

Sistem Pengenalan Nomor Pelat Kendaraan Menggunakan Jaringan Syaraf Tiruan Backpropagation

Afifatul Mukaroh^{*1)}, Priyo Sidik Sasongko^{*2)}

^{*}Jurusan Ilmu Komputer / Informatika, Universitas Diponegoro

¹⁾afifatul.mukaroh@gmail.com, ²⁾priyoss1234@gmail.com

Abstrak

Dalam pengawasan lalu lintas, pengenalan nomor pelat kendaraan menjadi penting untuk dilakukan. Hal ini dikarenakan pengenalan nomor pelat kendaraan memiliki banyak tujuan seperti identifikasi kendaraan curian, identifikasi kendaraan yang melanggar tata tertib, manajemen perparkiran, pengecekan keluar masuk kendaraan, dan lain sebagainya. Sayangnya jumlah kendaraan yang semakin tinggi membuat hal ini tidak bisa lagi dilakukan secara manual. Maka dari itu dibutuhkan sebuah machine vision yang dapat mengenali nomor pelat kendaraan dengan cepat dan akurasi yang tinggi. Pada penelitian ini dikembangkan sistem pengenalan nomor pelat kendaraan menggunakan jaringan syaraf tiruan backpropagation (JST BP). Sistem ini mencari model terbaik JST BP yang mampu melakukan pengenalan karakter-karakter pada pelat dengan waktu tercepat namun dengan akurasi yang tetap tinggi. Dari model terbaik yang didapatkan, sistem pengenalan nomor pelat kendaraan ini memiliki persentase keberhasilan sebesar 99.80% pada data pelat yang digunakan sebagai pembentuk model dan sebesar 96.10% pada data verifikasi.

Kata kunci : *Pengenalan Karakter, Nomor Pelat Kendaraan, Jaringan Syaraf Tiruan, Backpropagation*

Abstract

In traffic monitoring, vehicle plate number recognition is very important to do. This is caused vehicle plate number recognition has many purposes such as stealing vehicle identification, traffic violations identification, parking management, entry-exit checking, etc. Unfortunately, the high amount of vehicle makes it not be longer done manually. Therefore, it is needed a machine vision that able to recognize vehicle plate number with high speed and high accuracy. In this research it was developed a vehicle plate number recognition system using backpropagation neural network (BPNN). It found the best model of BPNN that perform characters recognition at a plate with high speed yet high accuracy. Based the best model obtained, it has 99.80% success percentage at generated data and 96.10% success percentage at verification data.

Keywords : *Character Recognition, Vehicle Plate Number, Artificial Neural Network, Backpropagation*

1 PENDAHULUAN

Dalam pengawasan lalu lintas, pengenalan nomor pelat kendaraan menjadi penting untuk dilakukan. Hal ini dikarenakan pengenalan nomor pelat kendaraan memiliki banyak tujuan seperti identifikasi kendaraan curian, identifikasi kendaraan yang melanggar tata tertib, manajemen parkir, pengecekan keluar masuk kendaraan, dan lain sebagainya. Sayangnya jumlah kendaraan yang semakin tinggi membuat hal ini tidak bisa lagi dilakukan secara manual. Maka dari itu dibutuhkan sebuah *machine vision* yang dapat mengenali nomor pelat kendaraan dengan otomatis. Bagian dari *machine vision* tersebut adalah sebuah sistem, yaitu sistem yang mampu melakukan pengenalan nomor pelat dengan cepat dan akurasi yang tinggi berdasarkan citra digital pelat yang ditangkap oleh kamera.

Sistem pada penelitian ini menggunakan beberapa tahapan pada pra-pengolahan seperti yang dilakukan pada penelitian Huang, et al. [1], Akoum, et al. [2], dan Pamungkas, et al. [3] yaitu dengan membaca histogram persebaran piksel secara horisontal maupun secara vertikal dari suatu citra pelat. Sedangkan pendekatan untuk proses klasifikasi, penelitian pada tugas akhir ini menggunakan jaringan syaraf tiruan backpropagation, seperti yang dilakukan Maarif & Sardy [4], Huang, et al [1], dan Asthana, et al [5].

Backpropagation terhitung ampuh dalam mengenali nomor pelat kendaraan karena memiliki akurasi yang tinggi. Sayangnya, backpropagation memiliki waktu komputasi yang lambat karena metode ini memiliki propagasi maju dan propagasi balik [6].

Waktu komputasi yang lambat pada jaringan syaraf tiruan backpropagation salah

satunya dipengaruhi oleh arsitektur jaringan yang kompleks, seperti jumlah layer atau jumlah neuron tiap layer-nya yang bisa jadi terlalu banyak. Jumlah neuron input pun dipengaruhi oleh panjang ciri yang diekstrak. Semakin panjang jumlah cirinya, semakin banyak jumlah neuron input.

Walau jumlah neuron input yang banyak membuat pola yang akan dikenali semakin jelas, yang itu berarti berpengaruh ke hasil akurasi yang juga semakin baik, namun hal tersebut juga berpengaruh pada waktu komputasi yang malah semakin buruk. Maka dari itu, penelitian pada tugas akhir ini mencari model terbaik untuk jaringan syaraf tiruan backpropagation, sehingga ditemukan waktu komputasi terbaik pada akurasi yang baik juga.

2 TINJAUAN PUSTAKA

Sistem pengenalan nomor pelat kendaraan ini melakukan pra-pengolahan dengan beberapa operasi citra seperti operasi histogram, binerisasi, opening, dan median filtering, sedangkan untuk klasifikasinya menggunakan jaringan syaraf tiruan backpropagation.

2.1. Histogram

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas piksel dari suatu citra atau bagian tertentu di dalam citra [7]. Secara matematis histogram citra dihitung dengan rumus 1.

$$h_i = \frac{n_i}{n}, i = 0, 1, \dots, L - 1 \quad (1)$$

n_i adalah jumlah piksel yang memiliki derajat keabuan i . n adalah jumlah seluruh piksel di dalam citra. L adalah derajat keabuan yang dimiliki citra, yaitu dari nilai 0 sampai $L - 1$ (misalnya pada citra dengan

kuantisasi derajat keabuan 8-bit, nilai derajat keabuan dari 0 sampai 255).

2.2. Binerisasi

Binerisasi adalah pengolahan citra yang bertujuan untuk mengubah citra biasa menjadi citra biner. Untuk melakukan binerisasi caranya dengan melakukan pengambangan atau *thresholding*. Operasi pengambangan mengelompokkan nilai derajat keabuan setiap piksel ke dalam dua kelas, hitam dan putih atau nol dan satu.

2.3. Opening

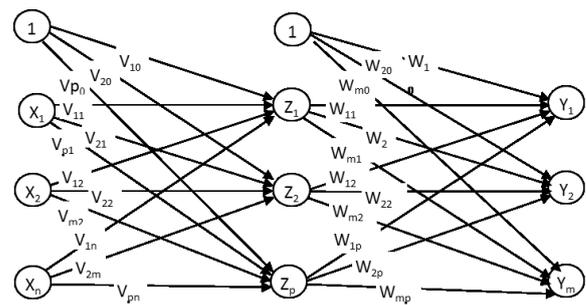
Opening merupakan operasi morfologi, yaitu metode analisis citra berdasarkan operasi ketetangaan nonlinier. Ketetangaan tersebut disebut *structuring element* (SE). Operasi dasar dari morfologi ini adalah erosi dan dilasi. Erosi dan dilasi sendiri biasanya jarang digunakan langsung. *Opening* yaitu kombinasi erosi dilasi dengan SE yang sama. Operasi ini dapat menghapus lubang putih pada objek hitam.

2.4. Median Filtering

Median filtering adalah operasi yang bertujuan untuk menghilangkan piksel-piksel noise pada suatu citra. Pada *median filtering*, suatu “jendela” (*window*) memuat sejumlah *pixel* (ganjil). Jendela digeser titik demi titik pada seluruh daerah citra. Pada setiap pergeseran dibuat jendela baru. Titik tengah dari jendela ini diubah dengan nilai median dari jendela tersebut [7].

2.5. Backpropagation

Backpropagation adalah salah satu metode pada Jaringan Syaraf Tiruan dan memiliki pelatihan jenis terkontrol (*supervised*) dimana menggunakan pola penyesuaian bobot untuk mencapai nilai kesalahan yang minimum antara keluaran hasil prediksi dengan keluaran yang nyata [8].



Gambar 1. Arsitektur Backpropagation

Langkah-langkah pelatihan backpropagation sesuai arsitektur jaringan pada Gambar 1. yaitu dengan node pada layer input berupa $X_i (i = 1, 2, 3, \dots, n)$, node pada layer tersembunyi berupa $Z_j (j = 1, 2, 3, \dots, p)$, node pada layer output berupa $Y_k (k = 1, 2, 3, \dots, m)$, bobot yang menghubungkan layer input dan layer tersembunyi berupa V_{pn} , dan bobot yang menghubungkan layer tersembunyi dan layer output berupa W_{mp} dijelaskan sebagai berikut [9]:

1. Inialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil)
2. Kerjakan langkah-langkah berikut selama kondisi pemberhentian belum terpenuhi :

Feedforward

- a. Tiap-tiap masukan menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Hitung semua keluaran di unit tersembunyi (Z_j):

$$z_{net_j} = V_{j0} + \sum_{i=1}^n X_i V_{ji} \quad (2)$$

Kemudian dengan menggunakan fungsi aktivasi sigmoid, hitung outputnya,

$$Z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (3)$$

dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit keluaran).

- c. Hitung semua jaringan di unit keluaran (Y_k)

$$y_{net_k} = W_{ko} + \sum_{j=1}^p X_j V_{kj} \quad (4)$$

Kemudian jika menggunakan fungsi aktivasi sigmoid, hitung output dengan :

$$Y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (5)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit keluaran).

Backward

- d. Unit keluaran y menerima target pola yang berhubungan dengan masukan pola pembelajarannya, hitung informasi kesalahannya:

$$\begin{aligned} \delta_k &= (t_k - Y_k) f'(y_{net_k}) \\ &= (t_k - Y_k) Y_k (1 - Y_k) \end{aligned} \quad (6)$$

Hitung suku perubahan bobot W_{kj} dengan laju perubahan α

$$\Delta W_{kj} = \alpha \delta_k Z_j \quad (7)$$

Kirimkan δ ini ke unit-unit yang ada di lapisan bawahnya.

- e. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi Z_j

$$\delta_{net_j} = \sum_{k=1}^m \delta_k W_{kj} \quad (8)$$

Faktor δ unit tersembunyi

$$\begin{aligned} \delta_j &= \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} Z_j \\ &(1 - Z_j) \end{aligned} \quad (9)$$

Hitung suku perubahan bobot V_{ji}

$$\Delta V_{ji} = \alpha \delta_j X_i \quad (10)$$

- f. Tiap – tiap unit keluaran memperbaiki bobot-bobot masukannya :
Perubahan bobot garis yang menuju unit keluaran

$$W_{kj} (\text{baru}) = W_{kj} (\text{lama}) + \Delta W_{kj} \quad (11)$$

Perubahan bobot garis yang menuju ke unit tersembunyi

$$V_{ji} (\text{baru}) = V_{ji} (\text{lama}) + \Delta V_{ji} \quad (12)$$

3 HASIL DAN PEMBAHASAN

Pembahasan berisi tentang hasil dari implementasi prapengolahan, ekstraksi ciri dan seleksi data, pelatihan, pengujian, dan verifikasi pelat pada sistem pengenalan nomor pelat kendaraan.

3.1. Prapengolahan

Pra-pengolahan yang dilakukan pada penelitian ini terdiri dari dua proses utama yaitu cropping pelat dan segmentasi karakter. Perubahan data citra pada setiap proses tersebut bisa dilihat pada Gambar 2.



(a) Citra input



(b) Citra hasil Cropping Pelat

Gambar 2. Proses Cropping Pelat

Cropping pelat adalah fungsi untuk mengubah citra input pada Gambar 2(a) menjadi citra hasil seperti pada Gambar 2(b). Cropping pelat memiliki beberapa operasi yang harus dilalui yaitu:

1. Binerisasi

Binerisasi pada penelitian ini merupakan binerisasi dengan pengambangan secara global, yang mana nilai ambang ditentukan secara otomatis dengan metode Otsu. Metode Otsu memilih nilai threshold dengan meminimalisir varians interclass dari piksel-piksel yang akan di-*threshold*.

2. Resizing

Resizing mengubah ukuran citra. Metode yang digunakan adalah metode *bicubic interpolation*. *Bicubic* menggunakan 4 x 4 piksel tetangga untuk mengambil informasi.

3. Cropping

Cropping atau pemotongan bertujuan untuk memotong citra pelat agar terpotong menjadi seperti gambar 1(b). *Region of Interest* (ROI) dari citra yang akan dipotong adalah berbentuk *rectangle*, yang mana dicari dengan membaca histogram persebaran piksel putih dan hitam pada citra.



Gambar 3. Proses Segmentasi Karakter

Segmentasi karakter mengubah sebuah citra input seperti pada Gambar 3(a) menjadi beberapa citra hasil seperti pada Gambar 3(b). Segmentasi karakter memiliki beberapa operasi yang harus dilalui yaitu:

1. Opening

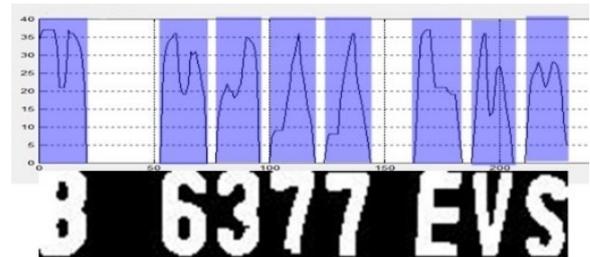
Opening bertujuan menghapus lubang putih pada objek hitam sehingga piksel-piksel *noise* di sekitar karakter merapat ke objek karakter. *Opening* merupakan operasi erosi dilanjutkan dilasi dengan Structure Element (SE) yang sama. SE yang digunakan pada penelitian ini adalah berbentuk *square*, berukuran 3 x 3, dan semua elemennya bernilai 1.

2. Median Filtering

Median Filtering bertujuan untuk menghilangkan piksel-piksel *noise* yang tersebar di seluruh citra dengan mengambil nilai median dari piksel matriks. Peneliti ini menggunakan piksel matriks 3 x 3.

3. Cropping

Proses segmentasi karakter juga menggunakan ROI berbentuk *rectangle*. Batas-batas *cropping* ditentukan dengan menghitung histogram persebaran piksel putih secara vertikal terlebih dahulu. Kemudian pada histogram tersebut dilakukan *scanning* dari kiri ke kanan. Jika ditemukan piksel putih, maka itu dianggap sebagai sebuah citra_karakter. Jika ditemukan piksel hitam, maka itu dianggap sebagai spasi antar citra_karakter. Dengan kata lain, suatu karakter di-identifikasi jika suatu *range* pada histogram bernilai tidak sama dengan nol (Pada Gambar 4 ditandai dengan *highlight* warna biru).



Gambar 4. Persebaran piksel hitam putih

3.2. Ekstraksi Ciri dan Seleksi Data

Ekstraksi ciri merupakan proses dimana citra_karakter hasil segmentasi diambil cirinya sehingga siap untuk dilakukan proses komputasi selanjutnya. Secara garis besar, langkah-langkah ekstraksi ciri pada penelitian ini adalah :

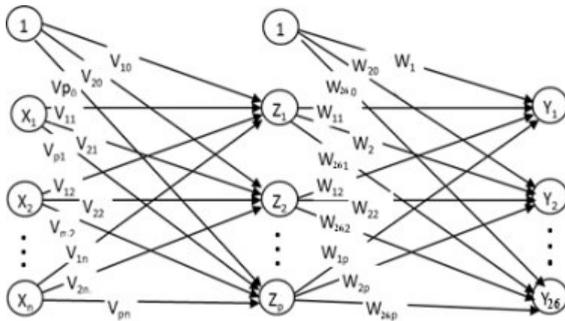
1. Membuat layer piksel berukuran m x n yang semua pikselnya bernilai nol.
2. citra_karakter yang ingin di-ekstrak, di-*resize* agar tingginya sama dengan m dan lebarnya mengikuti
3. Tempelkan citra karakter yang telah di-*resize* ke atas layer piksel dan sejajarkan sisi kirinya.
4. Layer piksel yang tersentuh piksel hitam dari citra karakter akan bernilai satu, sementara lainnya tetap bernilai nol.

- Ciri karakternya adalah piksel-piksel dari layer piksel yang dibaca dari ujung kiri atas ke ujung kanan bawah.

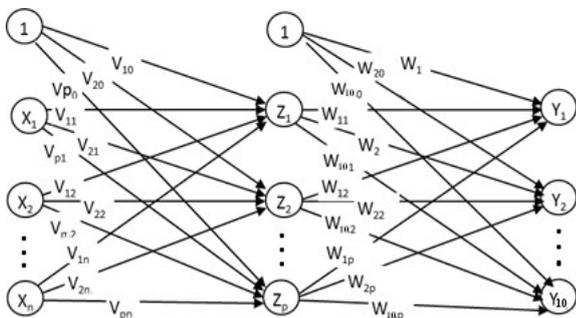
Seleksi data merupakan proses untuk membagi data ciri_karakter menjadi data latih dan data uji. Data latih digunakan dalam proses pelatihan, sedangkan data uji digunakan pada proses pengujian. Proses pembagian data menjadi data latih dan data uji menggunakan perbandingan 70% dan 30% untuk setiap jumlah karakternya.

3.3. Arsitektur Jaringan Backpropagation

Arsitektur jaringan syaraf tiruan backpropagation yang digunakan untuk Sistem Pengenalan Nomor Pelat Kendaraan ini dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Arsitektur Jaringan *Alphabet*



Gambar 6. Arsitektur Jaringan *Numeric*

Sistem pengenalan nomor pelat kendaraan ini memiliki dua arsitektur jaringan yang berbeda, yaitu arsitektur untuk jaringan *Alphabet* yang bisa dilihat gambar 5, dan arsitektur untuk jaringan *Numeric* yang bisa dilihat pada Gambar 6. Kedua arsitektur

ini terdiri dari 3 layer yaitu layer input, layer tersembunyi, dan layer output.

Jumlah neuron input baik pada jaringan *alphabet* maupun *numeric* adalah n . Nilai n adalah 35 jika panjang ciri yang digunakan adalah 35. Nilai n adalah 112 jika panjang ciri yang digunakan adalah 112. Nilai n adalah 252 jika panjang ciri yang digunakan adalah 252.

Jumlah neuron tersembunyi baik pada jaringan *alphabet* maupun *numeric* adalah p . Nilai p adalah 10, 20, atau 30 jika jumlah neuron input yang digunakan adalah 35. Nilai p adalah 20, 40, atau 60 jika jumlah neuron input yang digunakan adalah 112. Nilai p adalah 40, 80, atau 120 jika jumlah neuron input yang digunakan adalah 252.

Jumlah neuron output pada jaringan *alphabet* adalah 26. Dua puluh enam neuron output tersebut mewakili huruf A sampai huruf Z. Ketika karakter yang dikenali adalah huruf A, maka neuron pertama bernilai satu dan lainnya bernilai nol. Ketika karakter yang dikenali adalah huruf B, maka neuron kedua bernilai satu dan lainnya bernilai nol. Begitu seterusnya sampai ketika karakter yang dikenali adalah huruf Z, maka neuron ke-26 bernilai satu dan lainnya bernilai nol.

Jumlah neuron output pada jaringan *numeric* adalah 10. Sepuluh neuron output tersebut mewakili angka 0 sampai angka 9. Ketika karakter yang dikenali adalah angka 0, maka neuron pertama bernilai satu dan lainnya bernilai nol. Ketika karakter yang dikenali adalah angka 1, maka neuron kedua bernilai satu dan lainnya bernilai nol. Begitu seterusnya sampai ketika karakter yang dikenali adalah angka 9, maka neuron ke-10 bernilai satu dan lainnya bernilai nol.

3.4. Pelatihan

Pelatihan backpropagation pada penelitian ini menggunakan training dengan metode *gradient descent*. Parameter epoch

maksimum adalah 3000. Parameter untuk error maksimum adalah 0.0001. Parameter untuk learning rate adalah 0.1, 0.2, 0.3, ..., 0.9. Jumlah neuron pada layer input adalah 35, 112, dan 252. Jumlah neuron pada layer tersembunyi untuk neuron layer input 35 adalah 10, 20, dan 30, untuk neuron layer input 112 adalah 20, 40, dan 60, serta untuk neuron layer input 252 adalah 40, 80, dan 120.

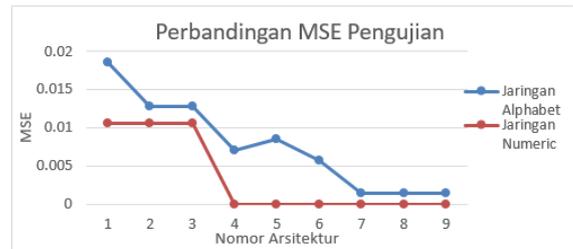
Total kombinasi pada penelitian ini adalah 81 sehingga pelatihan dilakukan sebanyak 2×81 , yaitu untuk pelatihan pada jaringan *alphabet* dan jaringan *numeric*. Dari 2×81 pelatihan tersebut akan dipilih MSE terkecil dari setiap kombinasi learning rate. Hingga nantinya tersisa 2×9 bobot hasil pelatihan.

3.5. Pengujian

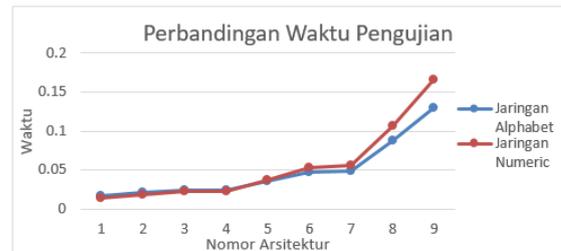
Bobot akhir sejumlah 2×9 yang dihasilkan pada proses pelatihan digunakan pada proses pengujian. Keluaran dari proses pengujian adalah nilai MSE dan waktu komputasi dari setiap bobot tersebut, yang mana bisa dilihat pada Tabel 1.

Tabel 1. Perbandingan MSE dan Waktu Pengujian

No	Node Input	Node Hidden	Jaringan <i>Alphabet</i>		Jaringan <i>Numeric</i>	
			MSE	Waktu(s)	MSE	Waktu(s)
1		10	0.018519	0.017745	0.010667	0.014826
2	35	20	0.012821	0.02187	0.010667	0.018817
3		30	0.012821	0.025002	0.010667	0.023384
4		20	0.007123	0.024729	0.000000	0.023436
5	112	40	0.008547	0.036122	0.000000	0.037205
6		60	0.005698	0.047883	0.000000	0.05333
7		40	0.001425	0.049542	0.000000	0.05574
8	252	80	0.001425	0.088173	0.000000	0.107181
9		120	0.001425	0.130008	0.000000	0.165719



Gambar 7. Perbandingan MSE Pengujian



Gambar 8. Perbandingan Waktu Pengujian

Berdasarkan grafik pada Gambar 7 dan Gambar 8, arsitektur pada jaringan *alphabet* memiliki MSE dengan nilai yang rendah pada nomor arsitektur ke-7, 8, dan 9, sedangkan waktu terkecil dari ketiga arsitektur tersebut adalah pada nomor arsitektur ke-7. Maka dari itu model terbaik untuk dipilih bobotnya adalah nomor arsitektur ke-7, yaitu jaringan dengan node input 252, node tersembunyi 40, dan learning rate 0.9. Nilai MSE pada jaringan tersebut adalah 0.001425 dan waktunya adalah 0.049542 detik.

Berdasarkan grafik pada Gambar 7 dan Gambar 8, arsitektur pada jaringan *numeric* memiliki MSE dengan nilai yang rendah pada nomor arsitektur ke-4, 5, 6, 7, 8, dan 9, sedangkan waktu terkecil dari keenam arsitektur tersebut adalah pada nomor arsitektur ke-4. Maka dari itu model terbaik untuk dipilih bobotnya adalah nomor arsitektur ke-4, yaitu jaringan dengan node input 112, node tersembunyi 20, dan learning rate 0.8. Nilai MSE pada jaringan tersebut adalah 0.000000 dan waktunya adalah 0.023436 detik.

3.5. Verifikasi

Verifikasi pengenalan pelat dilakukan dua kali yaitu pada data yang digunakan pada pembentukan model jaringan dan data yang tidak digunakan pada pembentukan model jaringan (data verifikasi).

Hasil verifikasi pada data yang digunakan pada pembentukan model jaringan, dari 62 data citra pelat, terdapat 61 citra pelat yang terkenal 100% dengan benar. Rata-rata persentase keberhasilan pada verifikasi ini adalah 99.80%. Hasil verifikasinya bias dilihat pada Tabel 2.

Tabel 2. Hasil Verifikasi pada Data Pembentuk Model

No	Observasi Visual	Hasil Pengenalan	Persentase Keberhasilan
1	AA4813HN	AA4813HN	100%
2	AB2966HI	AB2966HI	100%
3	AD2710XC	AD2710XC	100%
4	AD2901XO	AD2901XO	100%
...
15	D3614BGX	D3614BGT	87.5%
16	G2964TJ	G2964TJ	100%
17	G4639BL	G4639BL	100%
18	H2378TY	H2378TY	100%
...
59	KT5521DY	KT5521DY	100%
60	R2195SL	R2195SL	100%
61	R4776G	R4776G	100%
62	S3620CL	S3620CL	100%

Hasil verifikasi pada data verifikasi, dari 38 data citra pelat, terdapat 28 citra pelat yang terkenal 100% dengan benar. Rata-rata persentase keberhasilan pada verifikasi ini adalah 96.10%. Hasil verifikasinya bias dilihat pada Tabel 3.

Tabel 3. Hasil Verifikasi pada Data Verifikasi

No	Observasi Visual	Hasil Pengenalan	Persentase Keberhasilan
1	B6424UMO	B6424UNO	87.5%
2	B677PMV	Q6477PMK	75%
3	B6884UNH	B6884UNH	100%
4	H2158AUG	H2158AUC	87.5%
5	H2264AFG	H2264AFG	100%
...
14	AD4971RO	AD4971RD	87.5%
15	AD6393ZB	AD6393ZB	100%
16	H2981DH	H2981DH	100%
17	H3460AZ	W3460AZ	85.7%
...
35	F6049HZ	F6049HZ	100%
36	R6538VM	R6538VM	100%
37	AD3138XA	AD3138XA	100%
38	H2199TU	H2199TU	100%

4 KESIMPULAN

Sistem pengenalan nomor pelat kendaraan ini menunjukkan bahwa model jaringan syaraf tiruan backpropagation terbaik pada pengenalan karakter *alphabet* adalah arsitektur jaringan dengan jumlah node input 252, node tersembunyi 40, node output 26, dan learning rate 0.9, dimana MSE pengujiannya sebesar 0.001425 dan waktu pengujiannya sebesar 0.049542 detik.

Sedangkan model jaringan syaraf tiruan backpropagation terbaik pada pengenalan karakter *numeric* adalah arsitektur jaringan dengan jumlah node input 112, node tersembunyi 20, node output 10, dan learning rate 0.8, dimana MSE pengujiannya sebesar 0.000000 dan waktu pengujiannya sebesar 0.023436 detik.

Proses verifikasi berdasarkan model terbaik yang dihasilkan pada sistem pengenalan nomor pelat kendaraan, menunjukkan bahwa sistem ini memiliki persentase keberhasilan sebesar 99.80% pada data yang digunakan sebagai pembentuk model dan sebesar 96.10% pada data verifikasi.

Kesalahan pengenalan nomor pelat pada sistem ini mayoritas terjadi pada karakter *alphabet* dikarenakan beberapa karakter *alphabet* memiliki pola yang mirip, sedangkan karakter *numeric* seluruhnya

memiliki pola yang berbeda. Selain itu, kesalahan pada karakter alphabet juga disebabkan preprocessing yang kurang, seperti karena pemotongan yang kurang pas, angle karakter yang tidak pas, dan noise pada karakter, sehingga suatu karakter malah menyerupai karakter lain.

Penelitian selanjutnya bisa difokuskan pada proses prapengolahan citra karena proses tersebut juga berpengaruh pada akurasi dan waktu komputasi. Pengekstrakan ciri saat pengenalan nomor pelat kendaraan juga bisa dikembangkan lagi karena juga berpengaruh pada hasil pengenalan.

DAFTAR PUSTAKA

- [1] Huang, Y.-P., Chang, T.-W., Chen, Y.-R. & Sandnes, F. E., 2008. A BACK PROPAGATION BASED REAL-TIME LICENSE. *International Journal of Pattern Recognition*, 22(2), pp. 233-251.
- [2] Akoum, A., Daya, B. & Chauvet, P., 2009. TWO NEURAL NETWORKS FOR LICENSE NUMBER. *Journal of Theoretical and Applied Information Technology*, pp. 25-32.
- [3] Pamungkas, T. T., Isnanto, R. R. & Zahra, A. A., 2014. Pengenalan Plat Nomor Kendaraan Menggunakan Metode Template Matching dan Jarak Canberra. *Transient*, III(2), p. 167.
- [4] Maarif, H. A.-Q. & Sardy, S., 2006. *Plate Numer Recognition by Using Artificial Neural Network*. s.l., Prosiding Semiloka Teknologi Simulasi dan Komputasi serta Aplikasi.
- [5] Asthana, S., Sharma, N. & Singh, R., 2011. Vehicle number plate recognition using multiple. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, I(1), pp. 35-38.
- [6] Wuryandari, M. D. & Afrianto, I., 2012. Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization pada Pengenalan Wajah. *Jurnal Komputer dan Informatika (KOMPUTA)*, I(1), pp. 45-51.
- [7] Munir, R., 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Penerbit Informatika.
- [8] Suhandi, K. F., 2009. *Penerapan Jaringan Syaraf Tiruan Untuk Memprediksi Jumlah Pengangguran di Provinsi Kalimantan Timur Dengan Menggunakan Algoritma Pembelajaran Backpropagation*. [Online] Available at: <http://informatikamulawarman.files.wordpress.com/2010/02/08-jurnal-ilkom-unmul-v-5-1-0.pdf> [Accessed 19 March 2015].
- [9] Fausett, L., 1994. *Fundamentals of Neural Network, Architecture, Algorithms and Applications*. New Jersey: Prentice Hall.