

I. TUJUAN :

1. Mahasiswa dapat memahami prinsip-prinsip detection pada suatu image
2. Mahasiswa dapat menggunakan proses detection untuk mengetahui suatu gambar yang memiliki warna – warna tertentu
3. Mahasiswa dapat menggunakan static detection dan distance detection untuk mendeteksi suatu warna pada image.

II. DASAR TEORI :

Prinsip-Prinsip Deteksi

- Format Warna
 - RGB
 - r-g Color
 - Normalized RGB
 - HSV
 - YCrCb
 - TSL
- Deteksi Warna
 - Thresholding RGB
 - Thresholding YCrCb

Format Warna Pada Gambar

- Gambar (Digital) adalah sekumpulan titik yang disusun dalam bentuk matriks, dan nilainya menyatakan suatu derajat kecerahan (derajat keabuan/gray-scale). Derajat keabuan 8 bit menyatakan 256 derajat kecerahan.
- Pada gambar berwarna nilai setiap titiknya adalah nilai derajat keabuan pada setiap kompen warna RGB. Bila masing-masing komponen R,G dan B mempunyai 8 bit, maka satu titik dinyatakan dengan $(8+8+8)=24$ bit atau 2^{24} derajat keabuan

Format RGB

- Format RGB (Red, Green & Blue) adalah format dasar yang digunakan oleh banyak peralatan elektronik seperti monitor, LCD atau TV untuk menampilkan sebuah gambar.
- Pada format RGB, suatu warna didefinisikan sebagai kombinasi (campuran) dari komponen warna R, G dan B.

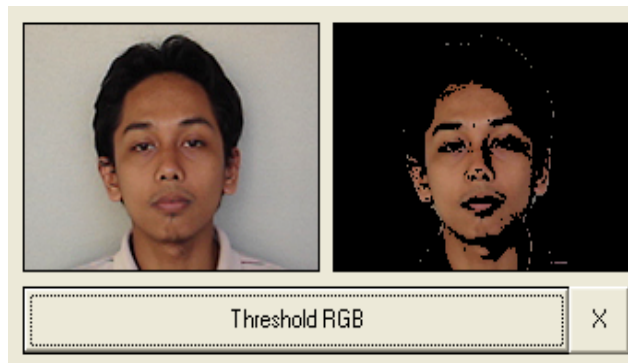
Deteksi Warna

- Mendeteksi adanya warna-warna tertentu
- Menentukan posisi pixel dengan warna yang ditentukan
- Aplikasi: Deteksi rambu-rambu lalu lintas, deteksi bola dengan warna tertentu, deteksi obyek berdasarkan warna, deteksi kulit (skin detection)

Threshold RGB

- Untuk warna-warna dasar, nilai RGB cukup efektif dalam melakukan deteksi meskipun cara ini bukan cara terbaik.
- Nilai threshold dapat dipelajari berdasarkan pola warna atau ditentukan secara intuitif, tetapi threshold RGB ini sangat rentan terhadap kestabilan cahaya
- Aplikasi untk menentukan nilai threshold terbaik dalam suatu permasalahan dapat dilakukan menggunakan teknik-teknik machine learning.

Nilai Threshold RGB



Nilai threshold : $112 < R < 168; 70 < G < 149, 44 < B < 118$

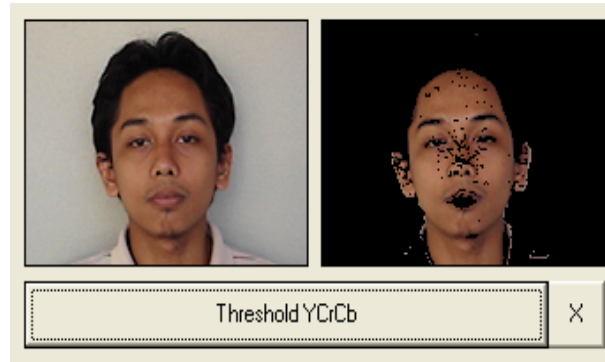
Threshold YCrCb

- YCrCb adalah format warna Yellow, Crominan Red dan Crominan Blue.
- Range nilai masing-masing komponen :

$$0 \leq Y \leq 255$$

$$-255 \leq Cr \leq 255$$

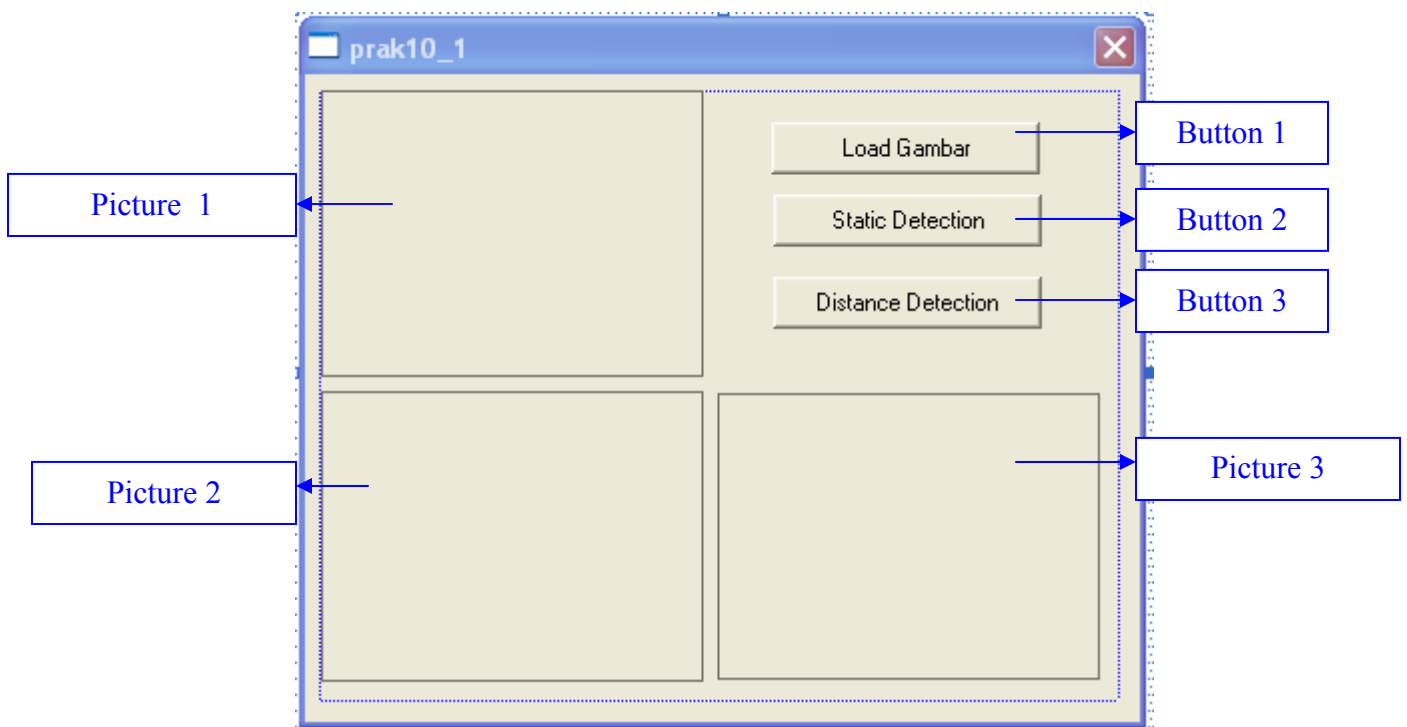
$$-255 \leq Cb \leq 255$$



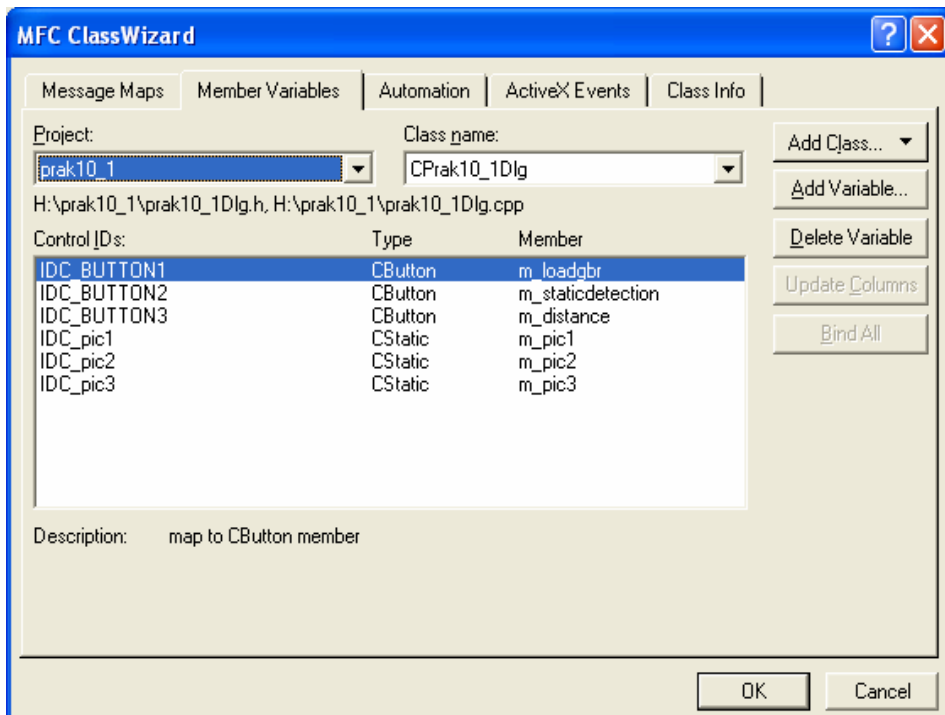
Nilai threshold : $41 < R < 145$; $10 < G < 52$, $-16 < B < -4$

III. HASIL DAN ANALISA PERCOBAAN :

1. Membuat suatu project transparan dua citra dengan melakukan proses deteksi warna dengan menggunakan static detection dan distance detection :
Langkah – langkah membuat project deteksi warna pada suatu citra seperti di atas adalah sebagai berikut :
 - a. Buat aplikasi AppWizard seperti pada praktikum sebelumnya dan beri nama project dengan prak10_1
 - b. Buat disain dialog dengan menggunakan 3 buah control picture, dan 3 buah button.



- c. Merubah properti control dan tambahkan class variabel control tipe dan membernya seperti berikut :



- d. Tambahkan program pada saat button1 click seperti di bawah ini, akan melakukan load gambar dalam kotak control picture dengan cara melakukan browse gambar, pilih gambar yang akan diproses.

```
void CPrak10_1Dlg::OnButton1()
{
    // TODO: Add your control notification handler code here
    static char BASED_CODE szFilter[]="Bitmap Files
    (*.bmp)|*.bmp|";

    CFileDialog m_ldFile(
    OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, szFilter);
    if(m_ldFile.DoModal()==IDOK)
    {
        name=m_ldFile.GetPathName();
    }

    CDC* pDC = m_pic1.GetDC();// mengarah kontrol picture
    CDC dcMem; //mengarahkan obyek yg sdh dipilih pd bitmap
    CRect rect;//kotak di picture
    BITMAP bm;//mendefinisikan fungsi2 yg ada d bitmap
    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);
    }
}
```

Variabel string yg digunakan untuk menyimpan nama gambar .bmp

```

}
m_pic1.GetClientRect(rect); //
m_bmpBitmap.GetBitmap(&bm); //
dcMem.CreateCompatibleDC(pDC);
//pDC->MoveTo(10,190);
//pDC->LineTo(300,190);
dcMem.SelectObject(&m_bmpBitmap);
pDC->StretchBlt(0,0,rect.Width(),rect.Height(),&dcMem,
0,0,bm.bmWidth,bm.bmHeight,SRCCOPY);
}

```

Load Gambar 1 disimpan pada var m_bmpBitmap

e. Menambah header file

- Buka file prak10_1Dlg.h
- Tambahkan program seperti dibawah ini

```

public:
    CPrak10_1Dlg(CWnd* pParent = NULL); // standard constructor

```

```

    CBitmap m_bmpBitmap;

```

Load Gambar disimpan pada var m_bmpBitmap

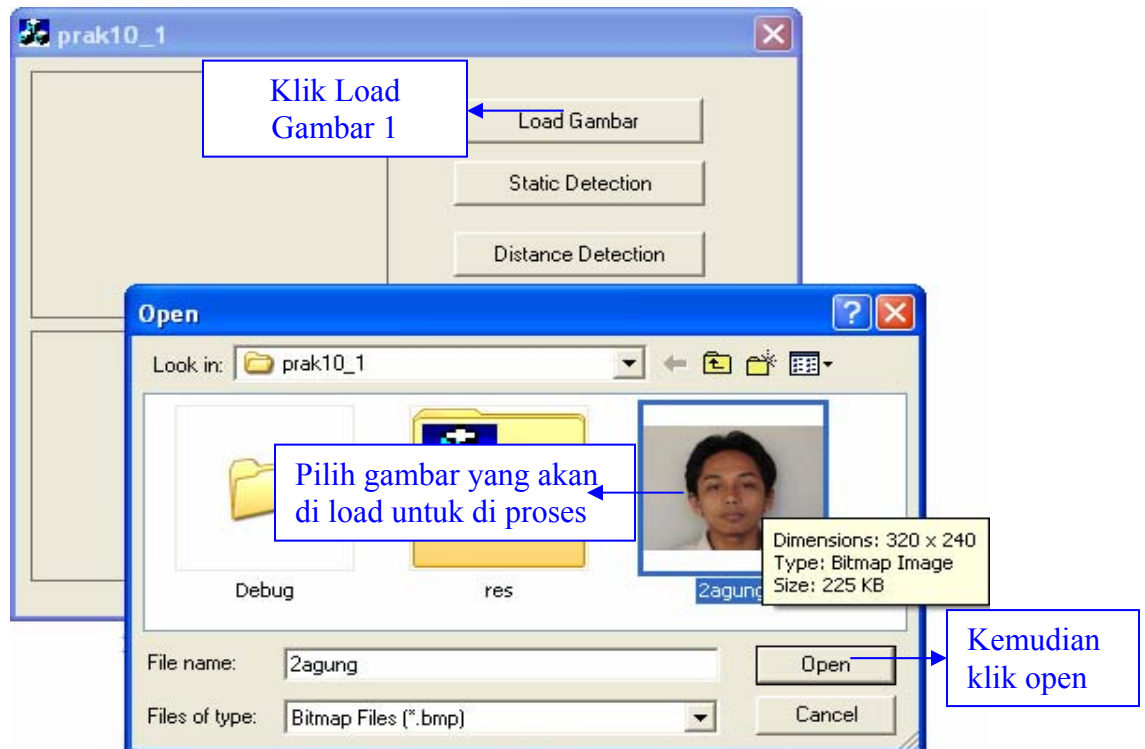
```

    CString name;

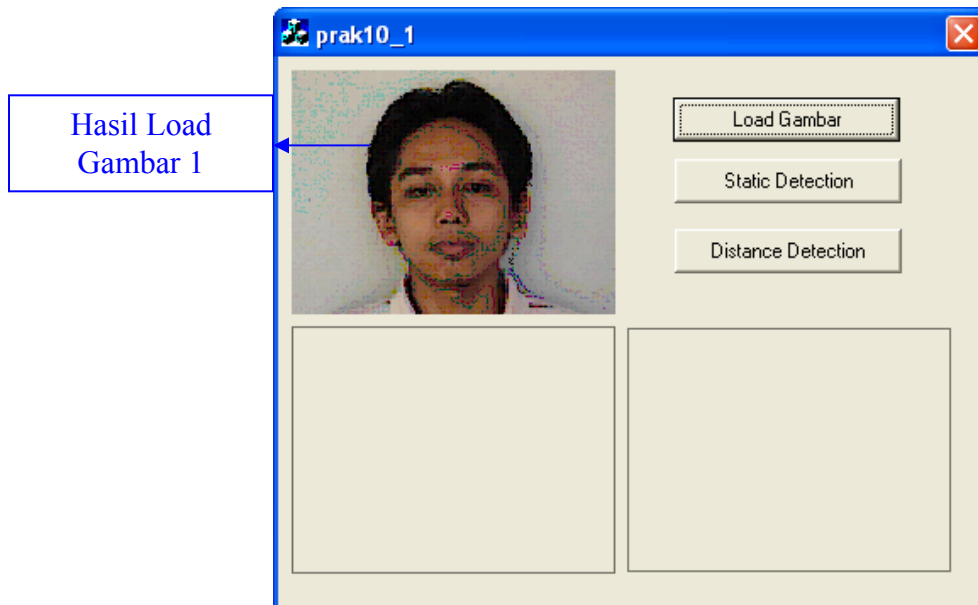
```

Variable temp untuk gambar

Saat Load Gambar 1 di klik, maka hasilnya akan seperti berikut ini :



Setelah di klik open :



- f. Tambahkan program pada saat button2 click seperti di bawah ini, akan melakukan static detection, yaitu deteksi warna secara statis :

```
void CPrak10_1Dlg::OnButton2()
{
    // TODO: Add your control notification handler code here
    int i,j;
    int red,green,blue,gray;
    long int warna, warnal;

    CDC* pDC = m_pic2.GetDC();// mengarah kontrol picture
    CDC dcMem; //mengarahkan obyek yg sdh dipilih pd bitmap
    CRect rect;//kotak di picture
    BITMAP bm;//mendefinisikan fungsi2 yg ada d bitmap

    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);
    }
    m_pic2.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++)
        {
            warna=dcMem.GetPixel(j,i);
            WarnaToRGB(warna,&red,&green,&blue);
            // 102-215, 20-158, 0-179 trafic merah
            //102-160, 70-100, 0-65fitur warna kulit

            if(((red>102)&&(red<160))&&((green>70)&&(green<100))&&
            (blue>0)&&(blue>65))
                warnal=warna;
            else
                warnal=RGBToWarna(0,0,0);

            dcMem.SetPixel(j,i,warnal);
        }
}
```

Fungsi untuk deteksi fitur warna kulit 102-160, 70-100, 0-65

```

pDC->StretchBlt(0,0,rect.Width(),rect.Height(),&dcMem,
0,0,bm.bmWidth,bm.bmHeight,SRCCOPY);
}

```

- g. Tambahkan fungsi untuk merubah data pixel ke RGB, dan fungsi untuk merubah dari RGB ke data pixel

```

// 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;
}

```

Fungsi untuk
membaca
warna RGB

```

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

```

Fungsi untuk
membuat
warna RGB

- h. Tambahkan program pada saat button3 click seperti di bawah ini, akan melakukan distance detection, yaitu deteksi warna berdasarkan pada jarak :

```

void CPrak10_1Dlg::OnButton3()
{
    // TODO: Add your control notification handler code here
    //int r=120, g=70, b=90; //untuk traffic merah

    int r=144, g=89, b=65;
    int k,j,red,green,blue,gray;
    long int warna, warnagray;
    double d;
    CDC* pDC = m_pic3.GetDC();// mengarah kontrol picture
    CDC dcMem; //mengarahkan obyek yg sdh dipilih pd bitmap
    CRect rect;//kotak di picture
    BITMAP bm;//mendefinisikan fungsi2 yg ada d bitmap

    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);
    }
    m_pic3.GetClientRect(rect);//
    m_bmpBitmap.GetBitmap(&bm);//

    dcMem.CreateCompatibleDC(pDC);
    dcMem.SelectObject(&m_bmpBitmap);

```

Inisialisai warna RGB yang
digunakan sebagai deteksi jarak

```

for(j=0;j<bm.bmHeight;j++)
    for(k=0;k<bm.bmWidth;k++)
    {
        warna=dcMem.GetPixel(k,j);
    }

```

Fungsi untuk deteksi fitur
warna kulit 102-160, 70-
100, 0-65

```

WarnaToRGB(warna, &red, &green, &blue);

//d=fabs(red-147)+fabs(green-44)+fabs(blue-48);

d=fabs(red-r)+fabs(green-g)+fabs(blue-b);
if(d<85)
    warna=RGBToWarna(red, green, blue);
else
    warna=RGBToWarna(0, 0, 0);

dcMem.SetPixel(k, j, warna);
}
pDC->StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem,
0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY);
}

```

Fungsi pemabnding untuk rata – rata area yang akan di deteksi dengan perhitungan jarak deteksi yang telah diinisialisasi warnanya

Hasil perhitungan Nilai rata – rata jarak deteksi warna

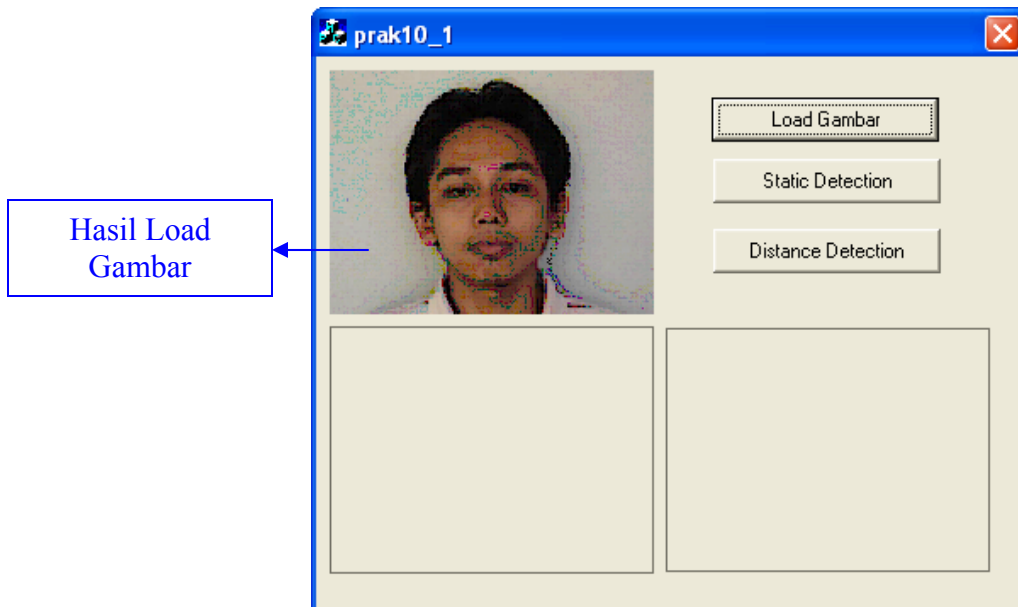
i. Cara menjalankan program

- Pilih menu : Build->Execute (!)
- Klik button Load Gambar : melakukan load gambar
- Klik button Static Detection : melakukan proses deteksi warna secara statis pada suatu image
- Klik button Transparan : melakukan proses deteksi warna berdasarkan perhitungan jarak pada suatu image
- Hasil running project prak10_1 seperti gambar di bawah ini :

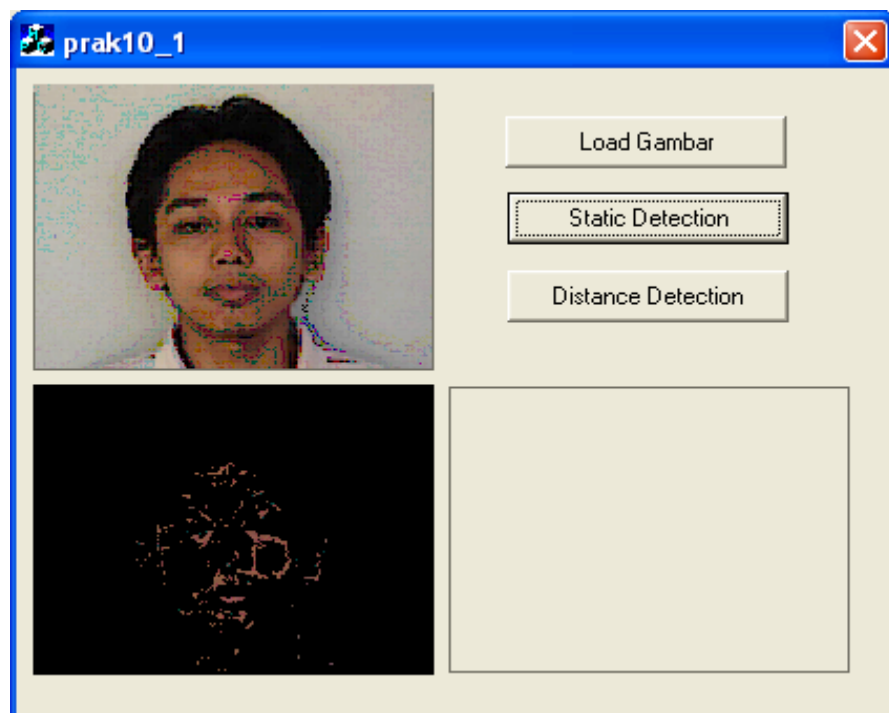
Hasil running, saat button Load Gambar di klik :



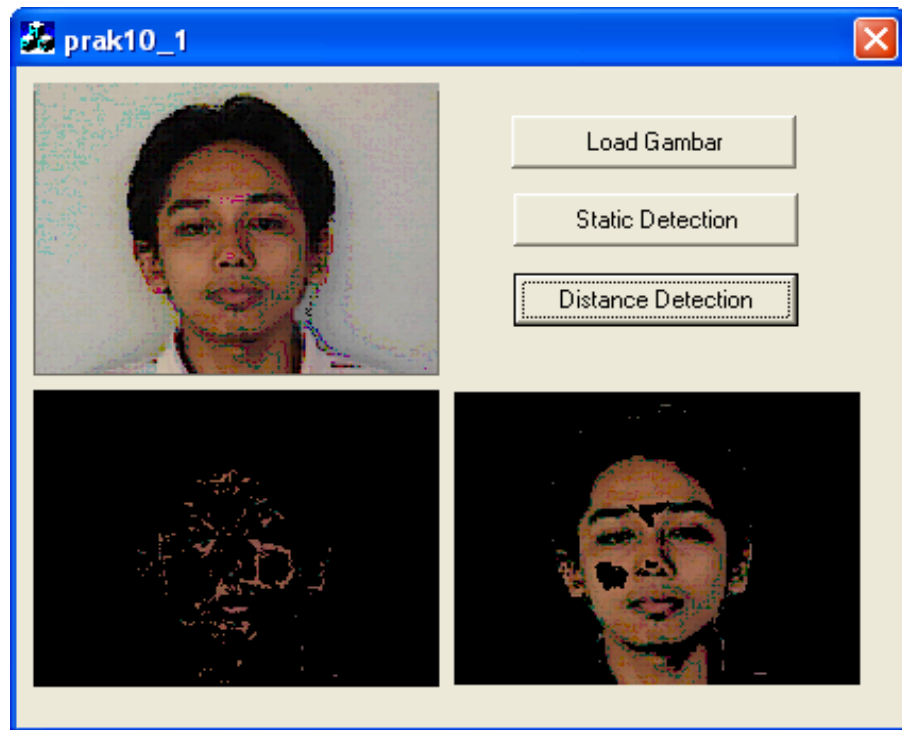
Klik open :



Hasil running, saat button Static Detection di klik :



Hasil running, saat button Distance Detection di klik :



IV. CONTOH OUTPUT DETEKSI WARNA

a. Objek Gambar : Boneka

Untuk Static Detection : R : 192 – 222, G : 183 – 220, B : 120 – 181

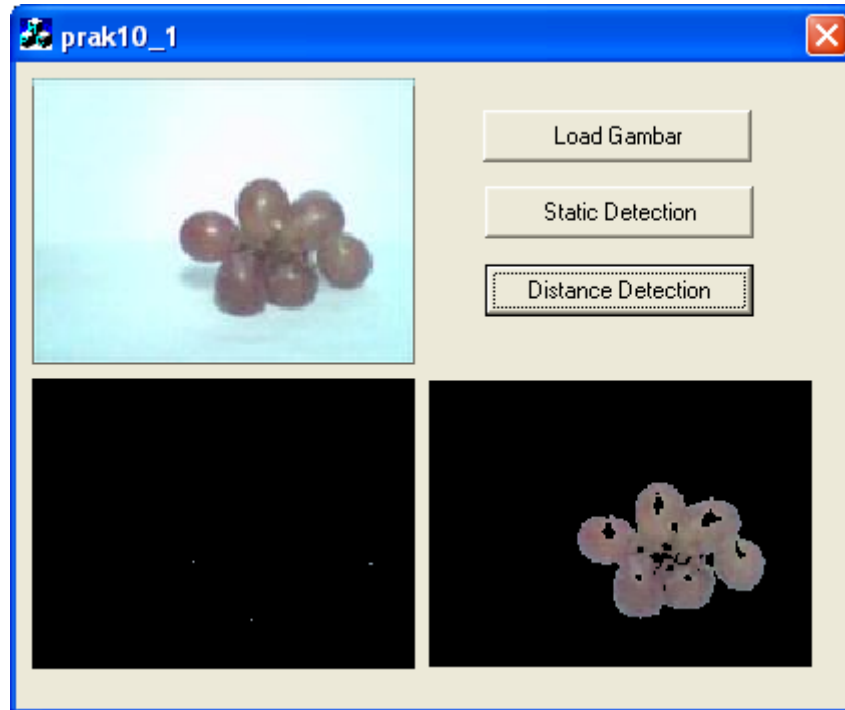
Untuk Distance Detection : R : 193, G : 192 , B : 154;



b. Objek Gambar : Buah

Untuk Static Detection : R : 192 – 222, G : 183 – 220, B : 120 – 181

Untuk Distance Detection : R : 193, G : 192 , B : 154;



V. KESIMPULAN

1. Buatlah kesimpulan dengan mengamati perbedaan hasil yang didapatkan pada static detection dan distance detection pada gambar kulit, buah-buahan, dan pemandangan