

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

Pada bab ini akan dibahas literatur dan landasan teori yang relevan dengan penelitian.

2.1 Tinjauan Pustaka

Sistem pengenalan pola plat nomor sebenarnya sudah ada sejak dahulu, tetapi baru di akhir tahun 1990-an pengenalan pola plat nomor menjadi aplikasi yang penting karena terjadinya peningkatan secara pesat terhadap jumlah kendaraan (Bailmare dan Gadicha, 2013; Shapiro et al., 2004). Sejak saat itu, banyak penelitian telah dilakukan dengan tujuan menemukan metode yang terbaik tidak hanya untuk mengenali karakter pada plat nomor tetapi juga metode untuk mengekstraksi lokasi plat nomor dari citra lingkungan.

Seperti penelitian oleh Mirashi et al. (2013) yang menggunakan beberapa operasi morfologi seperti *opening*, *dilatation*, *erosion*, *filling operation*, dan *closing* yang mampu menghasilkan objek-objek kandidat plat nomor. Kemudian objek yang memiliki kemungkinan terdapat plat nomor akan dihilangkan sehingga hanya tersisa satu daerah yang benar-benar mengandung plat nomor. Hasilnya, dari 50 citra yang diujikan, 49 citra (98%) berhasil di ekstrak.

Penelitian tentang ekstraksi lokasi plat nomor juga dilakukan oleh Ghosh et al. (2011) yang menggunakan deteksi tepi *Sobel* dan operasi morfologi seperti *dilatation* dan *erosion* untuk mengambil daerah-daerah kandidat plat nomor.

Kemudian untuk mengekstrak daerah plat nomor yang sebenarnya digunakan metode *Bounding Box Analysis*. Selanjutnya karakter pada plat nomor akan di segmentasi dengan *vertical scanning* dan *horizontal scanning* untuk kemudian dikenali menggunakan *feed-forward Neural Network*. Dari 300 gambar yang diujikan, metode ini mampu mengekstrak lokasi plat nomor dengan tingkat kesuksesan 84% dan mampu mengenali karakter plat nomor dengan tingkat kesuksesan 80%.

Pendekatan yang sedikit berbeda dilakukan oleh Na et al. (2012) untuk melakukan ekstraksi plat nomor. Pada penelitiannya digunakan *Pulse Coupled Neural Network* (PCNN) dan beberapa algoritma optimisasi *coupled parameter* untuk meningkatkan akurasi ekstraksi plat. Dari hasil pengujian, algoritma ini mampu melakukan ekstraksi plat nomor dengan kualitas yang baik. Akan tetapi algoritma ini tidak tahan terhadap citra dengan kontras rendah dan distribusi keabuan yang rumit.

Penelitian lain juga dilakukan oleh Khalifa et al. (2007) untuk membuat pengenalan pola plat nomor kendaraan di Malaysia. Pada penelitian ini digunakan metode deteksi tepi *Sobel*, dan *vertical and horizontal projection*, untuk menemukan daerah kandidat plat nomor. Kemudian daerah yang menjadi kandidat akan di verifikasi dengan memperhatikan rasio panjang dan lebar plat nomor untuk mengekstrak daerah plat nomor yang asli. Selanjutnya karakter pada plat nomor akan disegmentasi dengan metode *Connected Components* dan dikenali menggunakan *Multi-layer Perceptron Neural Network*. Metode yang diusulkan pada penelitian ini diuji dengan 150 citra dan berhasil melakukan

ekstraksi plat nomor dengan tingkat keberhasilan 92,1% dan akurasi dalam pengenalan karakter mencapai 93,2%.

Di negara Iran, penelitian tentang pengenalan pola plat nomor juga dilakukan oleh Kasaei et al. (2009) dan Kadkhoda dan Zefreh (2012). Pada penelitian yang pertama, digunakan beberapa operasi morfologi untuk melakukan ekstraksi dan segmentasi plat nomor. Sedangkan pada bagian pengenalan karakter plat nomor digunakan metode *template matching*. Hasilnya, sistem yang dibuat mampu mengekstrak plat nomor dengan akurasi 97,3% dan mampu mengenali karakter plat nomor dengan akurasi 92%. Sedangkan pada penelitian yang kedua, pendekatan yang sama dilakukan untuk melakukan ekstraksi plat nomor yaitu dengan operasi morfologi *dilatation* dan *opening operation* untuk mendapatkan daerah kandidat plat nomor. Perbedaanya terletak pada algoritma pengenalan karakter yang digunakan yaitu *Backpropagation Neural Network*. Metode yang disulkan ini berhasil mengekstraksi lokasi plat nomor dengan tingkat kesuksesan mencapai 87% dan mampu mengenali karakter dengan tingkat akurasi 91%.

Di negara Vietnam, Mai et al. (2013) menggunakan algoritma *Neural Network Backpropagation* untuk membuat sistem pengenalan plat nomor. Dari 600 citra yang terdiri dari 300 citra plat nomor berformat satu baris dan 300 citra plat nomor berformat dua baris yang diujikan, tingkat akurasi untuk plat nomor satu baris mencapai 98,33%, dan akurasi untuk plat nomor dengan format dua baris mencapai 97,67%. Dari hasil tersebut didapatkan tingkat akurasi sistem secara keseluruhan sebesar 98%.

Jenis algoritma *Neural Network* lain yang juga digunakan untuk melakukan pengenalan plat nomor kendaraan adalah *Radial Basis Neural Network* (RBFF) (Thakur dan Manoria, 2012). Penelitian ini mulanya menggunakan operasi-operasi morfologi dan deteksi tepi untuk melakukan ekstraksi dan segmentasi karakter pada plat nomor. Kemudian karakter hasil segmentasi akan dikenali menggunakan algoritma RBFF. Dari hasil pengujian, algoritma ini memiliki tingkat akurasi dalam mengenali plat secara keseluruhan sebesar 91,25%.

Selain itu, alihragam *Wavelet* juga sering digunakan pada pengenalan plat nomor terutama untuk mereduksi citra input sebelum masuk pada bagian pengenalan. Hal ini seperti yang dilakukan oleh Lee dan Hung (2013) yang menggunakan alihragam *Wavelet Haar* untuk mengenali plat nomor di Taiwan. Dari penelitiannya, proses pengenalan menggunakan alihragam *Wavelet Haar* hanya membutuhkan seperempat dari waktu biasa yang dibutuhkan untuk mengenali plat nomor tanpa *Wavelet*. Selain kecepatan pengenalan yang meningkat, hasil penelitian ini juga memiliki akurasi yang cukup tinggi yaitu sekitar 95,33%.

2.2 Landasan Teori

2.2.1 Citra Digital

Citra yang dapat dilihat oleh mata manusia tidak lain adalah sekumpulan cahaya yang dipantulkan dari suatu objek. Citra sendiri dibedakan menjadi dua jenis yaitu citra analog dan citra digital. Citra analog adalah citra yang bersifat

kontinu seperti gambar pada monitor, TV, dan foto yang tercetak di kertas foto. Sedangkan citra digital adalah sebuah array dua dimensi dengan nilai $f(x, y)$ yang telah dikonversi ke dalam bentuk diskret baik pada koordinat citra maupun intensitasnya. Elemen terkecil dari suatu citra digital adalah piksel yang menyimpan beberapa nilai antara lain koordinat dan intensitas.

2.2.2 Citra Grayscale

Citra *grayscale* atau citra beraras keabuan adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu. Hal ini melibatkan konversi dari citra berwarna menjadi citra abu-abu (Ahuja dan Kuldeepak, 2013). Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau, dan biru mempunyai intensitas yang sama. Pada citra beraras keabuan nilai intensitas dapat dinyatakan dengan nilai tunggal, dimana nilai 0 untuk merepresentasikan warna hitam dan 1 atau 255 merepresentasikan warna putih, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pikselnya.

2.2.3 Citra Biner

Citra biner adalah citra yang dikuantisasi menjadi dua nilai yaitu 0 dan 1 atau 0 dan 255 pada nilai piksel, yang mana mewakili warna hitam dan putih (Meenakshi dan Dubey, 2012). Citra biner merupakan citra yang telah melalui teknik *thresholding* yaitu pemisahan piksel berdasarkan derajat keabuan yang dimiliki. Teknik *thresholding* sendiri dikategorikan ke dalam 2 jenis yaitu *local thresholding* yang berarti ada beberapa nilai *threshold* untuk beberapa region dan

global thresholding yang berarti hanya ada satu nilai *threshold* untuk seluruh citra (Ibrahim et al., 2013). Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan seperti pada persamaan di bawah ini.

$$f(x, y)' = \begin{cases} a_1, & f(x, y) < T \\ a_2, & f(x, y) \geq T \end{cases} \quad (2.1)$$

Piksel dengan derajat keabuan lebih besar atau sama dengan nilai batas T akan diberi nilai 1 (a_2) dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0 (a_1).

2.2.4 Morfologi Matematika

Morfologi matematika adalah pendekatan berbasis topologi dan bentuk geometri untuk melakukan analisis pada citra dan menjadi suatu alat yang sangat berguna untuk mengekstrak struktur dan bentuk geometri di banyak aplikasi (Pandya dan Singh, 2011). Berikut adalah operasi-operasi yang menjadi dasar morfologi matematika:

Dilatation:

$$(F \oplus B)(x, y) = \max F(x - s, y - t) + B(s, t) \quad (2.2)$$

Operasi ini akan menyebabkan pertumbuhan ukuran objek dengan menukar tiap nilai piksel dengan nilai maksimum piksel tetangga di sekelilingnya. Hal ini akan meningkatkan jumlah piksel tepi objek di citra. Ukuran dan bentuk *structuring element* akan menentukan jumlah elemen yang akan ditambahkan ke dalam citra.

Erosion:

$$(F \ominus B)(x, y) = \min F(x + s, y + t) - B(s, t) \quad (2.3)$$

Operasi ini merupakan kebalikan dari operasi *dilatation* karena akan mengurangi jumlah piksel dengan menukar nilai tiap piksel dengan nilai minimum piksel tetangga di sekelilingnya. Hal ini berdampak pada hilangnya tepi objek pada citra.

Opening:

$$F \circ B = (F \ominus B) \oplus B \quad (2.4)$$

Operasi ini adalah operasi *erosion* yang diikuti dengan operasi *dilatation*. Operasi ini akan menghilangkan semua daerah piksel yang terang secara keseluruhan pada citra.

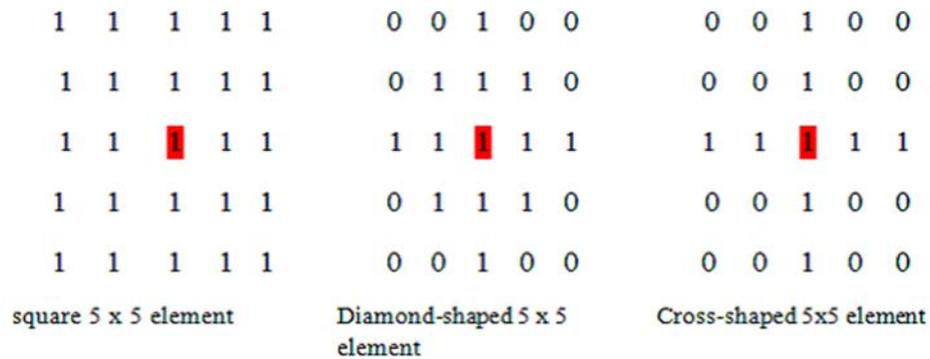
Closing:

$$F \circ B = (F \oplus B) \ominus B \quad (2.5)$$

Operasi ini adalah kebalikan dari operasi *opening* dengan menerapkan *dilatation* diikuti dengan *erosion*. Hasil operasi ini cenderung akan memperbesar daerah tepi dalam citra sekaligus mengecilkan warna *background* pada daerah tersebut. Operasi ini dapat digunakan untuk menjaga daerah *background* yang berbentuk sama dengan *structuring element*.

2.2.5 Structuring Element

Structuring element dibutuhkan untuk melakukan operasi morfologi seperti *dilatation* dan *erosion*. *Structuring element* adalah matriks ukuran $m \times n$ yang berisi nilai biner. Dalam operasi morfologi, nilai *structuring element* akan dibandingkan dengan nilai piksel tetangga pada citra input, dan diterjemahkan ke setiap piksel yang sesuai pada citra output (Radha dan Sumathi, 2012).



Gambar 2.1 Contoh Structuring Element Sederhana (Radha dan Sumathi, 2012)

2.2.6 Segmentasi Karakter pada Plat Nomor

Tahap segmentasi adalah tahapan yang penting dalam sistem pengenalan plat nomor (Shidore dan Narote, 2011; Solanki et al., 2013). Tahap segmentasi atau isolasi karakter ini bertujuan mengambil *region of interest* (dari gambar) dan membaginya menjadi karakter individu (Jusoh dan Zain, 2009). Terdapat dua jenis segmentasi yaitu segmentasi horisontal dan segmentasi vertikal (Harjani et al., 2013). Segmentasi horisontal biasanya digunakan untuk menghilangkan objek asing yang berada di atas dan bawah sekaligus menemukan batas atas dan bawah karakter pada plat nomor. Sedangkan segmentasi vertikal biasanya digunakan untuk menghilangkan gangguan yang berada di kiri dan kanan dan melakukan segmentasi untuk tiap karakter dengan tepat.

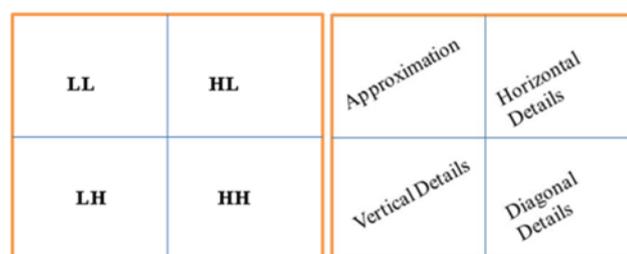


Gambar 2.2 Hasil Segmentasi Karakter (Bhalla et al., 2013)

Metode lain yang dapat digunakan untuk melakukan segmentasi karakter pada plat nomor adalah *Run Length Smearing Algorithm (RLSA)*. Pada metode ini citra inputan akan diproses secara vertikal dan horisontal (*scan-lines*). Jika jumlah piksel putih kurang dari batas ambang yang ditentukan atau melebihi batas ambang lainnya, maka piksel putih akan diubah menjadi hitam (Kumar et al., 2012).

2.2.7 Alihragam Wavelet

Wavelet adalah sinyal yang bersifat lokal dalam waktu dan skala dan umumnya memiliki bentuk yang tidak teratur (Chowdhury dan Khatun, 2012). Sedangkan alihragam *wavelet* adalah proses mendekomposisi sinyal pada frekuensi rendah dan frekuensi tinggi. Alihragam *wavelet* digolongkan menjadi *Continuous Wavelet Transforms (CWT)* dan *Discrete Wavelet Transforms (DWT)* (Madan et al., 2009). DWT adalah metode pengolahan citra digital yang sangat populer dalam beberapa tahun terakhir, terutama dalam representasi multiresolusi (Lee et al., 2012). DWT akan membagi suatu citra menjadi 4 sub-citra seperti yang ditunjukkan pada Gambar 2.3 berikut.



Gambar 2.3 Hasil Dekomposisi DWT Level-1 (Banik dan Bandyopadhyay, 2013)

Bagian LL akan berisi komponen frekuensi rendah arah vertikal dan horizontal, LH akan berisi komponen frekuensi rendah arah horizontal dan frekuensi tinggi arah vertikal, HL berisi komponen frekuensi tinggi horizontal dan frekuensi rendah vertikal, sedangkan HH berisi komponen frekuensi tinggi baik pada arah horizontal maupun vertikal. Beberapa contoh keluarga *Wavelet* adalah Haar, Daubechies, Symlets, Coiflets, BiorSplines, ReverseBior, Meyer, DMeyer, Gaussian, Mexican Hat, Morlet, Complex Gaussian, Shannon, Frequency B-Spline, Complex Morlet, dan Riyad.

2.2.8 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan adalah suatu algoritma pembelajaran bagi mesin yang terinspirasi dari struktur jaringan sel-sel saraf di dalam otak manusia karena kemampuannya melakukan perhitungan, pengenalan, pengamatan, serta pengambilan keputusan. Jaringan Saraf Tiruan dapat dibedakan menjadi dua jenis yaitu: Pembelajaran Terbimbing dan Pembelajaran Tak Terbimbing (Melin dan Castillo, 2007). Adapun letak perbedaan dari kedua jenis Jaringan Saraf Tiruan tersebut adalah sebagai berikut:

A. Pembelajaran Terbimbing (*Supervised Learning*)

Jaringan Saraf Tiruan jenis ini belajar dari sekumpulan pola masukan dan keluaran. Oleh sebab itu pada saat pelatihan, diperlukan pasangan pola yang terdiri dari vektor masukan dan vektor target yang diinginkan. Contoh dari Jaringan Saraf Tiruan yang menggunakan jenis pelatihan seperti ini dapat ditemui pada algoritma: Hebb Rule, Perceptron, Delta Rule, Backpropagation,

Hereroassociative Memory, Bidirectional Associative Memory (BAM), dan Learning Vector Quantization (LVQ).

B. Pembelajaran Tak Terbimbing (*Unsupervised Learning*)

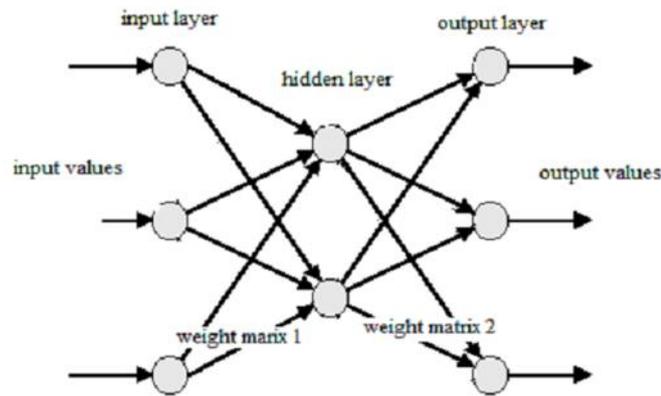
Pada Jaringan Saraf Tiruan dengan jenis Pembelajaran Tak Terbimbing tidak diperlukan adanya suatu vektor target. Vektor masukan diinputkan ke dalam Jaringan dan sistem akan mengatur dirinya sendiri sedemikian rupa sehingga dihasilkan keluaran yang konsisten seperti pola vektor masukan yang diberikan. Contoh Jaringan Saraf Tiruan yang menggunakan algoritma pelatihan ini adalah Adaptive Resonance Theory (ART) dan Competitive Learning Algorithm.

Penggunaan Jaringan Saraf Tiruan pada aplikasi pengenalan karakter secara dramatis dapat menyederhanakan kode dan meningkatkan kualitas pengenalan untuk mencapai kinerja yang baik (Patel, 2013).

2.2.9 Algoritma Momentum Backpropagation

Algoritma Backpropagation pertama kali ditemukan oleh Rumelhart, Hinton dan William pada tahun 1986 dan kemudian dikembangkan lagi bersama Mc Clelland di tahun 1988. Algoritma ini disebut sebagai propagasi balik (*Backpropagation*) karena pada saat diberikan pola pelatihan, maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk kemudian diteruskan ke unit-unit lapisan keluaran dan memberikan tanggapan atau yang disebut juga dengan keluaran. Saat keluaran jaringan tadi tidak sama dengan yang diharapkan, maka nilai keluaran akan dirambatkan mundur ke lapisan tersembunyi dan di teruskan ke unit pada lapisan masukan. Untuk lebih jelasnya, berikut penjelasan lebih rinci

dari tiga tahapan utama pada saat melakukan pelatihan, yaitu tahap propagasi maju, tahap propagasi mundur, dan tahap penyesuaian bobot.



Gambar 2.4 Arsitektur Jaringan Saraf Tiruan (Shaaban, 2011)

Pada tahap propagasi maju, sinyal masukan dipropagasikan ke lapisan tersembunyi menggunakan fungsi aktivasi yang ditentukan. Kemudian lapisan tersembunyi akan melakukan penghitungan dengan fungsi aktivasi:

$$z_j = f(v_{0j} + \sum_{i=1}^n x_i v_{ij}) \quad (2.6)$$

Hasil penghitungan tersebut akan dikirim ke lapisan sesudahnya (unit output). Sesampainya di unit output, unit tersebut melakukan perhitungan dengan fungsi aktivasi:

$$y_k = f(w_{0k} + \sum_{j=1}^p z_j w_{jk}) \quad (2.7)$$

Kemudian pada tahapan propagasi balik, Setiap unit output menerima sebuah pola target t_k lalu informasi kesalahan lapisan keluaran (δ_k) dihitung dan dikirim ke lapisan sebelumnya untuk menghitung besar koreksi bobot dan bias

(Δw_{jk} dan Δw_{0k}) antara lapisan tersembunyi dengan lapisan keluaran menggunakan rumus berikut:

$$\delta_k = (t_k - y_k) f'(w_{0k} + \sum_{j=1}^p z_j w_{jk}) \quad (2.8)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.9)$$

$$\Delta w_{0k} = \alpha \delta_k \quad (2.10)$$

Pada lapisan tersembunyi, akan dihitung informasi kesalahan lapisan tersembunyi (δ_j) dan digunakan untuk menghitung besar koreksi bobot dan bias (Δv_{ij} dan Δv_{0j}) antara lapisan masukan dan lapisan tersembunyi menggunakan rumus berikut:

$$\delta_j = \left(\sum_{k=1}^m \delta_k w_{jk} \right) f'(v_{0j} + \sum_{i=1}^n x_i v_{ij}) \quad (2.11)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.12)$$

$$\Delta v_{0j} = \alpha \delta_j \quad (2.13)$$

Selanjutnya pada tahap penyesuaian bobot, tiap unit pada lapisan output dan lapisan tersembunyi akan menyesuaikan nilai bias dan bobotnya menggunakan rumus berikut:

$$w_{jk(\text{baru})} = w_{jk(\text{lama})} + \Delta w_{jk} + (\beta \times \Delta w_{jk(\text{lama})}) \quad (2.14)$$

$$v_{ij(\text{baru})} = v_{ij(\text{lama})} + \Delta v_{ij} + (\beta \times \Delta v_{ij(\text{lama})}) \quad (2.15)$$

Proses ini akan berhenti jika nilai total error sudah lebih kecil dari nilai toleransi kesalahan (*error tolerance*) atau jumlah *epoch* melebihi maksimal.