

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Penelitian mengenai pengenalan pola telah banyak dilakukan sebelumnya, seperti pengenalan tokoh wayang, pengenalan pola angka, pengenalan pola wajah, pengenalan pola batik, dan sebagainya. Beberapa penelitian tersebut diantaranya adalah penerapan jaringan saraf tiruan *Backpropagation* dengan momentum untuk pengenalan tokoh wayang (Widitamtomo, 2003). Pada penelitian ini, sebagai masukan adalah ciri-ciri wayang untuk pengenalan pola, sehingga dapat memprediksi nama tokoh wayang tersebut. Kekurangan pada penelitian pengenalan tokoh wayang ini, yaitu proses pembelajaran yang lama karena harus menunggu sampai iterasi yang ditentukan selesai, dan penyimpanan data menggunakan \*.txt sehingga keamanan kurang terjamin.

Penelitian menggunakan metode *Backpropagation* lainnya yaitu pengenalan wajah (Nugroho, 2003). Obyek wajah yang dipakai didapat dari hasil foto, dan untuk mendapatkan pola wajahnya digunakan metode deteksi tepi *Prewitt* operator. Penelitian ini masih memiliki kekurangan, yaitu waktu pembelajaran yang lama, dan faktor pencahayaan yang mempengaruhi pengenalan wajah.

Penelitian lainnya tentang aplikasi pengenalan pola batik menggunakan jaringan saraf tiruan (Inge, 2005), yang dibuat dengan menggunakan Matlab, dan metode *Backpropagation*. Hasil penelitian ini adalah aplikasi yang dibuat mampu mengenali pola batik yang dikenai

*noise* seperti penghilangan parsial, *noise* secara poisson, *noise* secara *salt & pepper*. Tetapi pada penelitian ini hanya dibatasi untuk tiga pola batik saja, yaitu batik Solo, batik Jogja, batik Pekalongan.

Penelitian pengenalan pola lainnya adalah pengenalan angka 0-9 dengan metode LVQ (*Learning Vector Quantization*) dan transformasi *Wavelet* (Suryani, 2006). Pada penelitian yang menggunakan *Visual Basic 6* ini, telah mampu mengenali pola masukan dengan *noise* sampai 88%, tetapi juga masih memiliki kekurangan, yaitu belum mampu mengenali pola yang memiliki perbedaan warna polanya.

Penelitian lainnya adalah tentang pengenalan citra sidik jari berbasis transformasi *Wavelet* dan jaringan saraf tiruan *Backpropagation* (Wijaya, 2004). Tujuan dari penelitian ini untuk menentukan *Wavelet* yang terbaik untuk pengenalan citra sidik jari. Hasil penelitian menunjukkan bahwa pengenalan citra sidik jari menggunakan transformasi *Wavelet* dan jaringan syaraf tiruan memberikan hasil yang baik, dengan tingkat kesuksesan pengenalan diatas 90% dan waktu pengenalan yang singkat.

## 2.2. Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) adalah sebuah sistem pengolahan informasi yang karakteristik kinerjanya menyerupai jaringan saraf biologis. JST telah dikembangkan sebagai generalisasi model matematika dari pengertian manusia atau saraf biologi, berdasarkan pada asumsi-asumsi bahwa :

- a. pemrosesan informasi terjadi pada banyak elemen-elemen sederhana yang disebut saraf (*neuron*)
- b. sinyal-sinyal disampaikan antar saraf atas/pada jalur-jalur hubungan (*connection link*)
- c. setiap jalur hubungan mempunyai sebuah bobot hubungan, yang mengadakan sinyal transmisi pengiriman.
- d. Setiap saraf menggunakan fungsi aktivasi, untuk jaringan inputnya untuk menentukan sinyal output.

Sebuah jaringan saraf dicirikan oleh : pola hubungan antar saraf (disebut arsitektur), metode penentuan bobot pada hubungan-hubungannya (disebut pembelajaran, pengetahuan, atau algoritma), dan fungsi aktivasi (Santoso, 2000).

JST memiliki sifat mendasar yakni kecerdasan yang analog dengan kecerdasan manusia. JST dapat menyelesaikan persoalan rumit yang sulit atau tidak mungkin jika diselesaikan dengan menggunakan komputer secara konvensional.

Jaringan Saraf Tiruan merupakan bidang interdisipliner ilmu pengetahuan, baik untuk pengembangan maupun aplikasinya. Beberapa contoh aplikasinya yaitu : pemrosesan isyarat, pengendali /

kontrol, penghasihan ucapan (*speech production*), pengenalan ucapan, dan pengenalan pola.

### **2.2.1. Pengenalan Pola**

Pengenalan pola (*pattern recognition*) pada mulanya dimiliki manusia yang berbasis pada kemampuan alat indera manusia, dimana manusia mampu mengingat suatu informasi pola secara menyeluruh hanya berdasarkan sebagian informasi pola yang tersimpan di dalam ingatannya. Misalnya sebuah nada pendek yang dibunyikan dapat membuat seseorang mengingat sebuah lagu secara keseluruhan. Tetapi karena keterbatasan daya ingat manusia, pengenalan pola telah diterapkan dalam komputer.

Pengenalan pola merupakan pengenalan suatu objek dengan proses pengenalannya memiliki tingkat akurasi tinggi. Memiliki tingkat akurasi tinggi mengandung pengertian bahwa suatu objek yang secara manual (oleh manusia) tidak dapat dikenali, tetapi bila menggunakan salah satu metode pengenalan pola yang diaplikasikan pada komputer masih dapat dikenali.

Konsep pengenalan pola merupakan suatu proses pencocokkan antara ciri-ciri citra yang dicari (*citra query*) dengan ciri-ciri citra yang ada dalam basisdata (*citra pustaka*). Faktor yang menyebabkan proses pencocokkan/ pengenalan citra menjadi lebih sulit/ rumit adalah *citra query* sudah mengalami perubahan atau distorsi, antara lain *citra query* dapat mengalami pergeseran warna, resolusi yang rendah, berderau atau registrasi yang kurang sempurna (Wijaya, 2004).

Metode yang digunakan untuk pengenalan pola ada bermacam-macam, salah satunya adalah *Backpropagation*

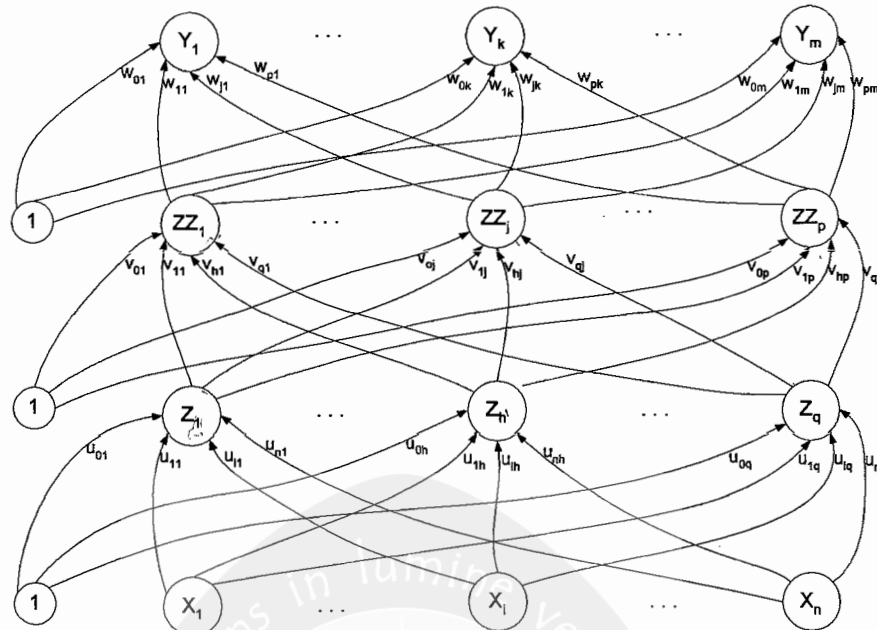
### **2.2.2. Backpropagation Momentum**

Salah satu algoritma pelatihan jaringan syaraf tiruan yang banyak dimanfaatkan dalam bidang pengenalan pola adalah algoritma *backpropagation*, umumnya digunakan pada jaringan syaraf tiruan yang berjenis multi-layer *feed-forward*, yang tersusun dari beberapa lapisan dan sinyal dialirkan secara searah dari input menuju output (Fausett, 1994).

*Backpropagation* dapat dipahami dalam beberapa tingkatan. Pada satu tingkatan merupakan kumpulan dari persamaan vektor, pada tingkat lain dapat dipandang sebagai suatu sistem berlapis dengan simpul-simpul yang saling berinteraksi. Algoritma *backpropagation*, yaitu metode dengan memperbaiki bobot pada saat sistem sudah mempunyai kesalahan rata-rata dari seluruh pasang data.

Salah satu metode paling awal yang diusulkan untuk mengatasi masalah lamanya waktu *training* adalah dengan menambahkan term momentum (dapat dilihat pada algoritma pelatihan pembaruan bobot), dimana parameter momentum akan menentukan besarnya pengaruh perubahan bobot pada iterasi sebelumnya. Cara ini dapat bekerja dengan baik pada kasus tertentu, namun tidak bersifat umum. Kadang-kadang malah diperoleh hasil yang lebih baik tanpa menggunakan momentum sama sekali (Riedmiller, 1994).

### 2.2.2.1. Arsitektur Two Hidden Layers



Gambar 2.1. Backpropagation dengan 2 hidden layers

Gambar di atas menampilkan jaringan saraf multilayer dengan 2 layer atau hidden units (unit Z dan ZZ). unit output (unit Y) dan unit tersembunyi dapat mempunyai bias. Bias pada unit output  $Y_k$  ditunjuk  $w_{0k}$ ; bias pada unit tersembunyi  $Z_j$  ditunjuk  $v_{0j}$ . Unit tersembunyi mempunyai bias yang berlaku sebagai bobot koneksi dari unit yang outputnya selalu 1.

### 2.2.2.2. Fungsi Aktivasi

Fungsi aktivasi pada *backpropagation* memiliki beberapa karakteristik penting yaitu kontinyu, dapat dideferensialkan, monoton tanpa penurunan. Fungsi aktivasi biasanya digunakan untuk mencari nilai asimot maksimum dan minimum. Salah satu contoh fungsi aktivasi adalah fungsi biner sigmoid yang mempunyai kisaran (0,1) dengan rumus :

$$f_1(x) = \frac{1}{1 + \exp(-x)} \quad (2.1)$$

dengan

$$f_1'(x) = f_1(x)[1 - f_1(x)] \quad (2.2)$$

Fungsi aktivasi lainnya adalah fungsi bipolar sigmoid dengan kisaran  $(-1, 1)$  dengan rumus :

$$f_2(x) = \frac{2}{1 + \exp(-x)} - 1 \quad (2.3)$$

dengan

$$f_2'(x) = \frac{1}{2}[1 + f_2(x)][1 - f_2(x)] \quad (2.4)$$

Sebagai catatan bahwa fungsi bipolar sigmoid berkaitan erat dengan fungsi :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.5)$$

### 2.2.2.3. Algoritma Pelatihan *Two Hidden Layers*

Algoritma pelatihan *backpropagation* pada dasarnya terdiri dari tiga tahapan (Fausett, 1994), yaitu:

1. Input nilai data pelatihan sehingga diperoleh nilai output
2. Propagasi balik dari nilai error yang diperoleh
3. Penyesuaian bobot koneksi untuk meminimalkan nilai error

Ketiga tahapan tersebut diulangi terus-menerus sampai mendapatkan nilai error yang diinginkan. Setelah *training* selesai dilakukan, hanya tahap pertama yang diperlukan untuk memanfaatkan jaringan syaraf tiruan tersebut.

L.0. Inisialisasi bobot (harga acak kecil)

L.1. Selama kondisi berhenti salah, melakukan langkah 2-9.

L.2. Untuk setiap pasang pelatihan, melakukan langkah 3-8.

**Feedforward :**

L.3. Setiap lapisan input ( $X_i, i = 1, \dots, n$ ) :

mengirim sinyal masukan ke unit tersembunyi.

L.4. Setiap lapisan tersembunyi ( $Z_h, h = 1, \dots, q$ ) :

Menghitung sinyal input.

$$z\_in_h = u_{oh} + \sum_{i=1}^n x_i u_{ih} \quad (2.6)$$

menggunakan fungsi aktivasi untuk menghitung sinyal keluaran.

$$z_h = f(z\_in_h) \quad (2.7)$$

dan mengirim sinyal ini ke semua unit tersembunyi yang kedua.

Setiap lapisan tersembunyi ( $zz_j, j = 1, \dots, p$ ) :

Menghitung sinyal input.

$$zz\_in_j = u_{oj} + \sum_{h=1}^n z_h u_{hj} \quad (2.8)$$

menggunakan fungsi aktivasi untuk menghitung sinyal keluaran.

$$zz_j = f(zz\_in_j) \quad (2.9)$$

dan mengirim sinyal output ke unit output.

L.5. Setiap unit output ( $Y_k, k = 1, \dots, m$ ) : menjumlahkan sinyal input bobotnya



$$y_{in_k} = w_{ok} + \sum_{j=1}^p zz_j w_{jk} \quad (2.10)$$

dan fungsi menggunakan fungsi aktivasi untuk menghitung sinyal keluaran.

$$y_k = f(y_{in_k}) \quad (2.11)$$

**Kesalahan Bacpropagation :**

L.6. Setiap unit output ( $Y_k$ ,  $k = 1, \dots, m$ ) :  
menghitung kesalahan

$$e_k = (t_k - y_k) \quad (2.12)$$

mengalikannya dengan turunan fungsi aktivasi

$$\delta_k = e_k f'(y_{in_k}) \quad (2.13)$$

menghitung koreksi bobot (digunakan untuk meng-update  $w_{jk}$ )

$$\Delta w_{jk} = \alpha \delta_k zz_j \quad (2.14)$$

menghitung koreksi bias

$$\Delta w_{ok} = \alpha \delta_k \quad (2.15)$$

dan mengirim nilai  $\delta_k$  ke unit tersembunyi ( $zz_j$ ,  $j = 1, \dots, p$ )

L.7. Setiap unit tersembunyi ( $zz_j$ ,  $j = 1, \dots, p$ ) :

menjumlahkan input bobot dari unit di lapisan atasnya

$$\partial_{in_j} = \sum_{k=1}^m \partial_k w_{kj} \quad (2.16)$$

mengalikan dengan turunan fungsi aktivasi

$$\partial_j = \partial_{in_j} f'(zz_{in_j}) \quad (2.17)$$

menghitung koreksi bobot

$$\Delta v_{ij} = \alpha \partial_j x_i \quad (2.18)$$

menghitung koreksi bias

$$\Delta v_{0j} = \alpha \delta_j \quad (2.19)$$

dan mengirim nilai  $\delta_j$  ke unit tersembunyi  
( $Z_h, h = 1, \dots, q$ )

Setiap unit tersembunyi ( $Z_h, h = 1, \dots, q$ )

menjumlahkan input bobot dari unit di lapisan atasnya

$$\partial_{in_h} = \sum_{j=1}^p \partial_j v_{hj} \quad (2.20)$$

mengalikan dengan turunan fungsi aktivasi

$$\partial_h = \partial_{in_h} f'(z_{in_h}) \quad (2.21)$$

menghitung koreksi bobot

$$\Delta u_{ih} = \alpha \delta_h x_i \quad (2.22)$$

menghitung koreksi bias

$$\Delta u_{0j} = \alpha \delta_j \quad (2.23)$$

#### Perbarui bobot dan bias :

L.8. Untuk setiap unit keluaran  $Y_k (j=0, \dots, p; k=1, \dots, m)$  :

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \partial_k z z_j + \mu [w_{jk}(t) - w_{jk}(t-1)]$$

atau (2.24)

$$\Delta w_{jk}(t+1) = \alpha \partial_k z z_j + \mu \Delta w_{jk}(t)$$

Untuk setiap unit tersembunyi  $z z_j (h=0, \dots, q; j=1, \dots, p)$  :

$$v_{hj}(t+1) = v_{hj}(t) + \alpha \partial_j z_h + \mu [v_{hj}(t) - v_{hj}(t-1)]$$

atau (2.25)

$$\Delta v_{hj}(t+1) = \alpha \partial_j z_h + \mu \Delta v_{hj}(t)$$

Untuk setiap unit tersembunyi  $Z_h (i=0, \dots, n; h=1, \dots, q)$  :

$$\begin{aligned}
 u_{ih}(t+1) &= u_{ih}(t) + \alpha \partial_h x_i + \mu [u_{ih}(t) - u_{ih}(t-1)] \\
 \text{atau} & \\
 \Delta u_{ih}(t+1) &= \alpha \partial_h x_i + \mu \Delta u_{ih}(t)
 \end{aligned}
 \tag{2.26}$$

L.9. Pengetesan kondisi berhenti.

#### 2.2.2.4. Prosedur Aplikasi

Setelah proses pelatihan, jaringan saraf *backpropagation* dipaplikasikan dengan fase *feedforward* dari algoritma pelatihan.

L.0. Inisialisasi bobot (dari algoritma pelatihan)

L.1. Untuk setiap vektor masukan, mengerjakan langkah 2-4

L.2. Untuk  $i=1, \dots, n$ : men-set aktivasi unit masukan  $x_i$

L.3. Untuk  $h=1, \dots, q$ :

$$\begin{aligned}
 z\_in_h &= v_{0h} + \sum_{i=1}^n x_i v_{ih} \\
 z_h &= f(z\_in_h)
 \end{aligned}
 \tag{2.27}$$

Untuk  $j=1, \dots, p$ :

$$\begin{aligned}
 z\_in_j &= v_{0j} + \sum_{h=1}^q z_h v_{hj} \\
 z_j &= f(z\_in_j)
 \end{aligned}
 \tag{2.28}$$

L.4. Untuk  $k=1, \dots, m$ :

$$\begin{aligned}
 y\_in_k &= w_{0k} + \sum_{j=1}^p z_j w_{jk} \\
 y_k &= f(y\_in_k)
 \end{aligned}
 \tag{2.29}$$

### 2.3. Citra

Citra merupakan suatu representasi, kemiripan, atau imitasi/tiruan dari suatu obyek atau benda (Kamus Webster). Ditinjau dari sudut pandang matematis, citra

merupakan fungsi penerus dari intensitas cahaya pada bidang 2-Dimensi. Pantulan obyek diterima oleh suatu sistem optis, dan kemudian akan menghasilkan suatu citra.

Contohnya seperti : foto yang mewakili entitas diri seseorang di depan kamera, foto sinar-X mewakili keadaan bagian tubuh seseorang.

Citra ada dua macam, yaitu citra tampak (foto keluarga, gambar teman, sesuatu yang tampak di layar monitor/TV, hologram (citra optis) dan citra tak tampak (citra gambar dalam file (citra digital), citra yang dinyatakan dalam fungsi matematis, citra fisik tak tampak: peta densitas material, citra distribusi panas bumi, kulit manusia). Citra tak tampak harus diubah menjadi citra tampak, dengan cara : menampilkan di monitor atau mencetaknya di kertas.

Citra yang dapat diolah dengan komputer disebut citra digital. Jika citra non-digital ingin diubah ke citra digital, maka harus dilakukan proses pencitraan (*imaging*), yaitu kegiatan untuk mengubah informasi citra fisik non-digital menjadi citra digital.

#### **2.4. Pengolahan Citra**

Pengolahan citra yaitu mentransformasikan suatu citra menjadi citra lain yang bertujuan memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau mesin. Pengolahan citra untuk citra digital dilakukan dengan menggunakan komputer, sehingga dihasilkan citra dengan kualitas citra yang lebih baik dan menghasilkan informasi yang lebih berarti. Pengolahan citra dapat dilakukan apabila :

- a. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung dalam citra.
- b. Elemen di dalam citra perlu dikelompokkan, dicocokkan atau diukur.
- c. Sebagian citra perlu digabung dengan bagian citra yang lain.

Pada citra digital yang dikelola/ diproses adalah elemen matriks citra digital. Elemen yang diolah dapat berupa : elemen piksel tunggal, elemen piksel yang bertetangga, sekelompok piksel tertentu, seluruh elemen matriks citra. Pada penelitian ini, pengolahan citra yang dilakukan adalah proses transformasi *Wavelet Haar*, proses *thresholding*, proses penambahan derau/ *noise*.

#### **2.4.1. Transformasi Wavelet**

*Wavelet* adalah suatu fungsi matematika yang membagi data menjadi berbagai komponen yang frekuensinya berbeda, kemudian mempelajari setiap komponen frekuensi dengan resolusi yang cocok untuk setiap ukuran. *Wavelet* merupakan suatu bentuk gelombang yang secara efektif memiliki batas durasi nilai rerata nol. *Wavelet* selain digunakan sebagai alat analisis yang baru di bidang matematika, juga merupakan alat yang praktis yang sangat handal untuk berbagai bidang aplikasi mulai untuk persamaan diferensial hingga untuk pengolahan citra (*image processing*). *Wavelet* dan alihragam *wavelet* merupakan sesuatu yang sangat penting dalam bidang grafika komputer. Beberapa aplikasi yang telah berhasil diwujudkan dengan memanfaatkan *wavelet* diantaranya

Kompresi data citra, watermarking, deteksi tepi / Sistem penglihatan robot, querying citra, dll.

Alasan mengapa alihragam *wavelet* menjadi begitu penting dalam berbagai bidang adalah karena sifat-sifat berikut:

- a. Waktu kompleksitasnya bersifat linear. Alihragam *wavelet* dapat dilakukan dengan sempurna dengan waktu yang bersifat linear
- b. Koefisien-koefisien *wavelet* bersifat jarang. Secara praktis, koefisien-koefisien *wavelet* kebanyakan bernilai kecil. Kondisi ini sangat memberikan keuntungan terutama dalam bidang kompresi data
- c. *Wavelet* dapat beradaptasi pada berbagai jenis fungsi, seperti fungsi tidak kontinyu, dan fungsi yang didefinisikan pada domain yang dibatasi

*Wavelet* merupakan sebuah fungsi yang dapat memuaskan kebutuhan matematika dan dapat digunakan untuk merepresentasikan data atau fungsi lain. Ide ini bukanlah sesuatu yang baru, sebab sebelumnya fungsi-fungsi lain telah dapat dinyatakan sebagai superposisi fungsi *sinus* dan *cosinus*, yang sangat terkenal dengan sebutan alihragam Fourier, akan tetapi, dalam analisis *wavelet*, skala yang digunakan untuk melihat data memainkan peranan yang sangat khusus. *Wavelet* mengolah data pada perbedaan skala dan resolusi.

Transformasi *wavelet* merupakan uraian dari suatu sinyal atau citra menggunakan fungsi *wavelet* dengan lokasi dan ukuran yang berbeda serta dihitung dengan menggunakan algoritma piramid.

Citra digital dinyatakan dalam bentuk diskret, maka sering digunakan bentuk alihragam wavelet dalam bentuk diskret (*Discrete Wavelet Transform / DWT*).

Alihragam balik wavelet sering disebut *Invers Discrete Wavelet Transform / IDWT*).



**Gambar 2.2. Transformasi Wavelet**

Ide dasar alihragam *Discrete Wavelet Transform* (DWT) 1-dimensi adalah seperti berikut :

- a. Isyarat dibagi menjadi dua bagian, frekuensi tinggi dan frekuensi rendah.
- b. Bagian frekuensi rendah dibagi lagi menjadi isyarat frekuensi tinggi dan rekuensi rendah.
- c. Proses ini diulang sampai isyarat tidak dapat didekomposisikan lagi atau sampai pada level yang telah ditentukan pengguna.

Pendekatan matematis transformasi wavelet :

Diberikan tapis lolos-bawah  $H(\omega)$

$$H(\omega) = \sum_{k \in \mathbb{R}} h_k e^{-jk\omega} \quad (2.30)$$

dengan  $h_k$  adalah koefisien tapis lolos-bawah dan

diberikan tapis lolos-atas  $G(\omega)$

$$G(\omega) = \sum_{k \in \mathbb{R}} g_k e^{-jk\omega} \quad (2.31)$$

dengan  $g_k$  adalah koefisien tapis lolos-atas

Isyarat  $F(n)$  dapat didekomposisi secara rekursif sebagai

$$f_{j-1}^{low}(k) = \sum_{n \in \mathbb{R}} h_{n-2k} f_j(n) \quad (2.32)$$

dan

$$f_{j-1}^{high}(k) = \sum_{n \in \mathbb{R}} g_{n-2k} f_j(n) \quad (2.33)$$

dengan  $j = J + 1, J, \dots, J_0$  dan  $f_{J+1}(k) = F(f)$ ,  $k \in \mathbb{Z}$ .  $J+1$  adalah indeks resolusi tertinggi dan  $J_0$  adalah indeks level resolusi terendah. Koefisien  $f_{J_0}^{low}(k), f_{J_0}^{high}(k), f_{J_0+1}^{high}(k), \dots, f_j^{high}(k)$  disebut dengan DWT atas isyarat  $F(n)$ , dengan  $f_{J_0}^{low}(k)$  adalah resolusi terendah (aproksimation) dan  $f_j^{high}(k)$  adalah detail isyarat pada tiap bidang frekuensi.

Isyarat  $F(n)$  dapat direkonstruksi berdasarkan koefisien DWT secara rekursif dengan

$$f_j^{low}(n) = \sum_{k \in \mathbb{R}} h_{n-2k} f_{j-1}^{low}(k) + \sum_{k \in \mathbb{R}} g_{n-2k} f_{j-1}^{high}(k) \quad (2.34)$$

Pada DWT dan IDWT diperlukan kondisi ortogonal antara tapis lolos-bawah dan tapis lolos-atas

$H(\omega)$  sehingga  $G(\omega)$

$$|H(\omega)|^2 + |G(\omega)|^2 = 1 \quad (2.35)$$

Sebagai contoh  $H(\omega)$  dan  $G(\omega)$  diberikan sebagai :

$$H(\omega) = \frac{1}{2} + \frac{1}{2} e^{-j\omega} \quad (2.36)$$

dan

$$G(\omega) = \frac{1}{2} - \frac{1}{2} e^{-j\omega} \quad (2.37)$$

yang sering disebut dengan tapis *wavelet Haar*.

#### 2.4.1.1. Wavelet Haar

*Wavelet Haar* diperkenalkan oleh Alfred Haar pada tahun 1909. Untuk *wavelet Haar* 1 dimensi, Fungsi basis untuk ruang  $V_j$  disebut dengan fungsi penyekalaan. Basis



sederhana pada  $V_j$  diberikan sebagai fungsi penyekalaan dan translasi sbb

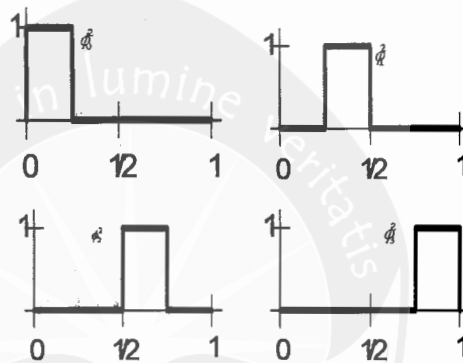
$$\phi_i^j(x) = \phi(2^j x - i), \quad i = 0, 1, 2, \dots, 2^j - 1 \quad (2.38)$$

dengan

$$\phi(x) = \begin{cases} 1, & \text{untuk } 0 \leq x < 1 \\ 0, & \text{untuk } x \text{ lainnya} \end{cases}$$

$\phi(x)$  sering disebut dengan fungsi penyekalaan (*scaling function*).

Sebagai contoh fungsi basis untuk ruang  $V_2$  seperti terlihat pada gambar berikut:



Gambar 2.3. Fungsi basis ruang  $V_2$

Fungsi wavelet yang sesuai dengan fungsi penyekalaan di atas disebut dengan *wavelet Haar*, yang diberikan dengan persamaan

$$\psi_i^j(x) = \psi(2^j x - i), \quad i = 0, 1, \dots, 2^j \quad (2.39)$$

dengan

$$\psi(x) = \begin{cases} 1, & \text{untuk } 0 \leq x < 1/2 \\ -1, & \text{untuk } 1/2 \leq x < 1 \\ 0, & \text{untuk } x \text{ lain} \end{cases}$$

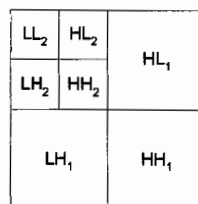
dan  $\psi(x)$  sering disebut dengan fungsi wavelet (*wavelet function* atau *mother wavelet*).

Untuk citra 2 dimensi, terdapat dua cara untuk mentransformasi atau mendekomposisi nilai-nilai pikselnya, yaitu dekomposisi standar dan takstandar. Masing-masing cara diperoleh berdasarkan alihragam *wavelet* 1-dimensi.

Untuk memperoleh dekomposisi standar suatu citra, mula-mula digunakan alihragam *wavelet* 1-dimensi pada setiap baris citra. Kemudian digunakan alihragam *wavelet* 1-dimensi pada setiap kolom citra.

Dekomposisi takstandar diperoleh dengan mengkombinasikan pasangan transformasi baris dan transformasi kolom secara bergantian. Pada langkah pertama diterapkan alihragam *wavelet* 1-dimensi pada baris, kemudian diterapkan alihragam *wavelet* 1-dimensi pada kolom. Proses tersebut diulang sesuai dengan level yang diinginkan.

Pada dekomposisi level ke-1, citra akan dibagi menjadi 4 *subband*, dinamakan *subband* HH, HL, LH, dan LL. *Subband* HH akan memberikan detail citra pada arah diagonal, *subband* HL akan memberikan detail citra pada arah horisontal, *subband* LH akan memberikan detail citra pada arah vertikal. Sedangkan *subband* LL adalah residu resolusi rendah yang mempunyai komponen frekuensi rendah, yang sering disebut dengan rata-rata citra. *Subband* LL ini akan dibagi lagi pada saat mendekomposisikan pada level yang lebih tinggi.



Gambar 2.4. Dekomposisi Citra

### 2.4.2. Operasi Pengambangan (*thresholding*)

Operasi pengambangan (*thresholding*) berfungsi mengatur jumlah derajat keabuan pada citra sesuai dengan keinginan. Derajat keabuan didapat dari mengubah nilai R, G, B dari citra berwarna menjadi sebuah nilai *grayscale*, dan hasilnya adalah citra *grayscale*. Untuk mendapatkan nilai *grayscale* yaitu :

$$s = \frac{R+G+B}{3} \quad (2.40)$$

S adalah nilai *grayscale*, R,G,B adalah *layer* matriks dari citra berwarna.

Pada operasi ini akan merubah citra menjadi citra biner (berwarna hitam / putih) berdasarkan nilai ambang (*thresholding factor*) tertentu. Jika derajat keabuan berada diatas nilai ambang akan diset hitam (dalam VB.Net nilai hitam adalah 0), dan jika derajat keabuan berada dibawah atau sama dengan nilai ambang akan diset putih (dalam VB.Net nilai putih adalah 255).

### 2.4.3. Derau (*noise*)

*Noise* adalah sinyal-sinyal yang tidak diinginkan pada citra. Biasanya sinyal-sinyal tersebut muncul pada saat proses digitalisasi citra. Pada citra digital, *noise* adalah bagian dari citra digital tersebut yang mempengaruhi kualitas dari citra tersebut. Hubungan citra digital dengan derau yang dimiliki dapat dituliskan dalam persamaan berikut :

$$f(x,y) = g(x,y) + n(x,y) \quad (2.40)$$

dengan  $f(x,y)$  adalah citra digital yang terekam, sedangkan  $g(x,y)$  adalah citra asli dan  $n(x,y)$  adalah *noise*.

Model *noise* yang dipakai pada penelitian ini adalah *Gaussian noise*. *Gaussian noise* merupakan *random noise* yaitu *noise* yang berbentuk titik-titik yang keberadaannya tersebar secara *random* pada citra digital. Pada *Gaussian noise*, warna yang ada pada tiap titik *noise* adalah berbeda, nilai warna untuk *noise* ini merupakan hasil pengurangan atau penambahan nilai warna piksel pada citra asli.

## 2.5. Bahasa Jepang

Bahasa Jepang merupakan bahasa resmi di Jepang. Bahasa Jepang juga digunakan oleh sejumlah penduduk negara yang pernah ditaklukkannya seperti Korea dan Republik Cina. Bahasa Jepang juga terdapat di Amerika Serikat (California dan Hawaii) dan Brasil akibat emigrasi orang Jepang ke sana. Namun keturunan mereka yang disebut *nisei* (generasi kedua), tidak lagi fasih dalam bahasa tersebut.

Bahasa Jepang dibagi menjadi dua bentuk yaitu *Hyoujungo* (pertuturan standar), dan *Kyoutsugo* (pertuturan umum). *Hyoujungo* adalah bentuk yang diajarkan di sekolah, digunakan di televisi dan untuk kebutuhan resmi ([http://id.wikipedia.org/bahasa\\_jepang](http://id.wikipedia.org/bahasa_jepang)).

Tulisan bahasa Jepang berasal dari tulisan bahasa China (kanji) yang diperkenalkan pada abad keempat Masehi. Sebelumnya, orang Jepang tidak mempunyai sistem penulisan sendiri. Tulisan Jepang dibagi menjadi tiga :

- a. aksara Kanji yang berasal dari China
- b. aksara Hiragana dan
- c. aksara Katakana

Selain itu, ada pula sistem alihaksara yang disebut Romaji. Romaji berasal dari kata Roma, yaitu Kerajaan Roma; dan Ji, yaitu Huruf. Jadi romaji itu huruf romawi atau alfabet.

Aksara Hiragana dan Katakana berunsur dari tulisan *kanji* dan dikembangkan pada abad kedelapan Masehi oleh rohaniawan Buddha untuk membantu melafazkan karakter-karakter China. Kedua aksara ini biasa disebut *kana* dan keduanya terpengaruhi fonetik bahasa Sansekerta, yang bisa dilihat dalam urutan aksara Kana.

Bahasa Jepang yang kita kenal sekarang ini, ditulis dengan menggunakan kombinasi aksara Kanji, Hiragana, dan Katakana. Kanji dipakai untuk menyatakan arti dasar dari kata (baik berupa kata benda, kata kerja, kata sifat, atau kata sandang). Hiragana ditulis sesudah kanji untuk mengubah arti dasar dari kata tersebut, dan menyesuaikan dengan peraturan tata bahasa Jepang. Katakana digunakan untuk menulis kata-kata yang berasal dari bahasa asing yang sudah diserap dalam bahasa Jepang (*gairaigo*).

Aksara Hiragana dan Katakana (*kana*) memiliki 48 set huruf/ suku kata. Keduanya tidak memiliki arti apapun, hanya melambangkan suatu bunyi tertentu, meskipun ada juga kata-kata dalam bahasa Jepang yang terdiri dari satu suku kata, seperti *me* (mata), *ki* (pohon), *ni* (dua), dsb.

### 2.5.1. Tulisan Katakana

Katakana adalah salah satu dari tiga cara penulisan bahasa Jepang, yang biasanya digunakan untuk menulis kata-kata yang berasal dari bahasa asing yang sudah diserap ke dalam bahasa Jepang (*gairaigo*).

Katakana diciptakan oleh Kibi Daishi, seseorang dengan jabatan tinggi di pemerintahan Kaisar Koujin, yang wafat pada tahun 776. Katakana digunakan di Jepang untuk menuliskan kata-kata serapan, onomatope, kata yang ditekankan, dan nama-nama flora dan fauna.

Katakana berasal dari karakter Cina yang dipersingkat dan dahulu digunakan oleh biksu Budha untuk menunjukkan cara pengucapakan yang benar dari tulisan Cina di abad ke-9. Suku kata tulisan Katakana, terdiri dari 48 suku kata dan dahulu disebut "tulisan pria". Tulisan Katakana adalah sebagai berikut :

ア A	カ KA	サ SA	タ TA	ナ NA	ハ HA	マ MA	ヤ YA	ラ RA	ワ WA	ン N
イ I	キ KI	シ SHI	チ CHI	ニ NI	ヒ HI	ミ MI		リ RI		
ウ U	ク KU	ス SU	ツ TSU	ヌ NU	フ FU	ム MU	ユ YU	ル RU		
エ E	ケ KE	セ SE	テ TE	ネ NE	ヘ HE	メ ME		レ RE		
オ O	コ KO	ソ SO	ト TO	ノ NO	ホ HO	モ MO	ヨ YO	ロ RO	ヲ WO	

Gambar 2.5. Huruf Dasar Katakana

Suku kata 'wi' dan 'we' sudah tidak digunakan lagi sejak ada perbaikan bahasa tahun 1946, oleh Departemen Pendidikan Jepang. Oleh karena itu, kini tulisan Katakana dasar hanya terdiri dari 46 suku kata.

Cara penulisan Katakana dasar yang benar yaitu :

A	ア ア ア	NE	ネ ネ ネ
I	イ イ イ	NO	ノ ノ
U	ウ ウ ウ	HA	ハ ハ
E	エ エ エ	HI	ヒ ヒ
O	オ オ オ	FU	フ フ
KA	カ カ カ	HE	ヘ ヘ
KI	キ キ キ	HO	ホ ホ ホ
KU	ク ク	MA	マ マ マ
KE	ケ ケ ケ	MI	ミ ミ ミ
KO	コ コ コ	MU	ム ム ム
SA	サ サ サ	ME	メ メ
SHI	シ シ シ	MO	モ モ モ
SU	ス ス ス	YA	ヤ ヤ ヤ
SE	セ セ	YU	ユ ユ ユ
SO	ソ ソ	YO	ヨ ヨ ヨ
TA	タ タ タ	RA	ラ ラ
CHI	チ チ チ	RI	リ リ リ

TSU	ツ	ッ	ツ							RU	ル	ル							
TE	テ	テ	テ							RE	レ	レ							
TO	ト	ト	ト							RO	ロ	ロ							
NA	ナ	ナ	ナ							WA	ワ	ワ							
NI	ニ	ニ	ニ							WO	ヲ	ヲ							
NU	ヌ	ヌ	ヌ							N	ン	ン							

Gambar 2.6. Cara Penulisan Katakana  
 ([http://www.learn-japanese.info/writing/kwriting\\_aiueo.html](http://www.learn-japanese.info/writing/kwriting_aiueo.html))

