

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pengenalan pola adalah proses mengelompokkan data numerik dan simbolik termasuk citra secara otomatis oleh komputer, tujuan dari pengelompokan adalah untuk mengenali suatu objek di dalam citra (Widodo, et al., 2006). Komputer akan menerima masukan berupa citra objek yang akan diidentifikasi kemudian memproses citra tersebut dan memberikan keluaran berupa informasi atau deskripsi objek di dalam citra. Dalam teknik pengenalan pola secara khusus digunakan untuk mengidentifikasi pola kain (Chang-chiun, et al., 2000). Terdapat banyak metode yang dapat digunakan untuk menangani masalah pengenalan pola, salah satunya adalah jaringan saraf tiruan [(Neruda & Vidnerova, 2009), (Prasad, et al., 2013)]. Jaringan syaraf tiruan adalah salah satu metode komputasi yang berusaha meniru cara kerja sistem syaraf otak manusia. Keunggulan metode ini dibanding metode lain adalah kemampuannya belajar dan memecahkan hubungan yang kompleks dan rumit, yang sulit untuk dideskripsikan antara data masukan dan data keluaran. Hal ini dimungkinkan karena pengetahuan yang ada pada jaringan syaraf tiruan tidak diprogram, namun dilatih berdasarkan informasi atau masukan yang diterimanya (Kusumaputro, 2000)

Algoritma pembelajaran yang digunakan dalam jaringan saraf tiruan adalah *Learning Vector Quantization* (LVQ) yang merupakan suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vector-vektor input (Naoum

& Al-Sultani, 2012). Sistem LVQ sangat fleksibel, mudah diterapkan, dan dapat diterapkan dalam masalah multi-kelas secara langsung (Purwanti & Widiyanti, 2012), tingkat kecocokan hasil pengenalan tergantung pada kombinasi nilai parameter yang digunakan dalam proses pembelajarannya dalam perbandingan metode jaringan syaraf tiruan *backpropagation* dan *learning vector quantization* pada pengenalan wajah dapat direkomendasikan dari segi akurasi dan waktu metode *learning vector quantization* lebih baik dari pada metode *backpropagation* dalam proses pengenalannya (Wuryandari & Afrianto, 2012).

Tetapi pengenalan pola citra dengan jaringan saraf tiruan akan menyebabkan proses pelatihan serta pengenalan pola membutuhkan waktu komputasi yang cukup lama (Zamani, et al., 2012), maka dibutuhkan sebuah proses ekstraksi ciri pada gambar dengan menggunakan metode transformasi gelombang singkat (*wavelet*). Transformasi *wavelet* memiliki dua jenis dalam pengembangannya yaitu *Continuous Wavelet Transform (CWT)* dan *Discrete Wavelet Transform (DWT)*. Semua fungsi yang digunakan dalam transformasi *CWT* dan *DWT* diturunkan dari *mother wavelet* melalui translasi dan kompresi. Standar diskrit transformasi wavelet (*DWT*) adalah suatu metode yang sangat ampuh dan berhasil digunakan untuk memecahkan berbagai masalah dalam sinyal dan pengolahan citra [(Narote, et al., 2009), (Mutter , et al., 2009)]. Proses ekstraksi ciri dilakukan dengan metode *Discrete Wavelet Transform (DWT)*, dimana metode ini telah banyak digunakan dalam berbagai penelitian dan memberikan hasil yang baik [(Murti, 2009), (Tawade & Warpe, 2011)]. Otomatis Segmentasi Citra menggunakan wavelet untuk membuat segmentasi cepat dan sederhana (Kumar, et

al., 2009), wavelet mempunyai kehandalan yang lebih jika dibandingkan fourier transform dalam menganalisis *image spatial* dan *frequency characteristic* (Gonzales , et al., 2009). Jenis DWT yang digunakan pada penelitian ini adalah Haar *Wavelets*.

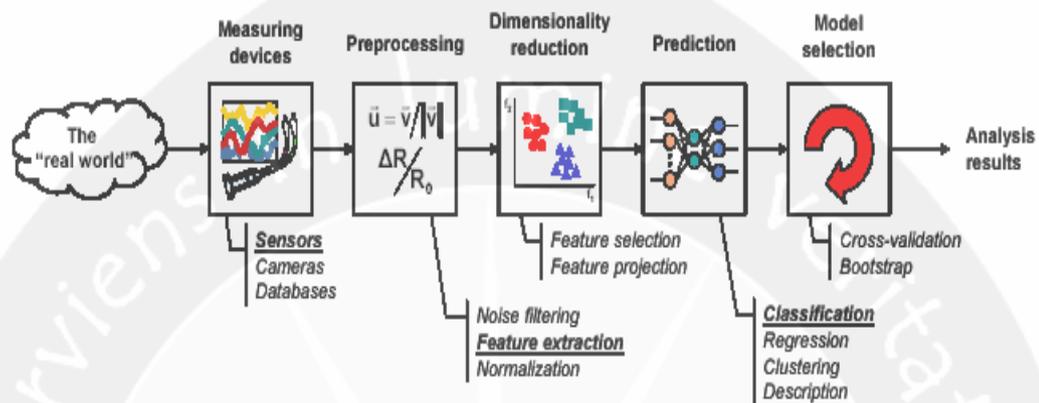
Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi (*edge*) dari objek di dalam citra. Tujuan operasi pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Terdapat banyak jenis algoritma deteksi tepi, seperti Canny, Sobel, Prewitt, Robert, dan Laplacian of Gaussian (LoG). Setiap algoritma deteksi tepi memiliki kelebihan dan kekurangan masing-masing. Diantara semua algoritma deteksi tepi yang ada, Canny merupakan algoritma deteksi tepi memberikan hasil yang paling optimal untuk digunakan pada hampir segala macam kondisi (Patil & Kulkarni, 2011).

2.2 Landasan Teori

2.2.1 Pengenalan Pola

Pengenalan pola memetakan suatu fitur, yang merupakan ciri utama suatu objek (bilangan) ke suatu kelas yang sesuai. Proses pemetaan ini menyangkut inferensi, baik secara eksplisit secara statistik, misalnya dalam aturan Bayesian maupun tak eksplisit dengan suatu jaringan keputusan misalnya, jaringan syaraf tiruan atau logika samar. Secara mendasar, suatu sistem pengenalan pola terdiri dari komponen-komponen berikut : sensor, mekanisme pre-processing, mekanisme penyari fitur (manual/ otomatis), algoritma pemilah dan sekumpulan contoh

pelatihan yang telah dipilah. Diagram blok dari sistem pengenalan pola dapat digambarkan sebagai berikut:



Gambar 2.1 Komponen Sistem Pengenalan Pola (Munir, 2004)

- 1) Sensor: berfungsi untuk menangkap objek dari dunia nyata menjadi sinyal-sinyal listrik (dan selanjutnya dalam bilangan-bilangan setelah proses digitasi).
- 2) Preprocessing: berfungsi untuk menonjolkan sinyal informasi dan menekan derau dalam sinyal.
- 3) Penyari fitur: mengambil besaran komponen tertentu dari sinyal yang mewakili sifat utama sinyal, sekaligus mengurangi dimensi sinyal menjadi sekumpulan bilangan yang lebih sedikit tetapi representatif.
- 4) Algoritma pemilah: melakukan assignment fitur ke kelas yang sesuai.
- 5) Sekumpulan contoh pelatihan: dipakai untuk mendapatkan representasi kelas.

Fitur adalah objek dari sebuah pola yang kuantitasnya dapat diukur, pengklasifikasiannya berdasar dari masing masing nilai dari fitur-fitur tersebut. Vektor fitur adalah sejumlah atau sekumpulan dari fitur, misalkan sebuah fitur adalah x maka kumpulan fitur dapat direpresentasikan sebagai x_1, \dots, x_i .

2.2.2 Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) adalah suatu metode komputasi yang meniru sistem jaringan saraf biologis. Metode ini menggunakan elemen perhitungan non linear dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran.

Layaknya neuron biologi, JST juga merupakan sistem yang bersifat “*fault tolerant*” dalam 2 hal, pertama dapat mengenali sinyal input yang agak berbeda dari yang pernah diterima sebelumnya. Sebagai contoh, manusia sering tidak mendapat mengenali seseorang yang wajahnya pernah dilihat dari foto atau dapat mengenali seseorang yang wajahnya agak berbeda karena sudah lama tidak dijumpainya. Kedua, tetap mampu bekerja meskipun beberapa neuronnya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak, neuron lain dapat dilatih untuk menggantikan fungsi neuron yang rusak tersebut.

Jaringan saraf tiruan, seperti manusia, belajar dari suatu contoh karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari suatu data-data sebelumnya dan mengenal pola data yang selalu berubah. Selain itu, JST merupakan sistem yang terprogram, artinya semua keluaran atau kesimpulan yang

tertarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran/ pelatihan. Hal yang ingin dicapai dengan melatih JST untuk dapat mencari keseimbangan antara kemampuan memorisasi dan generalisasi. Yang dimaksud kemampuan memorisasi adalah JST untuk dapat mengambil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan generalisasi adalah kemampuan JST untuk menghasilkan respon yang bisa diterima terhadap pola-pola input yang serupa (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam JST itu diinputkan informasi baru yang belum pernah dipelajari, maka JST itu masih akan tetap memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati (Puspitaningrum, 2006).

Jaringan saraf tiruan menyerupai otak manusia dalam dua hal, yaitu :

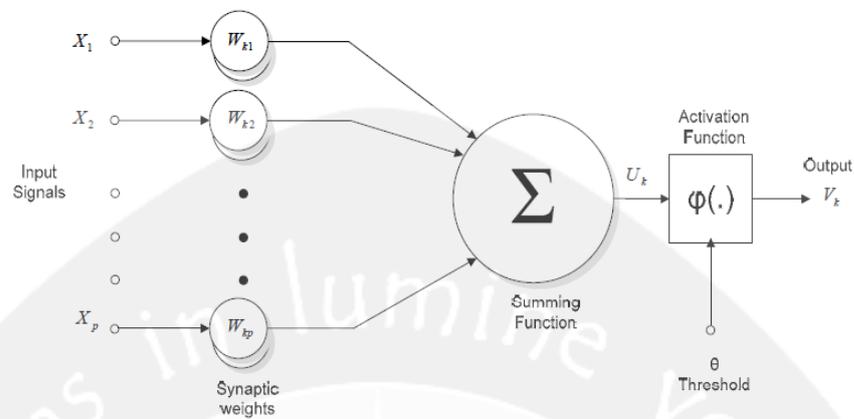
1. Pengetahuan diperoleh jaringan melalui proses belajar.
2. Kekuatan hubungan antar sel syaraf (neuron) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan.

JST ditentukan oleh 3 hal :

1. Pola hubungan antar neuron (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung.
3. Fungsi aktivasi, yaitu fungsi yang digunakan untuk menentukan keluaran suatu neuron.

2.2.2.1 Model neuron

Satu sel saraf terdiri dari tiga bagian, yaitu : fungsi penjumlah (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).



Gambar 2.2 Model Neuron (Puspitaningrum, 2006)

Jika dilihat pada gambar 2. 3, neuron buatan mirip dengan sel neuron biologis. Informasi (*input*) akan dikirimkan ke neuron dengan bobot tertentu. Input ini akan diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, jika tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot output ke semua neuron yang berhubungan dengannya.

2.2.3 Learning Vector Quantization

Pada penelitian kali ini metode yang digunakan adalah metode *Learning Vector Quantization*, dimana metode ini merupakan metode pembelajaran dengan pengarahan (*supervised*). Dengan metode ini maka pengklasifikasian pola

dilakukan dengan setiap unit keluaran mewakili satu kelas tertentu atau satu kategori tertentu.

Metode LVQ menggunakan vector acuan (*vector reference*) dari satu unit keluaran yang menjadi acuan bagi kelas/ kategori yang mewakili oleh keluaran tersebut. Pendekatan yang dilakukan adalah mengelompokkan vektor input berdasarkan kedekatan jarak vektor input terhadap bobot (metode kuadrat jarak euclidean minimum). *Learning Vector Quantization* merupakan salah satu metode jaringan saraf tiruan yang berbasis kompetisi dengan mekanisme *squared eucliden distance* dalam memilih vektor perwakilan pemenang untuk menentukan kategori vektor masukan. Proses pembelajaran LVQ merupakan pembelajaran *supervised* atau dengan kata lain menggunakan pengarahannya, dengan tujuan untuk mendapatkan vektor-vektor perwakilan yang akan melakukan kuantisasi terhadap vektor masukan (Kusumadewi, 2003).

Pada metode ini, pada saat menentukan vektor-vektor perwakilan untuk digunakan pada saat pelatihan, maka dengan pengarahannya vektor perwakilan tersebut akan mengenali target yang telah diberikan bersamaan dengan input. Dalam proses training, unit-unit output diarahkan kepada suatu *decision surface* teori dengan mengupdate bobot pada training. LVQ diarahkan untuk menentukan unit keluaran yang paling sesuai dengan target dari vektor masukannya melalui cara pergeseran posisi vektor perwakilan. Apabila vektor data training dikelompokkan sama dengan vektor perwakilan pemenang, maka vektor perwakilan digeser mendekati vektor training dengan notasi :

$$W(t+1) = w(t) + \alpha(t)(x(t)-w(t)) \quad (2.1)$$

Bila vektor data training dikelompokkan tidak sama dengan vektor perwakilan pemenang, maka vektor perwakilan digeser menjauhi vektor training yang dinyatakan dengan notasi :

$$W(t+1) = w(t) + \alpha(t)(x(t)-w(t)) \quad (2.2)$$

Dimana :

α = laju pembelajaran

w = posisi vektor perwakilan saat t

x = posisi vektor masukan saat t

Faktor-faktor pembelajaran pada LVQ :

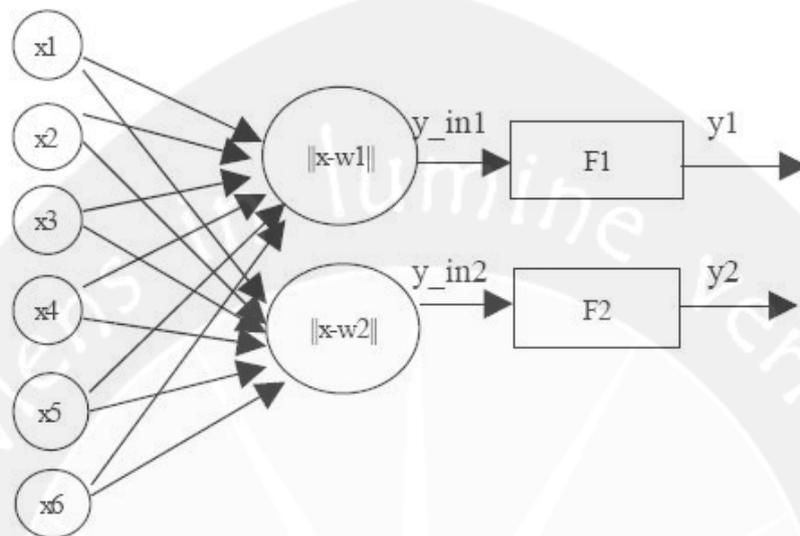
1) *Inisialisasi vektor perwakilan*

Inisialisasi pada LVQ dapat dilakukan dengan beberapa cara, antara lain memilih salah satu vektor training pada setiap kelas, memilih vektor training secara acak, cara random, dan inisialisasi awal '0'. Inisialisasi yang terbaik adalah dengan memilih masing-masing vektor training pada masing-masing kelas. Inisialisasi secara random akan mempersulit LVQ dalam melakukan pergeseran vektor perwakilan pada posisi yang benar, sehingga proses pembelajarannya menjadi lebih lama.

2) *Laju pembelajaran (α)*

Pada LVQ, laju pembelajaran akan berpengaruh pada saat pergeseran vektor perwakilan sesuai dengan persamaan (2.1) dan (2.2). oleh karena itu sebaiknya nilai dari laju pembelajaran tersebut jangan terlalu besar dan terlalu kecil.

2.2.3.1 Arsitektur Jaringan dan Algoritma LVQ



Gambar 2.3 Arsitektur Jaringan LVQ (Putri, 2012)

Gambar diatas merupakan contoh struktur jaringan LVQ yang memiliki empat input layer dengan dua unit neuron pada output layers. W_1 dan W_2 merupakan bobot yang menghubungkan input layer ke output layers. Setiap fungsi aktivitas F melakukan pemetaan setiap y diklasifikasikan y_1 atau y_2 . Pada F_1 , jika $|x - w_1| < |x - w_2|$, maka y_{in1} dipetakan ke $y_1=1$ dan dipetakan ke $y_1=0$ jika sebaliknya. Kondisi ini juga berlaku untuk F_2 dengan kondisi yang sesuai.

Secara garis besar, LVQ akan mencari unit keluaran yang paling mirip dengan vektor masukan. Jika vektor pelatihan adalah bagian dari kelas yang sama, maka vektor bobot digeser mendekati vektor masukan tersebut. Sebaliknya jika vektor pelatihan bukan bagian dari kelas yang sama, maka vektor bobot digeser menjauhi vektor masukan tersebut.

Berikut algoritma metode pembelajaran LVQ

Langkah 0 : Inisialisasi

Inisialisasi vektor-vektor acuan

Inisialisasi laju pembelajaran $\alpha=0$

Langkah 1 : Jika kondisi stop salah

Langkah 2 : Untuk setiap training vektor, lakukan langkah 2-6

Langkah 3 : Dapatkan nilai j sehingga $\|x-w_j\|$ bernilai minimum

$$D_i = \sqrt{\sum_{j=1}^m (W_{ji} - X_i)^2} \quad (2.3)$$

Langkah 4 : Update nilai bobot w_j

Jika $T = C_j$

$W_j(\text{baru}) = w_j(\text{lama}) + \alpha(x - w_j(\text{lama}))$

Jika $T \neq C_j$

$W_j(\text{baru}) = w_j(\text{lama}) + \alpha(x - w_j(\text{lama}))$

Langkah 5 : Update nilai learning rate

Langkah 6 : Uji kondisi stop

Uji kondisi stop ini dapat dilakukan berdasarkan jumlah iterasi tertentu (mulai dari langkah 1) atau setelah laju pembelajaran yang telah mencapai harga yang sangat kecil.

Dimana :

X = training vektor ($x_1, \dots, x_i, \dots, x_n$)

T = kategori training vektor yang benar untuk pelatihan

W_j = vektor bobot untuk unit keluaran ke- j ($w_1, \dots, w_j, \dots, w_n$)

C_j = kategori atau kelas yang mewakili oleh nilai unit keluaran ke- j
(hasil training)

$\|x-w_j\|$ = Euclidian distances antara 18 vektor masukan dan vektor bobot dari unit keluaran ke- j .

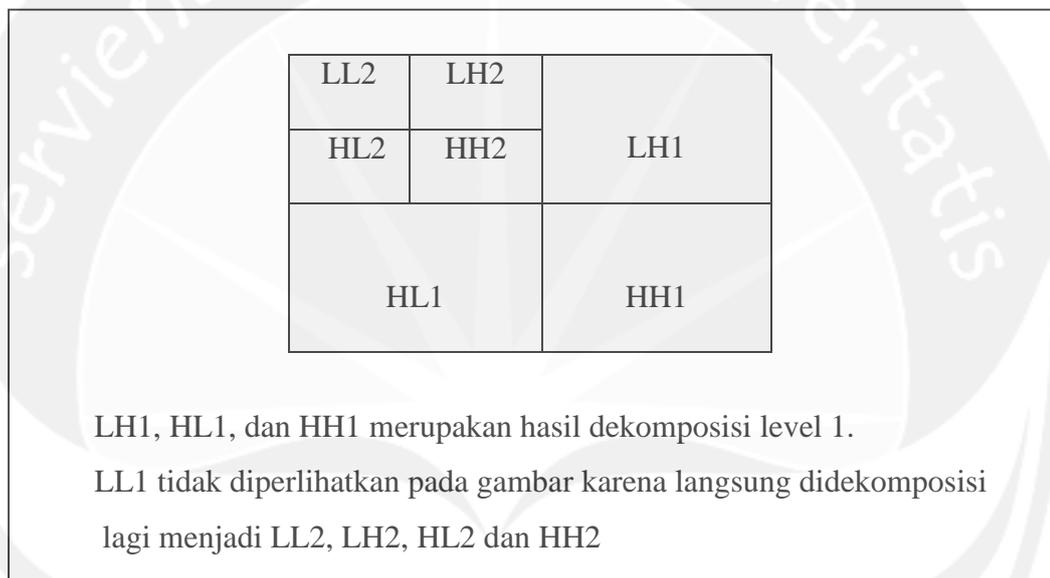
2.2.4 Wavelet

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 tuang (*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 subcitra 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 :



Gambar 2.4 Dekomposisi Citra (Santoso, 2011)

Pada citra 2 dimensi, terdapat dua cara untuk mentransformasi atau mendekomposisi nilai-nilai pikselnya, 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 diulangi dengan level yang diinginkan.

2.2.5 Deteksi Tepi (*Edge Detection*)

Deteksi tepi adalah metode yang dapat mendeteksi garis tepi, yaitu garis yang memisahkan antara objek dengan latar belakang (*background*). Deteksi tepi merupakan pengolahan citra tingkat dasar yang diperlukan untuk melakukan pengolahan citra pada tingkat yang lebih tinggi. Deteksi tepi banyak digunakan dalam analisa pengolahan citra untuk berbagai macam tujuan.

2.2.5.1 Deteksi Tepi Sobel

Sobel merupakan algoritma deteksi tepi yang menggunakan Sobel Operator, yaitu sepasang *kernel* berupa matriks berukuran 3 x 3 untuk mendeteksi tepi vertikal dan horizontal.

+1	+2	+1
0	0	0
-1	-2	-1

G_x

-1	0	+1
-2	0	+2
-1	0	+1

G_y

Gambar 2.5 Operator Sobel

Kombinasi kedua *kernel* tersebut digunakan untuk menghitung jarak absolut sebuah *gradient* namun bisa juga diaplikasikan secara terpisah untuk menghitung masing-masing proses vertikal dan horizontal. Untuk menghitung jarak *gradient*, digunakan persamaan berikut :

$$|G| = \sqrt{Gx^2 + Gy^2} \quad [2.1]$$

Kemudian untuk menghitung arah dari garis tepi yang dihasilkan, digunakan persamaan berikut :

$$\theta = \arctan(Gx/Gy) \quad [2.2]$$

2.2.5.2 Deteksi Tepi Prewitt

Prewitt merupakan algoritma deteksi tepi yang hampir serupa dengan Sobel, tetapi algoritma ini menggunakan Prewitt operator yang nilainya agak berbeda dengan Sobel operator.

+1	+1	+1
0	0	0
-1	-1	-1

Gx

-1	0	+1
-1	0	+1
-1	0	+1

Gy

Gambar 2.6 Operator Prewitt

2.2.5.3 Deteksi Tepi Canny

Canny adalah algoritma deteksi tepi yang banyak digunakan dalam berbagai penelitian karena dinilai sebagai algoritma deteksi tepi yang paling optimal. Langkah awal pada algoritma Canny adalah mengimplementasikan tapis *Gaussian* pada citra untuk menghilangkan derau. Kemudian dilanjutkan dengan melakukan deteksi tepi pada citra dengan salah satu algoritma deteksi tepi yang ada, misalnya Sobel atau Prewitt. Langkah berikutnya adalah membagi garis-garis yang ada

menjadi 4 warna terpisah dengan sudut masing-masing, lalu memperkecil masing-masing garis tepi agar menjadi tipis (*non maximum surpression*). Langkah terakhir adalah melakukan proses binerisasi berdasarkan nilai *low & high threshold* yang diberikan.

2.2.6 Android

Situs resmi Android (<http://developer.android.com>) mendefinisikan bahwa “Android merupakan sebuah perangkat lunak untuk piranti bergerak (*mobile*) yang meliputi sistem operasi, *middleware*, dan aplikasi.” Sebelum akhirnya dibeli oleh Google Inc., Android merupakan produk dari Android Inc. Dalam perkembangannya, terbentuklah suatu organisasi bernama Open Handset Alliance yang merupakan gabungan dari banyak perusahaan dengan tujuan mengembangkan Android. Untuk merancang aplikasi pada *platform* Android, Android telah menyediakan sebuah *tools* yang bernama *Android Software Development Kit (SDK)*. Android *SDK* menyediakan sekumpulan peralatan dan *Application Programming Interfaces (API)* yang digunakan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java (developer.android.com). Dengan menggunakan Android *SDK*, siapapun dapat mengembangkan aplikasi Android dengan bebas.

2.2.7 Kain Bentean

Kain tenun bentean keberadaannya di Minahasa, Sulawesi Utara sekitar hampir 200 tahun lalu, kain ini ditenun dengan menggunakan benang kapas dan diberi bahan pewarna alam. Umumnya kain tenun bentean pada saat itu, ditenun

dalam bentuk *Pasolongan* (bundar, seperti kain sarung namun tanpa sambungan/jahitan). Pembuatan kain tenun bentenan ini sangat sulit (teknik menenun yang tinggi), sehinggalah memakan waktu berbulan-bulan. Itu sebabnya, kain ini mempunyai nilai tinggi. Tapi bukan saja karena teknik pembuatannya yang mengakibatkan nilainya sangat tinggi, namun juga pada saat menenun, didendangkan lagu-lagu ritual dan dengan doa yang dipanjatkan sebelum penenunan dimulai. Oleh karenanya, pada waktu itu kain ini dipakai sebagai emas kawin.

Kain Tenun Bentenan adalah salah satu kain yang mutunya terbaik di dunia (Wenas, 2007). Data tulisan pertama mengenai kain ini terdapat di Koran Tjahaya Siang, 12 Agustus 1880, ketika kain tenun ini mulai menghilang dari Minahasa. Beberapa penulis luar seperti DR A.B. Meyer (1989); N Graafland (1898); D.E. Gasper (1919) direview oleh (Raturandang, 2007) menamakan kain ini kain bentenan, walaupun kain itu di tenun di Tombulu, Tondano, Ratahan, Tombatu dan wilayah lainnya di Minahasa. Di abad 15 sampai dengan 17, kota pelabuhan utama di Sulawesi Utara ialah di wilayah bentenan, Minahasa Tenggara. Melalui pelabuhan bentenanlah kain tenun ini di ekspor ke luar Minahasa. Kain tenun Minahasa pertama ditemukan di Bentenan, dan kain ini terakhir ditenun di daerah Ratahan pada tahun 1912, itulah sebabnya kain ini disebut kain tenun bentenan. (Raturandang, 2007)

Karena cara pembuatannya yang cukup sulit sehingga, kain tenun bentenan, saat itu hanya digunakan oleh orang-orang tertentu pada acara-acara tertentu pula, seperti oleh para pemimpin adat (*Tonaas*) dan pemimpin agama/suku (*Walian*)

dalam upacara adat dan upacara agama. Kain tenun bentenan berperan utama dalam lingkaran kehidupan manusia, seperti lahir, menikah dan meninggal. Bayi lahir dibungkus dengan kain tenun bentenan, ada upacara perkawinan juga menggunakan kain tenun bentenan, bahkan upacara pemakaman pun kain tenun bentenan digunakan untuk membungkus jenazah. Para pemimpin masyarakat menggunakan kain tenun Bentenan di kursi tamu dan dinding ruang tamu sebagai simbol status sosial.

Menurut motif hiasnya, kain tenun bentenan memiliki 7 motif, yaitu:

1. ***Tonilama*** (tenun dengan benang putih, tidak berwarna dan merupakan kain putih)
2. ***Sinoi*** (tenun dengan benang warna-warni/pelangi dan berbentuk garis-garis)
3. ***Pinatikan*** (tenun dengan garis-garis motif jala dan bentuk segi enam. Jenis ini pertama ditenun di Minahasa)
4. ***Tinompak Kuda*** (tenun dengan aneka motif berulang)
5. ***Tinonton Mata*** (tenun dengan gambar manusia)
6. ***Kaiwu Patola*** (tenun dengan motif patola India)
7. ***Kokera*** (tenun dengan motif bubga warna-warni bersulam maik-manik)

Saat ini masih ada 28 lembar kain tenun Bentenan yang tersisa di dunia. Untuk jenis *Pinatikan* terdapat 20 lembar. Dua lembar berada di Museum Nasional Jakarta, empat lembar di Tropenmuseum, Amsterdam, tujuh lembar di Museum voor Land-an Volkenkunde, Frankfurt-am-Main, Jerman, empat lembar di Ethnographical museum Dresden, dan satu lembar di Indonesisch Etnografisch Museum, Delfi. Sedangkan untuk jenis *Kaiwu Patola* terdapat delapan lembar di

dunia, dua lembar berada di Museum Nasional di Jakarta, empat lembar di Tropenmuseum, Amsterdam dan dua lembar di Rotterdam Ethnology Museum (Raturandang, 2007)

