

BAB II

TINJAUAN PUSTAKA

2.1. Pengertian Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, dan hasil CT *Scan*. Sedangkan citra digital adalah citra yang dapat diolah oleh komputer (Sutoyo et al, 2009).

Sebuah citra digital dapat mewakili sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut *pixel* yaitu elemen terkecil dari sebuah citra. *Pixel* mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas atau warna dari *pixel* di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk Persamaan 4 (Sutoyo et al, 2009).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & f(1,1) & \dots & f(1, M - 1) \\ \dots & \dots & \dots & \dots \\ f(N - 1,0) & f(N - 1,1) & \dots & f(N - 1, M - 1) \end{bmatrix} \quad (4)$$

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas f(x,y), dimana harga x (baris) dan y (kolom) merupakan koordinat posisi dan f(x,y) adalah nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari *pixel* di titik tersebut. Pada proses digitalisasi (*sampling* dan kuantitas) diperoleh besar baris M dan kolom N hingga citra membentuk matriks M x N dan jumlah tingkat keabuan *pixel* G (Sutoyo et al, 2009).

2.1.1. Elemen Citra

Citra digital mengandung 6 elemen dasar. Elemen-elemen dasar tersebut dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam *computer vision*. Elemen-elemen dasar tersebut adalah sebagai berikut.

1. Kecerahan (*brightness*)

Kecerahan adalah intensitas cahaya rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkat kecerahan (*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi dengan jangkauan sebesar 10^{10} .

2. Kontras (*contrast*)

Kontras adalah sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah citra. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur (*contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada setiap *pixel* yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra.

4. Warna (*color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (violet) mempunyai panjang gelombang paling rendah.

Persepsi sistem visual manusia terhadap warna sangat relatif sebab dipengaruhi oleh banyak kriteria, salah satunya disebabkan oleh adaptasi yang menimbulkan distorsi.

5. Bentuk (*shape*)

Bentuk adalah informasi yang ditangkap oleh sistem visual manusia mengenai citra suatu objek. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan

segmentasi citra. Salah satu tantangan utama pada *computer vision* adalah merepresentasikan bentuk, atau aspek-aspek penting dari bentuk.

6. Tekstur (*texture*)

Tekstur adalah distribusi spasial dari derajat keabuan di dalam *pixel-pixel* yang bertetangga. Jadi, tekstur tidak dapat didefinisikan untuk sebuah *pixel*. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap *pixel*, melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipersepsi.

2.1.2. Transformasi Geometri

Transformasi geometri adalah proses perubahan hubungan spasial antara setiap *pixel* pada sebuah citra. Secara umum, terdapat 4 transformasi geometri pada citra yaitu sebagai berikut (Putra, 2010).

1. Translasi

Operasi translasi adalah memindahkan setiap elemen *pixel* citra input ke posisi baru pada citra *output* dimana dimensi dari kedua citra (citra masukan dan citra *output*) pada umumnya adalah sama. Posisi baru dari suatu *pixel* ditentukan dari nilai variabel translasi (p, q).

Secara umum operasi translasi melakukan perubahan dengan cara menambahkan koordinat awal dengan nilai variabel translasi seperti pada Persamaan 5 dan 6 (Putra, 2010).

$$x_2 = x_1 + p \quad (5)$$

$$y_2 = y_1 + p \quad (6)$$

Jika ukuran citra *output* diset sama dengan citra input maka bila terdapat posisi hasil yang berbeda di luar batas citra *output*, *pixel* tersebut tidak dipetakan. Untuk posisi citra *output* yang tidak memiliki nilai *pixel* diset dengan nilai 0 atau warna hitam.

2. Rotasi

Rotasi merupakan suatu transformasi geometri memindahkan nilai-nilai *pixel* dari posisi awal menuju ke posisi akhir yang ditentukan melalui nilai variabel rotasi sebesar θ terhadap sudut θ atau garis horizontal dari citra. Proses rotasi dapat dilakukan dengan Persamaan 7 dan 8 (Putra, 2010).

$$x_2 = \cos(\theta) * (x_1 + x_0) - \sin(\theta) * (y_1 + y_0) + x_0 \quad (7)$$

$$y_2 = \sin(\theta) * (x_1 + x_0) + \cos(\theta) * (y_1 + y_0) + y_0 \quad (8)$$

Dimana (x_0, y_0) adalah koordinat titik pusat dari citra input dan 0 adalah sumbu putar. Sumbu putar pada umumnya memiliki arah putar searah jarum jam dengan garis horizontal. Seperti halnya operasi translasi, hasil perhitungan posisi hasil rotasi dapat memberikan nilai di luar batas citra *output* (apabila ukuran citra *output* sama dengan citra input). Untuk kasus seperti itu, ada beberapa implementasi yang membiarkan nilai *pixel* tersebut tanpa dipetakan ulang dan ada yang memetakan ke citra *output* sehingga menyebabkan ukuran citra membesar.

3. Refleksi

Refleksi atau pencerminan adalah proses pengolahan citra secara geometri dengan memindahkan nilai-nilai *pixel* pada posisi awal (x_1, y_1) menuju ke posisi baru di (x_2, y_2) pada citra *output* sesuai dengan posisi pencerminan.

Posisi pencerminan ada tiga jenis yaitu pencerminan terhadap sumbu x, pencerminan terhadap sumbu y, dan pencerminan terhadap sumbu x dan y.

Pencerminan terhadap sumbu x di posisi x_c ditentukan dengan Persamaan 9 dan 10 (Putra, 2010).

$$x_2 = -x_1 + (2 * x_0) \quad (9)$$

$$y_2 = y_1 \quad (10)$$

Sedangkan untuk pencerminan terhadap sumbu y di posisi y_0 ditentukan dengan Persamaan 11 dan 12 (Putra, 2010).

$$x_2 = x_1 \quad (11)$$

$$y_2 = -y_1 + (2 * y_0) \quad (12)$$

Dan untuk pencerminan yang dilakukan terhadap kedua sumbu baik x dan y di posisi (x_0, y_0) ditentukan dengan Persamaan 13 dan 14 (Putra, 2010).

$$x_2 = -x_1 + (2 * x_0) \quad (13)$$

$$y_2 = -y_1 + (2 * y_0) \quad (14)$$

4. Penskalaan

Penskalaan adalah sebuah operasi geometri yang memberikan efek memperbesar atau memperkecil ukuran citra *input* sesuai dengan variabel penskalaan citranya. Ukuran baru hasil penskalaan didapat melalui perkalian antara ukuran citra *input* dengan variabel penskalaan. Proses penskalaan dapat dilakukan dengan Persamaan 15 dan 16 (Putra, 2010).

$$P_0 = S_p * P_i \quad (15)$$

$$L_0 = S_l * L_i \quad (16)$$

Dimana (P_i, L_i) adalah ukuran citra *input*, (P_0, L_0) adalah ukuran citra *output*, dan (S_p, S_l) adalah variabel penskalaan yang diinginkan. Jika variabel penskalaan bernilai lebih besar dari 1 maka akan memperbesar ukuran citra, sebaliknya apabila variabel penskalaan lebih kecil dari 1 maka hasilnya akan memperkecil ukuran citra.

a. Metode Penskalaan untuk Memperkecil Ukuran Citra

Proses penskalaan dengan memperkecil citra dapat dilakukan dengan menggunakan metode *subsampling* atau *interpolation*.

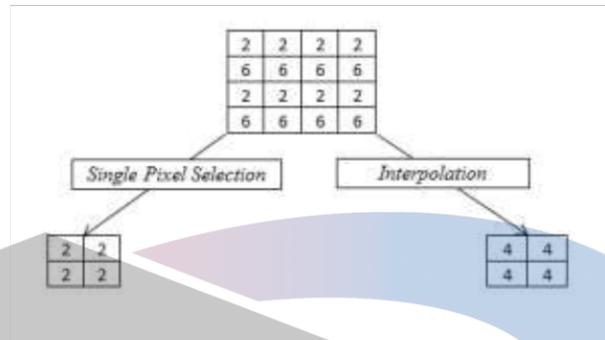
i. *Subsampling*

Metode ini memilih salah satu nilai dari suatu daerah citra untuk mewakili daerah tersebut. Pemilihan nilai dilakukan secara acak. Metode ini sangat sederhana dan cepat namun hasilnya kurang bagus, apalagi untuk proses dengan daerah *sampling* yang besar.

ii. *Interpolasi*

Metode kedua untuk memperkecil skala citra adalah *interpolation*. Metode ini menggunakan nilai rata-rata suatu region untuk mewakili region tersebut.

Gambar 2.1 berikut ini menunjukkan contoh pengolahan matrik untuk proses penurunan skala citra dengan metode *subsampling* dan *interpolation*.



Gambar 2.1 Metode Memperkecil Skala Citra (a) *Subsampling* (b) Interpolasi (Sumber : Putra, 2010)

Nilai 2 pada koordinat kiri atas pada citra hasil metode *subsampling* (lihat Gambar 2.1 (a)) diperoleh dengan memilih dari 4 nilai *pixel* pada pojok kiri atas citra asli, yaitu 2, 2, 6, 6. Aturan yang sama juga diberlakukan untuk nilai pada koordinat lain citra hasil *subsampling*.

Nilai 4 pada koordinat kiri atas pada citra hasil metode interpolasi (lihat Gambar 2.1 (b)) diperoleh dengan menghitung nilai rata-rata dari 4 nilai *pixel* pada pojok kiri atas citra asli, yaitu $(2 + 2 + 6 + 6)/4 = 4$. Aturan yang sama juga diberlakukan untuk nilai pada koordinat lain citra hasil interpolasi.

b. Metode Penskalaan dengan Memperbesar Ukuran Citra

Proses pembesaran dapat dilakukan dengan metode pengulangan *pixel* (*pixel replication*) dan interpolasi.

i. *Pixel Replication*

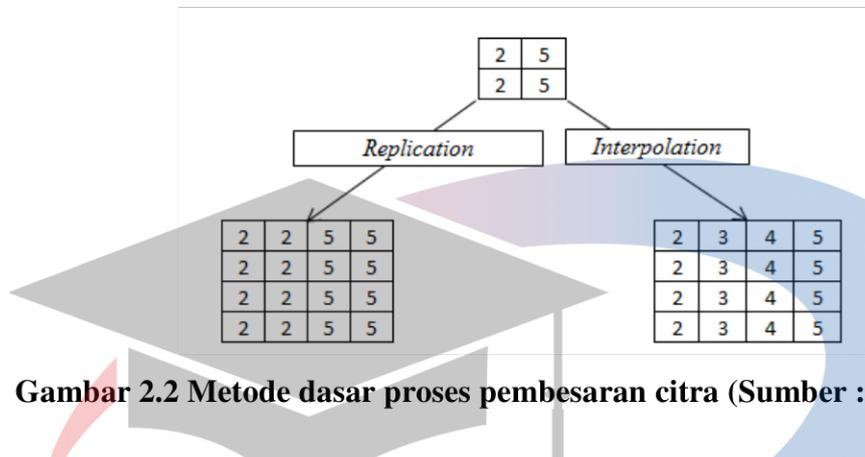
Pixel Replication akan mengulang nilai-nilai *pixel* pada daerah atau kelompok yang telah ditentukan sesuai dengan faktor penskalaannya.

ii. Interpolasi

Interpolasi pada proses perbesaran citra bekerja seperti halnya dalam proses penurunan skala citra namun kali ini digunakan untuk mengisi

nilai *pixel* hasil perbesaran. Nilai *pixel* ini dihitung dari rata-rata *pixel* tetangganya.

Gambar 2.2 berikut ini memperlihatkan proses perbesaran skala citra dengan menggunakan kedua metode di atas.



Gambar 2.2 Metode dasar proses pembesaran citra (Sumber : Putra, 2010)

2.2. *Image Retrieval*

Sistem Temu-Balik Citra (*Image Retrieval*) pada awal pengembangannya yaitu sekitar akhir 1970-an (Salton et al, 1975), masih menggunakan teks untuk menandai atau memberi keterangan (*annotation*) pada citra. Pertama-tama citra diberi keterangan berbentuk teks kemudian untuk melakukan proses temu-balik digunakan DBMS (*Database Management System*) berbasis teks. Pemberian keterangan tersebut memiliki kelemahan yaitu jika koleksi citra memiliki jumlah yang sangat besar, maka menjadi tidak efisien karena proses dilakukan secara manual dan keterangan yang diberikan pada citra bersifat subjektif, sangat tergantung pada persepsi pemberi keterangan. Untuk mengatasi persoalan tersebut maka, pada awal 1990-an mulai dikembangkan CBIR (*Content-Based Image Retrieval*) yang melakukan proses temu-balik berdasarkan muatan visual berupa komposisi warna yang dimiliki citra (Hsu et al, 1995).

2.2.1. Text Based Image Retrieval

Text Based Image Retrieval (TBIR) adalah sebuah teknik pencarian citra dengan menggunakan sekelompok informasi berupa teks yang menjelaskan suatu citra. Metode ini membandingkan sejumlah teks khusus seperti *keyword*. Metode

ini merupakan suatu teknik yang simpel, sehingga hanya membutuhkan waktu yang singkat untuk menampilkan hasil pencarian. Namun, untuk data yang besar, maka sangat diperlukan metode *indexing* yang efisien. Hal tersebut diperlukan, karena setiap citra harus melalui proses *indexing* ketika ingin dimasukkan ke dalam *database*. Proses ini akan membutuhkan waktu yang banyak (Salton et al, 1975).

2.2.2. Content Based Image Retrieval

Content Based Image Retrieval (CBIR) adalah sebuah teknik yang menggunakan fitur-fitur dari citra seperti warna, tekstur, bentuk dan fitur lainnya untuk mencari citra yang diinginkan pengguna dari *database* citra skala besar sesuai dengan *query* citra (Hsu et al, 1995). Proses CBIR mengekstrak fitur-fitur yang menjadi sifat dari citra. Nilai-nilai dari fitur citra akan disimpan di dalam *database*. Dalam sistem CBIR, *user* memberikan citra *query* dan sistem mengekstrak fitur yang ada, kemudian dibandingkan dengan nilai fitur dari citra yang ada di dalam *database*. Perbandingan dilakukan dengan menggunakan metode pengukuran jarak minimum.

Seiring dengan perkembangannya, terdapat banyak metode yang digunakan untuk menyelesaikan permasalahan CBIR, di antaranya *Ripplet Transform* (Xu et al, 2010), *Local Color Histogram* (Wang, 2001), dan *Edge Gradient Orientation Statistical Code* (Zeng et al, 2014). Dalam penelitian ini, metode yang digunakan adalah *Color Histogram for K-Means* (Lin et al, 2008) untuk mengolah fitur warna dan metode *Hu's Moment Invariants* (Huang dan Leng, 2010) untuk mengolah fitur bentuk.

2.2.2.1. Fitur Warna

Warna merupakan salah satu faktor yang paling menentukan ketika manusia melihat dan menganalisa sesuatu. Warna adalah hasil interaksi antara radiasi elektromagnetik (cahaya) dan permukaan objek (Waldman, G., 2002). Setiap warna mempunyai panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (violet)

mempunyai panjang gelombang paling rendah. Pada mata manusia, ada *photoreceptor* yang menerima cahaya untuk menentukan warna. *Photoreceptor* adalah sel kerucut, sel kerucut sangat sensitif terhadap 3 warna dasar, yaitu merah, hijau, biru (Chudler, E. H., 2013).

Pada CBIR, beberapa metode yang menggunakan fitur warna adalah sebagai berikut.

1. *Conventional Color Histogram*

Secara umum histogram menyatakan frekuensi kemunculan atau peluang keberadaan parameter dalam domain. Histogram warna menyatakan frekuensi kemunculan atau peluang setiap warna *pixel* di dalam sebuah citra. Untuk mengurangi waktu komputasi dan menghemat tempat penyimpanan, histogram warna menggunakan kuantisasi warna. Selain itu, kuantisasi warna juga dapat mengeliminasi komponen warna yang dapat dianggap sebagai *noise*. Banyaknya komponen kuantisasi (*bin*) dapat ditetapkan sesuai dengan kebutuhan pembuatan histogram. Peluang setiap bin dari histogram warna dapat ditentukan dengan Persamaan 17 dan 18 (Han dan Ma, 2002).

$$h_i = \frac{1}{N} \sum_{j=1}^N P_{i|j} \quad (17)$$

$$P_{i|j} = \begin{cases} 1, & \text{jika piksel ke } j \text{ dikuantisai ke bin } -i \\ 0, & \text{selainnya} \end{cases} \quad (18)$$

Keterangan :

h_i = nilai histogram bin ke- i

N = jumlah *pixel* dari citra

$P_{i|j}$ = peluang *pixel* ke- j dimasukkan ke bin- i .

2. *Fuzzy Color Histogram*

Fuzzy Color Histogram (FCH) merupakan salah satu metode untuk merepresentasikan informasi warna dalam citra digital ke dalam bentuk histogram. Metode ini mempertimbangkan persamaan warna pada tiap *pixel* warna yang terasosiasikan ke semua *bin* histogram melalui *fuzzy set membership function*. Pada FCH, satu warna dapat masuk ke dalam dua *bin* histogram atau

lebih dengan derajat keanggotaan yang berbeda. Inti dari FCH adalah bahwa tiap warna direpresentasikan dengan himpunan *fuzzy* (*fuzzy function*) dari himpunan *fuzzy*. Himpunan *fuzzy* F pada ruang ciri R^n didefinisikan oleh pemetaan $\mu_F : R^n \rightarrow [0,1]$ yang dinamakan *membership function*. Untuk setiap vector ciri $f \in R^n$, nilai dari $\mu_F(f)$ disebut derajat keanggotaan dari f terhadap himpunan *fuzzy* F . Nilai dari $\mu_F(f)$ yang mendekati 1 berarti vector ciri f lebih representatif terhadap himpunan *fuzzy* f (Han dan Ma, 2002).

3. *Color Histogram for K-Means (CHKM)*

Color Histogram for K-Means (Lin et al, 2008) merupakan metode pengelompokan warna-warna pada citra. Ada 2^{24} kemungkinan warna yang berbeda dari setiap *pixel* warna. Sebelumnya citra RGB terlebih dahulu telah melalui tahap *pre-processing* menjadi citra *grayscale*. Setiap *pixel* yang terdapat pada citra akan dikategorikan ke dalam 16 *cluster* menggunakan algoritma *K-Means*.

K-means clustering merupakan sebuah metode *clustering* berbasis jarak yang membagi data-data ke dalam sejumlah *cluster*, dimana proses tersebut dilakukan dengan memperhatikan kumpulan dari data-data yang akan dikelompokkan. Tahapan dalam algoritma *k-means clustering* adalah sebagai berikut (Malakar dan Mukherjee, 2013).

1. Definisikan jumlah k *cluster*
2. Inialisasi k *centroid* (pusat *cluster*) secara *random* atau dipilih dari k objek data.
3. Untuk setiap komponen data, hitung dan tandai jarak ke *centroid* awal kemudian masukkan data tersebut ke *centroid* yang paling dekat jaraknya.
4. Hitung dan ubah kembali *centroid* tiap *cluster* sebagai rata-rata dari seluruh kelompok *cluster* tersebut.
5. Cek semua data dan masukkan setiap data yang terdekat dengan *centroid* baru. Jika anggota tiap *cluster* tidak berubah (konvergen), maka langkah berhenti dan jika masih berubah, kembali ke langkah 3.

Selanjutnya, nilai CHKM dapat dihitung dengan Persamaan 19 (Lin et al, 2008).

$$g^k = \frac{N_k}{N} \quad (19)$$

Keterangan:

g^k = nilai CHKM untuk *cluster* ke-k

N_k = jumlah *pixel cluster* ke-k

N = total jumlah *pixel* pada citra

Tingkat kemiripan fitur warna antara citra *query* dengan citra di *database* dengan *color histogram for k-means* akan dihitung dengan Persamaan 20 (Lin et al, 2008).

$$\Delta^{CHKM} = \sum_{k=1}^{16} \left| \frac{g_k^q - g_k^d}{g_k^q + g_k^d + v} \right| \quad (20)$$

Keterangan :

g_k^q = nilai CHKM kluster ke-k pada citra *query*

g_k^d = nilai CHKM kluster ke-k pada citra *database*

v = konstanta

2.2.2.2. Fitur Bentuk

Bentuk adalah informasi yang ditangkap oleh sistem visual manusia mengenai citra suatu objek. Fitur bentuk merupakan salah satu fitur kunci dari *image retrieval*. Fitur bentuk menghasilkan ekstraksi yang efisien dan efektif. Hasil ekstraksi yang sama merepresentasikan bentuk yang sama dari citra terkait (Datta et al, 2005).

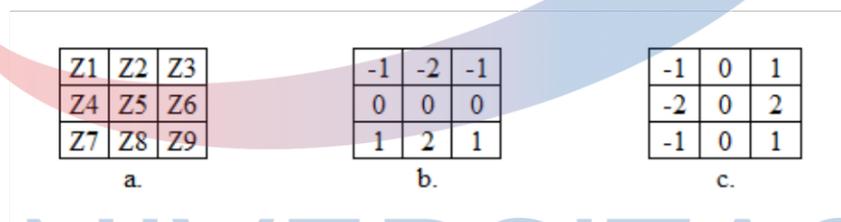
Pada CBIR, beberapa metode yang menggunakan fitur bentuk adalah sebagai berikut.

1. *Edge Detection*

Pendeteksian tepi (*edge detection*) adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edge*), yakni garis yang membatasi dua wilayah citra homogen berdasarkan tingkat kecerahan yang berbeda. Pendeteksian tepi merupakan salah satu langkah untuk meliput informasi di dalam citra. Tepi menampilkan batas-batas objek, oleh karena itu tepi berguna dalam proses segmentasi dan identifikasi objek di dalam citra.

Tujuan operasi pendeteksian tepi adalah untuk memperjelas garis batas suatu objek dari latar belakang di dalam citra. Karena tepi termasuk ke dalam komponen berfrekuensi tinggi, maka pendeteksian tepi dapat dilakukan dengan penapis lolos tinggi (*high pass filter*). Beberapa operator pendeteksi tepi yang umum digunakan, antara lain *Sobel*, *Prewitt*, *Roberts*, *Laplacian of a Gaussian (LoG)*, *Zero Crossings* dan *Canny* (Gonzalez et al, 2004).

Pendeteksi tepi Sobel (*Sobel edge detector*) merupakan salah satu metode pendeteksi tepi yang umum digunakan. Pendeteksi tepi Sobel menggunakan dua buah matriks konvolusi berukuran 3 x 3. Matrik konvolusi pada *pixel-pixel* tetangga berukuran 3 x 3, yang diperlihatkan pada Gambar 2.3.a. Matriks konvolusi pertama digunakan untuk mengestimasi *gradient* pada arah sumbu x, diperlihatkan pada Gambar 2.3.b. Matrik konvolusi kedua digunakan untuk menentukan *gradient* pada arah sumbu y, diperlihatkan pada Gambar 2.3.c (Gonzalez et al, 2004).



Gambar 2.3 Matriks Konvolusi Pendeteksi Tepi Sobel (a) Matrik Citra Tetangga (b) Matrik Konvolusi Arah Sumbu x (c) Matrik Konvolusi Arah Sumbu y (Sumber : Gonzalez et al, 2004)

Hasil operasi konvolusi dalam arah sumbu x terhadap citra I dinyatakan dengan G_x , dan hasil operasi konvolusi dalam arah sumbu y terhadap citra I dinyatakan dengan dinyatakan dengan G_y . Sehingga dengan menggunakan matrik konvolusi maka nilai G_x dan G_y berturut turut dapat dihitung dengan Persamaan 21 dan 22. Magnitudo (*edge strength*) dari gradien dapat dihitung dengan Persamaan 23 atau 24. Sebuah *pixel* akan dianggap sebagai tepi (bernilai satu) jika nilai magnitudonya lebih besar dari nilai ambang (*threshold*) yang ditetapkan. Arah tepi (*edge direction*) dapat dihitung menggunakan Persamaan 25. Setelah nilai *edge direction* diperoleh, langkah selanjutnya adalah menentukan *pixel-pixel* citra yang merupakan garis (*edge*). Sebuah *pixel* akan dianggap sebagai *edge* jika

nilai magnitudonya lebih besar dari nilai *threshold* yang ditetapkan (Gonzalez et al, 2004).

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (21)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_3) \quad (22)$$

$$|G| = |G_x| + |G_y| \quad (23)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (24)$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (25)$$

2. Transformasi Wavelet

Dalam proses ekstraksi ciri dilakukan transformasi citra untuk mendapatkan informasi yang lebih jelas yang terkandung dalam citra tersebut. Transformasi atau alih ragam citra pada bagian ini adalah perubahan ruang (*domain*) citra ke domain lainnya. Melalui proses transformasi, citra dapat dinyatakan sebagai kombinasi linear dari sinyal dasar (*basic signals*) yang sering disebut dengan fungsi basis (Putra, 2010).

Wavelet diartikan sebagai *small wave* atau gelombang singkat. Transformasi wavelet akan mengkonversi suatu sinyal ke dalam sederetan wavelet. Gelombang singkat tersebut merupakan fungsi yang terletak pada waktu berbeda. Transformasi wavelet mampu memberikan informasi frekuensi yang muncul dan memberikan informasi tentang skala atau durasi atau waktu. Wavelet dapat digunakan untuk menganalisa suatu bentuk gelombang (sinyal) sebagai kombinasi dari waktu (skala) dan frekuensi.

Proses transformasi pada wavelet dapat dicontohkan sebagai berikut, citra yang semula ditransformasikan dibagi (didekomposisi) menjadi empat sub citra baru untuk menggantikannya. Setiap sub citra berukuran $\frac{1}{3}$ kali dari citra asli. Tiga sub citra pada posisi kanan atas, kanan bawah dan kiri bawah akan tampak seperti versi kasar dari citra asli karena berisi komponen frekuensi tinggi dari citra asli. Sedangkan untuk sub citra pada posisi kiri atas tampak seperti citra asli dan lebih halus, karena berisi komponen frekuensi rendah dari citra asli. Sub citra pada

bagian kiri atas (frekuensi rendah) tersebut dibagi lagi menjadi empat sub citra baru. Proses diulang sesuai dengan level transformasi yang digunakan. Untuk lebih jelasnya ditunjukkan pada Gambar 2.4.

LL2	LH2	LH1
HL2	HH2	
HL1		HH1

Gambar 2.4 Proses Transformasi Wavelet (Sumber : Putra, 2010)

Hasil dekomposisi level 1 dinyatakan dalam LH1, HL1, dan HH1. Sementara itu L1 telah didekomposisi lagi menjadi LL2, LH2, HL2 dan HH2. Pada citra 2 dimensi, terdapat dua cara untuk mentransformasi atau mendekomposisi nilai-nilai *pixel*-nya, yaitu dekomposisi standar dan tak standar. Keduanya diperoleh berdasarkan transformasi wavelet 1 dimensi.

Dekomposisi standar menggunakan transformasi wavelet 1 dimensi pada tiap baris citra dan kemudian pada tiap kolom. Dekomposisi tak standar diperoleh dengan mengkombinasikan pasangan transformasi baris dan transformasi kolom secara bergantian. Pada langkah pertama diterapkan transformasi wavelet 1 dimensi pada kolom, proses tersebut diulang dengan level yang diinginkan.

3. Hu's Moment Invariants

Dalam metode ini, sebuah citra akan melalui 2 tahapan sebagai berikut.

a. Konversi Citra Grayscale ke Citra Biner dengan Metode Otsu

Citra biner adalah citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih (Putra, 2014). Proses konversi citra *grayscale* ke citra biner dilakukan dengan menggunakan metode Otsu (Otsu, 1979). Citra merupakan fungsi 2D intensitas *grayscale* dan mengandung N *pixel* dengan derajat keabuan 0 hingga L . Peluang dari sebuah tingkat keabuan didefinisikan dalam Persamaan 26 (Otsu, 1979).

$$P_i = \frac{f_i}{N} \quad (26)$$

Keterangan:

P_i = Peluang dari *gray level* i

f_i = jumlah tingkat keabuan i

Dalam kasus *bi-level thresholding* dari sebuah citra, *pixel-pixel* dibagi menjadi 2 bagian, C_1 dengan tingkat keabuan $[1, \dots, t]$ dan C_2 dengan tingkat keabuan $[t+1, \dots, L]$. Kemudian, peluang distribusi tingkat keabuan dibagi menjadi 2 kelas yaitu $C_1 = p_1 / \omega_1(t), \dots, p_t / \omega_1(t)$ and $C_2 = p_{t+1} / \omega_2(t), p_{t+2} / \omega_2(t), \dots, p_L / \omega_2(t)$ dimana ω_1 dan ω_2 dinyatakan berturut-turut dalam Persamaan 27 dan 28 (Otsu, 1979).

$$\omega_1(t) = \sum_{i=1}^t p_i \quad (27)$$

$$\omega_2(t) = \sum_{i=t+1}^L p_i \quad (28)$$

Keterangan :

ω_1 = distribusi kumulatif dari tingkat keabuan $[1, \dots, t]$

ω_2 = distribusi kumulatif dari tingkat keabuan $[t+1, \dots, L]$

Selanjutnya, mean dari C_1 dan C_2 didefinisikan berturut-turut dalam Persamaan 29 dan 30 (Otsu, 1979).

$$\mu_1 = \sum_{i=1}^t i p_i \quad (29)$$

$$\mu_2 = \sum_{i=t+1}^L i p_i \quad (30)$$

Keterangan :

μ_1 = mean dari dari C_1 dengan tingkat keabuan $[1, \dots, t]$

μ_2 = mean dari dari C_2 dengan tingkat keabuan $[t+1, \dots, L]$

Kemudian, mean dari intensitas untuk keseluruhan citra didefinisikan dalam Persamaan 31 (Otsu, 1979).

$$\mu_T = \sum_{i=1}^L i p_i \quad (31)$$

Variance dari C_1 dan C_2 didefinisikan oleh Otsu dalam Persamaan 32 (Otsu, 1979).

$$\sigma_B^2 = \frac{[\mu_T \omega_1(t) - \mu_1]^2}{\omega_1(t)[1 - \omega_1(t)]} \quad (32)$$

b. Menghitung Hu's Moment Invariants

Moment invariants banyak digunakan dalam pengenalan citra. *Hu's Moment Invariants* (Huang dan Leng, 2010) menghasilkan 7 nilai yang konstan terhadap transformasi geometri rotasi, skala dan translasi. Citra dengan ukuran $M \times N$ memiliki fungsi keabuan $f(x,y)$, maka momen dua-dimensi citra tersebut dapat didefinisikan dalam persamaan 33 (Huang dan Leng, 2010).

$$m_{pq} = \sum_{m=1}^M \sum_{n=1}^N x^p y^q f(x, y) \quad (33)$$

Keterangan :

x = nilai dari posisi *pixel* pada sumbu x

y = nilai dari posisi *pixel* pada sumbu y

$f(x,y)$ = derajat keabuan dari *pixel* pada posisi (x,y)

Selanjutnya, dihitung nilai *central moment*. Nilai dari *central moment* yang diperoleh bersifat konstan terhadap transformasi geometri translasi. *Central moment* didefinisikan dalam persamaan 34 (Huang dan Leng, 2010).

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (34)$$

Keterangan :

μ_{pq} = *central moment*

\bar{x} = titik pusat lokasi objek pada sumbu x

\bar{y} = titik pusat lokasi objek pada sumbu y

dimana $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$.

Keterangan :

m_{00} = momen ordo 0

m_{10} = momen ordo 1

m01 = momen ordo 1

Selanjutnya, *central moment* dinormalisasi agar konstan terhadap transformasi geometri skala dan rotasi yang didefinisikan dalam persamaan 35 (Huang dan Leng, 2010).

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\rho}} \quad (35)$$

Keterangan :

η_{pq} = *central moment* yang sudah dinormalisasi

dimana $\rho = (p+q)/2 + 1$

Pada akhirnya, 7 nilai *Hu's Moment Invariants* didefinisikan dalam Persamaan 36 (Huang dan Leng, 2010).

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} \\ &\quad - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} \\ &\quad + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} \\ &\quad - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (36)$$

Keterangan :

$\phi_1 - \phi_7$ = *moment invariant* ke-1 hingga ke-7

2.3. Tingkat Kemiripan

Tingkat kemiripan antara citra *query* dengan citra database merupakan masalah kedua terpenting dalam CBIR. Perbedaan antara citra *query* dengan citra database dihitung berdasarkan hasil ekstraksi fitur kedua citra. Hasil perhitungan yang kecil menunjukkan tingkat kemiripan yang tinggi antara kedua citra dan sebaliknya (Malik dan Baharudin, 2012).

2.3.1. Canberra Distance

Untuk setiap nilai 2 vektor yang akan dicocokkan, Canberra Distance membagi absolute selisih 2 nilai dengan jumlah dari absolute 2 nilai tersebut (Putra, 2010). Hasil dari dua nilai yang dicocokkan lalu dijumlahkan untuk mendapatkan *Canberra Distance*. Jika koordinat nol-nol((0,0)) diberikan definisi dengan $0/0=0$. *Canberra Distance* ini sangat peka terhadap sedikit perubahan dengan kedua koordinat mendekati nol. *Canberra distance* dinyatakan dalam Persamaan 37 (Singh, dan Hemachandran, 2012).

$$d(i, j) = \sum_{i=1}^n \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|} \quad (37)$$

Keterangan :

$d(i, j)$ = tingkat perbedaan

n = jumlah vektor

x_{ik} = vektor citra input

x_{jk} = vektor citra pembandingan /output

2.3.2. Euclidean Distance

Euclidean Distance adalah metrika yang paling sering digunakan untuk menghitung kesamaan dua vektor. Rumus euclidean Distance adalah akar dari kuadrat perbedaan 2 vektor. *Euclidean distance* dinyatakan dalam Persamaan 38 (Smith dan Chang, 1996).

$$d(i, j) = \sqrt{\sum_{i=1}^n [x_{ik} - x_{jk}]^2} \quad (38)$$

Keterangan :

$d(i, j)$ = tingkat perbedaan

n = jumlah vektor

x_{ik} = vektor citra *input*

x_{jk} = vektor citra pembandingan *loutput*