
1.1. Tujuan:

1. Mahasiswa dapat membuat program pengolahan citra menggunakan Visual C++ dengan MFC
2. Mahasiswa dapat membuat menu menggunakan Visual C++ dengan MFC.

1.2. Dasar Teori:

Image processing atau sering disebut dengan pengolahan citra digital merupakan suatu proses dari gambar asli menjadi gambar lain yang sesuai dengan keinginan kita. Misal suatu gambar yang kita dapatkan terlalu gelap maka dengan image processing gambar tersebut bisa kita proses sehingga mendapat gambar yang jelas. Secara garis besar bisa kita gambarkan seperti blok diagram pada gambar 1.1 dibawah ini:



Gambar 1.1 Blok Diagram Pengolahan Citra

1.3. Tugas Pendahuluan:

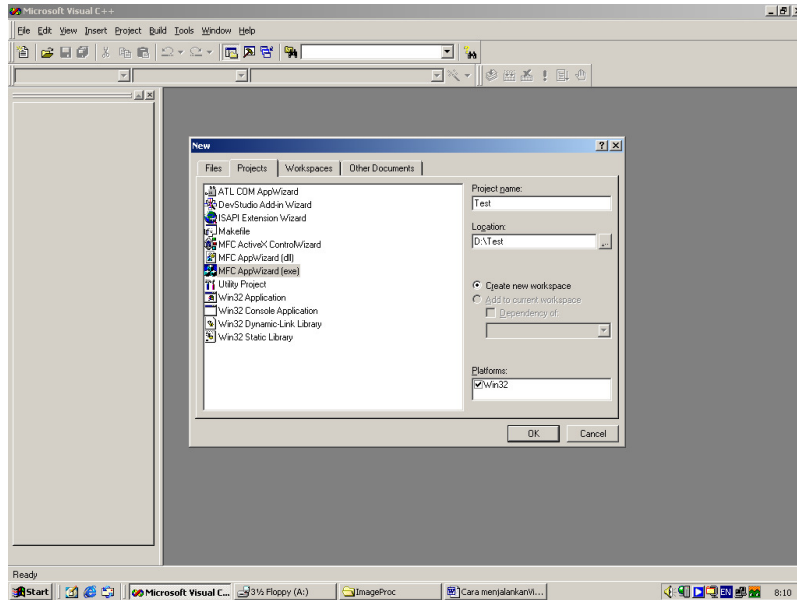
1. Tuliskan tujuan praktikum
2. Gambarkan blok diagram pengolahan citra
3. Buatlah ringkasan cara menjalankan Visual C++ dengan MFC
4. Buatlah ringkasan cara membuat menu

1.4. Percobaan:

1.4.1. Menjalankan Visual C++ dengan MFC

1. Membuka Visual C++ 6.0

- Pilih menu : Start->Programs->Microsoft Visual Studio 6.0->Microsoft Visual C++ 6.0
2. Memberi nama program
- Pilih menu : File->New->Projects->MFC AppWizard(exe)
 - Isi Project name misalnya dengan: **Test** (lihat gambar 1.2)
 - Tekan tombol OK



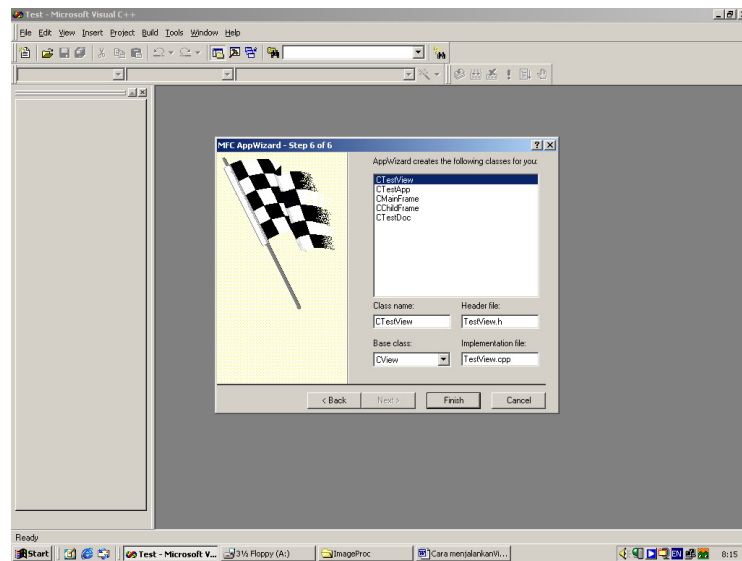
Gambar 1.2 Memberi nama program

3. Memilih isi program
- What tipe of application would you like to create (aplikasi apa yang ingin dibuat) pilih *Multiple documents* untuk pilihan standar. Tekan tombol Next
 - What database support would you like include (database apa yang ingin dibuat) pilih *None* untuk pilihan standar. Tekan tombol Next
 - What compound document support would you like include (dokumen apa yang ingin dibuat) pilih *None* untuk pilihan standar. Tekan tombol Next
 - What features would you like include

(tampilan apa yang ingin dibuat) pilih *Docking toolbar*, *Initial status bar*, *printing and print preview*, *3D control* untuk pilihan standar. Tekan tombol Next

- What style of project would you like
(gaya program apa yang ingin dibuat) pilih *MFC standard* untuk pilihan standar. Tekan tombol Next
- AppWizard creates the following classes for you
(AppWizard akan membuat class seperti dibawah ini)

CtestView
CtestApp
CmainFrame
CchildFrame
CtestDoc
Lihat gambar 1.3



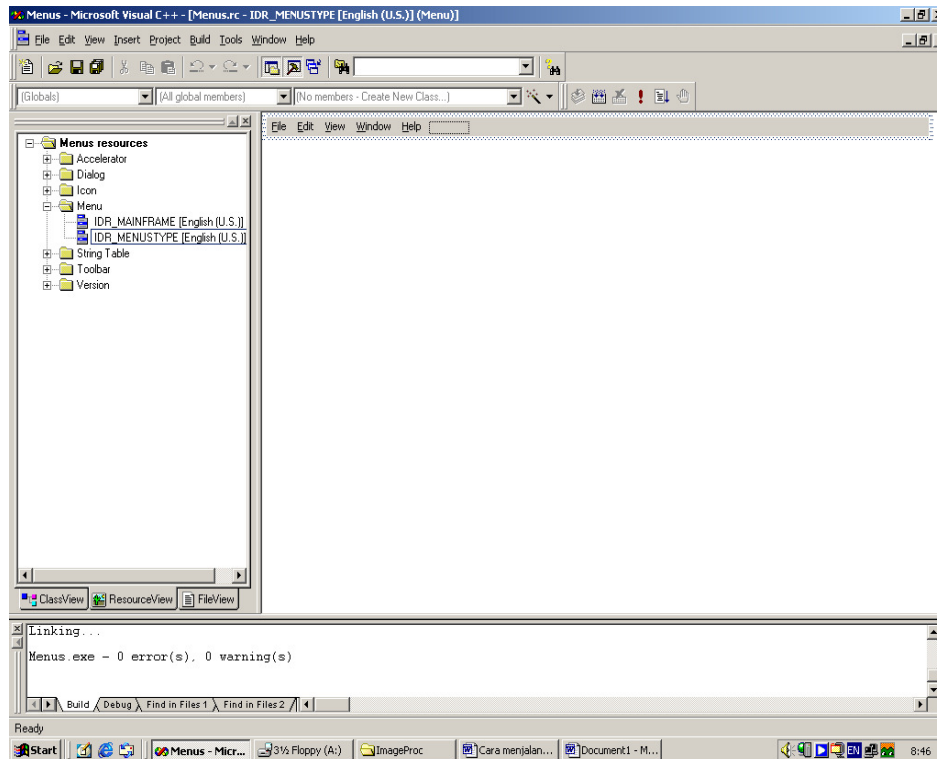
Gambar 1.3 Memilih isi program

- Tekan tombol Finish kemudian tombol OK
4. Cara menjalankan program
- Pilih menu : Build->Execute (!)
 - Tekan tombol Yes

1.4.2. Cara membuat menu dengan MFC

1. Cara membuat menu

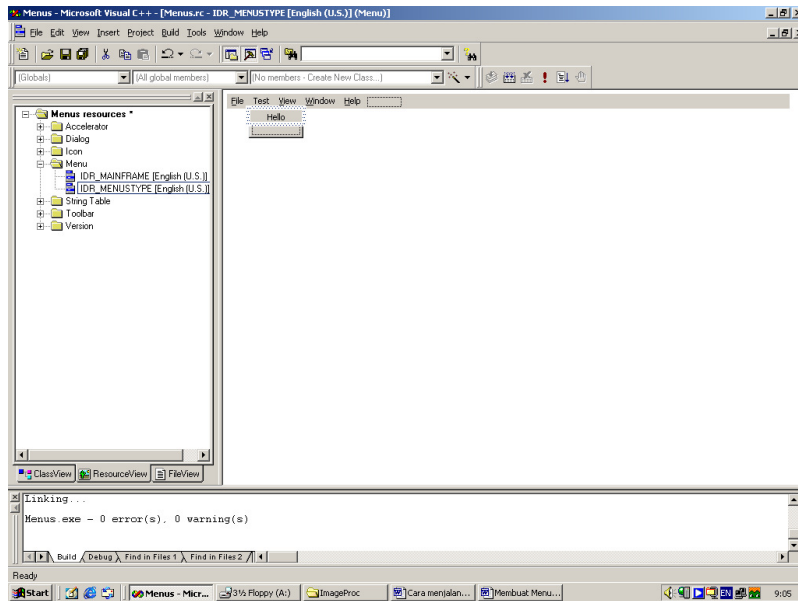
- Buat aplikasi AppWizard seperti pada praktikum 1 dan beri nama project dengan Menus
- Pilih ResourceView pada workspace
- Pilih folder paling atas dengan cara klik pada tanda +
- Pilih folder menu dengan cara klik pada tanda +
- Klik 2 kali IDR_MENUSTYPES[English[U.S.]] seperti gambar 1.4



Gambar 1.4 Membuat Menu

2. Menghapus dan menambah menu

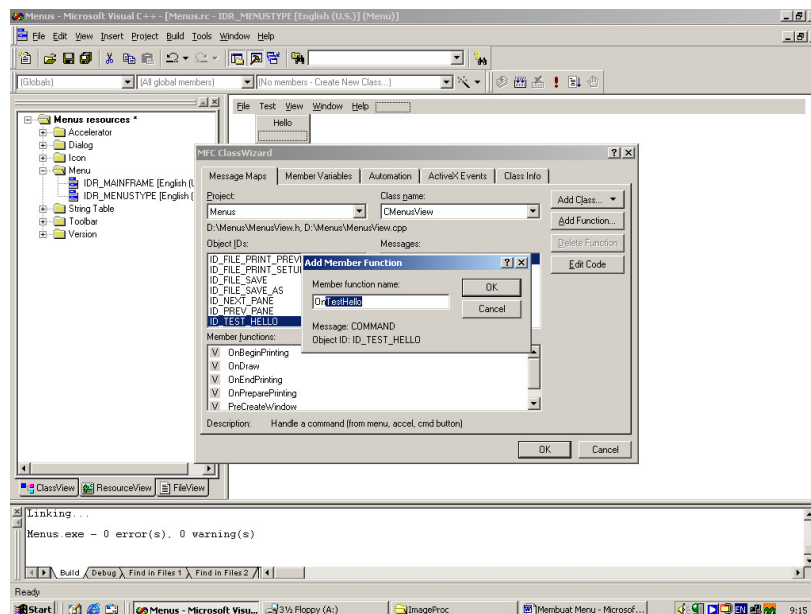
- Untuk menghapus menu : Pilih menu Edit kemudian tekan keyboard Delete kemudian tekan tombol OK
- Untuk Menambah menu: Tekan tombol keyboard Insert dan ketik misalnya Test kemudian dibawahnya ketik Hello seperti gambar 1.5



Gambar 1.5 Menghapus dan menambah menu

3. Mengaktifkan menu

- Pilih menu Hello dan klik kanan
- Pilih menu ClassWizard dan klik kiri
- Pilih message dengan COMMOND dan klik kiri dua kali seperti gambar 1.6 kemudian tekan tombol OK
- Untuk mengedit isi program tekan tombol Edit Code



Gambar 1.6 Mengaktifkan menu

4. Menampilkan fungsi

- Tambahkan program untuk menampilkan pesan seperti dibawah ini

```
////////////////////////////////////  
// CMenuView message handlers  
void CMenuView::OnTestHello()  
{  
    // TODO: Add your command handler code here  
    // Menampilkan pesan untuk pemakai  
    MessageBox("Hello");  
}
```

- Jalankan program dengan memilih menu Build->Execute (!)

1.5. Latihan:

1. Buat program untuk menampilkan Tulisan "Selamat Belajar Pengolahan Citra" dengan menggunakan MessageBox bila submenu yang dipilih.
2. Buatlah program untuk menampilkan Tulisan "Ini adalah Teks" pada fungsi OnDraw(CDC* pDC) dengan menggunakan pDC->TextOut(10,10,"Ini adalah Teks");
3. Buatlah program untuk menggambar sumbu x dan sumbu y pada fungsi OnDraw(CDC* pDC) dengan menggunakan pDC->MoveTo(10,10) dan pDC->LineTo(10,100)
4. Buatlah program untuk menampilkan persamaan linear $y=x$ pada fungsi OnDraw(CDC* pDC) dimana nilai x dari 0 sampai 90
5. Buatlah program untuk menampilkan persamaan kuadrat $y=x^2$ dimana nilai x dari 0 sampai 90 bila submenu yang dibuat dipilih.

1.6. Laporan Resmi:

Buatlah laporan resmi dari latihan-latihan diatas dengan cara membuat analisa dan kesimpulan.

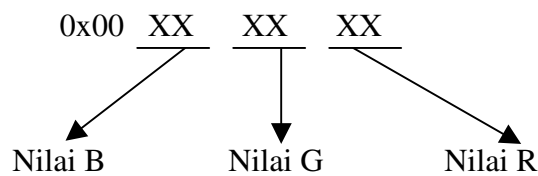
Dasar Pengolahan Citra (1)

2.1. Tujuan:

1. Mahasiswa dapat membuat program untuk menampilkan gambar
2. Mahasiswa dapat membuat program untuk memproses gambar dengan mengambil warna RGB
3. Mahasiswa dapat membuat program untuk memproses gambar dengan meletakkan warna RGB pada lokasi x dan y

2.2. Dasar Teori:

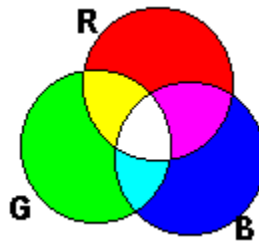
Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna di atas seperti gambar 2.1, variabel 0x00 menyatakan angka dibelakangnya adalah hexadecimal.



Gambar 2.1 Nilai warna RGB dalam hexadesimal

Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 255), atau mempunyai nilai derajat keabuan $256 = 2^8$. Dengan demikian range warna yang digunakan adalah $(2^8)(2^8)(2^8) = 2^{24}$ (atau yang dikenal dengan istilah True Colour pada Windows). Nilai warna yang digunakan di atas merupakan gandingan warna cahaya merah, hijau dan biru seperti yang terlihat pada

gambar 2.2. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.



Gambar 2.2 Komposisi warna RGB

Dari definisi diatas untuk menyajikan warna tertentu dapat dengan mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB, table 1. berikut memperlihatkan contoh-contoh warna yang bisa digunakan

Tabel 1. Contoh-contoh warna dalam hexadesimal

Nilai	Warna	Nilai	Warna
0x00000000	Hitam	0x0000AAFF	Orange
0x000000FF	Merah	0x00888888	Abu-abu
0x0000FF00	Hijau	0x00FF00AA	Ungu
0x00FF0000	Biru	0x00AAFF00	Hijau Muda
0x0000FFFF	Kuning	0x00AA00FF	Merah Muda
0x00FF00FF	Magenta	0x00AAFFFF	Kuning Muda
0x00FFFF00	Cyan	0x000088AA	Coklat
0x00FFFFFF	Putih	0x00AA0088	Ungu

Untuk mengetahui kombinasi warna, perlu dibuat suatu program yang dapat menampilkan warna sesuai dengan nilai yang dimasukkan sehingga dapat dicoba berbagai macam kombinasi warna RGB seperti gambar 2.2.

2.3. Tugas Pendahuluan:

1. Tuliskan tujuan praktikum
2. Jelaskan nilai warna RGB dalam hexadesimal
3. Sebutkan tiga komposisi warna dasar
4. Buatlah ringkasan mengenai class CFileDialog, CBitmap, CDC dan metoda setPixel dan getPixel di MSDN

2.4. Percobaan:

2.4.1. Menampilkan File Gambar

1. Cara membuka file

- Buat aplikasi AppWizard seperti pada praktikum 1 dan beri nama project dengan OpenFileDialog
- Buat Menu seperti pada praktikum 2 dengan tambahan Test dan submenunya OpenFileDialog
- Untuk mengedit isi program tekan tombol Edit Code atau buka file OpenFileDialog.cpp
- Tambahkan program untuk membuka file seperti dibawah ini

```
////////////////////////////////////  
// COpenFileView message handlers  
// Menampilkan file yang akan dibuka  
void COpenFileView::OnTestOpenfile()  
{  
    // TODO: Add your command handler code here  
    static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)*.bmp|";  
  
    CFileDialog m_IdFile(TRUE, "*.bmp", name,  
        OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);  
    if(m_IdFile.DoModal()!=IDOK)  
    {  
        name=m_IdFile.GetPathName();  
        LoadGambar();  
    }  
}  
  
// Menampilkan gambar hasil dari open file  
void COpenFileView::LoadGambar(void)  
{  
    CDC* pDC = GetDC();  
    CDC dcMem;  
    HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(), name,  
        IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE|LR_CREATEDIBSECTION);  
    if(hBitmap)  
    {  
        if(m_bmpBitmap.DeleteObject())  
            m_bmpBitmap.Detach();  
        m_bmpBitmap.Attach(hBitmap);  
    }  
    dcMem.CreateCompatibleDC(pDC);  
    dcMem.SelectObject(&m_bmpBitmap);  
    pDC->BitBlt(0,0,250,210,&dcMem,0,0,SRCCOPY);  
}
```

2. Menambah header file

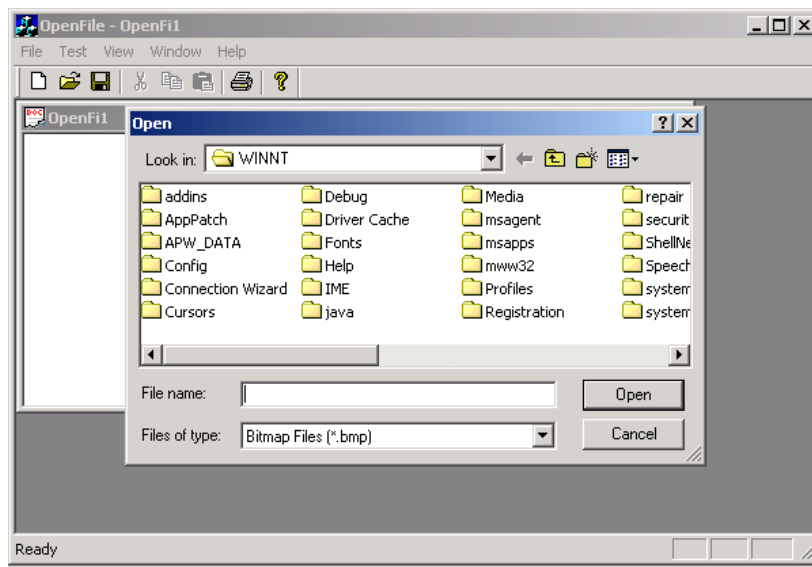
- Buka file OpenFileDialog.h
- Tambahkan program seperti dibawah ini

```
// Attributes
public:
    COpenFileDoc* GetDocument();
    CString name;
    CBitmap m_bmpBitmap;

// Operations
public:
    void LoadGambar(void);
```

3. Cara menjalankan program

- Pilih menu : Build->Execute (!)
- Pilih menu : Test->OpenFile ->pilih salah satu gambar misalnya gambar.bmp
- Hasilnya seperti gambar 2.3



Gambar 2.3 Membuka file gambar

2.4.2. Cara Memproses Gambar

1. Cara memproses gambar

- Buat aplikasi AppWizard seperti pada praktikum 1 dan beri nama project dengan Proses
- Buat Menu seperti pada praktikum 2 dengan tambahan Test sedangkan submenunya OpenFile dan Proses
- Untuk mengedit isi program tekan tombol Edit Code atau buka file ProsesView.cpp
- Tambahkan program untuk memproses gambar seperti dibawah ini

```
////////////////////////////////////  
// CProsesView message handlers  
  
void CProsesView::OnTestOpenfile()  
{  
    // TODO: Add your command handler code here  
    static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)|*.bmp|";  
  
    CFileDialog m_IdFile(TRUE, "*.bmp", name,  
        OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);  
    if(m_IdFile.DoModal()==IDOK)  
    {  
        name=m_IdFile.GetPathName();  
        LoadGambar();  
    }  
}  
  
// Menampilkan gambar hasil dari open file  
void CProsesView::LoadGambar(void)  
{  
    CDC* pDC = GetDC();  
    CDC dcMem;  
    HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(), name,  
        IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE|LR_CREATEDIBSECTION);  
    if(hBitmap)  
    {  
        if(m_bmpBitmap.DeleteObject())  
            m_bmpBitmap.Detach();  
        m_bmpBitmap.Attach(hBitmap);  
    }  
    dcMem.CreateCompatibleDC(pDC);  
    dcMem.SelectObject(&m_bmpBitmap);  
    pDC->BitBlt(0,0,250,210,&dcMem,0,0,SRCCOPY);  
}  
void CProsesView::OnTestProses()  
{  
    // TODO: Add your command handler code here  
    CDC* pDC = GetDC();
```

```

CDC dcMem;
int i,j;
long int warna;
char str[5];
dcMem.CreateCompatibleDC(pDC);
dcMem.SelectObject(&m_bmpBitmap);
for(i=0;i<210;i++)
    for(j=0;j<250;j++)
    {
        // memberi warna pada titik
        dcMem.SetPixel(j,i,0x000000ff);
    }
pDC->BitBlt(0,0,250,210,&dcMem,0,0,SRCCOPY);
// membaca warna pada titik
warna=dcMem.GetPixel(5,5);
sprintf(str,"%ld",warna);
pDC->TextOut(10,10,str);
}

```

2. Menambah header file

- Buka file ProsesView.h
- Tambahkan program seperti dibawah ini

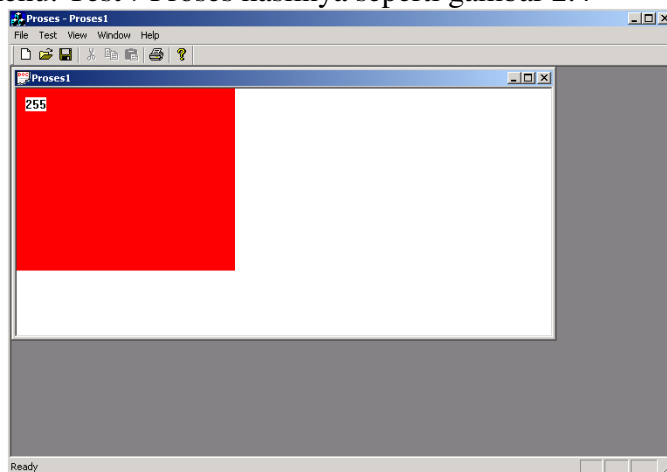
```

// Attributes
public:
    CProsesDoc* GetDocument();
    CString name;
    CBitmap m_bmpBitmap;
// Operations
public:
    void LoadGambar(void);

```

3. Cara menjalankan program

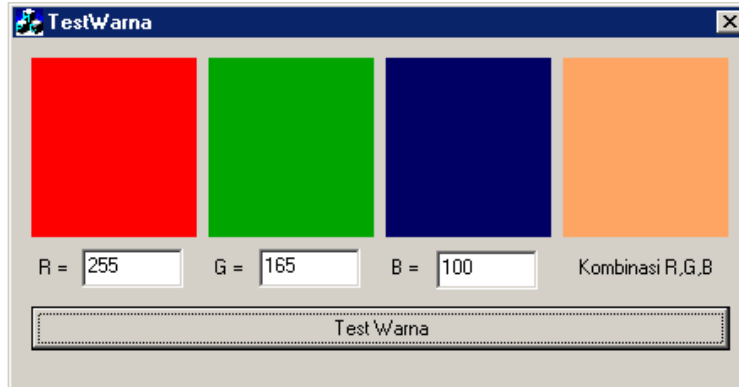
- Pilih menu : Build->Execute (!)
- Pilih menu : Test->OpenFile -> pilih salah satu gambar misalnya gambar.bmp
- Pilih menu: Test->Proses hasilnya seperti gambar 2.4



Gambar 2.4 Memproses gambar

2.5. Latihan:

1. Buatlah program untuk mengubah warna 3 buah picture-box dengan tiga macam nilai RGB yang masing-masing bernilai 0-255, sedangkan picture box yang keempat bernilai RGB yang merupakan kombinasi nilai-nilai R, G dan B seperti gambar berikut ini:



Gambar 2.5. Contoh program test warna

Cobalah mengisi dengan kombinasi berikut ini, apakah hasil dari kombinasi warna RGB berikut ini:

- (a) $R=0, G=255, B=128$
 - (b) $R=128, G=128, B=50$
 - (c) $R=100, G=100, B=255$
2. Buatlah program untuk melakukan perputaran citra yang setiap baris pada picture-box 1 menjadi kolom pada picture-box 2 dan setiap kolom pada picture box 1 menjadi baris pada picture box 2. Tampilan formnya sebagai berikut.



Gambar 2.6. Pembalikan gambar

2.6. Laporan Resmi:

Buatlah laporan resmi dari latihan-latihan diatas dengan cara membuat analisa dan kesimpulan.