

SOLVER PERMAINAN 24 MENGGUNAKAN ALGORITMA BRUTE FORCE

LAPORAN TUGAS KECIL

Disusun untuk memenuhi salah satu tugas kecil

mata kuliah Strategi Algoritma

IF2211-03

Oleh

Raditya Naufal A.

13521022



**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
INSTITUT TEKNOLOGI BANDUNG
2022**

DAFTAR ISI

DAFTAR ISI	1
BAB I	2
DESKRIPSI MASALAH	2
BAB II	3
DASAR TEORI	3
BAB III	4
BAB IV	5
4.1. Fungsi get_card_value	5
4.2. Fungsi Check Equation	6
4.3. Fungsi main	8
BAB V	9
BAB VI	12
PENUTUP	12
LAMPIRAN	13

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

DASAR TEORI

Brute force algorithm merupakan sebuah algoritma yang digunakan untuk pemecahan masalah dengan cara melakukan semua kemungkinan yang bisa dilakukan. Metode ini dapat digunakan untuk berbagai hal, salah satunya adalah kriptografi. Dalam kriptografi algoritma ini dapat memecahkan kata sandi dengan mencoba semua kemungkinan yang ada. Dalam hal ini salah satu kekurangan yang dari brute force algorithm yaitu waktu operasi yang lama akan terbayar dengan mendapatkan solusi yang pasti.

Selain dalam kriptografi, algoritma ini sangat mungkin dikembangkan ke bidang lain seperti optimasi dan berbagai pemecahan masalah. Namun, penggunaan Algoritma ini tidak terlalu disarankan karena dapat menghabiskan sumber daya komputasi yang cukup besar.

BAB III

ANALISIS PERSOALAN

Permainan kartu 24 merupakan sebuah permainan yang menggunakan 4 buah kartu dan operasi matematika dasar untuk membuat kartu bernilai 24. Permainan ini dapat diselesaikan menggunakan algoritma brute force dengan cara mencoba semua kombinasi perhitungan. Langkah-langkah yang digunakan dalam pemecahan 24 dengan menggunakan brute force adalah sebagai berikut.

1. Buatlah array yang berisikan 4 kartu tersebut
2. Buatlah loop untuk membuat permutasi angka angka dari 4 kartu tersebut
3. Buatlah loop untuk menyisipkan operasi di setiap celah antara angka
4. Di loop terakhir cek apakah hasil dari operasi sesuai dengan target yaitu 24
5. Apabila sesuai urutan angka dan operasi tersebut merupakan 1 solusi dari permainan 24

BAB IV

IMPLEMENTASI FUNGSI

4.1. Fungsi `get_card_value`

Fungsi ini digunakan untuk mengubah kartu jack, queen, king, as menjadi value 11, 12, 13, 1.

```
int get_card_value(char card[]) {  
    if (strcmp(card, "j") == 0) return 11;  
    else if (strcmp(card, "J") == 0) return 11;  
    else if (strcmp(card, "q") == 0) return 12;  
    else if (strcmp(card, "Q") == 0) return 12;  
    else if (strcmp(card, "k") == 0) return 13;  
    else if (strcmp(card, "K") == 0) return 13;  
    else if (strcmp(card, "A") == 0) return 1;  
    else if (strcmp(card, "a") == 0) return 1;  
    else return atoi(card);  
}
```

4.2. Fungsi Check Equation

Fungsi ini menggunakan algoritma brute force untuk menemukan semua kombinasi yang mungkin dari kartu yang diberikan dan menampilkannya di terminal.

```
void check_equation(int a, int b, int c, int d) {
    int nums[] = {a, b, c, d};
    int i, j, k, l, cnt=0;
    char ops[] = {'+', '-', '*', '/'};
    clock_t start_time = clock();

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            if (j == i) continue;
            for (k = 0; k < 4; k++) {
                if (k == i || k == j) continue;
                l = 6 - i - j - k;
                int num1 = nums[i];
                int num2 = nums[j];
                int num3 = nums[k];
                int num4 = nums[l];

                for (int op1 = 0; op1 < 4; op1++) {
                    for (int op2 = 0; op2 < 4; op2++) {
                        for (int op3 = 0; op3 < 4; op3++) {
                            int result = num1;
                            if (op1 == 0) result += num2;
                            else if (op1 == 1) result -= num2;
                            else if (op1 == 2) result *= num2;
                            else if (op1 == 3) {
                                if (num2 != 0) result /= num2;
                                else continue;
                            }

                            if (op2 == 0) result += num3;
                            else if (op2 == 1) result -= num3;
                            else if (op2 == 2) result *= num3;
                            else if (op2 == 3) {
                                if (num3 != 0) result /= num3;
                                else continue;
                            }

                            if (op3 == 0) result += num4;
                            else if (op3 == 1) result -= num4;
                            else if (op3 == 2) result *= num4;
                            else if (op3 == 3) {
                                if (num4 != 0) result /= num4;
                                else continue;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        for (int op1 = 0; op1 < 4; op1++) {
            for (int op2 = 0; op2 < 4; op2++) {
                for (int op3 = 0; op3 < 4; op3++) {
                    int result = num1;
                    if (op1 == 0) result += num2;
                    else if (op1 == 1) result -= num2;
                    else if (op1 == 2) result *= num2;
                    else if (op1 == 3) {
                        if (num2 != 0) result /= num2;
                        else continue;
                    }

                    if (op2 == 0) result += num3;
                    else if (op2 == 1) result -= num3;
                    else if (op2 == 2) result *= num3;
                    else if (op2 == 3) {
                        if (num3 != 0) result /= num3;
                        else continue;
                    }

                    if (op3 == 0) result += num4;
                    else if (op3 == 1) result -= num4;
                    else if (op3 == 2) result *= num4;
                    else if (op3 == 3) {
                        if (num4 != 0) result /= num4;
                        else continue;
                    }
                }
            }
        }

        if (result == TARGET) {
            fprintf(fp, "({d %c %d) %c %d = %d\n", num1, ops[op1], num2, ops[op2], num3, ops[op3], num4, TARGET);
        }
    }
}

fprintf(fp, "Total solutions: %d\n", cnt);
fprintf(fp, "Time used: %f seconds\n", time_used);
fclose(fp);
printf("Solution saved to 24_solution.txt\n");
} else if (save == 'n') {
    printf("Solution not saved\n");
} else {
    printf("Invalid input\n");
}
}

```

```

        if (result == TARGET) {
            printf("{d %c %d) %c %d = %d\n", num1, ops[op1], num2, ops[op2], num3, ops[op3], num4, TARGET);
            cnt++;
        }
    }
}

if (cnt == 0) printf("No solution found.\n");
else printf("Total solutions: %d\n", cnt);
clock_t end_time = clock();
double time_used = (double)(end_time - start_time) / CLOCKS_PER_SEC;
printf("Time used: %f seconds\n", time_used);
printf("Do you want to save the solution to a text file? (y/n): ");
char save;
scanf(" %c", &save);
if (save == 'y') {
    FILE *fp;
    fp = fopen("../test/24_solution.txt", "w");
    if (fp == NULL) {
        printf("Error opening file!\n");
        exit(1);
    }

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            if (j == i) continue;
            for (k = 0; k < 4; k++) {
                if (k == i || k == j) continue;
                l = 6 - i - j - k;
                int num1 = nums[i];
                int num2 = nums[j];
                int num3 = nums[k];
                int num4 = nums[l];
            }
        }
    }
}

```


4.3. Fungsi main

Fungsi ini merupakan fungsi yang dijalankan saat program/executable di run.

```
int main() {
    int a, b, c, d, temp;
    char at, bt, ct, dt;
    srand(time(NULL));
    printf("choose your input\n");
    printf("1. input from keyboard\n");
    printf("2. random\n");
    printf("input: ");
    scanf("%d", &temp);
    printf("\n");
    if (temp == 1) {
        char card1[10], card2[10], card3[10], card4[10];
        printf("Enter four cards: ");
        scanf("%s %s %s %s", card1, card2, card3, card4);
        int a = get_card_value(card1);
        int b = get_card_value(card2);
        int c = get_card_value(card3);
        int d = get_card_value(card4);
        check_equation(a, b, c, d);
    } else if (temp == 2) {
        a = rand() % 13+1;
        b = rand() % 13+1;
        c = rand() % 13+1;
        d = rand() % 13+1;
        check_equation(a, b, c, d);
    } else {
        printf("invalid input");
        return 0;
    }
    return 0;
}
```

BAB V
EKSPERIMEN

5.1. Input Manual 2 2 2 2 (No Solution)

24 SOLVER

choose your input

1. input from keyboard

2. random

input: 1

Enter four cards: 2 2 2 2

No solution found.

Time used: 0.000053 seconds

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

Solution not saved

5.2. Input Manual a j 7 4 (Not Save)

24 SOLVER

choose your input

1. input from keyboard

2. random

input: 1

Enter four cards: a j 7 4

$((1 + 4) * 7) - 11 = 24$

$((4 + 1) * 7) - 11 = 24$

Total solutions: 2

Time used: 0.000030 seconds

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

Solution not saved

5.3. Input Manual q k 3 8 (Save)

```
choose your input
1. input from keyboard
2. random
input: 1

Enter four cards: q k 3 8
((12 / 13) + 3) * 8 = 24
((12 / 13) + 8) * 3 = 24
((12 + 3) * 13) / 8 = 24
((13 - 12) * 3) * 8 = 24
((13 / 12) * 3) * 8 = 24
((13 - 12) * 8) * 3 = 24
((13 / 12) * 8) * 3 = 24
((13 + 3) * 12) / 8 = 24
((13 * 3) / 12) * 8 = 24
((13 / 3) + 12) + 8 = 24
((13 + 3) / 8) * 12 = 24
((13 - 3) - 8) * 12 = 24
((13 / 3) + 8) + 12 = 24
((13 * 8) / 12) * 3 = 24
((13 - 8) - 3) * 12 = 24
((3 + 12) * 13) / 8 = 24
((3 + 13) * 12) / 8 = 24
((3 * 13) / 12) * 8 = 24
((3 + 13) / 8) * 12 = 24
((8 * 13) / 12) * 3 = 24
Total solutions: 20
Time used: 0.000131 seconds
Do you want to save the solution to a text file? (y/n): y
Solution saved to 24_solution.txt
```

```
test > 24_solution.txt
You, 3 seconds ago | 1 author (You)
1 ((12 / 13) + 3) * 8 = 24
2 ((12 / 13) + 8) * 3 = 24
3 ((12 + 3) * 13) / 8 = 24
4 ((13 - 12) * 3) * 8 = 24
5 ((13 / 12) * 3) * 8 = 24
6 ((13 - 12) * 8) * 3 = 24
7 ((13 / 12) * 8) * 3 = 24
8 ((13 + 3) * 12) / 8 = 24
9 ((13 * 3) / 12) * 8 = 24
10 ((13 / 3) + 12) + 8 = 24
11 ((13 + 3) / 8) * 12 = 24
12 ((13 - 3) - 8) * 12 = 24
13 ((13 / 3) + 8) + 12 = 24
14 ((13 * 8) / 12) * 3 = 24
15 ((13 - 8) - 3) * 12 = 24
16 ((3 + 12) * 13) / 8 = 24
17 ((3 + 13) * 12) / 8 = 24
18 ((3 * 13) / 12) * 8 = 24
19 ((3 + 13) / 8) * 12 = 24
20 ((8 * 13) / 12) * 3 = 24
21 Total solutions: 20
22 Time used: 0.000131 seconds
23
```

5.4. Random Input (Not save)

```
choose your input
1. input from keyboard
2. random
input: 2

((11 * 12) / 4) - 9 = 24
((9 - 11) + 4) * 12 = 24
((9 * 12) - 11) / 4 = 24
((9 + 4) - 11) * 12 = 24
((12 * 11) / 4) - 9 = 24
((12 * 9) - 11) / 4 = 24
((12 / 4) * 11) - 9 = 24
((4 - 11) + 9) * 12 = 24
((4 + 9) - 11) * 12 = 24
Total solutions: 9
Time used: 0.000073 seconds
Do you want to save the solution to a text file? (y/n): n
Solution not saved
```

5.5. Random Input (Save)

```
choose your input
1. input from keyboard
2. random
input: 2

((4 * 5) + 4) * 1 = 24
((4 * 5) + 4) / 1 = 24
((4 * 5) * 1) + 4 = 24
((4 * 5) / 1) + 4 = 24
((4 * 1) * 5) + 4 = 24
((4 / 1) * 5) + 4 = 24
((5 * 4) + 4) * 1 = 24
((5 * 4) + 4) / 1 = 24
((5 * 4) * 1) + 4 = 24
((5 * 4) / 1) + 4 = 24
((5 * 4) + 4) * 1 = 24
((5 * 4) + 4) / 1 = 24
((5 * 4) * 1) + 4 = 24
((5 * 4) / 1) + 4 = 24
((5 * 1) * 4) + 4 = 24
((5 / 1) * 4) + 4 = 24
((5 * 1) * 4) + 4 = 24
((5 / 1) * 4) + 4 = 24
((4 * 5) + 4) * 1 = 24
((4 * 5) + 4) / 1 = 24
((4 * 5) * 1) + 4 = 24
((4 * 5) / 1) + 4 = 24
((4 * 1) * 5) + 4 = 24
((4 / 1) * 5) + 4 = 24
((1 * 4) * 5) + 4 = 24
((1 * 5) * 4) + 4 = 24
((1 * 5) * 4) + 4 = 24
((1 * 4) * 5) + 4 = 24
Total solutions: 28
Time used: 0.000142 seconds
Do you want to save the solution to a text file? (y/n): y
Solution saved to 24_solution.txt
```

```
test > 24_solution.txt
You, 4 seconds ago | 1 author (You)
1 ((4 * 5) + 4) * 1 = 24
2 ((4 * 5) + 4) / 1 = 24
3 ((4 * 5) * 1) + 4 = 24
4 ((4 * 5) / 1) + 4 = 24
5 ((4 * 1) * 5) + 4 = 24
6 ((4 / 1) * 5) + 4 = 24
7 ((5 * 4) + 4) * 1 = 24
8 ((5 * 4) + 4) / 1 = 24
9 ((5 * 4) * 1) + 4 = 24
10 ((5 * 4) / 1) + 4 = 24
11 ((5 * 4) + 4) * 1 = 24
12 ((5 * 4) + 4) / 1 = 24
13 ((5 * 4) * 1) + 4 = 24
14 ((5 * 4) / 1) + 4 = 24
15 ((5 * 1) * 4) + 4 = 24
16 ((5 / 1) * 4) + 4 = 24
17 ((5 * 1) * 4) + 4 = 24
18 ((5 / 1) * 4) + 4 = 24
19 ((4 * 5) + 4) * 1 = 24
20 ((4 * 5) + 4) / 1 = 24
21 ((4 * 5) * 1) + 4 = 24
22 ((4 * 5) / 1) + 4 = 24
23 ((4 * 1) * 5) + 4 = 24
24 ((4 / 1) * 5) + 4 = 24
25 ((1 * 4) * 5) + 4 = 24
26 ((1 * 5) * 4) + 4 = 24
27 ((1 * 5) * 4) + 4 = 24
28 ((1 * 4) * 5) + 4 = 24
29 Total solutions: 28
30 Time used: 0.000142 seconds
31
```

5.6. Random Input (Save)

```
choose your input
1. input from keyboard
2. random
input: 2

((4 + 9) / 5) * 12 = 24
((4 * 5) / 9) * 12 = 24
((9 + 4) / 5) * 12 = 24
((12 / 4) * 5) + 9 = 24
((12 / 9) + 5) * 4 = 24
((12 * 5) / 4) + 9 = 24
((12 * 5) / 9) * 4 = 24
((5 * 4) / 9) * 12 = 24
((5 * 12) / 4) + 9 = 24
((5 * 12) / 9) * 4 = 24
Total solutions: 10
Time used: 0.000099 seconds
Do you want to save the solution to a text file? (y/n): y
Solution saved to 24_solution.txt
```

```
test > 24_solution.txt
You, 1 second ago | 1 author (You)
1 ((4 + 9) / 5) * 12 = 24
2 ((4 * 5) / 9) * 12 = 24
3 ((9 + 4) / 5) * 12 = 24
4 ((12 / 4) * 5) + 9 = 24
5 ((12 / 9) + 5) * 4 = 24
6 ((12 * 5) / 4) + 9 = 24
7 ((12 * 5) / 9) * 4 = 24
8 ((5 * 4) / 9) * 12 = 24
9 ((5 * 12) / 4) + 9 = 24
10 ((5 * 12) / 9) * 4 = 24
11 Total solutions: 10
12 Time used: 0.000099 seconds
13
```

BAB VI





PENUTUP

6.1. Kesimpulan

24 solver merupakan program yang digunakan untuk menyelesaikan permainan kartu 24. Dalam program ini 24 solver dibuat menggunakan algoritma brute force dan menampilkan semua kemungkinan yang bisa terjadi. Waktu kalkulasi bukan menjadi masalah utama seperti kebanyakan kode dengan algoritma brute force karena dalam hal ini hanya dilakukan sekitar 4^7 operasi sehingga tidak memakan waktu yang terlalu banyak.

LAMPIRAN

Link Repository Github : https://github.com/Raditss/TUCIL1_13521022

POIN	YA	TIDAK
Program berhasil dikompilasi tanpa kesalahan		
Program berhasil <i>running</i>		
Program dapat membaca input / generate sendiri dan memberikan luaran		
Solusi yang diberikan program memenuhi (berhasil mencapai 24)		
Program dapat menyimpan solusi dalam file teks	