

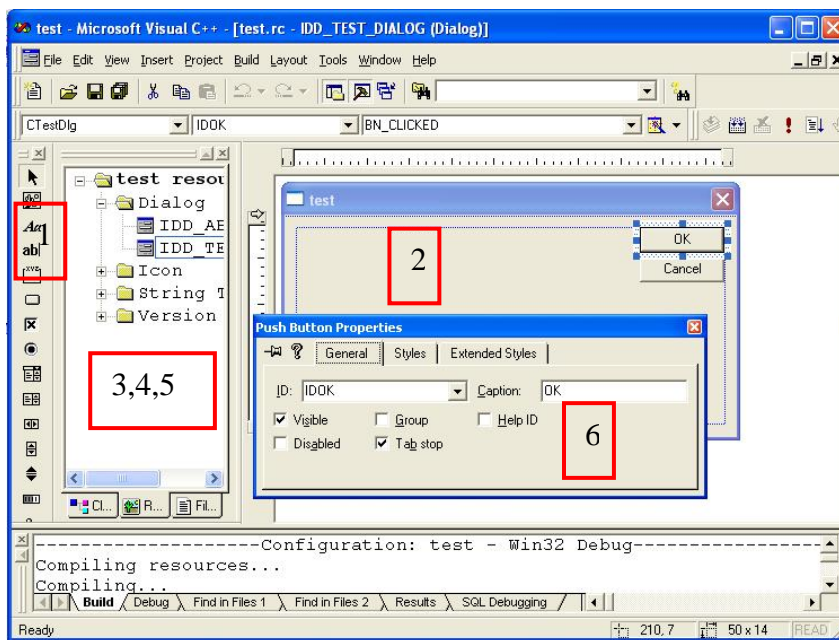
Pendahuluan

1. Tujuan:

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

2 Dasar Teori:

IDE (Integrated Development Environment) Visual C++ 6 dapat dilihat pada gambar 1 . Dengan macam-macam komponen di dalamnya dapat dilihat pada tabel 1.



Gambar 1.1. IDE Visual C++ 6

Macam-macam komponen yang terdapat pada IDE visual C++ 6 sbb :

Tabel 1.1.

Macam-Macam Komponen IDE VC++ 6

No	IDE Komponen	Deskripsi
1	Toolbox	Bermacam-macam control yang dapat digunakan di project Visual C++ (pada prak Citra yang biasa dipakai : picture, textbox, button,slider dsb).
2	Dialog Editor (MFC)	Pada area ini tempat membuat atau mengedit tampilan dialog box bukan form window.
3	Solution Explorer	Menampilkan organisasi project yang dibuat
4	Class View	Menampilkan simbol code project seperti : namespaces, classes, methods, dan functions
5	Resource View	Menampilkan resource file dari project
6	Properties Window	Window yang dapat dipakai untuk mengedit atau merubah sifat control pada waktu disain

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 1.2 dibawah ini:

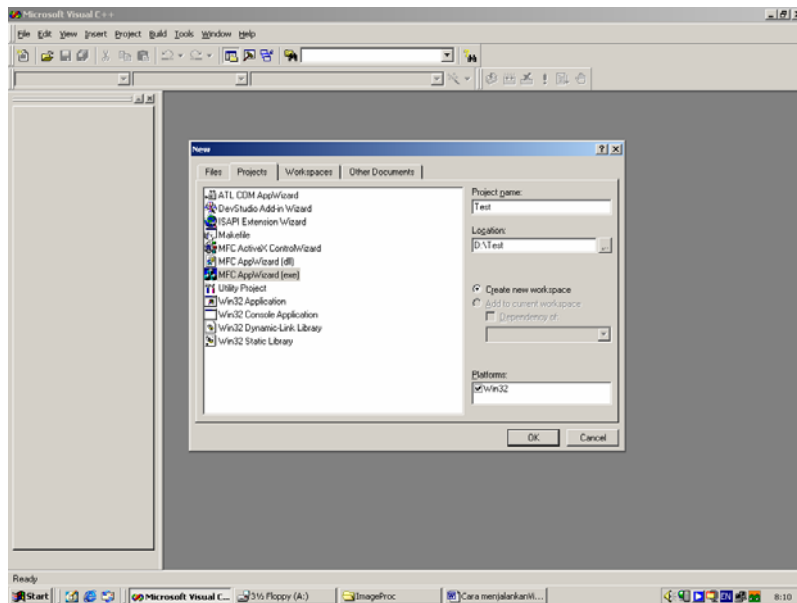


Gambar 1.2 Blok Diagram Pengolahan Citra

3. Percobaan:

3.1 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: **dialog** (lihat gambar 1.3)
 - Tekan tombol OK



Gambar 1.3 Memberi nama program

3. Memilih isi program
 - Step 1 : What tipe of application would you like to create (aplikasi apa yang ingin dibuat) pilih *dialog based* untuk membuat aplikasi memakai dialog. Perhatikan gambar 1.4. a dialog step 1. Tekan tombol Next.
 - Step 2 :
 1. What features would you like to include?

(fitur apa saja yang akan dimasukkan) pilih *About box* dan *3D controls* untuk pilihan standar.

2. What features would you like to include?

(mendukung aplikasi apa ?) pilih *ActiveX controls* untuk pilihan standar.

3. Would you like to include WOSA support ? *Window Sockets* tidak dipilih untuk pilihan standard.

4. Please enter a title for your dialog ? Title sudah berisi text sama dengan nama project untuk pilihan standard

Perhatikan gambar 1.4.b. dialog step 2. Tekan tombol Next.

• Step 3 :

1. What style of project would you like ?

Pilihan standard dan yang aktif hanya *MFC Standard*.

2. Would you like to generate source file comments ? pilihan standard Yes, Please

3. How would you like to use the MFC Library ? pilihan standard As a shared DLL

Perhatikan gambar 1.4.c. dialog step 3. Tekan tombol Next

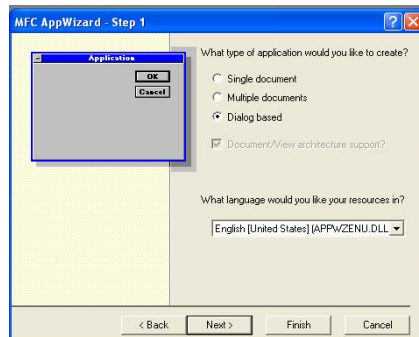
• Step 4 : AppWizard creates the following classes for you

(AppWizard akan membuat class seperti dibawah ini)

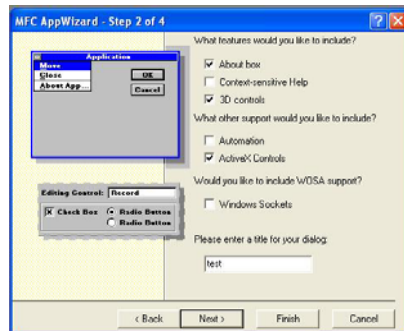
CtestApp

CtestDlg

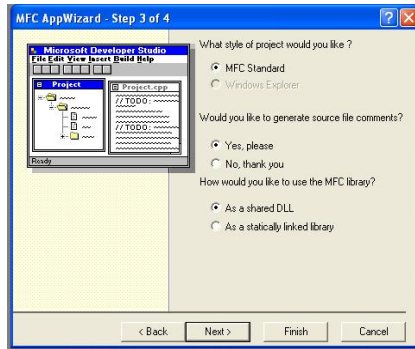
Perhatikan gambar 1.4.d. dialog step 4.



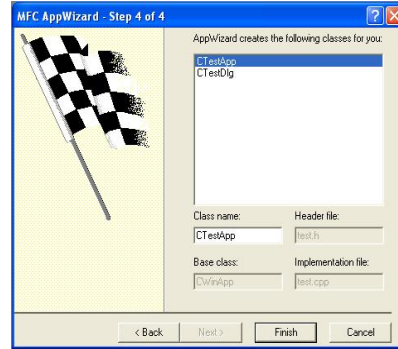
(a)



(b)



(c)

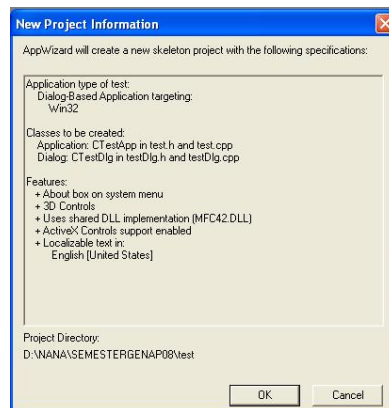


(d)

Gambar 1.4. Memilih Isi Program

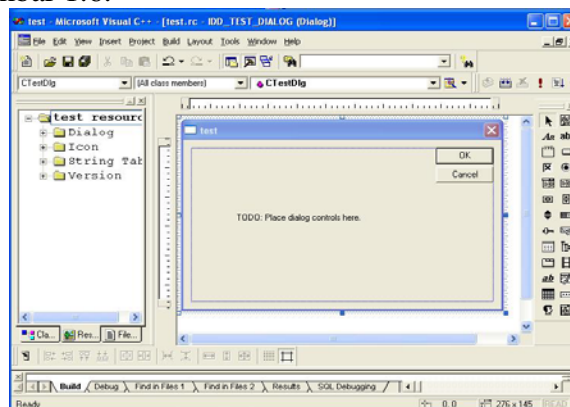
(a) Dialog Step 1 (b) Dialog Step 2 (c) Dialog Step 3 (d) Dialog Step 4

- Tekan tombol Finish akan muncul New Project Information. Perhatikan gambar 1.5.



Gambar 1.5. New Project Information

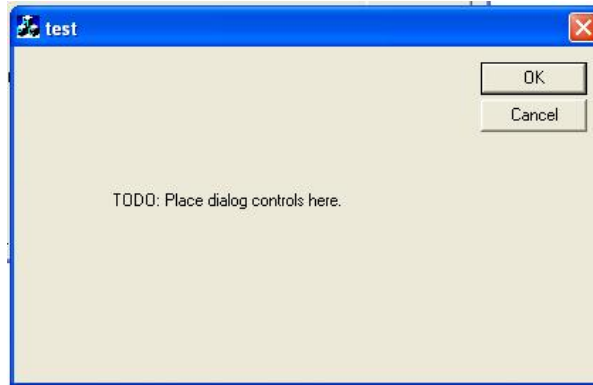
- Kemudian tekan tombol OK. Tampilan awal dialog akan dihasilkan perhatikan gambar 1.6.



Gambar 1.6. Awal Dialog

4. Cara menjalankan program

- Pilih menu : Build->Execute (!), perhatikan gambar 1.7. Dialog Hasil Execute
- Tekan tombol Yes, untuk menutup aplikasi.



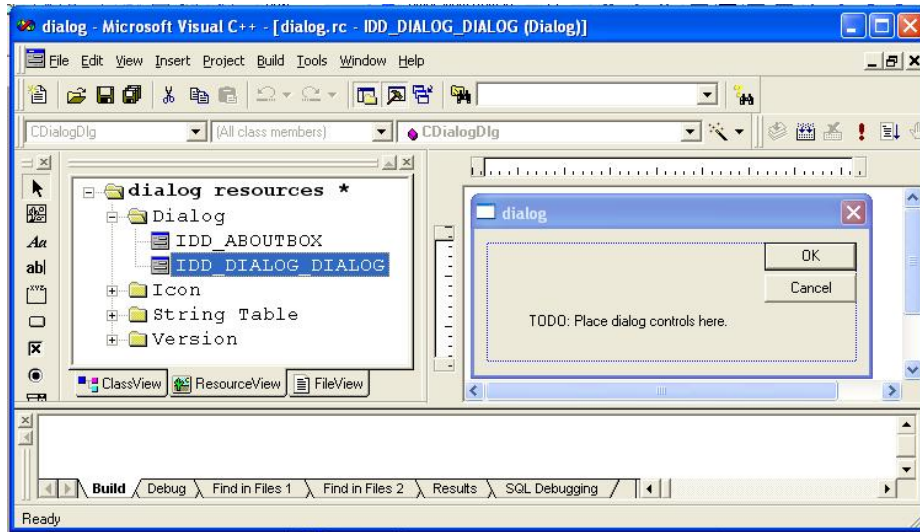
Gambar 1.7. Dialog Hasil Execute

1.1.1. Cara Mendisain Dialog

Catatan : Jika Dialog Editor tidak tampak buka pada Resource View (Ctrl-Shift-E) double click pada *IDD_TEST_DIALOG*.

1. Cara membuat dialog

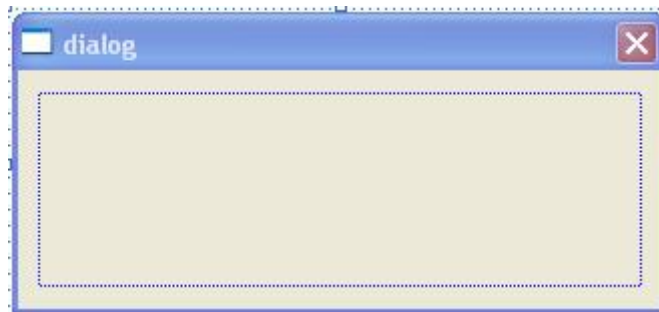
- Buat aplikasi AppWizard seperti pada praktikum 1 dan beri nama project dengan Dialog
- Pilih ResourceView pada workspace
- Pilih folder paling atas dengan cara klik pada tanda +
- Pilih folder dialog dengan cara klik pada tanda +
- Klik 2 kali *IDD_DIALOG_DIALOG* seperti gambar 1.4



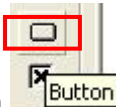
Gambar 1.8 Membuat Dialog

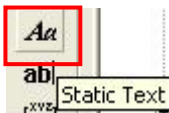
2. Menghapus dan menambah control pada dialog


- Untuk menghapus control : aktifkan control dengan cara klik pada area control lanjutkan dengan menekan tombol delete. Cobalah untuk menghapus control : text TODO, button OK dan Cancel. Hasil akhir pada gambar 1.9.



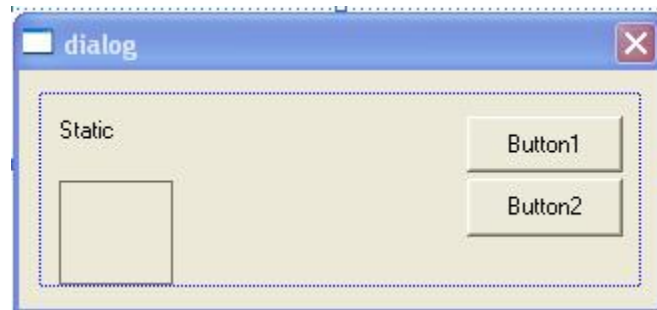
Gambar 1.9. Dialog dengan Semua Control Dihapus

- Untuk menambah control: (button) pilih icon  control pada toolbox drag drop ke editor dialog.

(static text) pilih icon  control pada toolbox drag drop ke editor dialog.

(picture box) pilih icon  control pada toolbox drag drop ke editor dialog.

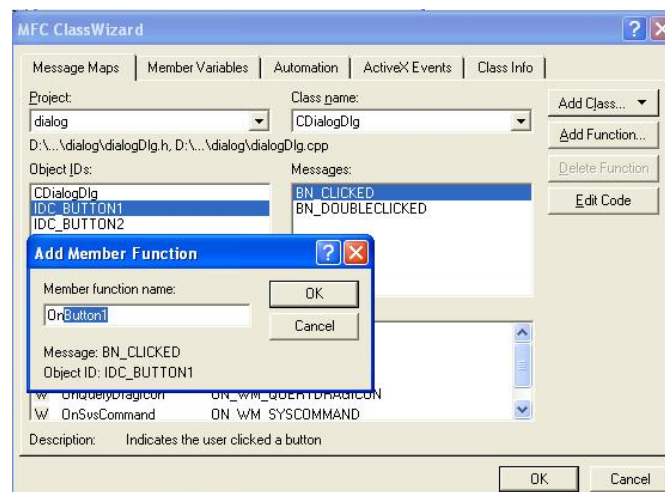
- Tambahkan control (dua button, 1 static text, 1 picture) pada dialog editor seperti hasil pada gambar 1.10.



Gambar 1.10 Disain Dialog

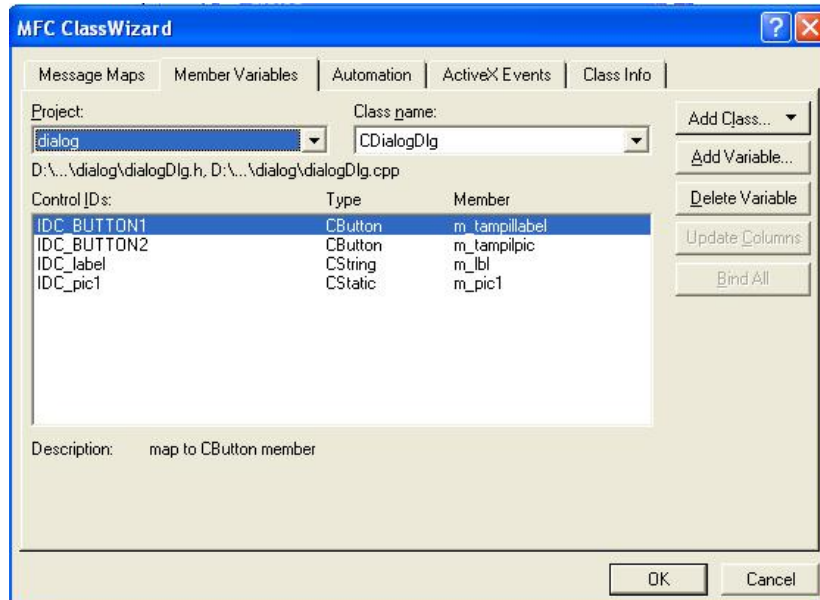
3. Mengaktifkan control

- Pilih control button1, double klik, pada dialog Add Member Function klik OK.
- Atau klik kanan, pilih klik ClassWizard, pada MFC Class Wizard Dialog Messages aktifkan BN_CLICKED, double klik, pada dialog Add Member Function klik OK. Gambar 1.11.



Gambar 1.11. Mengaktifkan Control Button1

4. Rubah terlebih dahulu ID dari static text = ID_label dan ID dari picture = ID_pic1, dengan cara aktifkan control dan klik kanan pilih properti.
5. Pada dialog MFC classwizard, Member Variabels edit type dan member dari control sebagai berikut gambar 1.12.



Gambar 1.12. MFC ClassWizard pada Member Variabel

6. Menulis fungsi pada button1 klik
 - Tambahkan program untuk menampilkan pesan pada static text seperti dibawah ini

```
void CDialogDlg::OnButton1()
{
    // TODO: Add your control notification handler code here
    SetDlgItemText(IDC_label, "Belajar Prak Citra dgn VC++ dan MFC");
}
```

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

7. Tambahkan deklarasi kelas CBitmap pada File View – Header Files – dialogDlg.h

```
// Construction
public:
    CDialogDlg(CWnd* pParent = NULL);    // standard constructor
    CBitmap m_bmpBitmap;
//
```

8. Menulis fungsi pada button2 klik

```
void CDialogDlg::OnButton2()  
{  
    // TODO: Add your control notification handler code here  
    CDC* pDC = m_pic1.GetDC();//  
    CDC dcMem1;  
    CRect rect;//kotak di picture  
    BITMAP bm;//  
    HBITMAP  
    hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),  
    "pens.bmp",IMAGE_BITMAP, 0, 0,  
    LR_LOADFROMFILE|LR_CREATEDIBSECTION);  
    if(hBitmap)  
    {  
        if(m_bmpBitmap.DeleteObject())  
            m_bmpBitmap.Detach();  
        m_bmpBitmap.Attach(hBitmap);  
    }  
    m_pic1.GetClientRect(rect);//  
    m_bmpBitmap.GetBitmap(&bm);//  
    dcMem1.CreateCompatibleDC(pDC);  
    dcMem1.SelectObject(&m_bmpBitmap);  
    pDC->StretchBlt(0,0,rect.Width(),rect.Height(),&dcMem1,  
    0,0,bm.bmWidth,bm.bmHeight,SRCCOPY);//  
}
```

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

Hasil akhir program dijalankan pada gambar 1.13.



Gambar 1.13. Hasil Execute Prak 1.

1.2. 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 100
5. Buatlah program untuk menampilkan persamaan kuadrat $y=x^2$ dimana nilai x dari 0 sampai 100 bila submenu yang dibuat dipilih.

1.6. Laporan Resmi:

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