

Pengaruh Jenis Penyisipan Bit di RGB pada Gambar

Eric Christopher / 13509037
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
Eric.c13@students.itb.ac.id

Abstract: Jenis Penyisipan RGB yang dimaksudkan disini adalah jumlah bit yang akan disisipkan dan jenis bit warna yang akan disisipkan oleh bit dari pesan.

Index: Kriptografi, Steganografi, Bit, RGB, LSB

I. LATAR BELAKANG

Pertama-tama penulis akan menjelaskan permasalahan yang akan penulis angkat di karya tulis ini. Setelah belajar tentang LSB (Least Significant Bit) dan setelah diberikan tugas besar untuk membuat sebuah program untuk mensteganografi sebuah pesan ke gambar, penulis mengira-ngira apakah metode steganografi ini dapat menjadi lebih baik dengan memanfaatkan kelemahan dari kepekaan mata manusia.

Pada karya tulis ini penulis ingin mencoba membandingkan perbedaan antara penyisipan bit pada bit LSB dari sebuah gambar dari berbagai kemungkinan. Misalnya hanya disisipkan di bit yang menyatakan warna merah, ataupun hanya di bit hijau, maupun biru. Meskipun nantinya akan memiliki *PSNR (Peak Signal-to-Noise Ratio)* yang relative sama tetapi secara kasat mata akan berbeda.

II. DASAR TEORI

Menurut Wikipedia, secara umum Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita [Bruce Schneier - Applied Cryptography]. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data [A. Menezes, P. van Oorschot and S. Vanstone - Handbook of Applied Cryptography].

Sedangkan di slide kriptografi di ITB, Kata cryptography berasal dari bahasa Yunani: *krupto* (hidden atau secret) dan *grapho* (writing) yang artinya "secret writing". Dan definisi lama dari kata kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat

dimengerti lagi maknanya.

Tujuan dari kriptografi ada empat yang juga merupakan aspek keamanan informasi yaitu :

- ➔ Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah dienkripsi.
- ➔ Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
- ➔ Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
- ➔ Non-repudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Sedangkan arti kata steganografi menurut Wikipedia adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Sebaliknya, kriptografi menyamarkan arti dari suatu pesan, tapi tidak menyembunyikan bahwa ada suatu pesan. Kata "steganografi" berasal dari bahasa Yunani *steganos*, yang artinya "tersembunyi atau terselubung", dan *graphein*, "menulis".

Di slide kriptografi, Steganography adalah ilmu dan seni menyembunyikan (embedded) informasi dengan cara menyisipkan pesan rahasia di dalam pesan lain. Sedangkan steganografi digital: steganografi pada data digital dengan menggunakan komputer digital.

Kelebihan steganografi jika dibandingkan dengan kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan. Seringkali, steganografi

dan kriptografi digunakan secara bersamaan untuk menjamin keamanan pesan rahasianya.

Meurut Wikipedia terdapat beberapa metode untuk steganografi. Pertama adalah Least Significant Bit Insertion (LSB). Metode ini dapat digunakan untuk menyembunyikan pesan pada media digital dengan cara menyisipkan bit-bit pesan pada bit paling rendah atau paling kanan. Keuntungan dari metode ini adalah cepat dan mudah.

Metode kedua adalah *Algorithms and Transformation*. Metode ini adalah metode steganografi dengan menyembunyikan data dalam fungsi matematika. Dua fungsi tersebut adalah *Discrete Cosine Transformation (DCT)* dan *Wavelet Transformation*. Fungsinya adalah mentransformasi data dari satu tempat (domain) ke tempat (domain) lain.

Metode berikutnya adalah *Redundant Pattern Encoding*. Metode ini menggambar pesan kecil pada kebanyakan gambar. Keuntungan metode ini adalah dapat bertahan dari *cropping* sehingga tidak mudah hilang. Sedangkan kerugiannya adalah tidak dapat menggambar pesan yang lebih besar.

Metode terakhir adalah *Spread Spectrum Method*. Metode ini memecah-mecah pesan secara acak tidak seperti LSB. Metode ini mudah diserang dengan penghancuran dari kompresi gambar.

III. PEMBAHASAN

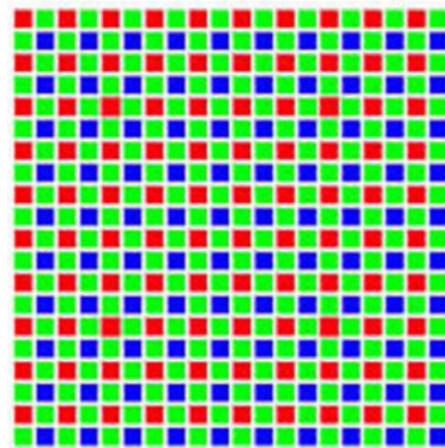
Pertama – tama penulis ingin mengingatkan bahwa pada karya tulis ini ia akan membahas tentang pengaruh jenis penyisipan pada RGB terhadap kualitas gambar. Perbandingan ini tidak dilihat secara matematis tetapi dilihat secara kasat mata manusia.

Untuk lebih mudahnya coba perhatikan gambar dibawah ini:



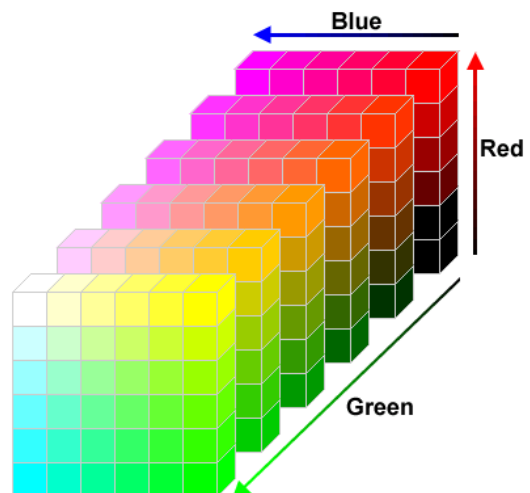
Sumber: Slide PC_Graphic_Adapter2.pptx (Slide Sistem Multimedia)

Setelah memperhatikan gambar tersebut apakah terlihat bahwa kita (dengan menggunakan mata manusia) lebih dapat membedakan perbedaan dalam warna hijau dibandingkan warna merah maupun biru. Hal tersebut terlihat dari warna hijau yang terlihat lebih panjang dari pada warna merah dan warna biru. Kelemahan pada mata manusia inilah yang juga mempengaruhi jenis sensor pada kamera sehingga sensor untuk warna hijau lebih banyak dari pada sensor untuk warna merah dan warna biru seperti pada gambar di bawah ini.



Sumber: Slide Camera_Data.pdf (slide Sistem Multimedia)

Begitu pula terdapat yang namanya web safe color yang hanya memiliki 6 jenis hex code : 00 ,33 ,66 ,99 ,CC ,FF yang membuatnya total menjadi 216 jenis warna saja (6x6x6). Warna selain kombinasi 6 jenis hex code tadi akan dipetakan sehingga menjadi kombinasi dari hex code tadi. Hal ini dapat membantu jika gambar yang sudah memiliki pesan rahasia dengan cara steganografi tadi akan diupload ataupun ditampilkan di web, sehingga metode LSB tadi akan lebih efektif dan tidak mudah dicurigai bahwa ada pesan didalam gambar tadi.



Sumber: Slide Slide PC_Graphic_Adapter2.pptx (Slide Sistem Multimedia)

Akan tetapi disini kita akan lebih fokus pada penyisipan bitnya. Maka dari itu penulis membuat sedikit program kecil untuk memudahkannya dalam mengedit gambar dan dapat membandingkan warna pada gambar.

Pertama-tama penulis membuat fungsi sederhana untuk menggenerate sebuah gambar warna blok. Disini sebagai contoh penulis menset semuanya adalah 255

```
public void initImage()
{
    height=200;
    width=400;
    bimg=new BufferedImage(width, height,
Image.SCALE_DEFAULT);
    int a,r,g,b;
    a=255;
    r=255;
    g=0;
    b=0;
    for(int x=0;x<width;x++)
    {
        for(int y=0;y<height;y++)
        {
            bimg.setRGB(x,y, ((a<<24)+(r<<16)+
(g<<8)+b));
        }
    }
}
```

Berikut adalah fungsi untuk mengubah bit dari rgb (sudah ditentukan mana yang harus diubah). Kembaliannya adalah int bit yang sudah diubah dengan bit terakhirnya sama dengan parameter string bit.

```
public int ubah(int rgb, String bit)
//mengembalikan ubahan rgb dengan bit sebagai bit
terakhirnya
{
    int hsl=0;
    String temp="";
    temp=Integer.toBinaryString(rgb);
    //potong
    temp=setbit(temp,8);
    temp=temp.substring(0, temp.length()-bit.length());
    temp+=bit;
    hsl=bitToInt(temp);
    return hsl;
}
```

Kemudian berikut adalah fungsi untuk memasukkan mode.

```
public void edit(int x, int y, String bit, String mode)
//1 r
//2 g
//3 b
//4 a
```

```
//gabungan
{
    int pixel,r,g,b,a;
    pixel=bimg.getRGB(x, y);
    b=pixel >> 0 & 0xFF;
    g=pixel >> 8 & 0xFF;
    r=pixel >> 16 & 0xFF;
    a=pixel >> 24 & 0xFF;

    //pengecekan
    if(mode.charAt(0)=='1')
    {
        r=ubah(r, bit);
    }
    else if(mode.charAt(0)=='2')
    {
        g=ubah(g, bit);
    }
    else if(mode.charAt(0)=='3')
    {
        b=ubah(b, bit);
    }
    else
    {
        a=ubah(a, bit);
    }
    //set
    bimg.setRGB(x,y, ((a<<24)+(b<<16)+(g<<8)+r));
}
```

Maka dengan memasukkan mode 1 maka akan disisipkan ke bit r, 2 untuk g, dan 3 untuk b.

Kemudian agar kita dapat mengamati perubahan warnanya maka penulis membuat beberapa fungsi lagi ,yang pertama adalah fungsi untuk mengubah n bit dibelakang menjadi inversnya (kemungkinan paling buruk adalah seluruh bit berubah).

```
public int inversbit(int rgb, int n)
{
    int hsl=0;
    String temp="";
    temp=Integer.toBinaryString(rgb);
    //potong
    temp=setbit(temp,8);

    int idx=7;
    for(int i=0;i<n;i++)
    {
        if(temp.charAt(idx)=='1')
        {
            temp=temp.substring(0, idx)
+'0'+temp.substring(idx+1);
        }
        else
        {
            temp=temp.substring(0, idx)
```

```

+'1'+temp.substring(idx+1);
    }
    idx--;
}
hsl=bitToInt(temp);
return hsl;
}

```

Kemudian prosedur untuk memilih bit mana yang akan di invers.

```

public void invers(int x, int y, int n, String mode)
{
    int pixel,r,g,b,a;
    pixel=bimg.getRGB(x, y);
    b=pixel >> 0 & 0xFF;
    g=pixel >> 8 & 0xFF;
    r=pixel >> 16 & 0xFF;
    a=pixel >> 24 & 0xFF;

    if(mode.equals("1"))
    {
        r=inversbit(r, n);
    }
    else if(mode.equals("2"))
    {
        g=inversbit(g, n);
    }
    else if(mode.equals("3"))
    {
        b=inversbit(b, n);
    }
    //set
    bimg.setRGB(x,y, ((a<<24)+(r<<16)+(g<<8)+b));
}

```

Dan yang terakhir adalah fungsi untuk mengganti bit di area-area tertentu dari gambar.

```

public void ganti()
{
    //ubah bit r
    for(int x=100;x<200;x++)
    {
        for(int y=0;y<25;y++)
        {
            invers(x, y, 1, "1");
        }
        for(int y=25;y<50;y++)
        {
            invers(x, y, 2, "1");
        }
        for(int y=50;y<75;y++)
        {
            invers(x, y, 3, "1");
        }
        for(int y=75;y<100;y++)

```

```

{
    invers(x, y, 4, "1");
}
for(int y=100;y<125;y++)
{
    invers(x, y, 5, "1");
}
for(int y=125;y<150;y++)
{
    invers(x, y, 6, "1");
}
for(int y=150;y<175;y++)
{
    invers(x, y, 7, "1");
}
for(int y=175;y<200;y++)
{
    invers(x, y, 8, "1");
}
}
//ubah bit g
for(int x=200;x<300;x++)
{
    for(int y=0;y<25;y++)
    {
        invers(x, y, 1, "2");
    }
    for(int y=25;y<50;y++)
    {
        invers(x, y, 2, "2");
    }
    for(int y=50;y<75;y++)
    {
        invers(x, y, 3, "2");
    }
    for(int y=75;y<100;y++)
    {
        invers(x, y, 4, "2");
    }
    for(int y=100;y<125;y++)
    {
        invers(x, y, 5, "2");
    }
    for(int y=125;y<150;y++)
    {
        invers(x, y, 6, "2");
    }
    for(int y=150;y<175;y++)
    {
        invers(x, y, 7, "2");
    }
    for(int y=175;y<200;y++)
    {
        invers(x, y, 8, "2");
    }
}
//ubah bit r

```

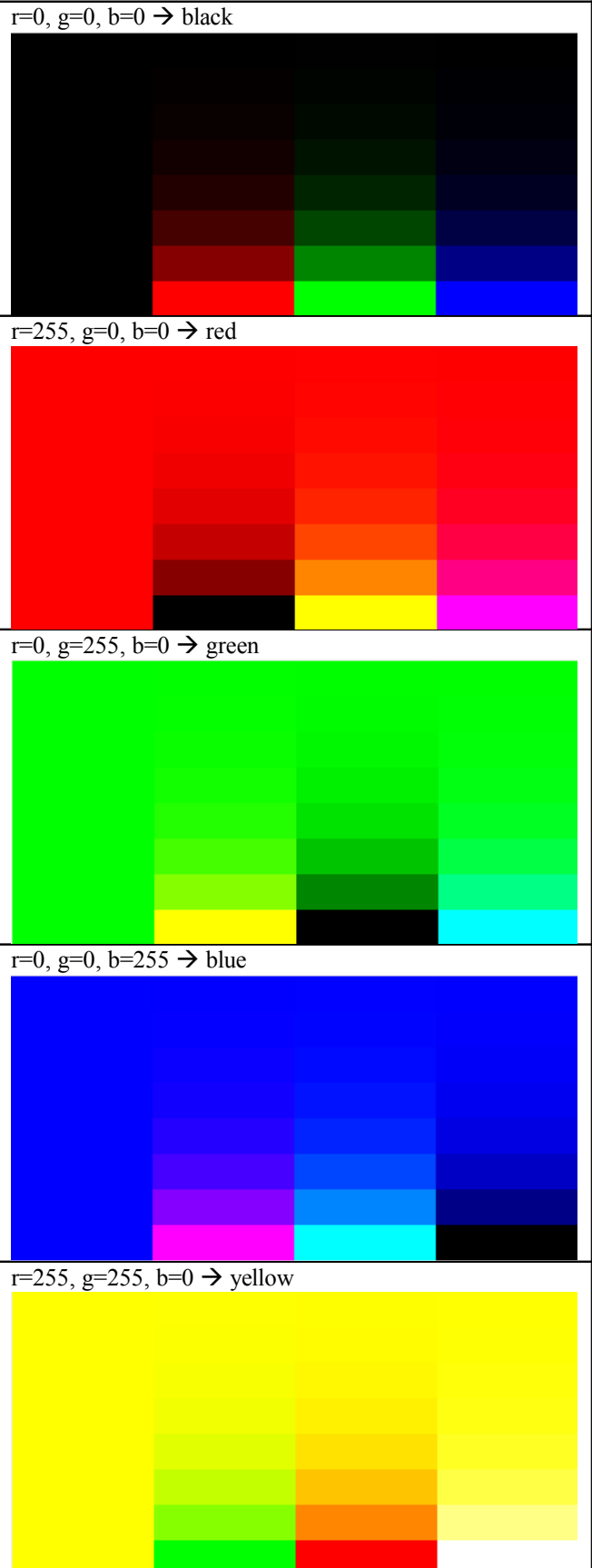
```

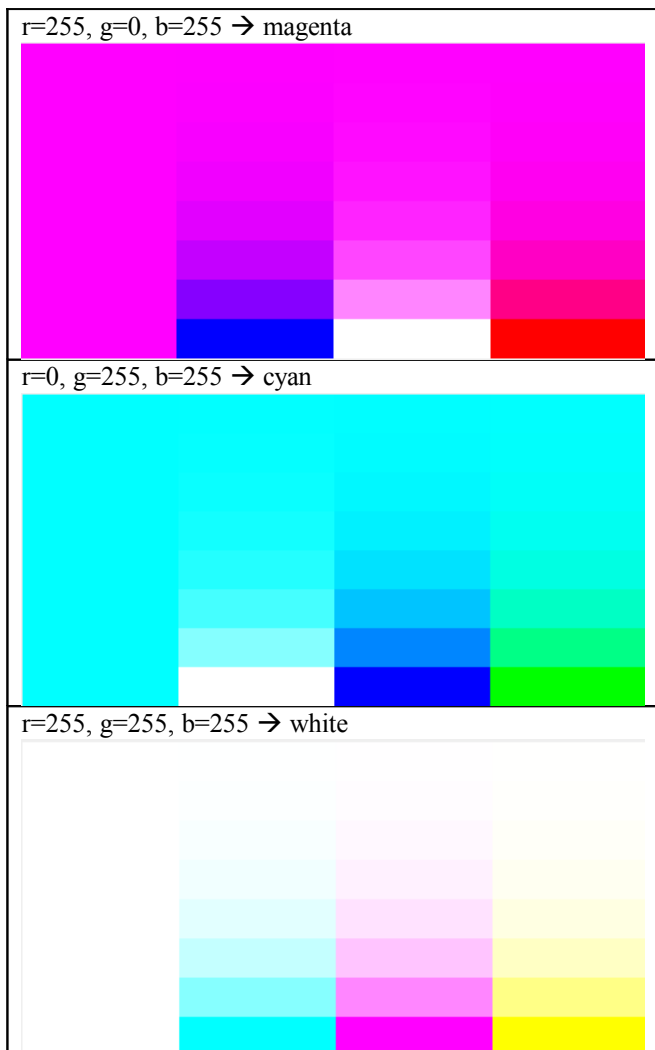
for(int x=300;x<400;x++)
{
  for(int y=0;y<25;y++)
  {
    invers(x, y, 1, "3");
  }
  for(int y=25;y<50;y++)
  {
    invers(x, y, 2, "3");
  }
  for(int y=50;y<75;y++)
  {
    invers(x, y, 3, "3");
  }
  for(int y=75;y<100;y++)
  {
    invers(x, y, 4, "3");
  }
  for(int y=100;y<125;y++)
  {
    invers(x, y, 5, "3");
  }
  for(int y=125;y<150;y++)
  {
    invers(x, y, 6, "3");
  }
  for(int y=150;y<175;y++)
  {
    invers(x, y, 7, "3");
  }
  for(int y=175;y<200;y++)
  {
    invers(x, y, 8, "3");
  }
}

```

Fungsi ganti diatas adalah fungsi yang penting sehingga kita dapat membandingkan perubahan bit. Jadi gambar tersebut nantinya akan dibagi menjadi 4 kolom dengan kolom pertama tetap, kolom kedua yang diinvers adalah bit warna merah sedangkan kolom ketiga warna hijau, dan kolom terakhir adalah warna biru. Sedangkan untuk kolom dibagi menjadi 8 bagian yaitu mulai dari atas jika yang diinvers adalah 1 bit paling belakang kemudian 2 bit di belakang sampai baris terakhir adalah semua bit akan diinvers.

Berikut adalah tabel perubahan warna dari beberapa warna yang sudah saya coba. Ini hanya baru terdiri dari warna-warna yang memiliki bit maksimum dan minimum saja.





jumlah bit yang paling maksimal untuk gambar masih tidak begitu terlihat perubahannya adalah 2 bit, lebih dari itu (mulai dari 3 bit ke atas) sudah lebih mudah untuk dikenali perubahannya (dalam contoh ini penulis menggunakan invers untuk kasus terburuk).

IV. KESIMPULAN

Kesimpulan yang dapat diambil adalah bit hijau termasuk bit yang rawan untuk digunakan dalam LSB karena lebih mudah teramati oleh mata manusia, sedangkan bit biru bisa disisipkan lebih karena tidak begitu kelihatan perubahannya. Kemudian jumlah masimum bit yang disisipkan agar tidak begitu berubah adalah 2 bit lebih dari itu akan kelihatan perubahannya jika terjadi kemungkinan terburuk.

Saran untuk pengembangan dari makalah ini adalah dengan memperbanyak sample warna yang diamati dan jumlah pengamat (voting) sehingga hasilnya lebih objektif dan lengkap. Sebab penulis mungkin masih mengambil sampel yang sedikit (baik warna maupun pengamat) karena keterbatasan waktu.

DAFTAR PUSTAKA

- [1] <http://id.wikipedia.org/wiki/Kriptografi>
- [2] <http://id.wikipedia.org/wiki/Steganografi>
- [3] Slide Kriptografi ITB 2012
- [4] Slide Sistem Multimedia ITB 2012

Kemudian berikut ini review yang dapat penulis ambil dengan melibatkan pendapat beberapa orang (setelah diberikan pertanyaan satu-persatu) dari percobaan-percobaan di atas tentang batas bit yang perbedaannya mulai terlihat dari tabel-tabel warna diatas.

R,G,B	R	G	B
0,0,0	3	3	3
255,0,0	3	4	4
0,255,0	5	3	5
0,0,255	3	3	5
255,255,0	3	3	6
255,0,255	3	5	4
0,255,255	5	3	5
255,255,255	3	3	5

Setelah dilihat ternyata memang warna hijau memang pada umumnya terdapat di tingkatan bit yang paling rendah daripada yang lain. Hanya pada warna merah dan magenta yang bit hijau tidak menjadi tingkatan terendah. Dan jika dilihat bit yang paling tidak kelihatan / memiliki tingkatan paling tinggi adalah bit untuk warna biru. Sehingga kita bisa mengurangi penyisipan pada bit hijau dan dipindah ke bit biru. Dari tabel juga terlihat bahwa

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Maret 2012

Eric Christopher / 13509037