16 – Warna (Bagian 1)

IF4073 Interpretasi dan Pengolahan Citra

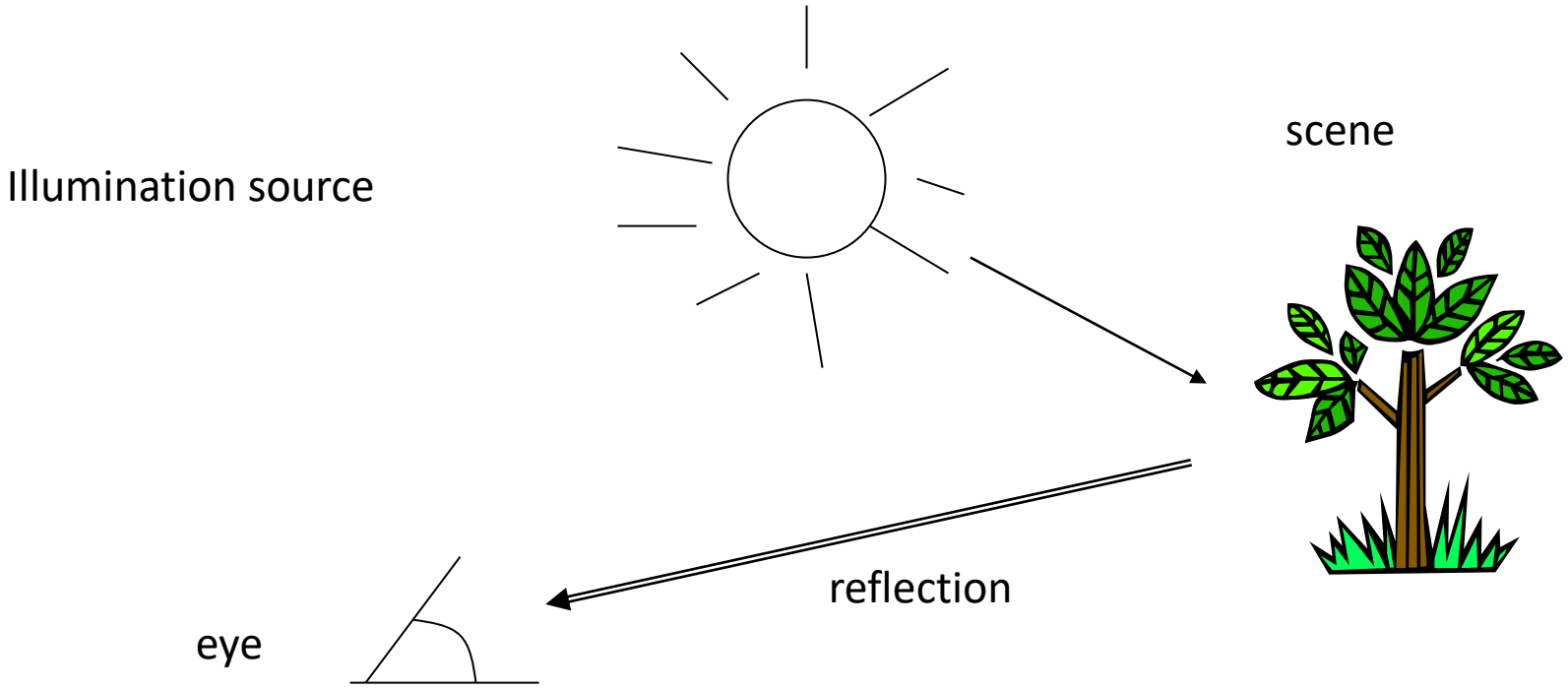
Oleh: Rinaldi Munir

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

Pendahuluan

- Citra berwarna memiliki informasi yang lebih kaya daripada citra *grayscale* atau citra biner.

- Warna yang diterima oleh mata (atau lensa kamera) dari sebuah objek ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut.



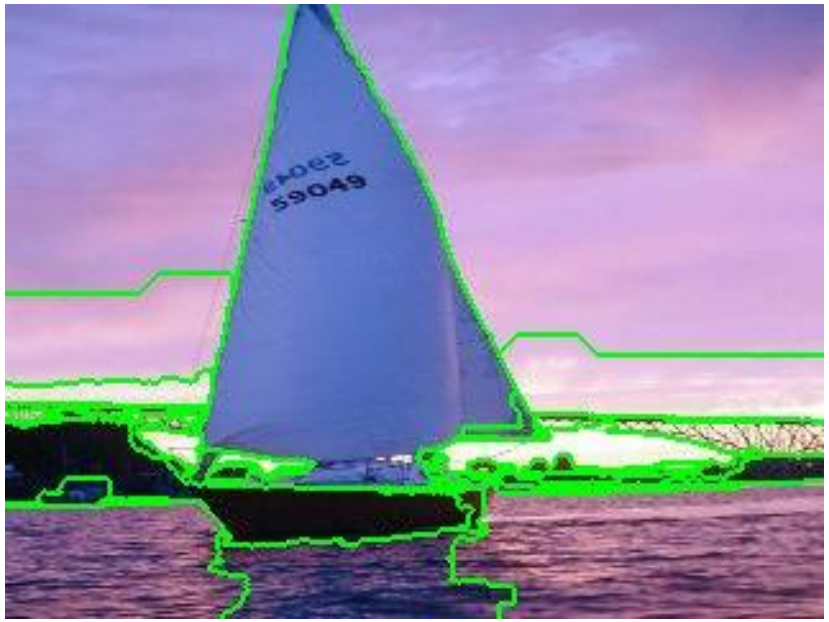
Sumber: Jen-Chang Liu, Spring 2006
Color Image Processing

- Sebagai contoh, suatu objek berwarna biru karena objek tersebut memantulkan sinar biru dengan panjang gelombang 450 sampai 490 nanometer (nm).

- Kenapa menggunakan warna di dalam pengolahan citra?
  - Warna adalah deskriptor yang berdayaguna (powerful descriptor)
    - Identifikasi objek dan ekstraksinya
    - Contoh: *Face detection* dengan menggunakan warna kulit

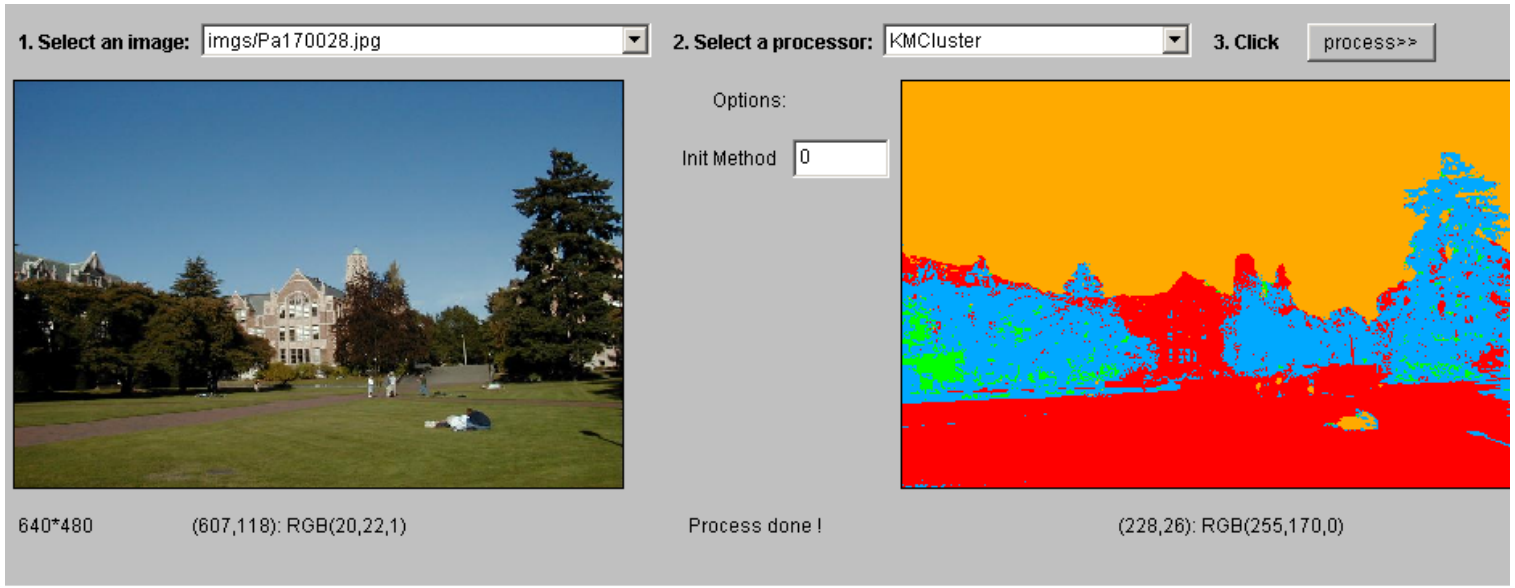


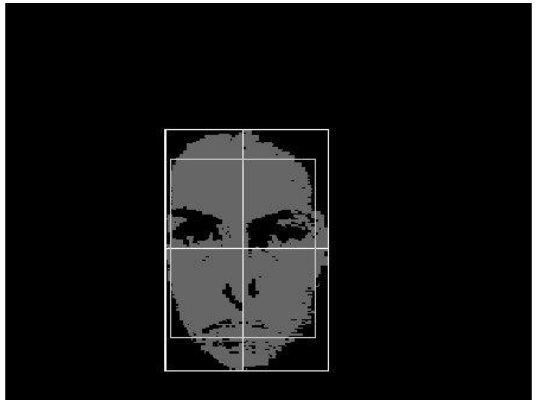
- Segmentasi citra menjadi region-region berdasarkan warna



Original RGB Image

Color Clusters by K-Means


1. Select an image:
imgs/Pa170028.jpg
2. Select a processor:
KMCluster
3. Click
process>>
Options:
Init Method
0
640*480
(607,118): RGB(20,22,1)
Process done !
(228,26): RGB(255,170,0)



Teori Warna

- 1666, Isaac Newton

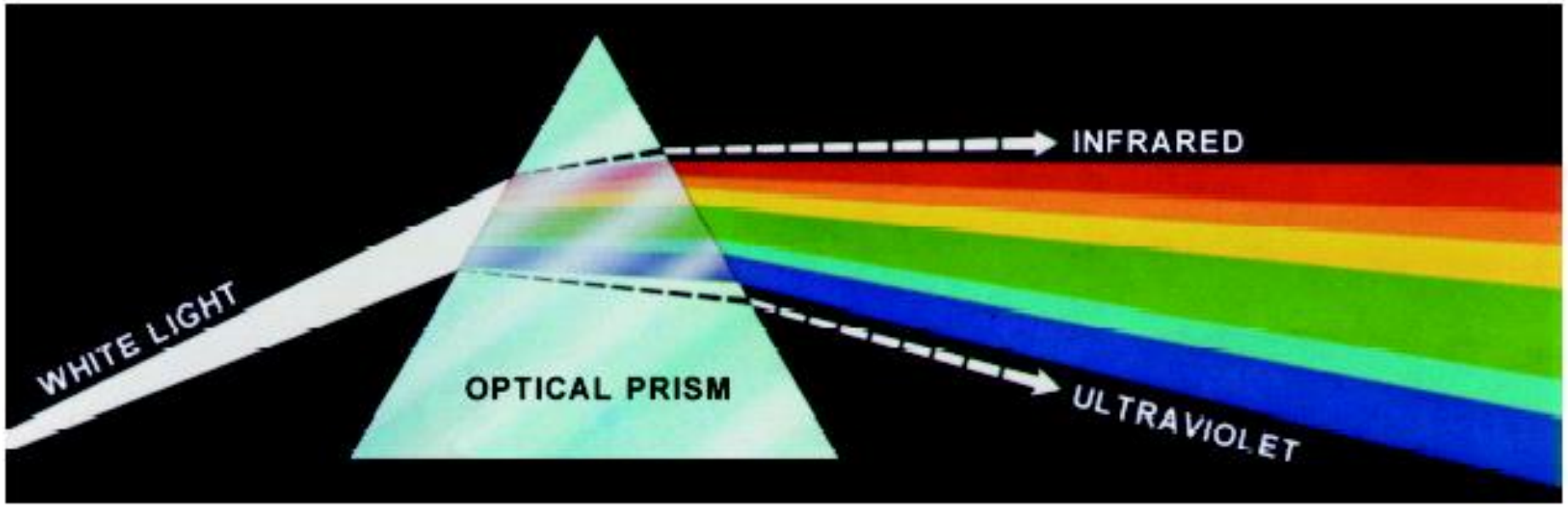
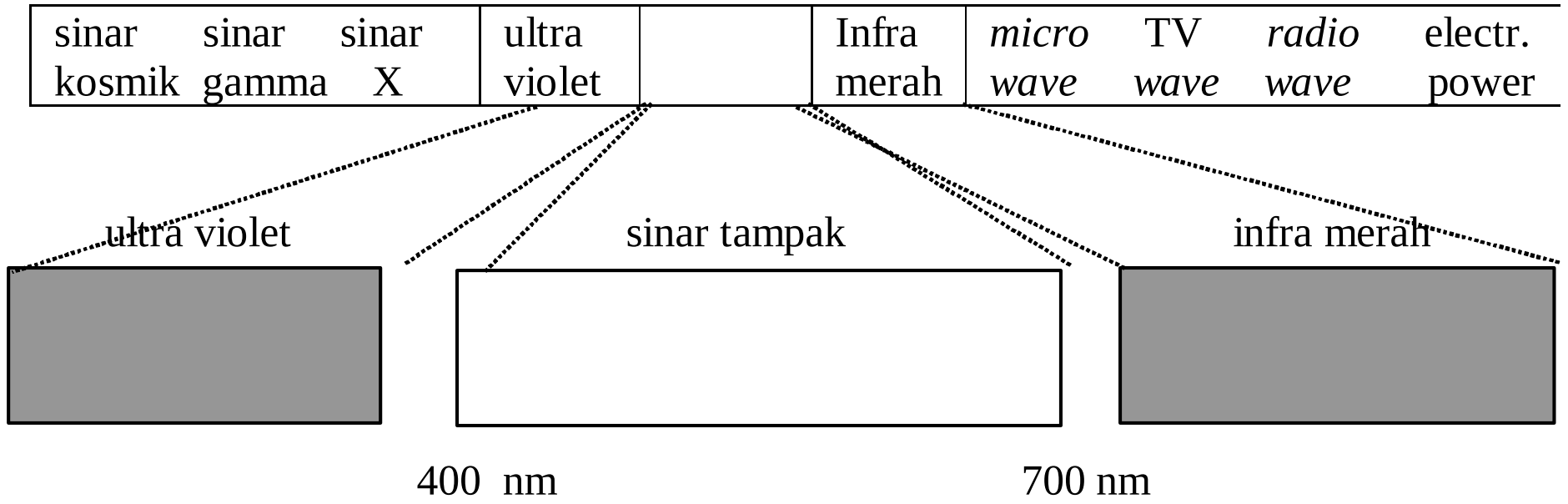


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

Warna Tampak

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



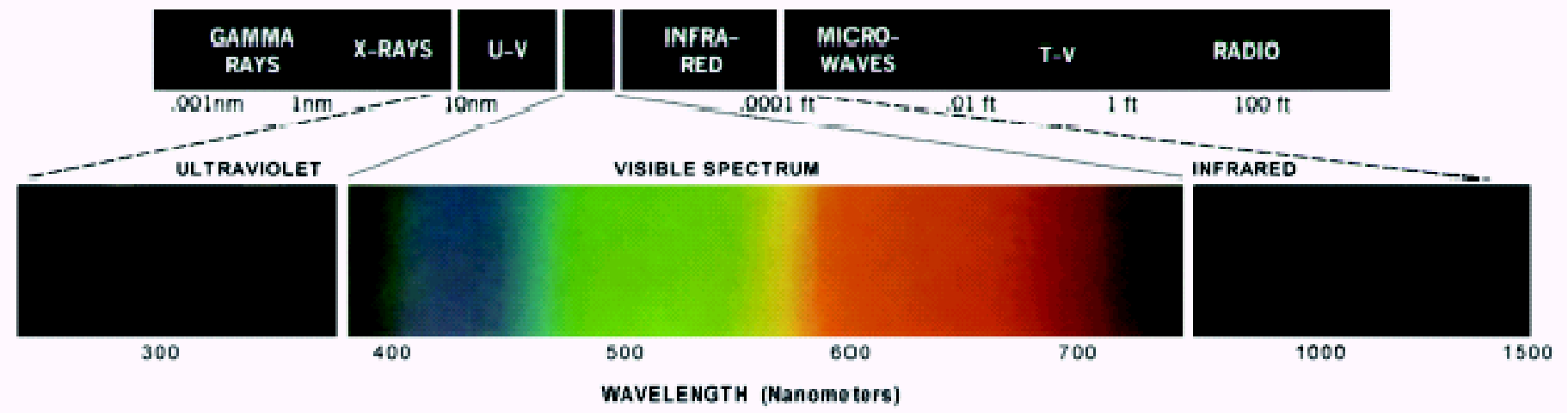


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

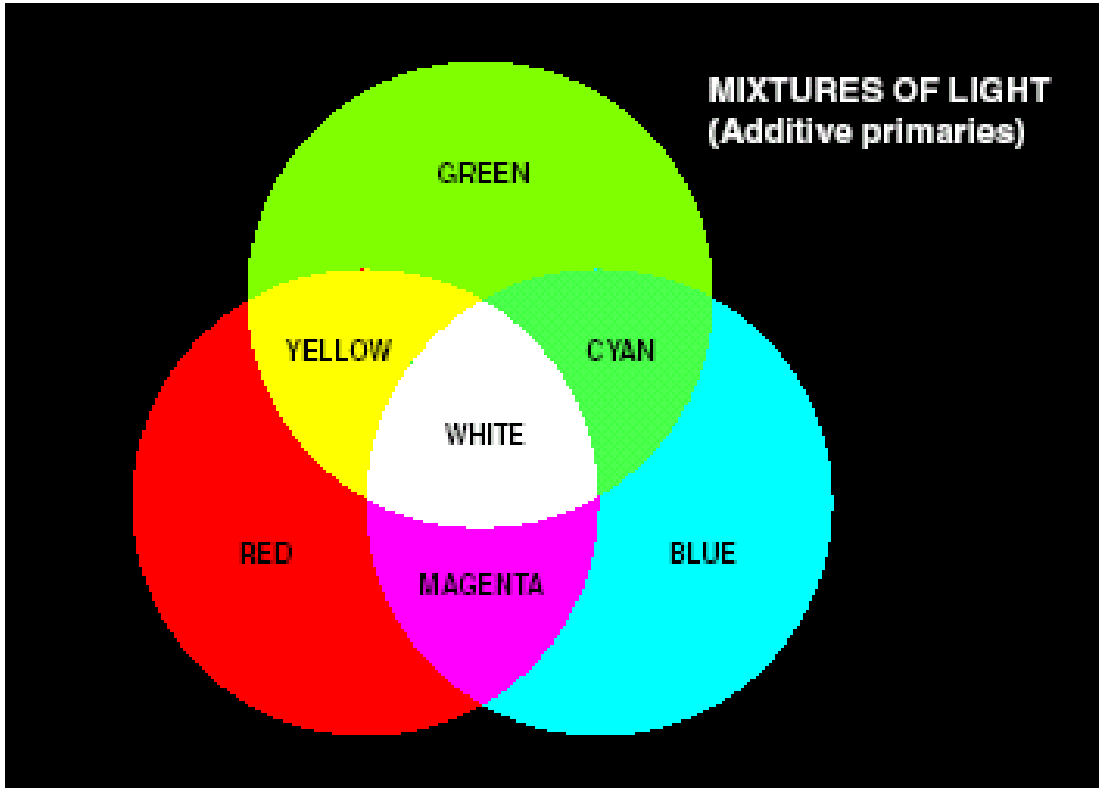
- Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (*R*), *green* (*G*), dan *blue* (*B*).

- Ketiga warna tersebut dinamakan **warna pokok** (*primary colors*), dan sering disingkat sebagai warna dasar *RGB.*

- Warna-warna lain dapat diperoleh dengan mencampurkan ketiga warna pokok (R, G, dan B) dengan perbandingan tertentu:

$$C = a\,R + b\,G + c\,B$$

- *Secondary colors*: G+B=Cyan, R+G=Yellow, R+B=Magenta


MIXTURES OF LIGHT
(Additive primaries)
GREEN
YELLOW
CYAN
WHITE
RED
MAGENTA
BLUE

Model Warna

- Istilah yang semakna: model warna (*color model*), ruang warna (*color space*), atau sistem warna (*color system*)
  - Menspesifikasikan warna dalam suatu cara yang baku
  - Sistem koordinat sehingga setiap warna disajikan dengan sebuah titik

- Model warna (atau ruang warna) yang digunakan di dalam pengolahan citra:
  1. Model warna RGB
  2. Model warna CMY dan CMYK
  3. Model YCbCr

  } Cocok untuk hardware atau aplikasi

  4. Model warna HSI — - Sesuai dengan deskripsi manusia
  5. Model warna XYZ — - Model warna fiktif, untuk proses antara

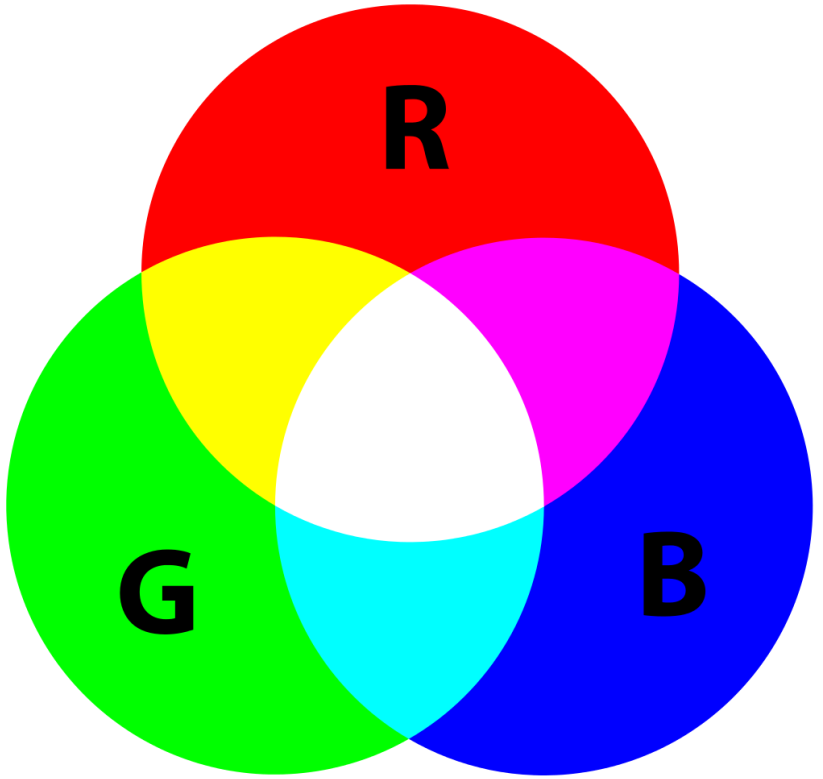
Model Warna RGB

- Model warna RGB digunakan untuk *display* pada layar komputer

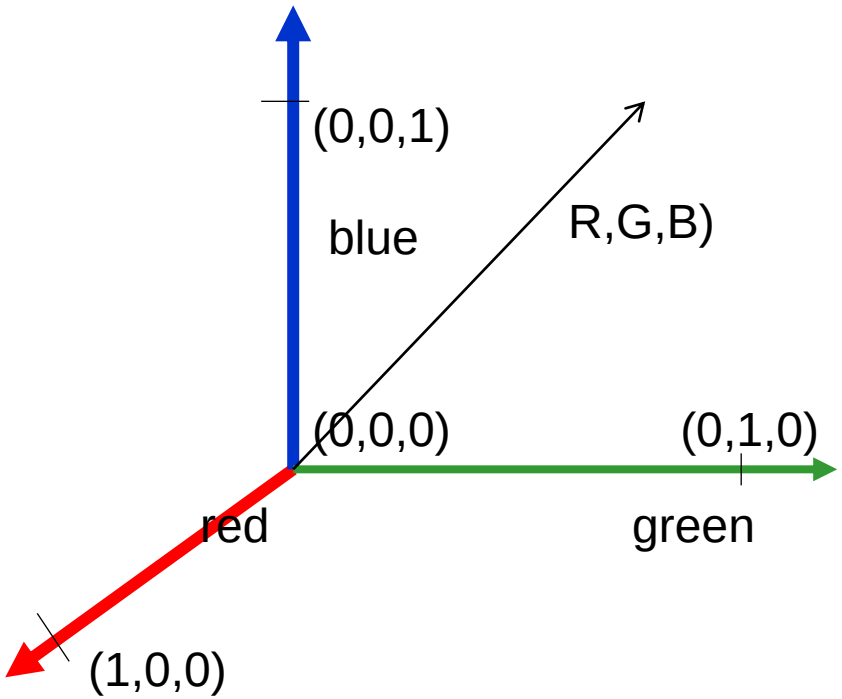
- *CIE* (*Commission International de l'Eclairage*) *atau International Lighting Committee* adalah lembaga yang membakukan warna pada tahun 1931.

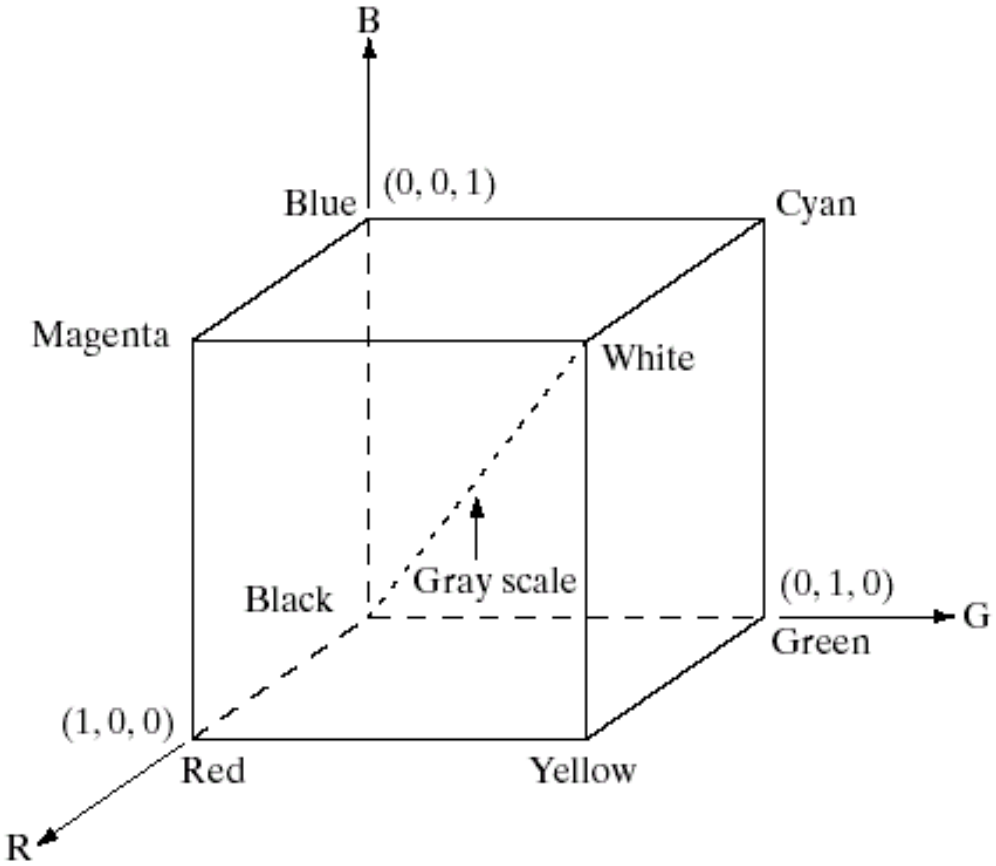
- *CIE* menstandarkan panjang gelombang warna-warna pokok sebagai berikut:

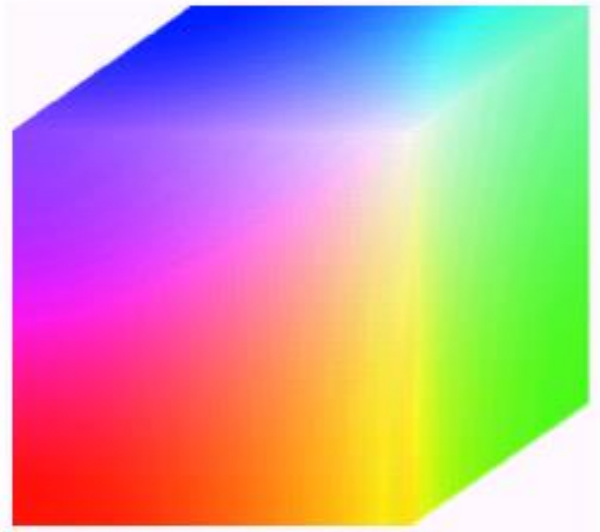
R	: 700 nm
G	: 546.1 nm
B	: 435.8 nm



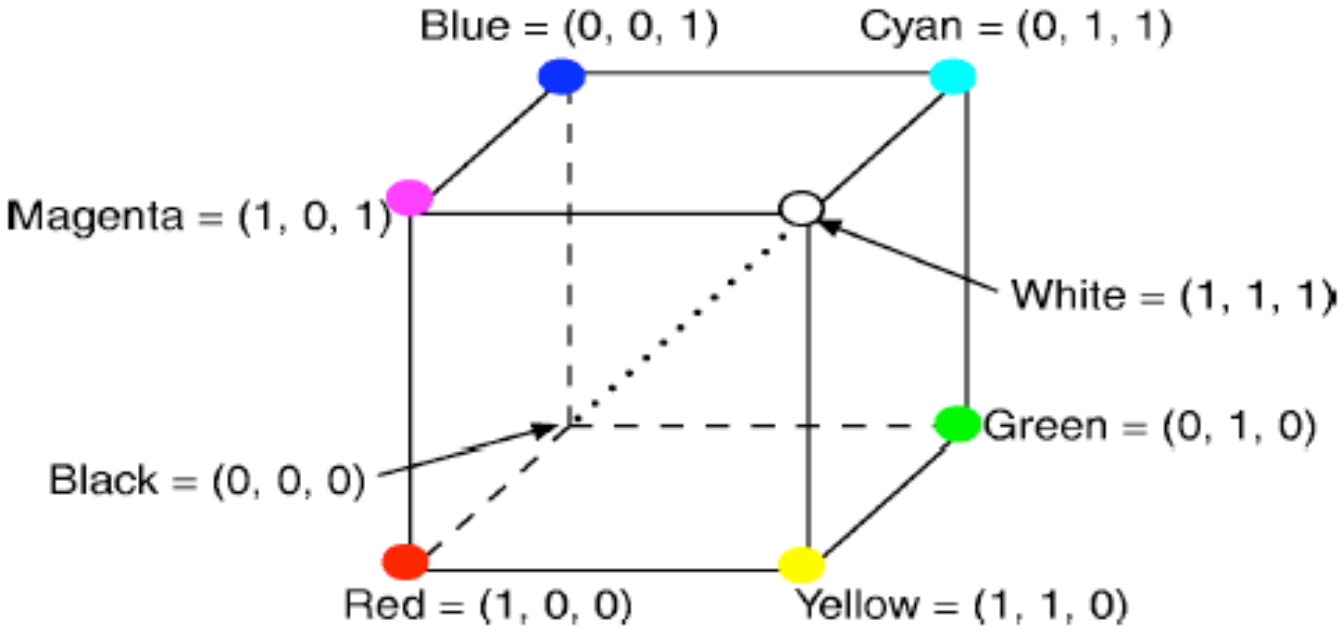
Ruang warna RGB







Catatan: (0, 0, 1) adalah nilai normalisasi dari (0, 0, 255) pada citra dengan kedalaman 24-bit

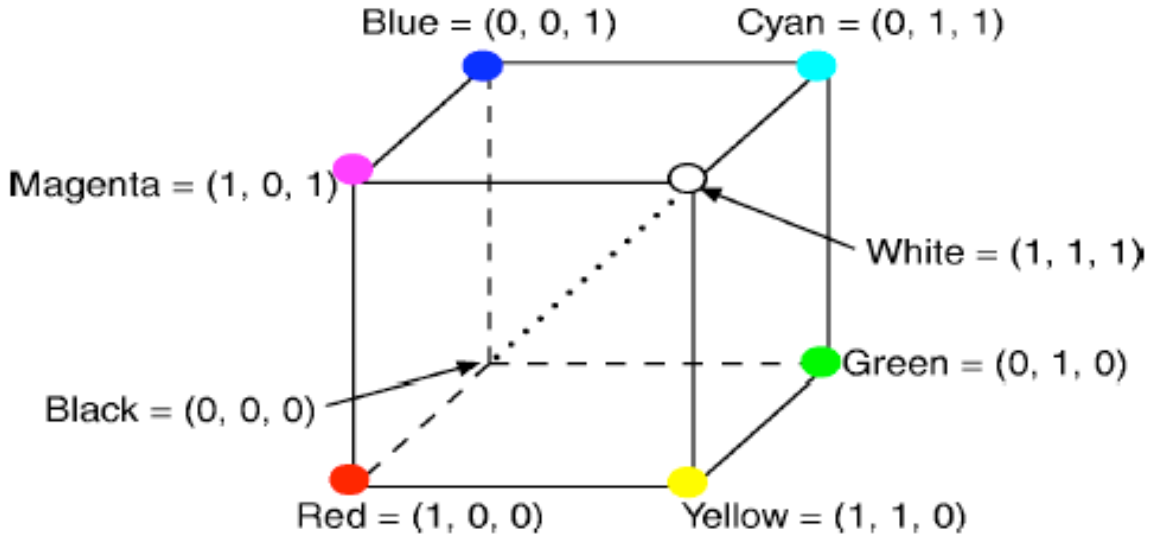


Warna-warna di dalam model RGB bersifat *additive:*

- $C = a\,R + b\,G + c\,B$
- G+B=Cyan, R+G=Yellow, R+B=Magenta
- Diagonal utama => gray levels
- Hitam adalah (0, 0, 0)
- Putih adalah (1, 1, 1)

Rentang warna (*gamut*) model RGB didefinisikan dengan rumus:

$$C = a\boldsymbol{R} + b\boldsymbol{G} + c\boldsymbol{B}$$



Nilai-nilai R, G, dan B, dalam berbagai sistem (dalam koordinat (x,y) pada model XYZ):

RGB	NTSC	CIE	Monitor
R	(0.67, 0.33)	(0.73, 0.26)	(0.62, 0.34)
G	(0.21, 0.71)	(0.27, 0.71)	(0.26, 0.59)
B	(0.14, 0.08)	(0.16, 0.01)	(0.15, 0.07)

Catatan:

- Format *NTSC* digunakan pada televisi di Amerika Serikat
- *National Television Systems Committee* (*NTSC*) menggunakan model warna *RGB* khusus untuk menampilkan citra berwarna pada layar *CRT*..

=

Red

Green

Blue

Red

148	162	175	182	189	194	195	193	195	195	197
148	164	174	176	185	189	191	191	196	194	195
144	159	167	176	178	185	188	191	196	194	197
128	147	157	168	173	179	182	184	191	191	192
119	134	148	160	164	170	179	176	181	189	185
145	124	142	151	160	168	169	174	180	182	183
172	120	140	153	157	169	171	178	180	182	182
196	120	129	144	152	158	167	170	177	176	178
204	144	116	134	142	149	155	165	165	170	171

Green

42	43	48	50	53	56	56	53	54	54	54
50	49	51	47	53	55	56	55	59	55	54
51	48	47	49	49	51	50	52	54	51	54
53	48	45	49	50	52	50	48	51	50	50
59	43	43	48	47	48	54	47	49	55	50
100	42	41	42	44	46	45	46	50	52	50
142	47	43	42	39	46	44	48	49	51	49
185	65	44	42	42	43	48	46	50	48	49
209	106	44	42	41	42	44	50	48	50	49

Blue

16	24	32	35	37	40	40	37	37	38	36
19	25	31	28	34	37	38	37	40	35	33
17	23	27	33	32	35	33	36	39	35	37
20	19	23	31	33	34	34	32	36	35	35
29	16	24	33	32	34	39	30	31	38	34
71	11	18	24	30	33	30	30	34	36	34
113	14	16	21	24	32	30	32	33	35	33
156	32	13	20	25	28	33	31	35	33	32
177	72	9	16	22	26	30	35	32	33	32

Kode program Matlab untuk dekomposisi R, G, B dari citra berwarna

```
function fig = imColorSep(A)
% IMCOLORSEP Displays the RGB decomposition of a full-color image
%
% Syntax: fig = imColorSep(A);
%
% Example: A = imread('peppers.png');
% fig = imColorSep(A);
%
%
% Written by: Rick Rosson, 2007 December 26
%
% Revised: Rick Rosson, 2008 January 2
%
% Copyright (c) 2007-08 Richard D. Rosson. All rights reserved.
%
%
% Number of gray scale values:
N = 256;
% Make sure data type of image array is 'uint8':
A = im2uint8(A);
% Create figure window:
fig = figure;
% Display full color image:
subplot(2,2,1);
imshow(A);
title('Full Color');

% Cell array of color names:
ColorList = { 'Red' 'Green' 'Blue' };

% Gray-scale column vector: % range [ 0 .. 1 ]
gr = 0:1/(N-1):1; % increment 1/(N-1)

% Display each of the three color components:
for k = 1:3
   % color map:
  cMap = zeros(N,3);
  cMap(:,k) = gr;

  % Display monochromatic image:
  subplot(2,2,k+1);
  imshow(ind2rgb(A(:,:,k),cMap));
  title(ColorList{k});
 end
end
```

Sumber: https://www.mathworks.com/matlabcentral/fileexchange/18125-rgb-image-decomposition

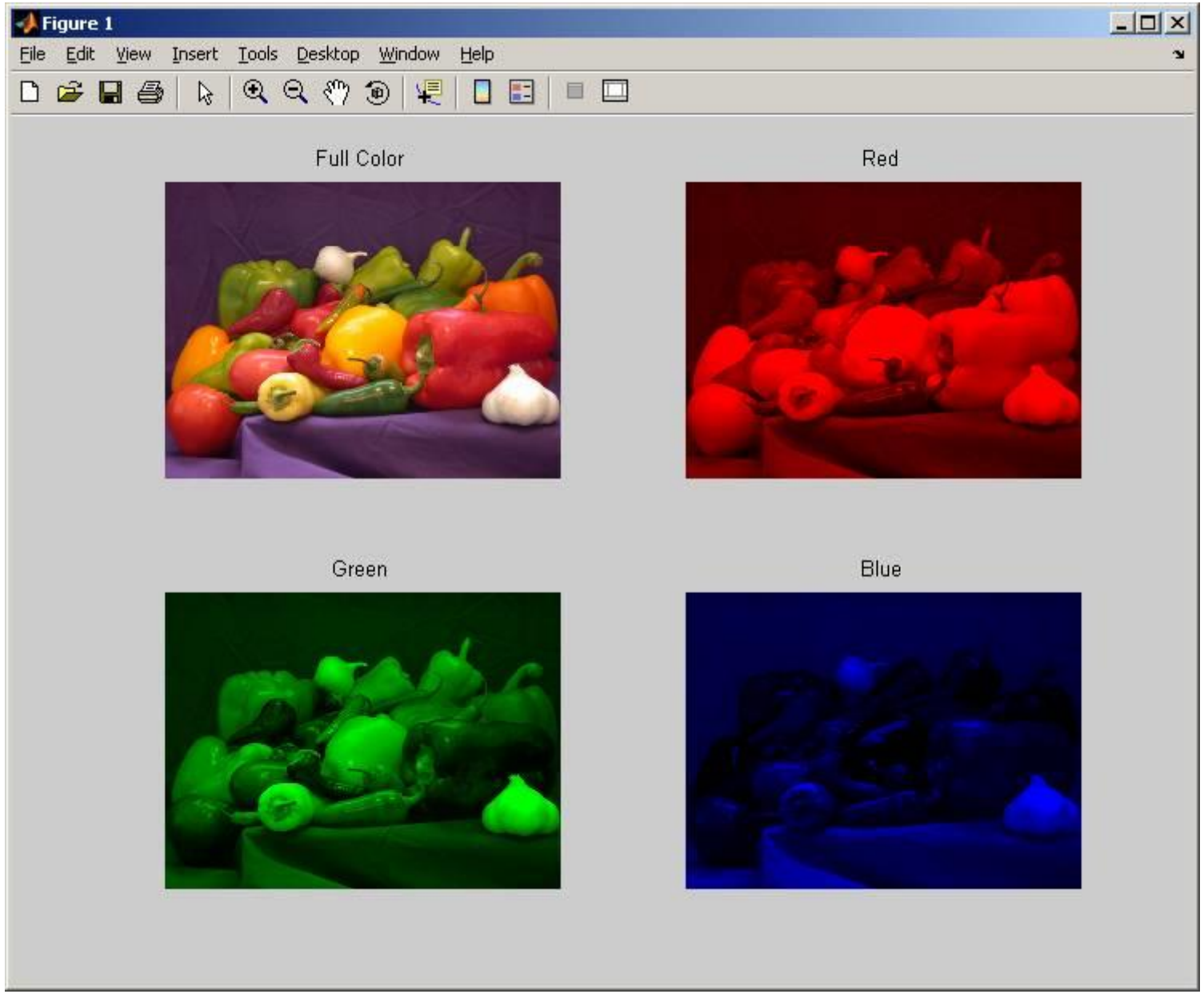


Figure 1
File Edit View Insert Tools Desktop Window Help
Full Color
Red
Green
Blue

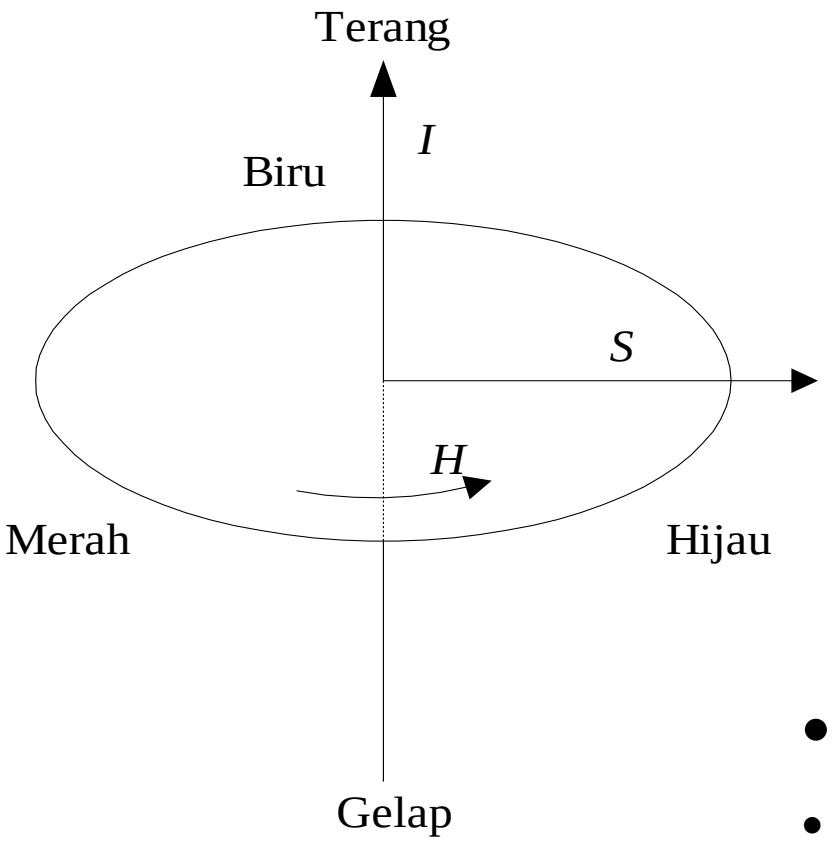
Model warna HSI

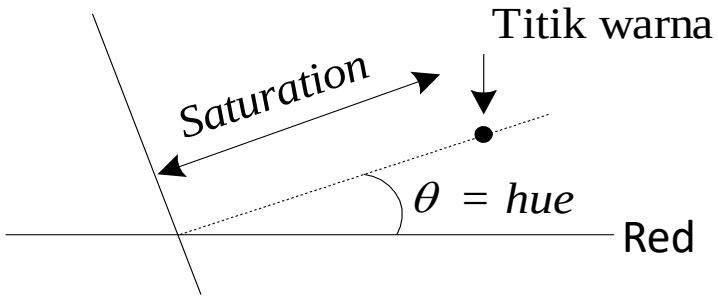
- Manusia tidak lazim mendeskripsikan warna dalam komponen R, G, dan B.

- Contoh: apakah anda akan mendeskripsikan warna ungu dalam persentase masing-masing R, G, dan B?

- Manusia mendekripsikan warna dengan atribut *hue* (*H*), dan *saturation* (*S*), dan *intensity* (*I*), sehingga dinamakan model HSI

Model HSI





- *Hue* dikodekan sebagai sudut (0 sampai 2π).
- *Saturation* adalah jarak ke sumbu vertikal (0 sampai 1).
- Intensity adalah tinggi sepanjang garis vertikal (0 sampai 1).

1. *Hue*

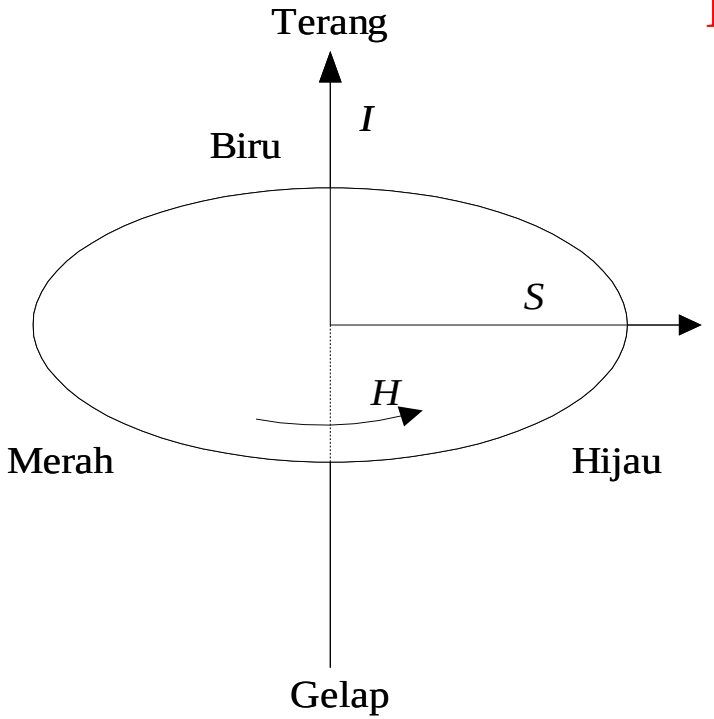
- Atribut yang menyatakan warna sebenarnya, seperti merah, violet, dan kuning.
- *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dsb, dari cahaya.
- *Hue* berasosiasi dengan panjang gelombang cahaya, dan bila kita menyebut warna merah, violet, atau kuning, kita sebenarnya menspesifikasikan *hue*-nya.
- *Hue* dikuantisasi dengan nilai dari 0 sampai 2π; 0 menyatakan merah, lalu memutar nilai-nilai spektrum tersebut kembali lagi ke 0 untuk menyatakan merah lagi.

2. Saturation

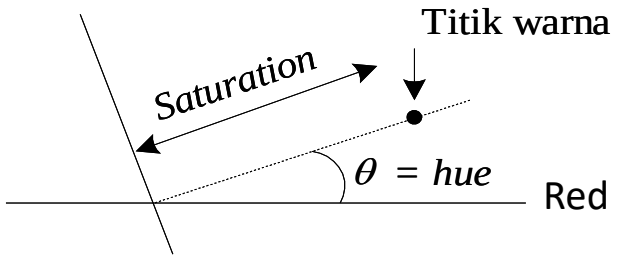
- Menyatakan tingkat kemurnian warna cahaya, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna.
- Sebagai contoh, warna merah adalah 100% warna jenuh (*saturated color*), sedangkan warna *pink* adalah warna merah dengan tingkat kejenuhan sangat rendah (karena ada warna putih di dalamnya).
- Jadi, jika *hue* menyatakan warna sebenarnya, maka *saturation* menyatakan seberapa dalam warna tersebut.
- Jika suatu warna mempunyai *saturation* = 0, maka warna tersebut tanpa *hue*, yaitu dibuat dari warna putih saja.
- Jika *saturation* = 1, maka tidak ada warna putih yang ditambahkan pada warna tersebut (*primary colors*)

3. *Intensity/brigthness/luminance*

- Atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa mempedulikan warna.
- Kisaran nilainya adalah antara gelap (hitam = 0) dan terang (putih = 1)

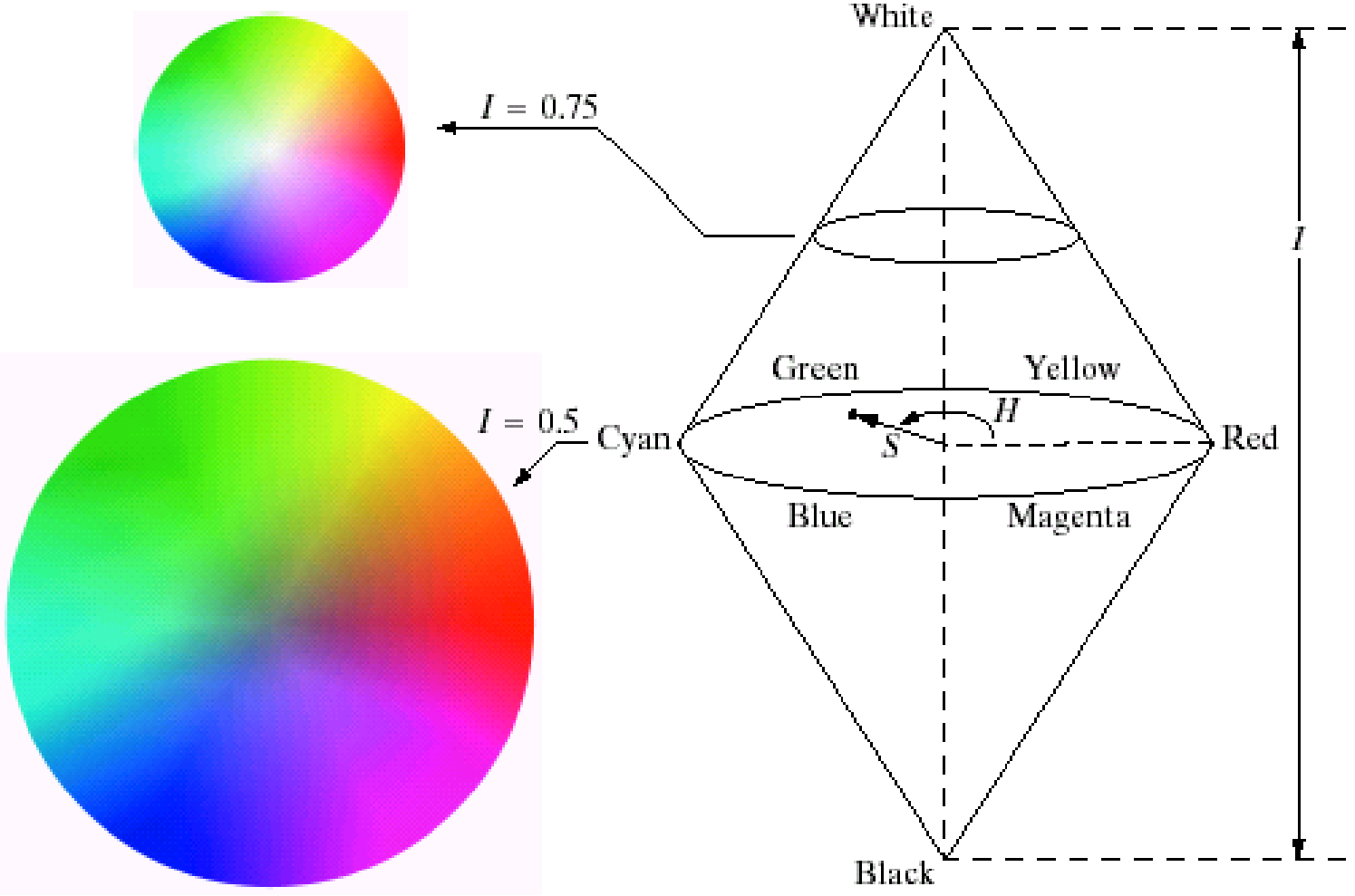


Model HSI



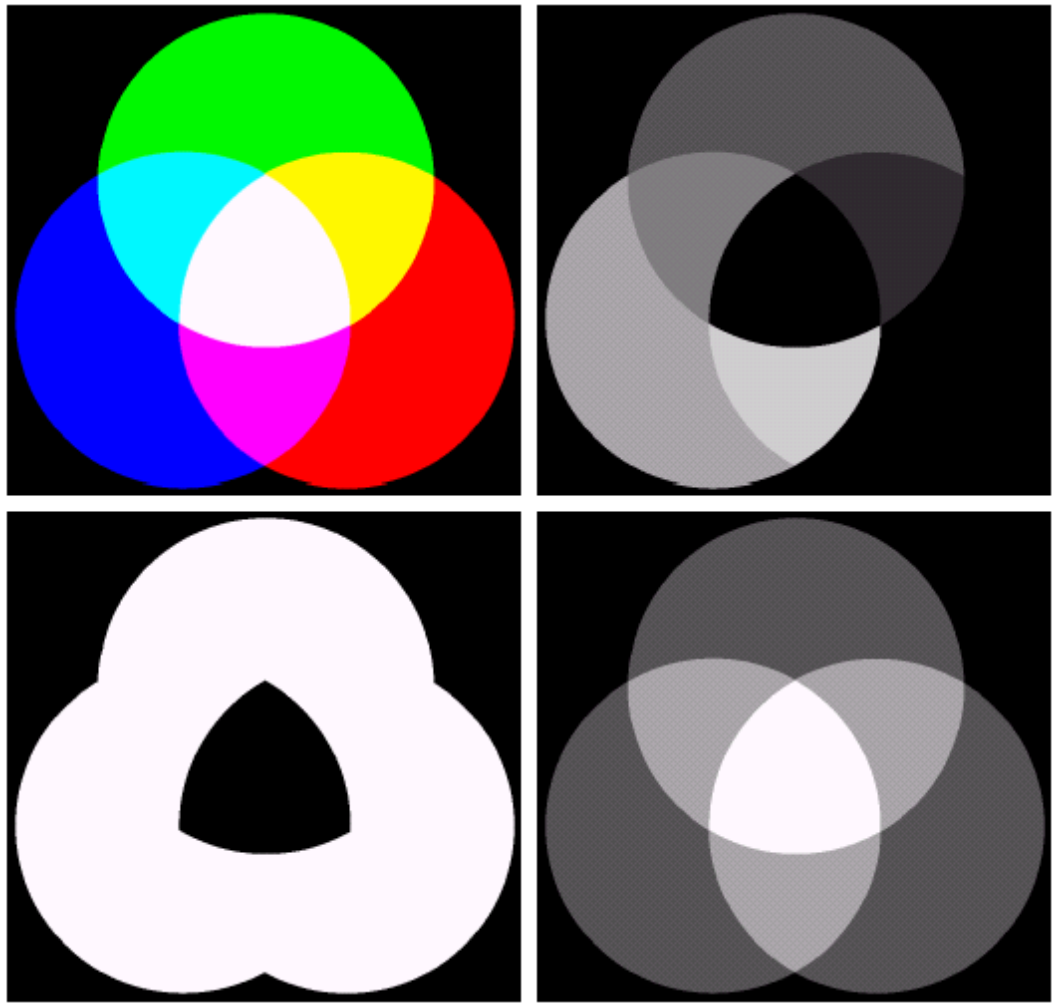
- *Hue* dikodekan sebagai sudut (0 sampai 2π).
- *Saturation*: jarak titik warna ke sumbu vertikal (0 sampai 1).
- Intensity: tinggi sepanjang garis vertikal (0 sampai 1).

Model HSI



Citra-citra yang menggambarkan komponen HSI:

R,G,B



Hue

saturation

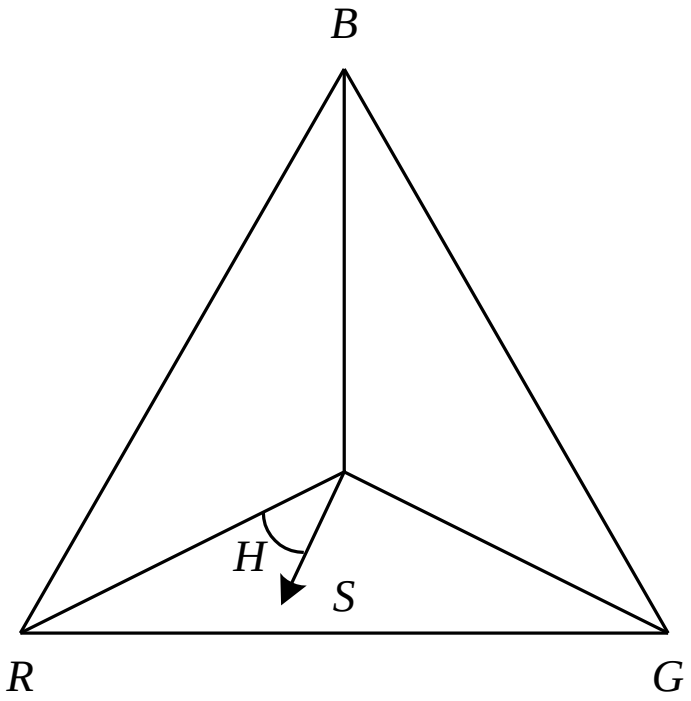
intensity

Transformasi Warna dari RGB ke HSI

- Meskipun basis *RGB* bagus untuk menampilkan informasi warna, tetapi ia tidak cocok untuk beberapa aplikasi pemrosesan citra.

- Misalnya pada aplikasi pengenalan objek, lebih mudah mengidentifikasi objek dengan perbedaan *hue*-nya dengan cara memberikan nilai ambang pada rentang nilai-nilai *hue* (panjang helombang spektrum) yang melingkupi objek.

- Pada aplikasi pemampatan citra, pemampatan citra berwarna lebih relevan bila warna *RGB*-nya dikonversikan ke *HSI* 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.



Segitiga RGB

- Segitiga menghubungkan tiga warna pokok *red*, *green*, *blue* .
- Titik-titik pada segitiga menyatakan warna yang dihasilkan dari pencampuran warna titik sudut.
- 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

$$H = \begin{cases} \theta & jika\ G \geq B \\ 360° - \theta & jika\ B > G \end{cases}$$

yang dalam hal ini $\theta = \cos^{-1} \dfrac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}}$

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

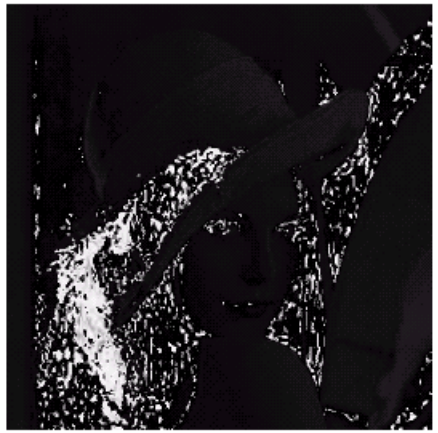
$$I = \frac{1}{3}(R+G+B)$$

original

R

G

B



H

S

I

Sumber: Jen-Chang Liu, Spring 2006
Color Image Processing

Transformasi warna dari HSI ke RGB

- Tergantung pada nilai H

1. Jika $0° \leq H < 120°$

$$B = I(1 - S)$$

$$R = I\left[1 + \frac{S \cos H}{\cos(60° - H)}\right]$$

$$G = 3I - (R + B)$$

2. Jika $120° \leq H < 240°$

Hitung $H = H - 120°$

$$R = I(1 - S)$$

$$G = I\left[1 + \frac{S \cos H}{\cos(60° - H)}\right]$$

$$B = 3I - (R + B)$$

3. Jika $240° \leq H \leq 360°$

Hitung $H = H - 240°$

$$G = I(1 - S)$$

$$B = I\left[1 + \frac{S \cos H}{\cos(60° - H)}\right]$$

$$R = 3I - (R + B)$$

```
%Fungsi transformasi warna citra dari model RGB menjadi model HSI
function f = rgb2hsi(J)
    Jred = double(J(:,:,1));
    Jgreen = double(J(:,:,2));
    Jblue = double(J(:,:,3));
    [M, N] = size(Jred);

    for i=1:M
        for j=1:N

            % Normalisasi nilai-nilai R, G, dan B ke dalam selang[0, 1]
            r = Jred(i,j)/255;
            g = Jgreen(i,j)/255;
            b = Jblue(i,j)/255;

            p = (r - g) + (r - b);
            q = (r-g)*(r - g) + (r - b)*(g - b);
            t = sqrt(double(q));
            tetha = acos(double(0.5*(p/t))) * 180/pi;
```

```
                if g >= b
                h = tetha;
            else
                h = 360 - tetha;
            end;

            minRGB = min(r, min(g, b));

            s = 1 - (3 * minRGB)/(r + g + b);

            i2 = (r + g + b)/3;

            h = double(h/360);
            s = double(s);
            i2 = double(i2);

            H(i,j) = h;
            S(i,j) = s;
            I(i,j) = i2;
        end
    end
    figure, imshow(H), title('H');
    figure, imshow(S), title('S');
    figure,imshow(I), title('I');

    Jhsi(:, :, 1) = H;
    Jhsi(:, :, 2) = S;
    Jhsi(:, :, 3) = I;

    f = Jhsi;
end
```

- Contoh pemanggilan fungsi rgb2hsi:

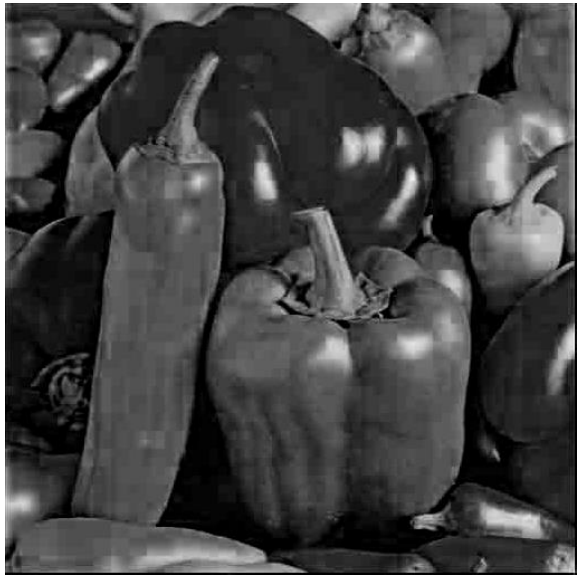
```
J = imread('peppers512.jpg');
R = J(:,:,1);
G = J(:,:,2);
B = J(:,:,3);
imshow(R), title('R');
figure, imshow(G), title('G');
figure, imshow(B), title('B');

Jhsi = rgb2hsi(J);
```

- Untuk fungsi transformasi ari HSI ke RGB, silakan dibuat sendiri

R

G

B



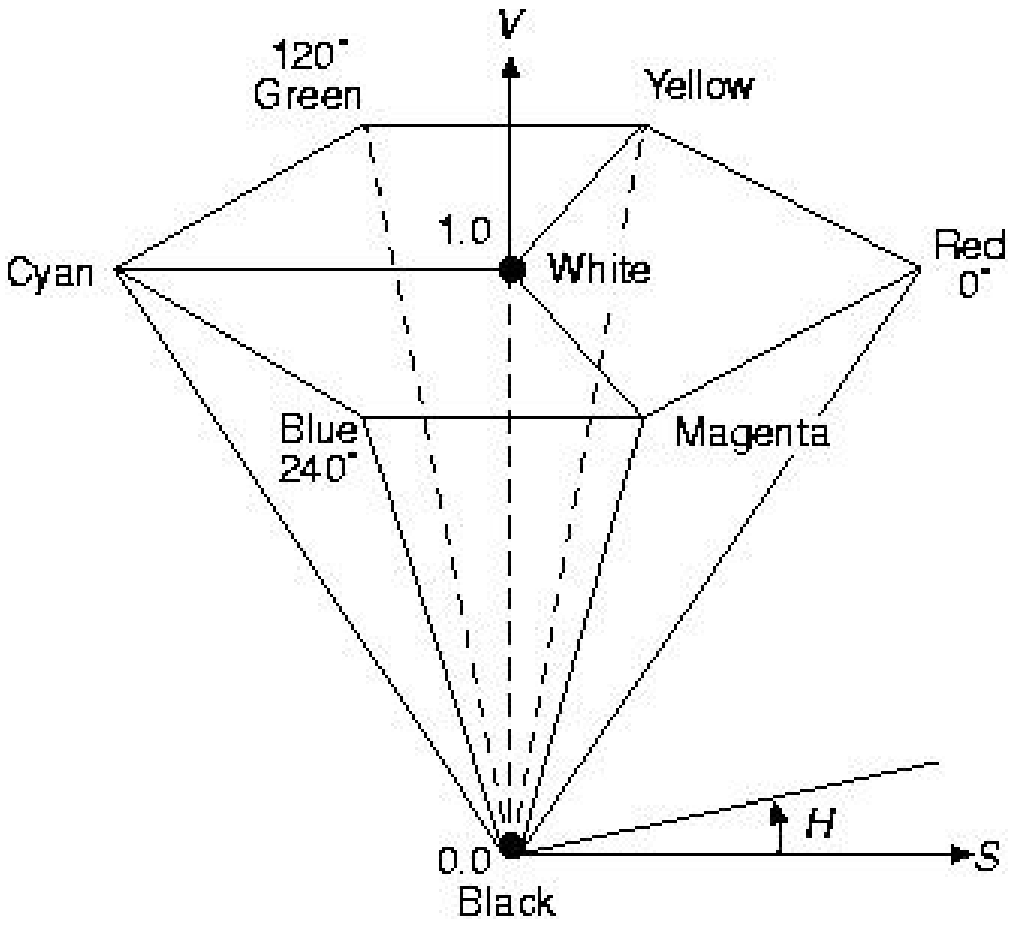
H

S

I

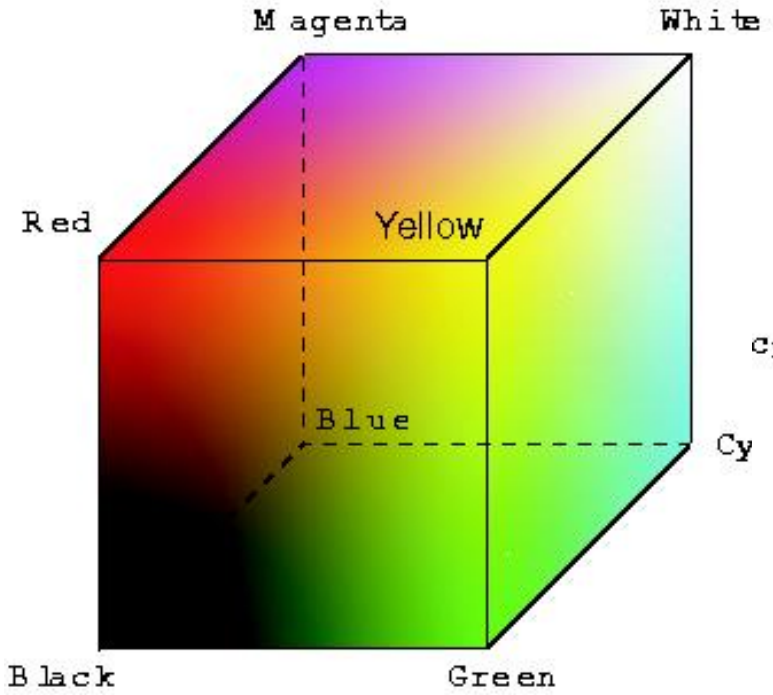
Model warna HSV

- Mirip dengan HSI.
- H = *Hue* = jenis warna sebenarnya (merah, ungu, dll), rentang nilai 0 sampai 2π
- S = *Saturation* = kemurnian warna, rentang nilai [0,1]
- V = Value = kecerahan (*brightness*) sebuah warna, rentang nilai [0, 1]

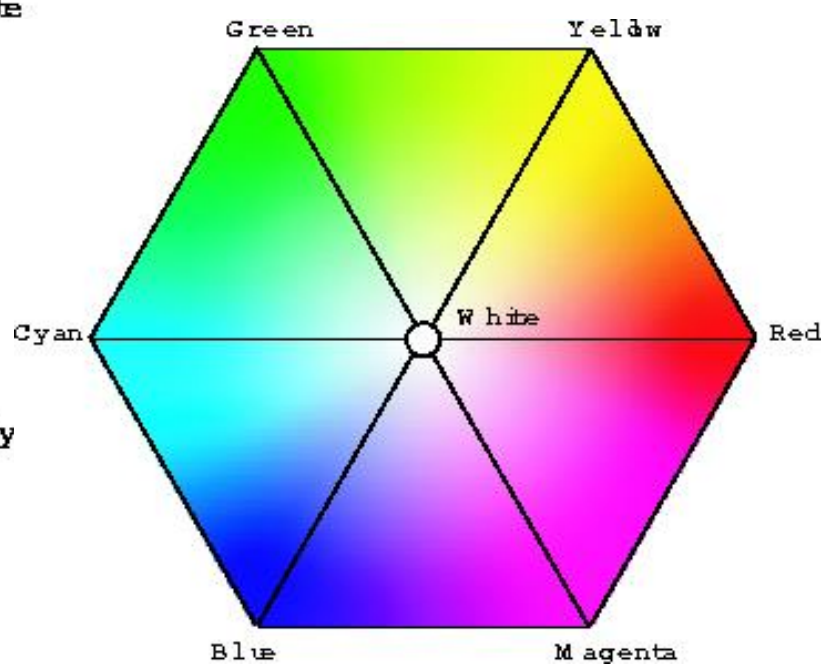


Sumber: Donald House, Color Principles for Computer Graphics

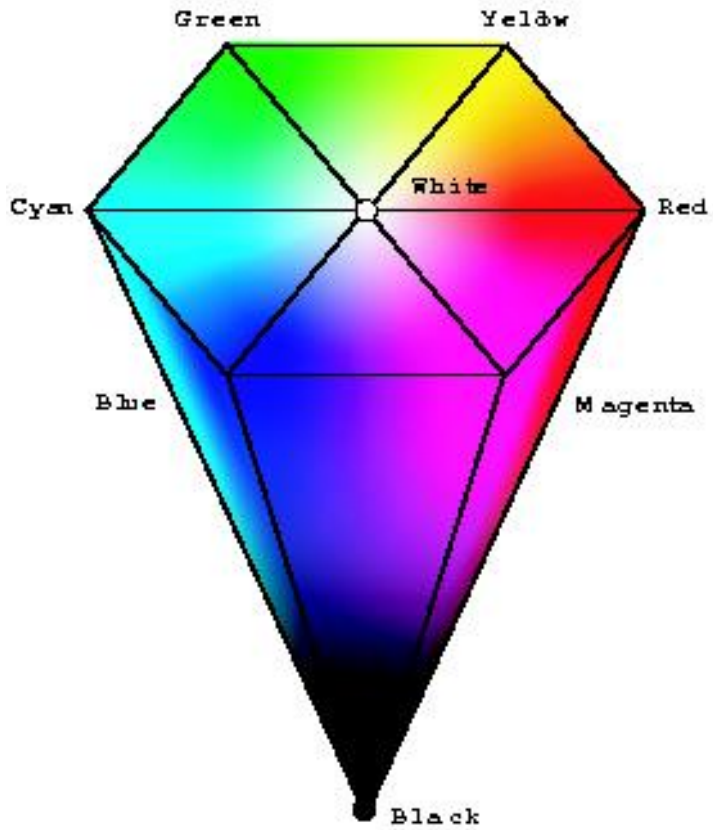
HSV Color Model



RGB cube



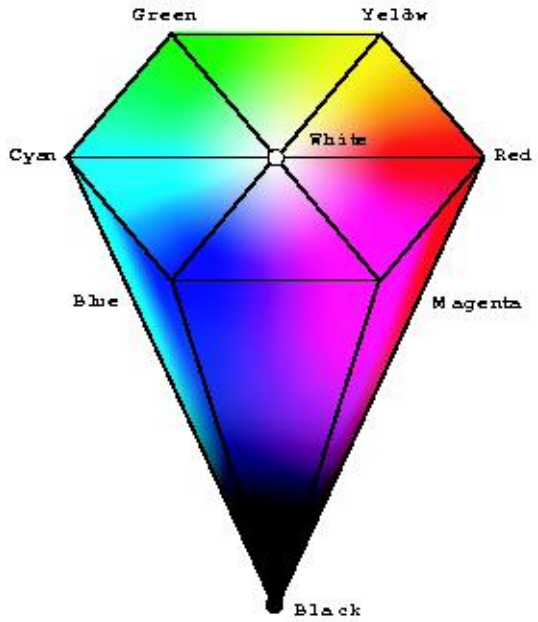
HSV top view

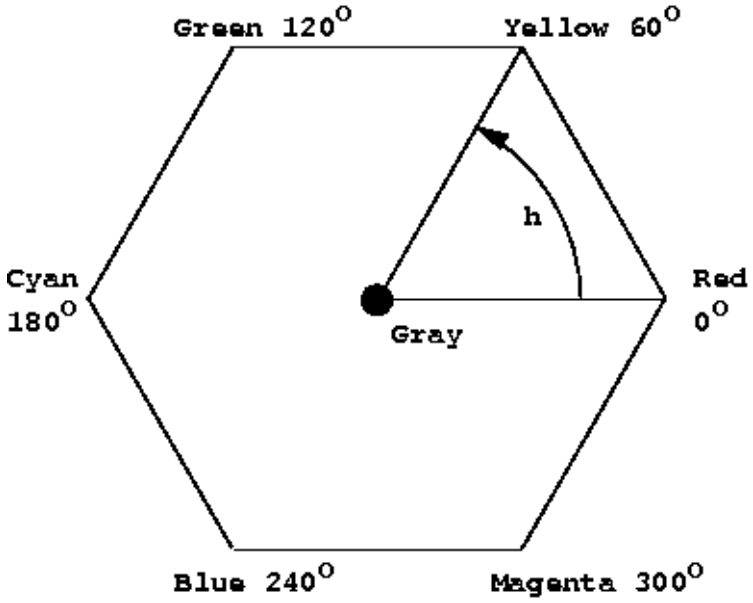


HSV cone

Sumber: Donald House, Color Principles for Computer Graphics

HSV Color Model

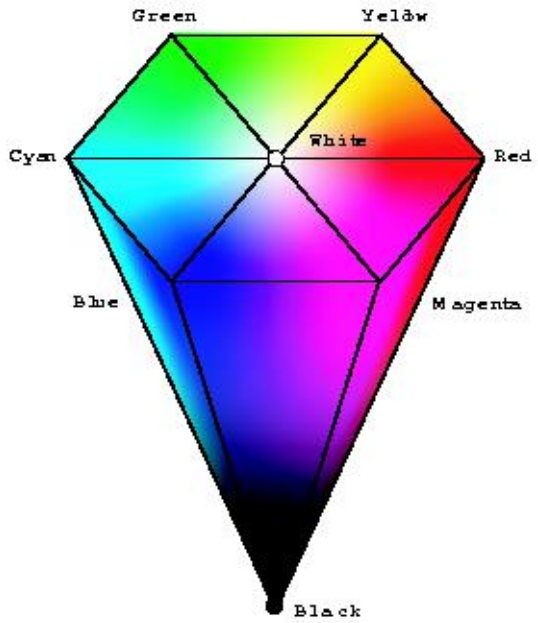


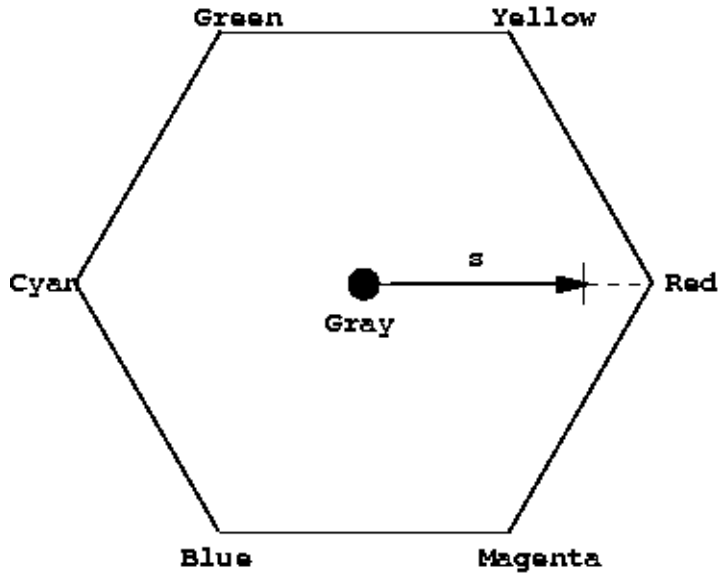


Hue, an angular measure (0 … 360)

Sumber: Donald House, Color Principles for Computer Graphics

HSV Color Model

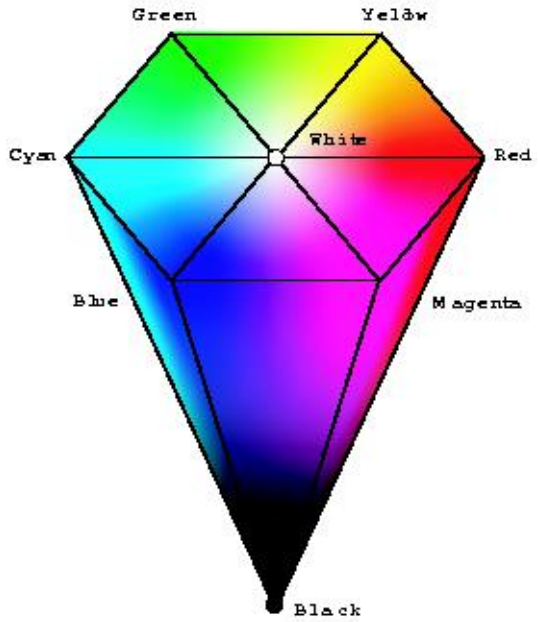


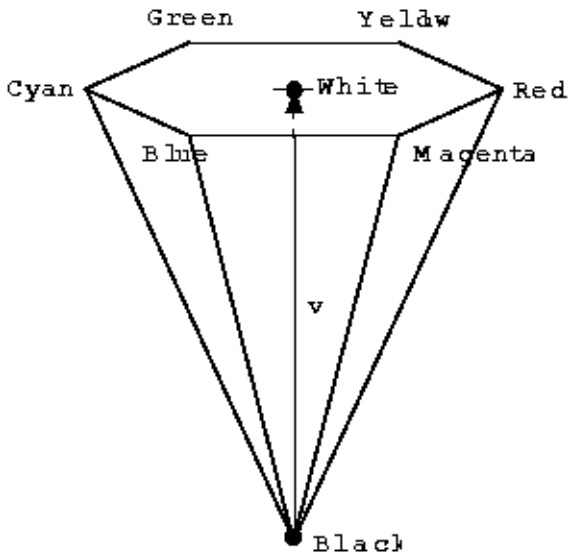


Saturation, a fractional measure (0.0 … 1.0)

Sumber: Donald House, Color Principles for Computer Graphics

HSV Color Model





Value, a fractional measure (0.0 … 1.0)

- Transformasi warna dari RGB ke HSV:

$$V = \max(R, G, B)$$

$$V_m = V - \min(R, G, B)$$

$$S = \begin{cases} 0 & , V = 0 \\ \dfrac{V_m}{V} & , V > 0 \end{cases}$$

$$H = \begin{cases} 0 & , S = 0 \\ 60° \cdot \left(\dfrac{G - B}{V_m}\right) mod\ 6 & , V = R \\ 60° \cdot \left(2 + \dfrac{B - R}{V_m}\right) & , V = G \\ 60° \cdot \left(4 + \dfrac{R - G}{V_m}\right) & , V = B \end{cases}$$

- Transformasi warna dari HSV ke RGB:

$$K = H/60^\circ$$

$$T = \mathrm{H}/60^\circ - K$$

$$X = V(1 - S),\ Y = \mathrm{V}(1 - ST),\ \ \mathrm{Z} = V(1 - S)(1 - T)$$

$$R, G, B = \begin{cases} V, Z, X & , K = 0 \\ Y, V, X & , K = 1 \\ X, V, Z & , K = 2 \\ X, Y, V & , K = 3 \\ Z, X, V & , K = 4 \\ V, X, Y & , K = 5 \end{cases}$$

Konversi RGB ke HSV

```
rgb = imread('lena.bmp');
hsv = rgb2hsv(rgb);
H = hsv(:,:,1)
imshow(H)
S = hsv(:,:,2);
figure, imshow(S);
V = hsv(:,:,3);
figure, imshow(V);
```

H

S

V

```
>> H(1:5,1:5)

ans =

    0.0198    0.0182    0.0075    0.0140    0.0283
    0.0198    0.0182    0.0093    0.0140    0.0283
    0.0198    0.0182    0.0093    0.0140    0.0283
    0.0198    0.0182    0.0093    0.0140    0.0283
    0.0198    0.0182    0.0093    0.0140    0.0283
```

```
>> S(1:5,1:5)

ans =

    0.4469    0.4469    0.3991    0.4260    0.4690
    0.4469    0.4469    0.4036    0.4260    0.4690
    0.4469    0.4469    0.4036    0.4260    0.4690
    0.4469    0.4469    0.4036    0.4260    0.4690
    0.4469    0.4469    0.4036    0.4260    0.4690
```

```
>> V(1:5,1:5)

ans =

    0.8863    0.8863    0.8745    0.8745    0.8863
    0.8863    0.8863    0.8745    0.8745    0.8863
    0.8863    0.8863    0.8745    0.8745    0.8863
    0.8863    0.8863    0.8745    0.8745    0.8863
    0.8863    0.8863    0.8745    0.8745    0.8863
```

Konversi RGB ke HSV

```
rgb = imread('lena.bmp');
hsv = rgb2hsv(rgb);     % Konversi RGB ke HSV
H = hsv(:,:,1); imshow(H), title ('H');
S = hsv(:,:,2); figure,imshow(S) , title ('S');
V = hsv(:,:,3); figure,imshow(V), title('V');
hsv2 = cat(3, H, S, V);
rgb2 = hsv2rgb(hsv2);  % Konversi HSV ke RGB
figure, imshow(rgb2), title ('Original image');
```

Original image

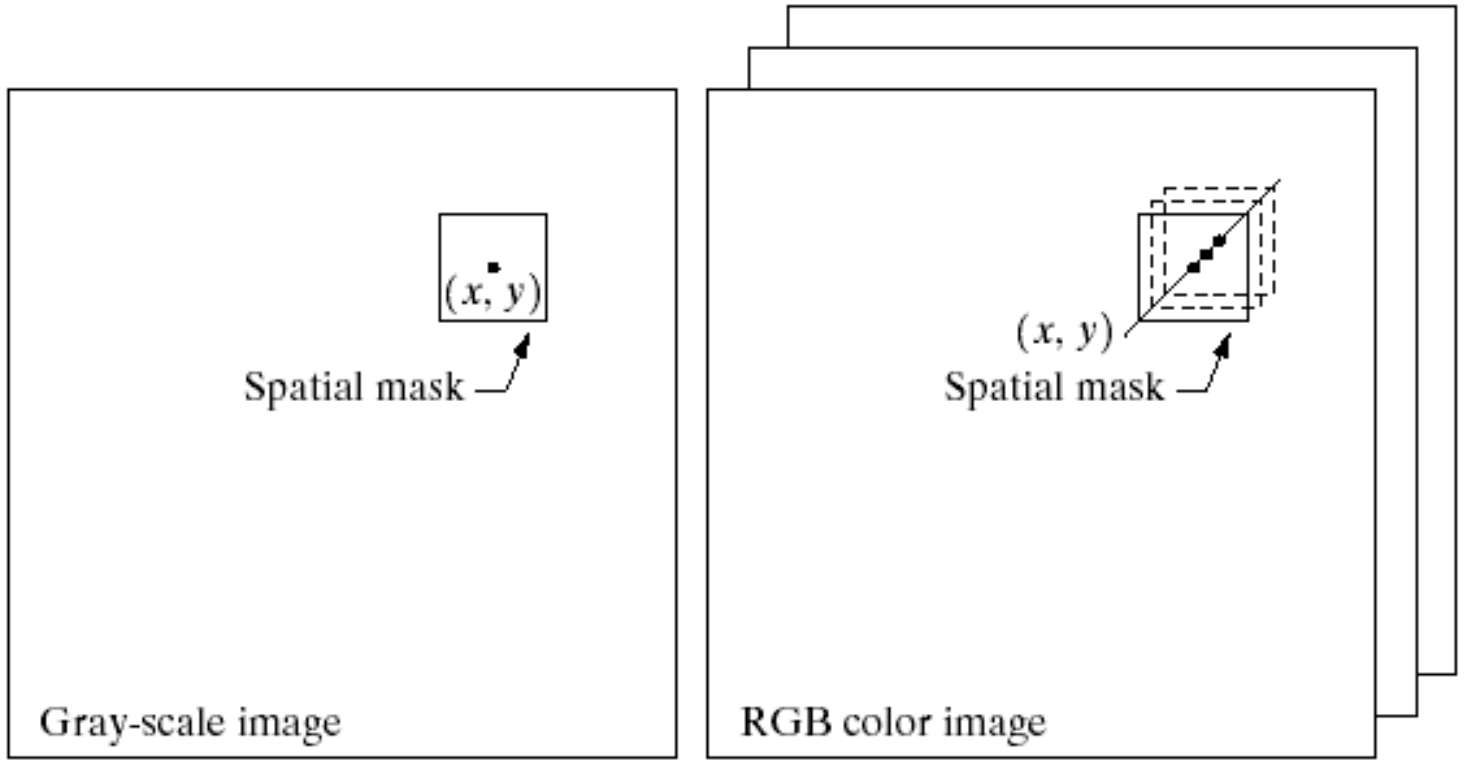
H

S

V

Pelembutan (*smoothing*) citra berwarna

- Neighborhood processing



Pelembutan citra berwarna: penapis rerata

$$\bar{\mathbf{c}}(x,y) = \frac{1}{K} \sum_{(x,y)\in S_{xy}} \mathbf{c}(x,y)$$

vector processing

Neighborhood
Centered at (x,y)

$$\bar{\mathbf{c}}(x,y) = \begin{bmatrix} \frac{1}{K} \sum_{(x,y)\in S_{xy}} R(x,y) \\ \frac{1}{K} \sum_{(x,y)\in S_{xy}} G(x,y) \\ \frac{1}{K} \sum_{(x,y)\in S_{xy}} B(x,y) \end{bmatrix}$$

per-component processing

Sumber: Jen-Chang Liu, Spring 2006
Color Image Processing

Sumber: Jen-Chang Liu, Spring 2006
Color Image Processing

Contoh: penapis rerata 5x5

RGB model

Smooth I
in HSI model

difference

a b c

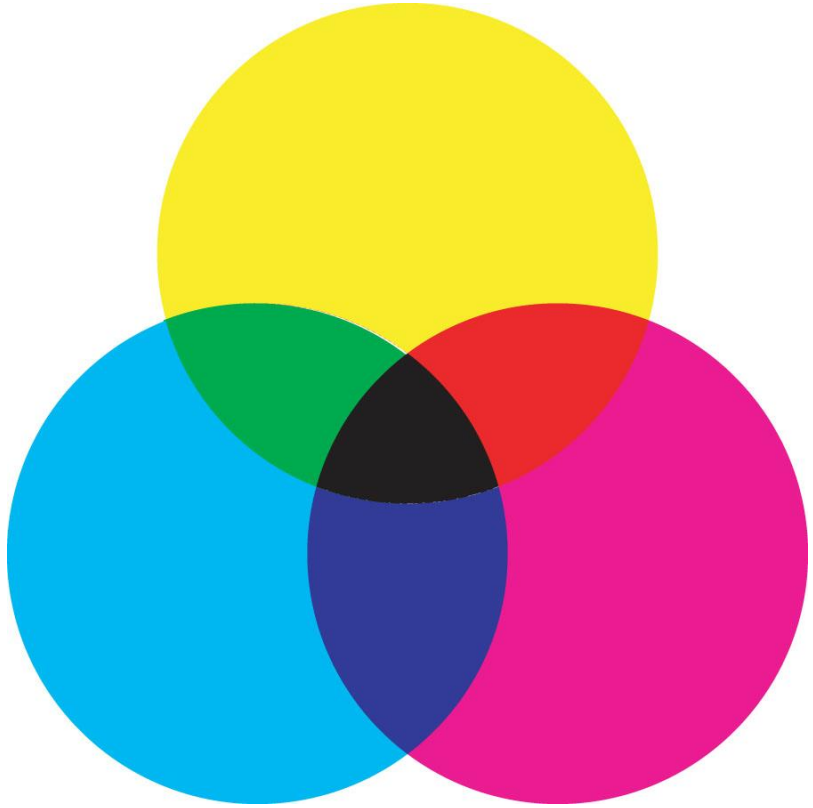
FIGURE 6.40 Image smoothing with a 5 × 5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

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 dari *green*



- Model warna CMY digunakan pada *printer* yang mencetak *hardcopy* citra berwarna.

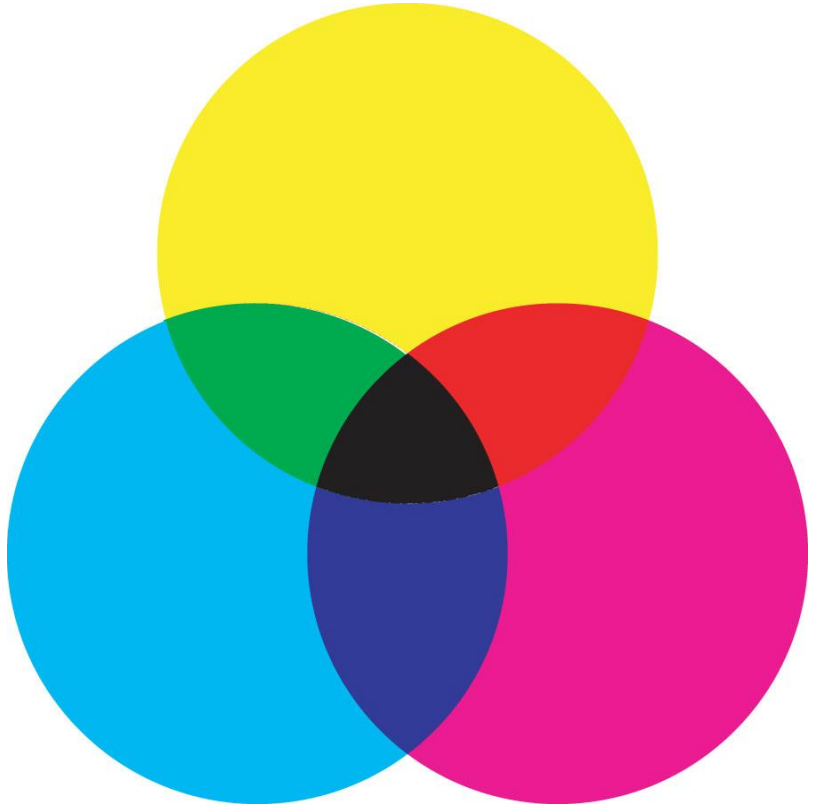
- CMY = White – RGB:

  C = 1 – R,

  M = 1 – G,

  Y = 1 – B

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



- CMY adalah model warna *substractive*

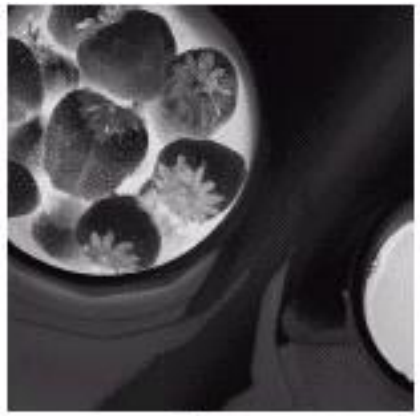
- Karena ketidaksempurnaan tinta, model *CMY* tidak dapat menghasilkan warna hitam dengan baik.

- Karena itu, model *CMY* disempurnakan menjadi model *CMYK*, yang dalam hal ini *K* menyatakan warna keempat (hitam), dengan rumus:

$$K = \min(C, M, Y)$$
$$C = C - K,$$
$$M = M - K,$$
$$Y = Y - K$$

```
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. Ketga mtariks 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];
    }
}
```

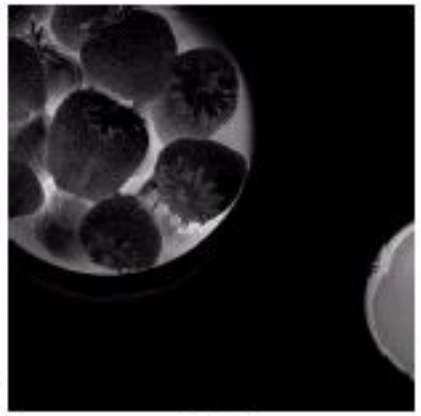
```
int CMYKtoRGB(citra c, citra y, citra m, citra k,
              citra r, citra g, citra b,
              int N, int M)

/* 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. Ketga mtariks 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];
    }
}
```



Cyan

Magenta

Yellow

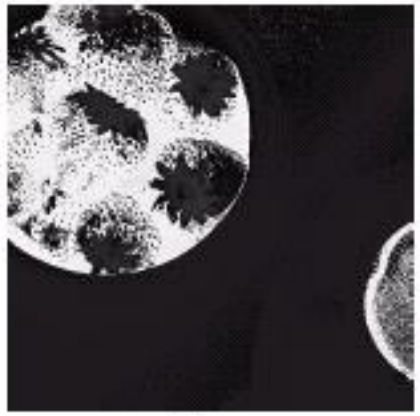


Black

Red

Green

Blue



Hue

Saturation

Intensity

Full color

Example: modify intensity of a color image

- **Example:** $g(x,y)=k\,f(x,y)$, $0<k<1$
- Components of color: input: r_1, r_2, r_3 ; output: s_1, s_2, s_3
- HIS/HSV color space (r_1 = H, r_2 = S, r_3 = I or V)
  - Intensity: $s_3 = k\,r_3$ where $s_1 = r_1$ and $s_2 = r_2$
  - Note: transform to HSI requires significant operations
- RGB color space (r_1 = R, r_2 = G, r_3 = B)
  - For each R,G,B component: $s_i = k\,r_i$
- CMY color space (r_1 = C, r_2 = M, r_3 = Y)
  - For each C,M,Y component: $s_i = k\,r_i + (1 - k)$

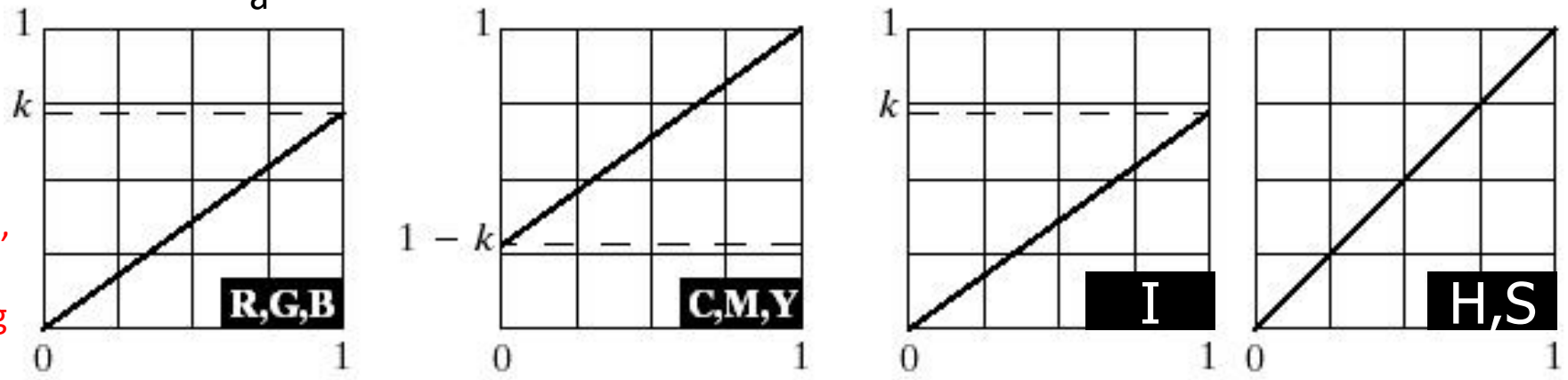
Sumber: Jen-Chang Liu, Spring 2006
Color Image Processing

k = 0.7

Original image

Output image (after decreasing its intensity by 30%)

a



Sumber: Jen-Chang Liu, Spring 2006
Color Image Processing

Menyunting saturasi warna:

(kiri) Citra makanan yang diambil dengan kamera digital;

(tengah) nilai saturation tiap pixel diturunkan 20%

(kanan) nilai saturation tiap pixel dinaikkan 40%

Modifikasi intensitas dalam model HSV

```
rgb = imread('peppers512.bmp');
imshow(rgb), title('Original image');
hsv = rgb2hsv(rgb);
H = hsv(:,:,1);
S = hsv(:,:,2);
V = hsv(:,:,3);
k = 0.7;
V = k*V;
hsv2 = cat(3, H, S, V);
rgb2 = hsv2rgb(hsv2);  % Konversi HSV ke RGB
figure, imshow(rgb2), title ('Output image');
```

Original image

Output image

Modifikasi intensitas dalam model RGB

```
rgb = imread('peppers512.bmp');
imshow(rgb), title('Original image');
k = 0.7;
for i=1:3
   rgb(:,:,i) = k * rgb(:,:,i);
end
rgb2 = cat(3, rgb(:,:,1), rgb(:,:,2),
rgb(:,:,3));
figure, imshow(rgb2), title ('Output image');
```

Original image

Output image

Modifikasi saturasi dalam model HSV

```
rgb = imread('peppers512.bmp');
imshow(rgb), title('Original image');
hsv = rgb2hsv(rgb);
H = hsv(:,:,1);
S = hsv(:,:,2);
V = hsv(:,:,3);
k = 1.5;        % Saturasi dinaikkan 50%
S = k*S;
hsv2 = cat(3, H, S, V);
rgb2 = hsv2rgb(hsv2);  % Konversi HSV ke RGB
figure, imshow(rgb2), title ('Output image');
```

Original image

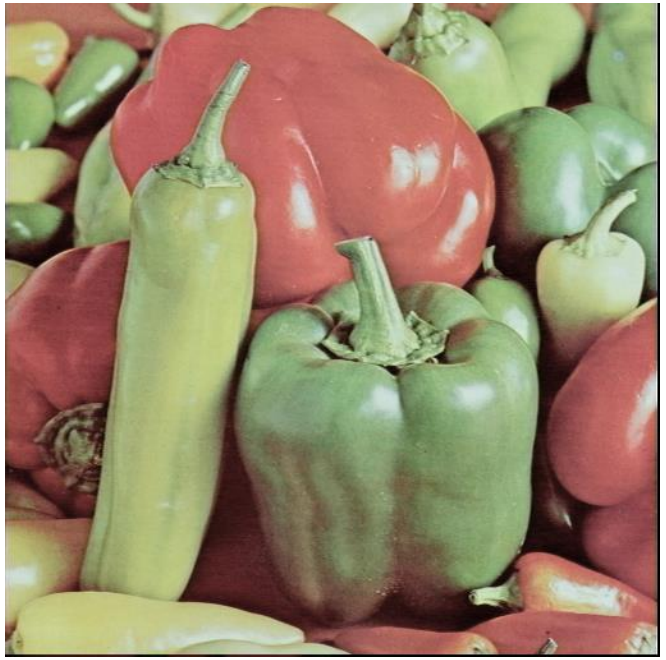
Output image

Modifikasi saturasi dalam model HSV

```
rgb = imread('peppers512.bmp');
imshow(rgb), title('Original image');
hsv = rgb2hsv(rgb);
H = hsv(:,:,1);
S = hsv(:,:,2);
V = hsv(:,:,3);
k = 0.4;  % Saturasi diturunkan menjadi 40% semula
S = k*S;
hsv2 = cat(3, H, S, V);
rgb2 = hsv2rgb(hsv2);  % Konversi HSV ke RGB
figure, imshow(rgb2), title ('Output image');
```

Original image

Output image



Modifikasi hue dalam model HSV

```
rgb = imread('peppers512.bmp');
imshow(rgb), title('Original image');
hsv = rgb2hsv(rgb);
H = hsv(:,:,1);
S = hsv(:,:,2);
V = hsv(:,:,3);
k = 0.5;     % Hue diturunkan menjadi 50% semula
H = k*H;
hsv2 = cat(3, H, S, V);
rgb2 = hsv2rgb(hsv2);  % Konversi HSV ke RGB
figure, imshow(rgb2), title ('Output image');
```

Original image

Output image

Bersambung