

Perbaikan Citra (Enhancement 1)

1. Tujuan:

1. Mahasiswa memahami konsep brightness, kontras dan autolevel
2. Mahasiswa dapat membuat program untuk memperjelas citra (brightness) dan mengatur kontras.
3. Mahasiswa dapat membuat program autolevel pada citra gray scale

2. Dasar Teori:

Brightness

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

$$x_b = x + b$$

dimana

x = adalah nilai derajat keabuan

b = nilai penambah

x_b = hasil brightness

Mengubah Kontras

Mengubah kontras dari suatu citra adalah proses pengaturan nilai range interval pada setiap nilai derajat keabuan, dan didefinisikan dengan :

$$x_k = k x$$

dimana

x = nilai derajat keabuan

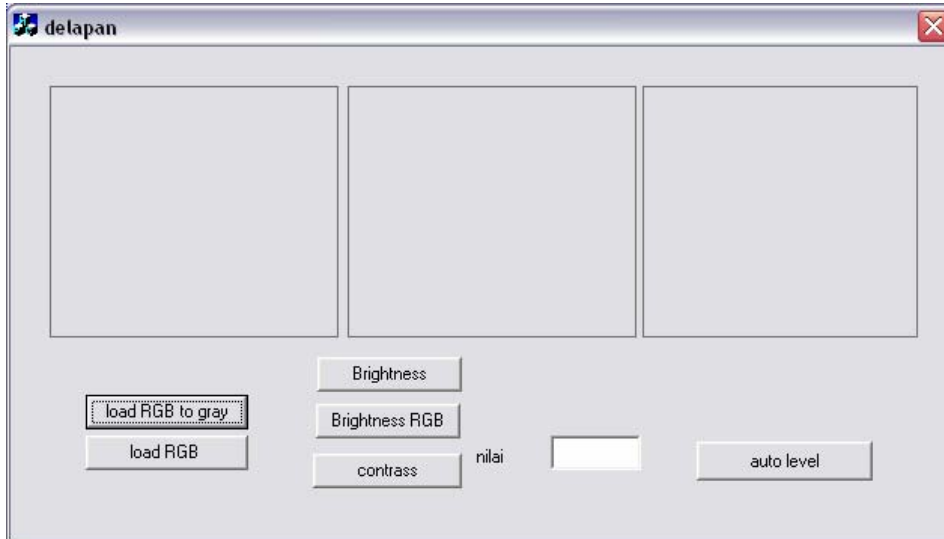
k = nilai kontras

x_k = nilai setelah pengaturan kontras

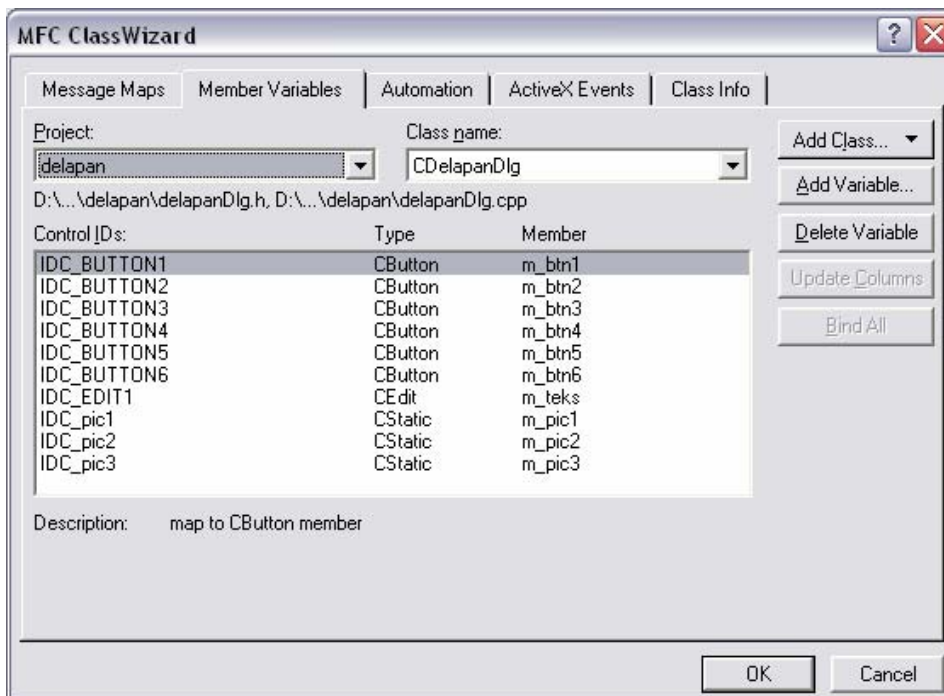
3. Percobaan:

Brighness (Memperjelas Gambar), Contrass, Auto Level

Buat projet dengan desain sebagai berikut:



member variabel:



Program di button load RGB to gray untuk menampilkan gambar yang akan diproses dalam format warna grayscale

```
int i, j, red, green, blue, gray;
long int warna, warnagray;
CDC* pDC = m_pic1.GetDC();
CDC dcMem1;
CRect rect;
BITMAP bm;
HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
"satu.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);
for(i=0; i<bm.bmHeight; i++)
    for(j=0; j<bm.bmWidth; j++)
    {
        warna=dcMem1.GetPixel(j, i);
        WarnaToRGB(warna, &red, &green, &blue);
        gray=int(red+green+blue)/3;
        warnagray=RGBToWarna(gray, gray, gray);
        dcMem1.SetPixel(j, i, warnagray);
    }
pDC->StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem1,
0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY);
```

Program di button load RGB untuk menampilkan gambar yang akan diproses dalam format warna RGB

```
CDC* pDC = m_pic1.GetDC();
CDC dcMem1;
CRect rect;
BITMAP bm;
HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
"satu.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);
```

Program di button brightness untuk mengatur brightness dari gambar grayscale sebesar nilai yang dimasukkan ke textbox 1

```
int i, j, red, green, blue, gray;
int k=GetDlgItemInt(IDC_EDIT1);
long int warna, warnagray;
CDC* pDC = m_pic2.GetDC();
CDC dcMem1;
CRect rect;
BITMAP bm;
HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
"satu.bmp", 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);
dcMem1.CreateCompatibleDC(pDC);
dcMem1.SelectObject(&m_bmpBitmap);
for(i=0; i<bm.bmHeight; i++)
    for(j=0; j<bm.bmWidth; j++)
    {
        warna=dcMem1.GetPixel(j, i);
        WarnaToRGB(warna, &red, &green, &blue);
        gray=int(red+green+blue)/3;
        gray=gray+k;
        if(gray>255)gray=255;
        if(gray<0)gray=0;
        warnagray=RGBToWarna(gray, gray, gray);
        dcMem1.SetPixel(j, i, warnagray);
    }
pDC->StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem1,
0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY);
```

Program di button contrast untuk mengatur contrast dari gambar grayscale sebesar nilai yang dimasukkan di textbox

```
int i, j, red, green, blue, gray;
float k=float(GetDlgItemInt(IDC_EDIT1))/10;
long int warna, warnagray;
CDC* pDC = m_pic3.GetDC();
CDC dcMem1;
CRect rect;
BITMAP bm;
HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
"satu.bmp", 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);
dcMem1.CreateCompatibleDC(pDC);
dcMem1.SelectObject(&m_bmpBitmap);
for(i=0; i<bm.bmHeight; i++)
    for(j=0; j<bm.bmWidth; j++)
    {
        warna=dcMem1.GetPixel(j, i);
        WarnaToRGB(warna, &red, &green, &blue);
        gray=int(red+green+blue)/3;
        gray=int(gray*k);
        if(gray>255)gray=255;
        warnagray=RGBToWarna(gray, gray, gray);
        dcMem1.SetPixel(j, i, warnagray);
    }
pDC->StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem1,
0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); //
```

Program di button brightness RGB untuk mengatur brightness dari gambar RGB sebesar nilai yang dimasukkan ke textbox 1

```
int i, j, red, green, blue;
int k=GetDlgItemInt(IDC_EDIT1);
long int warna, warna2;
CDC* pDC = m_pic2.GetDC();
CDC dcMem1;
CRect rect;
BITMAP bm;
HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
"satu.bmp", 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);
dcMem1.CreateCompatibleDC(pDC);
dcMem1.SelectObject(&m_bmpBitmap);
for(i=0; i<bm.bmHeight; i++)
    for(j=0; j<bm.bmWidth; j++)
    {
        warna=dcMem1.GetPixel(j, i);
        WarnaToRGB(warna, &red, &green, &blue);
        red=red+k;
        if(red>255)red=255;
        if(red<0)red=0;
        green=green+k;
        if(green>255)green=255;
        if(green<0)green=0;
        blue=blue+k;
        if(blue>255)blue=255;
        if(blue<0)blue=0;
        warna2=RGBToWarna(red, green, blue);
        dcMem1.SetPixel(j, i, warna2);
    }
pDC->StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem1,
0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); //
```

Program di button autolevel, mengatur contrast dan brightness secara otomatis

```
int i, j, red, green, blue, gray;
long int warna;
CDC* pDC = m_pic2.GetDC();
CDC dcMem1;
CRect rect;
BITMAP bm;
HBITMAP hBitmap=(HBITMAP)::LoadImage(AfxGetInstanceHandle(),
"satu.bmp", 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);
dcMem1.CreateCompatibleDC(pDC);
dcMem1.SelectObject(&m_bmpBitmap);
int xmax=300,xmin=0,d;
for(i=0;i<bm.bmHeight;i++)
    for(j=0;j<bm.bmWidth;j++)
    {
        warna=dcMem1.GetPixel(j,i);
        WarnaToRGB(warna,&red,&green,&blue);
        gray=int(red+green+blue)/3;
        if(gray<xmax)xmax=gray;
        if(gray>xmin)xmin=gray;
    }

d=xmin-xmax;
for(i=0;i<bm.bmHeight;i++)
    for(j=0;j<bm.bmWidth;j++)
    {
        warna=dcMem1.GetPixel(j,i);
        WarnaToRGB(warna,&red,&green,&blue);
        gray=int(red+green+blue)/3;
        gray=int(float(255/d)*(gray-xmax));
        warna=RGBToWarna(gray,gray,gray);
        dcMem1.SetPixel(j,i,warna);
    }
pDC->StretchBlt(0,0,rect.Width(),rect.Height(),&dcMem1,
0,0,bm.bmWidth,bm.bmHeight,SRCCOPY);
```

Fungsi mengubah warna 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 mengubah rgb ke warna

```
long int RGBToWarna(int Red, int Green, int Blue)
{
    return(Red+(Green<<8)+(Blue<<16));
}
```

Tambahan pada header file

```
public:
    Cbitmap m_bmpBitmap;
```

Penjelasan program:

Brightness untuk gambar grayscale

```
int k=GetDlgItemInt(IDC_EDIT1);
for(i=0;i<bm.bmHeight;i++)
    for(j=0;j<bm.bmWidth;j++)
    {
        warna=dcMem1.GetPixel(j,i);
        WarnaToRGB(warna,&red,&green,&blue);
        gray=int(red+green+blue)/3;
        gray=gray+k;
        if(gray>255)gray=255;
        if(gray<0)gray=0;
        warnagray=RGBToWarna(gray,gray,gray);
        dcMem1.SetPixel(j,i,warnagray);
    }
```

Nilai warna pada setiap pixel akan diambil rata-rata red, green dan bluenya kemudian nilai rata-rata itu akan ditambahkan dengan nilai brightness yang didapat dari textbox, nilai warna akan dibatasi dari 0 sampai dengan 255, kemudian semua titik akan diberikan nilai warna baru tersebut

Brightness untuk gambar RGB

```
int k=GetDlgItemInt(IDC_EDIT1);
for(i=0;i<bm.bmHeight;i++)
    for(j=0;j<bm.bmWidth;j++)
    {
        warna=dcMem1.GetPixel(j,i);
        WarnaToRGB(warna,&red,&green,&blue);
        red=red+k;
        if(red>255)red=255;
        if(red<0)red=0;
        green=green+k;
        if(green>255)green=255;
        if(green<0)green=0;
        blue=blue+k;
        if(blue>255)blue=255;
        if(blue<0)blue=0;
        warna2=RGBToWarna(red,green,blue);
        dcMem1.SetPixel(j,i,warna2);
    }
```

Untuk gambar RGB penambahan nilai brightness tidak dilakukan pada nilai rata-rata red, green dan bluenya melainkan penambahan nilai dilakukan di setiap nilai red, green dan bluenya. Batas nilai dari 0 sampai dengan 255 dan warna baru ini yang akan digunakan untuk nilai warna baru

Contrass untuk gambar grayscale

```
int k=GetDlgItemInt(IDC_EDIT1);
for(i=0;i<bm.bmHeight;i++)
    for(j=0;j<bm.bmWidth;j++)
    {
        warna=dcMem1.GetPixel(j,i);
        WarnaToRGB(warna,&red,&green,&blue);
        gray=int(red+green+blue)/3;
        gray=int(gray*k);
        if(gray>255)gray=255;
        warnagray=RGBToWarna(gray,gray,gray);
        dcMem1.SetPixel(j,i,warnagray);
    }
```

Sama dengan proses brightness pada gambar RGB tetapi nilai k tidak ditambahkan dengan rata-rata nilai red, green bluenya melainkan dikalikan

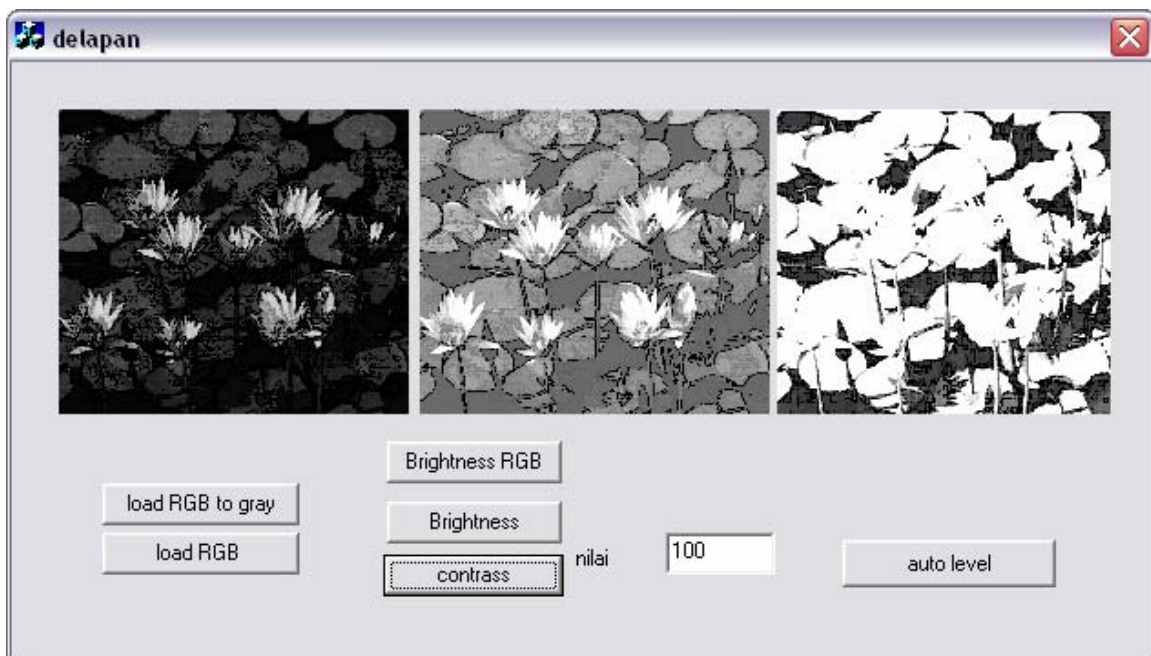
Autolevel

```
int xmax=300,xmin=0,d;  
for(i=0;i<bm.bmHeight;i++)  
    for(j=0;j<bm.bmWidth;j++)  
    {  
        warna=dcMem1.GetPixel(j,i);  
        WarnaToRGB(warna,&red,&green,&blue);  
        gray=int(red+green+blue)/3;  
        if(gray<xmax)xmax=gray;  
        if(gray>xmin)xmin=gray;  
    }  
  
d=xmin-xmax;  
for(i=0;i<bm.bmHeight;i++)  
    for(j=0;j<bm.bmWidth;j++)  
    {  
        warna=dcMem1.GetPixel(j,i);  
        WarnaToRGB(warna,&red,&green,&blue);  
        gray=int(red+green+blue)/3;  
        gray=int(float(255/d)*(gray-xmax));  
        warna=RGBToWarna(gray,gray,gray);  
        dcMem1.SetPixel(j,i,warna);  
    }
```

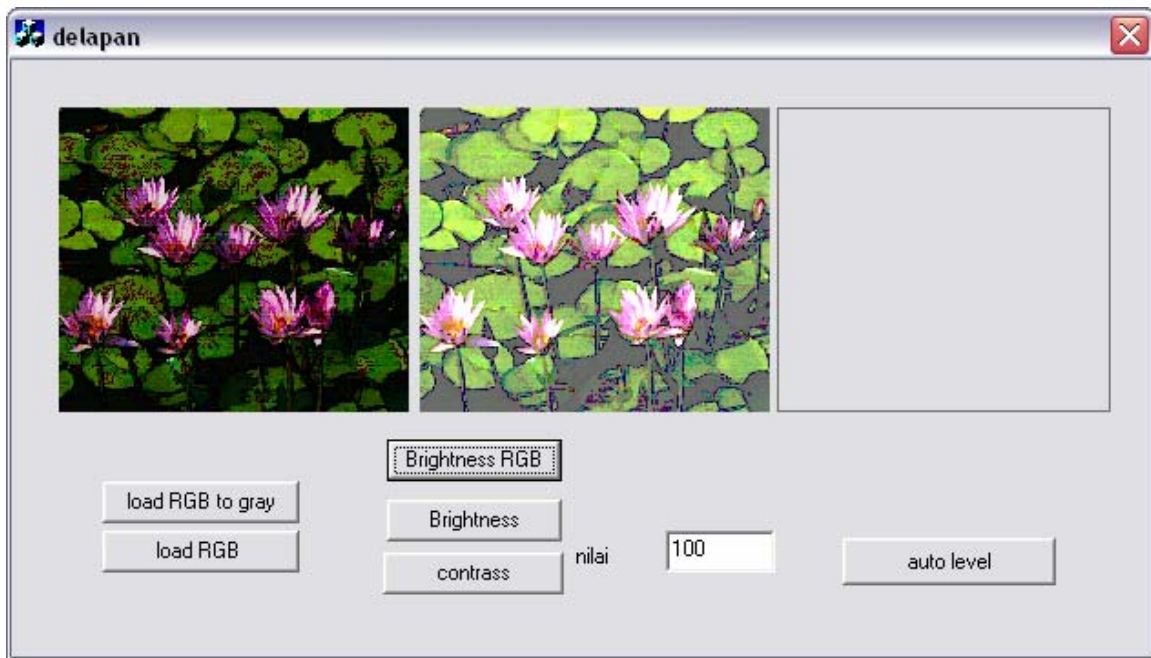
Auto level akan mengatur brightness dan contrast gambar secara otomatis, dengan cara mencari nilai terbesar dan terkecil rata-rata red green blue tiap-tiap pixel, setelah itu akan dicari jarak antara nilai terkecil dan terbesar yang akan dijadikan sebagai nilai untuk memperbaiki citra, dalam hal ini citra akan diubah ke format grayscale.

Menjalankan program

Mengatur brightness dan contrast gambar grayscale



Mengatur brighthness dan contrass gambar RGB

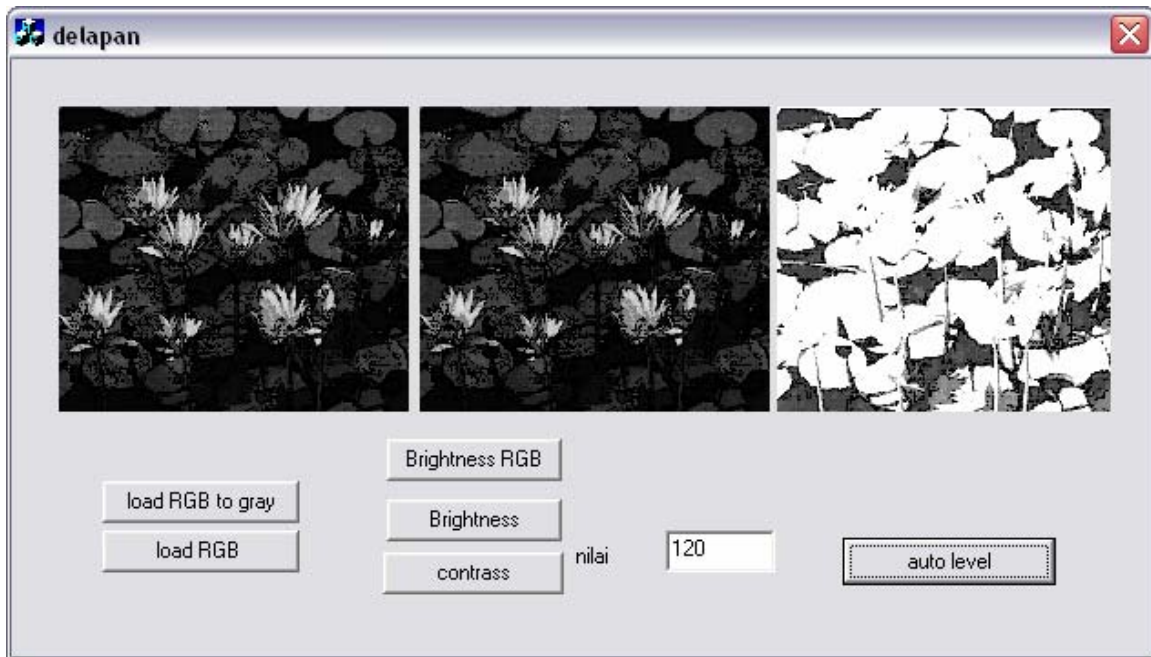


Mengatur gambar dengan auto level

Misal sebelumnya telah diatur nilai brightness dan contrassnya 120 secara manual dengan memasukkan nilai di textbox, dan hasilnya sebagai berikut



Ketika di auto level akan menjadi gambar di tengah



4. Latihan:

1. Ujilah program brightness yang sudah dibuat dengan gambar yang cenderung gelap pada penambahan 4 macam nilai konstanta, bagaimana hasilnya berikan kesimpulan. (tampilkan nilai K dan gambar hasilnya)
2. Ujilah program kontras yang sudah dibuat dengan gambar yang cenderung kabur pada pengaturan 4 macam nilai konstanta, bagaimana hasilnya berikan kesimpulan. (tampilkan nilai K dan gambar hasilnya)
3. Ujilah program autolevel yang sudah dibuat dengan gambar yang cenderung gelap dan kabur, bagaimana hasilnya berikan kesimpulan. (gunakan gambar yang low contrast)

5. Laporan Resmi:

Buatlah laporan resmi dari latihan-latihan diatas dengan menampilkan source code pada prosedur yang dibuat dan deklarasi lain yang diperlukan, dan sertakan penjelasan analisa dan kesimpulannya.