RANGKUMAN
PENGOLAHAN CITRA DIGITAL

Dibuat oleh :

Nama: Muhammad Fajrin Aljabar

Nim: F55121037

Kelas : A

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2023

Link Github : https://github.com/Fajrin21/F55121037_Muhammad-Fajrin-Aljabar.git

1. Dasar Warna

   Warna adalah cahaya yang dipantulkan oleh objek, sehingga cahaya inilah yang dapat dilihat oleh manusia. Spektrum cahaya kromatis berkisar antara 400-700 nm. Kromatis adalah kualitas warna cahaya yang ditentukan oleh panjang gelombang. Karakteristik persepsi mata manusia dalam yang membedakan antara satu warna dengan warna yang lain berupa hue, saturation, dan brightness.

   A. Hue adalah warna yang dikenal manusia seperti merah dan hijau. Properti ini mencerminkan warna yang ditangkap oleh mata manusia yang menanggapi berbagai nilai panjang gelombang cahaya. Contohnya jika mata menerima panjang gelombang antara 430 dan 480 nanometer, maka sensasi yang diterima adalah warna biru.

   B. Saturation menyatakan tingkat kemurnian warna atau seberapa banyak cahaya putih yang tercampur dengan hue. Setiap warna murni bersaturasi 100% dan tidak mengandung cahaya putih sama sekali. Jadi suatu warna murni yang bercampur dengan cahaya putih memiliki saturasi antara 0 dan 100%

   C. Brightness atau kecerahan yaitu menyatakan intensitas pantulan objek yang diterima oleh mata. Intensitas ini dinyatakan sebagai perubahan warna putih menuju abu abu dan terakhir mencapai ke warna hitam, atau dikenal dengan istilah aras keabuan.

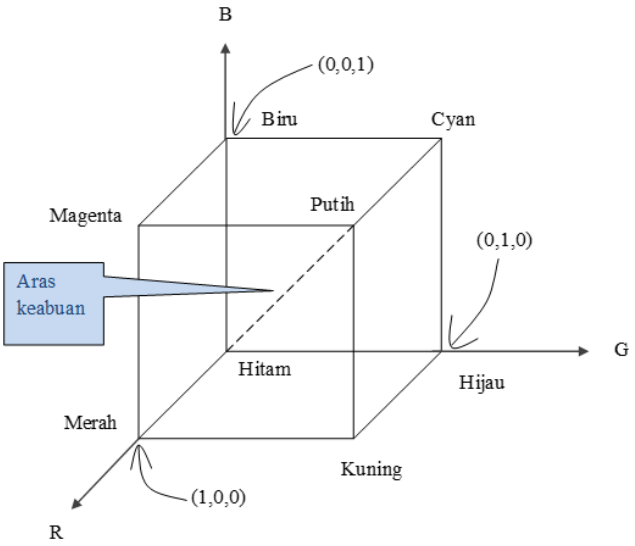
   Jadi dapat disimpulkan bahwa kromatik ini adalah gabungan antara hue dan saturation sedangkan istilah akromatik merujuk ke kecerahan.

2. Ruang Warna

   Gonzales & woods mendefinisikan ruang warna atau kadang disebut sistem warna atau model warna sebagai suatu spesifikasi sistem koordinat dan suatu subruang dalam sistem tersebut dengan setiap warna dinyatakan dengan satu titik di dalamnya. Tujuan dibentuknya ruang warna adalah untuk memfasilitasi spesifikasi warna dalam bentuk suatu standar. Ruang warna yang paling dikenal pada perangkat komputer adalah RGB, yang sesuai dengan watak manusia dalam menangkap warna. Namun, kemudian dibuat banyak ruang warna, antara lain HIS, CMY, LUV, dan YIQ.

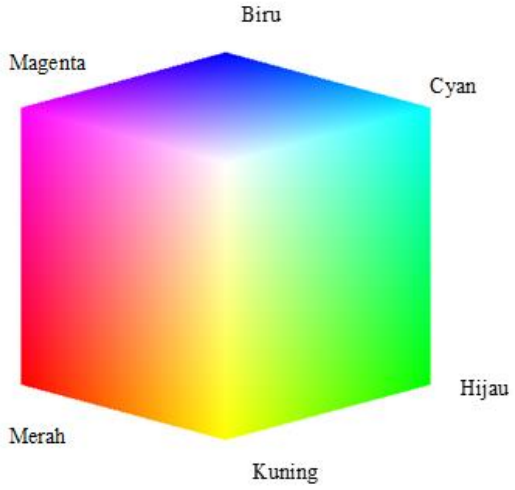
3. Ruang Warna RGB

   Ruang warna RGB biasanya digunakan pada monitor CRT dan kebanyakan sistem grafika komputer. Ruang warna menggunakan tiga komponen dasar yaitu merah (R), hijau (G), dan biru (B). Setiap piksel dibentuk oleh ketika komponen tersebut. Model RGB biasa disajikan dalam bentuk kubus tiga dimensi yang dimana tiap komponen itu mewakili tiap sumbu. Warna hitam berada pada titik asal dan warna putih berada di ujung kubus yang bersebrangan.



Gambar 9.1 Skema ruang warna RGB dalam bentuk kubus

Pada gambar berikut terdapat kubus warna secara nyata dengan resolusi 24 bit, yang dimana dengan menggunakan 24 bit, jumlah warna yang dapat dihasilkan mencapai 16.777.216.



Gambar 9.2 Kubus warna dengan 24 bit

RGB biasanya digunakan karena kemudahan dalam perancangan hardware, tetapi sebenarnya tidak ideal untuk beberapa aplikasi. Dikarenakan mengingat warna merah, hijau, dan biru, sesungguhnya terkorelasi erat, sangat sulit untuk beberapa algoritma pemrosesan citra. Contohnya, kebutuhan untuk memperoleh warna alamiah seperti merah dengan menggunakan RGB menjadi sangat kompleks mengingat komponen R dapat berpasangan dengan G dan B, dengan nilai berapa saja. Hal ini menjadi mudah jika menggunakan ruang warna HLS ataupun HSV.

4. Ruang Warna CMY/CMYK

Model warna CMY (cyan, magenta, yellow) mempunyai hubungan dengan RGB sebagai berikut

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

Dalam hal ini R, G, dan B berupa nilai warna yang sudah dinormalisasi, dengan jangkauan [0, 1]. Pada CMY, warna hitam diperoleh jika C, M, dan Y bernilai sama. Tetapi pada aplikasi printer, warna hitam ditambahkan tersendiri sehingga membentuk CMYK, dengan K menyatakan warna hitam. Hal ini dikarenakan warna hitam ini dapat langsung diambil dari tinta hitam tanpa perlu mencampur dengan warna lain. Lagipula, tinta warna hitam lebih murah daripada tinta berwarna.

(a) Penjumlahan warna pada sistem RGB (b) Pengurangan warna pada sistem CMY

Gambar 9.3 Warna-warna lain dapat dibentuk melalui kombinasi tiga warna dasar

Berikut ini salah satu rumus yang digunakan untuk CMY ke CMYK

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

5. Ruang Warna YIQ

Ruang warna YIQ, yang juga dikenal dengan nama ruang warna NTSC, dirumuskan oleh NTSC ketika mengembangkan sistem televisi berwarna di Amerika Serikat. Pada model ini, Y disebut luma (yang menyatakan luminans yaitu merupakan satu satunya komponen yang digunakan oleh televisi hitam putih), I serta Q disebut chroma (yaitu sinyal yang digunakan dalam video sistem untuk menyampaikan informasi warna dari gambar). Konversi YIQ berdasarkan RGB sebagai berikut.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0{,}596 & -0{,}274 & -0{,}322 \\ 0{,}211 & -0{,}523 & 0{,}312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Komposisi RGB untuk Y secara sistematis optimal untuk ditampilkan pada penerima TV hitam-putih

Matriks yang berisi koefisien konversi mempunyai ciri – ciri sebagai berikut

1. Jumlah elemen dalam baris pertama bernilai 1
2. Jumlah koefisien elemen dalam baris kedua maupun baris ketiga benilai nol

Berikut ini merupakan contoh untuk konversi YIQ ke RGB sebagai berikut

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1{,}000 & 0{,}956 & 0{,}621 \\ 1{,}000 & -0{,}272 & -0{,}647 \\ 1{,}000 & -1{,}106 & 1{,}703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

6. Ruang Warna YcbCr

Ruang warna YcbCr biasanya digunakan pada video digital. Yang dimana pada ruang warna ini, komponen Y menyatakan intensitas, sedangkan Cb dan Cr menyatakan informasi warna. Proses konversi dari RGB dilakukan dengan beberapa cara. Contohnya

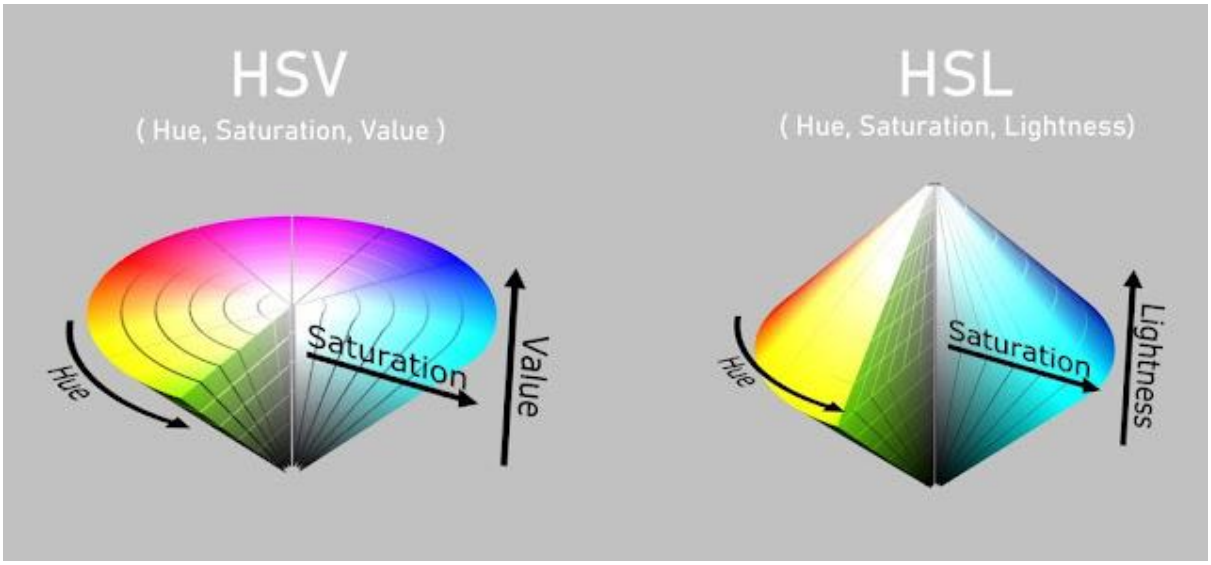
$$Y = 0{,}29900R + 0{,}58700G + 0{,}11400B$$
$$C_b = -0{,}16874R - 0{,}33126G + 0{,}5000B$$
$$C_r = +0{,}5000R - 0{,}41869G - 0{,}08131B$$

Adapun pengonversian dari YcbRr ke RGB sebagai berikut

$$R = Y + 1.40200C_r$$
$$G = Y - 0{,}34414C_b - 0{,}71414C_r$$
$$B = Y + 1{,}77200C_b$$

7. Ruang HIS, HSV, dan HSL

HSV dan HSL merupakan contoh ruang warna yang merepresentasikan warna seperti yang dilihat oleh mata manusia. H berasal dari kata “hue” yang berarti istilah untuk menggambarkan nama dari warna warna yang ada. S berasal dari “saturation” yang berarti intensitas warna atau sebuah kemurnian dari warna, sederhananya kepekatan warna. L berasal dari kata “lightness” yang berfungsi untuk mempercerah setiap warna. I berasal dari kata “intensity” yang merupakan istilah yang digunakan untuk menyatakan kualitas atau kekuatan warna. dan V berasal dari kata “Value” yang berarti nilai warna untuk menentukan kecerahan sebuah warna atau gelap terangnya sebuah warna.



Cara mendapatkan nilai H, S, V berdasarkan R, G, B

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right)$$

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

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

Namun, cara ini membuat *hue* tidak terdefinisikan kalau S bernilai nol. Cara kedua terdapat pada Acharya & Ray (2005). Rumus-rumus yang digunakan sebagai berikut

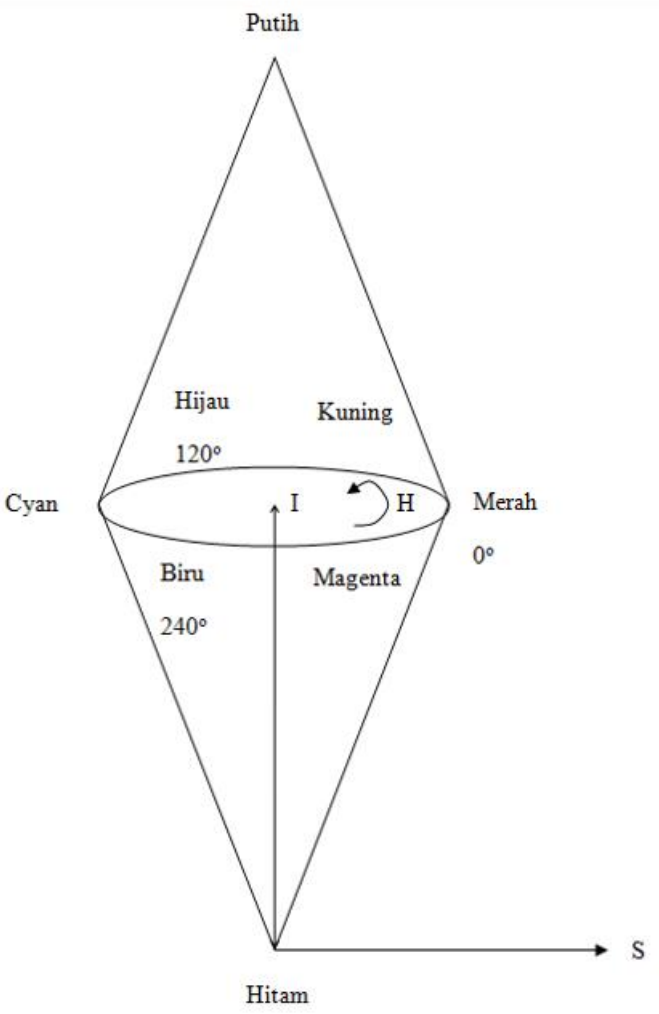
$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)}$$

$$V = \max(r,g,b)$$

$$S = \begin{cases} 0, & \text{jika V} = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & \text{V} > 0 \end{cases}$$

$$H = \begin{cases} 0, & jika\ S = 0 \\ \frac{60*(g-b)}{S*V}, & jika\ V = r \\ 60 * \left[2 + \frac{b-r}{S*V}\right], & jika\ V = g \\ 60 * \left[4 + \frac{r-g}{S*V}\right], & jika\ V = b \end{cases}$$

$$H = \text{H} + 360 \text{ jika H} < 0$$



Gambar 9.5 Model HSI

Gambar 9.5 memperlihatkan ruang warna HSI. Konversi dari RGB ke HSI dilakukan melalui rumus berikut (Gonzalez & Woods, 2002):

$$H = \begin{cases} 0, & jikaB \leq G \\ 360 - \theta, & jika\ B > G \end{cases}$$

Pada rumus di atas, H menyatakan *hue*. Adapun θ diperoleh melalui rumus berikut:

$$\theta = cos^{-1}\left\{\frac{1/2[(R-G)+(R-B)]}{[(R-G)^2(R-B)(G-B)]^{1/2}}\right\}$$

Selanjutnya, komponen *saturation* dihitung dengan menggunakan rumus:

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

dan komponen intensitas diperoleh melalui:

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

Untuk memperoleh RGB berdasarkan HSI, diperlukan beberapa aturan. Apabila H berada dalam sektor RG ($0^o \leq H < 120^o$), komponen R, G, dan B dihitung dengan menggunakan rumus-rumus berikut:

$$B = I(1 - S)$$
$$R = I\left(1 + \frac{S\cos H}{\cos(60^o - H)}\right)$$
$$G = 3I - (R + B))$$

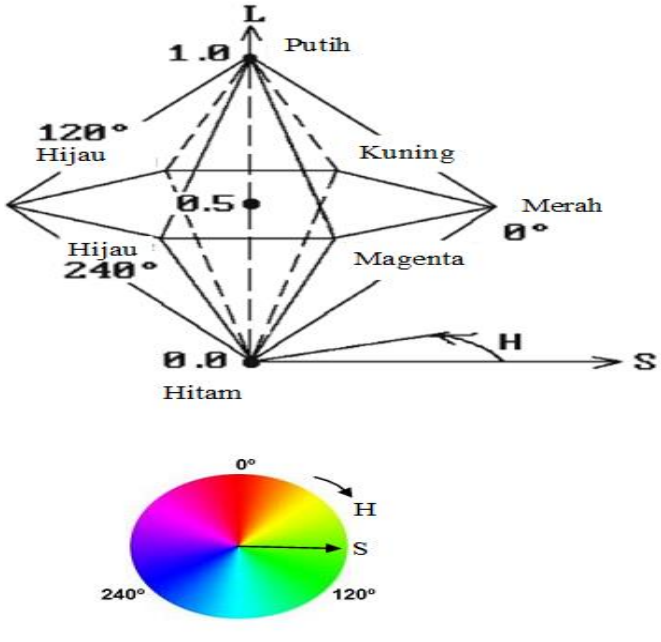
Apabila H berada di dalam sektor GB ($120^o \leq H < 240^o$), komponen R, G, dan B dihitung dengan menggunakan rumus-rumus berikut:

$$H = H - 120$$
$$R = I(1 - S)$$
$$G = I\left(1 + \frac{S\cos H}{\cos(60^o - H)}\right)$$
$$B = 3I - (R + G))$$

Apabila H berada di dalam sektor GB ($240^o \leq H < 360^o$), komponen R, G, dan B dihitung dengan menggunakan rumus-rumus berikut:

$$H = H - 240$$
$$G = I(1 - S)$$
$$B = I\left(1 + \frac{S\cos H}{\cos(60^o - H)}\right)$$
$$R = 3I - (R + G))$$

Perlu diketahui, mengingat nilai pada HSI berada di dalam jangkauan [0, 1], maka untuk mendapatkan nilai H yang berkisar antara 0^o-360^o, H perlu dikalikan terlebih dulu dengan 360. Dengan demikian, jangkauan H berada dalam [0, 360].



Gambar 9.6 Model HSL

Model ruang HSL diperlihatkan pada Gambar 9.6. Besaran kecerahan (dinamakan *lightness*) disimbolkan dengan L. Perhitungan komponen H, S, dan L berdasarkan komponen R,G, dan B adalah seperti berikut (Agoston, 2005).

$$C_{min} = \min(R, G, B) \quad (9.33)$$

$$C_{max} = \max(R, G, B) \quad (9.34)$$

$$L = \frac{C_{max} + C_{min}}{2} \quad (9.35)$$

$$S = \begin{cases} 0, & C_{min} = C_{max} \\ \frac{C_{max} - C_{min}}{C_{max} + C_{min}}, & L \leq 0{,}5 \\ \frac{C_{max} - C_{min}}{2 - (C_{max} + C_{\min})}, & L > 0{,}5 \end{cases} \quad (9.36)$$

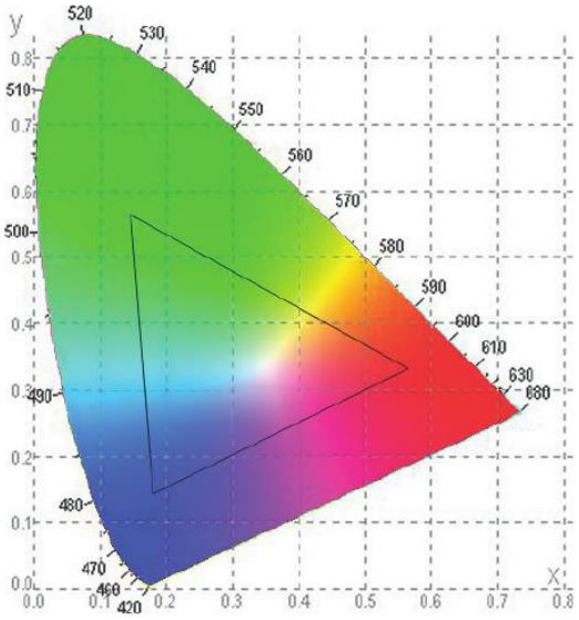
$$H = \begin{cases} UNDEFINED, & C_{min} = C_{max} \\ \frac{G - B}{C_{max} - C_{min}}, & R = C_{max} \\ 2 + \frac{B - R}{C_{max} - C_{min}}, & G = C_{max} \\ 4 + \frac{R - G}{C_{max} - C_{min}}, & untuk\ lainnya \end{cases} \quad (9.37)$$

$$H = H\ x\ 60 \quad (9.38)$$

$$Jika\ H < 0\ maka\ H = H + 360 \quad (9.39)$$

8. Ruang Warna CIELAB

CIELAB adalah nama lain dari CIE L*a*b. Pada diagram kromasitas CIE ditunjukkan bahwa setiap perpaduan x dan y menyatakan suatu warna. Namun hanya warna yang berada dalam area ladam (tapal kuda) yang bisa terlihat. Angka yang berada di tepi menyatakan panjang gelombang cahaya. Warna yang terletak di dalam segitiga menyatakan warna warna umum di monitor CRT, yang dapat dihasilkan oleh kompoenen warna merah, hijau, dan biru.



Gambar 9.7 Diagram kromasitas CIE

Transformasi RGB ke CIELAB dimulai dengan melakukan perhitungan sebagai berikut:

$$X = 0{,}412453R + 0{,}357580G + 0{,}180423B$$
$$Y = 0{,}212671R + 0{,}715160G + 0{,}072169B$$
$$Z = 0{,}019334R + 0{,}119193G + 0{,}950227B$$

Selanjutnya, L*a*b* didefinisikan sebagai berikut:

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16$$
$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right]$$
$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

Dalam hal ini, f(q) dihitung seperti berikut:

$$f(q) = \begin{cases} q^{\frac{1}{3}}, \ jika\ q > 0{,}008856 \\ 7{,}787q + 16/116, \ untuk\ yang\ lain \end{cases}$$

X_n, Y_n, Z_n diperoleh melalui R=G=B=1 dengan jangkauan R,G, B berupa [0, 1].

9. Memperoleh Statistika Warna

Fitur warna dapat diperoleh melalui perhitungan statistis seperti rerata, deviasi standar, *skewness*, dan kurtosis (Martinez & Martinez, 2002). Sebagai contoh, fitur-fitur tersebut dapat digunakan untuk kepentingan identifikasi tanaman hias (Kadir, dkk., 2011b dan Kadir, dkk., 2011c). Perhitungan dikenakan pada setiap komponen R, G. dan B.

Rerata memberikan ukuran mengenai distribusi dan dihitung dengan menggunakan rumus:

$$\mu = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} P_{ij}$$

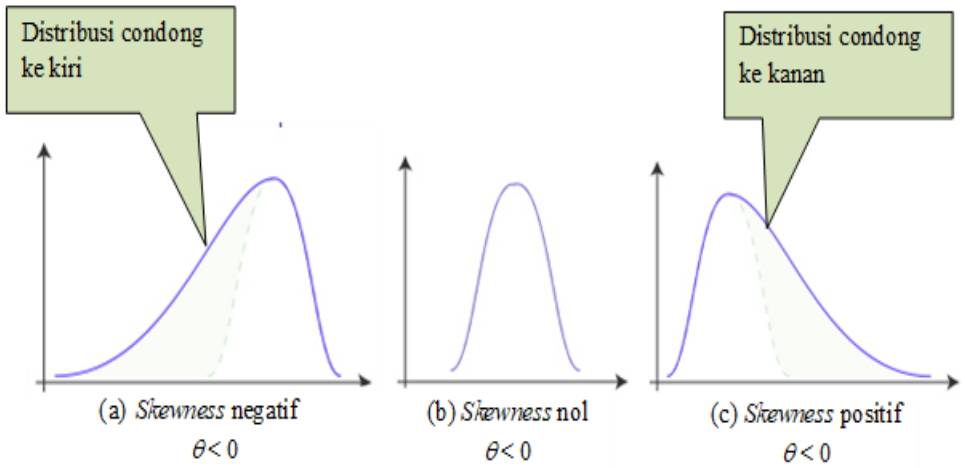
Varians menyatakan luas sebaran distribusi. Akar kuadrat varians dinamakan sebagai deviasi standar. Adapun rumus yang digunakan untuk menghitungnya sebagai berikut:

$$\sigma = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(P_{ij} - \mu)^2}$$

Skewness atau kecondongan menyatakan ukuran mengenai ketidaksimetrisan. Distribusi dikatakan condong ke kiri apabila memiliki nilai *skewness* berupa bilangan negatif. Sebaliknya, distribusi dikatakan condong ke kanan apabila memiliki nilai

skewness berupa bilangan positif. Jika distribusi simetris, koefisien *skewness* bernilai nol. Ilustrasi *skewness* dapat dilihat pada Gambar 9.8. *Skewness* dihitung dengan cara seperti berikut:

$$\theta = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(P_{ij}-\mu)^3}{MN\sigma^3}$$

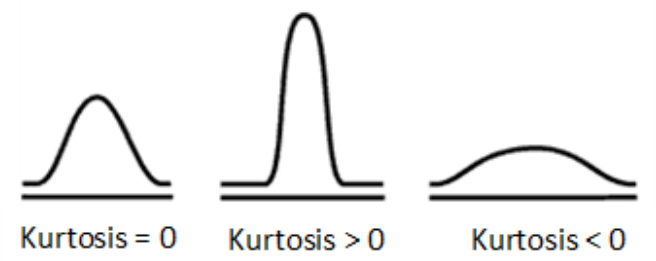


Gambar 9.8 Skewness menggambarkan kecondongan distribusi data

Kurtosis merupakan ukuran yang menunjukkan sebaran data bersifat meruncing atau menumpul. Perhitungannya seperti berikut:

$$\gamma = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(P_{ij}-\mu)^4}{MN\sigma^4} - 3$$

Definisi di atas membuat distribusi normal standar memiliki kurtosis nol. Nilai positif mengindikasikan bahwa distribusi bersifat lancip dan nilai negatif menyatakan distribusi yang datar (lihat Gambar 9.9). Perlu diketahui, pada Persamaan 9.47 hingga 9.50, M adalah tinggi citra, N menyatakan lebar citra, dan P_{ij} adalah nilai warna pada baris *i* dan kolom *j*.



Gambar 9.9 Kurtosis menggambarkan keruncingan distribusi normal

10. Mengatur Kecerahan dan Kontras

Untuk melihat efek kecerahan dan kontras, cobalah beberapa perintah berikut.

```
>> Img = imread('C:\Image\inns.png'); ↵
>> imshow(Img) ↵
>>
```

Kode di atas digunakan untuk melihat citra inns.png (Gambar 9.10.(a)). Selanjutnya,

```
>> C = Img + 30; ↵
>> imshow(C) ↵
>>
```

membuat citra dicerahkan sejauh 30. Hasilnya ditunjukkan pada Gambar 9.10(b). Lalu, cobalah kode berikut:

```
>> K = 2 * C; ↵
>> imshow(K) ↵
>>
```

Hasilnya dapat dilihat pada Gambar 9.10(c). Hal yang menarik dapat diperoleh dengan hanya memberikan kontras pada komponen R. Caranya:

```
>> K = C; ↵
>> K(:,:,1) = 2 * K(:,:,1); ↵
>>
```

Kode di atas mula-mula membuat K bernilai sama dengan C (efek pencerahan). Selanjutnya,

```
K(:,:,1) = 2 * K(:,:,1);
```

membuat hanya komponen R saja yang dinaikkan dua kali. hasilnya ditunjukkan pada Gambar 9.10(d).

(a) Citra inns.png

(b) Efek pencerahan melalui C = Img + 30;

(c) Efek kontras dan pencerahan melalui C = 2 x (Img + 30)

(d) Efek pencerahan melalui C = Img + 30 pada komponen R, G, dan B dan kontras sebesar 2 kali hanya pada komponen R

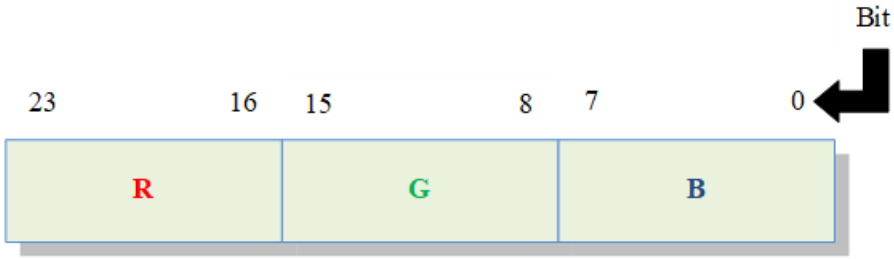
Gambar 9.10 Pencerahan dan peningkatan kontras pada citra berwarna

11. Menghitung Jumlah Warna

Penghitungan warna dilakukan dengan mula-mula menyusun komponen R, G, dan B untuk setiap piksel menjadi sebuah nilai dengan komposisi seperti terlihat pada Gambar 9.11. Untuk keperluan seperti itu, maka:

1. G perlu digeser ke kiri sebanyak 8 bit dan
2. R perlu digeser ke kiri sebanyak 16 bit.

Pada *Octave* dan *MATLAB*, penggeseran bit dilakukan melalui fungsi **bitshift**.



Gambar 9.11 Komposisi R, G, dan B dalam sebuah nilai

Setelah nilai gabungan R, G, dan B terbentuk dan diletakkan ke larik `Data`, isi larik tersebut diurutkan. Pengurutan tersebut dimaksudkan untuk mempermudah penghitungan jumlah warna. Implementasi penghitungan pada data yang telah urut seperti berikut:

```
jwarna = 1;
for i = 1 : jum - 1
    if Data(i) ~= Data(i+1)
        jwarna = jwarna + 1;
    end
end
```

Berdasarkan kode di atas, nilai `jwarna` dinaikkan sebesar satu sekiranya suatu nilai dan nilai berikutnya tidak sama.

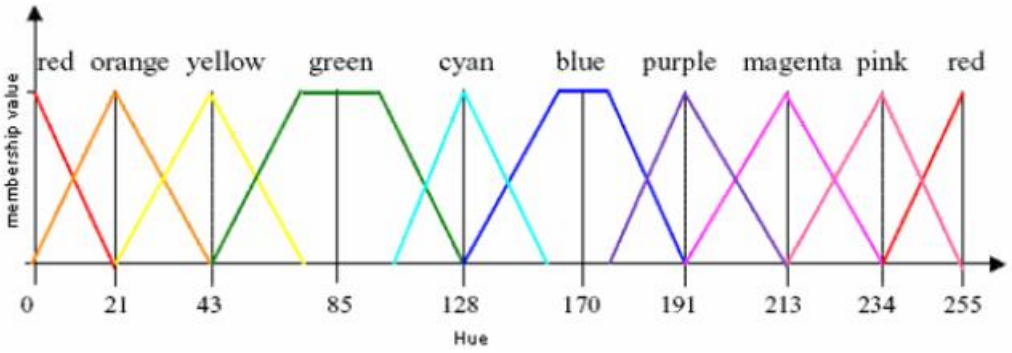
Contoh penggunaan fungsi `jumwarna`:

```
>> C = jumwarna('C:\Image\lapangan.png') ↵
C = 92475
>>
```

Hasil di atas menyatakan bahwa jumlah warna yang terkandung pada lapangan.png adalah 92.475.

12. Aplikasi Pencarian Citra Berdasarkan Warna Dominan

Agar pencarian menurut warna dominan seperti merah atau hijau dapat dilakukan, setiap piksel yang menyusun citra harus dapat dipetakan ke dalam warna alamiah semacam merah atau hijau. Hal ini dapat dilakukan kalau ruang warna yang digunakan berupa HSV. Pada sistem HSV, komponen *hue* sebenarnya menyatakan warna seperti yang biasa dipahami oleh manusia. Younes, dkk. (2007) membuat model *fuzzy* untuk menyatakan warna seperti terlihat pada Gambar 9.12.

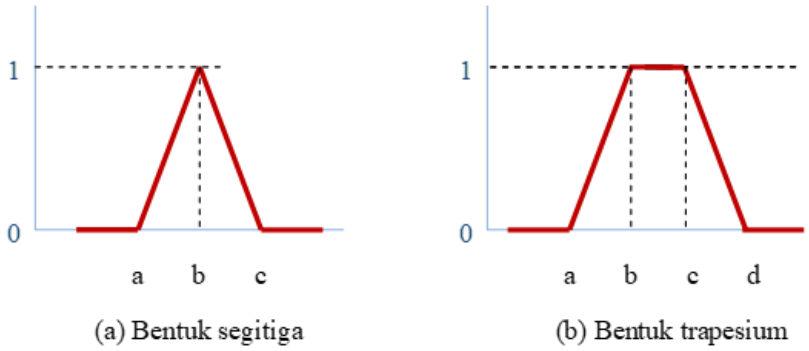


Gambar 9.12 Dimensi H

Berdasarkan Gambar 9.12, dimungkinkan untuk menerapkan *fuzzy logic* untuk menentukan suatu area warna beserta derajat keanggotaannya. Model tersebut didasarkan pada sumbu melingkar pada komponen *Hue* (H). Mengingat warna merah berada pada nilai H sama dengan nol, maka jangkauan warna merah berada di sekitar angka 0 dan 255.

Dua jenis fungsi keanggotaan *fuzzy* yang digunakan pada Gambar 9.12 berbentuk segitiga dan trapesium dan digambarkan kembali pada Gambar 9.13. Berdasarkan Gambar 9.13 tersebut, fungsi keangotaan berbentuk segitiga didefinisikan sebagai berikut:

$$\Lambda(x;a,b,c) = \begin{cases} 0 & x \le a, \\ (x-a)/(b-a) & a < x \le b, \\ (c-x)/(c-b) & b < x \le c, \\ 0 & x > c \end{cases}$$



Gambar 9.13 Fungsi keanggotan berbentuk segitiga dan trapesium

Adapun fungsi keanggotaan berbentuk trapesium didefnisikan sebagai berikut:

$$\Pi(x;a,b,c) = \begin{cases} 0 & x \le a, \\ (x-a)/(b-a) & a < x \le b, \\ 1 & b < x \le c, \\ (c-x)/(c-b) & c < x \le d, \\ 0 & x > d \end{cases}$$

Khusus untuk warna hitam, putih, dan abu-abu dilakukan penanganan secara khusus, yaitu dengan memperhatikan komponen S dan V. Sebagaimana tersirat pada Gambar 9.4, warna hitam diperoleh manakala V bernilai 0 dan warna putih diperoleh ketika S bernilai 0. Warna abu-abu diperoleh ketika S bernilai rendah. Semakin rendah nilai S maka warna abu-abu semakin terlihat tua. Secara garis besar, proses untuk memasukkan setiap piksel ke dalam suatu kategori warna (merah, hijau, dsb.) dilaksanakan melalui mekanisme seperti terlihat pada Gambar 9.14.

Gambar 9.14 Skema pemrosesan keanggotaan warna

Citra biner digunakan untuk menentukan area pada citra berwarna yang diproses khusus yang merupakan bagian daun. Dalam hal ini bagian yang berisi daun akan bernilai 1 pada citra biner dan 0 untuk latarbelakang. Selanjutnya, nilai H, S, V digunakan untuk menentukan kelompok warna piksel. Sebagai contoh, fungsi untuk menentukan warna hijau didefinisikan sebagai berikut:

```
function derajat=f_green(h, s, v)
   if (h==0) && (s==0)
       derajat = 0.0;
   else
      derajat = f_trapesium(43,65,105,128, h);
   end;
```

Dalam hal ini, f_trapesium adalah fungsi untuk mengimplementasikan Gambar 9.13(b). Definisinya seperti berikut:

```
function derajat=f_trapesium(a,b,c,d,h)

if (h>a) && (h<b)
        derajat=(h-a)/(b-a);
    else
        if (h>c) && (h<d)
            derajat=(d-h)/(d-c);
        else
            if (h>=b) && (h<=c)
                derajat=1.0;
            else
                derajat = 0.0;
            end
        end
    end
end
```

Pemrosesan yang dilakukan pada Gambar 9.14 kotak terbawah yaitu menghitung nilai untuk setiap komponen warna:

$$hijau = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f\_green(H_{ij}, S_{ij}, V_{ij})$$
$$merah = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f\_red(H_{ij}, S_{ij}, V_{ij})$$
$$kuning = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f\_yellow(H_{ij}, S_{ij}, V_{ij})$$

Setelah seluruh piksel pada daun diproses, setiap warna akan menyimpan komponen warna pada daun. Dengan demikian, warna yang memiliki nilai tertinggi menyatakan warna dominan pada daun.

Selanjutnya, *query* warna dominan dilakukan melalui

$warna\ domninan = \max(\text{hijau}, \text{merah}, \text{kuning})$

$query(warna)$: $warna\ domninan =$ warna

2 CONTOH PROGRAM

A. RGB KE HSV

Code :

```
%Muhammad Fajrin Aljabar
%F55121037
function [H,S,V] = RGBkeHSV(R,G,B)

R = double(R);
G = double(G);
B = double(B);

if max(max(R)) > 1.0 || max(max(G)) > 1.0 || ...
   max(max(B)) > 1.0
    R = double(R) / 255;
    G = double(G) / 255;
    B = double(B) / 255;
end

[tinggi, lebar] = size(R);
for m=1: tinggi
    for n=1: lebar
        minrgb = min([R(m,n) G(m,n) B(m,n)]);
        maxrgb = max([R(m,n) G(m,n) B(m,n)]);
        V(m,n) = maxrgb;
        delta = maxrgb - minrgb;

        if maxrgb == 0
            S(m,n) = 0;
        else
            S(m,n) = 1 - minrgb / maxrgb;
        end

        if S(m,n) == 0
            H(m,n) = 0;
        else
            SV = S(m,n) * V(m,n);

            if R(m,n) == maxrgb
                % Di antara kuning dan magenta
                H(m,n) = (G(m,n)-B(m,n)) / SV;
            elseif G(m,n) == maxrgb
                % Di antara cyan dan kuning
                H(m,n) = 2 + (B(m,n)-R(m,n)) / SV;
            else
                % Di antara magenta dan cyan
                H(m,n) = 4 + (R(m,n)-G(m,n)) / SV;
            end

            H(m,n) = H(m,n) * 60;
            if H(m,n) < 0
                H(m,n) = H(m,n)+360;
            end
        end
    end
end

% Konversikan ke jangkauan [0, 255] atau [0, 360]
H = uint8(H * 255/360);
S = uint8(S * 255);
V = uint8(V * 255);
```

Proses memasukkan angka RGB dan diubah ke HSV:

```
>> [H,S,V] = RGBkeHSV(50,90,30)

H =

  uint8

   71


S =

  uint8

   170


V =

  uint8

   90

>>
```

B. *RGB KE CMY*

Code :

```
%Muhammad Fajrin Aljabar
%F55121037
function [C,M,Y,K] = RGBkeCMY_F555121037(R,G,B)

R = double(R);
G = double(G);
B = double(B);

if max(max(R)) > 1.0 || max(max(G)) > 1.0 || ...
   max(max(B)) > 1.0
    R = double(R) / 255;
    G = double(G) / 255;
    B = double(B) / 255;
end

u = 0.5;
b = 1;
[tinggi, lebar] = size(R);
for m=1: tinggi
    for n=1: lebar
        Kb = min([(1-R(m,n)) (1-G(m,n)) (1-B(m,n))]);
        if Kb == 1
            C(m,n) = 0;
            M(m,n) = 0;
            Y(m,n) = 0;
        else
            C(m,n) = (1.0 - R(m,n) - u * Kb);
            M(m,n) = (1.0 - G(m,n) - u * Kb);
            Y(m,n) = (1.0 - B(m,n) - u * Kb);
            K(m,n) = b * Kb;
        end
    end
end

% Konversikan ke jangkauan [0,255]
C = uint8(C * 255);
M = uint8(M * 255);
Y = uint8(Y * 255);
K = uint8(K * 255);
```

Proses memasukkan angka RGB dan diubah ke CMY: