

Dasar Pengolahan Citra (1)

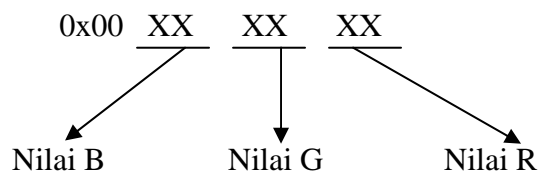
1. Tujuan:

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

2. Dasar Teori:

Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu.

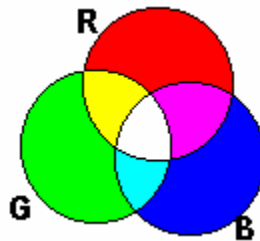
Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna di atas seperti gambar 2.1, variabel 0x00 menyatakan angka dibelakangnya adalah hexadecimal.



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 gabungan 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

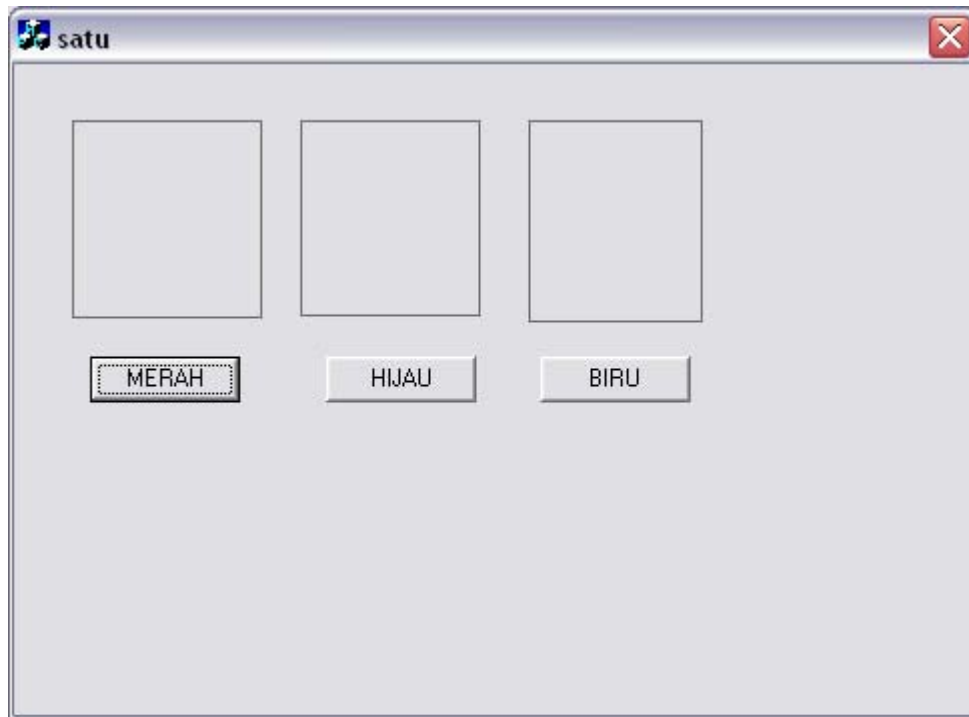
Untuk mengetahui kombinasi warna, perlu dibuat suatu program yang dapat menampilkan warna sesuai dengan nilai yang dimasukkan sehingga dapat dicoba berbagai macam kombinasi warna RGB seperti gambar 2.2.

3. Percobaan:

3.1. Cara Memproses Gambar

1. Cara memproses gambar

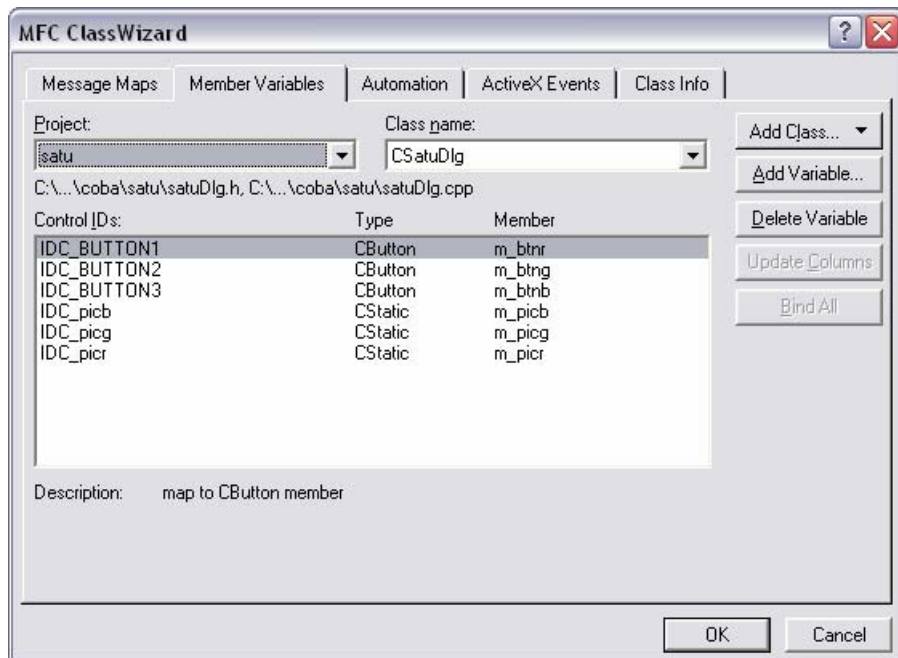
- Buat aplikasi AppWizard seperti pada praktikum 1 dan beri nama project dengan satu
- Buat disain dialog seperti pada gambar 2.3. dengan 3 buah control picture dan 3 buah button



Gambar 2.3. Disain dialog project dialogproses

- Rubah properti control dan tambahkan class variabel control tipe dan membernya seperti pada gambar 2.4

Gambar 2.4. Properti dan Class Member Control dialogproses



- Tambahkan program pada saat button1 click seperti di bawah ini, akan memberikan warna merah pada bitmap (fungsi setpixel) dan menampilkan kembali nilai dari piksel tertentu (fungsi getpixel).

Button1

```
void Cprak2Dlg::OnButton1()
{
    // TODO: Add your control notification handler code here
    CDC* pDC = m_picr.GetDC();
    CDC dcMem;
    CRect rect;
    BITMAP bm; //
    int i, j;
    long int warna;
    char str[5];
    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_picr.GetClientRect(rect);
    m_bmpBitmap.GetBitmap(&bm);
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);
    for(i=0; i<bm.bmHeight; i++)
        for(j=0; j<bm.bmWidth; j++)
        {
            // memberi warna pada titik
            dcMem.SetPixel(j, i, 0x000000FF);
        }
    pDC->StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem,
    0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY);
    // membaca warna pada titik
    warna=dcMem.GetPixel(5, 5);
    sprintf(str, "%ld", warna);
    pDC->TextOut(10, 10, str);
}
```

Button2

```
void Cprak2Dlg::OnButton2()
{
    // TODO: Add your control notification handler code here
    CDC* pDC = m_picg.GetDC();
    CDC dcMem;
    CRect rect;
    BITMAP bm;//
    int i,j;
    long int warna;
    char str[5];
    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_picg.GetClientRect(rect);
    m_bmpBitmap.GetBitmap(&bm);
    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);
    for(i=0;i<bm.bmHeight;i++)
        for(j=0;j<bm.bmWidth;j++)
        {
            // memberi warna pada titik
            dcMem.SetPixel(j,i,0x0000FF00);
        }
    pDC->StretchBlt(0,0,rect.Width(),rect.Height(),&dcMem,
    0,0,bm.bmWidth,bm.bmHeight,SRCCOPY);
    // membaca warna pada titik
    warna=dcMem.GetPixel(5,5);
    sprintf(str,"%ld",warna);
    pDC->TextOut(10,10,str);
}
}
```

Button3

```
void Cprak2Dlg::OnButton3()
{
    // TODO: Add your control notification handler code here
    CDC* pDC = m_picb.GetDC();
    CDC dcMem;
    CRect rect;
    BITMAP bm;//
    int i,j;
    long int warna;
    char str[5];
    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_picb.GetClientRect(rect);
m_bmpBitmap.GetBitmap(&bm);
dcMem.CreateCompatibleDC(pDC);
dcMem.SelectObject(&m_bmpBitmap);
for(i=0;i<bm.bmHeight;i++)
    for(j=0;j<bm.bmWidth;j++)
    {
        // memberi warna pada titik
        dcMem.SetPixel(j,i,0x00FF0000);
    }
pDC->StretchBlt(0,0,rect.Width(),rect.Height(),&dcMem,
0,0,bm.bmWidth,bm.bmHeight,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 prak2Dlg.h
- Tambahkan program seperti dibawah ini

```

// Construction
public:
    Cprak2Dlg(CWnd* pParent = NULL);    // standard constructor
    CBitmap m_bmpBitmap;

```

3. Cara menjalankan program

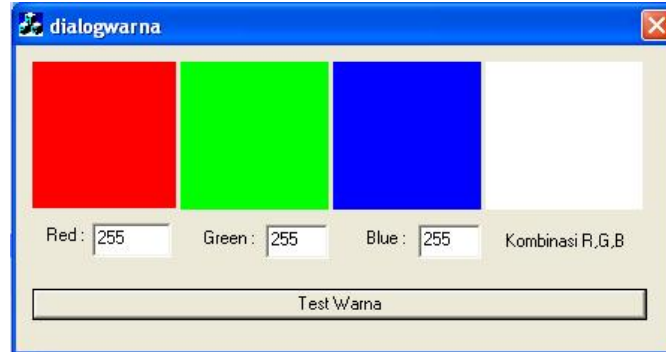
- Pilih menu : Build->Execute (!)
- Click button1 : picture 1 diberi warna merah
- Click button2 : picture 2 diberi warna hijau
- Click button3 : picture 3 diberi warna biru
- Hasil running project dialogproses seperti pada gambar 2.5.



Gambar 2.5 Output

2.5. Latihan:

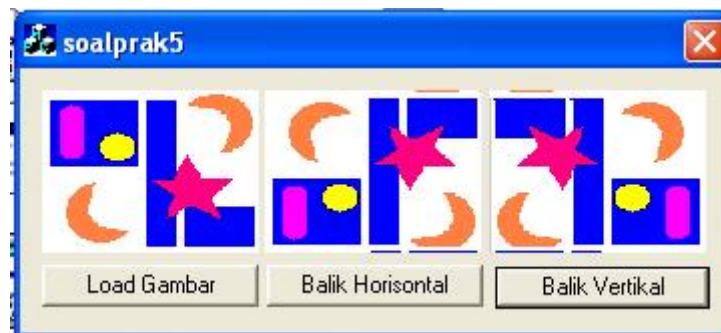
1. Buatlah program untuk mengubah warna 3 buah picture-box dengan tiga macam nilai RGB yang masing-masing bernilai 0-255, sedangkan picture box yang keempat bernilai RGB yang merupakan kombinasi nilai-nilai R, G dan B seperti gambar berikut ini:



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