

LAPORAN TUGAS KECIL I

STRATEGI ALGORITMA

PENYELESAIAN PERMAINAN KARTU 24 DENGAN ALGORITMA *BRUTE FORCE*

Kelas Mahasiswa K03

Dosen: Ir. Rila Mandala, M.Eng., Ph.D.

**Diajukan sebagai tugas kecil Mata Kuliah IF2211 Strategi Algoritma pada Semester II
Tahun Akademik 2022/2023**



Disusun Oleh:

Jason Rivalino (13521008)

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

2022

DAFTAR ISI

DAFTAR ISI	3
BAB I: DESKRIPSI MASALAH	4
BAB II: TEORI SINGKAT	5
I. Teori Singkat Algoritma <i>Brute Force</i>	5
BAB III: PENJELASAN ALGORITMA <i>BRUTE FORCE</i> & SOURCE PROGRAM YANG DIBUAT	6
I. Penjelasan Algoritma Program yang Dibuat.....	6
a) Splash.cpp	6
b) Inputing.cpp	7
c) Processing.cpp	8
d) Operation.cpp.....	10
e) Save.cpp	11
f) Main.cpp	11
BAB IV: EKSPERIMEN	14
I. Hasil Running Program.....	14
II. Uji Coba Hasil Program.....	16
BAB V: KESIMPULAN	19
REFERENSI	20
LAMPIRAN	21

BAB I

DESKRIPSI MASALAH

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.

BAB II

TEORI SINGKAT

I. Teori Singkat Algoritma *Brute Force*

Algoritma *Brute Force* merupakan algoritma yang memiliki pendekatan yang lempang (straightforward) untuk memecahkan suatu persoalan. Algoritma ini biasanya didasarkan pada pernyataan dalam persoalan (*problem statement*) dan juga definisi atau konsep yang dilibatkan. Algoritma *Brute Force* ini dapat memecahkan permasalahan dengan sangat sederhana, langsung, dan dengan cara yang jelas.

Beberapa contoh dari penerapan Algoritma *Brute Force* ini adalah sebagai berikut:

1. Berdasarkan pernyataan persoalan
 - Melakukan pencarian nilai elemen terbesar atau terkecil
 - Melakukan pencarian beruntun (*sequential search*)
2. Berdasarkan definisi/konsep yang terlibat
 - Menghitung a^n
 - Menghitung $n!$
 - Melakukan perkalian dua buah matriks berukuran $n \times n$
 - Menguji bilangan prima
 - Algoritma pengurutan (*bubble sort* dan *selection sort*)
 - Evaluasi polinom
 - Pencocokan string
 - Mencari persoalan titik dengan jarak terdekat

Algoritma *Brute Force* memiliki karakteristik antara lain tidak cerdas dan mangkus karena algoritma ini membutuhkan volume komputasi yang besar dan waktu yang lama untuk proses penyelesaiannya. Selain itu, algoritma ini lebih cocok untuk persoalan dengan ukuran masukan yang kecil. Terakhir, algoritma *Brute Force* dapat dipergunakan untuk menyelesaikan hampir semua persoalan.

BAB III

PENJELASAN ALGORITMA *BRUTE FORCE* & SOURCE PROGRAM YANG DIBUAT

I. Penjelasan Algoritma Program yang Dibuat

Pada program yang saya buat, program terdiri dari beberapa file yang tersusun atas satu file utama (*main file*) yaitu main.cpp dan terdapat beberapa file yang terdiri dari berbagai macam fungsi untuk menjalankan program utamanya antara lain splash.cpp, inputting.cpp, processing.cpp, operation.cpp, dan save.cpp.

a) Splash.cpp

Program pada file ini berfungsi untuk menampilkan splash screen saat *opening* dan *closing* dari program yang dijalankan.

```

1 #include <iostream>
2 #include <cstdlib>
3
4 using namespace std;
5
6 // Splash Screen Opening
7 void splashScreenOpening(){
8     cout << endl;
9     cout << "-----" << endl;
10    cout << "                                     " << endl;
11    cout << " | _||_/_| /_/_|_/_/_/_/_/_/_/_/_/_| " << endl;
12    cout << " | /_/_/_/_|'_/_/_/_/_/_/_/_/_/_/_| " << endl;
13    cout << " | /_/_/_/_| /_/_/_/_/_/_/_/_/_/_/_| " << endl;
14    cout << " |/_/_/_/_| |_/_/_/_/_| |_/_|_/_/_/_/_/_| " << endl;
15    cout << "                                     " << endl;
16    cout << "-----" << endl;
17    cout << endl;
18    cout << "===== " << endl;
19    cout << "           || " << endl;
20    cout << "           ||   PROGRAM INI DIBUAT OLEH   || " << endl;
21    cout << "           ||       JASON RIVALINO        || " << endl;
22    cout << "           ||   13521008 -- K03 JATINANGOR   || " << endl;
23    cout << "           ||                               || " << endl;
24    cout << "===== " << endl;
25    cout << endl;
26    cout << ("-----") << endl;
27    cout << ("Selamat datang di program permainan kartu 24") << endl;
28    cout << ("-----") << endl;
29    cout << endl;
30 }
31
32 // Splash Screen Closing
33 void splashScreenClosing(){
34     cout << endl;
35     cout << "-----" << endl;
36     cout << " |_____| ( ) |_____| ( ) |__| " << endl;
37     cout << " | |_____| |_____| |_____| |_____| | | | " << endl;
38     cout << " | |_____| |_____| |_____| |_____| | | | " << endl;
39     cout << " | |_____| |_____| |_____| |_____| | | | " << endl;
40     cout << " | |_____| |_____| |_____| |_____| | | | " << endl;
41     cout << " | |_____| |_____| |_____| |_____| | | | " << endl;
42     cout << " | |_____| |_____| |_____| |_____| | | | " << endl;
43     cout << "-----" << endl;
44     cout << endl;
45     cout << ("-----") << endl;
46     cout << ("Terima kasih telah menggunakan program ini !!") << endl;
47     cout << ("-----") << endl;
48     cout << endl;
49 }

```

IF2211

Strategi Algoritma

b) Inputting.cpp

Program pada file ini berfungsi untuk melakukan proses input angka kartu yang selanjutnya akan dicari nilai 24nya. Pada program input ini, terdapat fungsi untuk input angka secara manual dan juga fungsi untuk input angka secara acak. Serta ada juga fungsi untuk menentukan apakah input valid atau tidak.

Berikut merupakan fungsi untuk melakukan pengecekan validasi pada input program.

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <string>
4  #include <vector>
5  #include <ctime>
6  #include <climits>
7  using namespace std;
8
9  string num1, num2, num3, num4;
10 int num1Int, num2Int, num3Int, num4Int;
11 int cards[4];
12
13 // Fungsi untuk mengecek apakah inputan valid atau tidak
14 bool checkingCard(string card) {
15     return (card == "A") || (card == "2") || (card == "3") || (card == "4") || (card == "5") || (card == "6") ||
16            (card == "7") || (card == "8") || (card == "9") || (card == "10") || (card == "J") || (card == "Q") || (card == "K");
17 }
18
```

Fungsi untuk melakukan input secara manual adalah sebagai berikut:

```
19 // Fungsi opsi pertama untuk menginput angka secara manual
20 void inputNumber() {
21     cout << "-----" << endl;
22     cout << "Masukkan angka-angka kartu yang ingin dicari nilainya" << endl;
23     cout << "Format input: angka1 angka2 angka3 angka4" << endl;
24     cout << "Contoh: A 2 3 4" << endl;
25     cout << "-----" << endl;
26     cout << endl;
27     cout << "Input kartu: ";
28     cin >> num1 >> num2 >> num3 >> num4;
29     while (!(checkingCard(num1) && checkingCard(num2) && checkingCard(num3) && checkingCard(num4))) {
30         cout << "Input kartu: ";
31         cin >> num1 >> num2 >> num3 >> num4;
32     }
33     if (num1 == "A"){
34         num1 = "1";
35     }
36     if (num2 == "A"){
37         num2 = "1";
38     }
39     if (num3 == "A"){
40         num3 = "1";
41     }
42     if (num4 == "A"){
43         num4 = "1";
44     }
45     if (num1 == "J"){
46         num1 = "11";
47     }
48     if (num2 == "J"){
49         num2 = "11";
50     }
51     if (num3 == "J"){
52         num3 = "11";
53     }
54     if (num4 == "J"){
55         num4 = "11";
56     }
57     if (num1 == "Q"){
58         num1 = "12";
59     }
60     if (num2 == "Q"){
61         num2 = "12";
62     }
63     if (num3 == "Q"){
64         num3 = "12";
65     }
66     if (num4 == "Q"){
67         num4 = "12";
68     }
69 }
```

```
57     if (num1 == "K"){
58         num1 = "13";
59     }
60     if (num2 == "K"){
61         num2 = "13";
62     }
63     if (num3 == "K"){
64         num3 = "13";
65     }
66     if (num4 == "K"){
67         num4 = "13";
68     }
69     num1Int = stoi(num1);
70     num2Int = stoi(num2);
71     num3Int = stoi(num3);
72     num4Int = stoi(num4);
73
74     cout << "Angka-angka yang akan dicari hasilnya adalah: ";
75     cout << num1Int << " " << num2Int << " " << num3Int << " " << num4Int << endl;
76     cards[0] = num1Int;
77     cards[1] = num2Int;
78     cards[2] = num3Int;
79     cards[3] = num4Int;
80     // cout << cards[0]+cards[1]+cards[2]+cards[3] << endl; (Bwat ngecek bentuk udah integer apa belum)
81 }
```

Fungsi untuk input secara random adalah sebagai berikut:

```
80 // Fungsi opsi kedua untuk menginput angka secara random
81 void random(){
82     srand(time(NULL));
83     num1Int = rand() % 13 + 1;
84     num2Int = rand() % 13 + 1;
85     num3Int = rand() % 13 + 1;
86     num4Int = rand() % 13 + 1;
87     cout << "Angka-angka yang akan dicari hasilnya adalah: ";
88     cout << num1Int << " " << num2Int << " " << num3Int << " " << num4Int << endl;
89     cards[0] = num1Int;
90     cards[1] = num2Int;
91     cards[2] = num3Int;
92     cards[3] = num4Int;
93     // cout << cards[0]+cards[1]+cards[2]+cards[3] << endl; (Bwat ngecek bentuk udah integer apa belum)
94 }
```

c) Processing.cpp

Program pada file ini berfungsi untuk memproses angka-angka yang telah diinput sebelumnya untuk menghasilkan nilai 24. Fungsi yang terdapat pada program ini yaitu fungsi untuk melakukan pertukaran dan untuk melakukan permutasi untuk memasukkan kombinasi angka-angka yang dapat dibentuk kedalam bentuk array. Selain itu, terdapat fungsi untuk melakukan pengecekan secara *Brute Force* untuk menemukan segala kemungkinan pembentukan angka 24 dari angka dan operasi yang ada.

Berikut merupakan fungsi untuk melakukan pertukaran dan permutasi:


```
1  #include <iostream>
2  #include <cstdlib>
3  #include <string>
4  #include <vector>
5  #include <fstream>
6  #include <sstream>
7  #include <ctime>
8  #include <climits>
9  #include "inputing.cpp"
10 #include "operation.cpp"
11 using namespace std;
12
13
14 string hasilOperasi[1000]; // Untuk menyimpan hasil akhir operasi
15 int nums1[25], nums2[25], nums3[25], nums4[25]; // Untuk menyimpan angka-angka yang akan dicari hasilnya
16 int neff = 0;
17 int counts = 0;
18
19 // Fungsi untuk melakukan pertukaran
20 void swap (int *num1, int *num2){
21     int temp = *num1;
22     *num1 = *num2;
23     *num2 = temp;
24 }
25
26 // Fungsi untuk melakukan permutasi dan menyimpan hasilnya ke dalam array
27 void permutation (int *cards, int start, int end){
28     if (start == end){
29         neff++;
30         nums1[neff] = cards[0];
31         nums2[neff] = cards[1];
32         nums3[neff] = cards[2];
33         nums4[neff] = cards[3];
34     }
35     else{
36         for (int i = start; i <= end; i++){
37             swap((cards+start), (cards+i));
38             permutation(cards, start+1, end);
39             swap((cards+start), (cards+i));
40         }
41     }
42 }
43
```

Untuk fungsi untuk mencari operasi yang menghasilkan nilai 24 dengan menggunakan algoritma *Brute Force*, fungsi akan mencoba semua kemungkinan urutan angka dan operasi untuk dapat memperoleh nilai 24. Berikut merupakan programnya:

```
44 // Fungsi untuk mencari solusi dengan menggunakan Brute Force
45 void search24 (int a, int b, int c, int d){
46     int i, j, k;
47     float calc1, calc2;
48     float numHasil;
49     for (i = 1; i <= 4; i++){
50         for (j = 1; j <= 4; j++){
51             for (k = 1; k <= 4; k++){
52
53                 // Pengecekan segala kemungkinan operasi yang dapat dilakukan
54
55                 // ((w op x) op (y op z))
56                 calc1= calculation((float)a, (float)b, i);
57                 calc2= calculation((float)c, (float)d, k);
58                 numHasil= calculation(calc1, calc2, j);
59                 if (numHasil == 24){
60                     counts++;
61                     hasilOperasi[counts] = "(" + to_string(a) + " " + op(i) + " " + to_string(b) + " " + op(j) + " (" + to_string(c) + " " + op(k) + " " + to_string(d) + ")";
62                     cout << hasilOperasi[counts] << endl;
63                 }
64
65                 // ((w op x) op y) op z
66                 calc1= calculation((float)a, (float)b, i);
67                 calc2= calculation(calc1, (float)c, j);
68                 numHasil= calculation(calc2, (float)d, k);
69                 if (numHasil == 24){
70                     counts++;
71                     hasilOperasi[counts] = "(" + to_string(a) + " " + op(i) + " " + to_string(b) + " " + op(j) + " " + to_string(c) + " " + op(k) + " " + to_string(d);
72                     cout << hasilOperasi[counts] << endl;
73                 }
74             }
75         }
76     }
77 }
```

```

74 // (w op (x op y)) op z
75 calc1= calculation((float)b, (float)c, j);
76 calc2= calculation((float)a, calc1, i);
77 numHasil= calculation(calc2, (float)d, k);
78 if (numHasil == 24){
79     counts++;
80     hasilOperasi[counts] = "(" + to_string(a) + " " + op(i) + " (" + to_string(b) + " " + op(j) + " " + to_string(c) + ") " + op(k) + " " + to_string(d);
81     cout << hasilOperasi[counts] << endl;
82 }
83
84 // w op ((x op y) op z)
85 calc1= calculation((float)b, (float)c, j);
86 calc2= calculation(calc1, (float)d, k);
87 numHasil= calculation((float)a, calc2, i);
88 if (numHasil == 24){
89     counts++;
90     hasilOperasi[counts] = to_string(a) + " " + op(i) + " (" + to_string(b) + " " + op(j) + " " + to_string(c) + ") " + op(k) + " " + to_string(d);
91     cout << hasilOperasi[counts] << endl;
92 }
93
94 // w op (x op (y op z))
95 calc1= calculation((float)c, (float)d, k);
96 calc2= calculation((float)b, calc1, j);
97 numHasil= calculation((float)a, calc2, i);
98 if (numHasil == 24){
99     counts++;
100     hasilOperasi[counts] = to_string(a) + " " + op(i) + " (" + to_string(b) + " " + op(j) + " (" + to_string(c) + " " + op(k) + " " + to_string(d) + ") " + op(k) + " " + to_string(d);
101     cout << hasilOperasi[counts] << endl;
102 }
103 }
104 }
105 }
106 }
107 }

```

d) Operation.cpp

Program pada file ini berfungsi untuk menyimpan operasi-operasi yang akan dilakukan untuk membentuk angka 24. Operasi-operasi yang ada antara lain operasi pertambahan, pengurangan, perkalian, dan pembagian. Untuk programnya adalah sebagai berikut:

```

1  #include <iostream>
2  #include <cstdlib>
3  #include <string>
4  using namespace std;
5
6  // Fungsi untuk kalkulasi operasi
7  double calculation (double num1Int, double num2Int, int operators){
8      double hasil;
9      if (operators == 1){
10         hasil = num1Int + num2Int;
11     }
12     if (operators == 2){
13         hasil = num1Int - num2Int;
14     }
15     if (operators == 3){
16         hasil = num1Int * num2Int;
17     }
18     if (operators == 4){
19         hasil = num1Int / num2Int;
20     }
21     return hasil;
22 }
23
24 // Fungsi untuk menentukan operator
25 string op (int operators){
26     string opr;
27     if (operators == 1){
28         opr = "+";
29     }
30     if (operators == 2){
31         opr = "-";
32     }
33     if (operators == 3){
34         opr = "*";
35     }
36     if (operators == 4){
37         opr = "/";
38     }
39     return opr;
40 }

```

e) Save.cpp

Program pada file ini digunakan untuk melakukan penyimpanan output program yang dihasilkan dalam bentuk file.txt. Programnya adalah sebagai berikut:

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <string>
4  #include <vector>
5  #include <fstream>
6  #include <sstream>
7  #include "processing.cpp"
8  using namespace std;
9
10
11 // Fungsi untuk menyimpan hasil operasi ke dalam file
12 void saveFile(){
13     string namaFile;
14     cout << "Masukkan nama file: ";
15     cin >> namaFile;
16     ofstream myfile("../test/" + namaFile + ".txt");
17     myfile << "Angka-angka yang akan dicari hasilnya adalah: ";
18     myfile << num1Int << " " << num2Int << " " << num3Int << " " << num4Int << endl;
19     myfile << endl;
20     myfile << "Solusi yang didapat adalah: " << endl;
21     for (int i = 1; i <= counts; i++){
22         myfile << hasilOperasi[i] << endl;
23     }
24     myfile << endl;
25     if (counts == 0){
26         myfile << "No Solutions Found" << endl;
27     }
28     else if (counts == 1){
29         myfile << "1 Solution Found" << endl;
30     }
31     else{
32         myfile << counts << " Solutions Found" << endl;
33     }
34 }
```

f) Main.cpp

Program ini merupakan program utama yang berfungsi untuk menjalankan program-program yang telah dibuat pada file lainnya. Urutan proses programnya adalah program akan menampilkan splash screen pembuka dan menu untuk menanyakan kepada *user* mengenai input angka kartu pada program. Pengguna disini memiliki dua pilihan. Jika memilih manual, program akan masuk ke fungsi inputNumber pada file inputing. Begitupun jika memilih random, pengguna akan masuk dalam fungsi random. Setelah itu, program akan menjalankan proses pencarian solusi dengan menggunakan permutasi dari file processing dan kemudian menampilkan solusi-solusi apa saja yang didapat beserta jumlah solusinya dan waktu eksekusinya. Setelah itu, pengguna akan diberikan opsi untuk menyimpan hasil perhitungan dalam bentuk file.txt. Jika pengguna memilih “y”, program akan

IF2211

Strategi Algoritma

menjalankan fungsi savefile pada file save dan kemudian program menampilkan splash screen penutup. Untuk programnya adalah sebagai berikut:

```
1  #include <iostream>
2  #include <cstdlib>
3  #include "splash.cpp"
4  #include "save.cpp"
5  using namespace std;
6
7  int main(){
8      // Menampilkan splash screen dan menu pemilihan metode
9      splashScreenOpening();
10     cout << ("Silahkan memilih metode yang ingin digunakan: ") << endl;
11     cout << ("1. Memasukkan angka kartu sendiri secara manual") << endl;
12     cout << ("2. Memilih angka kartu secara acak (Randomize Number)") << endl;
13     cout << ("Masukkan pilihan anda: ");
14     int pilihan;
15     cin >> pilihan;
16     while (pilihan != 1 && pilihan != 2){
17         cout << ("Masukkan pilihan anda: ");
18         cin >> pilihan;
19     }
20
21     // Kondisi jika memilih metode manual
22     if (pilihan == 1){
23         cout << endl;
24         inputNumber();
25     }
26
27     // Kondisi jika memilih metode random
28     else if (pilihan == 2){
29         cout << endl;
30         random();
31     }
32
33     cout << endl;
34
35     // Melakukan proses pencarian solusi
36     cout << "Proses pencarian solusi..." << endl;
37     clock_t start = clock();
38
39     // Mencari solusi dengan cara permutasi
40     permutation(cards, 0, 3);
41
42     // Mencetak solusi yang didapat
43     cout << endl << "Solusi yang didapat adalah: " << endl;
44     int i;
45     for (i = 1; i <= neff; i++){
46         search24(nums1[i], nums2[i], nums3[i], nums4[i]);
47     }
48 }
```

```
49 // Mencetak jumlah solusi yang dihasilkan
50 if (counts == 0){
51     cout << endl << "No Solutions Found" << endl;
52 }
53 else if (counts == 1){
54     cout << endl << "1 Solution Found" << endl;
55 }
56 else{
57     cout << endl << counts << " Solutions Found" << endl;
58 }
59
60 printf("Time Execution: %.3f seconds\n", (float)(clock() - start)/CLOCKS_PER_SEC);
61
62 cout << endl;
63 string choose;
64 cout << "-----" << endl;
65 cout << endl;
66
67 // Opsi untuk menyimpan hasil ke dalam file
68 cout << "Apakah anda ingin menyimpan hasil ke dalam file? (y/n): ";
69 cin >> choose;
70 while (choose != "y" && choose != "n"){
71     cout << "Apakah anda ingin menyimpan hasil ke dalam file? (y/n): ";
72     cin >> choose;
73 }
74
75 if (choose == "y"){
76     saveFile();
77 }
78
79 cout << endl;
80 splashScreenClosing();
81 return 0;
82 }
```

Algoritma *Brute Force* yang diterapkan dalam program saya ini terdapat pada bagian file processing yaitu fungsi search24. Pada fungsi ini, penerapan dari algoritma *Brute Force*nya terdapat pada algoritma yang diterapkan, yaitu algoritma yang mencoba segala kemungkinan hingga mendapatkan hasil akhir yang diharapkan. Pada program saya ini, semua kemungkinan yang ada dicoba yaitu mulai dari susunan angka yang berbeda-beda yaitu ada sebanyak $4! = 24$ susunan. Lalu ada susunan posisi operasi dan tanda kurung yaitu ada 5 susunan yang berbeda

BAB IV

EKSPERIMEN

I. Hasil Running Program

a) Splash Screen dan menu

```
-----  
  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  
  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  
  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||  
-----  
  ||  PROGRAM INI DIBUAT OLEH  || | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  ||  JASON RIVALINO          ||  
  ||  13521008 -- K03 JATINANGOR  ||  
  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  ||  
-----  
Selamat datang di program permainan kartu 24  
-----  
Silahkan memilih metode yang ingin digunakan:  
1. Memasukkan angka kartu sendiri secara manual  
2. Memilih angka kartu secara acak (Randomize Number)  
Masukkan pilihan anda:
```

b) Opsi input angka secara manual

```
-----  
Selamat datang di program permainan kartu 24  
-----  
Silahkan memilih metode yang ingin digunakan:  
1. Memasukkan angka kartu sendiri secara manual  
2. Memilih angka kartu secara acak (Randomize Number)  
Masukkan pilihan anda: 1  
-----  
Masukkan angka-angka kartu yang ingin dicari nilainya  
Format input: angka1 angka2 angka3 angka4  
Contoh: A 2 3 4  
-----  
Input kartu: 2 3 4 5  
Angka-angka yang akan dicari hasilnya adalah: 2 3 4 5
```

- c) Opsi input angka secara random

```
-----
Selamat datang di program permainan kartu 24
-----

Silahkan memilih metode yang ingin digunakan:
1. Memasukkan angka kartu sendiri secara manual
2. Memilih angka kartu secara acak (Randomize Number)
Masukkan pilihan anda: 2

Angka-angka yang akan dicari hasilnya adalah: 5 2 5 2
```

- d) Program mengeluarkan solusi pencarian nilai 24

```
-----
Masukkan angka-angka kartu yang ingin dicari nilainya
Format input: angka1 angka2 angka3 angka4
Contoh: A 2 3 4
-----

Input kartu: 7 K 5 J
Angka-angka yang akan dicari hasilnya adalah: 7 13 5 11

Proses pencarian solusi...

Solusi yang didapat adalah:
(7 + 5) * (13 - 11)
(13 - 11) * (5 + 7)
(13 - 11) * (7 + 5)
(5 + 7) * (13 - 11)

4 Solutions Found
Time Execution: 0.004 seconds
```

- e) Program melakukan opsi penyimpanan dalam bentuk file.txt

[illegible]

- f) File.txt yang berisi hasil program yang telah tersimpan

```
1  Angka-angka yang akan dicari hasilnya adalah: 7 11 5 13
2
3  Solusi yang didapat adalah:
4  (7 + 5) * (13 - 11)
5  (5 + 7) * (13 - 11)
6  (13 - 11) * (5 + 7)
7  (13 - 11) * (7 + 5)
8
9  4 Solutions Found
```

II. Uji Coba Hasil Program

- a) testprogram1.txt (7 J 5 K)

```
Input kartu: 7 J 5 K
Angka-angka yang akan dicari hasilnya adalah: 7 11 5 13

Proses pencarian solusi...

Solusi yang didapat adalah:
(7 + 5) * (13 - 11)
(5 + 7) * (13 - 11)
(13 - 11) * (5 + 7)
(13 - 11) * (7 + 5)

4 Solutions Found
Time Execution: 0.004 seconds

-----
1  Angka-angka yang akan dicari hasilnya adalah: 7 11 5 13
2
3  Solusi yang didapat adalah:
4  (7 + 5) * (13 - 11)
5  (5 + 7) * (13 - 11)
6  (13 - 11) * (5 + 7)
7  (13 - 11) * (7 + 5)
8
9  4 Solutions Found
```

- b) testprogram2.txt (2 3 4 5)

```
Input kartu: 2 3 4 5
Angka-angka yang akan dicari hasilnya adalah: 2 3 4 5

Proses pencarian solusi...

Solusi yang didapat adalah:
2 * ((3 + 4) + 5)
2 * (3 + (4 + 5))
2 * ((3 + 5) + 4)
2 * (3 + (5 + 4))
2 * ((4 + 3) + 5)
2 * (4 + (3 + 5))
2 * ((4 + 5) + 3)
2 * (4 + (5 + 3))
2 * ((5 + 4) + 3)
2 * (5 + (4 + 3))
2 * ((5 + 3) + 4)
2 * (5 + (3 + 4))
```


IF2211
Strategi Algoritma

```

((3 - 2) + 5) * 4
(3 - (2 - 5)) * 4
((3 + 4) + 5) * 2
(3 + (4 + 5)) * 2
((3 + 5) + 4) * 2
(3 + (5 + 4)) * 2
((3 + 5) - 2) * 4
(3 + (5 - 2)) * 4
4 * ((3 - 2) + 5)
4 * (3 - (2 - 5))
((4 + 3) + 5) * 2
(4 + (3 + 5)) * 2
4 * ((3 + 5) - 2)
4 * (3 + (5 - 2))
4 * ((5 - 2) + 3)
4 * (5 - (2 - 3))
((4 + 5) + 3) * 2
(4 + (5 + 3)) * 2
4 * ((5 + 3) - 2)
4 * (5 + (3 - 2))
((5 + 3) + 4) * 2
(5 + (3 + 4)) * 2
((5 + 3) - 2) * 4
(5 + (3 - 2)) * 4
((5 + 4) + 3) * 2
(5 + (4 + 3)) * 2
((5 - 2) + 3) * 4
(5 - (2 - 3)) * 4

40 Solutions Found
Time Execution: 0.010 seconds

```

```

1 Angka-angka yang akan dicari hasilnya adalah: 2 3 4 5
2
3 Solusi yang didapat adalah:
4 2 * ((3 + 4) + 5)
5 2 * (3 + (4 + 5))
6 2 * ((3 + 5) + 4)
7 2 * (3 + (5 + 4))
8 2 * ((4 + 3) + 5)
9 2 * (4 + (3 + 5))
10 2 * ((4 + 5) + 3)
11 2 * (4 + (5 + 3))
12 2 * ((5 + 4) + 3)
13 2 * (5 + (4 + 3))
14 2 * ((5 + 3) + 4)
15 2 * (5 + (3 + 4))
16 ((3 - 2) + 5) * 4
17 (3 - (2 - 5)) * 4
18 ((3 + 4) + 5) * 2
19 (3 + (4 + 5)) * 2
20 ((3 + 5) + 4) * 2
21 (3 + (5 + 4)) * 2
22 ((3 + 5) - 2) * 4
23 (3 + (5 - 2)) * 4
24 4 * ((3 - 2) + 5)
25 4 * (3 - (2 - 5))
26 ((4 + 3) + 5) * 2
27 (4 + (3 + 5)) * 2
28 4 * ((3 + 5) - 2)
29 4 * (3 + (5 - 2))
30 4 * ((5 - 2) + 3)
31 4 * (5 - (2 - 3))
32 ((4 + 5) + 3) * 2
33 (4 + (5 + 3)) * 2
34 4 * ((5 + 3) - 2)
35 4 * (5 + (3 - 2))
36 ((5 + 3) + 4) * 2
37 (5 + (3 + 4)) * 2
38 ((5 + 3) - 2) * 4
39 (5 + (3 - 2)) * 4
40 ((5 + 4) + 3) * 2
41 (5 + (4 + 3)) * 2
42 ((5 - 2) + 3) * 4
43 (5 - (2 - 3)) * 4
44
45 40 Solutions Found

```

c) testprogram3.txt (A 8 9 Q)

```

Solusi yang didapat adalah:
((1 - 8) + 9) * 12
(1 - (8 - 9)) * 12
(1 * 8) * (12 - 9)
1 * (8 * (12 - 9))
((1 + 9) - 8) * 12
(1 + (9 - 8)) * 12
((1 * 12) - 9) * 8
(1 * (12 - 9)) * 8
1 * ((12 - 9) * 8)
(8 * 1) * (12 - 9)
8 * ((1 * 12) - 9)
8 * (1 * (12 - 9))
(8 / 1) * (12 - 9)
8 / (1 / (12 - 9))
(8 * (12 - 9)) * 1
8 * ((12 - 9) * 1)
8 * (12 - (9 * 1))
(8 * (12 - 9)) / 1
8 * ((12 - 9) / 1)
8 * (12 - (9 / 1))
8 * (12 - (1 * 9))
8 * ((12 * 1) - 9)
8 * ((12 / 1) - 9)

```

```

((9 - 8) + 1) * 12
(9 - (8 - 1)) * 12
((9 + 1) - 8) * 12
(9 + (1 - 8)) * 12
(12 - 9) * (8 * 1)
((12 - 9) * 8) * 1
(12 - 9) * (8 / 1)
((12 - 9) * 8) / 1
12 * ((9 - 8) + 1)
12 * (9 - (8 - 1))
(12 - 9) * (1 * 8)
((12 - 9) * 1) * 8
(12 - (9 * 1)) * 8
((12 - 9) / 1) * 8
(12 - (9 / 1)) * 8
(12 - 9) / (1 / 8)
12 * ((9 + 1) - 8)
12 * (9 + (1 - 8))
(12 - (1 * 9)) * 8
12 * ((1 + 9) - 8)
12 * (1 + (9 - 8))
((12 * 1) - 9) * 8
((12 / 1) - 9) * 8
12 * ((1 - 8) + 9)
12 * (1 - (8 - 9))

48 Solutions Found

```

IF2211
Strategi Algoritma

d) testprogram4.txt (8 2 10 9)

```
Input kartu: 8 2 10 9
Angka-angka yang akan dicari hasilnya adalah: 8 2 10 9

Proses pencarian solusi...

Solusi yang didapat adalah:
8 * ((2 + 10) - 9)
8 * (2 + (10 - 9))
8 * ((2 - 9) + 10)
8 * (2 - (9 - 10))
8 * ((10 + 2) - 9)
8 * (10 + (2 - 9))
8 * ((10 - 9) + 2)
8 * (10 - (9 - 2))
((8 + 9) * 2) - 10
(2 * (8 + 9)) - 10
((2 + 10) - 9) * 8
(2 + (10 - 9)) * 8
((2 - 9) + 10) * 8
(2 - (9 - 10)) * 8
(2 * (9 + 8)) - 10
((10 + 2) - 9) * 8
(10 + (2 - 9)) * 8
((10 - 9) + 2) * 8
(10 - (9 - 2)) * 8
((9 + 8) * 2) - 10

20 Solutions Found
Time Execution: 0.008 seconds
```

```
1  Angka-angka yang akan dicari hasilnya adalah: 8 2 10 9
2
3  Solusi yang didapat adalah:
4  8 * ((2 + 10) - 9)
5  8 * (2 + (10 - 9))
6  8 * ((2 - 9) + 10)
7  8 * (2 - (9 - 10))
8  8 * ((10 + 2) - 9)
9  8 * (10 + (2 - 9))
10 8 * ((10 - 9) + 2)
11 8 * (10 - (9 - 2))
12 ((8 + 9) * 2) - 10
13 (2 * (8 + 9)) - 10
14 ((2 + 10) - 9) * 8
15 (2 + (10 - 9)) * 8
16 ((2 - 9) + 10) * 8
17 (2 - (9 - 10)) * 8
18 (2 * (9 + 8)) - 10
19 ((10 + 2) - 9) * 8
20 (10 + (2 - 9)) * 8
21 ((10 - 9) + 2) * 8
22 (10 - (9 - 2)) * 8
23 ((9 + 8) * 2) - 10
24
25 20 Solutions Found
26
```

e) testprogram5.txt (K K K K)

```
Input kartu: K K K K
Angka-angka yang akan dicari hasilnya adalah: 13 13 13 13

Proses pencarian solusi...

Solusi yang didapat adalah:

No Solutions Found
Time Execution: 0.003 seconds
```

```
1  Angka-angka yang akan dicari hasilnya adalah: 13 13 13 13
2
3  Solusi yang didapat adalah:
4
5  No Solutions Found
```

f) testprogram6.txt (J Q 2 4)

```
Input kartu: J Q 2 4
Angka-angka yang akan dicari hasilnya adalah: 11 12 2 4

Proses pencarian solusi...

Solusi yang didapat adalah:
((11 - 2) * 4) - 12
(4 * (11 - 2)) - 12

2 Solutions Found
Time Execution: 0.004 seconds
```

```
1  Angka-angka yang akan dicari hasilnya adalah: 11 12 2 4
2
3  Solusi yang didapat adalah:
4  ((11 - 2) * 4) - 12
5  (4 * (11 - 2)) - 12
6
7  2 Solutions Found
```

BAB V

KESIMPULAN

Kesimpulan yang didapat dari pengerjaan tugas kecil ini adalah sebagai berikut:

- 1) Algoritma *Brute Force* dapat dipergunakan untuk menyelesaikan hampir sebagian besar persoalan
- 2) Permainan kartu 24 dapat menggunakan algoritma *Brute Force* untuk mencari solusi dari permasalahannya

REFERENSI

[1] R. Munir (2022). Algoritma Brute Force Bagian 1 [Powerpoint Slides]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)

LAMPIRAN

Pranala Github: https://github.com/jasonrivalino/Tucil1_13521008

Checklist:

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	✓	