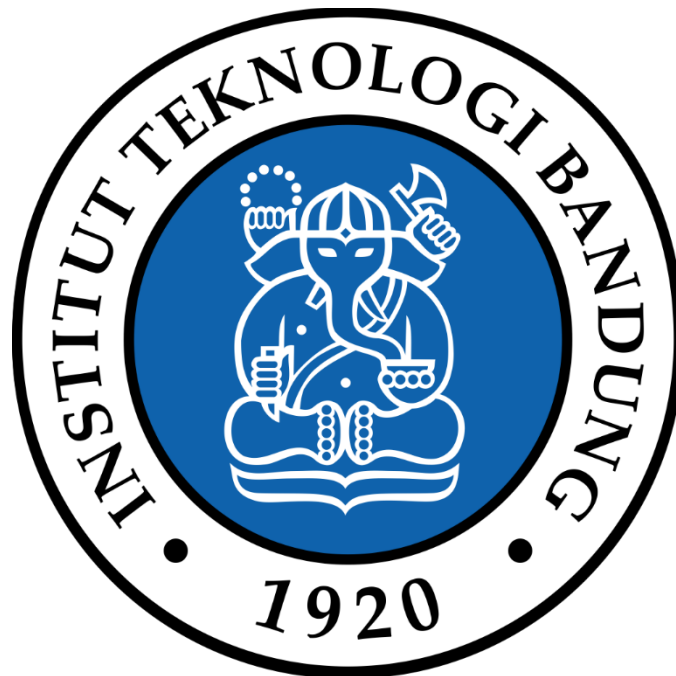


LAPORAN TUGAS KECIL 1
IF 2211 – STRATEGI ALGORITMA
SEMESTER II 2018-2019

Penyelesaian *24 Game* dengan Algoritma Brute Force

Disusun oleh

Jan Meyer Saragih / 13517131



PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

1. Deskripsi Program

a. Deskripsi Permainan 24

Permainan 24 adalah permainan di mana pemain menggunakan 4 buah angka dan mencoba mencari operasi aritmatika yang menghubungkan keempat angka tersebut sehingga terbentuk sebuah angka 24.

Operasi aritmatika yang dapat digunakan dalam permainan tersebut adalah tambah (+), kurang (-), kali (x), dan bagi (/). Setelah pemain mendapatkan keempat angka, maka pemain harus mencari operasi yang mungkin menggunakan keempat operator tersebut. Tentu saja operasi dari keempat angka tersebut haruslah kurang dari

b. Deskripsi Algoritma Brute Force

Dalam kesempatan ini, mata kuliah Strategi Algoritma menugaskan untuk mengaplikasikan strategi brute force untuk mencari setiap solusi 24 yang mungkin jika diberikan 4 bilangan bulat non-negatif. Strategi brute force adalah strategi di mana algoritma mengecek semua kemungkinan yang ada untuk memberikan solusi dari permasalahan.

Strategi brute force adalah strategi Strategi ini dinilai naif karena algoritma ini menyelesaikan persoalan langsung sesuai dengan permintaan serta langsung mengecek semua kemungkinan. Namun, ada beberapa persoalan yang hanya dapat diselesaikan menggunakan brute force dan penggunaan strategi ini dinilai lebih mudah dari strategi lain.

c. Deskripsi Logika Pengerjaan Program

Terdapat banyak kemungkinan yang ada menentukan operasi dari keempat angka tersebut untuk mendapatkan angka 24. Namun, pembagian kemungkinan tersebut dapat dibagi menjadi 3 tahap.

1. Kemungkinan susunan angka-angka dalam operasi
Dalam permainan 24, jumlah angka yang dapat digunakan dalam operasi haruslah 4. Maka dari itu, dengan menggunakan permutasi, terdapat $4! = 24$ kemungkinan susunan angka yang mungkin menghasilkan angka 24. Namun, ada kemungkinan terdapat duplikat.
2. Kemungkinan susunan operator-operator dalam operasi
Terdapat 4 buah operator yang dapat digunakan dalam permainan 24, yaitu tambah, kurang, kali, dan bagi. Karena terdapat 4 buah angka dalam permainan ini, maka terdapat 3 tempat di mana operator harus dimasukkan untuk menghasilkan operasi aritmatika, yaitu 1 di antara 2 angka.
Karena terdapat 4 operator dan 3 buah tempat yang tersedia, dengan aturan perkalian, maka kita mendapatkan $4 \times 4 \times 4 = 64$ buah kemungkinan susunan operator yang mungkin menghasilkan angka 24.
Namun, jika terdapat angka yang sama dari keempat angka tersebut, maka akan terdapat solusi yang sama dari solusi-solusi yang dihasilkan.
3. Kemungkinan penempatan tanda kurung dalam operasi
Penempatan tanda kurung dalam operasi aritmatika berfungsi untuk menentukan urutan operasi mana yang dikerjakan terlebih dahulu. Dari penempatan tanda kurung tersebut, terdapat 5 buah jenis solusi yang ada.
Jika dimisalkan salah satu susunan bilangan tersebut adalah a, b, c, dan d serta operasi yang digunakan adalah 3 operasi penjumlahan, maka susunan tanda kurung tersebut adalah sebagai berikut.

- a. $((a + b) + c) + d$
- b. $a + (b + (c + d))$
- c. $(a + b) + (c + d)$
- d. $(a + (b + c)) + d$
- e. $a + ((b + c) + d)$

Dengan ketiga poin di atas, maka dapat disimpulkan bahwa jumlah seluruh kemungkinan solusi dari program yang dibuat adalah $24 \times 64 \times 5 = 7.680$ buah kemungkinan solusi. Namun, hasilnya akan kurang dari 7.680 karena beberapa solusi tidak menghasilkan angka 24.

d. Deskripsi Langkah Kerja Program

1. Pertama-tama, keempat angka tersebut dibaca dengan menggunakan prosedur `readNumbers`, yang hanya berfungsi menerima input. Input yang dimasukkan haruslah 4 bilangan bulat non-negatif dipisahkan oleh koma.
2. Setelah mendapatkan keempat angka tersebut, maka dibuatlah kemungkinan susunan keempat angka tersebut di main program. Jumlah kemungkinan adalah $4! = 24$ kemungkinan susunan angka.
3. Lalu ke-24 kemungkinan ini dapat dimasukkan ke dalam array, yaitu array yang mengandung setiap kemungkinan susunan angka yang ada. Hal ini dilakukan dengan prosedur `AddToArray`. Namun, sebelum dimasukkan ke array, prosedur `AddToArray` mengecek terlebih dahulu jika susunan angka tersebut telah ada sebelumnya atau belum untuk menghindari duplikat.
4. Setiap kemungkinan susunan yang telah dimasukkan ke array kemudian dimasukkan ke prosedur `check24`. Prosedur `check24` ini adalah membuat setiap kemungkinan operator-operator aritmatika yang mungkin ditempatkan di antara susunan 4 angka tersebut. Jumlah kemungkinan yang mungkin untuk setiap susunan angka tersebut adalah 4^3 kemungkinan.
5. Setiap kemungkinan angka dan operator yang berada di `check24` itu dimasukkan ke prosedur `countResult`. Di dalam prosedur ini, setiap angka dan operator dihitung dengan 5 susunan tanda kurung yang berbeda. Tanda kurung merepresentasikan operasi yang dilakukan duluan. Di dalam tanda kurung terdapat 2 operan dan 1 operator. Karena proses operasi aritmatika menggunakan 2 buah operan dan 1 buah operator, maka digunakan fungsi `calculate`. Fungsi tersebut berfungsi memberikan hasil dari operasi dengan 2 operan dan 1 operator.
6. Hasil tersebut kemudian dicek. Jika hasil tersebut menghasilkan angka 24, maka hasil tersebut ditampilkan serta nilai counter, yang merupakan variabel global, ditambahkan. Dalam program yang dibuat, angka yang dicek tidak pas 24.
7. Setelah itu, program menampilkan jumlah solusi yang ada sesuai nilai counter dan menampilkan lamanya waktu eksekusi program.

2. Source Code

a. Asumsi Program

Asumsi yang dibuat untuk program ini adalah:

1. User memasukkan input angka dipisahkan oleh spasi.
2. Input yang dimasukkan haruslah 4 angka.
3. Input yang dimasukkan merupakan bilangan bulat non-negatif.
4. Jumlah kombinasi penempatan tanda kurung yang digunakan adalah 5 jenis tanda kurung. Sementara itu, jumlah kombinasi penempatan tanda kurung di spek adalah 4 buah.

5. Karena terdapat bilangan pecahan yang hasilnya harus dibulatkan oleh tipe data float, hasil perhitungan terkadang tidak pas 24. Contohnya adalah $3/24$. Maka dari itu, dibuat galat sebesar 0,00001 dari bilangan 24.

b. Source Code Program

Bahasa yang digunakan untuk menyelesaikan permainan 24 ini dengan cara brute force adalah bahasa C++. Source code yang digunakan adalah sebagai berikut.

```
/* NIM      : 13517131
   Nama     : Jan Meyer Saragih
   Mata Kuliah : Strategi Algoritma
   Kelas    : K-02 */

#include <iostream>    // Input dan output
#include <time.h>      // Menghitung waktu eksekusi
using namespace std;

// GLOBAL VARIABLES
float a, b, c, d;
int count = 0;

int e[25], f[25], g[25], h[25];
int Neff = 0;

// DECLARE VOIDS AND FUNCTIONS
void AddtoArray(int a, int b, int c, int d);
// Mencari tahu apakah susunan angka a, b, c, dan d terdapat dalam array
// Jika ada, maka angka tersebut tidak dimasukkan ke dalam array
// Jika tidak, maka angka tersebut dimasukkan ke dalam array dan Neff ditambah 1

void readNumbers();
// Membaca keempat angka untuk permainan 24

void check24 (int w, int x, int y, int z);
// Mengecek apakah hasilnya dapat mendapatkan angka 24 atau tidak

void countResult (int w, int x, int y, int z, int i, int j, int k);
// Menghitung hasil dari kombinasi angka dan operator yang direpresentasikan oleh angka
// w, x, y, z adalah keempat angka non negatif yang telah diinput
// i, j, k adalah operator-operator yang dihasilkan dari for di count24

float calculate (float a, float b, int sign);
// Menghitung operasi dengan 2 buah float dan 1 operator yang diwakili angka

char represent (int sign);
// Mengembalikan char yang direpresentasikan oleh angka
// Berfungsi untuk print solusi yang mungkin

// MAIN PROGRAM
int main()
{
    // KAMUS
    int i;

    // ALGORITMA

    // Input keempat angka
    cout << "TUGAS KECIL 1 STRATEGI ALGORITMA" << endl;
    cout << "Nama: Jan Meyer Saragih" << endl;
    cout << "NIM: 13517131" << endl;
    cout << "Kelas: K-02" << endl;
    cout << "Permainan 24" << endl << endl;
    readNumbers();

    // Mencari solusi
    cout << "Mencari solusi..." << endl;
    clock_t start = clock();

    // Karena terdapat 4 buah angka, maka terdapat 4! = 24 buah kemungkinan
    // susunan angka
    // Jika terdapat angka yang berbeda, maka terdapat solusi yang sama
    AddtoArray(a, b, c, d);
```

```

AddtoArray(a, b, d, c);
AddtoArray(a, c, b, d);
AddtoArray(a, c, d, b);
AddtoArray(a, d, b, c);
AddtoArray(a, d, c, b);
AddtoArray(b, a, c, d);
AddtoArray(b, a, d, c);
AddtoArray(b, c, a, d);
AddtoArray(b, c, d, a);
AddtoArray(b, d, a, c);
AddtoArray(b, d, c, a);
AddtoArray(c, a, b, d);
AddtoArray(c, a, d, b);
AddtoArray(c, b, a, d);
AddtoArray(c, b, d, a);
AddtoArray(c, d, a, b);
AddtoArray(c, d, b, a);
AddtoArray(d, a, b, c);
AddtoArray(d, a, c, b);
AddtoArray(d, b, a, c);
AddtoArray(d, b, c, a);
AddtoArray(d, c, a, b);
AddtoArray(d, c, b, a);

for (i = 1; i <= Neff; i++)
{
    check24(e[i], f[i], g[i], h[i]);
}
cout << endl << "Terdapat " << count << " buah solusi" << endl;

// Print lama waktu eksekusi
printf("Waktu eksekusi: %.5fs\n", (float)(clock() - start)/CLOCKS_PER_SEC);

return 0;
}

void readNumbers()
// Membaca keempat angka untuk permainan 24
{
    cout << "Masukkan keempat angka yang mungkin" << endl;
    cin >> a >> b >> c >> d;
    cout << "Angka yang diproses adalah..." << endl;
    cout << a << ", " << b << ", " << c << ", dan " << d << endl << endl;
}

void AddtoArray(int a, int b, int c, int d)
// Mencari tahu apakah susunan angka a, b, c, dan d terdapat dalam array
// Jika ada, maka angka tersebut tidak dimasukkan ke dalam array
// Jika tidak, maka angka tersebut dimasukkan ke dalam array dan Neff ditambah 1
{
    // Kamus lokal
    int i;
    bool found;

    // Algoritma
    found = false;
    i = 1;

    while (i <= Neff && not found)
    {
        if (a == e[i] && b == f[i] && c == g[i] && d == h[i])
        {
            found = true;
        }
        else
        {
            i++;
        }
    }
    if (not found)
    {
        Neff++;
        e[Neff] = a;
        f[Neff] = b;
        g[Neff] = c;
        h[Neff] = d;
    }
}

```

```

}

void check24 (int w, int x, int y, int z)
// Mengecek apakah hasilnya dapat mendapatkan angka 24 atau tidak
{
    // Berfungsi sebagai operator
    int i, j, k;

    // Terdapat 3 buah operator yang membatasi keempat angka karena
    // di antara 2 angka, terdapat 1 operator
    for (i = 1; i <= 4; i++)
    {
        for (j = 1; j <= 4; j++)
        {
            for (k = 1; k <= 4; k++)
            {
                countResult (w, x, y, z, i, j, k);
            }
        }
    }
}

void countResult (int w, int x, int y, int z, int i, int j, int k)
// Menghitung hasil dari kombinasi angka dan operator yang direpresentasikan oleh angka
// w, x, y, z adalah keempat angka non negatif yang telah diinput
// i, j, k adalah operator-operator yang dihasilkan dari for di count24
{
    // Float ini berfungsi menyimpan hasil perhitungan
    float a;

    // ((w + x) + y) + z
    a = calculate(calculate(calculate((float)w, (float)x, i), (float) y, j), (float)z, k);
    if (a > 23.99999 && a < 24.00001)
    {
        count++;
        cout << "(" << w << " " << represent(i) << " " << x << ") " <<
            represent(j) << " " << y << ") " << represent(k) << " " << z << endl;
    }
    // (w + x) + (y + z)
    a = calculate(calculate((float)w, (float)x, i), calculate((float) y, (float) z, k), j);
    if (a > 23.99999 && a < 24.00001)
    {
        count++;
        cout << "(" << w << " " << represent(i) << " " << x << ") " <<
            represent(j) << " (" << y << " " << represent(k) << " " << z << ")" << endl;
    }
    // (w + (x + y)) + z
    a = calculate(calculate((float)w, calculate((float)x, (float)y, j), i), (float) z, k);
    if (a > 23.99999 && a < 24.00001)
    {
        count++;
        cout << "(" << w << " " << represent(i) << " (" << x << " " << represent(j) <<
            " " << y << ") " << represent(k) << " " << z << endl;
    }
    // w + ((x + y) + z)
    a = calculate((float)w, calculate(calculate((float)x, (float)y, j), (float)z, k), i);
    if (a > 23.99999 && a < 24.00001)
    {
        count++;
        cout << w << " " << represent(i) << " (" << x << " " << represent(j) << " " <<
            y << ") " << represent(k) << " " << z << ")" << endl;
    }
    // w + (x + (y + z))
    a = calculate((float)w, calculate((float)x, calculate((float)y, (float)z, k), j), i);
    if (a > 23.99999 && a < 24.00001)
    {
        count++;
        cout << w << " " << represent(i) << " (" << x << " " << represent(j) << " (" <<
            y << " " << represent(k) << " " << z << ") " << endl;
    }
}

float calculate (float a, float b, int sign)
// Menghitung operasi dengan 2 buah float dan 1 operator yang diwakili angka
{
    if (sign == 1)
    {

```

```

        return (a + b);
    }
    else if (sign == 2)
    {
        return (a - b);
    }
    else if (sign == 3)
    {
        return (a * b);
    }
    else if (sign == 4)
    {
        return ((float)a / (float)b);
    }
}

char represent (int sign)
// Mengembalikan char yang direpresentasikan oleh angka
// Berfungsi untuk print solusi yang mungkin
{
    if (sign == 1)
    {
        return ('+');
    }
    if (sign == 2)
    {
        return ('-');
    }
    if (sign == 3)
    {
        return ('*');
    }
    if (sign == 4)
    {
        return ('/');
    }
}
}

```

c. Tabel Kompetensi

| | Poin | Ya | Tidak |
|---|------|----|-------|
| 1. Program berhasil dikompilasi | | ✓ | |
| 2. Program berhasil dieksekusi | | ✓ | |
| 3. Solusi 24 Game benar untuk semua data tes. | | ✓ | |

3. Test Case Program

a. Test Case 1

```

SourceCodeTucil1StrAlgo-13517131
TUGAS KECIL 1 STRATEGI ALGORITMA
Nama: Jan Meyer Saragih
NIM: 13517131
Kelas: K-02
Permainan 24

Masukkan keempat angka yang mungkin
1 1 1 1
Angka yang diproses adalah...
1, 1, 1, dan 1

Mencari solusi...

Terdapat 0 buah solusi
Waktu eksekusi: 0.00100s
Press any key to continue . . .

```

Test Case : Angka 1, 1, 1, dan 1.

Solusi : 0 buah

b. Test Case 2

```
SourceCodeTucil1StrAlgo-13517131
TUGAS KECIL 1 STRATEGI ALGORITMA
Nama: Jan Meyer Saragih
NIM: 13517131
Kelas: K-02
Permainan 24

Masukkan keempat angka yang mungkin
3 8 3 8
Angka yang diproses adalah...
3, 8, 3, dan 8

Mencari solusi...
8 / (3 - (8 / 3))

Terdapat 1 buah solusi
Waktu eksekusi: 0.00100s

Press any key to continue . . .
```

Test Case : Angka 3, 8, 3, dan 8

Kesimpulan : Terdapat 1 solusi

c. Test Case 3

```
SourceCodeTucil1StrAlgo-13517131
TUGAS KECIL 1 STRATEGI ALGORITMA
Nama: Jan Meyer Saragih
NIM: 13517131
Kelas: K-02
Permainan 24

Masukkan keempat angka yang mungkin
6 6 6 6
Angka yang diproses adalah...
6, 6, 6, dan 6

Mencari solusi...
((6 + 6) + 6) + 6
(6 + 6) + (6 + 6)
(6 + (6 + 6)) + 6
6 + ((6 + 6) + 6)
6 + (6 + (6 + 6))
(6 * 6) - (6 + 6)
((6 * 6) - 6) - 6

Terdapat 7 buah solusi
Waktu eksekusi: 0.00800s

Press any key to continue . . .
```

Test Case : Angka 6, 6, 6, dan 6

Kesimpulan : Terdapat 7 solusi

d. Test Case 4

```
SourceCodeTucil1StrAlgo-13517131
Masukkan keempat angka yang mungkin
12 5 2 6
Angka yang diproses adalah...
12, 5, 2, dan 6

Mencari solusi...
((12 * 5) / 2) - 6
(12 * (5 / 2)) - 6
(12 / (5 - 2)) * 6
12 / ((5 - 2) / 6)
12 * (5 - (6 / 2))
((12 / 2) * 5) - 6
(12 / (2 / 5)) - 6
(12 * 2) * (6 - 5)
12 * (2 * (6 - 5))
(12 * 2) / (6 - 5)
12 * (2 / (6 - 5))
(12 * (6 - 5)) * 2
12 * ((6 - 5) * 2)
(12 * 6) / (5 - 2)
12 * (6 / (5 - 2))
(12 / (6 - 5)) * 2
12 / ((6 - 5) / 2)
((5 * 12) / 2) - 6
(5 * (12 / 2)) - 6
((5 / 2) * 12) - 6
(5 / (2 / 12)) - 6
(5 * 6) - (12 / 2)
(5 - (6 / 2)) * 12
(2 * 12) * (6 - 5)
2 * (12 * (6 - 5))
(2 * 12) / (6 - 5)
2 * (12 / (6 - 5))
(2 * (6 - 5)) * 12
2 * ((6 - 5) * 12)
(2 / (6 - 5)) * 12
2 / ((6 - 5) / 12)
(6 * 12) / (5 - 2)
6 * (12 / (5 - 2))
((6 - 5) * 12) * 2
(6 - 5) * (12 * 2)
(6 * 5) - (12 / 2)
((6 - 5) * 2) * 12
(6 - 5) * (2 * 12)
(6 / (5 - 2)) * 12
6 / ((5 - 2) / 12)

Terdapat 40 buah solusi
Waktu eksekusi: 0.12800s
```

Test Case : 12, 5, 2, dan 6

Kesimpulan : Terdapat 40 solusi

e. Test Case 5

```

Select SourceCodeTucil1StrAlgo-13517131
TUGAS KECIL 1 STRATEGI ALGORITMA
Nama: Jan Meyer Saragih
NIM: 13517131
Kelas: K-02
Permainan 24

Masukkan keempat angka yang mungkin
1 2 3 4
Angka yang diproses adalah...
1, 2, 3, dan 4

Mencari solusi...
((1 + 2) + 3) * 4
(1 + (2 + 3)) * 4
((1 * 2) * 3) * 4
(1 * 2) * (3 * 4)
(1 * (2 * 3)) * 4
1 * ((2 * 3) * 4)
1 * (2 * (3 * 4))
((1 * 2) * 4) * 3
(1 * 2) * (4 * 3)
(1 * (2 * 4)) * 3
1 * ((2 * 4) * 3)
1 * (2 * (4 * 3))
((1 + 3) + 2) * 4
(1 + (3 + 2)) * 4
(1 + 3) * (2 + 4)
((1 * 3) * 2) * 4
(1 * 3) * (2 * 4)
(1 * (3 * 2)) * 4
1 * ((3 * 2) * 4)
1 * (3 * (2 * 4))
(1 + 3) * (4 + 2)
((1 * 3) * 4) * 2
(1 * 3) * (4 * 2)
(1 * (3 * 4)) * 2
1 * ((3 * 4) * 2)
1 * (3 * (4 * 2))
((1 * 4) * 2) * 3
(1 * 4) * (2 * 3)
(1 * (4 * 2)) * 3
1 * ((4 * 2) * 3)
1 * (4 * (2 * 3))
((1 * 4) * 3) * 2
(1 * 4) * (3 * 2)
(1 * (4 * 3)) * 2
1 * ((4 * 3) * 2)
1 * (4 * (3 * 2))

(4 * 3) * (2 * 1)
(4 * (3 * 2)) * 1
4 * ((3 * 2) * 1)
4 * (3 * (2 * 1))
((4 * 3) * 2) / 1
(4 * 3) * (2 / 1)
(4 * (3 * 2)) / 1
4 * ((3 * 2) / 1)
4 * (3 * (2 / 1))

Terdapat 242 buah solusi
Waktu eksekusi: 1.07200s
```

Test Case : 1, 2, 3, dan 4

Kesimpulan : Terdapat 242 solusi