

**LAPORAN PRAKTIKUM MODUL KE-10
MENGATUR KECERAHAN GAMBAR BITMAP
MENGUNAKAN OPERASI FILE BINARY**



**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**

A. PERMASALAHAN

Buatlah program untuk mengolah gambar dengan format file BMP Windows 24 bit. Program dapat membuat gambar yang lebih terang dari gambar aslinya.

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. 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  
}
```

5. 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. Adapun beberapa mode didalam operasi file yang dapat digunakan yaitu sebagai berikut :

MODE	KETERANGAN
r	Untuk membuka file teks dalam mode membaca
w	Untuk membuka atau membuat file teks dalam mode penulisan
a	Untuk membuka file teks dalam mode tambahkan
r+	Untuk membuka file teks dalam mode membaca dan menulis
w+	Untuk membuka file teks dalam mode membaca dan menulis
a+	Untuk membuka file teks dalam mode membaca dan menulis
rb	Untuk membuka file biner dalam mode membaca
wb	Untuk membuka atau membuat file biner dalam mode penulisan
ab	Untuk membuka file biner dalam mode tambahkan
rb+	Untuk membuka file biner dalam mode membaca dan menulis
wb+	Untuk membuka file biner dalam mode membaca dan menulis
ab+	Untuk membuka file biner dalam mode membaca dan menulis

6. Ruang Warna RGB

Ruang warna RGB adalah ruang warna tambahan apa pun yang didasarkan pada model warna RGB. Ruang warna RGB ditentukan oleh koordinat kromatisitas warna primer aditif merah, hijau, dan biru , titik putih yang biasanya merupakan iluminan standar , dan fungsi transfer yang juga dikenal sebagai kurva respons nada (TRC) atau gamma. Menerapkan hukum Grassmannaditivitas cahaya, ruang warna yang ditentukan dapat menghasilkan warna yang diapit oleh segitiga 2D pada diagram kromatisitas yang

ditentukan oleh koordinat primer tersebut. TRC dan titik putih selanjutnya menentukan kemungkinan warna, menciptakan volume dalam bentuk 3D yang tidak pernah melebihi batas segitiga. Mata manusia normal mengandung tiga jenis sel fotosensitif yang disebut kerucut, yang peka terhadap panjang gelombang cahaya yang umumnya kita kategorikan sebagai merah, hijau, dan biru. Ruang warna RGB menggunakan primer yang diterangi yang dipilih untuk merangsang setiap jenis kerucut sebebaskan mungkin. Dengan cara ini, pencampuran tiga lampu dalam proporsi yang berbeda dapat merangsang kerucut di mata dan menciptakan persepsi warna.

7. File BMP

Format file BMP, juga dikenal sebagai file gambar bitmap, format file bitmap device independent bitmap (DIB) dan bitmap, adalah format file gambar grafik raster yang digunakan untuk menyimpan gambar digital bitmap, terlepas dari perangkat tampilan (seperti adaptor grafik), terutama pada sistem operasi Microsoft Windows dan OS/2. Format file BMP mampu menyimpan gambar digital dua dimensi baik monokrom maupun berwarna, dalam berbagai kedalaman warna, dan opsional dengan kompresi data, saluran alfa, dan profil warna. Spesifikasi Windows Metafile (WMF) mencakup format file BMP. Adapun header dari file Bitmap ini yaitu sebagai berikut :

Offset hex	Offset dec	Size	Purpose
00	0	2 bytes	<p>The header field used to identify the BMP and DIB file is 0x42 0x4D in hexadecimal, same as BM in ASCII. The following entries are possible:</p> <p>BM Windows 3.1x, 95, NT, ... etc.</p> <p>BA OS/2 struct bitmap array</p> <p>CI OS/2 struct color icon</p> <p>CP OS/2 const color pointer</p> <p>IC OS/2 struct icon</p> <p>PT OS/2 pointer</p>
02	2	4 bytes	The size of the BMP file in bytes
06	6	2 bytes	Reserved; actual value depends on the application that creates the image, if created manually can be 0
08	8	2 bytes	Reserved; actual value depends on the application that creates the image, if created manually can be 0
0A	10	4 bytes	The offset, i.e. starting address, of the byte where the bitmap image data (pixel array) can be found.

C. PEMBAHASAN

Dalam pembahasan ini, akan menjelaskan proses atau tahapan yang dilakukan pada saat melakukan praktikum membuat program untuk mengolah gambar dengan format file BMP Windows 24 bit. Program dapat membuat gambar yang lebih terang dari gambar aslinya. Berikut bagian-bagian dari program yang telah dibuat :

1. Header



```
#include <stdio>
#include <stdlib.h>
```

Pada program ini menggunakan bahasa pemrograman C, yang dimana menggunakan standar library C dan library stdlib.h digunakan untuk bisa menggunakan perintah system("cls") didalam program untuk memperhalus/mempercantik tampilan program agar mudah untuk digunakan.

2. Operasi File & Tipe Data



```
FILE* fl;
FILE* flOut;
unsigned char varByte;
int brightness;
```

Dalam program ini menggunakan Operasi File yaitu File Binary. Tipe data yang saya gunakan yaitu unsigned char dan int. Penggunaan tipe data yang unsigned bertujuan untuk mendapatkan nilai yang tidak (-) minus. Karena didalam proses program nanti agar dapat menghasilkan nilai dari 0 – 255 sesuai rentang RGB. Berikut adalah tabel rentang nilai antara signed char dan unsigned char :

Tipe data	Rentang Nilai
signed char	-128 hingga 127
unsigned char	0 hingga 255

3. Main Program

Didalam main program ini adalah isi dari program yang dapat mengatur (menaikkan) nilai pada RGB di setiap pixel gambar sehingga hasil akhirnya diperoleh gambar yang lebih cerah dari yang sebelumnya.

```
int main()
{
    fl = fopen("Before_Edit.BMP", "rb");
    flOut = fopen("After_Edit.BMP", "wb");
    for (int i=0; i<54; i++)
    {
        fread(&varByte, 1, 1, fl);
        fwrite(&varByte, 1, 1, flOut);
    }

    printf("PERMASALAHAN MODUL KE-10");
    printf("\nOPERASI FILE BINARY (Mengatur Brightness File BMP)\n\n");
    printf("Masukkan Intensitas Cahaya (0-255) : ");
    scanf("%i", &brightness);
    while (fread(&varByte, 1, 1, fl)){
        if((varByte+brightness)>255){
            varByte = 255;
        }
        else{
            varByte = varByte+brightness;
        }
        fwrite(&varByte, 1, 1, flOut);
    }
    fclose(fl);
    fclose(flOut);
    system("cls");
    printf("OPERASI FILE BINARY (Mengatur Brightness File BMP)\n\n");
    printf("STATUS : BERHASIL\nHASIL : Brightness Gambar diatur menjadi +%i\n", brightness);
    return 0;
}
```

4. Read dan Write File BMP

```
fl = fopen("Before_Edit.BMP", "rb");
flOut = fopen("After_Edit.BMP", "wb");
for (int i=0; i<54; i++)
{
    fread(&varByte, 1, 1, fl);
    fwrite(&varByte, 1, 1, flOut);
}
```

Untuk bisa mengeksekusi gambar yang akan melewati proses edit pada program, kita harus menentukan source/tempat file BMP yang akan kita edit itu berada dimana dan destination/tujuan file BMP tersebut disimpan setelah di edit oleh program. Dalam hal ini, saya set source dan destinationnya berada dimana file program .cpp ini berada. Kemudian kita menggunakan perulangan for untuk membaca dan menulis kembali dari file BMP yang telah kita tentukan.

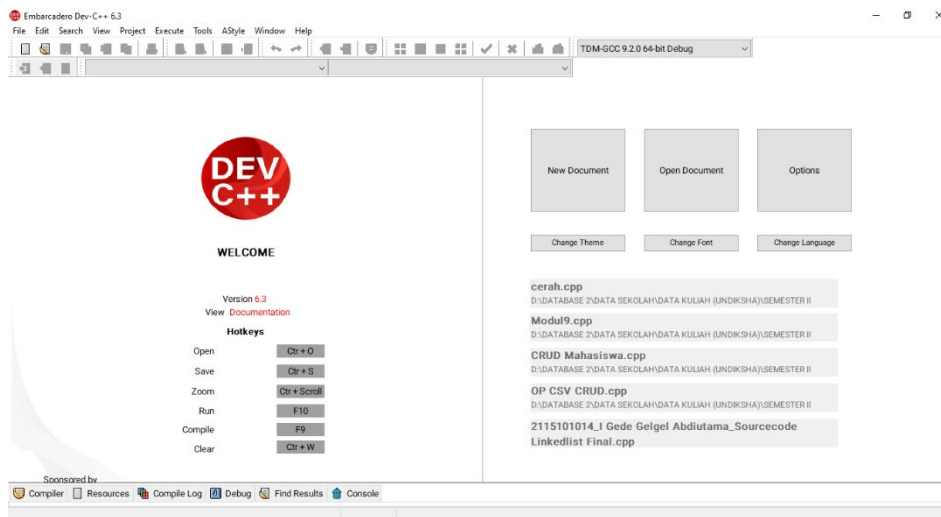
5. Set Kecerahan Gambar (Brightness)

```
printf("Masukkan Intensitas Cahaya (0-255) : ");
scanf("%i", &brightness);
while (fread(&varByte, 1, 1, fl)){
    if((varByte+brightness)>255){
        varByte = 255;
    }
    else{
        varByte = varByte+brightness;
    }
    fwrite(&varByte, 1, 1, flOut);
}
```

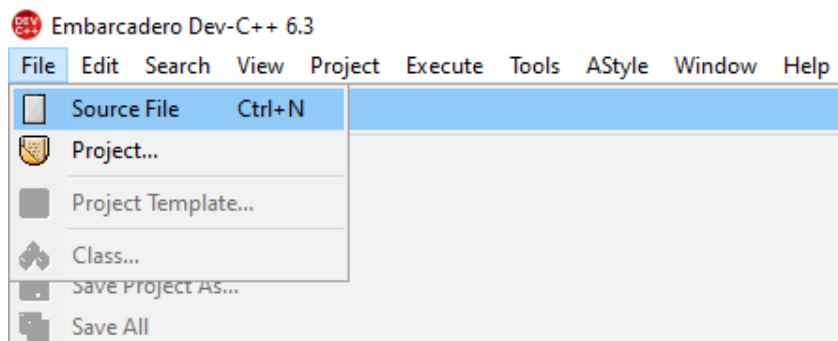
Hal terpenting dalam membuat program ini adalah dalam kita menentukan nilai dari intensitas cahaya/kecerahan yang diinginkan, dalam hal ini saya membuat seberapa banyak nilai RGB yang berdampak pada intensitas cahaya itu diatur sendiri oleh pengguna menggunakan metode Input Output. Pengguna akan memasukkan rentang nilai dari 0-255. Kemudian nilai yang telah di inputkan pengguna itu akan diproses pada perintah while, gambar akan dibaca terlebih dahulu oleh program, lalu dibandingkan dengan if else yaitu jika nilai varByte yang sudah di set brightnessnya lebih besar dari 255, sehingga output akan berhenti di 255. Dan jika kondisi itu tidak terpenuhi, maka akan memperoleh output dengan nilai baru varByte sama dengan varByte+brightness. Dan terakhir operasi file di close sehingga hasil dari gambar yang sudah melalui tahap proses edit akan ditulis ulang/disimpan dalam bentuk file bitmap yang ekstensinya adalah BMP.

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.

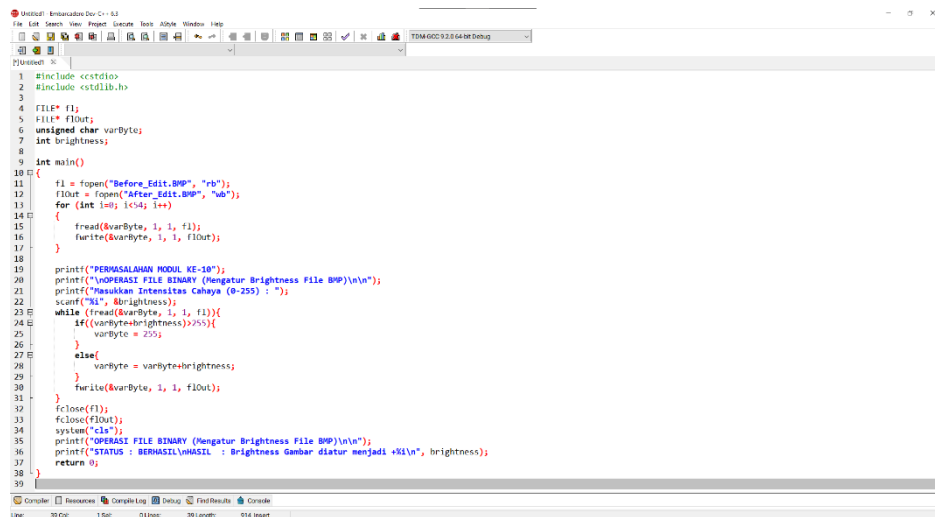


3. Kemudian kita akan eksekusi kode programnya, karena saya menggunakan bahasa pemrograman C++, maka kita membutuhkan sebuah kerangka dasarnya untuk bisa membentuk kode program lainnya, berikut merupakan header dan main program dari bahasa C

```
#include <iostream>

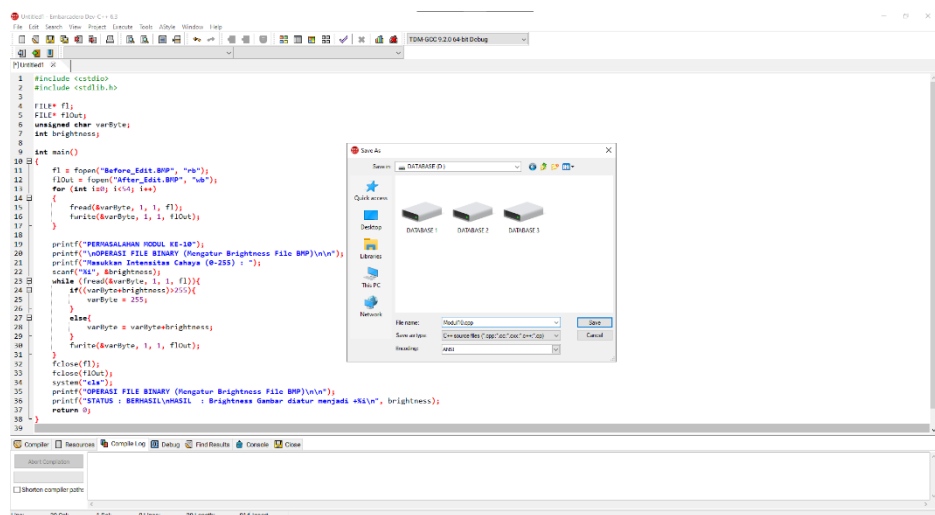
int main()
{
    std::cout << "Hello World";
    return 0;
}
```

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.

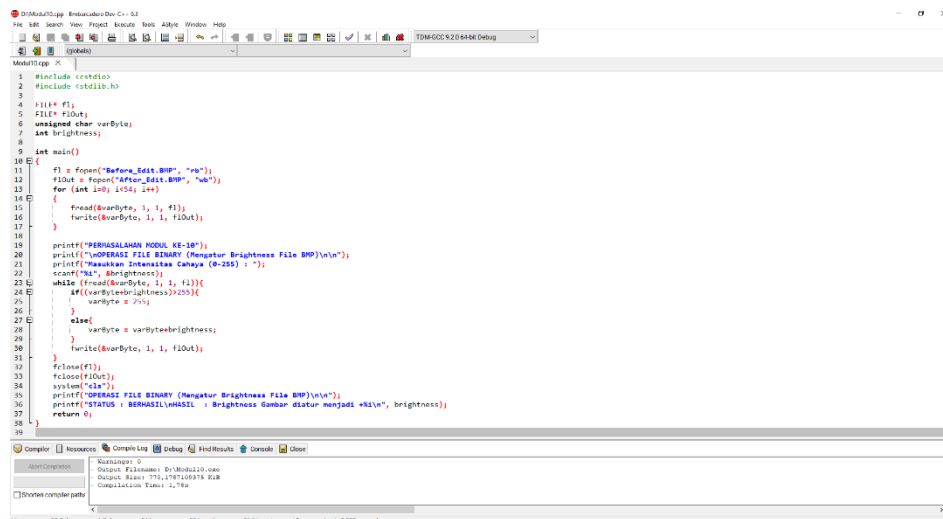


```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 FILE* fI;
5 FILE* fOut;
6 unsigned char varByte;
7 int brightness;
8
9 int main()
10 {
11     fI = fopen("Before_Edit.BMP", "rb");
12     fOut = fopen("After_Edit.BMP", "wb");
13     for (int i=0; i<54; i++)
14     {
15         fread(&varByte, 1, 1, fI);
16         fwrite(&varByte, 1, 1, fOut);
17     }
18
19     printf("PERMASALAHAN MODUL KE-10");
20     printf("\nOPERASI FILE BINARY (Mengatur Brightness File BMP)\n\n");
21     printf("Masukkan Intensitas Cahaya (0-255) : ");
22     scanf("%i", &brightness);
23     while (fread(&varByte, 1, 1, fI)){
24         if((varByte+brightness)>255){
25             varByte = 255;
26         }
27         else{
28             varByte = varByte+brightness;
29         }
30         fwrite(&varByte, 1, 1, fOut);
31     }
32     fclose(fI);
33     fclose(fOut);
34     system("cls");
35     printf("OPERASI FILE BINARY (Mengatur Brightness File BMP)\n\n");
36     printf("STATUS : BERNASIL/HAASIL : Brightness Gambar diatur menjadi %i\n", brightness);
37     return 0;
38 }
```

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 Modul10.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.



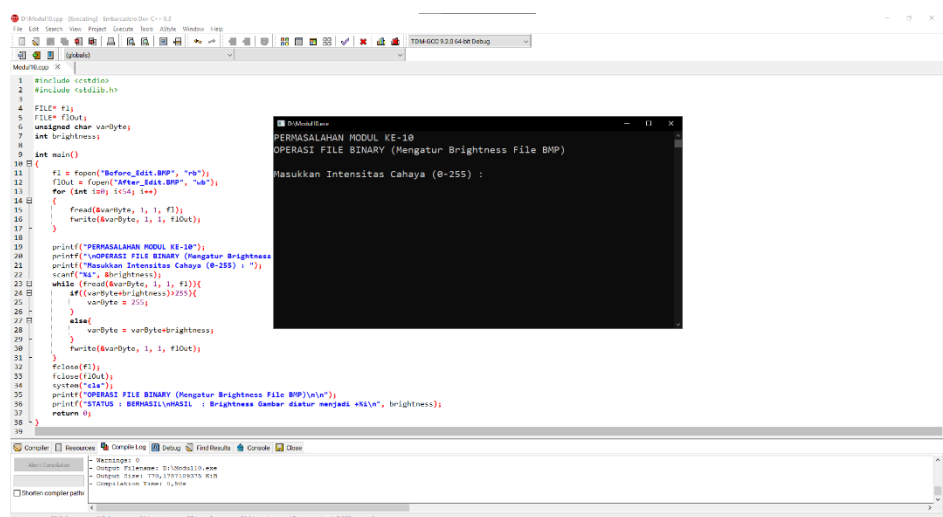
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 FILE* f1;
5 FILE* f2Out;
6 unsigned char varByte;
7 int brightness;
8
9 int main()
10 {
11     f1 = fopen("Before_Edit.BMP", "rb");
12     f2Out = fopen("After_Edit.BMP", "wb");
13     for (int i=0; i<54; i++)
14     {
15         fread(&varByte, 1, 1, f1);
16         fwrite(&varByte, 1, 1, f2Out);
17     }
18
19     printf("PERMASALAHAN MODUL KE-10");
20     printf("\nOPERASI FILE BINARY (Mengatur Brightness File BMP)\n");
21     printf("Masukkan Intensitas Cahaya (0-255) : ");
22     scanf("%d", &brightness);
23     while (fread(&varByte, 1, 1, f1))
24     {
25         if((varByte*brightness)>255){
26             varByte = 255;
27         }
28         else{
29             varByte = varByte*brightness;
30         }
31         fwrite(&varByte, 1, 1, f2Out);
32     }
33     fclose(f1);
34     fclose(f2Out);
35     printf("OPERASI FILE BINARY (Mengatur Brightness File BMP)\n");
36     printf("STATUS : BERHASIL\n");
37     return 0;
38 }
```

Compiler: g++ Resources: Compile Log Debug Find Results Console Close

Output File Name: D:\Modul10.exe
Output File: 770,197109375 KB
Compilation Time: 1,70s

Line: 38 Col: 1 Sel: 0 Lines: 38 Length: 914 Insert: Done parsing in 0.003 seconds

7. Ketika program sudah di run, maka langkah selanjutnya kita akan mengeksekusi program kita dengan cara memasukkan besar intensitas kecerahan pada gambar, maka program akan secara otomatis melakukan pemrosesan pada gambar dan menghasilkan output file BMP yang telah berhasil di edit.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 FILE* f1;
5 FILE* f2Out;
6 unsigned char varByte;
7 int brightness;
8
9 int main()
10 {
11     f1 = fopen("Before_Edit.BMP", "rb");
12     f2Out = fopen("After_Edit.BMP", "wb");
13     for (int i=0; i<54; i++)
14     {
15         fread(&varByte, 1, 1, f1);
16         fwrite(&varByte, 1, 1, f2Out);
17     }
18
19     printf("PERMASALAHAN MODUL KE-10");
20     printf("\nOPERASI FILE BINARY (Mengatur Brightness File BMP)\n");
21     printf("Masukkan Intensitas Cahaya (0-255) : ");
22     scanf("%d", &brightness);
23     while (fread(&varByte, 1, 1, f1))
24     {
25         if((varByte*brightness)>255){
26             varByte = 255;
27         }
28         else{
29             varByte = varByte*brightness;
30         }
31         fwrite(&varByte, 1, 1, f2Out);
32     }
33     fclose(f1);
34     fclose(f2Out);
35     printf("OPERASI FILE BINARY (Mengatur Brightness File BMP)\n");
36     printf("STATUS : BERHASIL\n");
37     return 0;
38 }
```

Compiler: g++ Resources: Compile Log Debug Find Results Console Close

Output File Name: D:\Modul10.exe
Output File: 770,197109375 KB
Compilation Time: 1,70s

Line: 38 Col: 1 Sel: 0 Lines: 38 Length: 914 Insert: Done parsing in 0.003 seconds

D. KESIMPULAN

Pembuatan program untuk mengolah gambar dengan format file BMP Windows 24 bit khususnya membuat program yang dapat membuat gambar yang lebih terang dari gambar aslinya ini menggunakan bahasa C. Pada program ini menggunakan operasi file binary sebagai kunci utama agar bisa melakukan pengeditan pada gambar. Untuk bisa mengatur kecerahan pada gambar agar bisa diatur kecerahannya, secara algoritma nilai RGB pada pixelnya yang ditingkatkan sesuai dengan inputan yang diberikan. Program ini dibuat sesederhana mungkin agar pengguna dapat menggunakannya dengan mudah dan menghasilkan hasil gambar yang maksimal.

E. LAMPIRAN

Kode Program:

```
#include <stdio>
#include <stdlib.h>

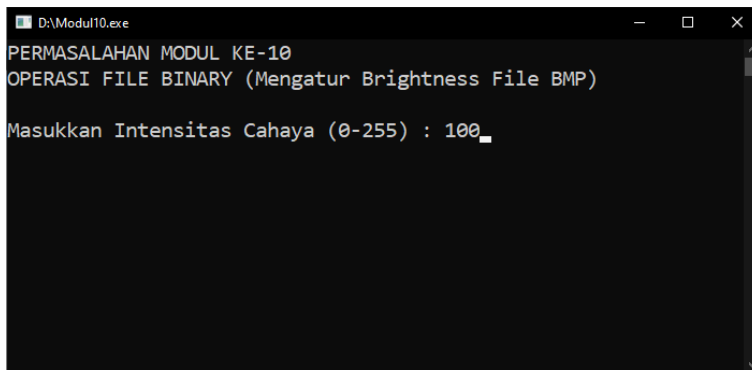
FILE* f1;
FILE* f1Out;
unsigned char varByte;
int brightness;

int main()
{
    f1 = fopen("Before_Edit.BMP", "rb");
    f1Out = fopen("After_Edit.BMP", "wb");
    for (int i=0; i<54; i++)
    {
        fread(&varByte, 1, 1, f1);
        fwrite(&varByte, 1, 1, f1Out);
    }

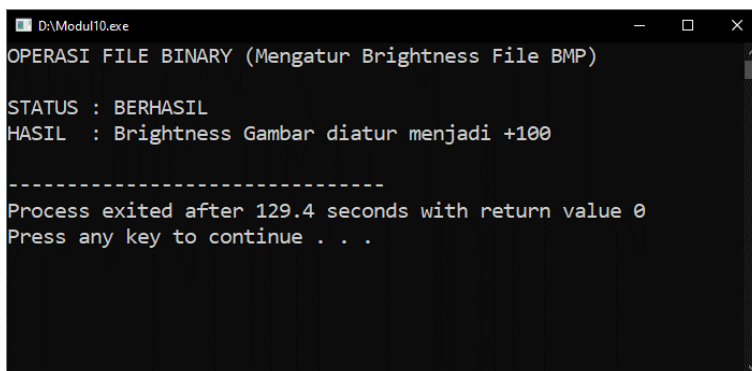
    printf("PERMASALAHAN MODUL KE-10");
    printf("\nOPERASI FILE BINARY (Mengatur Brightness File BMP)\n\n");
    printf("Masukkan Intensitas Cahaya (0-255) : ");
    scanf("%i", &brightness);
    while (fread(&varByte, 1, 1, f1)){
        if((varByte+brightness)>255){
            varByte = 255;
        }
        else{
            varByte = varByte+brightness;
        }
        fwrite(&varByte, 1, 1, f1Out);
    }
    fclose(f1);
    fclose(f1Out);
    system("cls");
    printf("OPERASI FILE BINARY (Mengatur Brightness File BMP)\n\n");
    printf("STATUS : BERHASIL\nHASIL : Brightness Gambar diatur menjadi\n%i\n", brightness);
    return 0;
}
```

Proses Run pada Command Prompt :

- Tampilan awal Program (Input Intensitas Cahaya)



- Tampilan Gambar berhasil di edit



- Before dan After Edit Kecerahan pada Gambar

