

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Deteksi tepi merupakan metode yang telah banyak diimplementasikan dalam berbagai penelitian dan pada aspek yang bermacam-macam [(Edoh, 2011), (Nadernejad, Sharifzadeh, & Hassanpour, 2008), (Yazdi & Toussi, 2011)]. 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. Algoritma Sobel memiliki kelebihan yaitu peka terhadap garis diagonal dibanding garis horizontal dan vertikal, sebaliknya algoritma Prewitt lebih peka terhadap garis horizontal dan vertikal daripada garis diagonal (Bin & Yeganeh, 2012). Diantara semua algoritma deteksi tepi yang ada, Canny merupakan algoritma deteksi tepi memberikan hasil yang paling optimal untuk digunakan pada hampir segala macam kondisi [(Aborisade, 2011), (Kaur & Singh, 2012), (Lee, Li, & Hong, 2012), (Patil & Kulkarni, 2011), (Saini & Garg, 2012), (Shuker, 2010), (Supriyanti, Setiawan, Widodo, & Murdyantoro, 2012)]. Hal tersebut dikarenakan algoritma Canny dapat memberikan garis hasil deteksi tepi yang tipis dengan ketebalan 1 piksel sehingga dapat dinyatakan paling akurat dibandingkan algoritma deteksi tepi yang lain [(Hussain & Hussain, 2011), (Li, Jiang, Liu, & Chen, 2012), (Vasudevan, Dharmendra, Sivaraman, & Karthick, 2010)]. Namun algoritma Canny memiliki sebuah kekurangan, yaitu proses komputasi deteksi tepi yang dilakukan pada

cukup kompleks sehingga memakan waktu cukup lama [(Chouhan & Shukla, 2011), (Jayachandra & Reddy, 2013), (Kitti, Jaruwat, & Chaiyapon, 2012), (Maini & Aggarwal, 2010), (Singh, Gandhi, & Singh, 2011)].

Pada penelitian ini selain menggunakan metode deteksi tepi, juga membutuhkan metode pengenalan pola untuk mengenali gambar wayang. Terdapat banyak metode yang dapat digunakan untuk menangani masalah pengenalan pola, salah satunya adalah jaringan saraf tiruan [(Abiyev & Altunkaya, 2008), (Ishwarya, 2012), (Ma, Li, & Wang, 2012), (Lakhal, Tlili, & Braiek, 2010), (Miry, Miry, & Majed, 2011), (Neruda & Vidnerova, 2009), (Pal, Garg, Chechi, Kumar, & Kumar, 2010), (Prasad, Nigam, Lakhotiya, & Umre, 2013), (Reddy, Babu, & Kishore, 2011)]. Tetapi secara umum metode jaringan saraf tiruan dinilai sebagai metode terbaik diantara semua metode yang ada [(Basu, Bhattacharyya, & Kim, 2010), (Kishore & Kaur, 2012), (Li G. , 2011), (Nawi, Hamid, Ransing, Ghazali, & Salleh, 2011), (Prasad, Nigam, Lakhotiya, & Umre, 2013), (Reddy, Babu, & Kishore, 2011)]. Jaringan saraf tiruan adalah sebuah konsep yang meniru cara kerja jaringan saraf pada manusia yang dapat dilatih untuk mempelajari sesuatu. Algoritma pembelajaran yang akan digunakan dalam jaringan saraf tiruan pada metode ini adalah *Backpropagation*, karena algoritma *Backpropagation* telah banyak digunakan dalam berbagai penelitian pengenalan pola dan memberikan hasil yang cukup baik [(Acharya, et al., 2010), (Anam, et al., 2009), (Chakraborty, 2012), (Devireddy & Rao, 2009), (Raheja & Kumar, 2010)]. Salah satu penelitian yang membahas mengenai penggunaan algoritma *Backpropagation* untuk mengenali pola karakter alfabet, mengatakan bahwa

Backpropagation memiliki keunggulan yaitu dapat mengenali karakter-karakter yang tidak pernah didefinisikan di awal, karena algoritma ini memiliki kemampuan untuk belajar dari data-data yang diberikan sebelumnya (Kosbatwar & Pathan, 2012). Tetapi pada kasus-kasus tertentu, algoritma *Backpropagation* mengalami penurunan tingkat keakuratan dalam mengenali pola-pola tertentu. Misalnya pada penelitian pengenalan pola ekspresi wajah, dimana pengenalan pola untuk ekspresi wajah sedih mengalami kendala dibanding ekspresi wajah yang lain seperti ekspresi senang atau bahagia (Shukla & Kumar, 2013). Sehingga dalam penelitian ini akan dilakukan analisa seberapa besar efisiensi algoritma *Backpropagation* yang digunakan untuk mengenali pola karakter wayang. Salah satu kelemahan dari pendekatan jaringan saraf tiruan yang digunakan untuk masalah pengenalan pola citra adalah terlalu banyak jumlah masukan yang diberikan kepada jaringan saraf, dimana hal ini menyebabkan proses pelatihan serta proses pengenalan pola membutuhkan waktu pemrosesan yang cukup lama (Hamid, Nawi, Ghazali, & Salleh, 2011). Karena itu sebelum sebuah gambar wayang dijadikan masukan pada jaringan saraf tiruan, dibutuhkan sebuah proses tambahan yaitu proses ekstraksi ciri pada gambar. Proses ekstraksi ciri dilakukan dengan metode *Discrete Wavelet Transform* (DWT), dimana metode ini telah banyak digunakan dalam berbagai penelitian dan memberikan hasil yang baik [(Kishk, Ahmed, & Helmy, 2011), (Lim, Kim, & Paik, 2009), (Murty, Reddy, & Babu, 2009), (Padma & Vijaya, 2010), (Shelke & Apte, 2011), (Tawade & Warpe, 2011), (Yi, 2011)]. Jenis DWT yang digunakan pada penelitian ini adalah Haar *Wavelets*.

Pada permasalahan pengenalan pola, tingkat keakuratan merupakan parameter penentu sukses atau tidaknya proses pengenalan pola tersebut. Tingkat keakuratan pada pengenalan pola dipengaruhi oleh beberapa faktor, salah satunya kualitas gambar yang menjadi masukan. Salah satu penelitian menunjukkan bahwa tingkat keakuratan pengenalan pola yang diaplikasikan pada perangkat *mobile* mengalami penurunan, hal ini disebabkan karena kualitas gambar yang dihasilkan oleh kamera pada sebagian besar perangkat *mobile* memiliki mutu yang terbatas (Romadhon, Ilham, Munawar, Tan, & Hedwig, 2012). Karenanya, untuk mendapatkan hasil yang optimal diperlukan perangkat *mobile* dengan kamera yang memiliki kualitas baik.

2.2. Landasan Teori

2.2.1. 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.

2.2.2. 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 2.1 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 :

$$|G| = \sqrt{G_x^2 + G_y^2} \quad [2.1]$$

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

$$\theta = \arctan(G_x/G_y) \quad [2.2]$$

2.2.3. 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

G_x

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

G_y

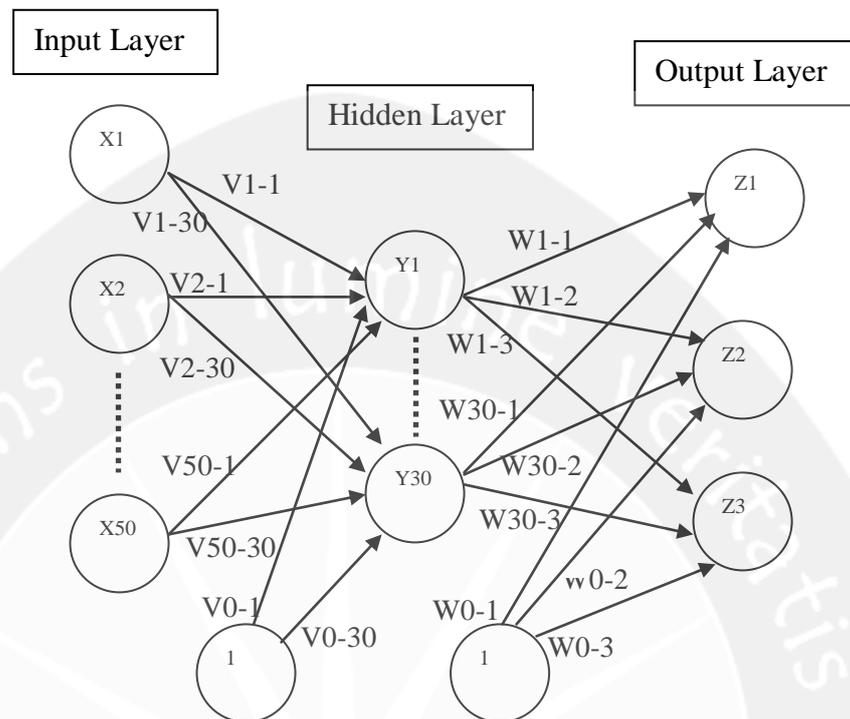
Gambar 2.2 Operator Prewitt

2.2.4. 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 suppression*). Langkah terakhir adalah melakukan proses binerisasi berdasarkan nilai *low & high threshold* yang diberikan.

2.2.5. Jaringan Saraf Tiruan (Artificial Neural Network)

Jaringan saraf tiruan adalah sebuah metode komputasi yang meniru cara kerja pada jaringan saraf pada manusia yang memiliki kemampuan untuk belajar. Metode ini memecah sebuah objek menjadi bagian-bagian kecil, kemudian masing-masing bagian akan terhubung satu sama lain seperti halnya pada *neuron*. Jaringan saraf tiruan dapat digunakan untuk mencari pola-pola data atau hubungan antar *neuron*, sehingga metode ini sangat cocok untuk digunakan dalam memecahkan masalah pengenalan pola. Terdapat dua jenis jaringan saraf tiruan, yaitu *supervised* dan *unsupervised*. *Supervised* adalah jaringan saraf tiruan yang telah dilatih dengan data-data yang disediakan sebelumnya (*training set*), kemudian jaringan saraf tiruan berusaha mengenali masukan yang diberikan berdasarkan pengetahuan yang didapat pada pelatihan. Sedangkan pada *unsupervised*, jaringan saraf tiruan tidak dilatih sebelumnya melainkan langsung belajar dari masukan-masukan yang diberikan. Pada arsitektur jaringan saraf tiruan terdapat lapisan masukan (*input layer*), lapisan keluaran (*output layer*), dan biasanya terdapat lapisan tersembunyi (*hidden layer* atau lapisan tersembunyi).



Gambar 2.3 Arsitektur Jaringan Saraf Tiruan (Kishore & Kaur, 2012)

2.2.6. Backpropagation

Backpropagation merupakan salah satu algoritma dalam pelatihan jaringan saraf tiruan. Algoritma ini bekerja dengan cara mundur, yaitu dari lapisan keluaran menuju lapisan masukan untuk memperbaiki nilai pada lapisan tersembunyi berdasarkan nilai galat yang didapat. Berikut adalah implementasi cara kerja dari algoritma *Backpropagation* (Kishore & Kaur, 2012):

1. Jaringan saraf tiruan terdiri dari tiga lapisan, yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran.

2. Setiap lapisan terdiri dari *neuron* yang saling terhubung satu sama lain.
3. Pada setiap koneksi antar *neuron* terdapat sebuah nilai bobot yang menggambarkan kekuatan dari koneksi tersebut. Bobot tersebut akan terus diperbarui berdasarkan nilai masukan dan nilai galat dengan cara propagasi mundur.
4. Lapisan tersembunyi berfungsi untuk memperbarui nilai bobot berdasarkan nilai masukan dan nilai galat.
5. Algoritma ini bekerja dalam dua tahap, tahap pertama adalah tahap pelatihan dengan sejumlah data pelatihan yang bertujuan untuk mendapatkan nilai-nilai bobot yang sesuai. Tahap kedua adalah tahap pengujian, dimana jaringan saraf tiruan telah memiliki nilai-nilai bobot tertentu dan siap digunakan untuk mengenali pola.

Persamaan untuk menghitung *neuron* pada lapisan tersembunyi :

$$Y_{in_j} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad [2.3]$$

$$Y_j = f(Y_{in_j}) \quad [2.4]$$

n = jumlah masukan

V = bobot

X = nilai masukan

Persamaan untuk menghitung neuron pada lapisan keluaran :

$$Z_{in_j} = W_{0j} + \sum_{i=1}^n Y_i W_{ij} \quad [2.5]$$

$$Z_j = f(Z_{in_j}) \quad [2.6]$$

n = jumlah masukan

W = bobot

Y = nilai masukan

Persamaan untuk menghitung nilai galat :

$$\delta_j = t_j - Z_j \quad [2.7]$$

t = nilai target

Z = nilai keluaran

Persamaan untuk fungsi aktivasi :

$$f(x) = \frac{1}{1+(e^{-x})} \quad [2.8]$$

$$f' = f(x)[1 - f(x)] \quad [2.9]$$

Persamaan untuk mengubah bobot :

$$\Delta W_{ij} = \alpha \delta_j Y_i \quad [2.10]$$

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{ij} \quad [2.11]$$

$$\Delta V_{ij} = \alpha \delta_j X_i \quad [2.12]$$

$$V_{jk}(new) = V_{jk}(old) + \Delta V_{ij} \quad [2.13]$$

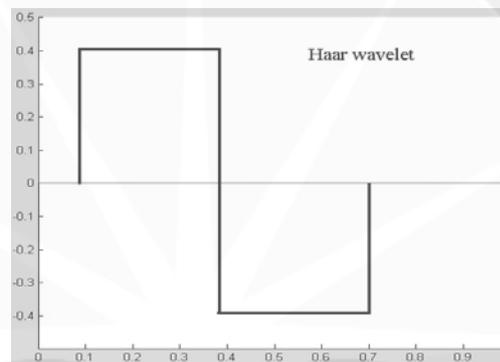
2.2.7. Ekstraksi Ciri (Feature Extraction)

Ekstraksi ciri merupakan salah satu bentuk khusus dari pengurangan dimensi (*dimensionality reduction*) yang banyak dijumpai pada kasus pengolahan citra dan pengenalan pola. Sebuah pola yang unik tersusun oleh elemen-elemen yang

berbeda antara pola yang satu dengan yang lain. Ekstraksi ciri dapat digunakan untuk mendapatkan elemen penting atau ciri khas yang dimiliki oleh sebuah citra.

2.2.8. Haar Wavelets

Haar merupakan salah satu jenis metode *wavelets* yang cukup sederhana dibandingkan dengan metode *wavelets* yang lain.



Gambar 2.4 Haar Wavelets

Haar banyak digunakan untuk menyelesaikan permasalahan yang mengutamakan waktu komputasi yang cepat. Fungsi Haar didefinisikan sebagai berikut :

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2, \\ -1 & 1/2 \leq t < 1, \\ 0 & \text{otherwise} \end{cases} \quad [2.14]$$

$$\varphi(t) = \begin{cases} 1 & 0 \leq t < 1, \\ 0 & \text{otherwise} \end{cases} \quad [2.15]$$

2.2.9. 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.