

LAPORAN PRAKTIKUM MODUL KE-2
MEMBUAT PROGRAM PERULANGAN MENGHITUNG BILANGAN FIBONACCI
DAN PERKALIAN MATRIK



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 beberapa Studi kasus yang akan diselesaikan yaitu membuat program dengan permasalahan sebagai berikut :

- **Permasalahan ke-1:**

Buatlah program untuk menampilkan bilangan Fibonacci sampai dengan bilangan ke-n, dimana n diinputkan oleh pengguna!

- **Permasalahan ke-2:**

Buatlah program untuk mengalikan dua buah matrik! Ukuran dan isi matrik dimasukkan oleh pengguna. Matrik yang dikalikan tidak harus merupakan matrik bujur sangkar.

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. 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.

4. 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 terdefinisi sebelumnya atau pun tidak. Struktur perulangan terdiri atas dua bagian :

- Kondisi perulangan yaitu ekspresi Boolean yang harus dipenuhi untuk melaksanakan perulangan
- Isi atau badan perulangan 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.

5. Matrik

Matriks adalah himpunan skalar (bilangan riil atau kompleks) yang disusun secara empat persegi maupun persegi panjang menurut baris-baris dan kolom-kolom. Skalar – skalar itu disebut elemen matriks. Untuk pembatasnya biasanya diberikan tanda [] atau () atau || ||. Notasi Matriks merupakan sebuah matriks yang diberi variabel A memiliki elemen-elemen baris dan kolom. Beberapa jenis matriks yang dipakai dalam proses pembuatan sistem dapat dibagi sebagai berikut:

1. Matriks Persegi (Bujur Sangkar)
2. Matriks Satuan (Identitas)

6. Metode 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  
}
```

7. Metode Pemrograman

Dalam proses pembuatan program ini saya menggunakan Bahasa Pemrograman C, dan memanfaatkan fungsi array hingga percabangan yaitu if else dan juga perulangan for khususnya nested for loop, karena untuk membuat program perhitungan perkalian matrik

ini membutuhkan alur program yang dapat menyimpan dalam bentuk matrik dan memilih kondisi dari apa yang diinputkan user serta memerlukan mekanisme perulangan agar program ini dapat bekerja sesuai dengan perhitungan perkalian matrik yang benar.

C. PEMBAHASAN

Dalam pembahasan ini, akan menjelaskan proses atau tahapan yang dilakukan pada saat melakukan Praktikum membuat Program Fibonacci dan Perkalian Matrik. Namun sebelum membahas mengenai langkah-langkah teknis pembuatan, akan dijelaskan terlebih dahulu bagaimana program ini dapat bekerja. Pada praktikum ini diberikan 2 buah permasalahan yaitu :

- **Permasalahan ke-1:**

Buatlah program untuk menampilkan bilangan Fibonacci sampai dengan bilangan ke- n , dimana n diinputkan oleh pengguna!

- **Permasalahan ke-2:**

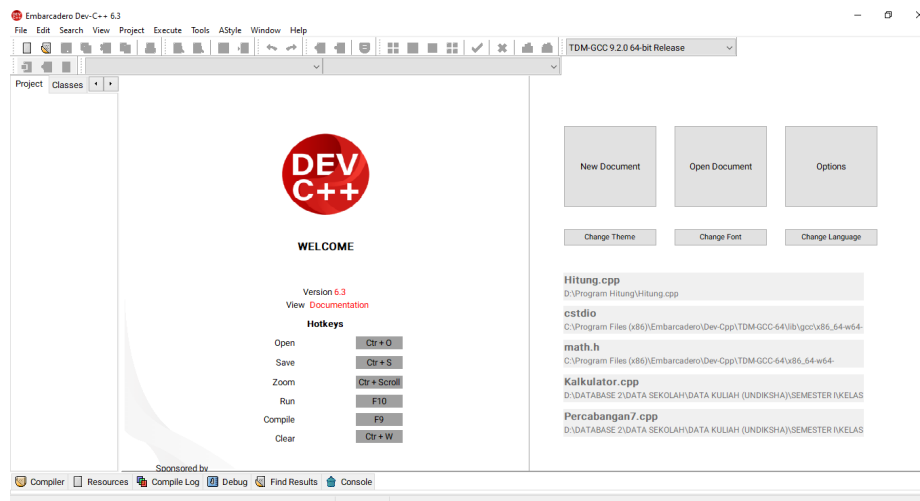
Buatlah program untuk mengalikan dua buah matrik! Ukuran dan isi matrik dimasukkan oleh pengguna. Matrik yang dikalikan tidak harus merupakan matrik bujur sangkar.

Dalam program ini dibuat menggunakan bahasa pemrograman C dengan library standar bahasa C (`#include <stdio>`) yang fungsinya sebagai header yang menampung kumpulan library yang ada sehingga bisa kita gunakan pada program kita. Bagian body yang akan membentuk kode program ini nantinya akan menggunakan metode percabangan if else dan perulangan for untuk membuat program ini bisa berjalan disaat ada proses pemilihan kondisi saat user memasukkan berbagai macam nilai, karena adanya 2 kondisi perhitungan khususnya pada program perkalian matrik dalam mengeksekusi program.

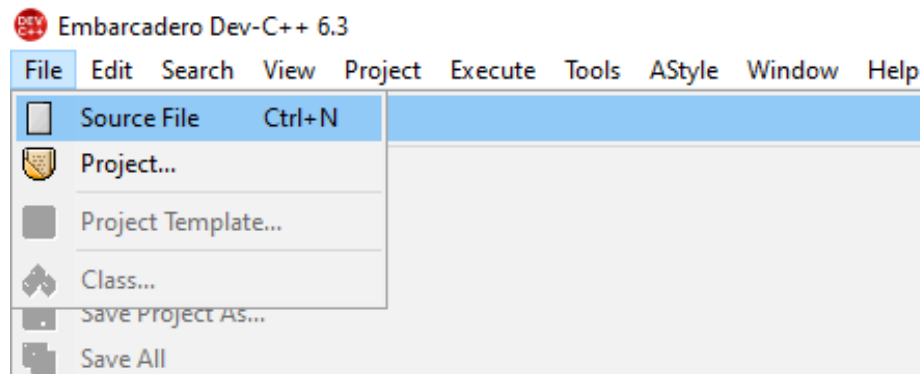
Program perhitungan Fibonacci dan perkalian matrik ini menggunakan tipe data array untuk menyimpan kumpulan data yang dapat diakses, sebagai contoh tipe data interger sesuai dengan fungsinya pada operasi bilangan, yaitu untuk menyimpan nilai bilangan bulat (negatif, 0, dan positif). Dan pada program perkalian matrik ini menggunakan nested for loop sehingga bisa membuat suatu perulangan yang bersarang. Sejumlah loop dapat didefinisikan di dalam loop lain, yaitu tidak ada batasan untuk mendefinisikan sejumlah loop.

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>
```

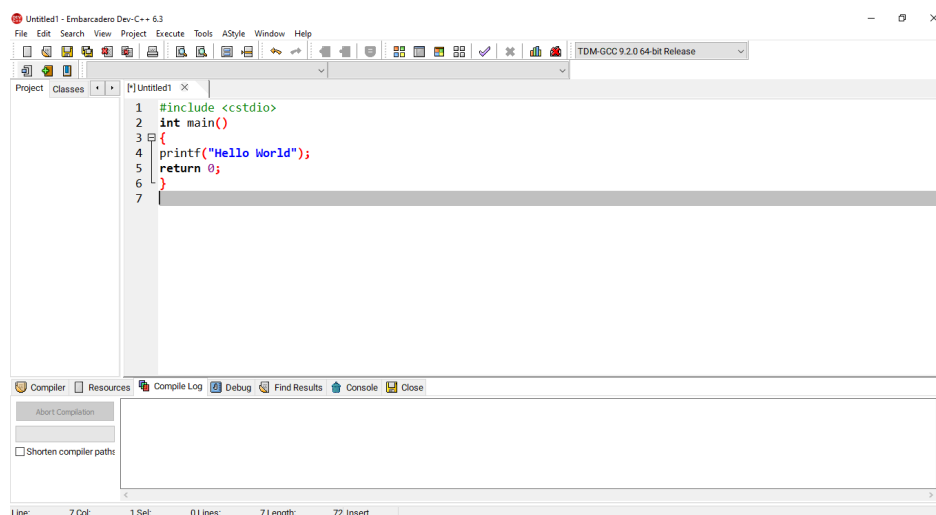
```
int main()
```

```
{
```

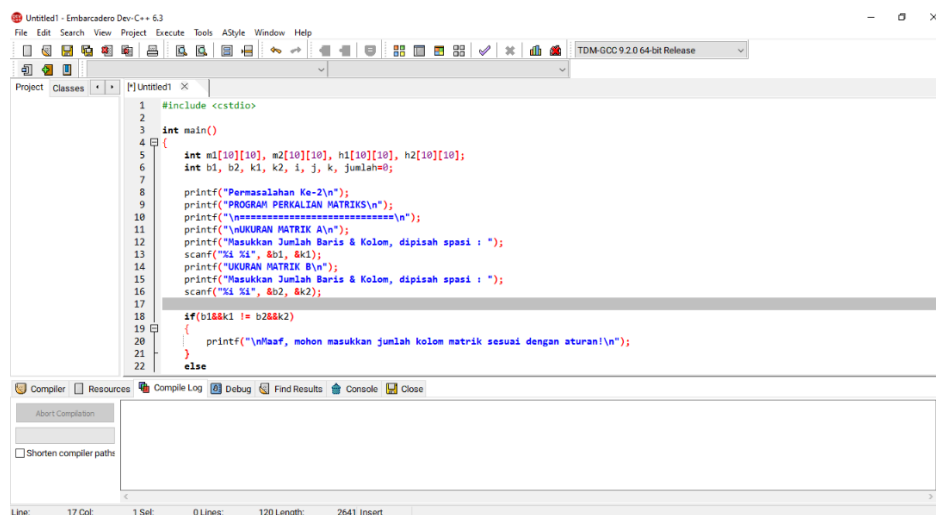
```
printf("Hello World");
```

```
return 0;
```

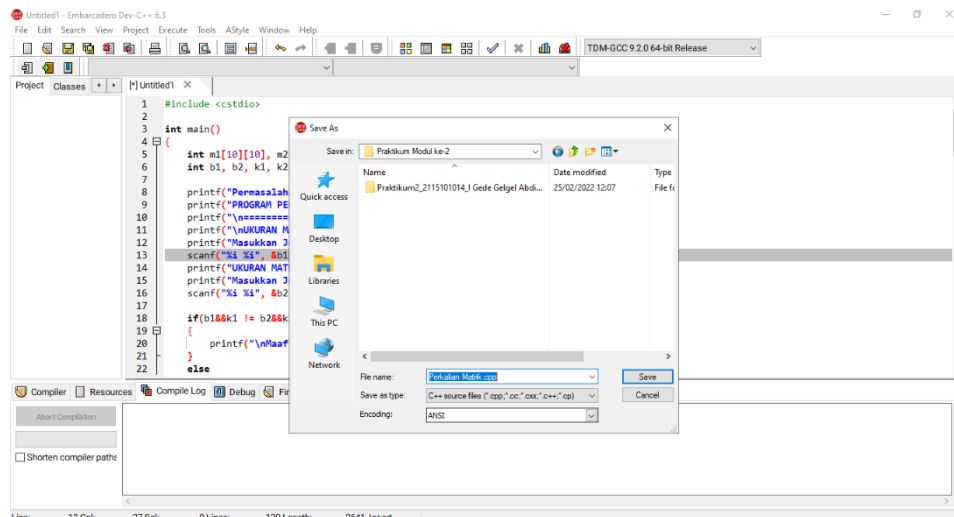
```
}
```



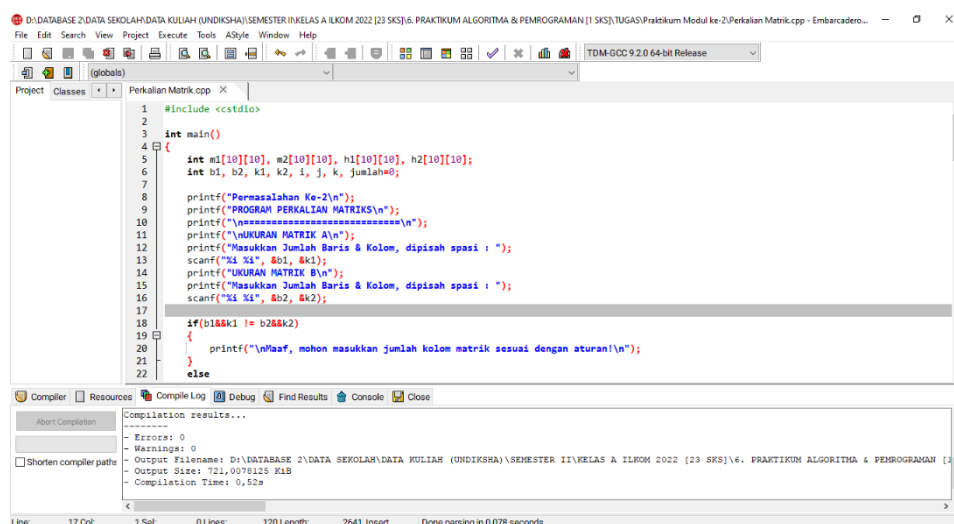
4. Pada tahap ini, kita bisa melanjutkan proses pembuatan program dengan melakukan koding sesuai dengan kode program yang akan dibuat. Untuk kode program Fibonacci maupun matrik akan dilampirkan pada halaman Lampiran paling akhir laporan ini.



5. Untuk melakukan compile sehingga program dapat dijalankan, kita klik Execute lalu pilih Compile atau bisa juga dengan menggunakan shortcut pada keyboard yaitu F9.

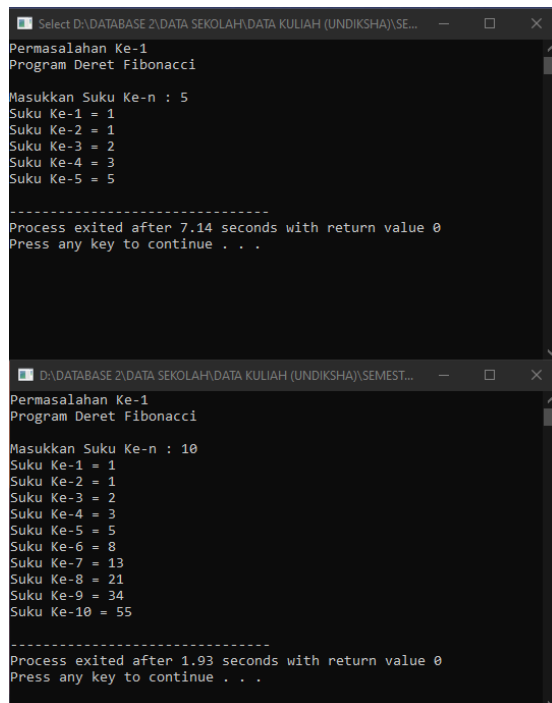


6. Berikutnya akan muncul tampilan yang meminta kita memilih lokasi untuk menyimpan file cpp sebelum di compile. Sebagai contoh disini akan saya simpan dengan nama Perkalian Matrik.cpp. Maka setelah itu hasil dari program yang telah kita buat akan di compile secara otomatis, perlu diperhatikan jika Errors 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 memasukkan inputan yang diminta sesuai dengan yang telah di set pada program yang telah dibuat lalu enter, maka di akhir akan muncul hasilnya. Sebagai contoh berikut adalah hasil eksekusi pada program Fibonacci dan Perkalian Matrik :

- Program Fibonacci



```
Permasalahan Ke-1
Program Deret Fibonacci

Masukkan Suku Ke-n : 5
Suku Ke-1 = 1
Suku Ke-2 = 1
Suku Ke-3 = 2
Suku Ke-4 = 3
Suku Ke-5 = 5

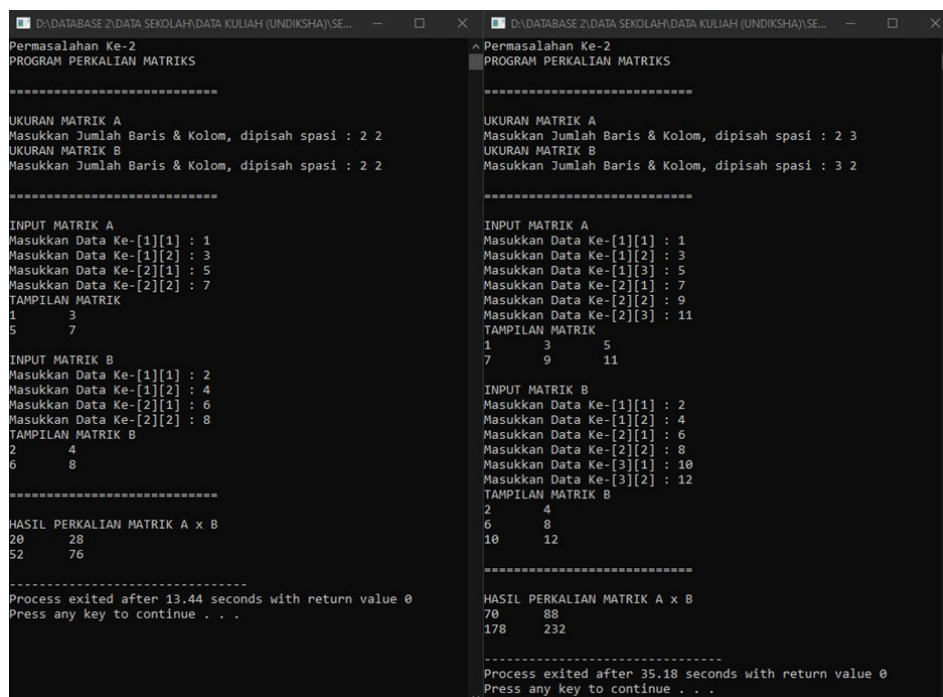
-----
Process exited after 7.14 seconds with return value 0
Press any key to continue . . .

Permasalahan Ke-1
Program Deret Fibonacci

Masukkan Suku Ke-n : 10
Suku Ke-1 = 1
Suku Ke-2 = 1
Suku Ke-3 = 2
Suku Ke-4 = 3
Suku Ke-5 = 5
Suku Ke-6 = 8
Suku Ke-7 = 13
Suku Ke-8 = 21
Suku Ke-9 = 34
Suku Ke-10 = 55

-----
Process exited after 1.93 seconds with return value 0
Press any key to continue . . .
```

- Program Fibonacci



```
Permasalahan Ke-2
PROGRAM PERKALIAN MATRIKS

=====
UKURAN MATRIK A
Masukkan Jumlah Baris & Kolom, dipisah spasi : 2 2
UKURAN MATRIK B
Masukkan Jumlah Baris & Kolom, dipisah spasi : 2 2

=====
INPUT MATRIK A
Masukkan Data Ke-[1][1] : 1
Masukkan Data Ke-[1][2] : 3
Masukkan Data Ke-[2][1] : 5
Masukkan Data Ke-[2][2] : 7
TAMPILAN MATRIK
1 3
5 7

INPUT MATRIK B
Masukkan Data Ke-[1][1] : 2
Masukkan Data Ke-[1][2] : 4
Masukkan Data Ke-[2][1] : 6
Masukkan Data Ke-[2][2] : 8
TAMPILAN MATRIK B
2 4
6 8

=====
HASIL PERKALIAN MATRIK A x B
20 28
52 76

-----
Process exited after 13.44 seconds with return value 0
Press any key to continue . . .

Permasalahan Ke-2
PROGRAM PERKALIAN MATRIKS

=====
UKURAN MATRIK A
Masukkan Jumlah Baris & Kolom, dipisah spasi : 2 3
UKURAN MATRIK B
Masukkan Jumlah Baris & Kolom, dipisah spasi : 3 2

=====
INPUT MATRIK A
Masukkan Data Ke-[1][1] : 1
Masukkan Data Ke-[1][2] : 3
Masukkan Data Ke-[1][3] : 5
Masukkan Data Ke-[2][1] : 7
Masukkan Data Ke-[2][2] : 9
Masukkan Data Ke-[2][3] : 11
TAMPILAN MATRIK A
1 3 5
7 9 11

INPUT MATRIK B
Masukkan Data Ke-[1][1] : 2
Masukkan Data Ke-[1][2] : 4
Masukkan Data Ke-[2][1] : 6
Masukkan Data Ke-[2][2] : 8
Masukkan Data Ke-[3][1] : 10
Masukkan Data Ke-[3][2] : 12
TAMPILAN MATRIK B
2 4
6 8
10 12

=====
HASIL PERKALIAN MATRIK A x B
70 88
178 232

-----
Process exited after 35.18 seconds with return value 0
Press any key to continue . . .
```

D. KESIMPULAN

Pembuatan program menghitung bilangan Fibonacci dan juga perkalian matrik ini kita perlu memahami bagaimana proses operasi daripada Fibonacci dan matrik itu sendiri, mulai dari alur perhitungannya hingga prosesnya seperti apa, sehingga dapat mengetahui apa saja yang akan digunakan atau dan dipakai di dalam kode programnya, pada program ini menggunakan sistem percabangan if-else dan juga perulangan for untuk membuat program ini bisa berjalan disaat ada proses pemilihan kondisi saat user memasukkan berbagai macam nilai. Program ini akan dieksekusi langsung melalui compiler Dev C++. Program ini dibuat secara sederhana dengan beberapa fungsi yang ada seperti user bisa memasukkan sendiri nilai/angka berapa yang user inginkan, ini dapat mempermudah user dalam menggunakan program ini.

E. LAMPIRAN

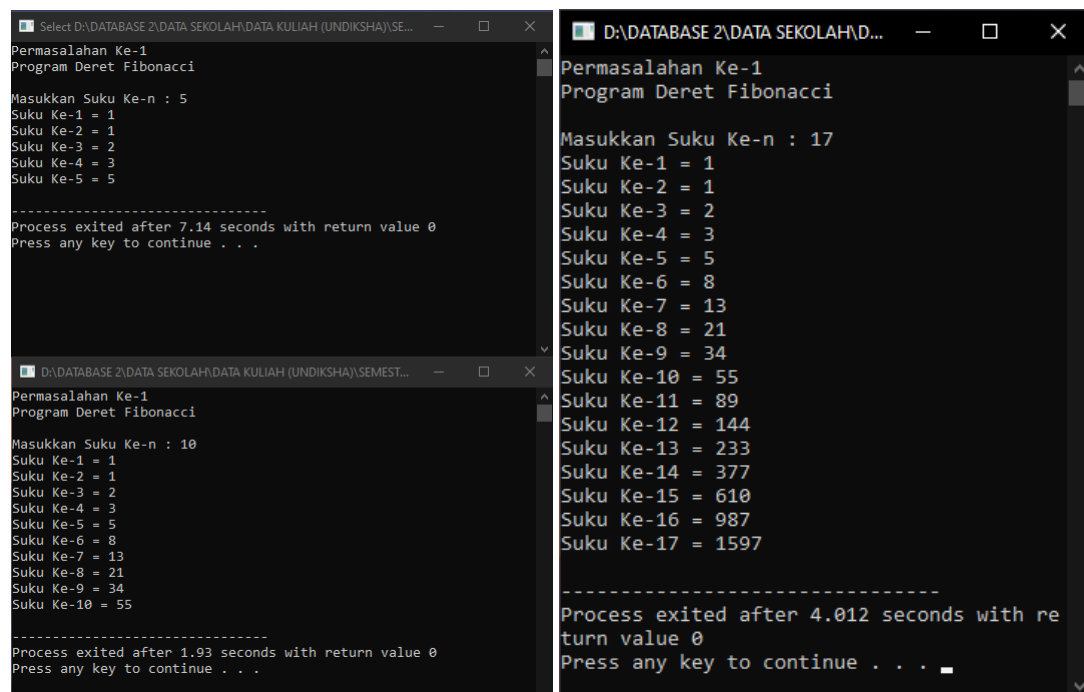
Kode Program Fibonacci:

```
#include <stdio>

int main()
{
    int Sn, hasil, i;
    int S0 = 0, S1 = 1;

    printf("Permasalahan Ke-1\n");
    printf("Program Deret Fibonacci\n\n");
    printf("Masukkan Suku Ke-n : ");
    scanf("%i", &hasil);
    printf("Suku Ke-%i = %i\n", S1, S1);
    for(i = 1 ; i < hasil ; i++)
    {
        Sn = S1 + S0;
        S0 = S1;
        S1 = Sn;
        printf("Suku Ke-%i = %i\n", i+1, Sn);
    }
    return 0;
}
```

Proses Run pada Command Prompt :



```
Permasalahan Ke-1
Program Deret Fibonacci
Masukkan Suku Ke-n : 5
Suku Ke-1 = 1
Suku Ke-2 = 1
Suku Ke-3 = 2
Suku Ke-4 = 3
Suku Ke-5 = 5
-----
Process exited after 7.14 seconds with return value 0
Press any key to continue . . .

Permasalahan Ke-1
Program Deret Fibonacci
Masukkan Suku Ke-n : 10
Suku Ke-1 = 1
Suku Ke-2 = 1
Suku Ke-3 = 2
Suku Ke-4 = 3
Suku Ke-5 = 5
Suku Ke-6 = 8
Suku Ke-7 = 13
Suku Ke-8 = 21
Suku Ke-9 = 34
Suku Ke-10 = 55
-----
Process exited after 1.93 seconds with return value 0
Press any key to continue . . .

Permasalahan Ke-1
Program Deret Fibonacci
Masukkan Suku Ke-n : 17
Suku Ke-1 = 1
Suku Ke-2 = 1
Suku Ke-3 = 2
Suku Ke-4 = 3
Suku Ke-5 = 5
Suku Ke-6 = 8
Suku Ke-7 = 13
Suku Ke-8 = 21
Suku Ke-9 = 34
Suku Ke-10 = 55
Suku Ke-11 = 89
Suku Ke-12 = 144
Suku Ke-13 = 233
Suku Ke-14 = 377
Suku Ke-15 = 610
Suku Ke-16 = 987
Suku Ke-17 = 1597
-----
Process exited after 4.012 seconds with re
turn value 0
Press any key to continue . . .
```

Kode Program Perkalian Matrik:

```
#include <stdio>

int main()
{
    int m1[10][10], m2[10][10], h1[10][10], h2[10][10];
    int b1, b2, k1, k2, i, j, k, jumlah=0;

    printf("Permasalahan Ke-2\n");
    printf("PROGRAM PERKALIAN MATRIKS\n");
    printf("\n===== \n");
    printf("\nUKURAN MATRIK A\n");
    printf("Masukkan Jumlah Baris & Kolom, dipisah spasi : ");
    scanf("%i %i", &b1, &k1);
    printf("UKURAN MATRIK B\n");
    printf("Masukkan Jumlah Baris & Kolom, dipisah spasi : ");
    scanf("%i %i", &b2, &k2);

    if(b1&&k1 != b2&&k2)
    {
        printf("\nMaaf, mohon masukkan jumlah kolom matrik sesuai dengan aturan!\n");
    }
    else
    {
        printf("\n===== \n");
        printf("\nINPUT MATRIK A\n");
        for(i = 0; i < b1; i++)
        {
            for(j = 0; j < k1; j++)
            {
                printf("Masukkan Data Ke-[%i][%i] : ", i+1, j+1);
                scanf("%i", &m1[i][j]);
            }
        }
        printf("TAMPILAN MATRIK A\n");
        for (int i=0; i<b1; i++ )
        {
            for (int j=0; j<k1; j++)
            {
                printf("%i\t", m1[i][j]);
            }
            printf("\n");
        }
        printf("\nINPUT MATRIK B\n");
        for(i = 0; i < b2; i++)
        {
            for(j = 0; j < k2; j++)
            {
                printf("Masukkan Data Ke-[%i][%i] : ", i+1, j+1);
                scanf("%i", &m2[i][j]);
            }
        }
        printf("TAMPILAN MATRIK B\n");
        for (int i=0; i<b2; i++ )
        {
            for (int j=0; j<k2; j++)
            {
```

```

        printf("%i\t", m2[i][j]);
    }
    printf("\n");
}

if(k1==b2)
{
    for(i = 0; i < b1; i++)
    {
        for(j = 0; j < k2; j++)
        {
            for(k = 0; k < k1; k++)
            {
                jumlah = jumlah + m1[i][k] * m2[k][j];
            }
            h1[i][j] = jumlah;
            jumlah = 0;
        }
    }
}
else if((b1==k1)&&(b2==k2)&&(b1==b2))
{
    for(i = 0; i < b1; i++)
    {
        for(j = 0; j < k2; j++)
        {
            for(k = 0; k < b2; k++)
            {
                jumlah = jumlah + m1[i][k] * m2[k][j];
            }
            h2[i][j] = jumlah;
            jumlah = 0;
        }
    }
    return 0;
}

if(k1==b2)
{
    printf("\n===== \n");
    printf("\nHASIL PERKALIAN MATRIK A x B\n");
    for(i = 0; i < b1; i++)
    {
        for(j = 0; j < k2; j++)
        {
            printf("%i\t", h1[i][j]);
        }
        printf("\n");
    }
}
else if((b1==k1)&&(b2==k2)&&(b1==b2))
{
    for(i = 0; i < b1; i++)
    {
        for(j = 0; j < k1; j++)
        {
            printf("%i\t", h2[i][j]);
        }
    }
}

```

```

        printf("\n\n");
    }
}
return 0;
}

```

Proses Run pada Command Prompt :

```

D:\DATABASE 2\DATA SEKOLAH\DATA KULIAH (UNDIKSHA)\SE...  D:\DATABASE 2\DATA SEKOLAH\DATA KULIAH (UNDIKSHA)\SE...
Permasalahan Ke-2                                         Permasalahan Ke-2
PROGRAM PERKALIAN MATRIKS                                  PROGRAM PERKALIAN MATRIKS
=====
UKURAN MATRIK A                                           UKURAN MATRIK A
Masukkan Jumlah Baris & Kolom, dipisah spasi : 2 2      Masukkan Jumlah Baris & Kolom, dipisah spasi : 2 3
UKURAN MATRIK B                                           UKURAN MATRIK B
Masukkan Jumlah Baris & Kolom, dipisah spasi : 2 2      Masukkan Jumlah Baris & Kolom, dipisah spasi : 3 2
=====
INPUT MATRIK A                                           INPUT MATRIK A
Masukkan Data Ke-[1][1] : 1                               Masukkan Data Ke-[1][1] : 1
Masukkan Data Ke-[1][2] : 3                               Masukkan Data Ke-[1][2] : 3
Masukkan Data Ke-[2][1] : 5                               Masukkan Data Ke-[1][3] : 5
Masukkan Data Ke-[2][2] : 7                               Masukkan Data Ke-[2][1] : 7
TAMPILAN MATRIK                                           Masukkan Data Ke-[2][2] : 9
1      3                                                    Masukkan Data Ke-[2][3] : 11
5      7                                                    TAMPILAN MATRIK
                                           1      3      5
                                           7      9      11
INPUT MATRIK B                                           INPUT MATRIK B
Masukkan Data Ke-[1][1] : 2                               Masukkan Data Ke-[1][1] : 2
Masukkan Data Ke-[1][2] : 4                               Masukkan Data Ke-[1][2] : 4
Masukkan Data Ke-[2][1] : 6                               Masukkan Data Ke-[2][1] : 6
Masukkan Data Ke-[2][2] : 8                               Masukkan Data Ke-[2][2] : 8
TAMPILAN MATRIK B                                           Masukkan Data Ke-[3][1] : 10
2      4                                                    Masukkan Data Ke-[3][2] : 12
6      8                                                    TAMPILAN MATRIK B
=====
HASIL PERKALIAN MATRIK A x B                               2      4
20     28                                                    6      8
52     76                                                    10     12
=====
Process exited after 13.44 seconds with return value 0
Press any key to continue . . .
=====
HASIL PERKALIAN MATRIK A x B                               70     88
178    232
=====
Process exited after 35.18 seconds with return value 0
Press any key to continue . . .

```