

TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA
Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*

IF2211 STRATEGI ALGORITMA
Semester II Tahun Ajaran 2020/2021

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



Disusun oleh :

Ridho Daffasyah

13519038

TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021

I. ALGORITMA BRUTE FORCE

Di dalam tugas kali ini, penyelesaian harus menggunakan algoritma brute force. Berikut langkah-langkah dalam membuat program ini :

1. Langkah pertama yang harus dilakukan adalah kita harus menyimpan semua huruf yang ada di baris saat pembacaan file.txt.

Misal : SEND

MORE+

MONEY

Maka kita bisa menyimpan semua huruf tersebut seperti huruf[] =

[SENDMOREMONEY], **ingat hanya hurufnya saja.**

2. Kedua, pisah setiap huruf yang ada di operan ke masing-masing list operan. Misal : operan1[] = [SEND], operan2[] = [MORE], operan3[] = [MONEY].
3. Ketiga, kita harus memisahkan huruf-huruf yang *double* menjadi huruf unik. Misal : hurufunik[] = [SENDMORY].
4. Keempat, setelah kita mendapatkan huruf-huruf yang unik, maka kita membuat juga list integer yang *range* nya sama seperti list hurufunik diatas.
5. Lalu kita lakukan *random integer* bisa dengan permutasi, kombinasi ataupun manual dengan looping. Angka disini hanya berkisar 0-9. Misal : nilaihuruf[] = [98565438].
6. Permutasi yang dilakukan itu permutasi 10 dengan jumlah huruf unik dalam hal ini hurufunik[] = [SENDMORY]. Maka lakukan pengacakan integer sebanyak $P(10, \text{panjang}(\text{hurfunik}[]))$.
7. Setelah itu, lakukan penyocokan dengan kondisi yang diinginkan seperti SEND+MORE=MONEY, maka angka acak yang merepresentasikan SEND dan MORE jika ditambah itu sama dengan angka yang merepresentasikan MONEY. Misal : SEND = 9567, MORE = 1082, MONEY = 10649.
8. Jika ditemukan yang sesuai dengan kondisi, maka output solusi tersebut ke layar.

9567

1082+

10649

II. SOURCE CODE PROGRAM

```
/* NAMA      : RIDHO DAFFASYAH
   NIM       : 13519038
   KELAS    : 01

   TUGAS KECIL 1 - IF2211 STRATEGI ALGORITMA

   Penyelesaian Cryptarithmic dengan Algoritma Brute Force
*/

/* HEADER */
#include <iostream>
#include <fstream>
#include <string.h>
#include <math.h>
#include <time.h>
#include <cstring>
#include <vector>

using namespace std;

/* KAMUS LOKAL */
ifstream in_stream;
vector<string> list_operan;
vector<char> list_huruf_unik;
string str;
char semuahurufdioperan[10];
int panjang = 0;
int nilaihuruf[10] = {0};
int i, j, k, angka1, angka2, angka3, percobaan, solusi;

/* FUNGSI-FUNGSI */
int faktorial(int n){
    if(n == 0){
        return 1;
    }
    else {
```

```

        return n*faktorial(n - 1);
    }
}

/* FUNGSI UNTUK Mencari nilai permutasi */
int permutasi(int n, int r){
    return faktorial(n)/faktorial(n - r);
}

/* FUNGSI Untuk memasukkan nilai ke suatu huruf */
void masukinnilaikeHuruf(){
    int i = panjang - 1;
    nilaihuruf[i] = nilaihuruf[i] + 1;
    while (nilaihuruf[i] == 10){
        nilaihuruf[i] = 0;
        i = i - 1;
        nilaihuruf[i] = nilaihuruf[i] + 1;
    }
}

/* FUNGSI Untuk mengetahui nilai suatu huruf dari posisi nya*/
int posisi(char str[], char c){
    int i, panjangchar;
    panjangchar = strlen(str);
    for(i = 0; i < panjangchar; i++){
        if(str[i] == c){
            return nilaihuruf[i];
        }
    }
}

/* Untuk mengetahui posisi suatu huruf */
int posisihuruf(char str[], char c){
    for(int i = 0; i < strlen(str); i++){
        if(c == str[i]){
            return i;
        }
    }
}

```

```

/* FUNGSI UNTUK MEMISAHKAN HURUF-HURUF YANG UNIK */
void hanyaunik(char str[])
{
    int i, j, panjangchar;
    panjangchar = strlen(str);
    for(i = 0; i < panjangchar; i++){
        for(j = panjang - 1; j >= 0; j--){
            if(semuahurufdioperan[j] == str[i]){
                break;
            }
        }
        if(j == -1){
            semuahurufdioperan[panjang] = str[i];
            panjang = panjang + 1;
        }
    }
}

/* PROGRAM UTAMA */
int main(){

    /* MEMBUKA FILE EKSTERNAL */
    in_stream.open("../test/soal.txt");

    /* MEMBACA ISI FILE EKSTERNAL */
    while (!in_stream.eof()){
        in_stream >> str;
        /* MENGISI LIST DENGAN CHAR DARI BELAKANG */
        list_operan.push_back(str);
    }
    list_operan.pop_back();

    if (list_operan.size()>4){
        exit(3);
        return 3;
    }

    /* MENAMPILKAN ISI TEXT */

```

```

        if (list_operan[0].size() < list_operan[1].size()){
            for (i = list_operan[0].size(); i < list_operan[1].size() -
1; i++){
                cout << " ";
            }
            cout << list_operan[0] << "\n";
            for (i = 1; i <= list_operan.size(); i++){
                cout << list_operan[i];
                cout << "\n";
            }
        }
        else{
            cout << list_operan[0] << "\n";
            for (i = list_operan[1].size(); i <= list_operan[0].size();
i++){
                cout << " ";
            }
            cout << list_operan[1] << "\n";
            for (i = 0; i<list_operan[3].size(); i++){
                cout << "-";
            }
            cout << "\n" << list_operan[3];
        }

    /* MENGHITUNG WAKTU KETIKA DIMULAI PROGRAM SAMPAI AKHIR PROGRAM
    */

    clock_t tStart = clock();

    /* MEMISAHKAN HURUF YANG UNIK / BERBEDA KE DALAM LIST HURUF UNIK
    */

    char operan1[list_operan.size()];
    char operan2[list_operan.size()];
    char operan3[list_operan.size()];

    for (i = 0; i < list_operan[0].size(); i++){
        operan1[i] = list_operan[0][i];
    }
    for (i = 0; i < list_operan[1].size() - 1; i++){
        operan2[i] = list_operan[1][i];
    }

```

```

}
for (i = 0; i<list_operan[3].size(); i++){
    operan3[i] = list_operan[3][i];
}

hanyaunik(operan1);
hanyaunik(operan2);
hanyaunik(operan3);

int huruf1 = 0;
int huruf2 = posisihuruf(semuahurufdioperan, operan2[0]);
int huruf3 = posisihuruf(semuahurufdioperan, operan3[0]);

/* JIKA JUMLAH HURUF DI DALAM OPERAND LEBIH DARI 10 */
if(panjang > 10){
    cout << "Jumlah huruf di dalam operand paling banyak 10
buah!";
    return 0;
}

/* Mencari kemungkinan angka dari list huruf dengan brute force
*/
if (list_operan.size()<4){
    percobaan = permutasi(10,panjang);
    cout << "\nPercobaan yang seharusnya dilakukan : " <<
percobaan;
}

solusi = 0;

for(i = 1; i <= percobaan; i++){
    masuknilaikeHuruf();
    for(j = 0; j < panjang; j++){
        for(k = j+1; k < panjang; k++){
            if(nilaihuruf[j] == nilaihuruf[k]){
                masuknilaikeHuruf();
                j=-1;
                break;
            }
        }
    }
}

```

```

    }
}

angka1 = 0; angka2 = 0; angka3 = 0;

/* MENGISI ANGKA SESUAI DENGAN POSISI HURUF DI OPERAN */
for(j = 0; j < list_operan[0].size(); j++){
    angka1 = (angka1*10) +
posisi(semuahurufdioperan,operan1[j]);
}
for(j = 0; j < list_operan[1].size()-1; j++){
    angka2 = (angka2*10) +
posisi(semuahurufdioperan,operan2[j]);
}
for(j = 0; j < list_operan[3].size(); j++){
    angka3 = (angka3*10) +
posisi(semuahurufdioperan,operan3[j]);
}

/* SOLUSI BERUPA HASIL PENJUMLAHAN DI KEDUA OPERAN DAN HURUF
PERTAMA TIDAK BOLEH 0*/
if((angka1 + angka2 == angka3) && (nilaihuruf[huruf1] != 0)
&& (nilaihuruf[huruf2] != 0) && (nilaihuruf[huruf3] != 0)){
    solusi++;
    if (solusi > 1){
        break;
    }
    else {
        cout << "\nTotal percobaan yang telah dilakukan : "
<< i << "\n\n";
        if (list_operan[0].size() < list_operan[1].size()){
            for (i = list_operan[0].size(); i <
list_operan[1].size() - 1; i++){
                cout << " ";
            }
            cout << angka1 << "\n";
            cout << angka2 << '+' << "\n";
            for (i = 0; i<list_operan[3].size(); i++){
                cout << "-";
            }
        }
    }
}

```



```

        }
        cout << "\n" << angka3;
        cout<<"\n\nWaktu yang diperlukan:"
<<endl<<(double)(clock() - tStart)/CLOCKS_PER_SEC<<" detik"<<endl;
    }
    else {
        cout << angka1 << "\n";
        for (i = list_operan[1].size(); i <=
list_operan[0].size(); i++){
            cout << " ";
        }
        cout << angka2 << '+' << "\n";
        for (i = 0; i<list_operan[3].size(); i++){
            cout << "-";
        }
        cout << "\n" << angka3;
        cout<<"\n\nWaktu yang diperlukan:"
<<endl<<(double)(clock() - tStart)/CLOCKS_PER_SEC<<" detik"<<endl;
    }
}
}
}
return 0;
}

```

III. SCREENSHOT INPUT OUTPUT

1. Soal 1

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
SEND
MORE+
-----
MONEY

Percobaan yang seharusnya dilakukan : 1814400
Total percobaan yang telah dilakukan : 1748230

9567
1085+
-----
10652

Waktu yang diperlukan:
6.106 detik
```

2. Soal 2

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
GO
TO+
---
OUT

Percobaan yang seharusnya dilakukan : 30240
Total percobaan yang telah dilakukan : 24577

81
21+
---
102

Waktu yang diperlukan:
0.012 detik

Process returned 0 (0x0)   execution time : 0.089 s
Press any key to continue.
```

3. Soal 3

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
USA
USSR+
-----
PEACE

Percobaan yang seharusnya dilakukan : 604800
Total percobaan yang telah dilakukan : 566904

932
9338+
-----
10270

Waktu yang diperlukan:
0.879 detik

Process returned 0 (0x0)   execution time : 1.734 s
Press any key to continue.
```

4. Soal 4

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
EVER
SINCE+
-----
DARWIN

Percobaan yang seharusnya dilakukan : 3628800
Total percobaan yang telah dilakukan : 2036031

5653
97825+
-----
103478

Waktu yang diperlukan:
287.363 detik
```

5. Soal 5

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
HERE
SHE+
-----
COMES
Percobaan yang seharusnya dilakukan : 604800
Total percobaan yang telah dilakukan : 575302

9454
894+
-----
10348

Waktu yang diperlukan:
0.713 detik
```

6. Soal 6

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
COCA
COLA+
-----
OASIS

Percobaan yang seharusnya dilakukan : 151200
Total percobaan yang telah dilakukan : 123695

8186
8106+
-----
16292

Waktu yang diperlukan:
0.274 detik
```

7. Soal 7

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
NUMBER
NUMBER+
-----
PUZZLE

Percobaan yang seharusnya dilakukan : 3628800
Total percobaan yang telah dilakukan : 728504

201689
201689+
-----
403378

Waktu yang diperlukan:
11.271 detik
```

8. Soal 8

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
TILES
PUZZLES+
-----
PICTURE

Percobaan yang seharusnya dilakukan : 3628800
Total percobaan yang telah dilakukan : 3328707

91542
3077542+
-----
3169084

Waktu yang diperlukan:
471.154 detik
```

9. Soal 9

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
MEMO
FROM+
-----
HOMER

Percobaan yang seharusnya dilakukan : 151200
Total percobaan yang telah dilakukan : 128687

8485
7358+
-----
15843

Waktu yang diperlukan:
0.252 detik
```

10. Soal 10

```
Select C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
CROSS
ROADS+
-----
DANGER

Percobaan yang seharusnya dilakukan : 3628800
Total percobaan yang telah dilakukan : 3519768

96233
62513+
-----
158746

Waktu yang diperlukan:
52.885 detik
```

11. Lebih dari 2 operan

Jika lebih dari 2 operan, maka program hanya menuliskan isi file saja.

```
C:\Users\Asus\OneDrive\Documents\Tucil1_13519038\src\Tucil1_13519038.exe
IF
STI
BIOMED+
-----
ITBNDG

Process returned 0 (0x0)   execution time : 0.074 s
Press any key to continue.
```

IV. ALAMAT DRIVE KODE PROGRAM

<https://github.com/ridhodaffasyah/TUGAS-KECIL-1---IF2211-STRATEGI-ALGORITMA---CRYPTARITHMATIC.git>

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (<i>no syntax error</i>)	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat membaca file masukan dan menuliskan luaran.	✓	
4. Solusi cryptarithmic hanya benar untuk persoalan <i>cryptarithmic</i> dengan dua buah operand.	✓	
5. Solusi cryptarithmic benar untuk persoalan <i>cryptarithmic</i> untuk lebih dari dua buah operand.		✓