BAB II
LITERATURE REVIEW

2.1 Pengertian Citra

Citra merupakan gambar dua dimensi yang memiliki variable *f(x,y)*, dimana x dan y adalah koordinat spasial dan nilai dari *f(x,y)* adalah intensitas citra atau pada *grey level* pada koordinat dengan tingkat kecerahan dari citra di titik tersebut (Gonzalez & Woods, 2002).

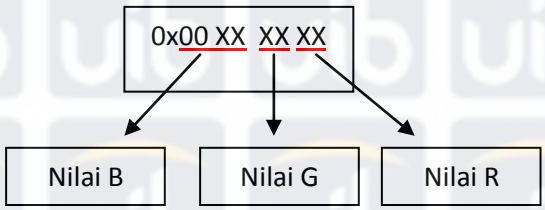
Dilihat dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Sumber cahaya yang menerangi objek tersebut memantulkan berkas cahaya dan pantulan cahaya di tangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, scanner, sensor satelit dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam (Munir, 2004) .

Menurut Munir (2004), citra di bagi menjadi 2 macam, yaitu citra diam (citra yang tidak bergerak / *still images*) dan citra bergerak (*moving images*). Citra diam adalah sebuah citra atau gambar yang tidak bergerak ataupun citra diam, sedangkan citra bergerak (*moving images*) merupakan citra diam yang ditampilkan secara beruntun sehingga memberikan kesan pada kita bahwa gambar tersebut bergerak.

2.2 Pengertian Pengolahan Citra

Meskipun sebuah citra yang sudah terekam sering kali mengandung cacat, *noise* (derau), warnanya terlalu kontras, kurang tajam, kabur, tentu saja membuat citra lebih sulit diinterpretasi. Untuk menghasilkan suatu citra yang bagus maka perlu manipulasi menjadi citra lain yang kualitasnya lebih baik, dalam hal ini dengan cara pengolahan citra (*image processing*).

Dalam pengolahan citra, warna dipresentasikan dengan nilai *hexadecimal* dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah warna seperti gambar dibawah ini, variable 0x00 menyatakan angka dibelakangnya adalah *hexadecimal*.



Gambar 2.1. Nilai Gambar RGB Pada *Hexadecimal*
Sumber: Budi Gunawan

Berikut ini adalah pendapat Gonzalez dan Woods (2001) tentang pengertian pengolahan citra :

When x, y and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels.

Masukannya adalah citra dan output keluarannya juga merupakan citra tapi dengan kualitas yang lebih bagus dari pada citra masukan. Pengolahan citra dapat dikelompokkan dalam dua jenis yaitu:

1. Memperbaiki kualitas gambar.
2. Mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis.

Di bidang ilmu komputer pengolahan citra dibagi menjadi 3 bagian dengan tujuan yang berbeda, antara lain:

1. Grafika Komputer (*Computer Graphics*).
2. Pengolahan Citra (*Image Processing*).
3. Pengenalan Pola (*Pattern Recognition*).

Grafika komputer bertujuan untuk menghasilkan citra dengan primitif-primitif geometri seperti garis, lingkaran, dan sebagainya. Primitif–primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar seperti koordinat titik, panjang garis, jari–jari lingkaran, tebal garis, warna, dan sebagainya.

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia ataupun mesin. Masukan input berupa citra tetapi keluarannya berupa citra juga, namun dengan kualitas yang lebih baik.

Pengenalan pola mengelompokkan data numerik dan simbolik secara otomatis oleh mesin. Tujuan pengelompokan adalah untuk mengenali suatu objek didalam citra. komputer menerima masukan berupa citra objek yang akan diidentifikasi,

memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra.

Dalam hal ini penulis akan menggunakan sistem pengenalan pola atau *pattern recognition* sebagai salah satu alat bantu dalam melakukan penelitian tersebut karena yang di deteksi merupakan objek berupa kaleng.

2.3 *Pattern Recognition*

Pattern atau pola merupakan entitas yang terdefinisi melalui ciri-cirinya, yaitu dimana komputer dapat mengenali suatu pola dan membandingkan kemiripan dengan pola lainnya.

Menurut Al Fatta (2009a), pola adalah komposit atau ciri yang merupakan sifat dari sebuah objek. Sehingga setiap objek atau benda memiliki ciri yang berbeda. Dalam klasifikasinya pola berupa sepasang variabel (x,y).

Beberapa contoh pola yang sudah berkembang:

1. Huruf, memiliki ciri-ciri seperti tinggi, tebal, titik sudut, dan lengkungan garis.
2. Suara, memiliki ciri-ciri seperti amplitudo, frekuensi, nada, dan intonasi.
3. Tanda tangan, memiliki ciri-ciri seperti panjang, kerumitan, dan tekanan.
4. Sidik jari, memiliki ciri-ciri seperti lengkungan, dan jumlah garis.

Khusus pada pola yang terdapat didalam citra, ciri-ciri yang dapat diperoleh berasal dari informasi :

1. Spasial, seperti intensitas piksel dan histogram.

2. Tepi, seperti arah dan keakuratan.
3. Kontur, seperti garis, ellips, dan lingkaran.
4. Wilayah atau bentuk, seperti keliling, luas dan pusat massa.
5. Hasil transformasi *Fourier*, seperti frekuensi.

Pengenalan objek pada dasarnya untuk membedakan objek yang satu dengan objek yang lainnya berdasarkan ciri masing-masing.

Pengenalan pola sendiri merupakan cabang dari kecerdasaan buatan (*Artificial Inteligence*). Beberapa definisi tentang pengenalan pola, diantaranya:

1. Penentuan suatu objek fisik atau kejadian ke dalam salah satu atau beberapa kategori. (Al Fatta, 2009b).
2. Ilmu pengetahuan yang menitikberatkan pada deskripsi dan kalsifikasi (pengenalan) dari suatu pengukuran dikemukakan oleh Schalkoff (1992).

Pengenalan pola merupakan cabang dari kecerdasan buatan yang saat ini berkembang pesat untuk mendukung aspek keamanan suatu sistem. Saat ini, aplikasi-aplikasi pengenalan pola juga sudah sangat beragam, diantaranya:

1. *Voice Recognition* yang menggunakan pengenalan suara sebagai kunci bagi pengguna sistem.
2. *Fingerprint Identification* yang menggunakan pengenalan sidik jari sebagai kunci telah dipakai secara luas sebagai pengganti password atau pin untuk mengakses sistem tertentu.

3. *Face Identification* yang menggunakan pengenalan wajah sebagai kunci bagi pengguna sistem, bahkan saat ini badan penegak hukum sedang mengembangkan sistem untuk mengidentifikasi para buronan dengan melakukan scanning pada wajah para pelaku kejahatan yang sudah di database-kan berdasarkan foto pelaku kejahatan tersebut.
4. *Handwriting Identification* yang menggunakan pengenalan tulisan yang telah secara luas digunakan oleh sistem perbankan untuk membuktikan pelaku transaksi adalah orang yang benar-benar berhak.
5. *Optical Character Recognition* (OCR) yang secara luas digunakan pada *counter* pengecekan barang.
6. *Robot Vision* yang digunakan oleh apliakasi *robotic* dalam mengenali objek tertentu pada lingkungan yang unik.

2.4 Pemrosesan Awal Pada *Image*

Pada awal suatu citra di tangkap kamera akan diproses dalam tahap pemrosesan image seperti tahap-tahap mengubaha suatu citra yang RGB menjadi *grayscale image* dan dilakukan proses *filtering*. Menurut Thiang dan Leonardo Indrotanoto (2008) pengubahan *image* dari RGB menjadi format *grayscale* dilakukan dengan menggunakan metode *illuminance grayscale* yang dipresentasikan dalam Persamaan 2.1 berikut ini:

$$Gray = 0.299R + 0.587G + 0.114B \quad \text{.....(2.1)}$$

Selain itu juga dapat dilakukan dengan *mean grayscale* dengan menghitung rata-rata nilai RGB yang ada. Dapat direpresentasikan dengan menggunakan Persamaan 2.2 berikut ini:

$$Gray = \frac{R+G+B}{3} \quad \text{.....(2.2)}$$

Setelah suatu citra telah menjadi *grayscale image* maka akan dilakukan proses binarisasi.

2.5 Metode Pengenalan Objek

Untuk metode pengenalan objek terdapat beberapa metode seperti dengan menggunnakan metode *Region Of Interest* (ROI), *Interpretation trees*, *Hypothesize and test*, *Pose consistency*, *Pose clustering Invariantce*, *Geometric hashing*, *Scale-invariantt feature transform* (SIFT*), Speeded Up Robust Features* (SURF).

Pada penelitian ini digunakan metode *Scale-invariantt feature transform* (SIFT) sebagai salah satu metode untuk membantu menyelesaikan penelitian tersebut.

2.6. Metode *Scale-Invariantt Feature Transform* (SIFT)

Scale-invariant feature transform (SIFT) yang diteliti oleh David G. Lowe (1999) merupakan sebuah algoritma untuk mengekstraksi fitur khas dari gambar yang dapat digunakan untuk melakukan pencocokan dari berbagai pandangan objek. Fitur yang dihasilkan yang *invariant* untuk skala gambar, rotasi, dan dapat mengubah sudut pandang 3D (tiga dimensi), sebagai tambahan fitur tersebut tahan akan *noise*, pergeseran, serta perubahan iluminasi (Mandava, 2009).

Menurut David G. Lowe (2004), dalam SIFT terdapat tahap-tahap utama dari perhitungan yang akan menghasilkan serangkaian fitur gambar yaitu:

1. *Scale-Space Extrema Detection*

   Tahap ini merupakan pencarian perhitungan atas semua skala dan lokasi gambar. Hal ini dapat diimplementasikan dengan menggunakan fungsi *difference-of-Gaussian* yang berfungsi untuk mengidentifikasi titik-titik yang berpotensi *invariant* terhadap skala dan rotasi.

   Menurut Mandava (2004), metode ini biasanya menghasilkan banyak jumlah *key point* tersebut.

2. *Keypoint Localization*

   Pada setiap kindidat lokasi, dibuat model/objek secara detail untuk menentukan lokasi dan skala. *Keypoint* dipilih secara ukuran stabilitasnya masing-masing.

3. *Orientation Assignment*

   Dalam mencapai *invariant* rotasi dan skala, akan diberikan ke setiap lokasi *keypoint* berdasarkan arah gradien gambar lokal.

4. *Keypoint Descriptor*

   Gambar gradien lokal diukur pada skala yang dipilih di wilayah sekitar *keypoint* masing-masing. Ini merupakan transformasi menjadi sebuah representasi yang mungkin untuk tingkat signifikan dari distorsi bentuk dan perubahaan iluminasi. Setelah dilakukan pengujian, David G. Lowe

(2003) menemukan bahwa hasil terbaik dapat dicapai dengan menggunakan *descriptor* 4x4.

2.6.1 *Scale-Space Extrema Detection*

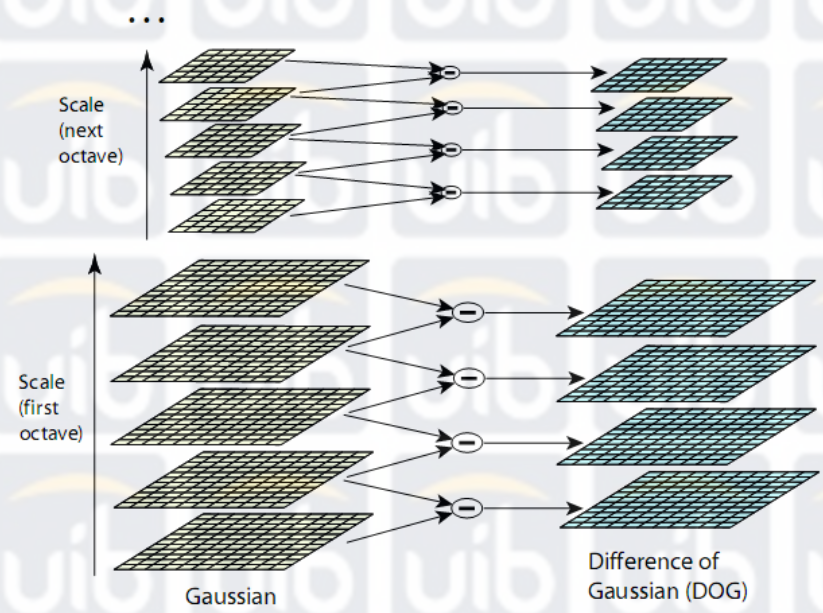
Menurut David G. Lowe (2003) tahap ini merupakan penyaringan untuk mengidentifikasi lokasi dan skala dari pandangan yang berbeda dengan objek yang sama. Hal ini dapat menggunakan fungsi *scale space* didasarkan dengan fungsi *Gaussian*. Sehingga *scale space* didefenisikan sebagai fungsi, L(x, y, σ), yang dihasilkan dari *convolution* variable skala *Gaussian*, G (x, y, σ) dengan gambar input, I (x, y). Dapat ditulis dengan:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad \text{.....(2.3)}$$

Dimana * merupakan operasi *convolution* dalam χ dan y.

Beberapa teknik dapat digunakan untuk mendeteksi lokasi *keypiont* yang stabil dalam *scale space* salah satunya adalah menggunakan *Difference of Gaussian*. Dimana lokasi *scale space extrema*, D (χ, y, σ) dengan menghitung perbedaan antara dua gambar yang dapat dihitung dari selisih dua skala terdekat oleh faktor konstan:

$$D(\chi, y, \sigma) = (G(\chi, y, k\sigma) - G(\chi, y, \sigma)) * I(\chi, y)$$
$$= L(\chi, y, k\sigma) - L(\chi, y, \sigma) \quad \text{.....(2.4)}$$



Gambar 2.2 *Scale-Space Extrema Detection*
Sumber: David G.Lowe (2003)

Pada Gambar 2.2 untuk setiap *octave of scale space*, gambar awal berulang-ulang menggulung dengan menggunakan *Gaussian* untuk menghasilkan serangkaian gambar *scale space* yang ditunjukkan di sebelah kiri. Gambar-gambar sebelah kiri yang menggunakan Gaussian dikurangi untuk menghasilkan *difference-of-Gaussian* yang ada disebelah kanan.

2.6.2 *Keypoint Localization*

Pada tahap ini apabila kandidat puncak telah ditemukan dengan membandingkan piksel dengan tetangganya, maka akan dilakukan pencocokan data terdekat untuk lokasi, deteksi tepi, dan besarnya puncak.

Interpolasi ini dengan menggunakan pemangkatan *Taylor* dari *difference of Gaussian* untuk fungsi *scale space*, D (χ, y, σ) dengan *keypoint* asalnya. Fungsi *Taylor* dapat definisikan sebagau berikut:

$$D(x) = D + (\partial D^T/\partial x)\, x + (1/2)\, x^T (\partial^2 D/\partial x^2)\, x \quad \text{.....(2.5)}$$

Dimana D merupakan turunan dari titik sampel dan x = (x, y, σ) adalah nilai *offset* dari titik. Lokasi dari extremum z dengan mengambil nilai derivatif fungsi, dengan defenisi:

$$Z = -(\partial^2 D^{-1}/\partial x^2)(\partial D/\partial x) \quad \text{.....(2.6)}$$

Jika nilai dari z di bawah dari nilai ambang titik tersebut maka dikecualikan. Hal ini untuk menghilangkan *extrema* dengan nilai kontras yang rendah.

2.6.3. *Orientation Assigment*

Dalam tahap ini, setiap *keypoint* diberikan satu atau lebih berdasarkan orientasi arah gradien dari gambar tersebut. Setelah eksperimen dengan jumlah pendekatan untuk menetapkan orientasi lokal yaitu skala *keypoint* yang menggunakan Gaussian untuk memperhaluskan gambar, L, dengan skala terdekat sehingga semua perhitungan harus dilakukan dengan cara skala invariant. Untuk setiap sampel gambar $L_{x,y}$, besarnya gradien, m, dan orientasi, θ, adalah prapenjumlah yang menggunakan perbedaan piksel (Lowe, 2003). Dapat dilihat dari Persamaan 2.7 :

$$m = \sqrt{(L_{x+1,\,y} - L_{x-1,\,y})^2 + (L_{x,\,y+1} - L_{x,\,y-1})^2}$$

$$\theta = \tan^{-1} \frac{(L_{x,y+1} - L_{x,y-1})}{(L_{x+1,y} - L_{x-1,y})} \quad \text{..... (2.7)}$$

Dari Persamaan 2.7 dapat dihitung besaran dan arah gradien dilakukan untuk setiap piksel dalam suatu wilayah *neighboring* di sekitar *keypoint* gambar L *Gaussian-blurred*. Sebuah histogram yang berorientasi membentuk 36 *bins*. Setelah diisi, orientasi yang sesuai dengan nilai puncak tertinggi dan puncak lokal yang berada dalam 80 % dari puncak tertinggi untuk *keypoint* tersebut.

2.6.4 *Keypoint Descriptor*

Dari hasil data gradien digunakan untuk membuat *keypoint descriptor*. *Keypoint descriptor* biasanya menggunakan 16 set histogram, selaras dengan 4 x 4 *grid* yang masing-masing mempunyai 8 *bins* orientasi. Histogram mempunyai masing-masing 8 bins dan setiap descriptor mempunyai *array* dari 4 histogram di sekitar keypoint. Ini menunjukkan fitur vektor pada SIFT dengan 4 x 4 x 8 = 128 elemen.

Gambar 2.3 *Keypoint Descriptor*
Sumber: David G.Lowe (2003)

Dari Gambar 2.3 dapat dilihat bahwa *keypoint descriptor* dibuat dengan menghitung besarnya gradien dan orientasi pada setiap titik sampel seperti yang ditunjukkan pada gambar sebelah kiri. Setelah sampel tersebut terakumulasi menjadi suatu histogram orientasi yang isinya menjadi besar, seperti yang ditunjukkan pada gambar sebelah kanan dengan anak panah masing-masing sesuai dengan jumlah besar gradien yang dekat dengan kawasan daerah tersebut.