










Pengolahan Citra :  
**Representasi Citra**

Universitas Gunadarma  
2006

## Representasi Citra dalam File (1/3)

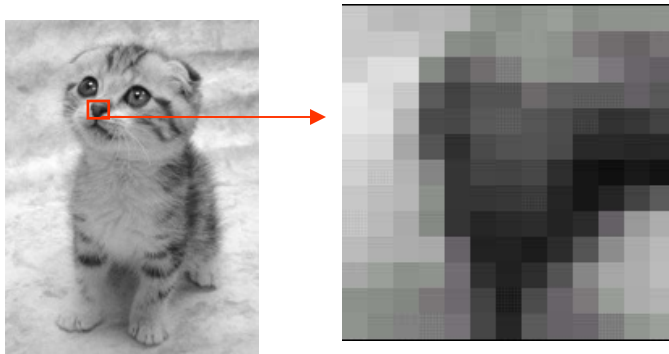
- Pertama-tama seperti halnya jika kita ingin melukis sebuah gambar, kita harus memiliki palet dan kanvas
  - **Palet:** kumpulan warna yang dapat membentuk citra, sama halnya seperti kita hendak melukis dengan cat warna, kita memiliki palet yang bisa kita isikan berbagai warna cat air
  - Setiap warna yang berbeda dalam palet tersebut kita beri nomor (berupa angka)
  - Contoh untuk citra monokrom (warnanya hanya putih-abuabu-hitam), berarti kita memiliki palet sbb:

								
0	1	2	.....	128	.....	.....	.....	255

- Setelah itu kita dapat menggambar menggunakan warna-warna dalam palet tersebut di atas sebuah **kanvas**
  - Sebuah kanvas dapat dianggap sebagai sebuah matriks dimana setiap elemen dari matriks tersebut bisa kita isikan dengan salah satu warna dari palet
- Informasi tentang palet (korespondensi antara warna dengan angka) disimpan dalam komputer (program pembuka citra seperti Paint, Photoshop, dll) sehingga sebuah file citra dalam komputer hanya perlu menyimpan angka-angka yang merepresentasikan sebuah warna.

# Representasi Citra dalam File (2/3)

- Sebuah citra direpresentasikan dalam sebuah matriks yang berisi angka-angka
- Contoh :



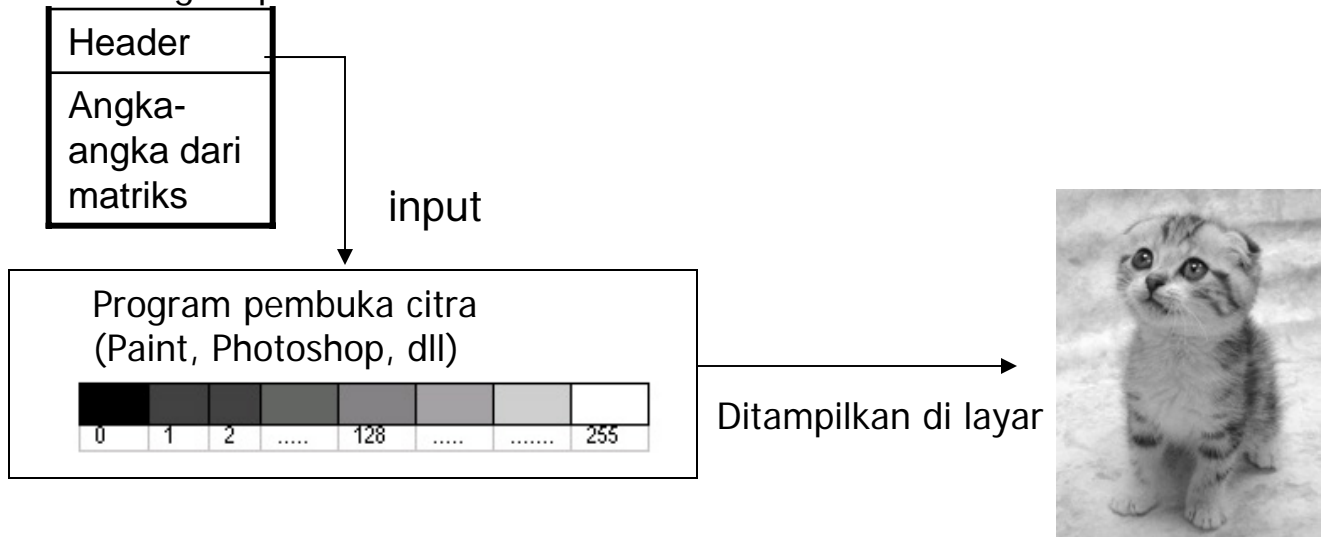
=

201	188	181	185	180	147	140	149	155	138	144	144	145
199	200	201	188	139	132	147	150	143	123	112	102	117
207	221	222	136	90	111	125	145	140	138	122	104	97
231	219	200	90	65	84	84	107	95	92	92	99	89
227	223	181	74	72	89	92	86	77	63	50	55	65
217	211	166	85	47	75	82	83	75	42	42	39	40
208	195	179	131	54	68	66	72	46	21	15	24	19
198	187	181	141	53	54	55	59	37	21	37	66	90
195	184	170	134	52	38	42	45	35	43	98	152	172
186	175	171	169	100	34	34	27	44	85	139	170	184
167	156	142	144	112	48	32	46	84	133	166	172	186
142	139	131	120	108	67	30	76	102	123	153	171	178
145	134	128	125	117	70	38	91	101	105	125	146	157

## Representasi Citra dalam File (3/3)

- Jika kita menyimpan gambar kucing tadi ke dalam sebuah file (kucing.bmp), maka yang disimpan dalam file tersebut adalah angka-angka yang diperoleh dari matriks kanvas.

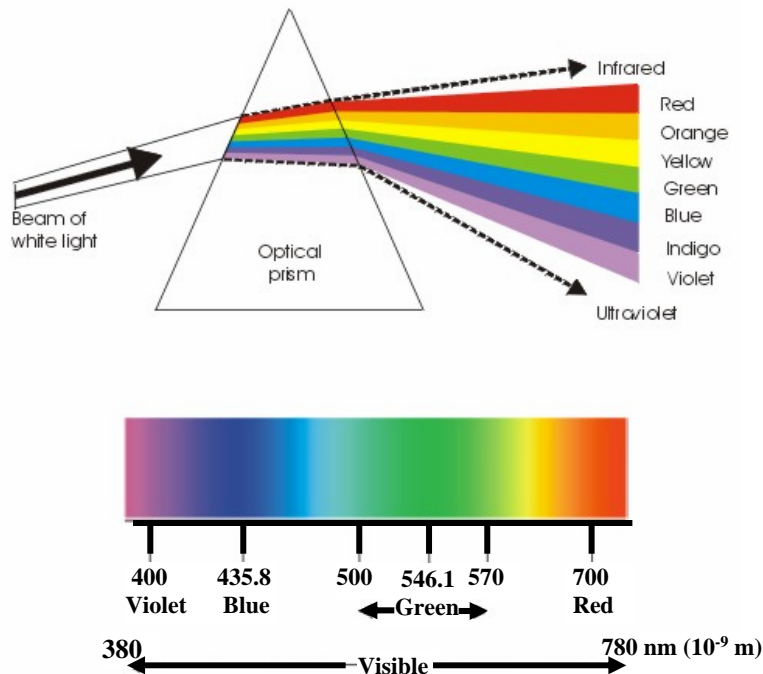
File kucing.bmp:



- Ada bermacam format representasi citra dalam file, seperti bmp, tif, jpg, dan sebagainya.
- Format BMP merupakan format yang kurang efisien, karena semua informasi angka dalam baris disimpan semua. Misalkan ukuran header adalah H byte, ukuran citra 100x100 byte monokrom, maka ukuran file bmp tersebut adalah :  $H + \text{data citra} = H + 10000 \text{ Byte}$
- Bagian data citra (10000 byte) sebenarnya bisa dikompresi agar ukuran file tidak terlalu besar. Salah satu cara kompresi adalah dengan terlebih dahulu mentransformasikan citra ke ruang yang berbeda (contoh: format file JPEG)

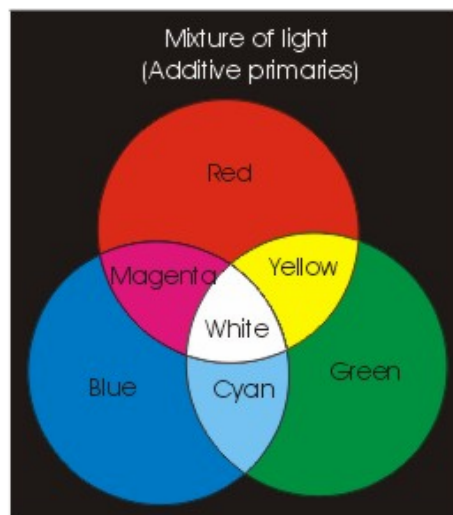
# Warna (1/2)

- Persepsi visual citra berwarna lebih kaya dan lebih disenangi dibandingkan citra hitam putih (greyscale).
- Warna yang diterima oleh mata dari sebuah objek ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut.
- Warna sinar yang direspon oleh mata adalah sinar tampak (visible spektrum) dengan panjang gelombang berkisar dari 400(biru) sampai 700 nm (merah)



## Warna (2/2)

- Warna-warna yang diterima oleh mata manusia merupakan hasil kombinasi dari panjang gelombang berbeda
- Rentang warna yang paling lebar adalah RED (R), GREEN (G) dan BLUE (B) yang disebut **Warna Pokok**.
- Warna lain diperoleh dengan mencampurkan ketiga warna pokok tersebut dengan perbandingan tertentu.



# Atribut Warna

- **Intensity / brightness / luminance**
  - Atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa mempedulikan warna.
  - Kisaran nilainya adalah antara gelap (hitam =0) dan terang (putih=255)
- **Hue**
  - Menyatakan warna sebenarnya, seperti merah violet dan kuning.
  - Digunakan untuk membedakan warna-warna dan menentukan kemerahan (redness), kehijauan(greeness), dsb dari cahaya.
  - Hue berasosiasi dengan panjang gelombang cahaya.
  - Hue dikuantisasi dengan nilai dari 0 sampai 255 (0 menyatakan merah lalu memutar nilai-nilai spektrum dan kembali lagi ke 0 untuk menyatakan merah lagi)
- **Saturation**
  - Menyatakan tingkat kemurnian warna cahaya, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna.
  - Contoh : warna merah adalah 100% warna jenuh (saturated color), sedangkan warna pink adalah warna merah dengan tingkat kejenuhan rendah (karena ada warna putih di dalamnya).
  - 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.

## Sistem Koordinat Warna (1/3)

- *CIE* ( Commission International de l'Eclairage) atau International Lighting Committee adalah lembaga yang membakukan warna pada tahun 1931.
- Standar panjang gelombang warna pokok (CIE)
  - R : 700 nm, G : 546.1 nm, B : 435.8 nm
- *CIE* mendenisikan model warna dengan menggunakan warna-warna fiktif yang dilambangkan dengan X,Y dan Z sehingga disebut model XYZ.
- Kromatisitas (*chromaticity of color*) masing-masing warna pokok, menunjukkan persentase relatif suatu warna pokok di antara warna pokok lainnya pada warna yang diberikan, didefinisikan sebagai :

$$x = \frac{X}{X+Y+Z} \quad (1)$$

$$y = \frac{Y}{X+Y+Z} \quad (2)$$

$$z = \frac{Z}{X+Y+Z} \quad (3)$$

Warna putih acuan dinyatakan dengan  $x = y = z = 1$ . Jumlah seluruh nilai kromatisitas warna adalah satu

$$x + y + z = 1$$



# Sistem Koordinat Warna (2/3)

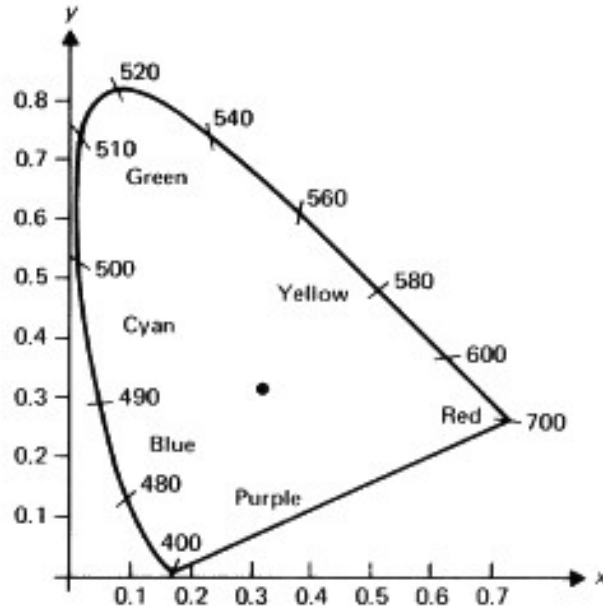
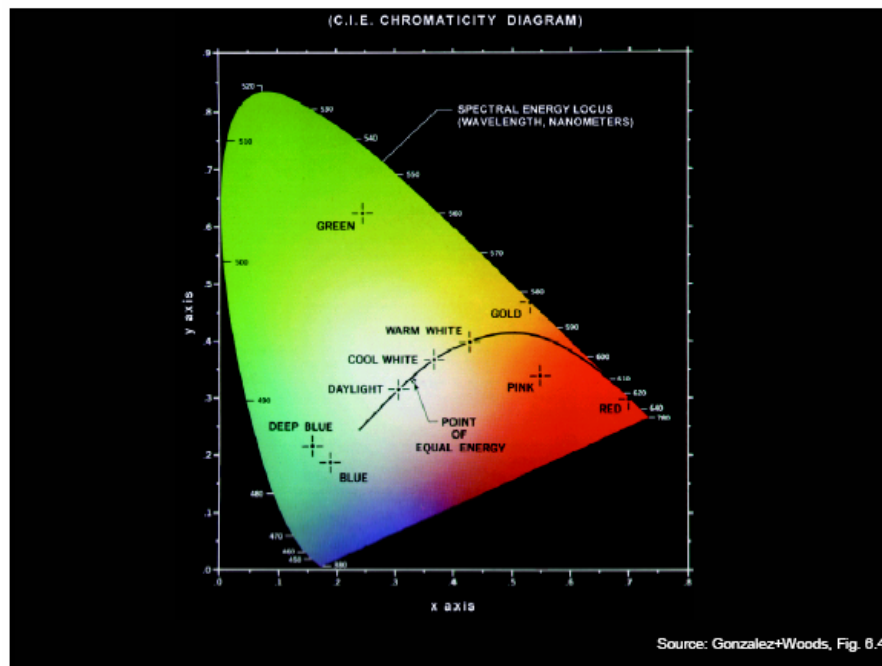


Fig. 17.20 The CIE chromaticity diagram, on the  $X + Y + Z = 1$  plane. Wavelengths are in nanometers.



# Sistem Koordinat Warna (3/3)

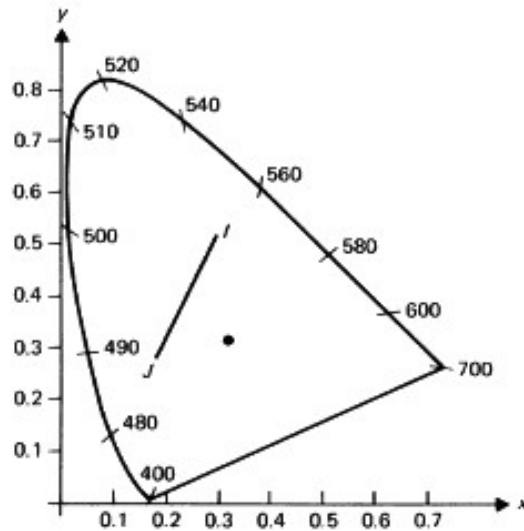


Fig. 17.24 Mixing two colors. All colors on connecting line can be created.

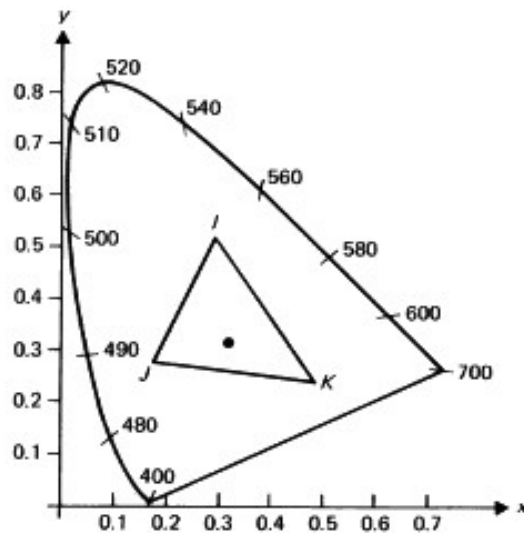


Fig. 17.25 Mixing three colors. All colors in triangle can be created.

# Model Warna

- Model warna yang banyak digunakan saat ini berorientasi hardware (contoh monitor dan printer) atau aplikasi dimana manipulasi warna menjadi tujuannya (kreasi warna grafik untuk animasi)
- Model Warna berorientasi Hardware :
  1. Model RGB (red, green, blue) untuk warna monitor dan warna pada kamera video
  2. Model CMY (cyan, magenta, yellow) untuk model printer;
  3. Model YIQ model, digunakan untuk standard televisi. Y berkoresponden dengan luminasi, I dan Q adalah dua komponen kromatik yang disebut inphase dan quarature .
- Model Warna berorientasi Software (hue, saturation, brightness) adalah manipulasi :
  1. Model HSV (hue, saturation, value);
  2. model HSI (hue, saturation, intensity);
  3. Model HLS (hue, lightness, saturation).

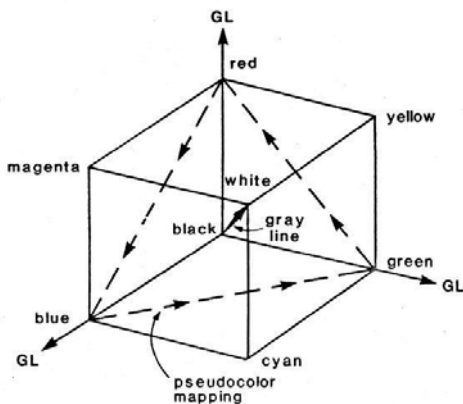


FIGURE 2-39. The RGB color cube.

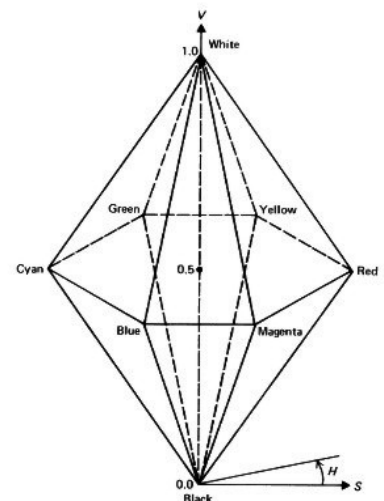


Fig. 17.31 Double hexcone HLS color model.

# Transformasi Sistem Koordinat Warna (1/)

- Transformasi warna dari basis *CIE RGB* ke *CIE XYZ* yang menjadi acuan untuk platform perangkat keras tertentu, dapat dilakukan sebagai berikut :

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.099 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}$$

- Transformasi sebaliknya dari *CIE XYZ* ke *CIE RGB* dilakukan sebagai berikut :

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 2.365 & -0.310 & -0.468 \\ -0.515 & 1.45 & 0.088 \\ 0.005 & -0.014 & 1.009 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

# Transformasi Sistem Koordinat Warna (2/)

- Model warna yang diusulkan oleh *National Television Systems Committee (NTSCC)* menampilkan citra berwarna pada layar CRT.
- Format NTSC terdiri dari 3 komponen : **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 untuk pixel  $I$ , maka nilai YIQ untuk pixel yang bersangkutan dihitung :
 
$$\begin{bmatrix} Y_i \\ I_i \\ Q_i \end{bmatrix} = \begin{bmatrix} 0.299 & 0.857 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}$$

- Nilai NTSC RGB semula dapat dihitung melalui transformasi :
 
$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.273 & -0.647 \\ 1.000 & -1.104 & 1.701 \end{bmatrix} \begin{bmatrix} Y_i \\ I_i \\ Q_i \end{bmatrix}$$

- Transformasi dari NTSC RGB ke CIE RGB :
 
$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 1.167 & -0.146 & -0.151 \\ 0.114 & 0.753 & 0.159 \\ -0.001 & 0.059 & 1.128 \end{bmatrix} \begin{bmatrix} R\_NTSC_i \\ G\_NTSC_i \\ B\_NTSC_i \end{bmatrix}$$

## Model Warna CMY dan CMYK

- Warna *cyan (C)*, *magenta (M)* dan *yellow (Y)* adalah **warna komplementer** terhadap *red*, *greend* dan *blue*.
- Dua buah warna komplementer jika dicampur dengan perbandingan yang tepat menghasilkan warna putih.

- Model CMY dapat diperoleh dari model RGB :

$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

- Model CMY digunakan untuk mencetak warna, namun karena ketidasesempurnaan tinta, CMY tidak dapat menghasilkan warna hitam. Karena itu model CMY disempurnakan menjadi model CMYK, dengan K menyatakan warna keempat dengan perhitungan :

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

$$C = C - K$$

$$M = M - K$$

$$Y = Y - K$$

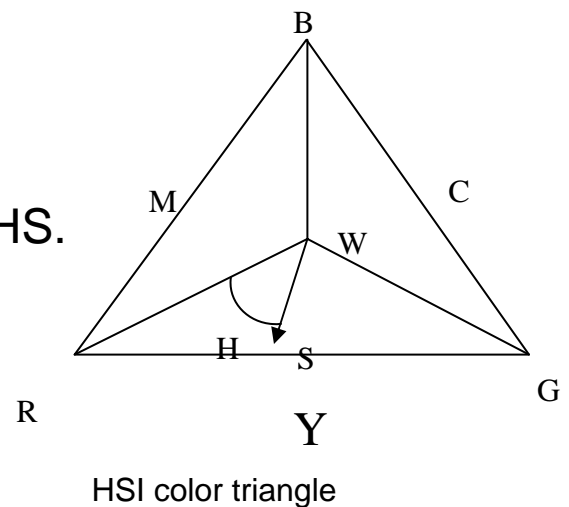
## Transformasi Warna RGB ke HIS (1/2)

- Basis RGB tidak cocok untuk beberapa aplikasi pengolahan citra
- Pada aplikasi pengenalan obyek akan lebih mudah mengidentifikasi obyek dengan perbedaan *hue* dengan memberikan nilai ambang pada rentang nilai-nilai *hue* yang melingkupi obyek.
- Diperlukan konversi dari RGB ke nilai intensity (I), hue (H) dan saturation (S) atau model warna IHS.

$$I = 1/3 (R+G+B)$$

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

$$S = 1 - \frac{3}{R + G + B}$$



## Transformasi Warna RGB ke HIS (2/2)

- Alternatif lain konversi RGB ke HIS
  1. Merotasikan koordinat RGB ke sistem koordinat  $(I, V_1, V_2)$  :
 
$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & \sqrt{3}/3 & \sqrt{3}/3 \\ 0 & 1\sqrt{2} & -1\sqrt{2} \\ 2\sqrt{6} & -1\sqrt{6} & -1\sqrt{6} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
  2. Menghitung H dan S dari koordinat  $(V_1, V_2)$ 

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

$$S = (V_1^2 + V_2^2)^{1/2}$$
- Transformasi dari model HIS ke model RGB dengan prosedur balikan :
 
$$V_1 = S \cos (H) \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & 0 & 2\sqrt{6} \\ \sqrt{3}/3 & 1\sqrt{2} & -1\sqrt{6} \\ \sqrt{3}/3 & -1\sqrt{2} & -1\sqrt{6} \end{bmatrix} \begin{bmatrix} R \\ V_1 \\ V_2 \end{bmatrix}$$

$$V_2 = S \sin (H)$$
- Transformasi citra dari basis RGB ke basis HIS dilakukan sebelum pengolahan citra. Citra yang sudah diproses dapat dikonversikan kembali ke basis RGB untuk tujuan display.