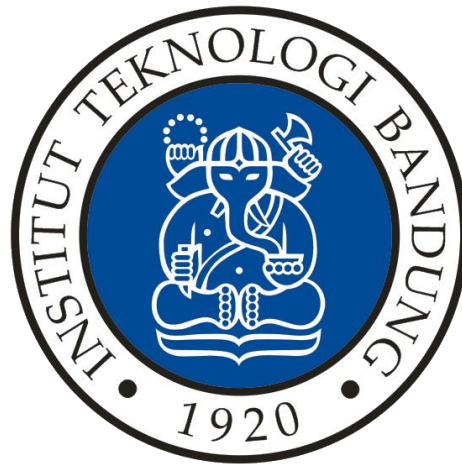


**LAPORAN TUGAS KECIL 1**  
**IF2211 - STRATEGI ALGORITMA**

**Topik/Judul : Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force***



**Nama** : Gilang Ardyamandala Al Assyifa  
**NIM** : 13515096  
**Kelas** : K03  
**Nama Dosen** : Rinaldi Munir  
**Dalam Bahasa** : C++  
**Deadline** : Rabu, 25 Januari 2017

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**BANDUNG**  
**2017**

## A. Algoritma *Brute Force* untuk menyelesaikan *Cryptarithmic*

1. Dapatkan semua huruf yang muncul pada operan maupun hasil penjumlahan.
2. Dapatkan faktor pengali dari masing-masing huruf yang didapatkan dari langkah 1 berdasarkan letaknya, dengan ketentuan:
  - a. Jika huruf muncul di operan, maka tambahkan ke faktor pengali.
  - b. Jika huruf muncul di hasil penjumlahan, maka kurangi dari faktor pengali.(misal : SEND + MORE = MONEY, maka faktor pengalinya: {<S,1000>, <E,100+1-10>, <N,10-100>, <D,1>, <M,1000-10000>, <O,100-1000>, <R,10>, <Y,-1>})
3. Enumerasi (*list*) semua pasangan instansiasi angka yang mungkin menjadi solusi dengan prinsip permutasi tak-berulang.
4. Evaluasi setiap kemungkinan solusi dari langkah 3 dengan kendala:

Penjumlahan dari setiap perkalian (*sum of product*) instansiasi angka dengan faktor pengalinya, jika hasilnya 0, maka pasangan instansiasi angka tersebut merupakan salah satu solusinya (catatan : suatu persoalan cryptarithmic dapat memiliki solusi lebih dari 1).

## B. Source Code Program

```
/*
 * Nama      : Gilang Ardyamandala Al Assyifa
 * NIM       : 13515096
 * Nama file  : Tucill_13515096.cpp
 * Tugas     : Penyelesaian Cryptarithmic dengan Algoritma Brute Force
 * Mata kuliah : IF2211 - Strategi Algoritma
 * Kelas/dosen : KO3/Rinaldi Munir
 */

#include <iostream>
#include <fstream>
#include <string.h>
#include <time.h>
using namespace std;

/***** FUNGSI DAN PROSEDUR TAMBAHAN *****/
/** FUNGSI PERPANGKATAN **/
int pangkat (int a, int b) {
    int tamp = 1;
    for (int i = 1; i <= b; i++) {
        tamp *= a;
    }
}
```

```

    }
    return tamp;
}

/** FUNGSI UNTUK MEMBENTUK STRING HANYA ABJAD DARI SUATI STRING **/
string hanyaABJAD (string sin) {
    int size = sin.length();
    char sout[sin.length()]; int n = 0;
    for (int i = 0; i < size; i++) {
        if (sin[i] >= 65 && sin[i] <= 90) {
            sout[n] = sin[i];
            n++;
        }
    }
    sout[n] = '\0';
    return sout;
}

/** FUNGSI UNTUK MENGECEK APAKAH SUATU KARAKTER ADA DI DALAM SUATU STRING **/
bool cari (char c, string s) {
    bool found = false; int i = 0;
    while (!found && i < s.length()) {
        if (s[i] == c) {
            found = true;
        }
        i++;
    }
    return found;
}

/** FUNGSI UNTUK Mencari POSISI SEBUAH KARAKTER PADA STRING, JIKA TIDAK ADA
KIRIMKAN -1 **/
int caripos (char c, string s) {
    bool found = false; int i = 0;
    while (!found && i < s.length()) {
        if (s[i] == c) {
            found = true;
        }
        i++;
    }
    if (found) {return (i-1);} else {return (-1);}
}

/** FUNGSI UNTUK MEMBENTUK STRING YANG ELEMENNYA UNIK DARI SUATU STRING
SESUAI URUTAN KEMUNCULANNYA **/
string makeUnique (string sin) {
    int size = sin.length();

```

```

    char sout[sin.length()]; int n = 0;
    for (int i = 0; i < size; i++) {
        if (!cari(sin[i],sout)) {
            sout[n] = sin[i];
            n++;
        }
    }
    sout[n] = '\\0';
    return sout;
}

/***** PROGRAM UTAMA *****/
int main() {
    /** KAMUS **/
    int c[10];           // Counter
    int k[10];           // Faktor pengali
    int m[10]; int s[10]; // Mulai dan Selesai
    int nSol = 0;        // Banyak solusi unik
    int nOperan = 0;     // Banyak operan

    string line;        // Pembacaan file eksternal
    string sn[10];      // String operan
    string sh;          // String hasil penjumlahan
    string sc;          // String unik dari operan dan hasil penjumlahan
    string semua;       // String yang mewakili keseluruhan file eksternal

    /* MEMBACA FILE EKSTERNAL */
    ifstream myfile ("soal.txt");
    if (myfile.is_open()) {
        do {
            getline (myfile,line);
            sn[nOperan] = line; semua += line + "\\n";
            nOperan++;
        } while (line[0] != '+');

        getline (myfile,line); semua += line + "\\n";

        getline (myfile,line);
        sh = line; semua += line + "\\n";
        myfile.close();
        cout << semua;
    } else {cout << "File tidak bisa diakses, silahkan cek dahulu file
    eksternalnya agar program ini berjalan dengan benar." << endl;}

    clock_t tStart = clock(); // Timer diaktifkan setelah pembacaan file
    eksternal

    /* MENGGABUNGKAN SEMUA STRING DAN Mencari string uniknya */

```

```

bool hasildulu = true;
for (int i = 0; i < nOperan; i++) {
    sc += hanyaABJAD(sn[i]);
    hasildulu &= (sn[i][1] == ' ');
}
if (hasildulu) {
    sc = hanyaABJAD(sh)+sc;
} else {sc += hanyaABJAD(sh);}

sc = makeUnique(sc); // Membuat string gabungan menjadi unik
int maxleng = sc.length(); // Panjang string

/* MENENTUKAN FAKTOR PENGALI DARI SUATU STRING */
for (int i = 10-maxleng; i < 10; i++) {
    m[i] = 0; s[i] = 10;
    k[i] = 0;
    string stamp;
    for (int b = 0; b < nOperan; b++) {
        stamp = hanyaABJAD(sn[b]);
        for (int j = stamp.length()-1; j >= 0; j--) {
            if (sc[i-(10-maxleng)] == stamp[j]) {
                k[i] += pangkat(10,stamp.length()-j-1);
                if (j == 0) {m[i] = 1;}
            }
        }
    }

    stamp = hanyaABJAD(sh);
    for (int j = stamp.length()-1; j >= 0; j--) {
        if (sc[i-(10-maxleng)] == stamp[j]) {
            k[i] -= pangkat(10,stamp.length()-j-1);
            if (j == 0) {m[i] = 1;}
        }
    }
}

// Inisiasi nilai variabel yang tidak dipakai agar tidak mengganggu
for (int i = 0; i < 10-maxleng; i++) {
    k[i] = 0;
    m[i] = 10*(i+1);
    s[i] = m[i] + 1;
}

/** ALGORITMA BRUTE FORCE */
for (c[0] = m[0]; c[0] < s[0]; c[0]++) {
    for (c[1] = m[1]; c[1] < s[1]; c[1]++) {
        for (c[2] = m[2]; c[2] < s[2]; c[2]++) {
            for (c[3] = m[3]; c[3] < s[3]; c[3]++) {

```

```

        for (c[4] = m[4]; c[4] < s[4]; c[4]++) {
            for (c[5] = m[5]; c[5] < s[5]; c[5]++) {
                for (c[6] = m[6]; c[6] < s[6]; c[6]++) {
                    for (c[7] = m[7]; c[7] < s[7]; c[7]++) {
                        for (c[8] = m[8]; c[8] < s[8]; c[8]++) {
                            for (c[9] = m[9]; c[9] < s[9];
c[9]++) {

                                // Mengecek kumpulan iteratornya
unik

                                if (c[2]!=c[3] && c[2]!=c[4] &&
c[2]!=c[5] && c[2]!=c[6] && c[2]!=c[7] && c[2]!=c[8] && c[2]!=c[9] &&
c[3]!=c[4] && c[3]!=c[5] && c[3]!=c[6] && c[3]!=c[7] && c[3]!=c[8] &&
c[3]!=c[9] && c[4]!=c[5] && c[4]!=c[6] && c[4]!=c[7] && c[4]!=c[8] &&
c[4]!=c[9] && c[5]!=c[6] && c[5]!=c[7] && c[5]!=c[8] && c[5]!=c[9] &&
c[6]!=c[7] && c[6]!=c[8] && c[6]!=c[9] && c[7]!=c[8] && c[7]!=c[9] &&
c[8]!=c[9] && c[0]!=c[1] && c[0]!=c[2] && c[0]!=c[3] && c[0]!=c[4] &&
c[0]!=c[5] && c[0]!=c[6] && c[0]!=c[7] && c[0]!=c[8] && c[0]!=c[9] &&
c[1]!=c[2] && c[1]!=c[3] && c[1]!=c[4] && c[1]!=c[5] && c[1]!=c[6] &&
c[1]!=c[7] && c[1]!=c[8] && c[1]!=c[9]) {

                                    // Mengecek apakah iterator
pada posisi ini adalah solusinya

                                    if (c[0]*k[0] + c[1]*k[1] +
c[2]*k[2] + c[3]*k[3] + c[4]*k[4] + c[5]*k[5] + c[6]*k[6] + c[7]*k[7] +
c[8]*k[8] + c[9]*k[9] == 0) {

                                        cout << endl;
                                        for (int i = 0; i <
semua.length(); i++) {

                                            if (semua[i] < 65 ||
semua[i] > 90) {

                                                cout <<

semua[i];

                                                } else {cout <<
c[caripos(semua[i],sc)+10-maxleng];}

                                            }
nSol++; // Solusi

bertambah 1

// Mencetak waktu yang
dibutuhkan untuk mendapatkan solusi

cout << "Waktu untuk
mendapatkan solusi ke-" << nSol << " : " << (double)(clock() -
tStart)/CLOCKS_PER_SEC << "s" << endl;

tStart = clock(); //

Waktu direset

}

}

}

}

}

```

```

    }
    }
    }
    }
    }

    // Mencetak jumlah solusi yang diperoleh
    if (nSol == 0) {
        cout << "\nTidak ada solusi yang memenuhi" << endl;
    } else {
        cout << "\nTotal solusi : " << nSol << endl;
    }

    cout << "\nTekan <enter> untuk keluar" << endl; getchar();
    return 0;
}

```

### C. *Input dan Output*

This tool reports detailed information about the DirectX components and drivers installed on your system.

If you know what area is causing the problem, click the appropriate tab above. Otherwise, you can use the "Next Page" button below to visit each page in sequence.

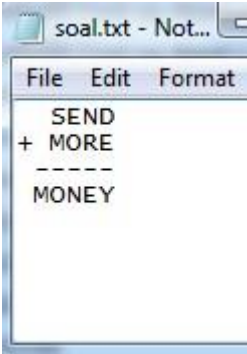

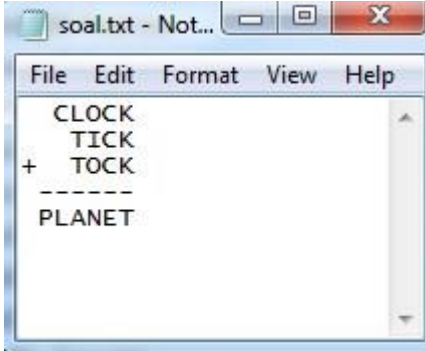
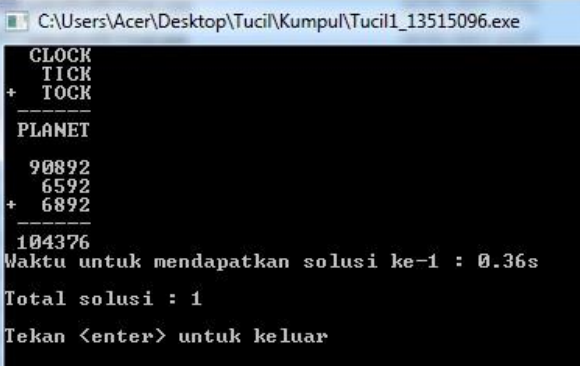
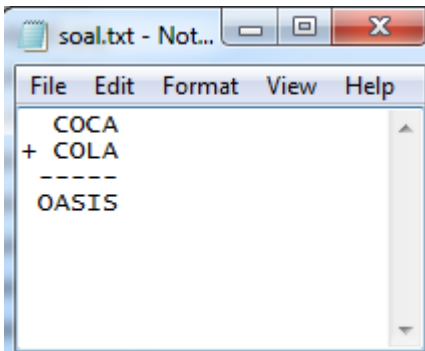
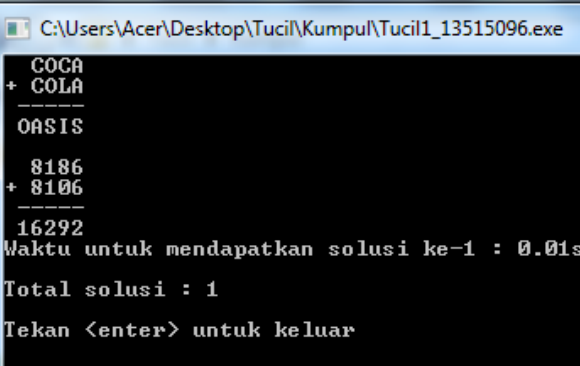
#### System Information

```

Current Date/Time: 25 Januari 2017, 14:18:45
Computer Name: ACER-PC
Operating System: Windows 7 Professional 64-bit (6.1, Build 7601)
Language: Bahasa Indonesia (Regional Setting: Bahasa Indonesia)
System Manufacturer: Acer
System Model: Aspire E1-532
BIOS: InsydeH2O Version 03.73.06.0055V2.12
Processor: Intel(R) Celeron(R) 2955U @ 1.40GHz (2 CPUs), ~1.4GHz
Memory: 2048MB RAM
Page file: 2065MB used, 1799MB available
DirectX Version: DirectX 11

```

*Gambar 1 - Spesifikasi komputer yang digunakan*

Input	Output
<p>1.</p> 	
<p>2.</p> 	
<p>3.</p> 	



4.

```
soal.txt - Not...
File Edit Format View Help
DOUBLE
DOUBLE
+ TOIL
-----
TROUBLE
```

```
C:\Users\Acer\Desktop\Tucil\Kumpul\Tucil1_13515096.exe
DOUBLE
DOUBLE
+ TOIL
-----
TROUBLE
798064
798064
+ 1936
-----
1598064
Waktu untuk mendapatkan solusi ke-1 : 0.53s
Total solusi : 1
Tekan <enter> untuk keluar
```

5.

```
soal.txt - Not...
File Edit Format View Help
THREE
THREE
TWO
TWO
+ ONE
-----
ELEVEN
```

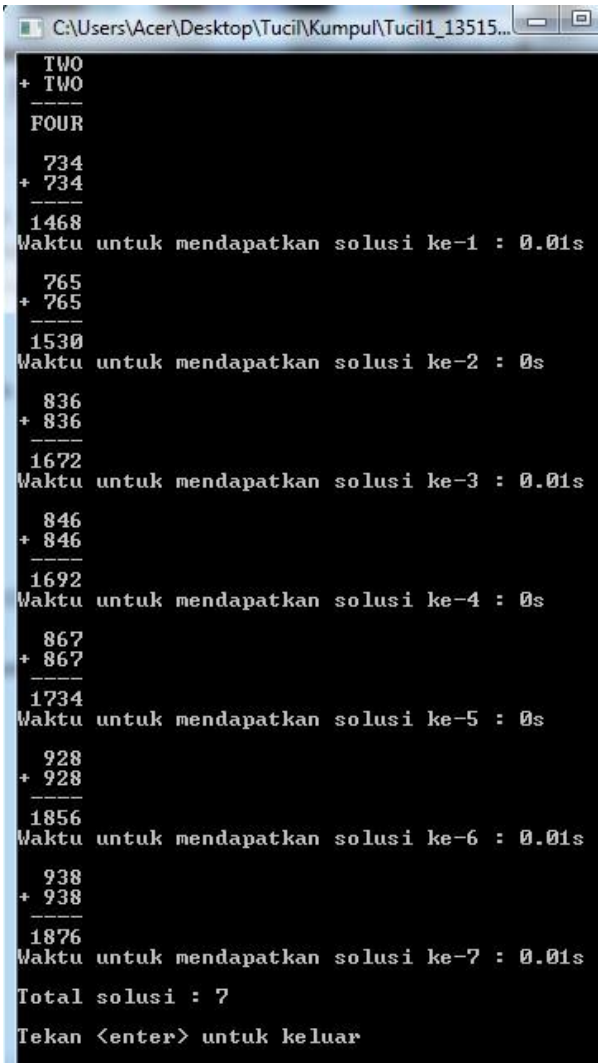
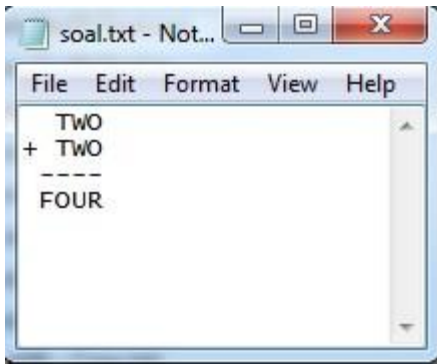
```
C:\Users\Acer\Desktop\Tucil\Kumpul\Tucil1_13515096.exe
THREE
THREE
TWO
TWO
+ ONE
-----
ELEVEN
84611
84611
803
803
+ 391
-----
171219
Waktu untuk mendapatkan solusi ke-1 : 0.585s
Total solusi : 1
Tekan <enter> untuk keluar
```

6.

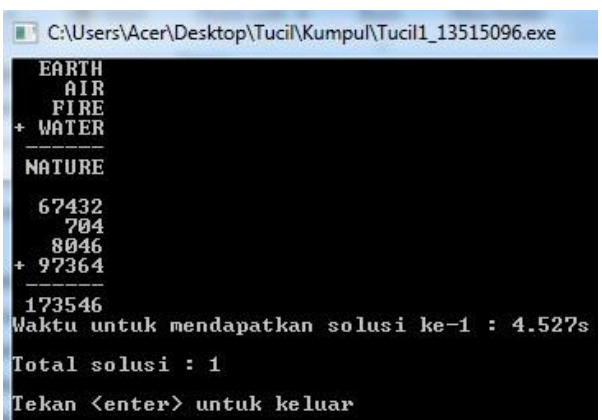
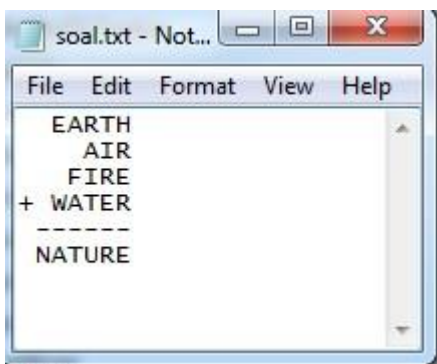
```
soal.txt - Not...
File Edit Format View Help
MEMO
+ FROM
-----
HOMER
```

```
C:\Users\Acer\Desktop\Tucil\Kumpul\Tucil1_13515096.exe
MEMO
+ FROM
-----
HOMER
8485
+ 7358
-----
15843
Waktu untuk mendapatkan solusi ke-1 : 0.01s
Total solusi : 1
Tekan <enter> untuk keluar
```

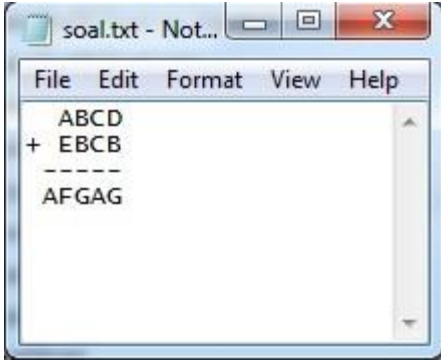
7.

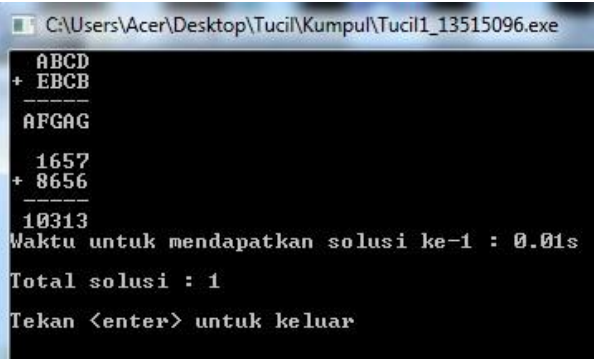


8.

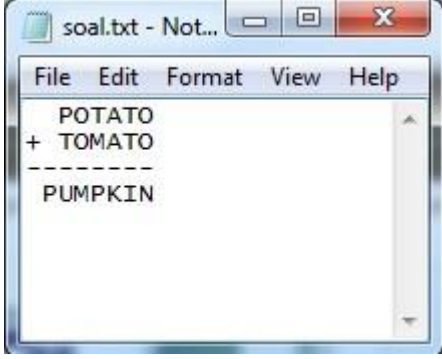


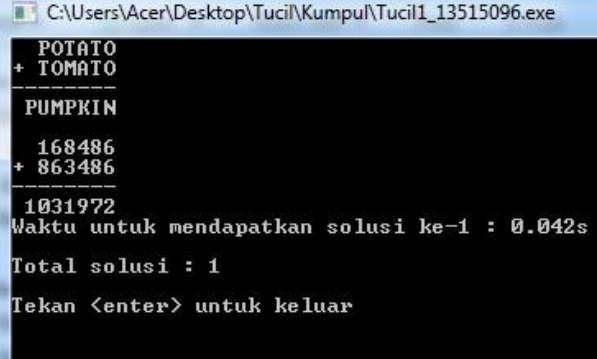
9.





10.





Poin	Ya	Tidak
1. Program berhasil dikompilasi	v	
2. Program berhasil <i>running</i>	v	
3. Program dapat membaca file masukan dan menuliskan luaran.	v	
4. Solusi <i>cryptarithmic</i> <b>hanya</b> benar untuk persoalan <i>cryptarihtmetic</i> dengan dua buah <i>operand</i> .		v
5. Solusi <i>cryptarithmic</i> benar untuk persoalan <i>cryptarihtmetic</i> untuk lebih dari dua buah <i>operand</i> .	v	