

Rancang Bangun Sistem Pengenalan Plat Nomor Kendaraan Menggunakan Jaringan Saraf Tiruan *Backpropagation*

Shabri Putra Wirman, Neneng Fitrya*, Rahmat Junaidi, Noviarni Gafura Rizki
Program Studi Fisika, Universitas Muhammadiyah Riau, Indonesia

*Correspondence e-mail: nenengfitrya@umri.ac.id

Abstract

The conventional parking system allows for errors in recording, the service takes a long time, and there is no history of vehicle users. The vehicle license plate recognition system is designed as an alternative parking system that is more accurate in recording, fast service, and the presence of vehicle user data. Vehicle license plate recognition system has been designed equipped with Backpropagation Artificial Neural Network (ANN). The recognition system that has been designed in advance goes through an image processing process with grayscale, black and white, segmentation, and principal component analysis (PCA) stages. The system is integrated with backpropagation ANN with multi layer network with the best results at layer1 550 and layer2 500. The number plate recognition system is equipped with a Graphical User Interface (GUI) to display the identification results of vehicle number plate identification. The results of identification number plate recognition resulted in 100% training accuracy and 97.95918367% testing accuracy.

Keywords: Vehicle number plate, Image processing, Artificial Neural Network

Abstrak

Sistem parkir manual memungkinkan adanya kesalahan dalam pencatatan, pelayanan yang membutuhkan waktu lama, dan tidak adanya riwayat pengguna kendaraan. Sistem pengenalan plat motor kendaraan dirancang sebagai alternatif sistem parkir yang lebih akurat dalam pencatatan, pelayanan yang cepat, dan adanya data pengguna kendaraan. Sistem pengenalan plat motor kendaraan telah dirancang dilengkapi dengan Jaringan Saraf Tiruan (JST) Backpropagation. Sistem pengenalan yang telah dirancang terlebih dahulu melalui proses pengolahan citra dengan tahap grayscale, black and white, segmentasi, dan principal component analysis (PCA). Sistem diintegrasikan dengan JST backpropagation dengan multi layer network dengan hasil terbaik pada layer1 550 dan layer2 500. Sistem pengenalan plat nomor dilengkapi dengan dengan Graphical User Interface (GUI) untuk menampilkan hasil identifikasi pengenalan plat nomor kendaraan. Hasil identifikasi pengenalan plat nomor menghasilkan akurasi pelatihan 100% dan akurasi pengujian 97,95918367% pengujian.

Kata kunci: Plat nomor kendaraan, Pengolahan citra, Jaringan saraf tiruan

1. Pendahuluan

Sistem parkir yang manual memungkinkan terjadi kesalahan dalam pencatatan, lama waktu pelayanan parkir dan tidak adanya data riwayat pengguna kendaraan. permasalahan tersebut dapat diatasi dengan menerapkan sistem yang dapat mengenali plat nomor kendaraan dengan tepat, cepat, dan tercatat pada database sistem tersebut. sistem semiotomatis sejenis sudah diterapkan di berbagai area parkir seperti pusat perbelanjaan dan rumah sakit, namun pengenalan berdasarkan nomor antrian dan masukan nomor plat kendaraan tetap secara manual oleh petugas parkir (Meilandanu, 2017). Hal ini dapat diatasi dengan sistem pengolahan citra sebagai subsistem dari sistem pengelolaan parkir otomatis. Teknik pengolahan citra dapat diterapkan pada gambar plat nomor kendaraan dengan cara mengubah format gambar menjadi bentuk teks atau tulisan. Proses diawali dengan mengubah gambar menjadi citra *grayscale*, gambar diubah menjadi warna hitam pada *background* dan warna putih pada tulisan atau yang sering disebut proses *black and white* (Bhandari et al., 2016). Proses berikutnya adalah segmentasi yaitu pemotongan secara persegi pada citra *black and white* karakter huruf dan angka (Wong et al., 2013). Tahap terakhir dari proses ini adalah *Principal Component Analysis* (PCA), dimana angka dan huruf yang telah dipotong persegi akan dipisahkan bagian tulisan tanpa *background*. Data inilah yang menjadi dasar untuk pengenalan karakter angka dan huruf pada sebuah plat nomor kendaraan.

Received: 23 Juni 2021, Accepted: 23 Agustus 2022 - Jurnal Photon Vol.12 No.2

DOI : <https://doi.org/10.37859/jp.v12i2.2586>

PHOTON is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

Hasil pengolahan citra masih dalam bentuk karakter sehingga diperlukan proses lebih lanjut yaitu identifikasi setiap karakter kemudian identifikasi semua karakter dalam sebuah plat. Salah satu metode yang dapat digunakan untuk pengenalan karakteristik angka dan huruf adalah Jaringan Saraf Tiruan (JST). Salah satu algoritma pembelajaran dalam JST adalah metode *backpropagation*. Metode *backpropagation* banyak digunakan sebagai metode untuk mempelajari pasangan data *input* dan *output*, agar terbentuk suatu sistem yang dapat mengenali data dengan baik dan memberikan *output* dengan nilai *error* yang kecil (Siang, 2009). Penelitian yang menggunakan metode *backpropagation* telah banyak dilakukan dengan tingkat keberhasilan maksimal. Penelitian lain juga dilakukan oleh Wong tahun 2013 pada sistem pembayaran tiket parkir berdasarkan pengenalan karakteristik plat dengan metode *Learning Vector Quantization* dengan tingkat keberhasilan 87% (Wong et al., 2013). Sementara, Bhandary tahun 2016 menambahkan metode *Optical Character Recognition* (OCR) untuk mengenali karakter dengan tingkat akurasi 89% (Bhandari et al., 2016). Percobaan yang telah dilakukan dapat dikatakan bahwa metode *backpropagation* sangat baik digunakan untuk mengklasifikasi.

Sistem pengenalan plat nomor yang akan dirancang menggunakan proses pengolahan citra dilengkapi dengan arsitektur jaringan saraf tiruan (JST) *Backpropagation* untuk meningkatkan akurasi pengenalan plat. Sistem akan dikontrol melalui antar muka grafis (*Graphical User Interface*, GUI) yang terhubung dengan PC atau komputer. GUI dibangun menggunakan *software* MATLAB R2012a untuk memudahkan pengguna dalam menggunakan sistem pengenalan plat nomor.

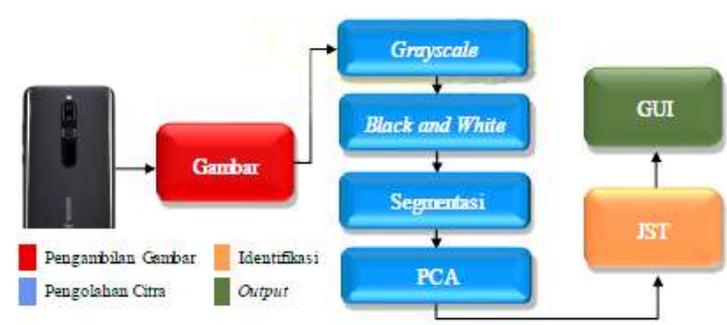
2. Metodologi

2.1 Pengambilan Image

Pengambilan *Image* yaitu proses pengambilan gambar plat nomor kendaraan di kota Pekanbaru sebanyak 120 plat dengan pemilihan model plat secara acak. Pengambilan gambar menggunakan kamera *handphone* 16 mega *pixel*. Karakter yang akan dikenali adalah angka 0 sampai 9 dan huruf A sampai Z, dimana pengenalannya dilakukan pada setiap karakter yang ada pada plat nomor kendaraan.

2.2 Perancangan Sistem

Sistem yang dibangun terdiri dari kamera dan perancangan perangkat lunak. Kamera bertujuan untuk pengambilan gambar citra plat nomor kendaraan. Perangkat lunak bertujuan untuk mengidentifikasi plat nomor kendaraan. Sistem mempunyai prinsip kerja seperti pada Gambar 1. Perancangan perangkat lunak menggunakan *software* MATLAB versi R2012a. Perangkat ini digunakan untuk mengolah citra gambar plat nomor kendaraan dan untuk mengenali masing-masing karakter yang ada pada gambar plat nomor kendaraan. Perancangan sistem ini akan ditampilkan pada *Graphical User Interface* (GUI). Perancangan *software* dilakukan dengan penetapan algoritma, pembuatan *flowchart*, dan pembuatan program.



Gambar 1. Blok diagram sistem

2.3 Pengolahan Citra

a. Konversi Citra *Grayscale*

Grayscale dilakukan dengan tujuan untuk mengkonversi citra warna RGB (*Red, Green, Blue*) menjadi citra skala keabuan. Konversi citra RGB menjadi citra *grayscale* dilakukan pada proses MATLAB. Pada MATLAB citra plat dengan warna RGB dan berekstensi JPG (*Joint Photographic Experts Group*) diinput dengan menggunakan perintah *imread*, selanjutnya dilakukan proses *grayscale*. Konversi citra RGB menjadi citra *grayscale* yaitu mencari nilai rata-rata dari total nilai RGB dengan memanfaatkan sintaks *rgb2gray* pada proses MATLAB.

b. Konversi Citra *Black and White*

Merubah citra plat ke *black and white* merupakan proses dimana citra plat yang dimasukan akan melalui proses perubahan dari citra *grayscale* menjadi citra hitam putih (*black and white*). Proses ini bertujuan untuk menentukan huruf atau angka yang dikenali nantinya adalah warna putih. Hasil citra *black and white* secara otomatis berisi nilai matriks biner yaitu nilai intensitas piksel 0 dan 1, dimana nilai intensitas piksel 0 adalah warna hitam dan nilai intensitas piksel 1 adalah warna putih.

c. Segmentasi

Segmentasi dilakukan dengan metode *thresholding* bertujuan untuk *cropping* huruf dan angka pada citra biner menggunakan proses MATLAB. Citra karakter yang dipotong terdiri dari angka sebanyak 10 karakter yaitu (0-9) dan huruf sebanyak 26 karakter yaitu (A-Z). Pada *Thresholding* dilakukan *resize image* pembentukan dimensi dari setiap karakter dengan ukuran 50 x 75 piksel. Proses segmentasi memanfaatkan fungsi sintaks *bwareaopen* untuk mendeteksi dimensi semua matriks sama dengan 1 yang mengandung warna putih. Secara otomatis citra biner akan tersegmentasi dan hasil citra segmentasi karakter akan tersimpan pada folder khusus.

d. Proses *Principal Component Analysis (PCA)*

Proses PCA sering dikenal dengan teknik pereduksi yang digunakan untuk memecahkan suatu dimensi menjadi *feature* yang lebih sedikit. Reduksi dimensi dilakukan agar mempunyai fungsi target dan *feature* yang tidak relevan dapat dihilangkan, menurunkan varian data dengan menghilangkan data yang tidak perlu dan memperbaiki nilai akurasinya.

Data yang dimasukkan pada PCA haruslah dalam bentuk vektor. Matriks gambar harus divektorisasi, semua vektor kemudian digabungkan dalam satu matriks besar. Matriks baru ini selanjutnya dipilih cirinya menggunakan fungsi PCA pada MATLAB. Proses PCA menggunakan rotasi nilai dari *eigen vector* yang menguraikan matriks kedalam sebuah *score*. *Score* dari matriks tersebut akan ditarik sebuah *feature* yang mempunyai nilai *eigen* tertinggi, maka satu matriks baru yang akan menjadi hasil reduksi.

2.4 Perancangan Arsitektur Backpropagation

Tahap ekstraksi menggunakan data gambar yang didalamnya terdapat angka dan huruf sebagai masukkan dalam metode *backpropagation* dalam menentukan suatu model pada setiap angka dan huruf. Arsitektur dibangun melalui serangkaian alur pelatihan (*training*) dan pengujian (*testing*) dengan parameter yang berbeda untuk menemukan arsitektur jaringan dengan *performance* terbaik. Parameter yang perlu diberikan sebelum melakukan pelatihan adalah Jumlah *Layer* dan *Neuron*, Metode Pelatihan, Fungsi Aktivasi, *Mean Squared (MSE)* dan Jumlah Iterasi.

a. Penentuan Jumlah *Layer* dan *Neuron*

Jumlah *hidden layer* yang akan digunakan pada sistem ini adalah 2 *hidden layer* dengan jumlah *neuron* yang divariasikan untuk menemukan *performance* dan akurasi jaringan yang optimal. Jaringan saraf tiruan ini menggunakan 2 *Hidden Layer* dengan variasi jumlah *neuron* yang berbeda-beda mulai dari 400 *neuron* hingga 600 *neuron*. Siang tahun 2009 mengatakan bahwa penggunaan lebih dari satu layer memiliki kelebihan untuk beberapa kasus, tetapi pelatihannya memerlukan waktu yang semakin lama.

b. Metode Pelatihan

Metode pelatihan yang biasa digunakan dalam MATLAB berbeda-beda, namun metode pelatihan yang akan digunakan dalam penelitian ini adalah metode *default* dari MATLAB dalam pelatihan *Backpropagation*. Hal ini didasari karena fungsi pelatihan ini memiliki kecepatan pelatihan yang tinggi (Siang, 2009).

c. Fungsi Aktivasi

Jaringan saraf tiruan menggunakan aktivasi untuk menentukan nilai keluaran pada setiap *layer* maupun pada *output*. Data yang digunakan berupa nilai 0 atau 1, sehingga fungsi aktivasi yang digunakan dalam sistem ini adalah *sigmoid biner*, baik pada *hidden layer* maupun *output*. *Sigmoid biner* dipilih karena *output* data yang diinginkan hanya berupa nilai dengan *interval* 0 hingga 1. Pada MATLAB *sigmoid biner* dituliskan sebagai *logsig* (Siang, 2009).

d. Mean Squared Error (MSE)

Pelatihan jaringan saraf tiruan membutuhkan parameter MSE (*Mean Squared Error*) untuk menentukan nilai toleransi kesalahan atau *error* dari jaringan yang akan dilatih. Siang tahun 2009 mengatakan bahwa pelatihan dengan *backpropagation* tidak akan menghasilkan MSE = 0. Berdasarkan hal tersebut, pada penelitian ini diharapkan MSE jaringan yang dibentuk dapat mencapai ketelitian 0,00001 dengan memberikan perintah *N.trainparam.goal* = 0,00001(Siang, 2009).

e. Jumlah Iterasi (Epoch)

Pelatihan jaringan saraf tiruan membutuhkan banyak perulangan (*iterasi*) untuk mencapai nilai MSE yang telah ditentukan. Data semakin banyak yang harus dikenali iterasi yang dibutuhkan semakin lama, terkadang untuk mengenali data yang rumit membutuhkan waktu berhari hari untuk mencapai nilai MSE yang telah ditentukan. Untuk menghindari proses iterasi yang berjalan terus menerus tanpa henti, maka jumlah iterasi perlu dibatasi. Jumlah iterasi ini disebut *Epoch* pada pemograman MATLAB (Siang, 2009). Pada penelitian ini iterasi dibatasi pada nilai 50000 dengan memberikan perintah *N.trainparam.epoch* = 50000.

f. Iterasi Bobot

Inisialisasi Bobot jaringan dilakukan sebelum memulai pelatihan,yaitu dengan menggunakan perintah *N = init(N)*. Bobot awal diberikan MATLAB secara otomatis dengan nilai *random* yang merupakan nilai acak kecil dengan interval-0,5 hingga 0,5 atau -1 hingga 1 atau dalam interval yang lain.

2.5 Pelatihan Arsitektur Backpropagation

Arsitektur *Backpropagation* dirancang dengan melakukan serangkaian pelatihan (*training*) menggunakan gambar plat nomor kendaraan sebanyak 90 plat. Tahap pelatihan terdiri dari 36 karakter sebagai data latih yang mewakili masing-masing huruf (A-Z) sebanyak 26 karakter dan angka (0-9) sebanyak 10 karakter. Data latih berupa karakter perlu dilakukan normalisasi kedalam *range* nol (0) hingga satu (1) sebelum dilakukan pelatihan. Hasil semua kumpulan citra latih tersebut kemudian akan diterapkan dalam sistem pengujian. Alur pelatihan dan pengujian *backpropagation* dapat dilihat pada Gambar 2.



Gambar 2. Alur pelatihan dan pengujian *backpropagation*

2.6 Pembuatan Graphical User Interface (GUI)

GUI merupakan tampilan grafis yang digunakan sebagai antar muka dalam mengoperasikan sistem yang akan dirancang. GUI pada sistem ini memiliki 4 bagian utama, yaitu bagian *Input*, *Processing*, *Output* dan Riwayat Pengguna Kendaraan. Pada bagian *Input* merupakan bagian untuk memasukkan file citra plat kendaraan kemudian menampilkannya. Bagian *processing* akan menampilkan hasil proses sistem konersi pengolahan

citra menjadi citra *grayscale* dan menampilkan hasil segmentasi citra per-karakter. Bagian *output* menampilkan hasil pengenalan jaringan saraf tiruan terhadap citra karakter, hasilnya ditampilkan berbentuk teks atau tulisan dari karakter yang telah dikenali. Bagian riwayat pengguna kendaraan juga merupakan bagian dari *output* yang akan menampilkan informasi pengguna yang memiliki kendaraan dengan plat nomor yang telah dikenali. Tampilan GUI dapat dilihat pada Gambar 3



Gambar 3. Tampilan GUI

2.7 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui seberapa akurat performa metode yang digunakan untuk pengenalan karakter. Akurasi yang paling ideal adalah 100%, semakin akurasinya mendekati ke 100% akurasi dinyatakan baik. Pengujian sistem bertujuan untuk mengetahui apakah sistem yang telah dibuat sudah sesuai dengan yang diharapkan. Pada tahap pengujian ini akan ditampilkan pada GUI menggunakan *software* MATLAB R2012a hasil dari pengujian jaringan saraf tiruan *backpropagation*.

3. Hasil dan Pembahasan

Hasil penelitian yang dilakukan peneliti di Ruang Sebayang Rumah Sakit Jiwa Tampan Provinsi Riau pada tanggal 03 - 13 Maret 2020 memperoleh hasil mayoritas kemampuan mengontrol halusinasi pendengaran responden setelah diberikan Terapi Aktivitas Kelompok dalam kategori mampu mengontrol yang berjumlah 16 responden (100.0%), sedangkan responden yang memiliki kategoritidak mampu mengontrol berjumlah 0 responden (0.0%). Hal ini dapat dilihat bahwa rata-rata nilai mean sebelum dilakukan Terapi Aktivitas Kelompok adalah 14.37%, dan sesudah dilakukan Terapi Aktivitas Kelompok adalah 20.68% dengan selisih 6.31% dan dibuktikan dengan P-Value = 0,01 < 0,05, yang artinya ada pengaruh Terapi Aktivitas Kelompok : Stimulasi Persepsi Sesi 1-3 Terhadap Kemampuan Mengontrol Halusinasi Pendengarann Pada Pasien Skizofrenia.

3.1 Hasil Pengolahan Citra

Data citra yang didapatkan dari proses pengumpulan data berupa citra plat nomor kendaraan dengan format *.jpg. Citra plat nomor kendaraan yang dihasilkan berbentuk citra warna (data RGB). Data RGB plat nomor kemudian dikonvers menjadi citra *grayscale* (warna keabu-abuan) menggunakan program MATLAB. Hasil konversi citra RGB ke *grayscale* dapat dilihat pada Gambar 4.



Gambar 4. Hasil proses citra *grayscale*

3.2 Hasil Pengolahan Citra *Black and White*

Received: 23 Juni 2021, Accepted: 23 Agustus 2022 - Jurnal Photon Vol.12 No.2

DOI : <https://doi.org/10.37859/jp.v12i2.2586>

PHOTON is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

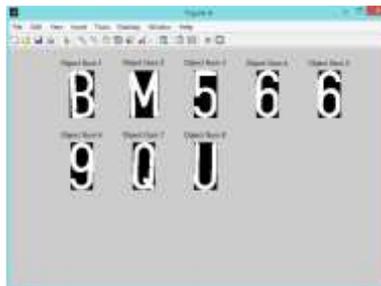
Hasil citra grayscale kemudian dikonversi kembali menjadi citra black and white (warna hitam dan putih) untuk mendapatkan nilai biner (nilai 0 dan 1) dari citra plat nomor kendaraan. proses konversi dari citra *grayscale* menjadi citra *black and white* dilakukan menggunakan program MATLAB dengan perintah *thresholding*. (Tjolleng, 2017). Hasil proses citra *black and white* dapat dilihat pada Gambar 5.



Gambar 5. Hasil proses citra *black and white*

3.3 Hasil Pengolahan Citra Segmentasi

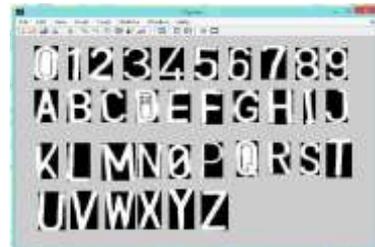
Proses selanjutnya pada citra adalah melakukan proses segmentasi. Segmentasi dilakukan untuk memotong plat nomor kendaraan menjadi beberapa bagian sehingga dihasilkan karakter angka dan huruf yang mewakili setiap karakter yang ada pada nomor plat. Hasil *cropping* akan diperoleh nilai vektor karakter angka dan huruf dari gambar plat kendaraan yang akan digunakan pada proses pelatihan dan pengujian pada Jaringan Saraf Tiruan *Backpropagation*. Hasil citra segmentasi dapat dilihat pada Gambar 6.



Gambar 6. Hasil proses citra segmentasi

3.4. Arsitektur Jaringan Saraf Tiruan *Backpropagation*

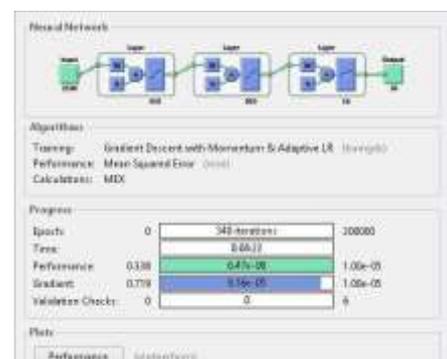
Arsitektur *Backpropagation* dirancang untuk dapat mengenali perbedaan masing-masing karakter. Proses ini menentukan dimana banyaknya *training* yang dilakukan dan terhadap jumlah *layer* yang digunakan dengan menggunakan nilai bobot dan bias. Pengacakan nilai bobot dan bias dilakukan pada tahap proses pelatihan (*training*). Bobot ditentukan dari lapisan masukan menuju lapisan tersembunyi dan bias yang berasal dari lapisan tersembunyi menuju lapisan keluaran pada acak (*random*). *Random* dilakukan dengan angka kecil mulai dengan *range* -1 sampai 1, kemudian menentukan maksimum *epoch*, *layer*, *neuron*, toleransi *error* sebagai tanda kondisi berhenti, dan fungsi aktivasi. Fungsi aktivasi menggunakan *Logsig*, *Tansig* dan *Mean Squared Error* (MSE) dapat menghitung rata-rata dari *error* yang sangat kecil yang dihasilkan prediksi (Siang, 2009). *Input* yang digunakan untuk setiap *neuron* dimulai dari 400 hingga 600. *Input* yang digunakan adalah 36 karakter terdiri dari huruf (A-Z) dan angka (0-9) dengan jumlah *output* yang sama yaitu 36 (jumlah semua karakter). Masing-masing karakter dapat dilihat pada Gambar 7.



Gambar 7. Karakter angka dan huruf

3.5 Hasil Pelatihan Arsitektur Jaringan

Data karakter yang telah didapatkan dari pengolahan citra digunakan sebagai data *input* pada proses JST dan didapat data kumpulan karakter dari 90 plat sebagai data latih sebanyak 638 data. Pelatihan arsitektur jaringan dilakukan dengan metode pelatihan, yaitu *trainingdx*. Metode pelatihan dilakukan dengan 6 variasi jumlah *hidden layer* dan *neuron*. Hasil pelatihan *trainingdx* menghasilkan *performance* dengan ketelitian MSE. Tampilan proses pelatihan dapat dilihat pada Gambar 8.



Gambar 8. Hasil *running* pelatihan JST

Arsitektur yang dirancang pada *Multi layer Neural Network* (550, 500, 36) dengan hasil pelatihan menunjukkan nilai iterasi maksimum (*epoch*) berhenti pada 348 (*epoch*) dalam waktu 4 menit 22 detik, sedangkan nilai target *error* (*performa*) yang didapatkan yaitu 0,00000647.

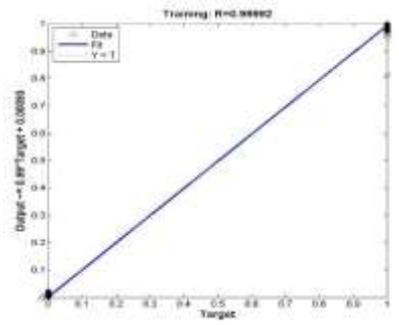
3.6 Hasil Metode Pelatihan *Trainingdx*

Pelatihan dengan *tools trainingdx* rata-rata tidak mencapai *performance* yang diharapkan. Variasi layer yang menghasilkan nilai MSE dibawah 0,00001 adalah pada layer1 50 dan layer2 500. Perbandingan *performance* untuk masing-masing variasi jumlah *layer* dan *neuron* dapat dilihat pada Tabel 1.

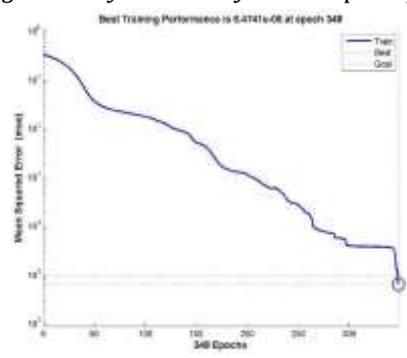
Tabel 1. *Performance* pelatihan *trainingdx*

Percobaan	Variasi		<i>Performance</i> (MSE)
	Layer 1	Layer 2	
1	500	400	0,000035
2	500	450	0,00026
3	550	450	0,000077
4	550	500	0,0000064
5	600	500	0,000034
6	600	550	0,000035

Berdasarkan Tabel 1 dapat dilihat tingkat *performance* yang didapat setiap variasi *layer* dan *neuron* berbeda, semakin mendekati *error* (0,00001) semakin baik jaringan dalam mengenali data latih. Dapat dilihat *performance* pada variasi yang ketiga *Layer1* 550 *Layer2* 500 memiliki nilai MSE jauh lebih kecil dari variasi lainnya, sehingga memiliki nilai keluaran yang jauh lebih dekat dengan target yang diharapkan. Data keluaran jaringan pada *Layer1* 550 *Layer2* 500 terdistribusi ke nilai yang mendekati target, seperti pada grafik *regression* pada Gambar 9. Distribusi hasil keluaran jaringan mendekati daerah nol dan satu seiring bertambahnya iterasi, kemudian pada iterasi (*epoch*) ke-348 MSE telah mencapai <0,00001 dan iterasi dihentikan. Grafik *performance* dapat dilihat pada Gambar 10.



Gambar 9. Grafik *regression* *layer1* 550 *layer2* 500 pada pelatihan *trainingdx*



Gambar 10. Grafik *performance* *layer1* 550 *layer2* 500 pada pelatihan *trainingdx*

3.7 Analisis dan Pengujian Arsitektur Jaringan

Arsitektur jaringan yang telah dilatih memerlukan analisis dan pengujian untuk melihat keakuratan dari *performance* yang diperoleh (Siang, 2009). Pengujian dilakukan dalam dua tahap, yaitu menguji jaringan dengan data latih dan menguji jaringan dengan data uji. Pengujian yang pertama dilakukan dengan memasukkan data latih pada jaringan yang telah dibangun, kemudian menganalisis keluaran yang akan dihasilkan. Pengujian yang kedua dilakukan dengan memasukkan data uji pada jaringan, kemudian menganalisis keluaran yang telah dihasilkan.

a. Pengujian dengan Data Latih

Pengujian jaringan dengan data latih dilakukan setelah siklus pelatihan selesai. Simulasi dilakukan dengan melakukan proses *feedforward* dari algoritma *backpropagation* menggunakan arsitektur pelatihan dan bobot hasil pelatihan. Data *Input* yang digunakan yaitu data yang sama dengan data pelatihan, bertujuan untuk menganalisis keakurasian jaringan yang telah dilatih dalam mengenali data latih (Siang, 2009). Analisa dilakukan dengan kuantitatif yaitu menghitung persentase keluaran dari hasil simulasi yang memiliki nilai mendekati target. Hasil uji seluruh variasi metode pelatihan dan jumlah *layer* dan *neuron* dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian dengan data latih

Variasi <i>neuron</i>	<i>Performance</i>	Akurasi jaringan	Total benar dari
-----------------------	--------------------	------------------	------------------

Layer 1	Layer 2	(MSE)		638 data
500	400	0,00003511530	99,8432602	637
500	450	0,00026794349	98,7460815	630
550	450	0,00007005127	99,5297806	635
550	500	0,0000647405	100	638
600	500	0,00003766680	98,5903417	629
600	550	0,00003591074	99,2163009	633

Akurasi jaringan tertinggi mencapai 100% dari 638 data total benar diperoleh dari jaringan yang dibentuk dengan metode pelatihan *traingdx* yaitu pada *Layer1* 550 *Layer2* 500.

b. Pengujian dengan Data Uji

Pengujian jaringan dengan data uji diperlukan untuk mengetahui apakah jaringan masih dapat mengenali data yang tidak termasuk dalam data latih, kemudian dapat dikelompokkan pada kategori yang benar. Tahap ini hampir serupa dengan melakukan pengujian dengan data latih, yaitu hanya memproses algoritma *feedforward*. Apabila semua data keluaran mendekati data target, maka jaringan *backpropagation* yang dibentuk dapat dikatakan berhasil (Siang, 2009). Hasil uji seluruh variasi metode pelatihan dan jumlah *layer* dan *neuron* dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian dengan data uji

Variasi Neuron		Performance	Akurasi jaringan	Total benar dari
Layer 1	Layer 2	(MSE)		245 data
500	400	0,0001033	97,1428571	238
500	450	0,0000706	91,8367347	225
550	450	0,0002016	82,4489796	202
550	500	0,0000720	97,9591837	240
600	500	0,0000366	92,2448981	226
600	550	0,0000694	96,3265306	202

Akurasi jaringan terbaik dalam mengenali data uji diperoleh dari hasil pelatihan metode *traingdx* pada variasi *Layer1* 550 *Layer2* 500 dengan tingkat akurasi sebesar 97,95918367%. Jaringan ini dapat mengenali 240 data dengan benar dari 245 total data uji.

3.8 Tampilan Identifikasi Plat Nomor Kendaraan pada GUI

Model JST yang memiliki hasil terbaik pada *layer1* 550 *layer2* 500 digunakan sebagai sistem pengenalan plat nomor kendaraan. Sistem dilengkapi dengan *Graphycal User Interface* (GUI). Pada GUI terdapat *pushbotton Open File* sebagai membuka dokumen gambar plat nomor. Program yang *run* akan tertampil item *input* gambar, proses pengolahan citra, hasil pengenalan plat nomor kendaraan dari riwayat pengguna sebagai *database* pemilik kendaraan. Tampilan GUI identifikasi plat nomor kendaraan dapat dilihat pada Gambar 11.

3.9 Pengujian Sistem

Pengujian sistem pada jaringan dengan data uji untuk mengetahui apakah jaringan masih dapat mengenali data yang tidak termasuk dalam data latih, kemudian dapat dikelompokkan pada kategori yang benar. Tahap ini hampir serupa dengan melakukan pengujian dengan data latih, yaitu memproses algoritma *feedforward* (Siang, 2009). Gambar 11 menunjukkan bahwa identifikasi plat nomor kendaraan dengan tingkat akurasi keberhasilan 100%. Riwayat pengguna pada Gambar 11 menunjukkan sebagai pengguna dari plat nomor kendaraan tersebut.



Gambar 11. Tampilan GUI identifikasi plat nomor kendaraan

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut bahwa sistem pengenalan plat nomor kendaraan yang telah berhasil dirancang dilengkapi dengan JST *backpropagation* memiliki hasil terbaik pada jumlah layer 550,500 (layer1, layer2) dengan tingkat akurasi pelatihan sebesar 100% dan akurasi pengujian 97,95918367%.

Ucapan Terimakasih

Peneliti mengucapkan terimakasih kepada Lembaga Penelitian dan Pengabdian masyarakat Universitas Muhammadiyah Riau atas seluruh bantuannya.

Daftar Pustaka

- Bhandari, A. K., Kumar, A., Chaudhary, S., & Singh, G. K. (2016). A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms. *Expert Systems with Applications*, 63, 112–133. <https://doi.org/10.1016/j.eswa.2016.06.044>
- Meilandanu, A., Rasmana, S. T., & Susanto, P. (2018). Pengenalan Plat Nomor Kendaraan Menggunakan Metode Diagonal Distance Feature. *Journal of Control and Network Systems*, 7(1), 50-61.
- Siang, J. J. (2009). Jaringan Syaraf Tiruan Dan Pemrogramannya Dengan Matlab. Yogyakarta: Andi Yogyakarta.
- Tjolleng, A. (2017). Pengantar pemrograman MATLAB: Panduan praktis belajar MATLAB. Jakarta: Elex Media Komputindo.
- Wong, N. P., Hardy, & Maulana, A. (2013). Aplikasi Pengenalan Karakter Pada Plat Nomor Kendaraan Bermotor Dengan Learning Vector Quantization. *Prosiding Seminar Nasional Sistem Informasi Indonesia 2013*, 2–4.