

# Mengenal Lebih Jauh Apa Itu Point Process

**Faisal Ridwan**

FaizalLeader99@yahoo.com

***Lisensi Dokumen:***

*Copyright © 2003-2007 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

## Pendahuluan

Perkembangan dunia teknologi dan komunikasi sekarang ini telah menuntut semua pekerjaan dilakukan dengan cepat dan efisien. Agar semua dapat berjalan dengan mudah maka diperlukannya suatu alat atau fasilitas yang dapat memberi kemudahan tersebut. Salah satunya adalah komputer, Dengan komputer hampir semua pekerjaan dapat dilakukan, tentu saja semua pekerjaan yang berhubungan dengan informasi, komunikasi, citra digital, dsb.

Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra yang banyak melibatkan persepsi visual dengan bantuan komputer . Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin.

Artikel ini membahas tentang pengolahan citra digital yaitu point proses untuk mengubah nilai pixel yang asli menjadi nilai pixel yang baru. Pada kesempatan ini penulis akan menjelaskan apa itu point proses dan mengaplikasikan Pengolahan Citra Digital ini ke dalam Bahasa Pemrograman C# menggunakan Microsoft Visual Studio.

## Point Processing

Point process merupakan salah satu teknik untuk melakukan pengolahan citra digital dengan cara mengubah nilai pixel yang asli berdasarkan nilai pixel tersebut.

Yang termasuk pengolahan citra digital dengan teknik point process adalah:

- a. Brightness
- b. Contrast Stretching
- c. Gray Scale
- d. Negasi
- e. Black & White

### 1. Pencerahan citra (image brightening)

Pada dasarnya merubah nilai warna dari gelap menuju terang atau sebaliknya merubah citra yang terlalu cemerlang/pucat menjadi gelap. Nilai brightness satu pixel dalam suatu citra, yang menunjukkan tingkat kecerahannya dari hitam sampai putih. Tingkat kecerahan biasanya dinilai dari 0 (hitam) hingga 255 (putih). Sesuai kutipan dari situs *Kamus Komputer dan Teknologi Informasi* yang mendefinisikan Brightness sebagai berikut [1]:

*“Brightness Atau Kecerahan. Adalah nilai satu pixel dalam suatu citra, tingkat kecerahannya dari hitam sampai putih. Tingkat kecerahan biasanya dinilai dari 0 (hitam) hingga 255 (putih).”*

Dengan pertolongan GST fungsi, dapat ditarik formula linier :

$$L_o = L_i + T \text{ atau } f(x,y)' = f(x,y) + T$$

Dimana L = Nilai derajat keabuan

T = Nilai penambahan

L<sub>o</sub> = Hasil Brightness

Dimana C adalah suatu konstanta yang bernilai positif untuk meningkatkan kecermerlangan citra, bernilai negatif untuk mengurangi kecermerlangan citra.

Untuk citra true color :

$$R_o = R_i + CR$$

$$G_o = G_i + CG$$

$$B_o = B_i + CB$$

Algoritma pencerahan citra (brightness) adalah sebagai berikut :

```
private void brightnessToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Deklarasi variabel lokal
    int i, j;
    if (img != null)
    {
        img2 = new Bitmap(img);
        //Struktur bersarang untuk meload semua pixel yang ada pada citra awal
        for (i = 0; i <= img2.Width - 1; i++)
        {
            for (j = 0; j <= img2.Height - 1; j++)
            {
                //Mendapatkan data pixel pada area form
                Color pixelColor = img2.GetPixel(i, j);
                nRed = pixelColor.R + 100;
                nGreen = pixelColor.G + 100;
                nBlue = pixelColor.B + 100;
                if (nRed > 255)
                    nRed = 255;
                if (nGreen > 255)
                    nGreen = 255;
                if (nBlue > 255)
                    nBlue = 255;
                //Deklarasi pixel baru dari data pixel yang telah diseleksi
                Color newpixelColor = Color.FromArgb(nRed, nGreen, nBlue);
                img2.SetPixel(i, j, newpixelColor);
            }
        }
        //Menyimpan citra baru pada pictureBox2
        pictureBox2.Image = img2;
    }
}
```

Hasil operasi pencerahan citra pada citra adalah sebagai berikut :



Gambar 1. Citra awal



Gambar 2. Hasil citra brightness

## 2. Contrast Stretching

Contrast stretching ini adalah teknik yang digunakan untuk mendapatkan citra baru dengan kontras lebih baik daripada kontras dari citra asalnya. Proses contrast stretching termasuk proses perbaikan citra yang bersifat point processing, yang artinya proses ini hanya tergantung dari nilai intensitas (gray level) satu pixel, tidak tergantung dari pixel lain yang ada di sekitarnya. Sesuai kutipan dari *Setyo Nugroho* yang mendefinisikan contrast stretching sebagai berikut [2];

*“Contrast stretching ini adalah teknik yang digunakan untuk mendapatkan citra baru dengan kontras yang lebih baik daripada kontras dari citra asalnya. Proses contrast stretching termasuk proses perbaikan citra yang bersifat point processing, yang artinya proses ini hanya tergantung dari nilai intensitas (gray level) satu pixel, tidak tergantung dari pixel lain yang ada di sekitarnya. Disini diasumsikan bahwa citra memiliki range gray level dari 0 sampai 255.”*

Jika sebuah citra yang mempunyai nilai keabuan yang tidak terlalu berbeda untuk semua titik, dimana titik tergelap dalam citra tidak mencapai hitam pekat dan titik paling terang dalam citra tidak berwarna putih cemerlang. Dengan

peningkatan kontras maka titik yang cenderung gelap menjadi lebih gelap dan yang cenderung terang menjadi lebih cemerlang.

Peningkatan kontras dapat dilakukan dengan bermacam rumus, salah satunya adalah :

$$L_o = G (L_i - Q) + Q$$

Dimana : G = Koefisien penguatan kontras

Q = Nilai skala keabuan yang dipakai sebagai pusat pengontrasan

Algoritma contrast stretching adalah sebagai berikut :

```
private void colorStretchingToolStripMenuItem_Click(object sender, EventArgs e)
{
    int c1, d1, c2, d2;
    c1 = 10;
    d1 = 100;
    c2 = 10;
    d2 = 100;
    brightnessToolStripMenuItem.Checked = false;
    grayToolStripMenuItem.Checked = false;
    invertToolStripMenuItem.Checked = false;
    thresholdToolStripMenuItem.Checked = false;
    grayToolStripMenuItem.Checked = false;
    blackAndWhiteToolStripMenuItem.Checked = false;
    colorStretchingToolStripMenuItem.Checked = true;
    if (img != null)
    {
        img2 = new Bitmap(img);
        for (i = 0; i <= img2.Width - 1; i++)
        {
            for (j = 0; j <= img2.Height - 1; j++)
            {
                Color pixelColor = img2.GetPixel(i, j);
                nRed = pixelColor.R;
                if (nRed < c1)
                {
                    nRed = nRed * d1 / c1;
                }
                else
                {
                    if (nRed < c2)
                    {

```

```
        nRed = d1 + (nRed - c1) * (d2 - d1) / (c2 - c1);  
    else  
        nRed = d2 + (nRed - c2) * (255 - d2) / (255 - c2);  
  
    nGreen = pixelColor.G;  
    if (nGreen < c1)  
        nGreen = nGreen * d1 / c1;  
    else  
        if (nGreen < c2)  
            nGreen = d1 + (nGreen - c1) * (d2 - d1) / (c2 - c1);  
        else  
            nGreen = d2 + (nGreen - c2) * (255 - d2) / (255 - c2);  
    nBlue = pixelColor.B;  
    if (nBlue < c1)  
        nBlue = nBlue * d1 / c1;  
    else  
        if (nBlue < c2)  
            nBlue = d1 + (nBlue - c1) * (d2 - d1) / (c2 - c1);  
        else  
            nBlue = d2 + (nBlue - c2) * (255 - d2) / (255 - c2);  
    Color newPixelColor = Color.FromArgb(nRed, nGreen, nBlue);  
    img2.SetPixel(i, j, newPixelColor);  
} }  
pictureBox2.Image = img2;  
}
```

Hasil operasi contrast stretching pada citra adalah sebagai berikut :



Gambar 3. Citra awal



Gambar 4. Hasil citra contrast stretching

### 3. Gray Scale

Grayscale adalah konversi citra true color menjadi citra keabuan. Suatu istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam dan putih. Grayscale menunjukkan jumlah warna (dari abu-abu, hingga hitam - putih) yang ada dalam satu citra. Sesuai kutipan dari situs *Kamus Komputer dan Teknologi Informasi* yang mendefinisikan grayscale sebagai berikut [3];

*“Grayscale berarti Skala Keabu-abuan. Suatu istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam dan putih. Grayscale menunjukkan jumlah warna (dari abu-abu, hingga hitam - putih) yang ada dalam satu citra. Makin besar angka grayscale, citra yang terbentuk makin mendekati kenyataan.”*

Operasi konversi citra true color ke keabuan dengan rumus :

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

Bisa juga dengan memberi bobot ( $w$ ) pada RGB karena mata manusia lebih sensitif pada warna hijau, kemudian merah, terakhir biru.

$$L_o = w_r R + w_g G + w_b B$$

Berdasarkan NTSC (National Television System Committee), dimana :

$$w_r = 0.299$$

$$w_g = 0.587$$

$$w_b = 0.144$$

Algoritma grayscale adalah sebagai berikut :

```
private void grayScaleToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Deklarasi variabel lokal
    int i, j, rata;
    if (img != null)
    {
        //Deklarasi citra baru berdasarkan citra awal
        img2 = new Bitmap(img);
        //Struktur bersarang untuk meload semua pixel yang ada pada citra awal
        for (i = 0; i <= img2.Width - 1; i++)
        {
            for (j = 0; j <= img2.Height - 1; j++)
            {
                //Mendapatkan data pixel pada area form
                Color pixelColor = img2.GetPixel(i, j);
                nRed = pixelColor.R;
                nGreen = pixelColor.G;
                nBlue = pixelColor.B;
                /*Menghitung nilai rata dari ketiga warna dengan terlebih dahulu mengubahnya menjadi bentuk integer dengan menggunakan fungsi Convert.ToInt32 */
                rata = Convert.ToInt32((nRed + nBlue + nGreen) / 3);
                //Deklarasi pixel baru dari data pixel yang data yang di proses
                Color newpixelColor = Color.FromArgb(rata, rata, rata);
                img2.SetPixel(i, j, newpixelColor);
            }
        }
        //Menyimpan citra baru di pictureBox2
        pictureBox2.Image = img2;
    }
}
```

Hasil operasi grayscale pada citra adalah sebagai berikut :



Gambar 5. Citra awal



Gambar 6. Hasil citra grayscale



#### 4. Negasi

Negasi merupakan operasi mendapatkan citra negatif (Negatif Image) adalah suatu citra seperti halnya meniru film negatif pada fotografi, yaitu titik yang berwarna putih pada citra warna hitam pada film negatifnya. Demikian juga sebaliknya dengan cara mengurangi nilai intensitas pixel dari nilai keabuan maksimum. Misalnya citra dengan 256 derajat keabuan (8 bit). Sesuai kutipan dari *Chandra Wijaya* yang mendefinisikan negasi sebagai berikut [4];

*“Negatif Image adalah suatu citra seperti halnya meniru film negatif pada fotografi dengan cara mengurangi nilai intensitas pixel dari nilai keabuan maksimum. Misal citra dengan 256 derajat keabuan (8 bit), maka citra negatif diperoleh dengan persamaan “*

*“ $X = (R+G+B) / \text{bilangan pembagi}$*

*$Y = 255 - X$ “*

maka citra negatif diperoleh dengan persamaan.

$$L_o = L_{max} - L_i$$

Misal pada citra dengan 256 derajat keabuan (8 bit)  $L_{max} = 255$  maka

$$L_o = 255 - L_i \text{ atau } f(x,y)' = 255 - f(x,y)$$

Algoritma negasi adalah sebagai berikut :

```
private void invertToolStripMenuItem_Click(object sender, EventArgs e)
{
    //deklarasi variabel lokal
    int i, j;
    //Seleksi kondisi jika ada citra
    if (img != null)
    //Deklarasi citra baru berdasarkan citra awal
    {
        img2 = new Bitmap(img);
        //Struktur bersarang untuk meload semua pixel berdasarkan citra awal
        for (i = 0; i <= img2.Width - 1; i++)
        {
            for (j = 0; j <= img2.Height - 1; j++)
            {
                //Mendapatkan data pixel pada area form
                Color pixelColor = img2.GetPixel(i, j);
                nRed = 255 - pixelColor.R;
                nGreen = 255 - pixelColor.G;
                nBlue = 255 - pixelColor.B;
                //Deklarasi pixel baru dari data pixel yang telah di proses
                Color newpixelColor = Color.FromArgb(nRed, nGreen, nBlue);
                //Menyimpan warna baru untuk citra yang baru
                img2.SetPixel(i, j, newpixelColor);
            }
        } //Menyimpan citra baru di pictureBox2
        pictureBox2.Image = img2;
    }
}
```

Hasil operasi negasi pada citra adalah sebagai berikut :



Gambar 7. Citra awal



Gambar 8. Hasil citra negasi

## 5. Tresholding

Thresholding adalah operasi non-linear yang mengubah gambar gray-scale atau citra bederajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan background dari citra secara jelas. Dimana dua tingkat ditugaskan untuk piksel yang berada di bawah atau diatas nilai ambang batas yang ditentukan. Citra hasil thresholding biasanya digunakan lebih lanjut untuk proses pengenalan obyek serta ekstraksi fitur. Sesuai kutipan dari situs *Wavemetric* yang mendefinisikan thresholding sebagai berikut [5] ;

*" Thresholding is a non-linear operation that converts a gray-scale image into a binary image where the two levels are assigned to pixels that are below or above the specified threshold value."*

Proses thresholding ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan thresholding dengan derajat keabuan dapat digunakan rumus:

$$X = b \cdot \text{int}(w/b)$$

dimana : w adalah nilai derajat keabuan sebelum thresholding  
x adalah nilai derajat keabuan setelah thresholding

$$b = \text{int}(256/a)$$

Operasi `ImageThreshold` yang juga menyediakan beberapa metode untuk menemukan "optimal" nilai ambang batas untuk gambar yang diberikan. `ImageThreshold` menyediakan metode berikut untuk menentukan nilai ambang.

## Metode Thresholding

1. Secara otomatis menghitung nilai ambang dengan menggunakan metode iteratif.
2. Perkiraan histogram dari gambar sebagai distribusi bimodal dan memilih nilai titik tengah sebagai ambang batas.
3. Evaluasi ambang berdasarkan 8 pixel terakhir di setiap baris, menggunakan baris bergantian. Perhatikan bahwa metode ini tidak didukung ketika digunakan sebagai bagian dari operasi ImageEdgeDetection.
4. Fuzzy thresholding menggunakan entropi sebagai ukuran untuk "ketidakjelasan".
5. Fuzzy thresholding menggunakan metode yang meminimalkan sebuah "ketidakjelasan" mengukur melibatkan tingkat abu-abu berarti dalam objek dan latar belakang.
6. Default metode mana Anda harus menggunakan flag / T untuk menentukan nilai ambang.

Operasi pengambangan digunakan untuk mengubah citra dengan format skala keabuan, yang mempunyai kemungkinan nilai lebih dari 2 ke citra biner yang memiliki 2 buah nilai (yaitu 0 dan 1).

### **Pengambangan Tunggal** Memiliki sebuah nilai batas ambang

Fungsi GST-nya

0, jika  $K_i < \text{ambang}$  (0 = hitam)

$K_o = 1$ , jika  $K_i \geq \text{ambang}$  (1 = putih)

atau 0, jika  $K_i \geq \text{ambang}$

$K_o = 1$ , jika  $K_i < \text{ambang}$

Algoritma thresholding adalah sebagai berikut :

```
private void thresholdToolStripMenuItem_Click(object sender, EventArgs e)
{
    int i, j;
    if (img != null) {
        img2 = new Bitmap(img);
        for (i = 0; i <= img2.Width - 1; i++) {
            for (j = 0; j <= img2.Height - 1; j++)
            {
                Color pixelColor = img2.GetPixel(i, j);
                nRed = pixelColor.R;
                nGreen = pixelColor.G;
                nBlue = pixelColor.B;
                if (nRed < 127)
                    nRed = 0;
                else nRed = 255;
                if (nGreen < 127)
                    nGreen = 0;
                else nGreen = 255;
                if (nBlue < 127)
                    nBlue = 0;
                else nBlue = 255;
                Color newpixelColor = Color.FromArgb(nRed, nGreen, nBlue);
                img2.SetPixel(i, j, newpixelColor);
            }
        }
        pictureBox2.Image = img2;
    }
}
```

Hasil operasi thresholding pada citra adalah sebagai berikut :



Gambar 9. Citra awal



Gambar 10. Hasil citra thresholding

## 6. Black & White

Black and white adalah sebuah citra pixel-pixel yang nilai intensitasnya di bawah 128 diubah menjadi hitamsedangkan pixel-pixel yang nilai intensitasnya di atas 128 diubah menjadi putih. Dalam pengolahan citra digital Black & White disebut juga dengan citra biner. Sesuai kutipan dari *Chandra Wijaya* yang mendefinisikan black & white sebagai berikut [4];

*“Black and white adalah sebuah citra pixel-pixel pada gambar yang nilai intensitasnya di bawah 128 diubah menjadi hitam (nilai intensitas = 0) jadi hanya berwarna hitam atau putih saja, sedangkan pixel-pixel yang nilai intensitasnya di atas 128 diubah menjadi putih (nilai intensitas =1). Black and white disebut juga citra biner.”*

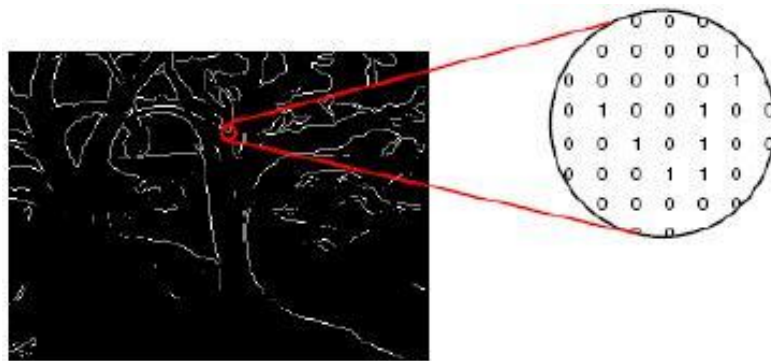
Citra biner adalah citra digital yang hanya memiliki dua nilai yang mungkin untuk setiap pixel . Biasanya dua warna yang digunakan untuk citra biner hitam dan putih meskipun setiap dua warna bisa digunakan. Warna yang digunakan untuk objek pada gambar adalah warna foreground sedangkan sisanya gambar adalah warna latar belakang.

Citra biner adalah citra yang melalui proses pemisahan piksel – piksel berdasarkan derajat keabuan yang dimiliki. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. *Pixel* dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0. Sesuai kutipan dari *Ferdinand Simanjuntak* yang mendefinisikan citra biner sebagai berikut [6] ;

*“Citra biner merupakan citra yang telah melalui proses pemisahan piksel – piksel berdasarkan derajat keabuan yang dimiliki. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. Pixel dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0”*

$$f(x, y)' = \begin{cases} a_1, f(x, y) < T \\ a_2, f(x, y) \geq T \end{cases}$$

Jika  $a_1 = 0$  dan  $a_2 = 1$ , serta  $T = \text{gray level}/2$ , maka operasi di atas mentransformasikan suatu citra menjadi citra biner.



Algoritma black & white adalah sebagai berikut :

```
private void blackAndWhiteToolStripMenuItem_Click(object sender, EventArgs e)
{
    int i, j, rata;
    if (img != null)
    {
        img2 = new Bitmap(img);
        for (i = 0; i <= img2.Width - 1; i++)
        {
            for (j = 0; j <= img2.Height - 1; j++)
            {
                Color pixelColor = img2.GetPixel(i, j);
                nRed = pixelColor.R;
                nGreen = pixelColor.G;
                nBlue = pixelColor.B;
                rata = Convert.ToInt32((nRed + nBlue + nGreen) / 3);
                if (rata > 127)
                    rata = 255;
                else
                    rata = 0;
                Color newpixelColor = Color.FromArgb(rata, rata, rata);
                img2.SetPixel(i, j, newpixelColor);
            }
        }
        pictureBox2.Image = img2;
    }
}
```

Hasil operasi black & white pada citra adalah sebagai berikut :



Gambar 9. Citra awal



Gambar 10. Hasil citra black & white



## **Kesimpulan**

Point process merupakan salah satu teknik untuk melakukan pengolahan citra digital dengan cara mengubah nilai pixel yang asli berdasarkan nilai pixel tersebut. Di dalam point proses memiliki beberapa pengolahan citra digital seperti brightness, contrast stretching, grayscale, negasi(inverst), black & white(binary image).

Brightness berarti pencerahan citra yang dasarnya merubah nilai warna dari gelap menuju terang atau sebaliknya merubah citra yang terlalu cemerlang/pucat menjadi gelap. Contrast stretching merupakan teknik yang digunakan untuk mendapatkan citra baru dengan kontras lebih baik daripada kontras dari citra asalnya. Sedangkan grayscale adalah konversi citra true color menjadi citra keabuan.

Inverst atau negasi atau sering disebut citra negatif image adalah suatu citra sepeti halnya meniru film negatif pada fotografi dengan cara mengurangi nilai intensitas pixel dari nilai keabuan maksimum. Dan yang terakhir black & white sebuah citra pixel-pixel yang nilai intensitasnya di bawah 128 diubah menjadi hitam (nilai intensitas = 0).

## Referensi

- [1]. Feri. (2004). *Brightness*. [On-line] Tersedia: <http://www.total.or.id/info.php?kk=Brightness> [Diakses 23 Februari 2011]
- [2]. Nugroho , Setyo . (2005) . *Implementasi Metoda Contrast Stretching Untuk Memperbaiki Kontras Citra*. [On-line]. Tersedia :  
<http://www.scribd.com/doc/4854230/Implementasi-Metode-Contrast-Stretching-untuk-Memperbaiki-Kontras-Citra-Setyo-Nugroho> [Diakses 23Februari 2011]
- [3]. Feri. (2004) . *Grayscale* . [On-line]. Tersedia: <http://www.total.or.id/info.php?kk=Grayscale> [Diakses 23 Februari 2011]
- [4]. Wijaya, Chandra. (2010). *Metode Pengolahan Citra*. [On-line]. Tersedia: <http://bugis.blogspot.com/2010/12/metode-pengolahan-citra.html> [Diakses 26 Februari 2011]
- [5]. (2009). *Image Threshold*. [On-line]. Tersedia:<http://www.wavemetrics.com/products/igorpro/imageprocessing/thresholding.htm> [Diakses 26 Februari 2011]
- [6]. Simanjuntak , Ferdinand. (2009). *Pengolahan Citra Digital*. [On-line] Tersedia:  
[http://www.ittelkom.ac.id/library/index.php?view=article&catid=15%3Apemrosesan-sinyal&id=573%3Apengolahan-citra-digital&option=com\\_content&Itemid=15](http://www.ittelkom.ac.id/library/index.php?view=article&catid=15%3Apemrosesan-sinyal&id=573%3Apengolahan-citra-digital&option=com_content&Itemid=15)  
[Diakses 26 Februari 2011]

## Biografi Penulis



**Faisal Ridwan.** Lahir di sebuah kota yang pernah menjadi tempat konferensi Asia-Afrika, tepatnya di Bandung, 10 Agustus 1990. Duduk di Sekolah Dasar di SDN KORPRI II. Melanjutkan pendidikan Sekolah Menengah Pertama di SLTPN 1 Baleendah. Dan menyelesaikan pendidikan Sekolah Menengah Umum di SMUN 1 Baleendah. Hingga sedang menyelesaikan program D3 jurusan Teknik Informatika di Universitas Padjadjaran tahun 2008 . Bercita-cita melanjutkan ekstensi ke Universitas Indonesia jurusan ilmu komputer. Namun masih proses penyelesaian program D3 dan sekarang sedang giat-giatnya menyusun tugas akhir.