

Bab 12

Warna

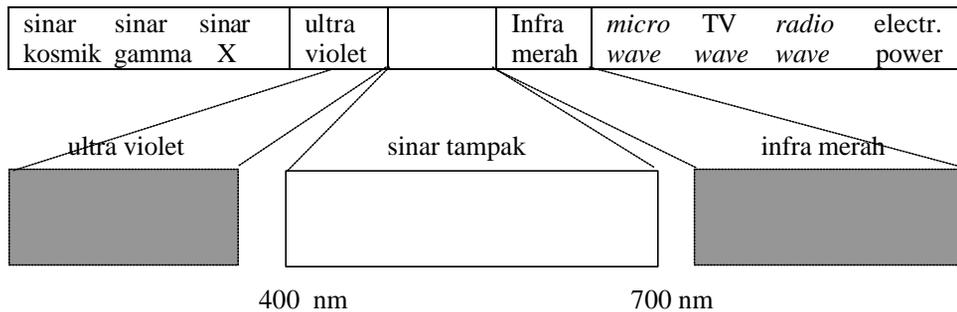
Persepsi visual citra berwarna (*color images*) umumnya lebih kaya dibandingkan dengan citra hitam putih (*greyscale*), karena itu citra berwarna lebih disenangi daripada citra hitam putih. Citra berwarna menampilkan warna objek seperti warna aslinya (meskipun tidak selalu tepat demikian). Bab ini akan menguraikan konsep warna, model warna, dan transformasi warna dari satu model ke model lainnya.

12.1 Dasar-dasar Warna

Warna yang diterima oleh mata dari sebuah objek ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut. Sebagai contoh, suatu objek berwarna hijau karena objek tersebut memantulkan sinar biru dengan panjang gelombang 450 sampai 490 nanometer (nm).

Warna sinar yang direspon oleh mata adalah sinar tampak (*visible spectrum*) dengan panjang gelombang berkisar dari 400 (biru) sampai 700 nm (merah). Lihat Gambar 12.1.

10^{-5} 10^{-1} 1 10 400 700 2500 nm 1000 m



Gambar 12.1 Spektrum cahaya

Warna-warna yang diterima oleh mata manusia merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (*R*), *green* (*G*), dan *blue* (*B*). Ketiga warna tersebut dinamakan **warna pokok** (*primaries*), dan sering disingkat sebagai warna dasar *RGB*. Warna-warna lain dapat diperoleh dengan mencampurkan ketiga warna pokok tersebut dengan perbandingan tertentu (meskipun tidak sepenuhnya benar, karena tidak semua kemungkinan warna dapat dihasilkan dengan kombinasi *RGB* saja), sesuai dengan teori Young (1802) yang menyatakan bahwa sembarang warna dapat dihasilkan dari percampuran warna-warna pokok C_1 , C_2 , dan C_3 dengan persentase tertentu [PIT93]:

$$C = a C_1 + b C_2 + c C_3 \tag{12.1}$$

Bila citra warna didigitasi, maka tiga buah *filter* digunakan untuk mengekstraksi intensitas warna merah, hijau, dan biru, dan bila ketiganya dikombinasikan kita memperoleh persepsi warna.

12.2 Atribut Warna

Selain *RGB*, warna juga dapat dimodelkan berdasarkan atribut warnanya. Setiap warna memiliki 3 buah atribut, yaitu *intensity* (*I*), *hue* (*H*), dan *saturation* (*S*).

a. *Intensity/brightness/luminance*

Atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna. Kisaran nilainya adalah antara gelap (hitam) dan terang (putih).

b. *Hue*

Menyatakan warna sebenarnya, seperti merah, violet, dan kuning. *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dsb, dari cahaya. *Hue* berasosiasi dengan panjang gelombang cahaya, dan bila kita menyebut warna merah, violet, atau kuning, kita sebenarnya menspesifikasikan *hue*-nya.

c. *Saturation*

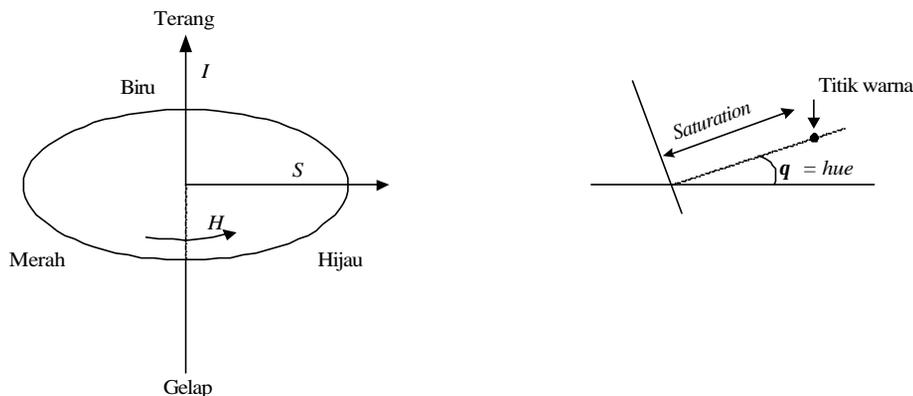
Menyatakan tingkat kemurnian warna cahaya, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Sebagai contoh, warna merah adalah 100% warna jenuh (*saturated color*), sedangkan warna *pink* adalah warna merah dengan tingkat kejenuhan sangat rendah (karena ada warna putih di dalamnya). Jadi, jika *hue* menyatakan warna sebenarnya, maka *saturation* menyatakan seberapa dalam warna tersebut.

Dalam praktek, *hue* dikuantisasi dengan nilai dari 0 sampai 255; 0 menyatakan merah, lalu memutar nilai-nilai spektrum tersebut kembali lagi ke 0 untuk menyatakan merah lagi. Ini dapat dipandang sebagai sudut dari 0° sampai 360° .

Jika suatu warna mempunyai *saturation* = 0, maka warna tersebut tanpa *hue*, yaitu dibuat dari warna putih saja. Jika *saturation* = 255, maka tidak ada warna putih yang ditambahkan pada warna tersebut. *Saturation* dapat digambarkan sebagai panjang garis dari titik pusat lingkaran ke titik warna.

Intensity nilainya dari gelap sampai terang (dalam praktek, gelap = 0, terang = 255). *Intensity* dapat digambarkan sebagai garis vertikal yang menembus pusat lingkaran.

Ketiga atribut warna (*I*, *H*, dan *S*) digambarkan dalam model *IHS* (ada juga yang menyebutnya model *HSV*, dengan $V = Value = I$) yang diperlihatkan pada Gambar 12.2.



Gambar 12.2 Model IHS

12.3 Sistem Koordinat Warna

CIE (*Commission International de l'Eclairage*) atau *International Lighting Committee* adalah lembaga yang membakukan warna pada tahun 1931. *CIE* mula-mula menstandarkan panjang gelombang warna-warna pokok sebagai berikut [MEN89]:

<i>R</i>	: 700 nm
<i>G</i>	: 546.1 nm
<i>B</i>	: 435.8 nm

Warna-warna lain dapat dihasilkan dengan mengkombinasikan ketiga warna pokok tersebut. Model warna yang digunakan sebagai acuan dinamakan model *RGB*.

RGB bukan satu-satunya warna pokok yang dapat digunakan untuk menghasilkan kombinasi warna. Warna lain dapat juga digunakan sebagai warna pokok (misalnya *C* = *Cyan*, *M* = *Magenta*, dan *Y* = *Yellow*).

Karena itu *CIE* mendefinisikan model warna dengan menggunakan warna-warna fiktif (yaitu, warna yang secara fisik tidak dapat direalisasikan), yang dilambangkan dengan *X*, *Y*, dan *Z*. Model warna tersebut dinamakan model *XYZ*. Warna-warna dispesifikasikan dengan jumlah relatif warna pokok fiktif. Keuntungan utama dari model ini adalah *luminance* atau *brightness* sinyal disediakan langsung oleh *Y*. Jadi, nilai *Y* memberikan citra *greyscale* dari citra berwarnanya.

Kromatisitas (*chromaticity of color*) masing-masing warna pokok, menunjukkan persentase relatif suatu warna pokok di antara warna pokok lainnya pada warna yang diberikan, yang didefinisikan sebagai

$$x = \frac{X}{X + Y + Z} \quad (12.2)$$

$$y = \frac{Y}{X + Y + Z} \quad (12.3)$$

$$z = \frac{Z}{X + Y + Z} \quad (12.4)$$

Warna putih acuan dinyatakan dengan $X = Y = Z = 1$. Jumlah seluruh nilai kromatisitas warna adalah satu:

$$x + y + z = 1 \quad (12.5)$$

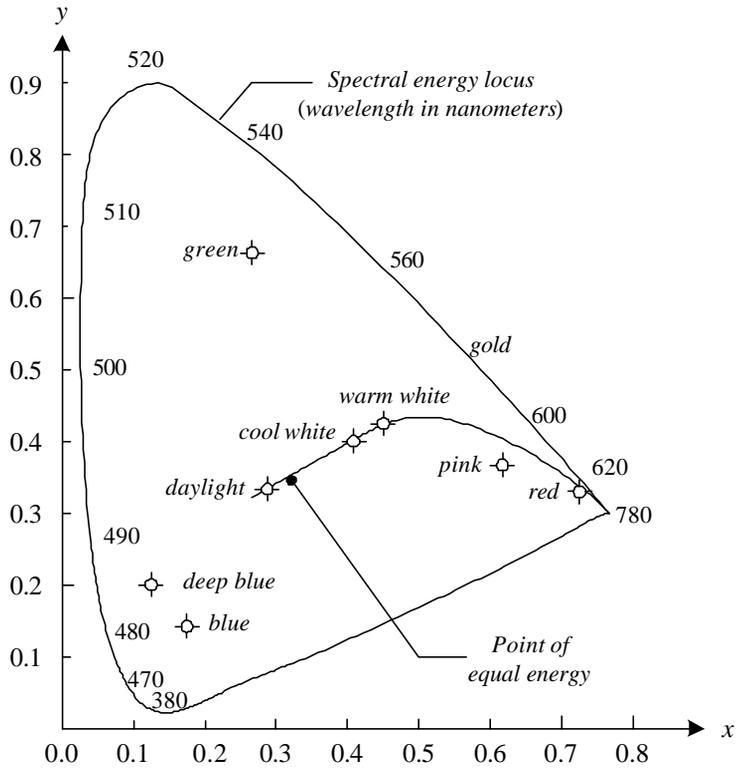
atau

$$z = 1 - (x + y) \quad (12.6)$$

Jelas hanya dua nilai x dan y yang dibutuhkan untuk menspesifikasikan kromatisitas warna, karena jika x dan y diketahui, z dapat dihitung dengan persamaan 12.6. Warna lebih tepat dinyatakan dengan kromatisitas x dan y dan luminansi Y . Koordinat kromatisitas (12.2 dan 12.3) digunakan untuk menggambarkan **diagram kromatisitas** pada Gambar 12.3.

Titik yang ditandai “green” pada diagram Gambar 12.3 memiliki kira-kira 62% green dan 25% red. Ini sesuai dengan persamaan 12.6 bahwa komposisi *blue* kira-kira 13%. Posisi bermacam-macam spektrum warna, dari *violet* 380 nm sampai *red* 780 nm dinyatakan pada sisi (*boundary*) diagram yang berbentuk setengah elips. Ini adalah warna-warna “*pure*”. Titik yang tidak terletak pada sisi tetapi masih di dalam diagram menyatakan campuran spektrum warna. Titik **energi setara** (*equal energy*) berkoresponden dengan nilai fraksi yang sama dari ketiga warna pokok X , Y , Z . Titik energi setara menyatakan bakuan *CIE* untuk cahaya putih. Sembarang titik yang terletak pada sisi diagram dikatakan jenuh (*saturated*). Semakin jauh titik-titik itu meninggalkan sisi dan mendekati titik energi setara, itu berarti semakin banyak cahaya putih ditambahkan pada warna dan ia menjadi kurang jenuh (*less saturated*). Kejenuhan titik energi setara adalah nol.

Garis lurus yang menghubungkan dua titik di dalam diagram mendefinisikan semua variasi warna berbeda yang dapat diperoleh dengan mengkombinasikan dua warna ini. Sebagai contoh, garis lurus dari *red* ke *green*. Jika lebih banyak cahaya *red* daripada *green*, maka titik warna baru terletak pada segmen garis, tetapi ia akan lebih dekat ke *red* daripada ke *green*.



Gambar 12.3 Diagram kromatisitas CIE

Hue dari warna tertentu diperoleh dengan menarik garis dari putih ke sisi elips melalui warna tersebut. Misalkan digambar garis dari posisi *W* (putih) melalui warna tertentu, *P*, ke sisi elips pada posisi *H*. Nilai *hue* adalah *H*, dan *saturation* adalah panjang *WP* relatif terhadap *WH*. Warna *P* dapat dipandang sebagai campuran warna putih dan *hue*:

$$P = SH + (1 - S)W \quad (12.7)$$

yang dalam hal ini *S* mengendalikan perbandingan relatif warna putih dan *hue*.

12.4 Transformasi Sistem Koordinat Warna

Transformasi warna dari basis *CIE RGB* ke *CIE XYZ* dapat dilakukan sebagai berikut: Diberikan triplet *RGB* (R_i, G_i, B_i) untuk *pixel* i , maka triplet *XYZ* (X_i, Y_i, Z_i) dihitung dengan

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.99 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} \quad (12.8)$$

Transformasi sebaliknya dari *CIE XYZ* ke *CIE RGB* dapat dilakukan dengan persamaan

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 2.365 & -0.3896 & -0.468 \\ -0.515 & 1.425 & 0.088 \\ 0.005 & -0.014 & 1.009 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (12.9)$$

Algoritma transformasi citra dari model warna *CIE RGB* ke model warna *CIE XYZ* diperlihatkan pada Algoritma 12.1, sedangkan algoritma transformasi balikan dari model warna *CIE XYZ* ke model warna *CIE RGB* diperlihatkan pada Algoritma 12.2 [PIT93]. Kedua algoritma tersebut mengasumsikan bahwa komponen *RGB* disimpan masing-masing di dalam matriks r , g , dan b . Hasil transformasi masing-masing disimpan di dalam matriks x , y , dan z . Ukuran citra adalah $N \times M$.

```
int cieRGB_toXYZ(citra r, citra g, citra b,
                 citra x, citra y, citra z, int N, int M)
/* Transformasi citra dari model warna CIE RGB ke model CIE XYZ
   Masukan: citra dengan kompoenen RGB masing-masing disimpan di dalam
            matriks r, g, dan b. Ketga matriks ini berukuran N x M.
   Keluaran: citra dengan komponen XYZ masing-masing disimpan di dalam
            matriks x, y, dan z.
*/
{
    int i, j; double R, G, B; double X, Y, Z;

    for (i=0; i<=N-1; i++)
        for (j=0; j<=M-1; j++)
        {
            R = (double)r[i][j]; G=(double)g[i][j]; B=(double)b[i][j];
            X = 0.490*R+0.310*G+0.200*B;
            Y = 0.177*R+0.813*G+0.011*B;
            Z = 0.010*G+0.990*B;
            if (X > 255.0) x[i][j]=255; else x[i][j]=(unsigned char)X;
            if (Y > 255.0) y[i][j]=255; else y[i][j]=(unsigned char)Y;
            if (Z > 255.0) z[i][j]=255; else z[i][j]=(unsigned char)Z;
        }
}
```

Algoritma 12.1 Transformasi citra dari model warna *CIE RGB* ke model *CIE XYZ*

```

int XYZ_to_cieRGB(citra x, citra y, citra z,
                 citra r, citra g, citra b, int N, int M)
/* Transformasi citra dari model warna CIE XYZ ke model CIE RGB
Masukan: citra dengan komponen XYZ masing-masing disimpan di dalam
matriks x, y, dan z. Ketga matriks ini berukuran N x M.
Keluaran: citra dengan komponen RGB masing-masing disimpan di dalam
matriks r, g, dan b.
*/
{
    int i, j; double R, G, B; double X, Y, Z;

    for (i=0; i<=N-1; i++)
        for (j=0; j<=M-1; j++)
            { X = (double)x[i][j]; Y =(double)y[i][j]; Z =(double)z[i][j];
              R = 2.365*X-0.896*Y-0.468*Z;
              G = -0.515*X+1.425*Y+0.088*Z;
              B = 0.005*X-0.014*Y+1.009*Z;
              if (R > 255.0) r[i][j]=255;
              else if (R<0.0) r[i][j]=0;
              else r[i][j]=(unsigned char)R;
              if (G > 255.0) g[i][j]=255;
              else if (G<0.0) g[i][j]=0;
              else g[i][j]=(unsigned char)G;
              if (B > 255.0) b[i][j]=255;
              else if (B<0.0) b[i][j]=0;
              else b[i][j]=(unsigned char)B;
            }
}

```

Algoritma 12.2 Transformasi citra dari model warna CIE XYZ ke model CIE RGB

Model warna *RGB* dan *XYZ* yang dikemukakan di atas adalah bakuan dari *CIE*. Ada juga model warna yang diusulkan untuk *platform* perangkat keras tertentu untuk menampilkan gambar. Misalnya *National Television Systems Committee (NTSC)* menggunakan model warna *RGB* khusus untuk menampilkan citra berwarna pada layar *CRT*. Format *NTSC* digunakan pada televisi di Amerika Serikat. Salah satu keuntungan utama dari format ini adalah data *greyscale* dipisahkan dari data warnanya, sehingga sinyal yang sama dapat digunakan baik untuk layar berwarna maupun layar hitam putih.

Pada format *NTSC*, data citra terdiri atas tiga komponen, yaitu *luminance (Y)*, *hue (I)*, dan *saturation (Q)*. Komponen pertama, yaitu *Y* menyatakan data *greyscale*, sedangkan dua komponen terakhir membentuk *chrominance*. Jika diberikan triplet *NTSC RGB (R_i, G_i, B_i)* untuk *pixel i*, maka nilai *YIQ* untuk *pixel* yang bersangkutan dihitung dengan

$$\begin{bmatrix} Y_i \\ I_i \\ Q_i \end{bmatrix} = \begin{bmatrix} 0.299 & 0.857 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} \quad (12.10)$$

Nilai *NTSC RGB* semula dapat dihitung dengan menggunakan transformasi balikan:

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.273 & -0.647 \\ 1.000 & -1.104 & 1.701 \end{bmatrix} \begin{bmatrix} Y_i \\ I_i \\ Q_i \end{bmatrix} \quad (12.11)$$

Transformasi dari *NTSC RGB* ke *CIE RGB* dihitung dengan persamaan:

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 1.167 & -0.146 & -0.151 \\ 0.114 & 0.753 & 0.159 \\ -0.001 & 0.059 & 1.128 \end{bmatrix} \begin{bmatrix} R_NTSC_i \\ G_NTSC_i \\ B_NTSC_i \end{bmatrix} \quad (12.12)$$

Sistem *NTSC RGB* dapat ditransformasikan ke sistem *CIE XYZ* dengan persamaan:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{bmatrix} \begin{bmatrix} R_NTSC_i \\ G_NTSC_i \\ B_NTSC_i \end{bmatrix} \quad (12.13)$$

sedangkan transformasi dari *CIE XYZ* ke *NTSC RGB* dihitung dengan persamaan:

$$\begin{bmatrix} R_NTSC_i \\ G_NTSC_i \\ B_NTSC_i \end{bmatrix} = \begin{bmatrix} 1.910 & -0.533 & -0.288 \\ -0.985 & 2.000 & -0.028 \\ 0.058 & -0.118 & 0.896 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (12.14)$$

12.5 Model Warna *CMY* dan *CMYK*

Warna *cyan* (*C*), *magenta* (*M*), dan *yellow* (*Y*) adalah warna komplementer terhadap *red*, *green*, dan *blue*. Dua buah warna disebut komplementer jika dicampur dengan perbandingan yang tepat menghasilkan warna putih. Misalnya, *magenta* jika dicampur dengan perbandingan yang tepat dengan *green* menghasilkan putih, karena itu *magenta* adalah komplemen dari *green*.

Model *CMY* dapat diperoleh dari model *RGB* dengan perhitungan berikut:

$$C = 1 - R \quad (12.15)$$

$$M = 1 - G \quad (12.16)$$

$$Y = 1 - B \quad (12.17)$$

Model *CMY* dapat digunakan untuk mencetak citra berwarna, tetapi karena ketidaksempurnaan tinta, model *CMY* tidak dapat menghasilkan warna hitam

dengan baik. Karena itu, model *CMY* disempurnakan menjadi model *CMYK*, yang dalam hal ini *K* menyatakan warna keempat, dengan perhitungan sebagai berikut [PIT93]:

$$K = \min(C, M, Y) \quad (12.18)$$

$$C = C - K \quad (12.19)$$

$$M = M - K \quad (12.20)$$

$$Y = Y - K \quad (12.21)$$

Algoritma transformasi citra dari model *RGB* ke model *CMYK* ditunjukkan pada Algoritma 12.3, sedangkan algoritma transformasi balikan dari model *CMYK* ke model *RGB* ditunjukkan pada Algoritma 12.4 [PIT93]. Kedua algoritma tersebut mengasumsikan komponen *RGB* disimpan masing-masing di dalam matriks *r*, *g*, dan *b*. Hasil transformasi masing-masing disimpan di dalam matriks *c*, *m*, *y* dan *k*. Ukuran citra adalah $N \times M$.

```
int RGB_toCMYK(citra r, citra g, citra b,
               citra c, citra m, citra y, citra k,
               int N, int M)

/* Transformasi citra dari model warna RGB ke model CMYK.
   Masukan: citra dengan komponen RGB masing-masing disimpan di dalam
   matriks r, g, dan b. Ketiga matriks ini berukuran N x M.
   Keluaran: citra dengan komponen CMYK masing-masing disimpan di dalam
   matriks c, y, m, dan k.
*/

{
  int i, j;

  for (i=0; i<=N-1; i++)
    for (j=0; j<=M-1; j++)
      {
        c[i][j]=(unsigned char)255 - r[i][j];
        m[i][j]=(unsigned char)255 - g[i][j];
        y[i][j]=(unsigned char)255 - b[i][j];
        k[i][j]=c[i][j];
        if (m[i][j]<k[i][j]) k[i][j]=m[i][j];
        if (y[i][j]<k[i][j]) k[i][j]=y[i][j];
        c[i][j]=c[i][j]-k[i][j];
        m[i][j]=m[i][j]-k[i][j];
        y[i][j]=y[i][j]-k[i][j];
      }
}
```

Algoritma 12.3 Transformasi citra dari model warna *RGB* ke model *CMYK*.

```

int CMYKtoRGB(citra c, citra y, citra m, citra k,
              citra r, citra g, citra b,
              int N, int M)

/* Transformasi citra dari model warna CMYK ke model RGB.
   Masukan: citra dengan komponen CMYK masing-masing disimpan di dalam
            matriks c, y, m, dan k. Ketiga matriks ini berukuran N x M.
   Keluaran: citra dengan komponen RGB masing-masing disimpan di dalam
            matriks r, g, dan b.
*/
{
    int i, j;

    for (i=0; i<=N-1; i++)
        for (j=0; j<=M-1; j++)
            {
                c[i][j]=c[i][j]+k[i][j];
                m[i][j]=m[i][j]+k[i][j];
                y[i][j]=y[i][j]+k[i][j];
                k[i][j]=c[i][j];
                r[i][j]=(unsigned char)255 - c[i][j];
                g[i][j]=(unsigned char)255 - m[i][j];
                b[i][j]=(unsigned char)255 - y[i][j];
            }
}

```

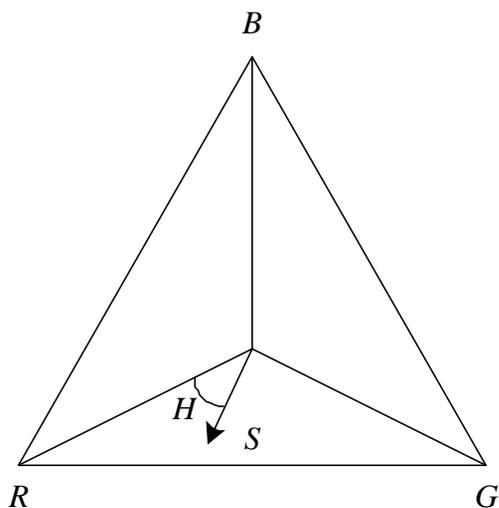
Algoritma 12.4 Transformasi citra dari model warna CMYK ke model RGB

17.6 Transformasi Warna RGB ke IHS

Meskipun basis *RGB* bagus untuk menampilkan informasi warna, tetapi ia tidak cocok untuk beberapa aplikasi pemrosesan citra. Pada aplikasi pengenalan objek, lebih mudah mengidentifikasi objek dengan perbedaan *hue*-nya dengan cara memberikan nilai ambang pada rentang nilai-nilai *hue* (panjang gelombang spektrum) yang melingkupi objek. Masalahnya, bagaimana melakukan pengembangan pada ruang warna *RGB* dan apa rumus untuk mengaplikasikannya? Masalah ini lebih mudah dipecahkan bila nilai *RGB* dikonversi ke nilai *intensity* (*I*), *hue* (*H*), dan *saturation* (*S*).

Aplikasi yang lain misalnya dalam pemampatan citra. Melakukan pemampatan secara terpisah pada setiap nilai *R*, *G*, dan *B* tidak disarankan, karena data yang dimampatkan 3 kali lebih banyak dan waktu pemampatannya 3 kali lebih lama daripada waktu pemampatan citra skala-abunya. Pemampatan citra berwarna lebih relevan bila warna *RGB*-nya dikonversikan ke *IHS* karena algoritma pemampatan pada citra skala-abu dilakukan pada komponen *I*, sedangkan nilai *H* dan *S* dikodekan dengan cara yang lain dengan sedikit atau sama sekali tidak ada degradasi.

Model warna *IHS* merepresentasikan warna dalam terminologi *intensity*, *hue*, dan *saturation*. Dari diagram kromatisitas, buatlah segitiga yang menghubungkan tiga warna pokok *red*, *green*, *blue* (Gambar 12.4). Titik-titik pada segitiga menyatakan warna yang dihasilkan dari pencampuran warna titik sudut, sedangkan titik-titik di dalam segitiga menyatakan warna yang dapat dihasilkan dengan mengkombinasikan tiga warna titik sudut. Titik tengah segitiga menyatakan warna putih, yaitu pencampuran warna pokok dengan fraksi yang sama.



Gambar 12.4 Segitiga *IHS*

Komponen *RGB* dari citra berwarna dapat dikonversikan ke model warna *IHS*. Dengan mengasumsikan komponen *RGB* telah dinormalisasikan ke 1, maka *I* dihitung dengan rumus:

$$I = \frac{1}{3}(R + G + B) \quad (12.22)$$

Persamaan 12.22 di atas sering digunakan untuk mengubah citra berwarna menjadi citra skala-abu.

H dihitung dengan rumus:

$$H = \cos^{-1} \frac{2R - G - B}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \quad (12.23)$$

Nilai S dihitung dengan rumus:

$$S = 1 - \frac{3}{R+G+B} \min(R, G, B) \quad (12.24)$$

Alternatif lain mengubah model RGB ke model IHS adalah sebagai berikut. Konversi dari model RGB ke model IHS dilakukan dalam dua tahap. Tahap pertama adalah merotasikan koordinat RGB ke sistem koordinat (I, V_1, V_2) dengan transformasi:

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & \sqrt{3}/3 & \sqrt{3}/3 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (12.25)$$

Langkah kedua adalah menghitung H dan S dari koordinat (V_1, V_2) :

$$H = \tan^{-1}(V_2/V_1) \quad (12.26)$$

$$S = (V_1^2 + V_2^2)^{1/2} \quad (12.27)$$

Nilai H adalah dalam selang $[0, 2\pi]$ atau setara dengan $[0, 360^\circ]$.

Transformasi dari model IHS ke model RGB dapat dilakukan dengan prosedur balikan:

$$V_1 = S \cos(H) \quad (12.28)$$

$$V_2 = S \sin(H) \quad (12.29)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & 0 & 2/\sqrt{6} \\ \sqrt{3}/3 & 1/\sqrt{2} & -1/\sqrt{6} \\ \sqrt{3}/3 & -1/\sqrt{2} & -1/\sqrt{6} \end{bmatrix} \begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} \quad (12.30)$$

Dengan transformasi RGB ke IHS , maka algoritma pemrosesan citra yang semula untuk citra skala-abu dapat diterapkan pada komponen *intensity*, sedangkan algoritma segmentasi citra dapat dilakukan pada komponen H .

Transformasi citra dari basis *RGB* ke basis *IHS* dilakukan sebelum pemrosesan citra. Citra yang sudah diproses dapat dikonversikan kembali ke basis *RGB* untuk tujuan *display*.

National Television Systems Committee (NTSC) menggunakan model *YIQ* untuk mentransformasikan model *RGB* ke *IHS*, yang dalam hal ini *Y* menyatakan *intensity*, *I* menyatakan *hue*, dan *Q* menyatakan *saturation* [PIT93].