

PELATIHAN IMAGE PROSESING

MATERI

IMAGE PROSESING DAN APLIKASINYA



Oleh : Bima Sena Bayu Dewantara, SST

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2006

MODUL 1

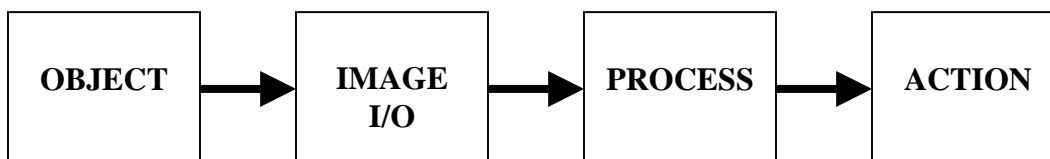
Image I/O Dengan Menggunakan MS Visual C++ 6

1. Tujuan

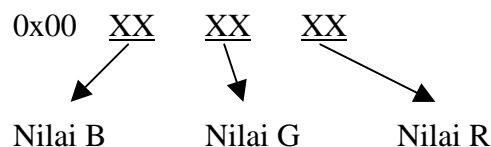
- Mahasiswa dapat membuat program pengolahan citra untuk kepentingan robot vision dengan menggunakan MS Visual C++ menggunakan MFC (Microsoft Foundation Class).
- Mahasiswa dapat membuat menu-menu yang berhubungan dengan pengolahan citra menggunakan Visual C++ dengan MFC.
- Mahasiswa dapat menampilkan dan memproses informasi RGB dari gambar pada layar monitor sebagai dasar robot vision.

2. Teori Penunjang

Image I/O merupakan proses pemasukan dan pengeluaran data yang berupa gambar atau citra kedalam sebuah program aplikasi dalam hal ini MS Visual C++ 6.0. Sehingga didapatkan informasi tertentu yang untuk selanjutnya dapat diolah, dianalisa dan diberikan sebuah keputusan yang tepat untuk pengendalian, dalam hal ini sebagai pandangan (vision) dari sebuah perangkat keras atau robot.



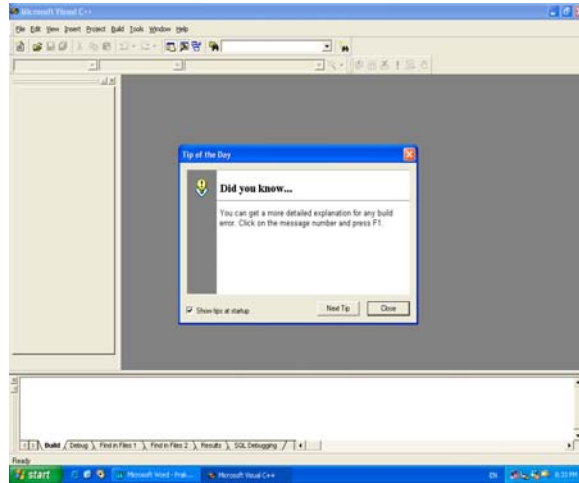
Dasar dari robot vision adalah pengolahan citra dimana informasi yang dapat diambil dari sebuah perangkat I/O adalah berupa informasi warna RGB (Red, Green dan Blue) pada posisi tertentu. Dalam pengolahan citra warna direpresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff.



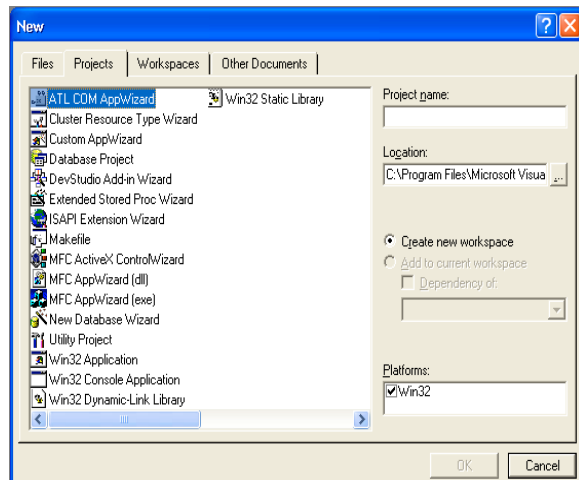
3. PROSEDUR PERCOBAAN

1. Membuat Menu dengan MS Visual C++ menggunakan MFC

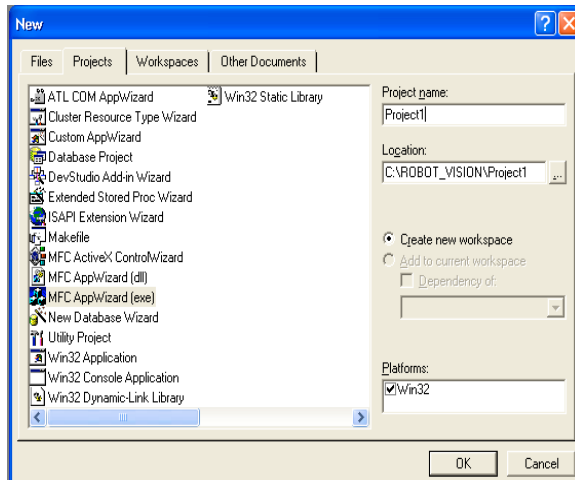
- a) Pada dekstop, tekan *start* → *Programs* → *Microsoft Visual Studio 6.0* → *Microsoft Visual C++ 6.0*. Sehingga akan muncul layar sebagai berikut :



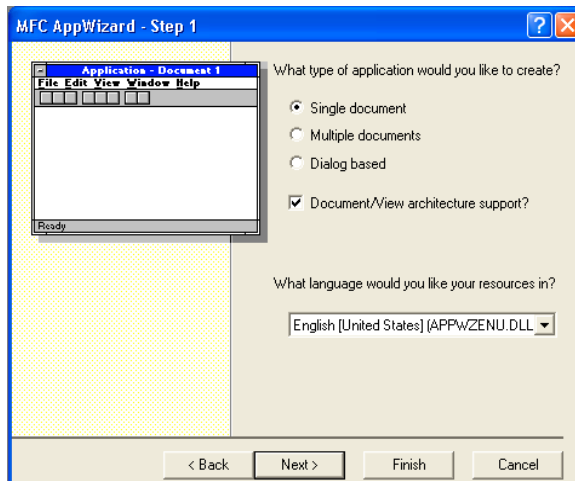
- b) Pada menu program, tekan *File* → *New*, sehingga muncul window baru sebagai berikut :



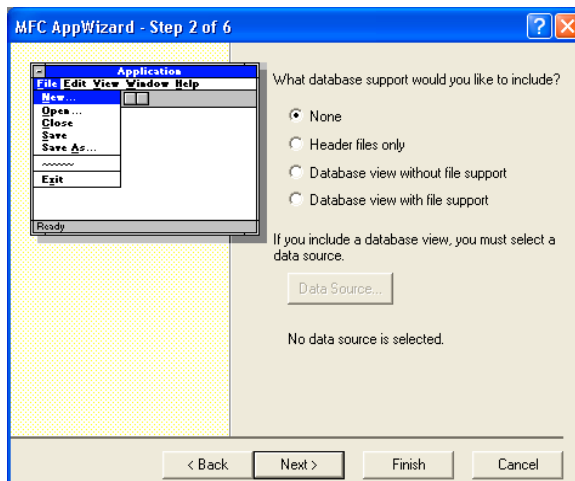
- c) Pada tab menu, pilih *Projects*, kemudian pilih menu *MFC AppWizard(exe)*, kemudian isilah kolom editor Project name untuk memberikan identitas dari nama setiap project yang anda buat, misalkan "*project1*", dsb. Pada kolom editor Location, letakkan atau ketikkan pada *C:\robot_vision* (dengan terlebih dahulu membuat folder tersebut pada C:\



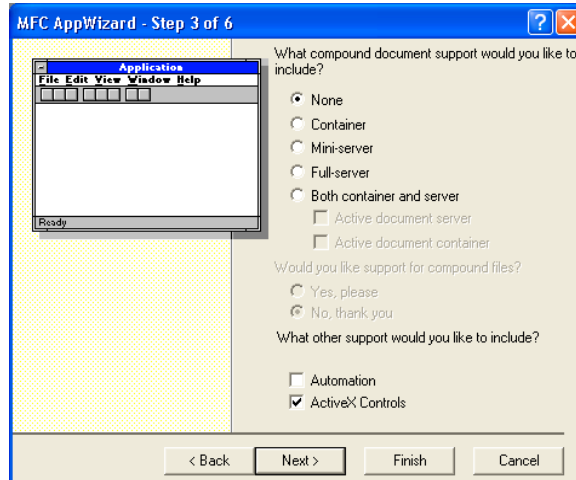
- d) Setelah selesai, tekan **OK**, sehingga akan muncul window baru sebagai berikut dan pilih radio button **“Single Document”**, kemudian tekan **next** :



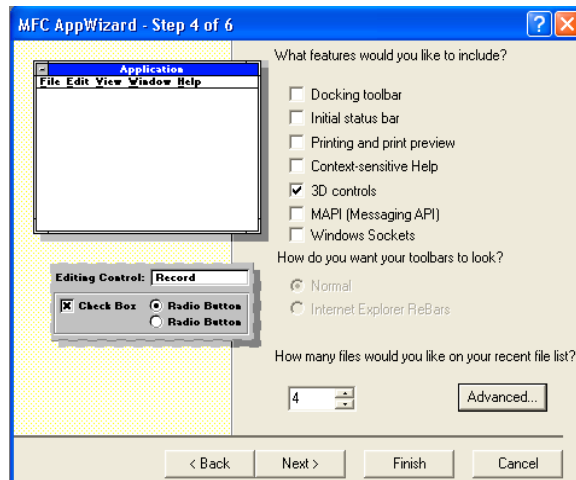
- e) Kemudian muncul window sebagai berikut dan tekan **next** :



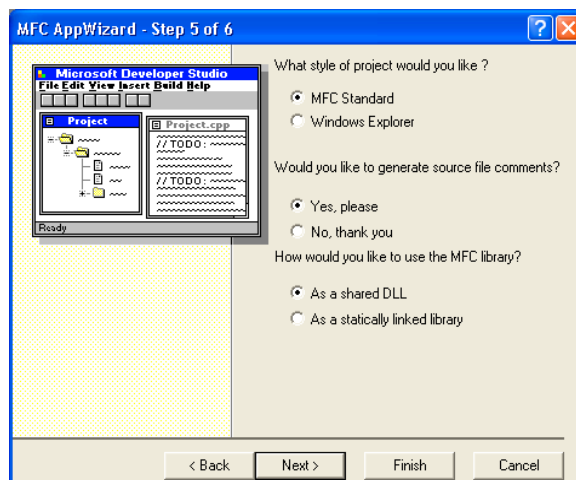
f) Kemudian muncul window sebagai berikut dan tekan *next* :



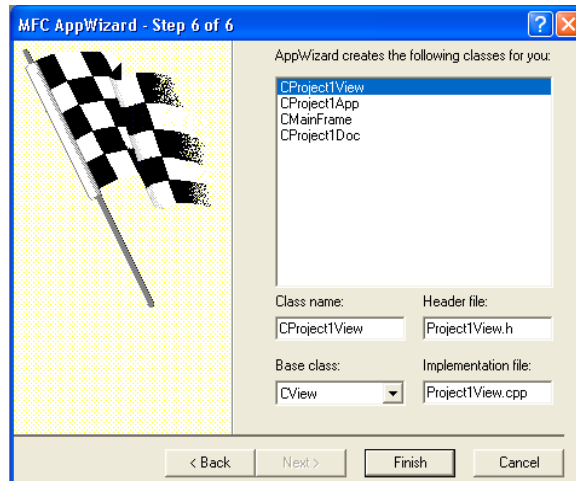
g) Kemudian muncul window sebagai berikut, hilangkan tanda “*cawang*” pada check box “*Docking Toolbar*”, “*Initial Status Bar*”, serta “*Printing and Print Preview*” dan tekan *next* :



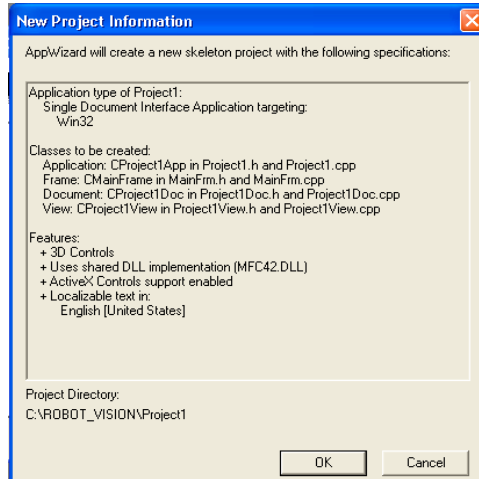
h) Kemudian muncul window sebagai berikut dan tekan *next* :



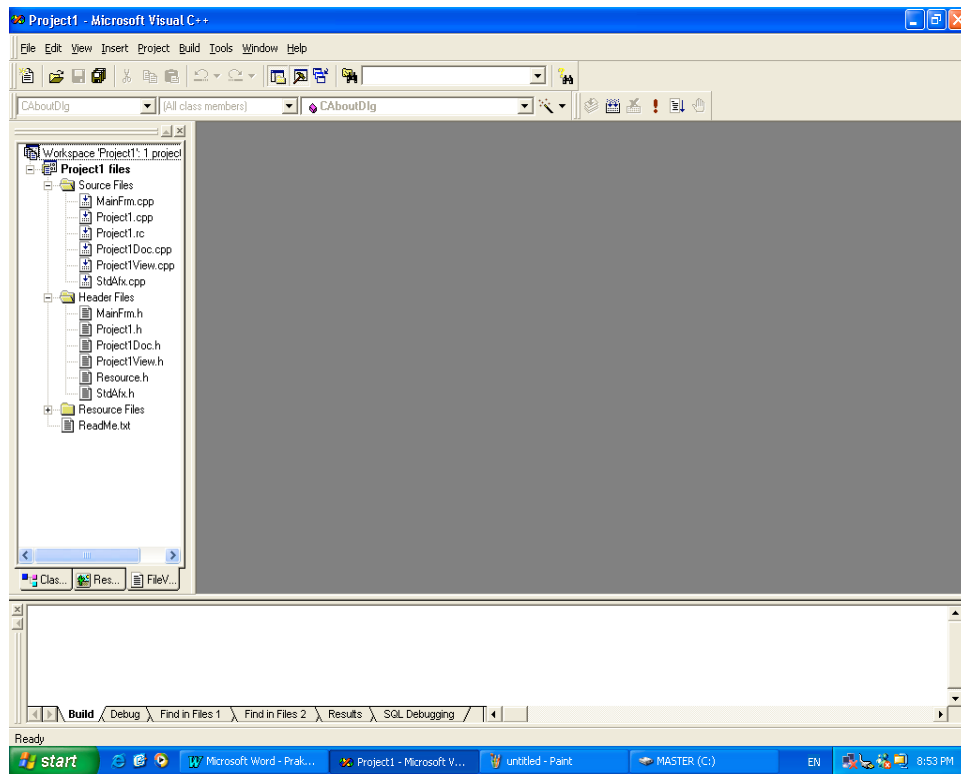
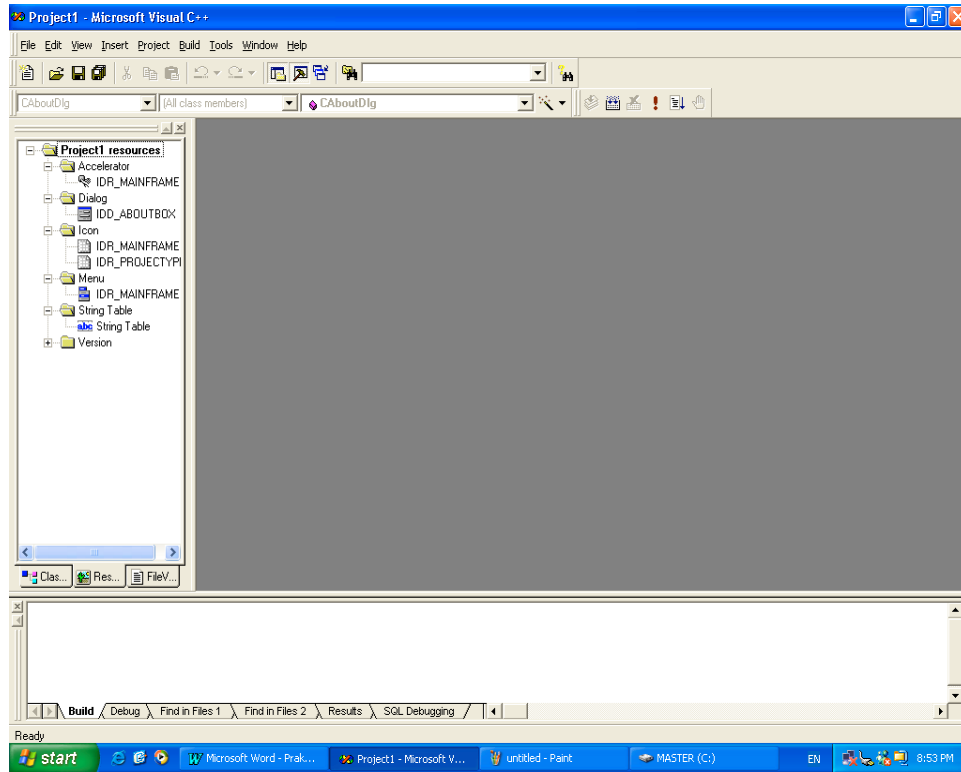
i) Kemudian muncul window sebagai berikut dan tekan *finish* :



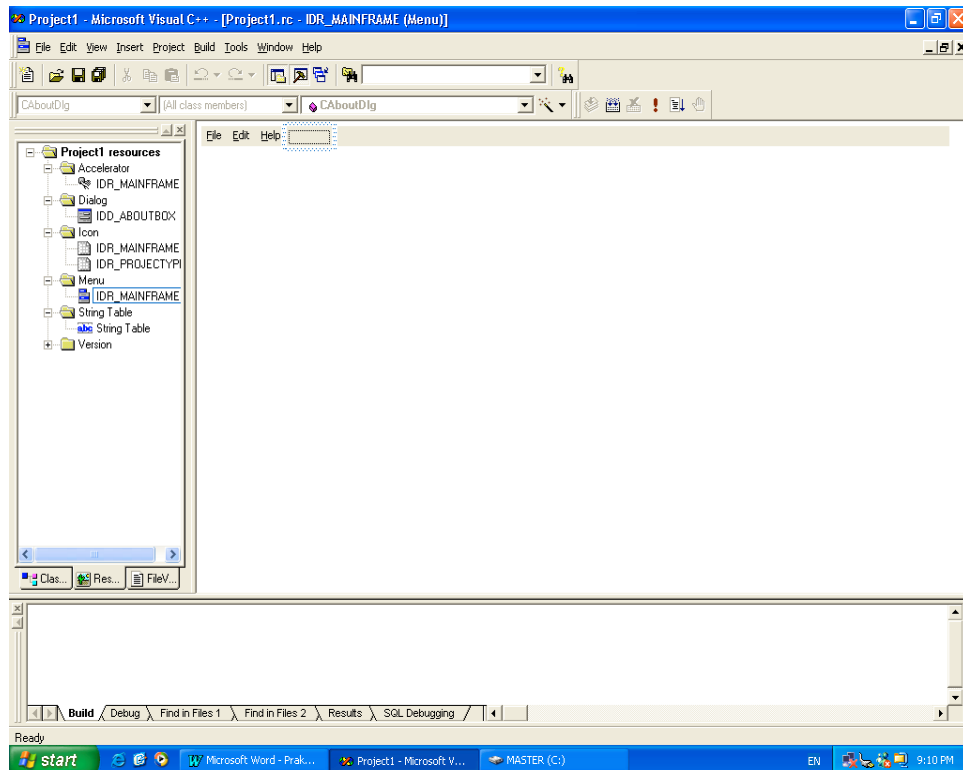
j) Kemudian muncul window sebagai berikut dan tekan *OK* :



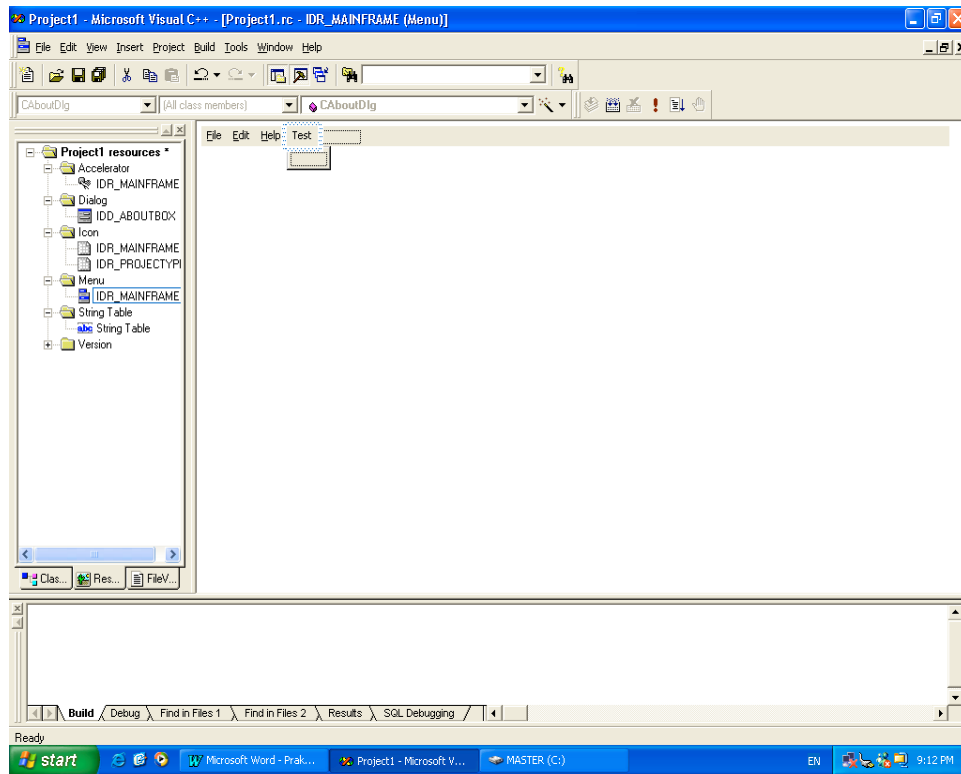
k) Kemudian muncul window sebagai berikut dan MFC siap untuk digunakan sebagai berikut :



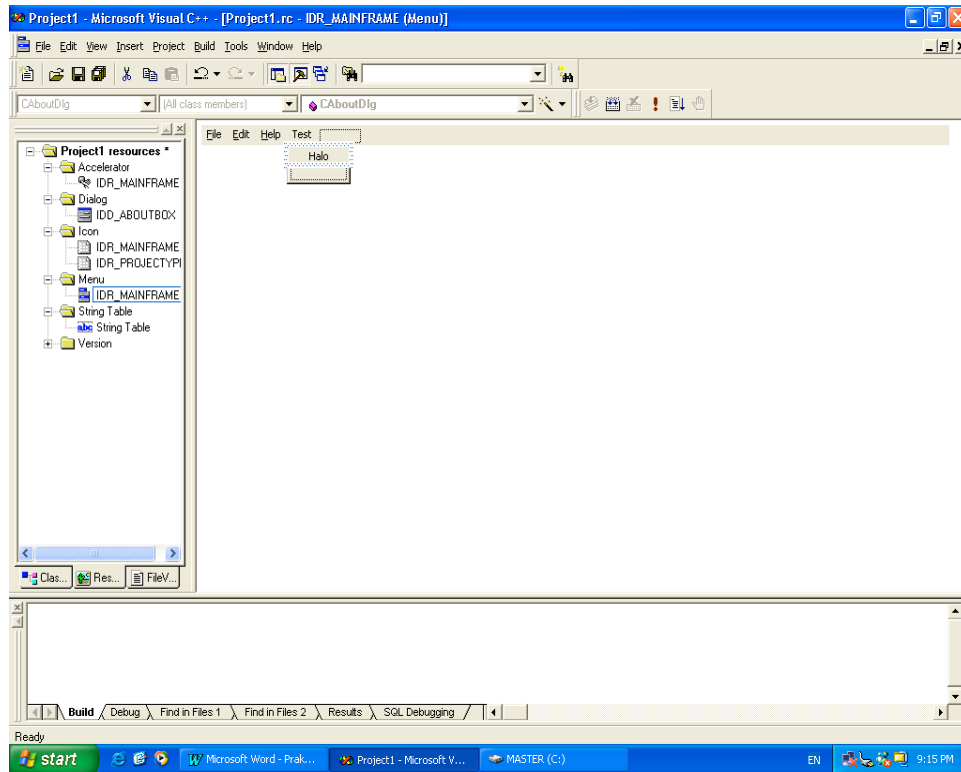
- l) Pada kedua window diatas, terdapat window untuk resources dan window untuk files, dimana window untuk resources terdiri dari Accelerator, Dialog, Icon, Menu, String Table dan Version. Sedangkan pada window untuk files terdiri dari Sources File dan Header File.
- m) Untuk membuat menu baru pada mainframe, buka window resources pada pilihan Menu dan *klik dua kali* pada menu tersebut. Akan muncul window sebagai berikut :



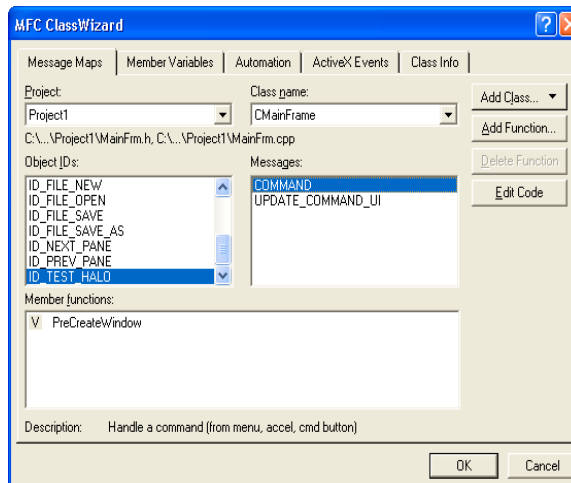
- n) Untuk menambahkan menu baru misalkan *“Test”* maka ketikkan langsung pada tempat yang kosong pada baris untuk toolbar project anda. Dan secara otomatis akan muncul baris baru dibawahnya sebagai sub menu dari menu yang telah anda beri nama tadi, sebagai berikut :



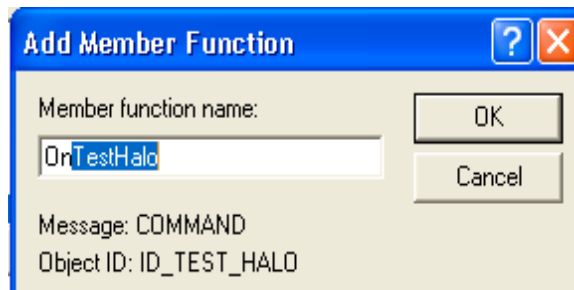
o) Pada baris baru tersebut buatlah sub menu baru dengan nama ***“Halo”*** sebagai berikut :



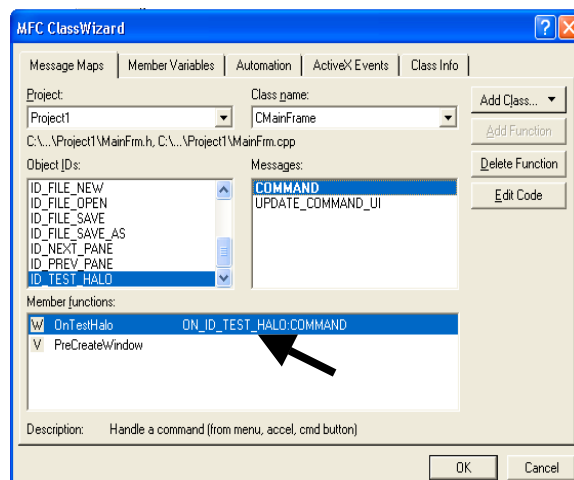
- p) Untuk mengaktifkan menu tersebut, maka klik kanan pada sub menu baru anda tersebut dan pilih menu ClassWizard, sehingga muncul window sebagai berikut :



- q) Pastikan Class name adalah **CmainFrame** dan pada Object ID's, **ID_TEST_HALO** akan di-**highlight** dan pilih pada Messages "**COMMAND**". Klik pada COMMAND akan menghasilkan message box sebagai berikut :



- r) Tekan **OK** dan pastikan bahwa ID telah diaktifkan seperti gambar berikut :



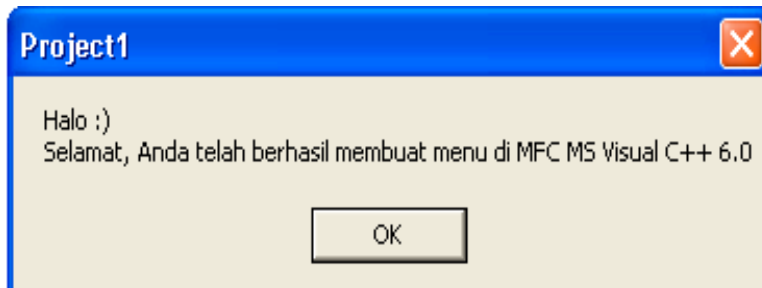
- s) Untuk mengetahui hasil perubahan dari hal yang telah anda lakukan, tekan **Edit Code** dan pastikan bahwa pada file **MainFrame.cpp** telah bertambah dengan sebuah blok fungsi perintah sebagai berikut :

```
IAINFRAME      #endif // _DEBUG
PROJECTYPEI
IAINFRAME      // CMainFrame message handlers
e
Table         void CMainFrame::OnTestHalo()
              {
              // TODO: Add your command handler code here
              }
```

- t) Pada blok tersebut ketikkan pesan sebagai berikut :

```
////////////////////////////////////
// CMainFrame message handlers
void CMainFrame::OnTestHalo()
{
    // TODO: Add your command handler code here
    // Menampilkan pesan untuk pemakai
    MessageBox("Halo :) \nSelamat, Anda telah berhasil membuat menu di
MFC MS Visual C++ 6.0");
}
```

- u) Jalankan program anda dengan menekan tombol execute atau !



2. Menampilkan File Gambar

- a) Ulangi langkah 1 (Membuat Menu dengan MS Visual C++ menggunakan MFC) dengan memberi nama project anda "Project2".
- b) Pada sub menu yang telah anda buat sebelumnya (OpenFile, lihat langkah 1.o), isikan cuplikan program berikut ini pada Class menu ProjectView untuk memanggil file gambar (format .bmp) :

```
void CProject2View::OnTestOpenFile ()
{
    // TODO: Add your command handler code here
    static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)|*.bmp|";

    CFileDialog m_ldFile(TRUE, "*.bmp", name,
    OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);
    if(m_ldFile.DoModal()==IDOK)
    {
        name=m_ldFile.GetPathName();
        LoadGambar ();
    }
}

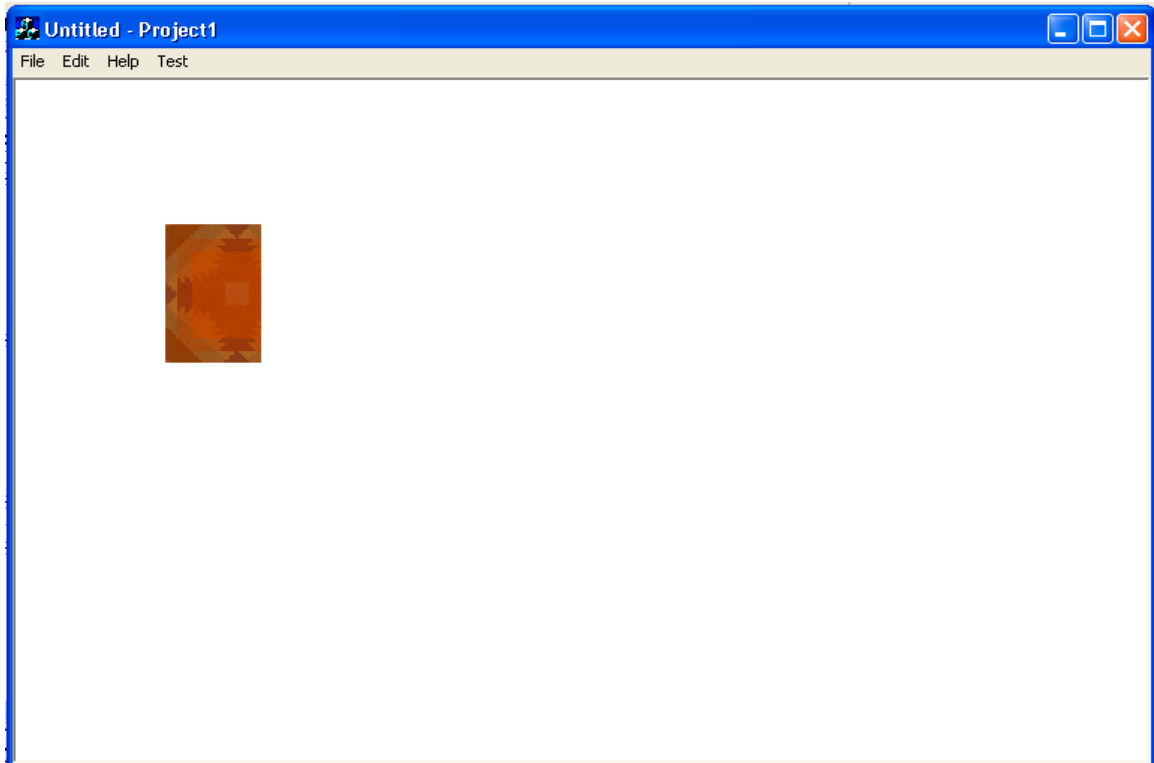
void CProject2View::LoadGambar (void)
{
    CDC* pDC = GetDC();
    CDC dcMem;
    HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
    name, IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE|LR_CREATEDIBSECTION);
    if(hBitmap)
    {
        if(m_bmpBitmap.DeleteObject())
            m_bmpBitmap.Detach();
        m_bmpBitmap.Attach(hBitmap);
    }
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);
    pDC->BitBlt(100,100,300,300,&dcMem,0,0,SRCCOPY);
}
```

- c) Pada header file Project2View.h, tambahkan cuplikan program berikut ini :

```
// Attributes
public:
    CProject2Doc* GetDocument();
    CString name;
```

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

- d) Jalankan program dengan menekan tombol execute atau !
- e) Pilih menu *Test* → *OpenFile* → pilih salah satu gambar yang ada pada komputer anda!



4. TUGAS PRAKTIKUM

Buatlah sebuah project yang didalamnya terdapat menu-menu untuk pemrosesan citra (buat atas inisiatif anda sendiri) dimana semua kode program dimasukkan kedalam Class menu : CnamaprojectView dan proses pengambilan gambar diletakkan didalam menu File → Open !

MODUL 2

PRE-PROCESSING – IMAGE ENHANCEMENT

1. TUJUAN

- a) Mahasiswa dapat menyebutkan macam-macam proses dalam image enhancement
- b) Mahasiswa dapat memahami proses dalam image enhancement
- c) Mahasiswa dapat menggunakan fasilitas image enhancement untuk memperbaiki gambar agar menjadi lebih baik

2. DASAR TEORI

Pre – Processing – Image Enhancement merupakan teknik yang digunakan dalam perbaikan citra atau gambar dari berbagai sebab dengan tujuan untuk mendapatkan hasil citra atau gambar yang jauh lebih baik, sehingga memudahkan untuk diproses untuk keperluan tertentu. Pada bab ini, akan dijelaskan beberapa macam image enhancement seperti tersebut dibawah ini :

a) Intensity Compensation – Brightness Enhancement

Brightness adalah proses penambahan kecerahan dari suatu nilai derajat keabuan. Proses brightness ini dilakukan dengan menambahkan nilai derajat keabuan dengan suatu nilai penambah :

$$xb = x + b$$

dimana

x = nilai derajat keabuan

b = nilai penambah

xb = hasil brightness

b) Contrast Enhancement

Contrast adalah proses pengaturan nilai range interval pada setiap nilai derajat keabuan, dan didefinisikan dengan :

$$xk = k \bullet x$$

dimana

x = nilai derajat keabuan

b = nilai penambah

k = nilai kontras

x_k = nilai setelah pengaturan kontras

c) Noise Reduction

- Noise pada citra

Pada saat proses capture (pengambilan gambar), ada beberapa gangguan yang mungkin terjadi, seperti kamera tidak fokus atau munculnya bintik-bintik yang bisa jadi disebabkan oleh proses capture yang tidak sempurna. Setiap gangguan pada citra dinamakan noise. Noise pada citra tidak hanya terjadi karena ketidaksempurnaan dalam proses capture, tetapi juga bisa disebabkan oleh kotoran-kotoran yang terjadi pada citra. Berdasarkan bentuk dan karakteristiknya, noise pada citra dibedakan menjadi beberapa macam, yaitu :

- Gaussian

Noise gaussian merupakan noise yang mengikuti distribusi normal standar dengan rata-rata nol dan standar deviasi 1. Efeknya adalah pada gambar muncul titik-titik berwarna yang jumlahnya sama dengan prosentase noise.

- Speckle

Noise Speckle merupakan model noise yang memberikan warna hitam pada titik yang terkena noise.

- Salt & Pepper

Salt & Pepper merupakan model noise seperti halnya taburan garam, akan memberikan titik-titik putih pada titik yang terkena noise.

Terkadang dalam pengolahan citra, noise dibangkitkan dengan metode pembangkitan noise tersendiri.

- Reduksi Noise

Ada berbagai macam cara atau teknik untuk mengurangi (mereduksi) noise, salah satunya adalah menggunakan filter rata-rata (mean filter). Dalam pengertian noise sebagai suatu nilai yang berbeda dengan semua titik tetangganya maka dapat dikatakan noise merupakan nilai-nilai yang berada pada frekuensi tinggi. Untuk

mengurangnya digunakan low pass filter (LPF) yang salah satu contohnya adalah filter rata-rata (mean filter).

Filter rata-rata merupakan filter H dalam bentuk matrik yang berukuran mxn, dan nilainya adalah sama untuk setiap elemen dan karena bersifat LPF maka jumlah seluruh elemen adalah satu dan dituliskan sebagai berikut :

$$H(i, j) = \frac{1}{m, n}, 1 \leq i \leq m, 1 \leq j \leq n$$

Filter rata-rata berukuran 3x3 adalah :

$$H = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \text{ atau } H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot 1/9$$

d) Gray Scale

Proses awal yang banyak dilakukan dalam image processing adalah mengubah citra berwarna menjadi gray scale, hal ini digunakan untuk menyederhanakan model citra. Seperti telah dijelaskan pada percobaan sebelumnya, citra berwarna terdiri dari tiga layer matrik, yaitu R-layer, G-layer dan B-layer. Bila proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga kali perhitungan yang caranya sama. Sehingga konsep itu dirubah dari 3-layer ke 1-layer saja dan hasilnya dinamakan gray scale. Dalam citra model ini, tidak ada lagi warna, tetapi derajat keabuan.

Untuk mengubah citra berwarna menjadi gray scale maka nilai masing-masing matrik R, G dan B diubah ke sebuah nilai x, dimana :

$$x = \frac{R + G + B}{3}$$

3. PROSEDUR PERCOBAAN

- A. Sebelumnya, anda buat dahulu sebuah project seperti pada praktikum 1.
- B. Kemudian aktifkan menu **File – Open** pada classwizard untuk membuka file gambar yang ada pada harddisk anda, dengan menambahkan cuplikan program berikut ini pada file CnamaprojectView.cpp sbb :


```

void CNamaprojectView::OnFileOpen()
{
    // TODO: Add your command handler code here
    static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)|*.bmp|";

    CFileDialog m_ldFile(TRUE, "*.bmp", name,
    OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);
    if(m_ldFile.DoModal()!=IDOK)
    {
        name=m_ldFile.GetPathName();
        LoadGambar();
    }
}

void CNamaprojectView::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(100,50,300,250,&dcMem,0,0,SRCCOPY);
}

```

Dan CnamaprojectView.h sbb :

```

// Attributes
public:
    CNamaprojectDoc* GetDocument();
    void LoadGambar (void);

// Operations
public:
    CString name;
    CBitmap m_bmpBitmap;

```

- C. Kemudian tambahkan cuplikan program dibawah ini untuk konversi data dari pixel ke RGB dan dari RGB ke pixel pada file CnamaprojectView.cpp sbb :

```
// merubah data pixel ke RGB
void WarnaToRGB(long int warna,int *Red, int *Green, int *Blue)
{
    *Red = warna & 0x000000FF;
    *Green = (warna & 0x0000FF00) >> 8;
    *Blue = (warna & 0x00FF0000) >> 16;
}

//merubah RGB ke data pixel
long int RGBToWarna(int Red, int Green, int Blue)
{
    return(Red+(Green<<8)+(Blue<<16));
}
```

Dan pada file CnamaprojectView.h sbb :

```
// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void WarnaToRGB(long int warna,int *Red, int *Green, int *Blue);
    long int RGBToWarna(int Red, int Green, int Blue);
```

- D. Kemudian anda buat dan aktifkan sub menu pada docking bar pada classwizard yang meliputi proses-proses sebagai berikut :

- Brightness Enhancement
- Contrast Enhancement
- Noise Reduction
- Gray Scale

- E. Pada sub-sub menu yang telah anda buat pada langkah C tersebut, isilah cuplikan program berikut ini pada file CnamaprojectView.cpp dan CnamaprojectView.h

- Brightness Enhancement

Pada file CnamaprojectView.cpp sbb :

```
void CnamaprojectView::Brightness()
{
    // TODO: Add your command handler code here
    CDC* pDC = GetDC();
```

```

CDC dcMem;
int i,j,red,green,blue;
long int warna;
dcMem.CreateCompatibleDC(pDC);
dcMem.SelectObject(&m_bmpBitmap);

for(i=0;i<200;i++)
    for(j=0;j<200;j++)
    {
        warna=dcMem.GetPixel(j,i);
        // merubah data pixel ke RGB
        WarnaToRGB(warna,&red,&green,&blue);

        // memperjelas gambar dengan menambah 50
        red=red+50;
        if(red>255) red=255;
        green=green+50;
        if(green>255) green=255;
        blue=blue+50;
        if(blue>255) blue=255;
        // merubah RGB ke data pixel
        warna=RGBToWarna(red,green,blue);
        dcMem.SetPixel(j,i,warna);
    }
pDC->BitBlt(500,50,700,250,&dcMem,0,0,SRCCOPY);
}

```

Pada file CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void Brightness (void);

```

➤ Contrast Enhancement

Pada file CnamaprojectView.cpp sbb :

```

void CnamaprojectView::Contrast()
{
    // TODO: Add your command handler code here
    CDC* pDC = GetDC();
    CDC dcMem;
    int i,j,red,green,blue;
    long int warna;
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);

```

```

for(i=0;i<200;i++)
    for(j=0;j<200;j++)
    {
        warna=dcMem.GetPixel(j,i);
        // merubah data pixel ke RGB
        WarnaToRGB(warna,&red,&green,&blue);

        // memperjelas gambar dengan mengalikan 2
        red=red*2;
        if(red>255) red=255;
        green=green*2;
        if(green>255) green=255;
        blue=blue*2;
        if(blue>255) blue=255;
        // merubah RGB ke data pixel
        warna=RGBToWarna(red,green,blue);
        dcMem.SetPixel(j,i,warna);
    }
pDC->BitBlt(500,50,700,250,&dcMem,0,0,SRCCOPY);
}

```

Pada file CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void Contrast (void);

```

➤ Noise Reduction

Pada file CnamaprojectView.cpp sbb :

```

void CnamaprojectView::NoiseReduction()
{
    // TODO: Add your command handler code here
    CnamaprojectDoc* pDoc = GetDocument();
    CDC* pDC = GetDC();
    CDC dcMem;
    int i,j,k,l,resultr,resultg,resultb;
    long int w, mat[3][3];
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);

    for(i=0;i<200;i++)
        for(j=0;j<200;j++)

```

```

    {
        mat[0][0]=dcMem.GetPixel(j-1,i-1);
        mat[0][1]=dcMem.GetPixel(j,i-1);
        mat[0][2]=dcMem.GetPixel(j+1,i-1);
        mat[1][0]=dcMem.GetPixel(j-1,i);
        mat[1][1]=dcMem.GetPixel(j,i);
        mat[1][2]=dcMem.GetPixel(j+1,i);
        mat[2][0]=dcMem.GetPixel(j-1,i+1);
        mat[2][1]=dcMem.GetPixel(j,i+1);
        mat[2][2]=dcMem.GetPixel(j+1,i+1);

        resultr=0;resultg=0;resultb=0;
        for(k=0;k<3;k++)
            for(l=0;l<3;l++)
                {
                    WarnaToRGB(mat[k][l],&r,&g,&b);
                    resultr=resultr+r;
                    resultg=resultg+g;
                    resultb=resultb+b;
                }
            resultr=resultr/9;
            resultg=resultg/9;
            resultb=resultb/9;
            if(resultr>255)resultr=255;
            if(resultg>255)resultg=255;
            if(resultb>255)resultb=255;
            w=RGBToWarna(resultr,resultg,resultb);
            dcMem.SetPixel(j,i,w);
    }
    pDC->BitBlt(500,50,700,250,&dcMem,0,0,SRCCOPY);
}

```

Pada file CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void NoiseReduction (void);

```

➤ Gray Scale

Pada file CnamaprojectView.cpp sbb :

```

void CnamaprojectView::GrayScale()
{
    // TODO: Add your command handler code here

```

```

CDC* pDC = GetDC();
CDC dcMem;
int i,j,red,green,blue,gray;
long int warna;
dcMem.CreateCompatibleDC(pDC);
dcMem.SelectObject(&m_bmpBitmap);

for(i=0;i<200;i++)
    for(j=0;j<200;j++)
    {
        warna=dcMem.GetPixel(j,i);
        // merubah data pixel ke RGB
        WarnaToRGB(warna,&red,&green,&blue);

        // mengubah warna menjadi gray scale
        gray=(red+green+blue)/3;
        // merubah RGB ke data pixel
        warna=RGBToWarna(gray, gray, gray);
        dcMem.SetPixel(j,i,warna);
    }
pDC->BitBlt(500,50,700,250,&dcMem,0,0,SRCCOPY);
}

```

Pada file CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void GrayScale (void);

```

F. Ujilah hasil program yang telah anda buat !

4. TUGAS PRAKTIKUM

- a) *Dari program yang telah anda buat, tambahkan menu pop-up pada sub menu brightness enhancement untuk perbaikan / memperjelas gambar dengan pilihan 10%, 25% dan 50% brightness.*
- b) *Tambahkan menu pop-up pada sub menu contrast enhancement untuk perbaikan kontras gambar dengan pilihan 50%, 100% dan 150% kontras.*

MODUL 3

FEATURE EXTRACTION

1. TUJUAN

- Mahasiswa dapat menyebutkan macam-macam proses dalam image feature extraction / ekstraksi ciri citra.
- Mahasiswa dapat memahami proses dalam image feature extraction / ekstraksi ciri citra.
- Mahasiswa dapat menggunakan fasilitas image feature extraction / ekstraksi ciri citra untuk mengenali atau mengidentifikasi citra.

2. DASAR TEORI

a) *Histogram*

Banyak sekali proses pengolahan citra yang melibatkan distribusi data, seperti pada contoh konversi biner. Bahkan dalam image enhancement (perbaikan citra), distribusi dari nilai derajat keabuan pada citra menjadi suatu acuan dasar. Untuk menyatakan distribusi data dari nilai derajat keabuan ini dapat digunakan nilai histogram. Histogram adalah fungsi yang menyatakan jumlah kemunculan dari setiap nilai.

Misalkan diketahui data sbb :

$$X = 1 \ 3 \ 2 \ 5 \ 3 \ 0 \ 2 \ 1 \ 2 \ 4 \ 3 \ 2$$

Maka histogramnya adalah kemunculan dari angka 0 = 1 kali, angka 1 = 2 kali, angka 2 = 4 kali dst. Karena citra mempunyai derajat keabuan 256 yaitu dari 0 sampai 255, maka histogram menyatakan jumlah kemunculan nilai tertentu antara 0 - 255.

b) *Edge Detection*

Deteksi tepi (Edge Detection) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek citra, tujuannya adalah :

- Untuk menandai bagian yang menjadi detail citra
- Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra.

Suatu titik (x,y) dikatakan sebagai tepi (edge) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

Berdasarkan prinsip-prinsip filter pada citra, maka tepi suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF) yang mempunyai karakteristik :

$$\sum_y \sum_x H(x, y) = 0$$

Untuk mencoba perhitungan diatas, dapat dilakukan dengan cara manual menggunakan perhitungan konvolusi.

Macam-macam metode untuk proses deteksi tepi antara lain :

➤ Metode Robert

Metode Robert adalah nama lain dari teknik differensial yang dikembangkan diatas, yaitu differensial pada arah horisontal dan differensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Kernel filter yang digunakan :

$$H = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad \text{dan} \quad V = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

➤ Metode Prewitt

Metode Prewitt merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kernel filter yang digunakan :

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

➤ Metode Sobel

Metode Sobel merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Kernel filter yang digunakan :

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

c) Image Segmentation

Sebuah citra dapat direduksi ukurannya dengan cara mengelompokkan beberapa nilai red, green dan blue pada beberapa pixel sehingga dapat dijadikan menjadi satu nilai derajat keabuan yang sama.

Kernel dari fungsi image segmentation adalah sebagai berikut :

$$Seg_red = \frac{\sum_x \sum_y red}{x \bullet y}$$

$$Seg_green = \frac{\sum_x \sum_y green}{x \bullet y}$$

$$Seg_blue = \frac{\sum_x \sum_y blue}{x \bullet y}$$

Jika ingin didapatkan sebuah nilai gray saja dari ketiga nilai tersebut diatas, maka persamaan diatas dapat diubah menjadi :

$$Seg_gray = \frac{Seg_red + Seg_green + Seg_blue}{3}$$

d) Fast Fourier Transform

Sekarang kita dapat menampilkan data frekuensi dari citra, bagaimana caranya? Masing-masing pixel pada spektrum menyatakan perubahan frekuensi spasial dari satu siklus per lebar citra. Jika setiap pixel pada citra adalah gray, hanya memiliki satu nilai spektrum. Pixel berikutnya ke kanan dari pixel asli menyatakan satu siklus per lebar citra. Pixel berikutnya ke kanan dari pixel asli menyatakan dua siklus per lebar citra dan seterusnya. pixel paling kanan dari pixel asli adalah menyatakan frekuensi spasial yang lebih tinggi.

TABLE 7.2 Savings when using the FFT on 2-dimensional data.

Image Size	DFT Multiplications	DFT Time	FFT Multiplications	FFT Time
256 X 256	4.3 E 9	71 min	1,048,576	1.0 sec
512 X 512	6.8 E 10	19 hr	4,718,592	4.8 sec
1024 X 1024	1.1 E 12	12 days	20,971,520	21.0 sec
2048 X 2048	1.8 E 13	203 days	92,274,688	92.2 sec

3. PROSEDUR PERCOBAAN

A. Sebelumnya, anda buat dahulu sebuah project seperti pada praktikum 1.

B. Kemudian aktifkan menu *File – Open* pada classwizard untuk membuka file gambar yang ada pada harddisk anda, dengan menambahkan cuplikan program berikut ini pada file CnamaprojectView.cpp sbb :

```
void CnamaprojectView::OnFileOpen()
{
    // TODO: Add your command handler code here
    static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)*.bmp|";

    CFileDialog m_ldFile(TRUE, "*.bmp", name,
    OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);
    if(m_ldFile.DoModal()=IDOK)
    {
        name=m_ldFile.GetPathName();
        LoadGambar();
    }
}

void CnamaprojectView::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(100,50,300,250,&dcMem,0,0,SRCCOPY);
    }

```

Dan CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void LoadGambar (void);

// Operations
public:
    CString name;
    CBitmap m_bmpBitmap;

```

- C. Kemudian tambahkan cuplikan program dibawah ini untuk konversi data dari pixel ke RGB dan dari RGB ke pixel pada file CnamaprojectView.cpp sbb :

```

// merubah data pixel ke RGB
void CnamaprojectView::WarnaToRGB(long int warna,int *Red, int *Green, int *Blue)
{
    *Red = warna & 0x000000FF;
    *Green = (warna & 0x0000FF00) >> 8;
    *Blue = (warna & 0x00FF0000) >> 16;
}

//merubah RGB ke data pixel
long int CnamaprojectView::RGBToWarna(int Red, int Green, int Blue)
{
    return(Red+(Green<<8)+(Blue<<16));
}

```

Dan pada file CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void WarnaToRGB(long int warna,int *Red, int *Green, int *Blue);
    long int RGBToWarna(int Red, int Green, int Blue);

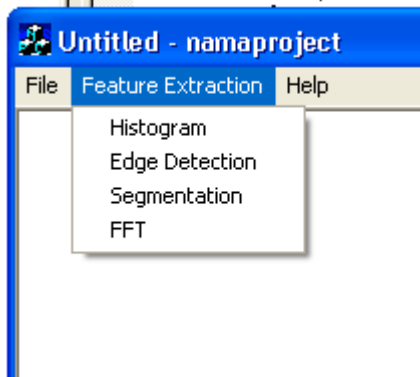
```

- D. Kemudian anda buat sebuah menu “Feature Extraction” dan aktifkan sub menu pada docking bar tersebut pada classwizard yang meliputi proses-proses sebagai berikut :

➤ Histogram

- Edge Detection
- Segmentation
- Fast Fourier Transform (FFT)

Seperti pada gambar berikut ini :



E. Pada sub-sub menu yang telah anda buat pada langkah C tersebut, isilah cuplikan program berikut ini pada file CnamaprojectView.cpp dan CnamaprojectView.h

- Histogram

```
void CnamaprojectView::OnFeatureextractionHistogram()
{
    // TODO: Add your command handler code here
    CDC* pDC = GetDC();
    CDC dcMem;
    int i,j,k,red,green,blue,hist[256],gray,maks;
    long int warna;
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);

    for(i=0;i<256;i++)
        hist[i]=0;

    for(j=0;j<200;j++)
    {
        for(k=0;k<200;k++)
        {
            warna=dcMem.GetPixel(j,k);

            // merubah pixel ke data RGB
            WarnaToRGB(warna,&red,&green,&blue);

            // merubah warna menjadi gray scale
            gray=(red+green+blue)/3;
            hist[gray]=hist[gray]+1;
        }
    }
}
```

```

maks=hist[0];
for(i=0;i<255;i++)
{
    if(maks<hist[i])
        maks=hist[i];
}

pDC->MoveTo(300,100);
pDC->LineTo(300,300);
pDC->LineTo(560,300);

pDC->MoveTo(300,300);
for(i=0;i<255;i++)
{
    pDC->LineTo(300+i,300-hist[i]*200/maks);
}
}

```

➤ Edge Detection

```

void CNamaprojectView::OnFeatureextractionEdgedetection()
{
    // TODO: Add your command handler code here
    CDC* pDC = GetDC();
    CDC dcMem;
    int i,j,r,g,b;
    int resultr,resultg,resultb;
    int r1,r2,r3,r4,r5,r6,r7,r8,r9;
    int g1,g2,g3,g4,g5,g6,g7,g8,g9;
    int b1,b2,b3,b4,b5,b6,b7,b8,b9;
    long int mat[3][3],temp[250][210];
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);
    for(i=0;i<200;i++)
    {
        for(j=0;j<200;j++)
        {
            mat[0][0]=dcMem.GetPixel(j-1,i-1);
            mat[0][1]=dcMem.GetPixel(j,i-1);
            mat[0][2]=dcMem.GetPixel(j+1,i-1);
            mat[1][0]=dcMem.GetPixel(j-1,i);
            mat[1][1]=dcMem.GetPixel(j,i);
            mat[1][2]=dcMem.GetPixel(j+1,i);
            mat[2][0]=dcMem.GetPixel(j-1,i+1);
            mat[2][1]=dcMem.GetPixel(j,i+1);
            mat[2][2]=dcMem.GetPixel(j+1,i+1);

            WarnaToRGB(mat[0][0],&r,&g,&b);r1=r;g1=g;b1=b;
            WarnaToRGB(mat[0][1],&r,&g,&b);r2=r;g2=g;b2=b;
            WarnaToRGB(mat[0][2],&r,&g,&b);r3=r;g3=g;b3=b;
            WarnaToRGB(mat[1][0],&r,&g,&b);r4=r;g4=g;b4=b;
            WarnaToRGB(mat[1][1],&r,&g,&b);r5=r;g5=g;b5=b;
            WarnaToRGB(mat[1][2],&r,&g,&b);r6=r;g6=g;b6=b;
            WarnaToRGB(mat[2][0],&r,&g,&b);r7=r;g7=g;b7=b;

```

```

        WarnaToRGB(mat[2][1],&r,&g,&b);r8=r;g8=g;b8=b;
        WarnaToRGB(mat[2][2],&r,&g,&b);r9=r;g9=g;b9=b;

        // metode edge detection prewitt
        resultr=abs(r3+r6+r9-r1-r4-r7)+abs(r7+r8+r9-r1-r2-r3);
        resultg=abs(g3+g6+g9-g1-g4-g7)+abs(g7+g8+g9-g1-g2-g3);
        resultb=abs(b3+b6+b9-b1-b4-b7)+abs(b7+b8+b9-b1-b2-b3);

        if(resultr>=255)resultr=255;
        if(resultg>=255)resultg=255;
        if(resultb>=255)resultb=255;

        temp[j][i]=RGBToWarna(resultr,resultg,resultb);
    }
}

for(i=0;i<200;i++)
{
    for(j=0;j<200;j++)
    {
        dcMem.SetPixel(j,i,temp[j][i]);
    }
}

pDC->BitBlt(500,100,700,300,&dcMem,0,0,SRCCOPY);
}

```

➤ Segmentation

```

void CNamaprojectView::OnFeatureextractionSegmentation()
{
    // TODO: Add your command handler code here
    CDC* pDC = GetDC();
    CDC dcMem;
    int i,j,k,l,red,green,blue;
    int resultr,resultg,resultb;
    long int warna;
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);

    for(i=0;i<200;i+=10)
    {
        for(j=0;j<200;j+=10)
        {
            resultr=0;resultg=0;resultb=0;
            for(k=0;k<10;k++)
            {
                for(l=0;l<10;l++)
                {
                    warna=dcMem.GetPixel(i+k,j+l);

                    // merubah pixel ke data RGB
                    WarnaToRGB(warna,&red,&green,&blue);

                    resultr+=red;

```

```

        resultg+=green;
        resultb+=blue;
    }
}

resultr/=100;
resultg/=100;
resultb/=100;

for(k=0;k<10;k++)
{
    for(l=0;l<10;l++)
    {
        // merubah RGB ke data pixel
        warna=RGBToWarna(resultr,resultg,resultb);
        dcMem.SetPixel(i+k,j+l,warna);
    }
}
}
pDC->BitBlt(500,100,700,300,&dcMem,0,0,SRCCOPY);
}

```

➤ Fast Fourier Transform

Pada file **CnamaprojectView.cpp** :

```

void CnamaprojectView::OnFeatureextractionFft()
{
    // TODO: Add your command handler code here
    CDC* pDC = GetDC();
    CDC dcMem;
    int i,j,red,green,blue,gray,nx=128,ny=128,dir=1,mag;
    long int warna;
    float temp[250][250];
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);

    for(i=0;i<128;i++)
    {
        for(j=0;j<128;j++)
        {
            warna=dcMem.GetPixel(j,i);
            // merubah data pixel ke RGB
            WarnaToRGB(warna,&red,&green,&blue);

            // mengubah warna menjadi gray scale
            gray=(red+green+blue)/3;
            temp[i][j]=(float)gray;
            // merubah RGB ke data pixel
            warna=RGBToWarna(gray, gray, gray);
            dcMem.SetPixel(j,i,warna);
        }
    }
}

```

```

pDC->TextOut(300,50,"Gray Scale");
pDC->BitBlt(300,100,500,300,&dcMem,0,0,SRCCOPY);

for(i=0;i<128;i++)
{
    for(j=0;j<128;j++)
    {
        cov.c[i][j].re=temp[i][j];
        cov.c[i][j].im=0;
    }
}

FFT2D(nx,ny,dir);

for(i=0;i<128;i++)
{
    for(j=0;j<128;j++)
    {
        mag=(int)sqrt(pow(result.c[i][j].re,2)+pow(result.c[i][j].im,2));
        warna=RGBToWarna(mag,mag,mag);
        dcMem.SetPixel(j,i,warna);
    }
}

pDC->TextOut(500,50,"FFT");
pDC->BitBlt(500,100,700,300,&dcMem,0,0,SRCCOPY);
}

void CNamaprojectView::FFT2D(int rows,int cols,int dir)
{
    int Row,Col;
    float real[256],imag[256];
    int i,j;

    // Compute power of 2s
    Row=(int)(log10(rows)/log10(2));
    Col=(int)(log10(cols)/log10(2));

    // Transform the rows
    for(i=0;i<cols;i++)
    {
        for(j=0;j<rows;j++)
        {
            real[j]=cov.c[j][i].re;
            imag[j]=cov.c[j][i].im;
        }
        FFT(real,imag,Row,dir);
        for(j=0;j<rows;j++)
        {
            result.c[j][i].re=real[j];
            result.c[j][i].im=imag[j];
        }
    }

    // Transform the columns
    for(i=0;i<rows;i++)

```



```

    {
        for(j=0;j<cols;j++)
        {
            real[j]=result.c[j][i].re;
            imag[j]=result.c[j][i].im;
        }
        FFT(real,imag,Col,dir);
        for(j=0;j<cols;j++)
        {
            result.c[j][i].re=real[j];
            result.c[j][i].im=imag[j];
        }
    }
}

void CNamaprojectView::FFT(float *x,float *y,int l,int mode)
{
    int np,lmx,lo,lix,lm,li,j1,j2,nv2,npm1,i,j,k;
    float scl,arg,c,s,t1,t2;

    for(i=0;i<pow(2,l);i++)
        y[i]=0;

    --x;
    --y;

    /* ===== radix-2 fft ===== */

    np=(int)pow(2.0,(float)l);
    lmx=np;
    scl=(float)(6.283185303/(float)np);

    for(lo=1;lo<=l;++lo)
    {
        lix=lmx;
        lmx=(int)(lmx/2.0);
        arg=0.0;
        for(lm=1;lm<=lmx;++lm)
        {
            c=(float)cos(arg);
            s=(float)(mode*sin(arg));
            arg=arg+scl;
            for (li=lix; lix<0 ? li>=np : li<=np; li+=lix)
            {
                j1=li-lix+lm;
                j2=j1+lmx;
                t1=x[j1]-x[j2];
                t2=y[j1]-y[j2];
                x[j1]=x[j1]+x[j2];
                y[j1]=y[j1]+y[j2];
                x[j2]=c*t1+s*t2;
                y[j2]=c*t2-s*t1;
            }
        }
        scl=(float)(2.0*scl);
    }
}

```

```

/* ===== bit reversal ===== */

j=1;
nv2=(int)(np/2.0);
npm1=np-1;
for(i=1;i<=npm1;++i)
{
    if(i>=j)
        goto L30;
    t1=x[j];
    t2=y[j];
    x[j]=x[i];
    y[j]=y[i];
    x[i]=t1;
    y[i]=t2;

    L30:
    k=nv2;

    L40:
    if(k>=j)
        goto L50;

    j=k;
    k=(int)(k/2.0);
    goto L40;

    L50:
    j=j+k;
}
}

```

Pada file **CnamaprojectView.h** :

```

// namaprojectView.h : interface of the CnamaprojectView class
//
////////////////////////////////////
#ifdef AFX_NAMAPROJECTVIEW_H__4F5CCAD0_7A91_492F_929E_1F1312882AD9__INC
LUDED_
#define AFX_NAMAPROJECTVIEW_H__4F5CCAD0_7A91_492F_929E_1F1312882AD9__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

struct complex
{
    float re,im;
};

struct cover
{

```

```
        struct complex c[256][256];
};

struct cover cov,result;

// Operations
public:
    void FFT2D(int ,int ,int );
    void FFT(float *,float *,int ,int );
```

4. Ujilah hasil program yang telah anda buat !

MODUL 4

IMAGE RECOGNITION

1. TUJUAN

- Mahasiswa dapat menyebutkan macam-macam proses dalam image recognition / pengenalan citra.
- Mahasiswa dapat memahami proses dalam image recognition / pengenalan citra.
- Mahasiswa dapat menggunakan fasilitas image recognition / pengenalan citra untuk mengenali atau mengidentifikasi citra.

2. DASAR TEORI

a) *RGB Distance*

Ciri warna suatu gambar dapat dinyatakan dalam bentuk histogram dari gambar tersebut yang dituliskan dengan : $H(r,g,b)$, dimana $H(r,g,b)$ adalah jumlah munculnya pasangan warna r (red), g (green) dan b (blue) tertentu.

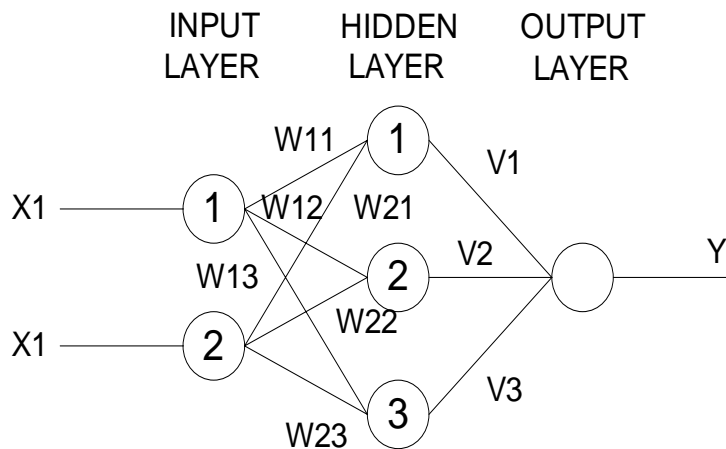
Pada pembahasan tentang RGB Distance ini, dua gambar dapat dikatakan mirip jika nilai histogramnya mempunyai jarak yang kecil dan dituliskan sebagai berikut :

$$\sqrt{\sum_{r=0}^{255} \sum_{g=0}^{255} \sum_{b=0}^{255} (H1(r, g, b) - H2(r, g, b))^2} < \varepsilon$$

dimana ε adalah nilai threshold yang ditentukan sebagai derajat kemiripan.

b) *Jaringan Syaraf Tiruan (JST)*

Salah satu metode pengenalan dan identifikasi modern yang sering digunakan dalam image recognition adalah Jaringan Syaraf Tiruan atau ANN (Adaptif Neural Network). Metode ANN merupakan tiruan dari pola otak manusia, dimana didalamnya terdiri atas neuron, dendrite, dsb. Dalam kenyataannya, pola ANN dapat digambarkan sebagai berikut :



Dimana setiap lapisan menggambarkan urutan dari lapisan dimana sinyal masukan datang, kemudian diteruskan ke lapisan tengah dan terakhir ke lapisan keluaran atau hasil. Seperti halnya manusia, semua proses dapat dimengerti dan dipahami setelah melalui proses pembelajaran, yaitu proses pemahaman sistem ANN atau otak manusia terhadap sesuatu dengan cara mempelajari dari contoh (dalam hal ini adalah target). Salah satu metode pembelajaran yang cukup terkenal dalam ANN adalah Backpropagasi, dimana terdiri dari dua arah pembelajaran, yaitu maju dan mundur. Pada arah maju, terjadi proses pemasukan sinyal masukan sampai menuju kearah hasil. Selanjutnya hasil dibandingkan dengan target, jika terjadi kesalahan / error, maka error dikembalikan / dirambatkan mundur / back propagate kearah sebaliknya / kearah masukan. Pada saat yang sama, bobot dari tiap dendrite dikoreksi berdasarkan kesalahan / error dari keluaran. Tentunya proses pembelajaran seperti ini tidak cukup hanya sekali dua kali saja, tetapi sampai ratusan, bahkan ribuan kali, yang dinamakan sebagai iterasi atau epoch. Tetapi untuk komputer hal ini tidaklah menjadi sebuah masalah serius.

3. PROSEDUR PERCOBAAN

- A. Sebelumnya, anda buat dahulu sebuah project seperti pada praktikum 1.
- B. Kemudian aktifkan menu *File – Open1* dan *File – Open2* pada classwizard untuk membuka file gambar yang ada pada harddisk anda, dengan menambahkan cuplikan program berikut ini pada file CnamaprojectView.cpp sbb :

```
void CnamaprojectView::OnFileOpen1()
{
```

```

// TODO: Add your command handler code here
static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)|*.bmp|";

CFileDialog m_IdFile(TRUE, "*.bmp", name1,
OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);
if(m_IdFile.DoModal()==IDOK)
{
    name1=m_IdFile.GetPathName();
    LoadGambar1();
}
}

void CNamaprojectView::OnFileOpen2()
{
// TODO: Add your command handler code here
static char BASED_CODE szFilter[]="Bitmap Files (*.bmp)|*.bmp|";

CFileDialog m_IdFile(TRUE, "*.bmp", name2,
OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);
if(m_IdFile.DoModal()==IDOK)
{
    name2=m_IdFile.GetPathName();
    LoadGambar2();
}
}

```

Dan CnamaprojectView.h sbb :

```

// Attributes
public:
    CNamaprojectDoc* GetDocument();
    CString name1,name2;
    CBitmap m_bmpBitmap1,m_bmpBitmap2;
    char buf[20];

```

- C. Kemudian tambahkan cuplikan program dibawah ini untuk konversi data dari pixel ke RGB dan dari RGB ke pixel pada file CnamaprojectView.cpp sbb :

```

// merubah data pixel ke RGB
void CNamaprojectView::WarnaToRGB(long int warna,int *Red, int *Green, int *Blue)
{
    *Red = warna & 0x000000FF;
    *Green = (warna & 0x0000FF00) >> 8;
    *Blue = (warna & 0x00FF0000) >> 16;
}

//merubah RGB ke data pixel
long int CNamaprojectView::RGBToWarna(int Red, int Green, int Blue)
{
    return(Red+(Green<<8)+(Blue<<16));
}

```

Dan pada file CnamaprojectView.h sbb :

```

// Attributes
public:
    CnamaprojectDoc* GetDocument();
    void LoadGambar1(void);
    void LoadGambar2(void);
    void WarnaToRGB(long int warna,int *Red, int *Green, int *Blue);
    long int RGBToWarna(int Red, int Green, int Blue);

```

D. Kemudian anda buat sebuah menu “Recognition” dan aktifkan sub menu pada docking bar tersebut pada classwizard yang meliputi proses-proses sebagai berikut :

- RGB Distance
- Jaringan Syaraf Tiruan

Seperti pada gambar berikut ini :



E. Pada sub-sub menu yang telah anda buat pada langkah C tersebut, isilah cuplikan program berikut ini pada file CnamaprojectView.cpp dan CnamaprojectView.h

- RGB Distance

```

void CnamaprojectView::OnRecognitionRgbdistance()
{
    // TODO: Add your command handler code here
    int selisihRed,selisihGreen,selisihBlue;
    int histRed1[256],histGreen1[256],histBlue1[256];
    int histRed2[256],histGreen2[256],histBlue2[256];
    long int warna1,warna2;
    int i,j,k,red1,green1,blue1,red2,green2,blue2;
    float accRed1=0.0,accGreen1=0.0,accBlue1=0.0;
    float accRed2=0.0,accGreen2=0.0,accBlue2=0.0;
    int tempRed1[256],tempGreen1[256],tempBlue1[256];
    int tempRed2[256],tempGreen2[256],tempBlue2[256];

    CDC* pDC = GetDC();
    CDC dcMem;
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap1);

    for(i=0;i<256;i++)
    {

```

```

        histRed1[i]=0;
        histGreen1[i]=0;
        histBlue1[i]=0;
    }
    for(j=0;j<200;j++)
    {
        for(k=0;k<200;k++)
        {
            warna1=dcMem.GetPixel(k,j);
            WarnaToRGB(warna1,&red1,&green1,&blue1);
            histRed1[red1]=histRed1[red1]+1;
            histGreen1[green1]=histGreen1[green1]+1;
            histBlue1[blue1]=histBlue1[blue1]+1;
        }
    }
    for(i=0;i<256;i++)
    {
        accRed1=accRed1+(float)histRed1[i]/(250*210);
        accGreen1=accGreen1+(float)histGreen1[i]/(250*210);
        accBlue1=accBlue1+(float)histBlue1[i]/(250*210);
        tempRed1[i]=int(accRed1*255);
        tempGreen1[i]=int(accGreen1*255);
        tempBlue1[i]=int(accBlue1*255);
        if(tempRed1[i]>255)
            tempRed1[i]=255;
        if(tempGreen1[i]>255)
            tempGreen1[i]=255;
        if(tempBlue1[i]>255)
            tempBlue1[i]=255;
    }

    dcMem.SelectObject(&m_bmpBitmap2);

    for(i=0;i<256;i++)
    {
        histRed2[i]=0;
        histGreen2[i]=0;
        histBlue2[i]=0;
    }

    for(j=0;j<200;j++)
    {
        for(k=0;k<200;k++)
        {
            warna2=dcMem.GetPixel(k,j);
            WarnaToRGB(warna2,&red2,&green2,&blue2);
            histRed2[red2]=histRed2[red2]+1;
            histGreen2[green2]=histGreen2[green2]+1;
            histBlue2[blue2]=histBlue2[blue2]+1;
        }
    }

    for(i=0;i<256;i++)
    {

```



```

        accRed2=accRed2+(float)histRed2[i]/(250*210);
        accGreen2=accGreen2+(float)histGreen2[i]/(250*210);
        accBlue2=accBlue2+(float)histBlue2[i]/(250*210);
        tempRed2[i]=int(accRed2*255);
        tempGreen2[i]=int(accGreen2*255);
        tempBlue2[i]=int(accBlue2*255);
        if(tempRed2[i]>255)
            tempRed2[i]=255;
        if(tempGreen2[i]>255)
            tempGreen2[i]=255;
        if(tempBlue2[i]>255)
            tempBlue2[i]=255;
    }

    selisihRed=0;selisihGreen=0;selisihBlue=0;
    for(i=0;i<256;i++)
    {
        selisihRed=selisihRed+abs(tempRed1[i]-tempRed2[i]);
        selisihGreen=selisihGreen+abs(tempGreen1[i]-tempGreen2[i]);
        selisihBlue=selisihBlue+abs(tempBlue1[i]-tempBlue2[i]);
    }

    sprintf(buf,"Selisih Red = %g",selisihRed/40000.f);
    pDC->TextOut(100,300,buf);
    sprintf(buf,"Selisih Green = %g",selisihGreen/40000.f);
    pDC->TextOut(300,300,buf);
    sprintf(buf,"Selisih Blue = %g",selisihBlue/40000.f);
    pDC->TextOut(500,300,buf);
}

```

F. Ujilah hasil program yang telah anda buat !

4. TUGAS PRAKTIKUM

- A. Modifikasilah program yang telah anda buat untuk menambahkan sebuah nilai threshold sehingga program anda dapat memberikan keputusan berdasarkan nilai RGB distance-nya dan keputusannya dicetak pada layar view! Misal jika nilai RGB distance < 0.15 maka gambar diputuskan mirip, tetapi jika lebih maka dikatakan berbeda.
- B. Menurut anda, bagaimanakah kira-kira algoritma dari JST atau ANN untuk mengidentifikasi sebuah citra atau gambar? Jelaskan!

MODUL 5

IMAGE CAPTURING VIA WEB CAM

1. TUJUAN

- Mahasiswa dapat memahami proses dalam image capturing / penangkapan citra melalui media video / webcam.
- Mahasiswa dapat menggunakan fasilitas image capturing / penangkapan citra sebagai masukan sistem.

2. DASAR TEORI

Kamera merupakan sebuah alat atau media pengambilan atau penangkapan gambar secara online. Pada dasarnya, video atau gambar bergerak hasil tangkapan sebuah kamera adalah berupa gambar dua dimensi yang terdiri dari kumpulan frame dalam jumlah tertentu per detiknya sehingga dapat membentuk sebuah urutan gambar yang bergerak.

Pada praktikum berikut ini, disajikan cuplikan program untuk meng-on-kan dan meng-off-kan kamera webcam serta meng-capture video menjadi sebuah gambar berformat *.bmp.

3. PROSEDUR PERCOBAAN

- A. Sebelumnya, install driver kamera webcam yang akan anda gunakan ke dalam harddisk anda agar sistem dapat mengenali jenis kamera yang anda pakai. (Untuk informasi lebih lanjut tanyakan kepada Dosen atau Asisten Praktikum).
- B. Selanjutnya, anda buat dahulu sebuah project dengan MFCAppWizard(Exe) bernama “*Video*” seperti pada praktikum 1.
- C. Kemudian buatlah menu “**Kamera**” yang memiliki sub-sub menu “*On*”, “*Off*” dan “*Capture*” pada classwizard untuk meng-on-kan dan meng-off-kan kamera serta meng-capture video menjadi gambar *.bmp, dengan menambahkan cuplikan program berikut ini pada file *CVideoView.cpp* sbb :

```
void CVideoView::OnKameraOff()
{
    // TODO: Add your command handler code here
    m_VFWImageProc.Destroy();
}

void CVideoView::OnKameraOn()
{
```

```

        // TODO: Add your command handler code here
        CAPDRIVERCAPS Caps;
        m_VFWImageProc.Initialize();
        m_VFWImageProc.DriverGetCaps(&Caps);
        m_VFWImageProc.EnablePreviewVideo(*this, 0,0);
    }

void CVideoView::OnKameraCapture()
{
    // TODO: Add your command handler code here
    CDC *pDC=GetDC();
    WndToBmpFile(pDC,"temp.bmp");
    m_VFWImageProc.Destroy();
}

//*****Window to bmp file*****//

BOOL CVideoView::WndToBmpFile(CDC *pDC, CString szFile)
{
    CString fname=szFile;
    CBitmap bmp,*pOldBmp;
    CRect rect;
    CWnd *pWnd;
    BOOL flg=0;
    CPalette pal;
    LOGPALETTE *pLp;

    if(pDC==NULL)        //if pDC is NULL return
    {
        return FALSE;
    }

    pWnd=pDC->GetWindow();    //Get Window of PDC

    pWnd->GetClientRect(&rect); //Get dimension of Window

    if(fname.IsEmpty())
        return FALSE;

    CDC memdc;

    memdc.CreateCompatibleDC(pDC); //Make Compatible DC for memdc
    bmp.CreateCompatibleBitmap(pDC,rect.Width(),rect.Height());
    //Create Compatible DDB
    pOldBmp=memdc.SelectObject(&bmp);
    memdc.BitBlt(0,0,rect.Width(),rect.Height(),pDC,0,0,SRCCOPY);

    //The following code will detect whether the BMP uses a Raster
    //palette or not.

    if(pDC->GetDeviceCaps(RASTERCAPS) & RC_PALETTE)
    {
        int nSize;
        nSize=sizeof(LOGPALETTE) + sizeof(PALETTEENTRY) * 256;
        pLp=(LOGPALETTE*) new BYTE[nSize];
        pLp->palVersion=0x300;
    }
}

```

```

        pLp->palNumEntries=GetSystemPaletteEntries(
            pDC->m_hDC,0,255,pLp->palPalEntry);
        pal.CreatePalette(pLp);

        delete [] pLp;
    }
    memdc.SelectObject(pOldBmp);

    //will convert bitmap from DDB to DIB see DDBToDIB()
    // See DDBToDIB function for more..

    HANDLE hDIB=DDBToDIB bmp,BI_RGB,&pal);
    if(hDIB==NULL)
        return FALSE;

    //*****
    //This code writes the BMP file

    CFile m_file;
    if(!m_file.Open(fname,CFile::modeWrite | CFile::modeCreate,NULL))
        return FALSE;
    BITMAPFILEHEADER hdr;
    LPBITMAPINFOHEADER lpbi;
    lpbi=(LPBITMAPINFOHEADER ) hDIB;
    int nColors= 1 << lpbi->biBitCount;
    hdr.bfType= ((WORD) ('M' << 8) | 'B');
    hdr.bfSize=sizeof(hdr) + GlobalSize(hDIB);
    hdr.bfReserved1=0;
    hdr.bfReserved2=0;
    hdr.bfOffBits=(DWORD) sizeof(hdr) + nColors * sizeof( RGBQUAD);
    m_file.Write(&hdr,sizeof(hdr));
    m_file.Write(lpbi,GlobalSize(hDIB));
    m_file.Close();
    //*****

    //It will open the BMP file in Binary Mode. Then, it will read the first 14
    //bytes from the bitmap file.
    //It will change the 11th byte from 11 to 36.
    //It will change the 14th byte from 4 to 0 because this is the
    //basic requirement for the bitmap format.
    //So, it will support all PaintBrush Tools and thumbnail views
    //of Windows Explorer.

    CFile m_tempFile; //CBinFile is derived from CFile
    BYTE dummy=0;//14 //14
    BYTE pBuf[14]; //11

    BOOL fres=m_tempFile.Open(fname,CFile::modeReadWrite |
        CFile::typeBinary);

    if(fres==0)
        return FALSE;

    UINT tt=m_tempFile.Read(pBuf,14);
    pBuf[13]=dummy;//will replace from 04 to 00
    m_tempFile.SeekToBegin();

```

```

        m_tempFile.Write(pBuf,14);

        m_tempFile.Close();

        return flg;

        //it will capture wnd and save into a bmp file
        //End of the code
    }

HANDLE CVideoView::DDBToDIB(CBitmap &bitmap, DWORD dwCompression, CPalette
*pPal)
{
    BITMAP        bm;
    BITMAPINFOHEADER bi;
    LPBITMAPINFOHEADER lpbi;
    DWORD        dwLen;
    HANDLE        hDIB;
    HANDLE        handle;
    HDC           hDC;
    HPALETTE      hPal;

    ASSERT( bitmap.GetSafeHandle() );

    // The function has no arg for bitfields
    if( dwCompression == BI_BITFIELDS )
        return NULL;

    // If a palette has not been supplied, use default palette
    hPal = (HPALETTE) pPal->GetSafeHandle();
    if (hPal==NULL)
        hPal = (HPALETTE) GetStockObject(DEFAULT_PALETTE);

    // Get bitmap information
    bitmap.GetObject(sizeof(bm),(LPSTR)&bm);

    // Initialize the bitmap infoheader
    bi.biSize      = sizeof(BITMAPINFOHEADER);
    bi.biWidth     = bm.bmWidth;
    bi.biHeight    = bm.bmHeight;
    bi.biPlanes    = 1;
    bi.biBitCount  = bm.bmPlanes * bm.bmBitsPixel;
//bm.bmPlanes * bm.bmBitsPixel;
    bi.biCompression = dwCompression;
    bi.biSizeImage  = 0;
    bi.biXPelsPerMeter = 0;
    bi.biYPelsPerMeter = 0;
    bi.biClrUsed    = 0;
    bi.biClrImportant = 0;

    // Compute the size of the infoheader and the color table

```

```

int nColors = (1 << bi.biBitCount);
if( nColors > 256 )
    nColors = 0;
dwLen = bi.biSize + nColors * sizeof(RGBQUAD);

// We need a device context to get the DIB from
hDC = ::GetDC(NULL);
hPal = SelectPalette(hDC,hPal,FALSE);
RealizePalette(hDC);

// Allocate enough memory to hold bitmap infoheader and
// color table
hDIB = GlobalAlloc(GMEM_FIXED,dwLen);

if (!hDIB){
    SelectPalette(hDC,hPal,FALSE);
    ::ReleaseDC(NULL,hDC);
    return NULL;
}

lpbi = (LPBITMAPINFOHEADER)hDIB;

*lpbi = bi;

// Call GetDIBits with a NULL lpBits param, so the device
// driver will calculate the biSizelImage field
GetDIBits(hDC, (HBITMAP)bitmap.GetSafeHandle(), 0L,
          (DWORD)bi.biHeight,
          (LPBYTE)NULL, (LPBITMAPINFO)lpbi,
          (DWORD)DIB_RGB_COLORS);

bi = *lpbi;

// If the driver did not fill in the biSizelImage field, then
// compute it
// Each scan line of the image is aligned on a DWORD (32bit)
// boundary
if (bi.biSizelImage == 0){
    bi.biSizelImage = (((bi.biWidth * bi.biBitCount) + 31)
        & ~31) / 8 * bi.biHeight;

    // If a compression scheme is used, the result may in fact
    // be larger
    // Increase the size to account for this.
    if (dwCompression != BI_RGB)
        bi.biSizelImage = (bi.biSizelImage * 3) / 2;
}

// Realloc the buffer so that it can hold all the bits
dwLen += bi.biSizelImage;
if (handle = GlobalReAlloc(hDIB, dwLen, GMEM_MOVEABLE))
    hDIB = handle;
else{
    GlobalFree(hDIB);

    // Reselect the original palette

```

```

        SelectPalette(hDC,hPal,FALSE);
        ::ReleaseDC(NULL,hDC);
        return NULL;
    }

    // Get the bitmap bits
    lpbi = (LPBITMAPINFOHEADER)hDIB;

    // FINALLY get the DIB
    BOOL bGotBits = GetDIBits( hDC, (HBITMAP)bitmap.GetSafeHandle(),
    0L, // Start scan line
    (DWORD)bi.biHeight, // # of scan lines
    (LPBYTE)lpbi // address for bitmap bits
    + (bi.biSize + nColors * sizeof(RGBQUAD)),
    (LPBITMAPINFO)lpbi, // address of bitmapinfo
    (DWORD)DIB_RGB_COLORS); // Use RGB for color table

    if( !bGotBits )
    {
        GlobalFree(hDIB);

        SelectPalette(hDC,hPal,FALSE);
        ::ReleaseDC(NULL,hDC);
        return NULL;
    }

    SelectPalette(hDC,hPal,FALSE);
    ::ReleaseDC(NULL,hDC);

    return hDIB;

    //End of the function
}

```

Dan *CVideoView.h* sbb :

```

// videoView.h : interface of the CVideoView class
//
///////////////////////////////////////////////////////////////////
#ifdef AFX_VIDEOSVIEW_H__886DCE03_6A2E_4C91_968B_E5ABF8A828D1__INCLUDED_
#define AFX_VIDEOSVIEW_H__886DCE03_6A2E_4C91_968B_E5ABF8A828D1__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "VFWImageProcessor.h"

class CVideoView : public CView
{
protected: // create from serialization only
    CVideoView();
    DECLARE_DYNCREATE(CVideoView)

```

```

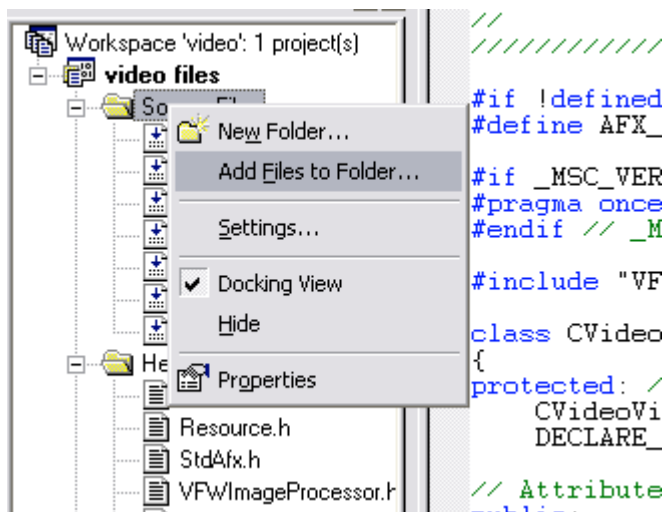
// Attributes
public:
    CVideoDoc* GetDocument();
    HANDLE m_hIDComDev;
    CVFWImageProcessor m_VFWImageProc;

// Operations
public:
    BOOL WndToBmpFile(CDC *pDC, CString szFile);
    HANDLE DDBToDIB(CBitmap &bitmap, DWORD dwCompression, CPalette *pPal);

// Overrides
...

```

- D. Kemudian tambahkan / copy-kan program VFWImageProcessor.cpp dan VFWImageProcessor.h kedalam folder project anda.
- E. Kemudian tambahkan file-file tersebut ke dalam project yang telah anda buat dengan cara dibawah ini :



- F. Ujilah hasil program yang telah anda buat !