

BAB 2 LANDASAN TEORI

Pada bab ini akan dijelaskan tentang berbagai teori yang digunakan untuk melakukan penelitian ini. Penelitian ini pada dasarnya adalah segmentasi citra, maka diawali dengan penjelasan mengenai segmentasi, kemudian mengenai operasi-operasi pada matematika morfologi, kemudian dijelaskan pula morfologi pada citra keabuan, selanjutnya *fuzzy* morfologi dan yang terakhir mengenai pengolahan warna.

2.1 Segmentasi

Segmentasi citra yaitu pembagian citra menjadi beberapa daerah, berdasarkan sifat-sifat tertentu dari citra yang dapat dijadikan pembeda.

2.1.1 Daerah dan Objek

Suatu daerah dalam citra adalah sekumpulan piksel yang terkoneksi satu sama lain dan mempunyai sifat-sifat yang secara umum sama. Suatu citra mungkin mengandung beberapa objek dan tiap objek mungkin mengandung beberapa daerah yang berkorespondensi pada bagian berbeda pada objek tersebut. Agar daerah yang didapat benar-benar berkorespondensi dengan objek yang ingin dianalisis, maka pengetahuan tentang objek yang bersifat khusus harus digunakan agar objek dapat dipisahkan dari latar belakangnya.

Salah satu pendekatan untuk mempartisi citra ke dalam beberapa daerah adalah segmentasi berdasarkan daerah itu sendiri, yang biasanya mempunyai nilai intensitas yang berbeda dengan latar belakang.

Dua kriteria yang penting dalam segmentasi daerah yaitu:

1. Kemiripan nilai intensitas

Lebih banyak digunakan karena lebih dominan. Kemiripan nilai intensitas dua piksel dapat diukur berdasarkan perbedaan nilai intensitas keduanya.

2. Kedekatan posisi

Digunakan bila sebagian dari objek dan latar belakang mempunyai kemiripan nilai. Kedekatan posisi dua piksel dapat diukur melalui jaraknya, biasanya menggunakan jarak Euclidean.

Apabila citra mengandung beberapa objek, segmentasi secara sederhana dapat dilakukan dengan binerisasi melalui proses *thresholding* dan pelabelan komponen yang bertujuan memisahkan daerah milik sebuah atau beberapa objek dan latar belakang untuk menghasilkan citra biner.

2.1.2 Segmentasi Daerah

Idealnya suatu segmen citra mewakili satu atau sebagian dari objek. Jadi daerah adalah bagian dari citra dan segmentasi adalah pengelompokan piksel-piksel dalam suatu citra menjadi beberapa daerah.

Konversi suatu citra abu-abu menjadi citra biner adalah bentuk sederhana dari segmentasi citra, dimana citra dipartisi menjadi dua bagian. Algoritma binerisasi, untuk mendapatkan citra biner dapat digeneralisasikan untuk level lebih dari dua bagian atau dua kelompok piksel. Pada algoritma tersebut nilai *threshold* dipilih oleh perancang sistem berdasarkan kondisi citra yang akan dikenakan operasi binerisasi. Dalam membuat proses segmentasi yang cepat untuk berbagai kondisi, algoritma seharusnya mempunyai kemampuan untuk memilih nilai *threshold* yang sesuai secara otomatis menggunakan informasi intensitas pada citra. Ketidaktepatan dalam menentukan nilai batas untuk operasi binerisasi dapat menghasilkan citra biner yang tidak sesuai dengan harapan.

2.1.3 Binerisasi Manual

Binerisasi manual pada citra abu-abu dilakukan dengan menggunakan kode program, besarnya nilai batas untuk operasi binerisasi dimasukkan secara manual melalui keyboard. Dengan demikian proses binerisasi tidak dapat terjadi dengan sendirinya di dalam program, karena menunggu sampai nilai *threshold* dimasukkan oleh pemakai program. Operasi dengan menggunakan metode ini tidak dapat digunakan bila sejumlah citra abu-abu yang diproses mempunyai karakteristik intensitas yang berbeda, dan citra biner yang dihasilkan akan

langsung dijadikan data dalam proses berikutnya. Metode ini lebih cocok untuk program pengolahan citra yang bersifat tunda yang berarti operasi dapat dilakukan secara berulang-ulang hingga menghasilkan citra biner yang sesuai dengan keperluan.

2.1.4 Binerisasi Otomatis

Selain menggunakan binerisasi manual alternatif lainnya adalah dengan memberikan suatu nilai yang tetap dalam algoritma program, tetapi penggunaannya terbatas hanya pada sekelompok citra dengan sifat-sifat pencahayaan yang mirip satu sama lain. Untuk sejumlah citra yang mempunyai sifat-sifat pencahayaan berbeda satu sama lain, akan sulit menemukan satu nilai *threshold* yang cocok untuk semua kondisi citra yang akan dikenakan operasi binerisasi. Penentuan nilai *threshold* untuk melakukan operasi binerisasi dapat dilakukan secara otomatis dengan mempelajari terlebih dahulu karakteristik histogram dari citra yang akan dikenakan operasi binerisasi. Pengetahuan tentang karakteristik citra meliputi:

1. Sifat intensitas dari objek
2. Ukuran objek
3. Bagian dari citra yang ditempati objek
4. Jumlah dan jenis yang berbeda dari objek-objek yang muncul dalam citra.

Salah satu metode dalam binerisasi otomatis adalah *threshold ganda* (*double thresholding*). Metode ini berguna untuk operasi binerisasi pada citra dimana objek dari citra tersebut berada diantara 2 nilai yaitu batas atas dan batas bawah. Semua nilai diantara batas atas dan batas bawah yang ditentukan akan bernilai 1 yang berarti berwarna putih (daerah objek) dan sisanya akan bernilai 0 yang berarti berwarna hitam (daerah latar belakang).

2.2 Morphologi Biner

Morphologi digital merupakan suatu cara untuk menganalisis atau mendeskripsikan bentuk dari objek digital dalam citra digital. Gagasan yang mendasari morphologi digital adalah kenyataan bahwa citra mengandung

serangkaian piksel-piksel yang membentuk sekumpulan data dalam struktur dua dimensi dengan menggunakan operasi matematika sehingga piksel-piksel tersebut lebih mudah dikenali atau dihitung.

Morphologi biner ialah operasi-operasi morfologi diterapkan pada citra biner. Citra biner ialah citra yang hanya terdiri dari 2 jenis piksel yaitu hitam sebagai objek dan putih sebagai latar belakang.

Secara umum tujuan dari morfologi biner adalah memperbaiki bentuk objek, jadi operasi morfologi bukanlah operasi yang mempunyai tujuan analisis, melainkan suatu operasi yang memfasilitasi analisis yang akan dilakukan pada suatu objek. Contoh penghilangan *noise*.

Ide dasar dari matematika morfologi adalah teori himpunan. Dalam subbab ini akan dijelaskan operasi-operasi dasar seperti translasi, refleksi, dilasi, erosi, *opening*, *closing* dan juga operasi-operasi kompleks seperti *iterative rule* yang berguna untuk dekomposisi citra menjadi elemen yang lebih sederhana, penghilangan *noise*, dan *edge detection*.

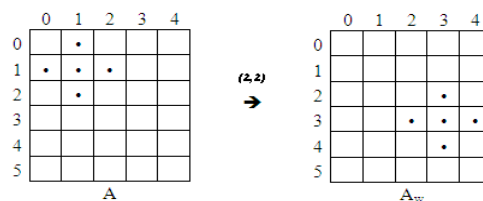
2.2.1 Translasi

Translasi berarti perpindahan. Misalkan A adalah kumpulan piksel dalam citra *binary* dan $w=(x,y)$ adalah sebuah titik koordinat, sedangkan A_w adalah A yang ditranslasikan sejauh w . Definisi secara matematika dituliskan sebagai berikut

$$A_w = \{(a,b) + (x,y) : (a,b) \in A\}$$

Persamaan 2.1 Translasi

Contoh translasi sejauh (2,2) bisa dilihat seperti gambar 2.1



Gambar 2.1 Translasi sejauh (2,2)

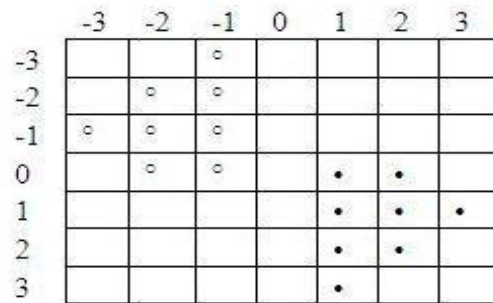
2.2.2 Refleksi

Refleksi didapat dengan mencerminkan setiap titik dalam A kepada origin (titik (0,0)). Definisi secara matematika dituliskan sebagai berikut

$$\hat{A} = \{(-x, -y) : (x, y) \in A\}$$

Persamaan 2.2 Refleksi

Contoh refleksi bisa dilihat seperti gambar 2.2



Gambar 2.2 Refleksi

2.2.3 Dilasi

Dilasi merupakan salah satu dari 2 operasi dasar dari matematika morfologi. Operasi dasar yang satunya adalah erosi yang akan dijelaskan pada subbab 2.2.4. Operasi morfologi lainnya dapat dibentuk oleh kedua operasi dasar ini.

Misalkan A dan B adalah himpunan dari piksel-piksel, dilasi A oleh B didefinisikan sebagai hasil *union* (gabungan) dari setiap titik di A di translasikan oleh setiap koordinat di B, secara matematika didefinisikan sebagai berikut

$$A \oplus B = \bigcup_{x \in B} A_x$$

Persamaan 2.3 Definisi Dilasi 1

Atau

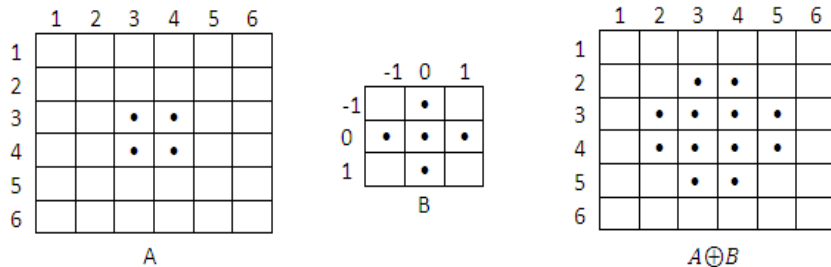
$$A \oplus B = \{(x, y) + (u, v) : (x, y) \in A, (u, v) \in B\}$$

Persamaan 2.4 Definisi Dilasi 2

Dilasi bersifat komutatif, berarti dilasi A oleh B akan sama hasilnya dengan dilasi B oleh A. Secara umum kita dapat berasumsi A adalah citra yang

akan proses dan B adalah himpunan kecil dari piksel-piksel yang biasa disebut elemen struktur.

Contoh dilasi bisa dilihat seperti gambar 2.3



Gambar 2.3 Dilasi

2.2.4 Erosi

Erosi A oleh B berisi semua piksel-piksel $w=(x,y)$ dimana B_w didalam A. Untuk menghasilkan erosi kita dapat menggerakkan B didalam A, cari semua tempat didalam A dimana B fit, dan disetiap tempat itu tandai piksel (0,0) dari B. Erosi secara matematika didefinisikan sebagai berikut

$$A \ominus B = \bigcap_{b \in B} A_b$$

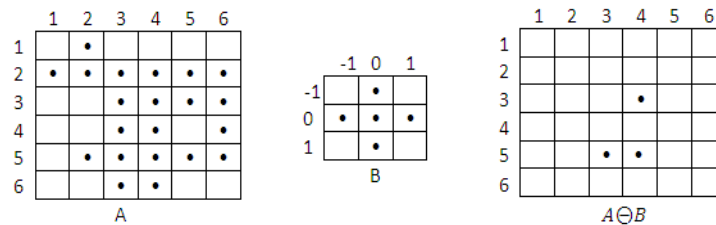
Persamaan 2.5 Definisi Erosi 1

Atau

$$A \ominus B = \{w : B_w \subseteq A\}$$

Persamaan 2.6 Definisi Erosi 2

Contoh erosi dapat dilihat pada gambar 2.4



Gambar 2.4 Erosi

2.2.5 Opening

Opening adalah erosi yang dilanjutkan dilasi dengan elemen struktur yang sama. Misalkan terdapat *image* A dan elemen struktur B, *opening* A oleh B didefinisikan sebagai berikut

$$A \circ B = (A \ominus B) \oplus B$$

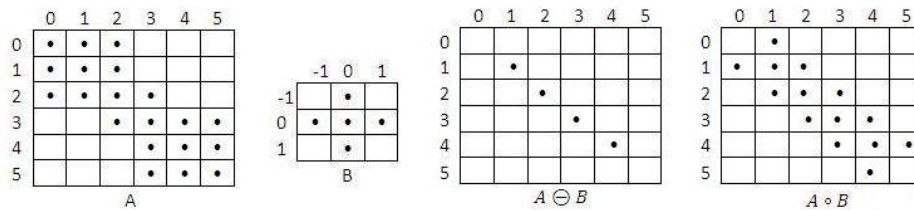
Persamaan 2.7 Definisi *Opening*

Operasi *opening* harus memenuhi properti di bawah ini:

1. $(A \circ B) \subseteq A$
2. $(A \circ B) \circ B = A \circ B \rightarrow$ Sifat idempoten
3. Jika $A \subseteq C$, maka $(A \circ B) \subseteq (C \circ B)$

Persamaan 2.8 Properti *Opening*

Operasi *opening* dapat dilihat seperti gambar 2.5



Gambar 2.5 *Opening*

2.2.6 Closing

Berkebalikan dengan *opening*, *closing* adalah dilasi yang dilanjutkan dengan erosi. Secara matematika *closing* didefinisikan sebagai berikut

$$A \bullet B = (A \oplus B) \ominus B$$

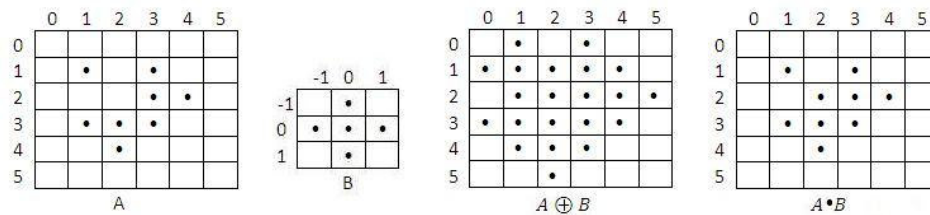
Persamaan 2.9 Definisi *Closing*

Operasi *closing* harus memenuhi properti dibawah ini:

1. $A \subseteq A \bullet B$
2. $(A \bullet B) \bullet B = A \bullet B \rightarrow$ Sifat idempoten
3. Jika $A \subseteq C$, maka $(A \bullet B) \subseteq (C \bullet B)$

Persamaan 2.10 Properti *Closing*

Operasi *closing* dapat dilihat seperti gambar 2.6



Gambar 2.6 Closing

2.2.7 Iterative Rule

Dekomposisi *morphological* adalah sebuah teknik berdasarkan perulangan erosi dilanjutkan dilasi yang mendekomposisi input kedalam komponen yang lebih sederhana.

Algoritma dari *iterative rule* sendiri ditemukan oleh Pitas dan Maglara (1991) yang diimplementasikan sebagai berikut

1. *Copy* input data kedalam S
2. Lakukan erosi berturut-turut terhadap S sehingga menghasilkan citra tererosi T, lakukan terus sampai nilai maksimum dari T lebih kecil dari nilai minimum dari S. Jumlah erosi dilambangkan dengan n.
3. Lakukan n dilasi terhadap T untuk menghasilkan U
4. Hitung residual R dengan mengurangi S terhadap U
5. *Copy* R ke S
6. Ulangi langkah nomor 2 sampai n=1

Algoritma 2.1 Iterative Rule

2.2.8 Penghilangan Noise (Noise Removal)

Noise merupakan bagian citra hasil segmentasi yang tidak diinginkan. Operasi *closing* menyebabkan lubang-lubang pada objek berhasil ditutup karena elemen struktur yang digunakan lebih besar dari lubang tersebut, namun bagian-bagian bukan objek lainnya juga menjadi semakin kompak sehingga semakin sulit untuk dibuang. Sebaliknya hasil operasi *opening*, bagian-bagian bukan objek akan terhapus, namun lubang-lubang pada objek nampak belum tertutup. Untuk

mendapatkan hasil penghilangan *noise* yang baik dapat dilakukan dengan melakukan operasi *opening* kemudian dilanjutkan dengan operasi *closing*.

$$NR = (A \circ B) \bullet B$$

Persamaan 2.11 Noise Removal

2.2.9 Edge Detection

Suatu objek yang berada dalam bidang citra dan tidak bersinggungan dengan batas bidang citra berarti objek tersebut dikelilingi oleh latar belakang. Pertemuan antara bagian objek dan bagian latar belakang disebut tepi objek (*edge*).

Dalam matematika morfologi pendeteksian tepi (*edge detection*) dapat dilakukan dengan *morphological gradien* seperti persamaan 2.12

$$MG = (A \oplus B) - (A \ominus B)$$

Persamaan 2.12 Morphological Gradient

2.3 Fuzzy Morphologi

Segmentasi citra sel positif pulsan imunohistokimia pada kanker payudara merupakan tugas yang memiliki komplikasi tinggi karena intensitas warna dari objek yang ingin disegmentasi sangat heterogen. Metode-metode segmentasi yang biasa digunakan seperti *thresholding* dan *region growing* tidak berkerja baik karena kedua metode ini hanya dapat bekerja dengan baik pada objek yang telah terdefinisi dengan baik dan homogen. Untuk mengakomodasi keheterogenan intensitas warna dari pulsan sel positif itu maka dibutuhkan parameter-parameter yang didefinisikan dalam algoritma segmentasi itu menjadi fleksibel dan penggunaan intuisi manusia yang memungkinkan melakukan segmentasi secara manual sebaiknya digabungkan dalam metode segmentasi yang dilakukan agar tercipta metode segmentasi yang baik dan metode inilah yang disebut *fuzzy* morfologi.

Fuzzy morfologi merupakan perluasan dari matematika morfologi berkaitan dengan perluasan penggunaan morfologi biner menjadi morfologi

keabuan. *Fuzzy* morfologi didefinisikan menggunakan teori himpunan dan logika *Fuzzy*. Teknik ini telah dimanfaatkan untuk memperluas area *fuzzy*, mensegmentasi objek, mendeteksi *edge* dan menganalisis struktur dengan sangat jelas.

Perbedaan *fuzzy* morfologi dengan morfologi biasa salah satunya adalah bagaimana menganggap elemen struktur. Pada morfologi biasa, elemen struktur dianggap sebagai citra, sedangkan pada *fuzzy* morphology dianggap sebagai himpunan *fuzzy*.

Dalam logika *fuzzy*, operator konjungsi (C) dan implikasi (I) diperluas dari domain boolean $\{0,1\} \times \{0,1\}$ ke domain baru yaitu $[0,1] \times [0,1]$. Sebuah fungsi $C: [0,1] \times [0,1] \rightarrow [0,1]$ ditulis dengan $C(s, t) = s \wedge t$ disebut sebagai *fuzzy conjunction* jika memenuhi properti $C(0,0) = C(1,0) = C(0,1) = 0$ dan $C(1,1) = 1$. fungsi $I: [0,1] \times [0,1] \rightarrow [0,1]$ ditulis dengan $I(s, t) = s \rightarrow t$ disebut sebagai *fuzzy implication* jika memenuhi properti $I(0,0) = I(0,1) = I(1,1) = 1$ dan $I(1,0) = 0$.

Menurut [6] dan [7] ada 2 formula yang biasa digunakan untuk untuk *fuzzy conjunction* dan *fuzzy implication* yaitu:

1. Godel-Brouwer

$$C(a,t) = a \wedge t$$

$$I(a,s) = s, \text{ jika } s < a, \text{ dan } I(a,s) = 1 \text{ jika } s \geq a$$

Persamaan 2.13 Konjungsi dan Implikasi Godel-Brouwer

2. Kleene-Dienes

$$C(a,t) = 0, \text{ jika } t \leq 1-a, \text{ dan } C(a,t) = t \text{ jika } t > 1-a$$

$$I(a,s) = (1-a) \vee s$$

Persamaan 2.14 Konjungsi dan Implikasi Kleene-Dienes

Dalam tugas akhir ini akan digunakan definisi konjungsi dan implikasi dari Kleene-Dienes.

Misalkan terdapat citra f dan sebuah elemen struktur B pada titik x dalam citra, maka *fuzzy erosi* dapat didefinisikan sebagai berikut

$$\mathcal{E}^F(f, B)(x) = \inf_{y \in f} \{I(B(y), f(y))\}$$

Persamaan 2.15 Fuzzy Erosi

Sedangkan *fuzzy dilasi* didefinisikan sebagai berikut

$$\delta^F(f, B)(x) = \sup_{y \in f} \{C(B(y), f(y))\}$$

Persamaan 2.16 Fuzzy Dilasi

Kemudian *fuzzy opening* didefinisikan sebagai berikut

$$\gamma^F(f, B) = \delta^F(\mathcal{E}^F(f, B), B)$$

Persamaan 2.17 Fuzzy Opening

Fuzzy closing sebagai berikut

$$\phi^F(f, B) = \mathcal{E}^F(\delta^F(f, B), B)$$

Persamaan 2.18 Fuzzy Closing

Fuzzy top hat oleh opening didefinisikan sebagai berikut

$$\text{Top} - \text{Hat}_\gamma^F = \gamma^F(f, B) - f$$

Persamaan 2.19 Fuzzy Top Hat oleh Opening

Dan terakhir *fuzzy top hat oleh closing* didefinisikan sebagai berikut

$$\text{Top} - \text{Hat}_\phi^F = f - \phi^F(f, B)$$

Persamaan 2.20 Fuzzy Top Hat oleh Closing

Dengan operasi-operasi *fuzzy* yang telah didefinisikan secara matematis diatas maka kemudian citra dapat dilakukan proses morfologi dengan elemen struktur B yang merupakan himpunan *fuzzy*.

2.4 Pengolahan Warna

Munculnya warna tergantung dari 3 faktor yaitu:

- Sifat pantulan spektrum dari permukaan yang menentukan bagaimana suatu permukaan memantulkan gelombang cahaya sehingga menampilkan warna.

- Kandungan spektrum dari cahaya yang menyinari yaitu warna-warna yang terkandung dalam suatu gelombang cahaya yang menyinari permukaan.
- Respon spektrum dari peralatan sistem visual seperti mata dan kamera.

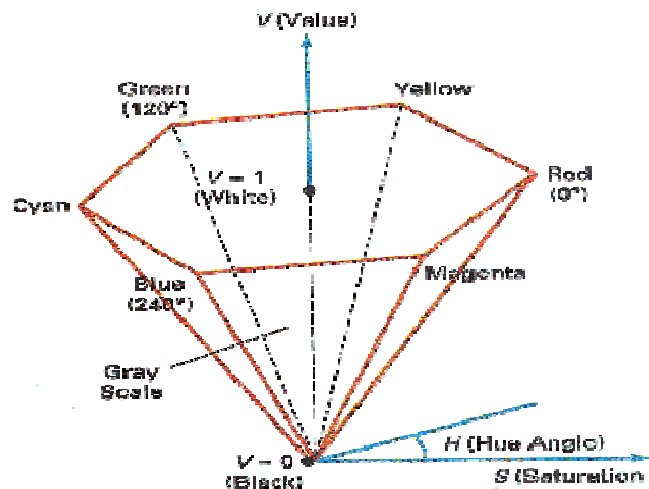
Untuk merepresentasikan warna maka dibuatlah model warna. Ada bermacam-macam model warna. Lihat table 2.1 untuk penjelasannya

Tabel 2.1 Model Warna dan Deskripsi

Model Warna	Deskripsi
RGB	Merah, hijau dan biru, sebuah model warna pokok yang biasa digunakan pada sistem display.
CMY(K)	Cyan, magenta, kuning, sebuah model warna subtraktif yang biasa digunakan pada printer.
YCbCr	Luminasi (Y) dan 2 komponen kromasiti (Cb dan Cr) biasa digunakan dalam siaran gelombang televisi.
HSV	Hue, saturation dan value berdasarkan persepsi manusia pada warna. Hue merupakan warna sejati. Saturation merupakan banyaknya warna putih terhadap warna yang dihasilkan komponen Hue. Value merupakan derajat kecerahan.
La*b*(CIELAB)	Model warna yang juga mendekati persepsi visual manusia

Representasi warna yang biasa digunakan adalah RGB, namun representasi ini tidak mendekati persepsi visual manusia tentang warna. Model warna yang mendekati persepsi visual manusia antara lain adalah La*b*(CIELAB) dan HSV. Permodelan warna dengan Lab sangat memakan waktu karena tidak ada transformasi linier yang dapat mentransformasikan dari RGB ke LAB, selain itu tidak ada fungsi invers *transform* yang unik karena sangat tergantung pada bagaimana mendefinisikan warna putih.

Model warna HSV merupakan model warna yang menggambarkan warna dengan konsep yang intuitif, bukan himpunan warna primer. Spesifikasi warna dengan konsep intuitif dilakukan dengan memilih warna spektrum dan jumlah warna putih dan hitam yang ditambahkan pada warna. Model warna HSV digambarkan sebagai *hexcone* seperti tampak pada Gambar 2.7.



Gambar 2.7 *Hexcone* Model Warna HSV

Komponen Hue dinyatakan sebagai sudut terhadap sumbu vertikal dari 0 sampai 360. titik-titik sudut heksagonal dipisahkan oleh sudut sebesar 60 derajat. Parameter Saturation atau S digunakan untuk menyatakan keaslian (purity) dari warna. Warna spektral asli memiliki nilai $S=1$ dan akan menurun sampai $S=0$ pada garis keabuan pada pusat *hexcone*. Nilai V bervariasi dari 0 sampai 1. Nilai $V=0$ ada didasar cone dan nilai $V=1$ ada dipuncak bidang *hexcone*.

Berdasarkan [3] konversi komponen warna RGB ke HSV dapat dilakukan dengan algoritma 2.2:

- 1 Setiap elemen dari citra RGB dibagi dengan 255 bertujuan agar input bernilai antara 0 sampai dengan 1
- 2 Untuk Setiap piksel dari citra RGB lakukan operasi berikut
 - 2.1 $V = \max(R,G,B)$
 - 2.2 $\delta = V - \min(R,G,B)$

$$2.3 \quad S = \frac{\delta}{V}$$

$$2.4 \quad \text{Jika } V = \delta = 0, \text{ maka } H = 0$$

$$2.5 \quad \text{Jika } V = R, \text{ maka } H = \frac{1}{6} \frac{G-B}{\delta}$$

$$2.6 \quad \text{Jika } V = G, \text{ maka } H = \frac{1}{6} \left(2 + \frac{B-R}{\delta} \right)$$

$$2.7 \quad \text{Jika } V = B, \text{ maka } H = \frac{1}{6} \left(4 + \frac{R-G}{\delta} \right)$$

$$2.8 \quad \text{Jika nilai } H < 0 \text{ maka } H = H + 1$$

- 3 Setiap elemen dari citra HSV dikali dengan 255 bertujuan agar input bernilai antara 0 sampai dengan 255

Algoritma 2.2 Konversi RGB ke HSV

Untuk mengembalikan ke citra warna RGB, diperlukan algoritma konversi dari HSV ke RGB seperti algoritma 2.3

- 1 Setiap elemen dari citra HSV dibagi dengan 255 bertujuan agar input bernilai antara 0 sampai dengan 1
- 2 Untuk Setiap piksel dari citra HSV lakukan operasi berikut
 - 2.1 $H' = [6H]$
 - 2.2 $F = 6H - H'$
 - 2.3 $P = V(1 - S)$
 - 2.4 $Q = V(1 - SF)$
 - 2.5 $T = V(1 - S(1 - F))$
 - 2.6 Jika $H' = 0$, maka $R = V; G = T; B = P$
 - 2.7 Jika $H' = 1$, maka $R = Q; G = V; B = P$
 - 2.8 Jika $H' = 2$, maka $R = P; G = V; B = T$
 - 2.9 Jika $H' = 3$, maka $R = P; G = Q; B = V$
 - 2.10 Jika $H' = 4$, maka $R = T; G = P; B = V$
 - 2.11 Jika $H' = 5$, maka $R = V; G = P; B = Q$
- 3 Setiap elemen dari citra RGB dikali dengan 255 bertujuan agar input bernilai antara 0 sampai dengan 255

Algoritma 2.3 Konversi HSV ke RGB