

**LAPORAN PRAKTIKUM MODUL KE-7**  
**MEMBUAT OPERASI FILE CSV PADA PROGRAM CRUD MAHASISWA**



**Dosen Pengampu :**  
**I Ketut Purnamawan, S.Kom., M.Kom.**

**Disusun Oleh :**  
**I Gede Gelgel Abdiutama ; 2115101014**

**MATA KULIAH PRAKTIKUM ALGORITMA DAN PEMROGRAMAN**  
**UNIVERSITAS PENDIDIKAN GANESHA**  
**SINGARAJA**  
**TA. 2022**

## **A. PERMASALAHAN**

Pada kegiatan Praktikum ini terdapat Studi kasus yang akan diselesaikan yaitu membuat program yang bisa meng-input/menambah, meng-edit, menghapus dan menampilkan data mahasiswa yang terdiri dari: NIM, nama, jenis kelamin, tahun lahir, tahun masuk, dan IPK. Program bisa menampilkan data seluruh mahasiswa, maupun satu mahasiswa saja berdasarkan NIM yang di-input-kan user, kalau mahasiswa dengan NIM yang diinputkan ada. Kalau data mahasiswa dengan NIM yang di-input-kan tidak ada, maka program menampilkan pemberitahuan bahwa mahasiswa dengan NIM tersebut tidak ada. Dan program tersebut dibuat dalam bentuk operasi file (disimpan dalam bentuk yaitu .csv) File CSV yang dihasilkan harus bisa juga dibuka dengan benar menggunakan Microsoft Excel. Program juga harus bisa membuka file CSV data mahasiswa yang dibuat menggunakan Microsoft Excel, dan bisa mengolah datanya (menambah, mengubah, menghapus, dan menampilkan).

## **B. KAJIAN TEORI**

### **1. Bahasa Pemrograman C**

Bahasa pemrograman C dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Meskipun C dibuat untuk memprogram sistem dan jaringan komputer, bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan ekstensi dari C. Bahasa C terdiri dari beberapa versi seperti C K&R, ANSI C & ISO C, dan C99.

Contoh program bahasa C:

```
#include <stdio>

int main()
{
    printf("Hello World");
    return 0;
}
```

## **2. Bahasa Pemrograman C++**

Bahasa pemrograman C++ adalah bahasa pemrograman komputer yang di buat oleh Bjarne Stroustrup, yang merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, bahasa itu diturunkan dari bahasa B yang ditulis oleh Ken Thompson pada tahun 1970 yang diturunkan dari bahasa sebelumnya yaitu BCL. Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix. Pada perkembangannya, versi ANSI (American National Standards Institute) pada bahasa pemrograman C menjadi versi dominan, meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk embedded system. Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan support system untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (object-oriented programming).

Contoh program bahasa C++:

```
#include <iostream>

int main()
{
    std::cout << "Hello World";
    return 0;
}
```

## **3. Tipe Data**

Data types atau tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Untuk mengembangkan program dalam bahasa pemrograman C atau C++ terdapat berbagai jenis tipe data yang dapat dipilih dan digunakan sesuai dengan kebutuhan dan karakter nilai yang ingin disimpan di dalam variable. Jenis tipe data yang sering digunakan yaitu Boolean, Character, String, Integer, Floating Point, dan Double Floating Point.

**a. Boolean**

Tipe data Boolean merupakan tipe yang memiliki dua nilai yaitu benar (true) atau salah (false). Nilai yang digunakan pada tipe ini sangat penting dalam mengambil keputusan suatu kejadian tertentu.

**b. Character**

Tipe data character merupakan salah satu tipe data yang memungkinkan kita untuk memesan memori berformat text (huruf, angka, dan simbol) dengan karakter tunggal. Dibutuhkan 1 byte atau 8 bit ruang di dalam memori agar dapat menyimpan sebuah karakter.

**c. String**

Tipe data string terdiri dari kumpulan karakter dengan panjang tertentu, dan seringkali dianggap sebagai tipe data dasar. Hal ini dikarenakan hingga saat ini tipe data string paling sering digunakan oleh para programmer.

**d. Integer**

Jenis tipe data integer dapat didefinisikan sebagai bilangan bulat. Artinya, suatu program yang menggunakan tipe data integer ini tidak mendukung penggunaan huruf. Selain itu, bilangan yang digunakan juga haruslah bulat (tidak mengandung pecahan decimal).

**e. Floating Point**

Tipe data floating point atau real number merupakan tipe data angka yang memiliki bagian decimal di akhir angka. Tipe data float cocok digunakan untuk variable yang akan berisi angka pecahan.

**f. Double Floating Point**

Sama halnya dengan floating point, yang bersifat menyatakan bilangan pecahan. Bedanya adalah penyimpangan angka maksimal lebih besar daripada float, otomatis double juga akan membutuhkan memori yang lebih besar.

**4. Array**

Dalam beberapa literatur, array sering disebut (diterjemahkan) sebagai larik. Array adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Nilai-nilai data di suatu array disebut dengan elemen-elemen array. Letak urutan dari elemen-elemen array ditunjukkan oleh suatu subscript atau indeks. Array bisa berupa array berdimensi satu, dua, tiga atau lebih. Array berdimensi satu (one-dimensional array) mewakili bentuk suatu vektor. Array

berdimensi dua (twodimensional array) mewakili bentuk dari suatu matriks atau table. Array berdimensi tiga (three-dimensional array) mewakili bentuk suatu ruang.

## 5. Percabangan

Percabangan adalah sebuah tahap dimana program akan melakukan pengecekan kondisi. Kondisi ini bisa digunakan untuk menentukan bagian program/statement mana yang akan dijalankan jika kondisi tertentu terpenuhi. Di dalam bahasa C, kita dapat membuat seleksi dengan if else.

### a. if

Pernyataan if : “Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan.”

```
if(kondisi){  
//statement  
}
```

### b. if else

Pernyataan if else : “Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2”

```
if(kondisi){  
//statement  
}else{  
//statement  
}
```

## 6. Perulangan

Perulangan data atau yang biasa disebut dengan “looping” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan. Biasanya bila dalam perulangan tersebut tidak disertakan batasnya maka syntax akan error karena proses itu akan berulang terus hingga tak terhingga sementara variabel dalam komputer masih terbatas. Dalam bahasa C tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan kita. Struktur perulangan dalam bahasa C mempunyai bentuk yang bermacam-macam. Sebuah/kelompok instruksi diulang untuk jumlah perulangan tertentu. Baik yang terdefiniskan sebelumnya atau pun tidak. Struktur perulangan terdiri atas dua bagian :

- Kondisi perulangan yaitu ekspresi Boolean yang harus dipenuhi untuk melaksanakan perulangan
- Isi atau badan pengulangan yaitu satu atau lebih pernyataan (aksi) yang akan diulang

#### **a. Statement For**

Struktur perulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan for tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Pernyataan for digunakan untuk melakukan looping. Pada umumnya looping yang dilakukan oleh for telah diketahui batas awal, syarat looping dan perubahannya. Selama kondisi terpenuhi, maka pernyataan akan terus dieksekusi.

#### **b. Statement While**

Perulangan WHILE banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar ( $\neq 0$ ) dan akan berhenti bila kondisinya bernilai salah ( $=0$ ).

#### **c. Statement Do-While**

Perintah DO ... WHILE hampir sama dengan WHILE sebelumnya. Perbedaan dengan WHILE sebelumnya yaitu bahwa pada DO WHILE statement perulangannya dilakukan terlebih dahulu baru kemudian di cek kondisinya. Sedangkan WHILE kondisi dicek dulu baru kemudian statement perulangannya dijalankan. Akibat dari hal ini adalah dalam DO WHILE minimal terdapat 1x perulangan. Sedangkan WHILE dimungkinkan perulangan tidak pernah terjadi yaitu ketika kondisinya langsung bernilai FALSE.

### **7. Operasi File**

Penggunaan operasi FILE dalam sebuah program sangat dibutuhkan dalam pembuatan program yang sesungguhnya. Membuat operasi file sebagai tempat penampung data-data selama operasi program. Jika menggunakan variabel biasa, data yang biasanya diolah program hanya akan tersimpan sementara dalam memory komputer dan akan hilang ketika program close seperti sifat dasar pada RAM. Berbeda dengan memory, penyimpanan data berbasis file akan tersimpan terus walaupun program telah diclose maupun komputer telah di shutdown. Pada dasarnya, operasi file terbagi 3 jenis. Read (R), Write(W), dan Append(A). Read artinya membaca isi file, write menulis data ke file dan append menambahkan data ke baris terakhir dalam file.

### C. PEMBAHASAN

Dalam pembahasan ini, akan menjelaskan proses atau tahapan yang dilakukan pada saat melakukan Praktikum membuat Program yang dapat melakukan operasi CRUD (Create, Read, Update, dan Delete) untuk bisa meng-input/menambah, meng-edit, menghapus dan menampilkan data mahasiswa. Sebenarnya ini adalah pengembangan dari tugas sebelumnya, namun bedanya program ini bisa menyimpan data yang sifatnya permanen tersimpan di lokal disk komputer kita, tidak lagi tersimpan dalam memori komputer yang sifatnya hanya sementara, sebelum membahas mengenai langkah-langkah teknis pembuatan, akan dijelaskan terlebih dahulu bagaimana program ini dapat bekerja.

Dalam program ini dibuat menggunakan bahasa pemrograman C++ dengan library standar bahasa C++ (`#include <iostream>`) yang fungsinya sebagai header yang menampung kumpulan library yang ada sehingga bisa kita gunakan pada program kita. Dan juga terdapat beberapa library lainnya seperti `conio.h` dan `stdlib.h`, yang dimana tambahan library ini berfungsi untuk memaksimalkan fungsi yang dibuat pada program. Bagian dalam yang membentuk kode program ini yaitu menggunakan array dan struct yang terdapat tipe data char dan integer untuk menampung atribut-atribut variabel pendukung nantinya akan digunakan didalam fungsi program. Program ini menggunakan perulangan for dan percabangan if else agar program ini bisa berjalan disaat ada proses pemilihan kondisi saat user memasukkan berbagai macam pilihan. Program ini menggunakan fungsi yang dimana beberapa operasi-operasi yang ada ditempatkan pada void variabel() sehingga pada main program kita bisa lebih mudah memanggil operasi yang ada dengan hanya mengetikkan fungsi yang telah dibuat.

Program CRUD Data Mahasiswa ini menggunakan string namun tidak termasuk tipe data dasar dari program, jika diartikan string itu sebuah array satu dimensi yang berisi kumpulan dari huruf atau karakter dan diakhiri dengan karakter kosong(NULL). Program ini menggunakan bahasa pemrograman C++ namun tidak memiliki tipe data string, melainkan kita tetap bisa membuat string dengan tipe data char. Sebenarnya ada beberapa fungsi yang bisa digunakan untuk memanipulasi string sesuai dengan kebutuhan contohnya saja pada saat membuat program ini saya menggunakan fungsi `strcmp()`. Fungsi `strcmp()` digunakan untuk membandingkan string dengan string yang lainnya dan akan menghasilkan nilai 0 apabila kedua string yang dibandingkan sama lalu jika berbeda akan menghasilkan angka yang berbeda-beda.

Didalam program ini juga menggunakan beberapa perintah/syntax tambahan untuk memperhalus program saat digunakan, diantaranya sebagai berikut :

1. **getch();** yaitu berfungsi untuk Tidak memberikan efek pergantian baris secara otomatis, getch() akan membaca nilai yang dimasukkan user tetapi tidak menampilkannya di layar. Ketika user memberikan karakter input, itu tidak ditampilkan di layar dan tanpa menunggu tombol enter, output printf ditampilkan ke layar pada posisi berikutnya. Ini hanya ditampilkan karena adanya fungsi printf.
2. **system("cls");** yaitu berfungsi membersihkan layar pada program yang akan dijalankan sehingga pem-rogram dapat menentukan saat untuk menghapus data yang telah dijalankan tanpa harus menutup program tersebut dan membukanya kembali.
3. **goto;** yaitu berfungsi sebagai pernyataan yang memungkinkan kita untuk mengatur arahnya aliran pengekseskusion program kita. Bisa dikatakan juga untuk memerintahkan program melompat ke baris manapun berdasarkan label yang telah dibuat.

#### Fungsi yang digunakan dalam file processing :

Function	Deskripsi
fopen()	Membuka file
fclose()	Menutup file
fprintf()	Mencetak file
fscanf()	Membaca isi file
feof()	Mendeteksi baris
fputc()	Menulis 1 karakter pada file
fgetc()	Membaca 1 karakter pada file
fseek()	Mengatur pointer file ke posisi tertentu
fputw()	Menulis 1 angka pada file
fgetw()	Membaca 1 angka pada file
ftell()	Return ke posisi sekarang
rewind()	Mengatur file pointer ke awal file

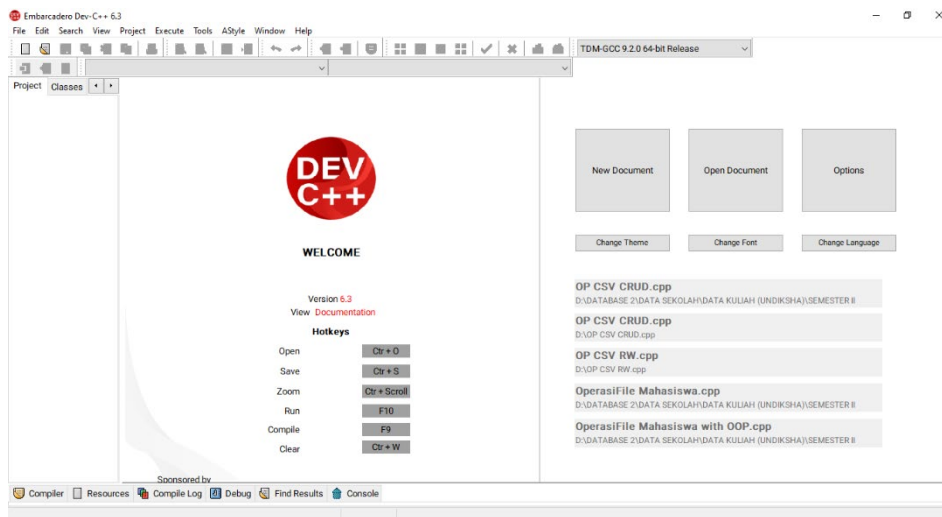
#### Operasi file yang digunakan dalam file processing :

Operasi	Deskripsi
r	Membuka sebuah file untuk dibaca
w	Membuat sebuah file untuk ditulis
a	Membuka file untuk penambahan data
r+	Membuka sebuah file untuk dibaca/ditulis
w+	membuat sebuah file untuk dibaca/ditulis
a+	Membuka file untuk penambahan data/dibaca
rb	Membuka sebuah file binary untuk dibaca
wb	Membuat sebuah file binary untuk ditulis
ab	Membuka file binary untuk penambahan data
rb+	Membuka sebuah file binary untuk dibaca/ditulis
wb+	Membuat sebuah file binary untuk dibaca/ditulis
ab+	Membuka file binary untuk penambahan data/dibaca

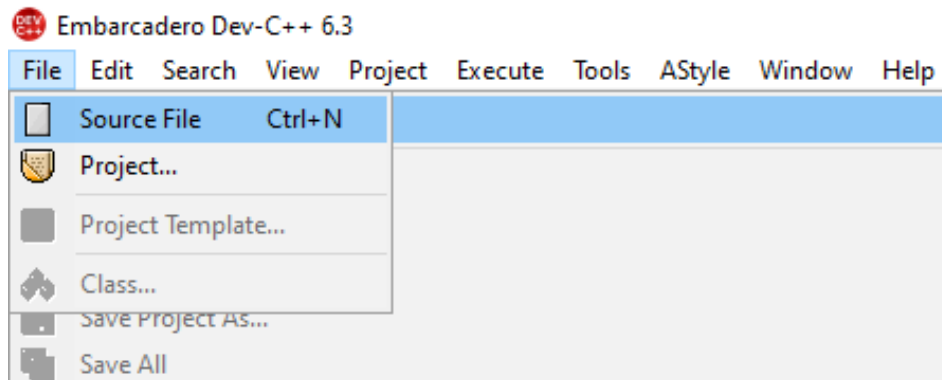


## Berikut Langkah-langkah praktikum

1. Pertama, kita buka terlebih dahulu Compiler yang digunakan. Sebagai contoh disini saya menggunakan Dev C++ sebagai compiler nya.



2. Setelah terbuka, kita buat source file dengan cara memilih pada File > New > Source File, atau bisa juga dengan menggunakan shortcut pada keyboard yaitu CTRL + N.

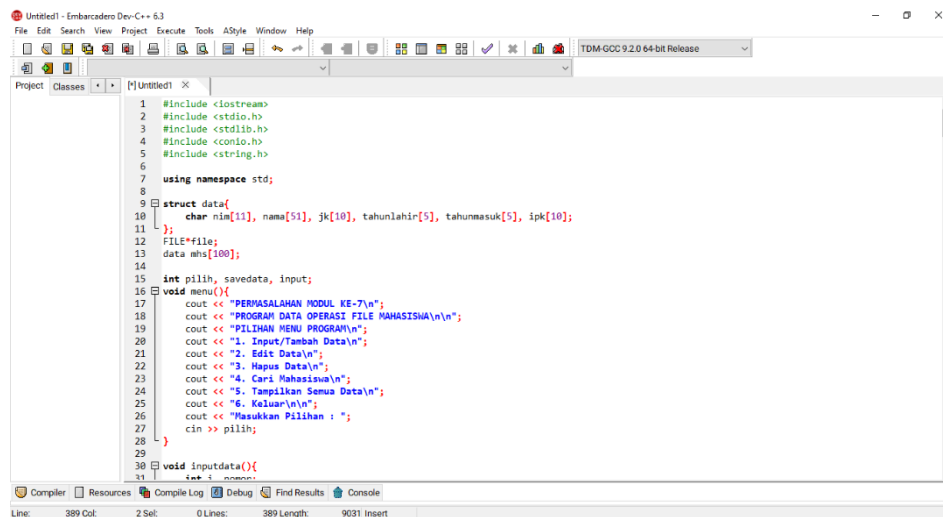


3. Kemudian kita akan eksekusi kode programnya, karena saya menggunakan bahasa pemrograman C, maka kita membutuhkan sebuah kerangka dasarnya untuk bisa membentuk kode program lainnya, berikut merupakan header dan main program dari bahasa C

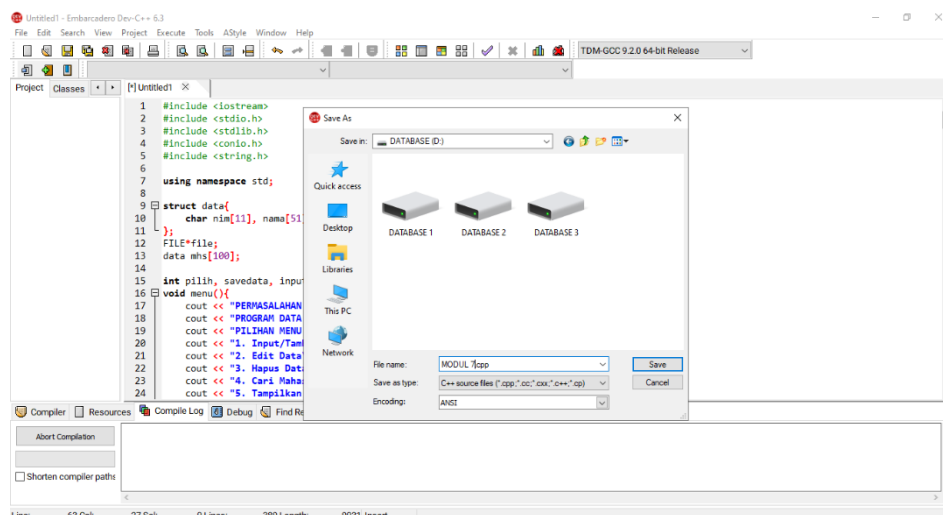
```
#include <stdio>

int main()
{
    printf("Hello World");
    return 0;
}
```

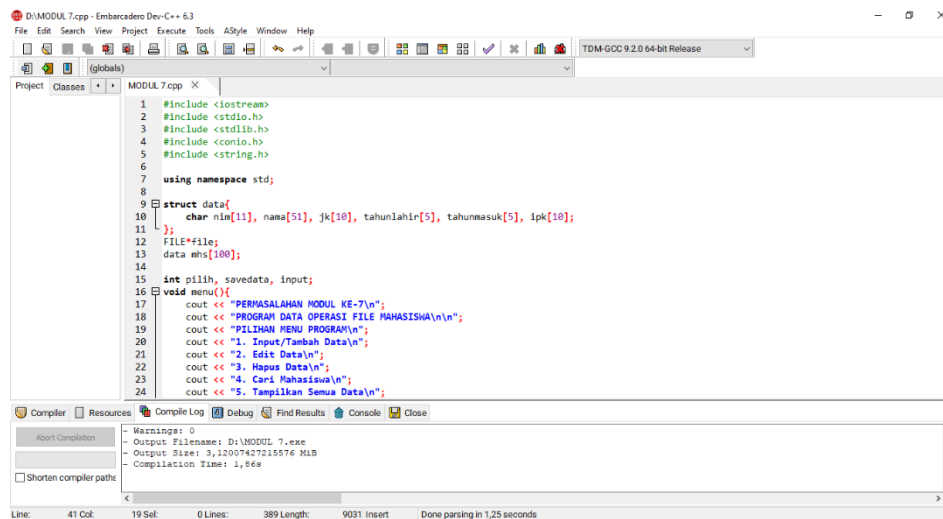
4. Pada tahap ini, kita bisa melanjutkan proses pembuatan program dengan melakukan koding sesuai dengan kode program yang akan dibuat. Untuk kode program akan dilampirkan pada halaman Lampiran paling akhir laporan ini.



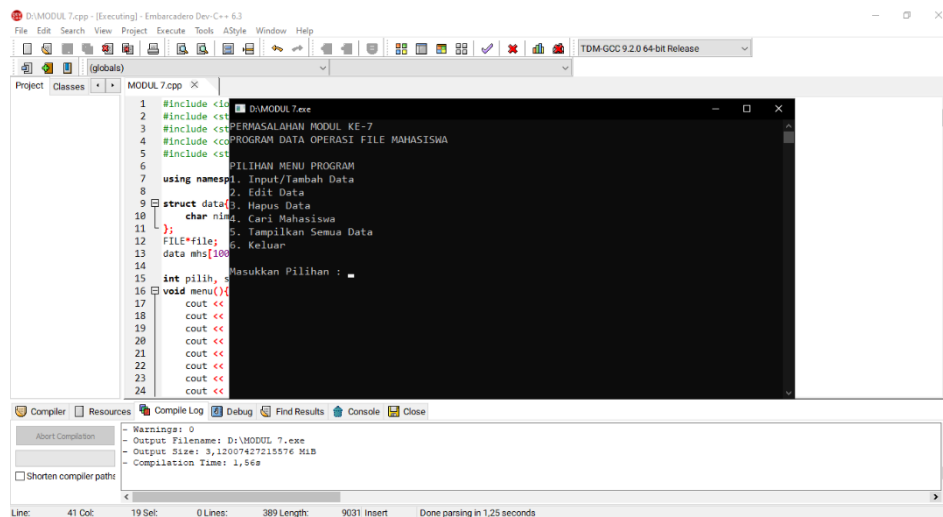
5. Untuk melakukan compile sehingga program dapat dijalankan, kita klik Execute lalu pilih Compile atau bisa juga dengan menggunakan shortcut pada keyboard yaitu F9. Berikutnya akan muncul tampilan yang meminta kita memilih lokasi untuk menyimpan file cpp sebelum di compile. Sebagai contoh disini akan saya simpan dengan nama MODUL 7.cpp.



6. Maka setelah itu hasil dari program yang telah kita buat akan di compile secara otomatis, perlu diperhatikan jika Errors 0 dan Warnings 0 maka program siap untuk di Run dengan cara klik Execute lalu pilih Run menekan tombol F10.



7. Ketika program sudah di run, maka langkah selanjutnya kita akan mengeksekusi program kita dengan cara memilih menu pilihan operasi pada program dengan memasukkan nomor opsinya lalu enter, maka akan langsung diarahkan ke operasi yang terkait, sehingga kita bisa memasukkan perintah selanjutnya sesuai alur program.



#### **D. KESIMPULAN**

Pembuatan program yang bisa meng-input/menambah, meng-edit, menghapus dan menampilkan data mahasiswa atau biasa disebut program CRUD (Create, Read, Update, dan Delete) dan ada tambahan program ini dapat menyimpan file nya kedalam lokal disk komputer yang dinamakan dengan Operasi File. Dalam membuat program ini perlu memahami yang paling utama yaitu operasi file dan yang umum yaitu dalam pemilihan tipe data, menggunakan perulangan, merancang percabangan, konsep data array dan struct, karena itu semua merupakan kunci agar bisa lanjut membuat program ini. Program ini akan dieksekusi langsung melalui compiler Dev C++ dan hasil dari inputan user akan disimpan dalam bentuk file excel yaitu .csv yang dimana secara umum akan dibuka menggunakan Microsoft Excel, sehingga data yang disimpan akan identik dipisah dengan menggunakan tanda koma atau titik koma disetiap datanya. Program ini dibuat sesederhana mungkin sehingga user bisa langsung meng-input/menambah, meng-edit, menghapus dan menampilkan data mahasiswa yang ingin dikelola, dibuatnya program ini juga sebagai solusi agar nantinya dapat berkembang menjadi program yang siap untuk digunakan sehingga mempermudah user/pengguna dalam mengelola database mahasiswa dan keperluan lainnya.

## E. LAMPIRAN

Kode Program:

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

using namespace std;

struct data{
    char nim[11], nama[51], jk[10], tahunlahir[5], tahunmasuk[5], ipk[10];
};
FILE*file;
data mhs[100];

int pilih, savedata, input;
void menu(){
    cout << "PERMASALAHAN MODUL KE-7\n";
    cout << "PROGRAM DATA OPERASI FILE MAHASISWA\n\n";
    cout << "PILIHAN MENU PROGRAM\n";
    cout << "1. Input/Tambah Data\n";
    cout << "2. Edit Data\n";
    cout << "3. Hapus Data\n";
    cout << "4. Cari Mahasiswa\n";
    cout << "5. Tampilkan Semua Data\n";
    cout << "6. Keluar\n\n";
    cout << "Masukkan Pilihan : ";
    cin >> pilih;
}

void inputdata(){
    int i, nomor;
    cout << "Masukkan Jumlah Data Yang Akan Diinput : ";
    cin >> input;
    nomor = 0;
    cout << endl;
    for(i=0; i<input; i++)
    {
        nomor=nomor+1;
        cout << "Data Ke-" << nomor << endl;
        gets(mhs[savedata].nim);
        input_nim:
        cout << "NIM\t\t: ";
        gets(mhs[savedata].nim);
        for (int i=0; i<savedata; i++)
        {
            if (strcmp(mhs[i].nim, mhs[savedata].nim) == 0)
            {
                cout << "Maaf, NIM Sudah Ada\n";
                cout << "Silahkan Masukkan NIM Yang Berbeda!" << endl;
                goto input_nim;
            }
        }
        cout << "Nama\t\t: ";
        gets(mhs[savedata].nama);
        cout << "Jenis Kelamin\t: ";
```

```

        gets(mhs[savedata].jk);
        cout << "Tahun Lahir\t: ";
        cin >> mhs[savedata].tahunlahir;
        cout << "Tahun Masuk\t: ";
        cin >> mhs[savedata].tahunmasuk;
        cout << "IPK\t\t: ";
        cin >> mhs[savedata].ipk;
        cout << "" << endl;
        savedata++;
    }
    fclose(file);
}

void lihatdata(){
    int i, j;
    if(savedata ==0)
    {
        cout <<"Belum Ada Data Tersimpan\n";
        cout <<"Silahkan Input Data Terlebih Dahulu!"<< endl;
        system ("cls");
    }
    else
    {
        j=1;
        file=fopen("Database Mahasiswa.csv","w");
        for(i=0; i<savedata; i++)
        {
            printf ("\nData Ke-%i", j);
            printf ("\nNIM\t\t: %s",&mhs[i].nim);
            printf ("\nNama\t\t: %s",&mhs[i].nama);
            printf ("\nJenis Kelamin\t: %s",&mhs[i].jk);
            printf ("\nTahun Lahir\t: %s",&mhs[i].tahunlahir);
            printf ("\nTahun Masuk\t: %s",&mhs[i].tahunmasuk);
            printf ("\nIPK\t\t: %s\n", mhs[i].ipk);

            fprintf(file,"%s,%s,%s,%s,%s,%s\n", mhs[i].nim, mhs[i].nama,
mhs[i].jk, mhs[i].tahunlahir, mhs[i].tahunmasuk, mhs[i].ipk);
            j++;
        }
        printf("\n");
        fclose(file);
    }
    FILE* file = fopen("Database Mahasiswa.csv", "r");

    if (!file)
    {
        printf("Belum Ada File Tersimpan\n\n");
    }

    else
    {
        char buffer[1024];
        int row = 1;
        int column = 0;

        while (fgets(buffer, 1024, file))
        {
            column = 0;
            row++;
        }
    }
}

```

```

        if(row == 1)
            continue;

        char* value = strtok(buffer, ", ");

        while (value) {
            if (column == 0) {
                printf("NIM \t\t:");
            }

            if (column == 1) {
                printf("\nNama \t\t:");
            }

            if (column == 2) {
                printf("\nJenis Kelamin \t:");
            }

            if (column == 3) {
                printf("\nTahun Masuk \t:");
            }

            if (column == 4) {
                printf("\nTahun Lahir \t:");
            }

            if (column == 5) {
                printf("\nIPK \t\t:");
            }

            printf(" %s", value);
            value = strtok(NULL, ",");
            column++;
        }
        printf("\n");
    }
    fclose(file);
}

void hapusdata(){
    FILE* file = fopen("Database Mahasiswa.csv", "a");
    int i, j;
    char pilih;
    if (savedata == 0)
    {
        cout << "Belum Ada Data Tersimpan\n";
        cout << "Silahkan Input Data Terlebih Dahulu!" << endl;
    }
    else
    {
        system ("cls");
        char hapusdata[11];
        cout << "Masukkan Data NIM Yang Ingin Dihapus : ";
        cin >> hapusdata;
        for(i=0; i<savedata; i++)
        {
            if(strcmp(hapusdata, mhs[i].nim) ==0)

```

```

    {
        system ("cls");
        cout << "NIM\t\t: " << mhs[i].nim << "\n";
        cout << "Nama\t\t: " << mhs[i].nama << "\n";
        cout << "Jenis Kelamin\t: " << mhs[i].jk << "\n";
        cout << "Tahun Lahir\t: " << mhs[i].tahunlahir << "\n";
        cout << "Tahun Masuk\t: " << mhs[i].tahunmasuk << "\n";
        cout << "IPK\t\t: " << mhs[i].ipk << endl;

        mhs[i] = mhs[i+1];
        cout << "\nData Berhasil Di Hapus" << endl;

        savedata--;
        break;
    }
    for(int i=0; i<savedata; i++)
    {
        if(mhs[i].nim != hapusdata)
        {
            system ("cls");
            cout << "Maaf NIM Yang Anda Tidak Valid\n";
            cout << "Tidak Ada Data Yang Berhasil Dihapus!"
<< endl;
        }
    }
}

void caridata(){
    FILE* file = fopen("Database Mahasiswa.csv", "r");
    system("cls");
    char carinim[11];
    int i,j;
    char pilihan;
    j=1;
    if (savedata == 0)
    {
        cout << "Belum Ada Data Tersimpan\n";
        cout << "Silahkan Input Data Terlebih Dahulu!" << endl;
    }
    else
    {
        cout << "Masukkan Data NIM Yang Ingin Dicari: ";
        cin >> carinim;
        for(i=0; i<savedata; i++)
        {
            if(strcmp(mhs[i].nim, carinim)==0)
            {
                system ("cls");
                cout <<"NIM Berhasil Ditemukan" << endl;
                cout << "NIM\t\t: " << mhs[i].nim << endl;
                cout << "Nama\t\t: " << mhs[i].nama << endl;
                cout << "..." << endl;
                cout <<"Ingin Menampilkan Data Lengkap?(Y/N): ";

                cin >> pilihan;
                if(pilihan == 'y' || 'Y')
                {

```



```

        system ("cls");
        cout << "NIM\t\t: " << mhs[i].nim << "\n";
        cout << "Nama\t\t: " << mhs[i].nama << "\n";
        cout << "Jenis Kelamin\t: " << mhs[i].jk << "\n";
        cout << "Tahun Lahir\t: " << mhs[i].tahunlahir <<
"\n";
        cout << "Tahun Masuk\t: " << mhs[i].tahunmasuk <<
"\n";
        cout << "IPK\t\t: " << mhs[i].ipk << endl;
    }
    else
    {
        system("cls");
        menu();
    }
    break;
}
else if(mhs[i].nim != carinim)
{
    system ("cls");
    cout << "Maaf NIM Yang Anda Cari Tidak Dapat
Ditemukan\n";
    cout << "Silahkan Coba Lagi!" << endl;
}
}
}

void editdata(){
    FILE* file = fopen("Database Mahasiswa.csv", "a");
    system("cls");
    int i;
    if (savedata == 0)
    {
        cout << "Belum Ada Data Tersimpan\n";
        cout << "Silahkan Input Data Terlebih Dahulu!" << endl;
    }
    else
    {
        char edit[11];
        cout << "Masukkan Data NIM Yang Ingin Diedit : ";
        cin >> edit;
        for(i=0; i<savedata; i++)
        {
            if(strcmp(mhs[i].nim, edit) ==0)
            {
                system ("cls");
                cout << "NIM Tervalidasi\n";
                cout << "Silahkan Edit Data\n\n";
                cout << "NIM\t\t: " << mhs[i].nim << "\n";
                cout << "Nama\t\t: ";
                gets(mhs[i].nama);
                gets(mhs[i].nama);
                cout << "Jenis Kelamin\t: ";
                gets(mhs[i].jk);
                cout << "Tahun Lahir\t: ";
                cin >> mhs[i].tahunlahir;
                cout << "Tahun Masuk\t: ";
                cin >> mhs[i].tahunmasuk;
            }
        }
    }
}

```

```

        input_ipk:
        cout << "IPK\t\t: ";
        cin >> mhs[i].ipk;
        cout << "\nData Berhasil Diubah" << endl;
        break;
    }

    else if(edit != mhs[i].nim)
    {
        system("cls");
        cout << "Maaf NIM Yang Anda Masukkan Tidak
Valid\n";

        cout << "Tidak Ada Data Yang Berhasil Di Edit" <<
endl;
    }
}
}

int main(){
    awal:
    menu();
    switch(pilih)
    {
        case 1:
        {
            system("cls");
            cout << "INPUT/TAMBAH DATA MAHASISWA" << endl;
            inputdata();
            cout << "\nTekan ENTER untuk kembali...";
            getch();
            system("cls");
            goto awal;
        }

        case 2:
        {
            system("cls");
            cout << "EDIT DATA MAHASISWA" << endl;
            editdata();
            cout << "\nTekan ENTER untuk kembali...";
            getch();
            system("cls");
            goto awal;
        }

        case 3:
        {
            system("cls");
            cout << "HAPUS DATA MAHASISWA" << endl;
            hapusdata();
            cout << "\nTekan ENTER untuk kembali...";
            getch();
            system("cls");
            goto awal;
        }

        case 4:
        {
            system("cls");
            cout << "MENCARI DATA MAHASISWA" << endl;
            caridata();

```

```

        cout << "\nTekan ENTER untuk kembali...";
        getch();
        system("cls");
        goto awal;
    }
    case 5:
    {
        system("cls");
        cout << "MENAMPILKAN SEMUA DATA MAHASISWA" << endl <<

endl;

        lihatdata();
        cout << "\nTekan ENTER untuk kembali...";
        getch();
        system("cls");
        goto awal;
    }
    case 6:
    {
        system("cls");
        int exit;
        system("cls");
        cout << "Apakah Anda Yakin Keluar Dari Program Ini?

(Y/N) : ";

        scanf ("%s", &exit);
        if(exit=='y' || exit=='Y')
        {
            system("cls");
            cout << "Terimakasih telah menggunakan program

ini";

        }
        else
        {
            system("cls");
            goto awal;
        }
        return 0;
    }
    default:
    {
        system("cls");
        printf("\nMaaf, pilihan yang anda masukkan tidak ada!\n\n");
        cout << "\nTekan ENTER untuk mengulang program...";
        getch();
        system("cls");
        goto awal;
    }
}
return 0;
}

```

Proses Run pada Command Prompt dan melihat File CSV yang dibuat program:

- Tampilan Pilihan Menu

```
D:\MODUL 7.exe
PERMASALAHAN MODUL KE-7
PROGRAM DATA OPERASI FILE MAHASISWA

PILIHAN MENU PROGRAM
1. Input/Tambah Data
2. Edit Data
3. Hapus Data
4. Cari Mahasiswa
5. Tampilkan Semua Data
6. Keluar

Masukkan Pilihan :
```

- Tampilan Menu Input/Tambah Data

```
D:\MODUL 7.exe
INPUT/TAMBAH DATA MAHASISWA
Masukkan Jumlah Data Yang Akan Diinput : 3

Data Ke-1
NIM      : 2115101014
Nama     : I Gede Gelgel Abdiutama
Jenis Kelamin : L
Tahun Lahir : 2002
Tahun Masuk : 2021
IPK      : 3.9

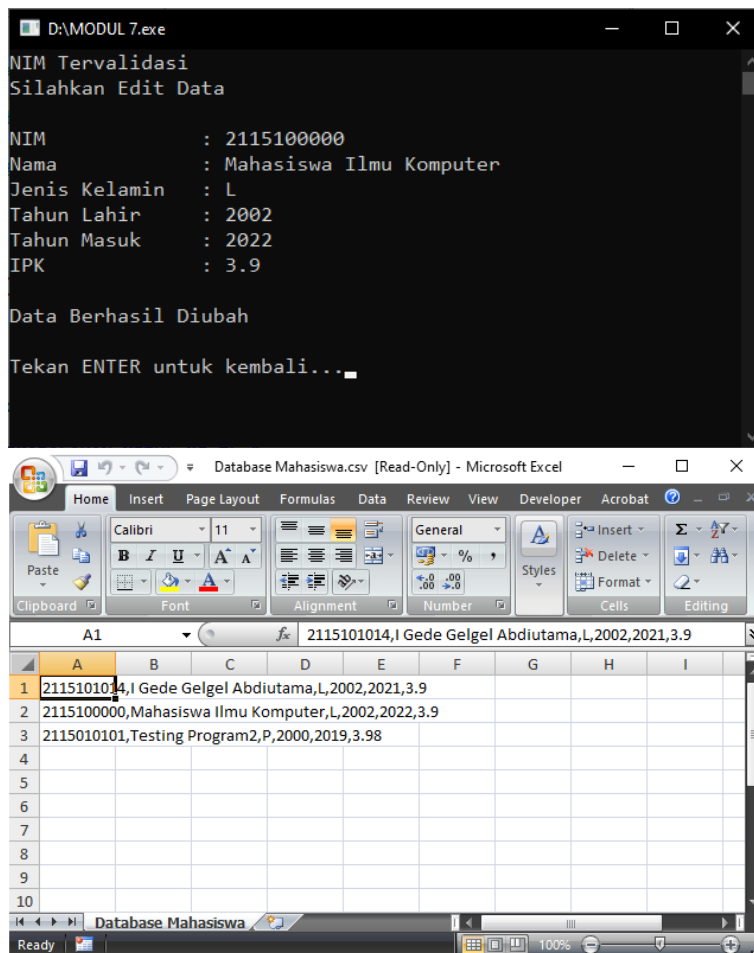
Data Ke-2
NIM      : 2115100000
Nama     : Testing Program1
Jenis Kelamin : P
Tahun Lahir : 1998
Tahun Masuk : 2016
IPK      : 4

Data Ke-3
NIM      : 2115010101
Nama     : Testing Program2
Jenis Kelamin : P
Tahun Lahir : 2000
Tahun Masuk : 2019
IPK      : 3.98

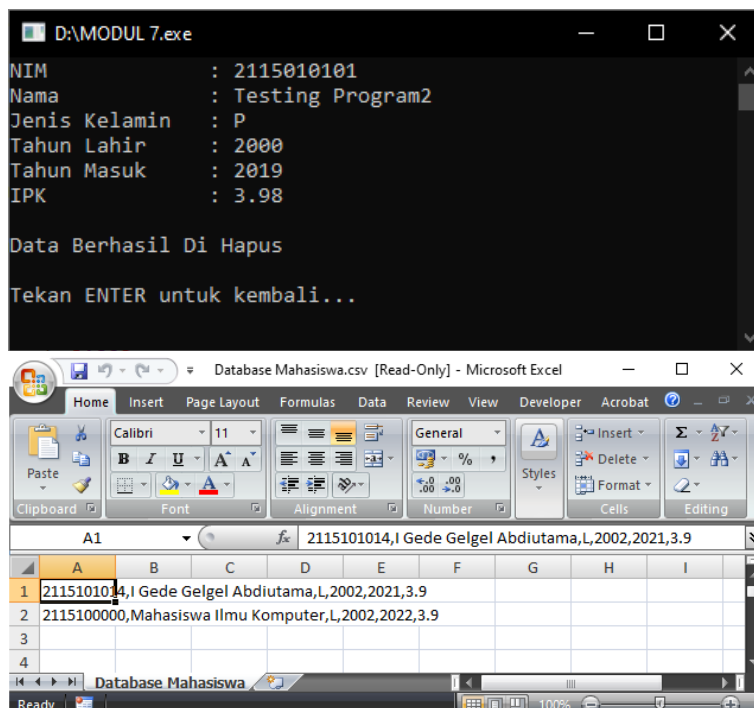
Tekan ENTER untuk kembali...
```

	A	B	C	D	E	F	G	H	I
1	2115101014	I Gede Gelgel Abdiutama	L	2002	2021	3.9			
2	2115100000	Testing Program1	P	1998	2016	4			
3	2115010101	Testing Program2	P	2000	2019	3.98			
4									
5									
6									
7									
8									
9									
10									

- Tampilan Menu Edit Data



- Tampilan Menu Hapus Data



- Tampilan Menu Cari Mahasiswa

```

D:\MODUL 7.exe
NIM Berhasil Ditemukan
NIM      : 2115101014
Nama     : I Gede Gelgel Abdiutama
...
Ingin Menampilkan Data Lengkap?(Y/N):

D:\MODUL 7.exe
NIM      : 2115101014
Nama     : I Gede Gelgel Abdiutama
Jenis Kelamin : L
Tahun Lahir : 2002
Tahun Masuk : 2021
IPK      : 3.9
Tekan ENTER untuk kembali...

```

- Tampilan Menu Tampilkan Semua Data

```

D:\MODUL 7.exe
MENAMPILKAN SEMUA DATA MAHASISWA

Data Ke-1
NIM      : 2115101014
Nama     : I Gede Gelgel Abdiutama
Jenis Kelamin : L
Tahun Lahir : 2002
Tahun Masuk : 2021
IPK      : 3.9

Data Ke-2
NIM      : 2115100000
Nama     : Mahasiswa Ilmu Komputer
Jenis Kelamin : L
Tahun Lahir : 2002
Tahun Masuk : 2022
IPK      : 3.9

```

- Tampilan Menu Keluar

```

D:\MODUL 7.exe
Apakah Anda Yakin Keluar Dari Program Ini? (Y/N) : Y

D:\MODUL 7.exe
Terimakasih telah menggunakan program ini
-----
Process exited after 417.5 seconds with return value 0
Press any key to continue . . .

```