

# Pengenalan Plat Kendaraan Bermotor Menggunakan Metode Gradien Karakter dan BPNN (Backpropagation Neural Network)

(*Licences Recognition Using Gradient Character and Backpropagation Neural Network*)

Muhammad Ridwan Ali<sup>[1]\*</sup> dan Ario Yudo Husodo<sup>[2]</sup>

<sup>1</sup>Dinas Komunikasi, Informatika dan Statistik Provinsi NTB  
Jalan Udayana No. 14 Mataram, Lombok NTB, INDONESIA

<sup>2</sup>Jurusan Teknik Informatika, Fakultas Teknik, Universitas Mataram  
Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia

*Email:* ridwanali@unram.ac.id, ario@unram.ac.id

*\*Penulis korespondensi*

**Abstract** License plates are a unique feature to identify a vehicle in the combination between letters and numbers. Feature extraction needed to identify each letter and number in a digital image. There are several methods in feature extraction, one of them uses a gradient feature extraction. In this research, an application program to identify the license plate is a character gradient method and backpropagation neural network (BNN). First, the digital image is cropped to get a license plate then segmented to generate each character. The next step is the extraction feature using Character gradient to get a particular feature from each character. Backpropagation neural networks are used as data classification. This research consists of two types of testing: performance analysis based on hidden layers and feature quantity in training datasets and license plate data. From the result, we can conclude that the quantity of features affects the system performance. The highest performance rate in the first scenario test is feature 48 with 60 hidden layers, and in the second scenario, the highest is feature 108 with 60 hidden layers. The lowest performance rate is shown in feature 12 with 20 hidden layers.

**Key words:** License plate, character gradient method, backpropagation neural network, feature, hidden layer

## I. PENDAHULUAN

Plat kendaraan bermotor merupakan ciri atau tanda pengenal dari suatu kendaraan yang diberikan oleh kepolisian. Semua jenis kendaraan bermotor memiliki nomor polisi pada plat kendaraan yang berbeda-beda untuk mengidentifikasi kendaraan tersebut.

Pengenalan plat kendaraan bermotor secara manual yaitu dengan cara melihat dan mencatat di komputer saat ini masih banyak dilakukan. Hal ini melibatkan manusia yang memungkinkan terjadinya kesalahan dalam hal pencatatan. Dengan cara ini kemungkinan terjadinya kesalahan akibat kelalaian manusia (*human error*) dalam proses memasukkan data cukup tinggi dan waktu pengenalan plat kendaraan yang lebih lama.

Dengan adanya teknologi pengenalan pola dan pengolahan citra, data berupa citra dari plat kendaraan yang diambil melalui kamera dapat diolah menjadi teks. Pengolahan citra dilakukan dengan memotong (*cropping*) latar belakang citra sehingga dihasilkan citra plat kendaraan bermotor. Selanjutnya dilakukan pemisahan (*segmentation*) pada plat kendaraan bermotor tersebut sehingga dihasilkan pemisahan setiap karakternya. Kemudian dari masing-masing karakter tersebut dilakukan ekstraksi ciri (*feature extraction*) dengan mengambil ciri khusus menggunakan metode character gradient yang ada pada citra karakter tersebut sehingga masing-masing citra karakter tersebut memiliki ciri khas yang membedakannya dengan citra karakter lain. Dengan pengenalan pola dilakukan klasifikasi dari hasil ekstraksi ciri dengan menggunakan metode *Backpropagation Neural Network* (BPNN).

Berdasarkan penelitian sebelumnya tentang pengenalan karakter alpabet dari tulisan tangan berdasarkan fitur geometri[1], maka paper ini merupakan implementasi dari penelitian tersebut dengan menambahkan karakter numerik dengan studi kasus plat kendaraan. Penelitian ini bertujuan untuk mengetahui hasil pengenalan karakter yang didapatkan menggunakan metode klasifikasi BPNN dengan ekstraksi fitur gradien karakter berdasarkan fitur tepi dan gradiennya[2] pada plat kendaraan.

## II. TINJAUAN PUSTAKA

Saat ini telah berkembang beberapa metode untuk pengenalan objek untuk kasus-kasus tertentu, seperti yang bisa dilihat pada penelitian yang berjudul *Real time face recognition engine using compact features for electronics key*[3], *Porn picture files scanning method based on the number of picture files in a directory*[4], *Real Time Face Recognition Based on Face Descriptor and Its Application oleh*[5], dan *Fast pornographic image recognition using*

*compact holistic features and multi-layer neural network*[6].

Penelitian terdahulu lainnya yang terkait dengan penelitian ini antara lain; penelitian tentang *A Feature Extraction Technique Based on Character Geometry for Character Recognition*. Dalam penelitian ini, dilakukan proses ekstraksi fitur berdasarkan geometri dari karakter untuk pengenalan karakter. Metode yang diusulkan tersebut telah dilakukan uji coba menggunakan JST dengan 650 citra basis data. Algoritma tersebut juga telah dilakukan uji coba dengan 130 citra uji dan 6 di antaranya terjadi kesalahan pengenalan[1].

Penelitian tentang *A Review of Gradient-based and Edge-based Feature Extraction Methods for Object Detection* melakukan studi tentang berbagai teknik ekstraksi fitur berbasis gradien. Kesimpulannya, telah diulas beberapa paper penelitian tentang deteksi objek, khususnya kita membagi penelitian deteksi objek yang konvensional ke dalam 3 kategori berdasarkan objek yang dipelajari. Ketiga kategori tersebut adalah: deteksi wajah, deteksi obyek dan deteksi manusia. Dan juga, membagi fitur gambar yang digunakan menjadi fitur berbasis gradien dan fitur berbasis tepi. dari ulasan beberapa paper, dapat disimpulkan bahwa fitur berbasis gradien pada umumnya lebih disukai dibandingkan dengan fitur berbasis tepi. Kesimpulan ini berdasarkan beberapa observasi yaitu pertama, fitur berbasis gradien lebih banyak digunakan dalam pengenalan pola. Kedua, walaupun fitur berbasis tepi telah terbukti sebagai pendekatan yang sangat baik, akan tetapi komponen gradien dibutuhkan untuk meningkatkan ketahanan pada suatu permasalahan atau tugas yang lebih kompleks. Penelitian selanjutnya akan menyelidiki lebih dalam kedua pendekatan ini dan menelusuri perbaikan yang potensial dari suatu pengenalan objek[2].

Penelitian yang berjudul *Character Recognition Using Matlab's Neural Network Toolbox* melakukan studi tentang berbagai teknik klasifikasi berbasis fitur untuk pengenalan karakter tulisan tangan secara offline. Penelitian ini mengusulkan teknik pengenalan karakter yang optimal. JST biasa digunakan untuk melakukan pengenalan karakter karena toleransi noise yang tinggi dan memiliki kemampuan untuk menghasilkan hasil klasifikasi yang sangat baik. Langkah ekstraksi fitur pada pengenalan karakter adalah yang paling penting. Fitur-fitur yang buruk akan menghasilkan tingkat klasifikasi yang buruk oleh JST[7].

Penelitian tentang *Neural Based Handwritten Character Recognition* melakukan beberapa pendekatan inovatif untuk pengenalan karakter tulisan tangan. Ekstraksi fitur yang digunakan yaitu Fusion method. Fusion method memberikan cara unik untuk menggabungkan potensi pendekatan berbasis ring dan sector serta simetri karakter digabungkan ke dalam JST untuk *recognition rate* yang lebih baik. *Recognition rate* yang diperoleh pada penelitian ini sangat memuaskan[8].

Penelitian yang berjudul *Sistem Deteksi dan Pengenalan Karakter pada Plat Nomor Kendaraan Dengan*

Metode Backpropagation. Dalam penelitian tersebut hasil rata-rata pengenalan karakter pada plat nomor kendaraan sebesar 77,44 % dari 30 plat nomor pengujian. Hasil pengujian terkendala pencahayaan yang ada pada citra input jika kondisi pencahayaan baik atau cerah maka karakter mudah untuk dikenali, tetapi jika pencahayaan kurang karakter pada citra input susah untuk dikenali. Posisi mobil saat pengambilan citra sebagai input juga berpengaruh untuk sempurnanya pengenalan karakter pada plat nomor mobil tersebut, seperti posisi citra miring dan posisi pengambilan tidak tegak lurus[9].

Penelitian tentang *Realisasi Pengenalan Plat Nomor Kendaraan Dengan Metode Histogram Citra dan Jaringan Syaraf Tiruan Backpropagation* telah melakukan suatu sistem pengenalan plat nomor kendaraan menggunakan metode jaringan syaraf tiruan dengan program MATLAB, data berupa gambar yang mengandung plat nomor kendaraan bermotor melalui kamera gambar diambil dan dapat diambil informasinya serta dikonversikan ke dalam bentuk gambar yang lebih rinci yaitu dalam bentuk teks. Sistem yang dibuat sudah cukup baik dengan tingkat rata-rata kesalahan (*error*) pelatihan 1.907% dan 1.963% serta tingkat pengujian sebesar 88% dan 60% [10].

Berdasarkan penelitian-penelitian terdahulu, metode BPNN memiliki kemampuan untuk menghasilkan hasil klasifikasi yang sangat baik. Paper ini merupakan implementasi dari pengembangan dari teknik klasifikasi fitur berdasarkan geometri huruf[1] dengan menambahkan karakter numerik dengan studi kasus plat kendaraan. Penelitian ini bertujuan untuk mengetahui hasil pengenalan karakter yang didapatkan menggunakan metode klasifikasi BPNN dengan ekstraksi fitur gradien karakter pada plat kendaraan.

### III. METODE PENELITIAN

#### A. Gradient Feature Extraction

Gradien merupakan salah satu teknik ekstraksi fitur. Ekstraksi fitur gradien mengukur nilai magnitude dan direction dari perubahan intensitas terbesar dari setiap pixel tetangga. Pada penelitian ini, gradien akan dihitung menggunakan operator Sobel. Template Sobel digunakan untuk menghitung komponen horizontal (X) dan vertikal (Y) dari gradien seperti pada Gambar 1.

1	2	1
0	0	0
-1	-2	-1

Horizontal Component

-1	0	1
-2	0	2
-1	0	1

Vertical Component

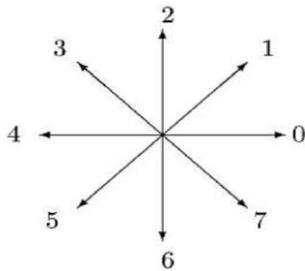
Gambar 1. Sobel Mask untuk Gradien

Misalkan terdapat citra masukan dengan ukuran  $D1 \times D2$ , setiap pixel tetangga akan dilakukan konvolusi dengan template tersebut untuk menentukan komponen X dan Y, yaitu  $S_x$  dan  $S_y$ . Persamaan (1) dan (2) merepresentasikan nilai  $S_x$  dan  $S_y$  secara berurutan. Di mana  $(i, j)$  merupakan range dari citra I dengan baris ( $D1$ )

dan kolom (D2). Gradien magnitude dan direction dapat dihitung dari gradien vektor [Sx, Sy]. Setelah mendapatkan gradien vektor dari setiap pixel, citra gradien didekomposisi menjadi 4 orientation planes atau 8 orientation planes (chain-code directions) seperti pada Gambar 2.

$$S_{(i,j)} = I_{(i-1,j+1)} + 2 \times I_{(i,j+1)} + I_{(i+1,j+1)} - I_{(i-1,j-1)} - 2 \times I_{(i,j-1)} - I_{(i+1,j-1)} \quad (1)$$

$$S_{(i,j)} = I_{(i-1,j-1)} + 2 \times I_{(i-1,j)} + I_{(i-1,j+1)} - I_{(i+1,j-1)} - 2 \times I_{(i+1,j)} - I_{(i+1,j+1)} \quad (2)$$



Gambar 2. Arah Chain-Code

### B. Neural Network

Jaringan syaraf tiruan (JST) atau juga disebut Simulated Neural Network (SNN), atau umumnya hanya disebut Neural Network (NN), adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia. NN merupakan sistem adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana, NN adalah sebuah alat permodelan data statistik non-linier. NN dapat digunakan untuk memodelkan hubungan yang kompleks antara masukan dan keluaran untuk menemukan pola-pola pada data. Menurut Ref. [11] NN dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa:

- 1) Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron);
- 2) Sinyal dikirimkan di antara neuron-neuron melalui penghubung;
- 3) Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal;

Untuk menentukan keluaran, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linear) yang dikenakan pada jumlahan masukan yang diterima. Besarnya keluaran ini selanjutnya dibandingkan dengan suatu batas ambang.

Ref. [11] juga menuliskan bahwa NN ditentukan oleh beberapa hal. Pertama, pola hubungan antar neuron (disebut arsitektur jaringan). Kedua, metode untuk menentukan bobot penghubung (disebut metode training atau belajar algoritma).

### C. Back Propagation

Pelatihan propagasi balik (*backpropagation*) meliputi tiga fase yaitu:

- **Fase I :** Propagasi maju. Selama propagasi maju, sinyal masukan (xi) dipropagasikan kelapisan tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit lapisan tersembunyi (zj) tersebut selanjutnya dipropagasikan maju lagi ke lapisan tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan (yk). Berikutnya, keluaran jaringan (yk) dibandingkan dengan goal yang harus dicapai (tk). Selisih tk – yk adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka setiap bobot dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.
- **Fase II :** Propagasi mundur. Berdasarkan kesalahan tk – yk, dihitung faktor δj (k = 1, 2, ..., m) yang dipakai untuk mendistribusikan kesalahan di unit yk ke semua unit tersembunyi yang terhubung langsung dengan yk. δk juga dipakai untuk mengubah bobot yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor δj di setiap unit di lapisan tersembunyi sebagai perubahan semua bobot yang berasal dari unit tersembunyi di lapisan di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.
- **Fase III :** Perubahan bobot. Setelah semua faktor δ dihitung, semua bobot akan dimodifikasi bersamaan. Perubahan suatu bobot didasarkan atas faktor δ neuron di lapisan atasnya. Sebagai contoh, perubahan bobot yang menuju ke lapisan keluaran didasarkan atas δk yang ada di unit keluaran. Ketiga fase tersebut diulang-ulang hingga kondisi penghentian terpenuhi. Secara umum kondisi penghentian diatur oleh jumlah iterasi dan toleransi kesalahan yang telah ditentukan. Iterasi dihentikan jika jumlah iterasi sudah sama dengan jumlah maksimum iterasi yang ditentukan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diperbolehkan.

Berikut algoritma dari JST Backpropagation:

- a) Inisialisasi bobot (ambil bobot awal dengan nilai acak yang cukup kecil), tetapkan maksimum iterasi, target eror dan laju pembelajaran.
- b) Jika iterasi < maksimum iterasi dan MSE (*Mean Square Error*) > target eror maka proses perhitungan akan terus berlanjut.
- c) Tiap-tiap unit input (Xi, i=1,2...n) menerima sinyal Xi dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

- d) Tiap unit pada suatu lapisan tersembunyi ( $Z_j$ ,  $j=0,1,2,\dots,p$ ) menjumlahkan sinyal-sinyal input terbobot menggunakan Persamaan (3).

$$Z_{net\ j} = V_{0j} + \sum_{i=1}^p X_i V_{ij} \quad (3)$$

di mana  $Z_{net}$  adalah nilai input untuk hidden layer,  $V$  adalah bobot awal dari unit input ke unit tersembunyi, dan  $X$  adalah unit input.

Menggunakan fungsi aktivasi sigmoid untuk menghitung sinyal Output menggunakan Persamaan (4).

$$Z_j = f(Z_{net\ j}) \quad (4)$$

di mana  $Z$  adalah Output sinyal dari hidden layer. Kemudian mengirim sinyal tersebut ke semua unit di lapisan atasnya (unit-unit Output).

- e) Tiap-tiap unit Output  $Y_k$  ( $k=0,1,2,\dots,m$ ) menjumlahkan sinyal-sinyal input terbobot menggunakan Persamaan (5).

$$Y_{net\ k} = W_{0j} + \sum_{k=0}^m Z_j W_{jk} \quad (5)$$

Menggunakan fungsi aktivasi untuk menghitung sinyal Output menggunakan Persamaan (6).

$$Y_k = f(Y_{net\ k}) \quad (6)$$

Mengirim sinyal tersebut ke semua unit di lapisan atasnya (unit-unit Output). Perhitungan ini dilakukan sesuai dengan jumlah unit tersembunyi, di mana  $Y_{net}$  adalah sinyal input dari hidden layer ke unit Output.  $Y$  adalah Output sistem.  $W$  adalah bobot hidden layer untuk unit Output.

- f) Tiap-tiap unit Output  $Y_k$  ( $k=0,1,2,\dots,m$ ) menerima target pola yang berhubungan dengan pola input pelatihan, kemudian menghitung informasi erornya menggunakan persamaan (7).

$$\delta_k = (t_k - Y_k) f'(Y_{net\ k}) \quad (7)$$

Kemudian menghitung suku perubahan bobot yang akan digunakan untuk mengubah bobot  $W_{jk}$  dengan laju pembelajaran  $\alpha$  menggunakan Persamaan (8).

$$\Delta w_{jk} = \alpha \delta_k Z_j \quad (8)$$

Di mana  $\delta$  adalah informasi kesalahan yang akan digunakan dalam perubahan bobot layer,  $\alpha$  adalah laju perubahan (modifikasi) bobot di setiap iterasi,  $\Delta w$  adalah suku perubahan bobot pada hidden layer dan unit Output,  $t$  adalah target Output sistem.

- g) Menghitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $Z_j$  ( $j=0,1,2,\dots,p$ ) menggunakan Persamaan (9).

$$\delta_{net\ j} = \sum_{k=0}^m \delta_k W_{jk} \quad (9)$$

Mengalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi eror menggunakan Persamaan (10).

$$\delta_j = \delta_{net\ j} f'(z_{net\ j}) \quad (10)$$

Kemudian menghitung suku perubahan bobot  $V_{ij}$  yang akan digunakan untuk mengubah nilai  $V_{ij}$  ( $j=0,1,2,\dots,p$ ;  $i=0,1,2,\dots,n$ ) menggunakan Persamaan (11).

$$\delta_j = \delta_{net\ j} f'(z_{net\ j}) \quad (11)$$

di mana  $\delta_{net\ j}$  adalah informasi kesalahan pada unit tersembunyi,  $\Delta v$  adalah suku perubahan bobot pada unit input dan unit hidden layer.

- h) Menghitung semua perubahan bobot ( $k=0,1,2,\dots,m$ ) memperbaiki bias dan bobot ( $j=0,1,2,\dots,p$ ) menggunakan Persamaan (12).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (12)$$

Tiap-tiap unit tersembunyi  $Z_j$  ( $j=0,1,2,\dots,p$ ) memperbaiki bias dan bobotnya ( $i=0,1,2,\dots,n$ ) menggunakan Persamaan (13).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (13)$$

- i) Menghitung MSE (*Mean Square Error*) menggunakan Persamaan (14).

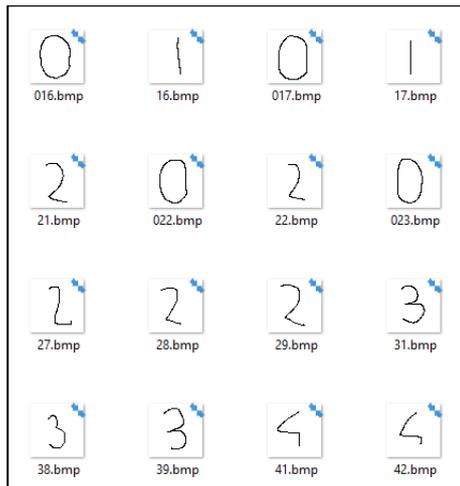
$$MSE = \sum (t - Y_k)^2 / n \quad (14)$$

di mana  $n$  merupakan jumlah dari unit input.

#### IV. HASIL DAN PEMBAHASAN

##### A. Pengumpulan Data

Langkah pertama yang dilakukan dalam penelitian ini adalah pengumpulan data latih berupa gambar yang dibuat menggunakan Microsoft Paint. Data latih dibuat dengan cara menggambar karakter menggunakan tools Pencil pada canvas dengan ukuran 50x50 pixel, ukuran 50x50 dipilih untuk mempermudah proses preprocessing dan karakter dengan ukuran 50x50 masih dapat dibaca, gambar tersebut disimpan dalam format .jpg. Data latih yang digunakan berjumlah 900 data yang terdiri dari 26 karakter alfabet dan 10 karakter numerik dan masing-masing karakter tersebut berjumlah 25 gambar. Contoh data latih dapat dilihat pada terlampir pada Gambar 3. Selanjutnya untuk pengumpulan data uji berupa citra plat kendaraan dari gambar yang diambil menggunakan kamera. Sebelum digunakan untuk pengujian, dilakukan proses cropping secara manual menggunakan Microsoft Paint untuk mendapatkan citra plat kendaraan saja. Contoh hasil cropping dapat dilihat pada terlampir pada Gambar 4.



Gambar 3. Data Latih



Gambar 4. Data Uji

## B. Pengujian

Proses pengujian dilakukan dalam 2 skenario berbeda yaitu:

- 1) Skenario pengujian pertama: Pada proses pelatihan, sampel yang digunakan sejumlah  $36 \times 25 = 900$  data sampel yang selanjutnya dibagi menjadi data latih dan data uji. Data latih yang digunakan sejumlah 80% yaitu  $36 \times 20 = 720$  data latih dan sisanya digunakan sebagai data uji sejumlah 20% yaitu  $36 \times 5 = 180$  data uji.
- 2) Skenario pengujian kedua: Proses pengujian kedua menggunakan network pada skenario pengujian pertama dengan data uji berupa plat kendaraan untuk mengetahui presisi dan kecepatan proses pengenalan citra.

### B.1. Pengujian Skenario Pertama

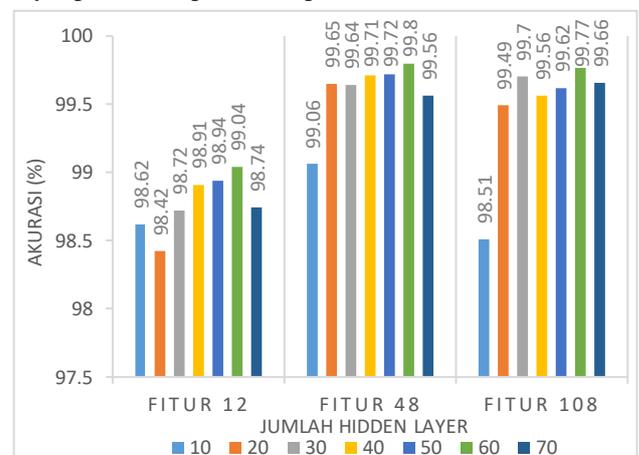
Pengujian skenario pertama dilakukan untuk mengetahui pengaruh hidden layer terhadap performa sistem, dilakukan manipulasi jumlah hidden layer di mana hidden layer yang digunakan yaitu 10, 20, 30, 40, 50, 60,

dan 70 dengan jumlah fitur masing-masing 12, 48, dan 108 fitur. Penentuan performa dapat dilihat dari beberapa parameter yaitu :

- 1) Tingkat Akurasi
- 2) Tingkat Presisi
- 3) True Postitive Rate (TPR), True Negative Rate (TNR), False Negative Rate (FNR), dan True Postive Rate (TPR).

#### B.1.1. Tingkat Akurasi

Semakin tinggi nilai akurasi dari masing-masing *hidden layer* maka nilai performa sistem juga semakin tinggi (baik), nilai akurasi didapatkan dengan menggunakan formula (3 – 4) pada section 3. Hasil persentase akurasi jumlah hidden layer per fitur dapat dilihat pada Gambar 5.



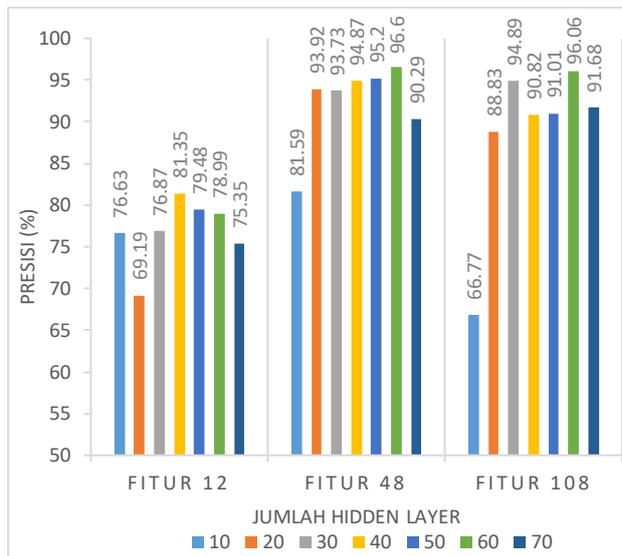
Gambar 5. Perbandingan tingkat akurasi berdasarkan jumlah fitur

Berdasarkan Gambar 5 dapat dilihat bahwa terjadi peningkatan akurasi seiring dengan bertambahnya jumlah fitur, tingkat nilai akurasi maksimum terdapat pada fitur 48 dengan jumlah hidden layer 60 dengan nilai akurasi 99,80%. Fitur 12 dengan jumlah hidden layer 20 memiliki nilai akurasi terendah yaitu 98,42%, dan pada fitur 12 tidak dilakukan zoning seperti yang dilakukan pada fitur lainnya, bertambahnya fitur maka bertambah pula zoning yang digunakan. Pada Gambar 5 penambahan jumlah hidden layer masing-masing fitur tidak mempengaruhi peningkatan nilai akurasi, di mana terjadi penurunan dan peningkatan nilai akurasi saat dilakukan penambahan jumlah *hidden layer*.

#### B.1.2. Tingkat Presisi

Penentuan nilai presisi dapat dilakukan dengan cara menghitung jumlah data true positive (TP) dibagi dengan jumlah true positive (TP) ditambah dengan jumlah true negative (TN) yang diperoleh, sehingga data persentase presisi yang diperoleh adalah seperti pada Gambar 6. Persentase tingkat presisi pada Gambar 6 menunjukkan bahwa tingkat presisi paling tinggi terdapat pada fitur 48 dengan nilai presisi yaitu 96,60% dengan jumlah hidden layer 60. Jumlah fitur mempengaruhi nilai presisi yang

didapatkan, semakin meningkat jumlah fitur, maka tingkat presisi yang didapatkan meningkat pula. Seperti terlihat pada Gambar 6 fitur 12 memiliki nilai presisi tertinggi yaitu 81,35% pada hidden layer 30, kemudian nilai presisi meningkat pada fitur 48 mencapai 96,60% pada hidden layer 60, hal ini juga menunjukkan bahwa jumlah hidden layer tidak mempengaruhi peningkatan nilai presisi.

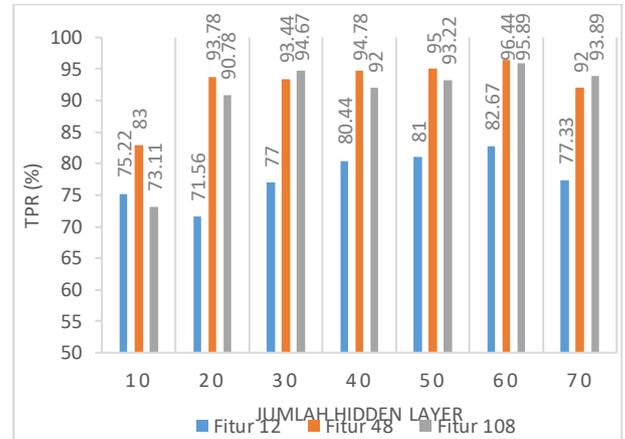


Gambar 6. Perbandingan tingkat presisi berdasarkan jumlah hidden layer

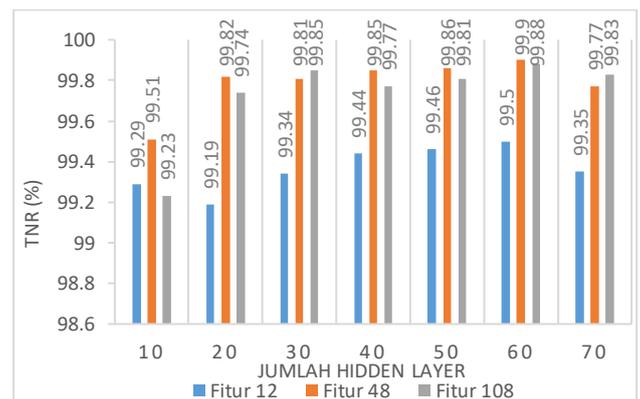
B.1.3. TPR, TNR, FNR, dan FPR

- 1) **TPR**: TPR menunjukkan kemampuan sistem dalam memprediksi hasil dengan benar, dengan kata lain TPR menunjukkan tingkat deteksi dari sistem, pada Gambar 7 peningkatan nilai TPR juga dipengaruhi dengan bertambahnya jumlah fitur yang digunakan, hal ini dapat dilihat pada data dengan jumlah hidden layer 30 dan 70 dengan nilai TPR fitur 12, 48, dan 108 yaitu 77%, 93,44%, dan 94,67%.
- 2) **TNR**: TNR menunjukkan nilai data salah yang berhasil diidentifikasi salah oleh sistem, nilai TNR dapat diukur dengan menggunakan formulasi (3 – 8) pada Bab III. Gambar 8 menampilkan data TNR dengan nilai diatas 99%, dan nilai TNR tertinggi yaitu 99,90% yang terdapat pada fitur 48 dengan jumlah hidden layer 60.
- 3) **FNR**: FNR atau disebut pula tingkat kesalahan sistem dalam memprediksi hasil dari input yang masuk ke dalam sistem, pada Gambar 9 tingkat kesalahan terendah dimiliki oleh fitur 48 pada hidden layer 60 dengan persentase 3,56%, sedangkan untuk tingkat kesalahan tertinggi dimiliki oleh fitur 12 pada hidden layer 20 dengan persentase mencapai 28,44%.
- 4) **FPR**: FPR merupakan tingkat di mana sistem membaca data yang seharusnya salah menjadi data yang benar, sehingga semakin rendah nilai FPR sistem maka semakin bagus. Gambar 10 menunjukkan bahwa fitur 48 memiliki FPR

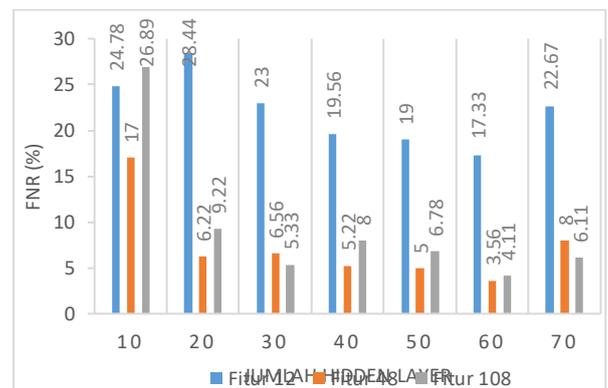
terendah yaitu 0.10% pada hidden layer 60, sedangkan FPR tertinggi terdapat pada fitur 10 dengan jumlah hidden layer 20, yang menunjukkan bahwa peningkatan jumlah fitur mempengaruhi penurunan nilai FPR.



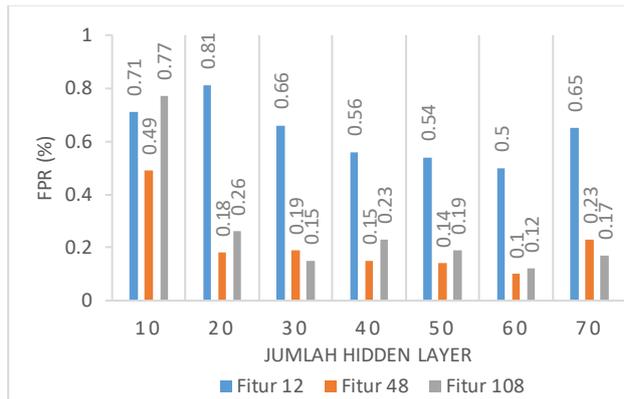
Gambar 7. Persentase TPR berdasarkan jumlah hidden layer.



Gambar 8. Persentase TNR berdasarkan jumlah hidden layer.



Gambar 9. Persentase FNR berdasarkan jumlah hidden layer.



Gambar 10. Persentase FPR berdasarkan jumlah *hidden layer*.

### B.2. Pengujian Skenario Kedua

Pengujian skenario kedua dilakukan dengan menggunakan data citra plat kendaraan, terdapat 10 citra yang akan dilakukan pengujian untuk melihat output dan tingkat presisi dengan menggunakan fitur 12, 48, dan 108 dengan jumlah hidden layer yaitu 10, 20, 30, 40, 50, 60, dan 70 *hidden layer*. Hasil pengujian masing-masing fitur ditunjukkan pada Tabel 1 dan Tabel 2. Berdasarkan data Tabel 2 yang menampilkan hasil presisi pengujian masing-masing citra, diperoleh presisi rata-rata dari masing-masing fitur sesuai dengan jumlah *hidden layer*-nya adalah seperti pada Gambar 11.

Rata-rata nilai presisi pada Gambar 11 menunjukkan bahwa peningkatan jumlah fitur mempengaruhi hasil presisi dari pengujian citra tersebut, sehingga grafik pada Gambar 4.9 nilai presisi terus meningkat sebanding dengan jumlah fitur yang digunakan pada semua jumlah hidden layer. Hal ini menunjukkan bahwa jumlah fitur yang digunakan sangat mempengaruhi performa dari metode gradien karakter, karena pada metode ini pengenalan fitur dilakukan dengan menghitung nilai gradient magnitude dan arah dari citra masukan, sehingga semakin banyak fitur yang digunakan pada citra maka semakin jelas sistem menentukan nilai gradien dan arah pada citra.

Nilai presisi maksimum yang diperoleh dari pengujian skenario kedua yaitu 58,13%, hal ini menunjukkan bahwa metode karakter gradien yang digunakan pada ekstraksi fitur plat kendaraan yang menjadi citra uji dengan cara memanipulasi jumlah hidden layer dan jumlah fitur tidak mampu menghasilkan nilai presisi yang cukup baik. Hal ini disebabkan karena pada metode ekstraksi fitur ini mengenali ciri citra dengan melihat nilai dan arah gradien, sehingga nilai pada tahap *preprocessing* yaitu *grayscale*, *morphology*, dan *binarization* dan tahap *thinning* sangat mempengaruhi proses penentuan nilai dan arah gradien pada metode karakter gradien. Hasil dari proses *thinning* pada tiap karakter yang sama akan memiliki banyak perbedaan arah gradien, padahal karakter pada plat kendaraan memiliki standar ukuran dan jenis tulisan yang sama. Sehingga untuk mendapatkan nilai presisi yang lebih baik dari metode ini dapat dilakukan dengan cara

melakukan perbaikan dengan tidak hanya menggunakan arah gradien saja, tetapi juga dengan menambahkan jumlah persimpangan (*intersects*) karakter dan jumlah lubang karakter (*number of holes*), solusi lain dapat juga dengan menggunakan metode fitur ekstraksi lain yang mendukung pengenalan citra plat kendaraan.

Pada penelitian ini fungsi training neural network yang digunakan yaitu fungsi *Levenberg-Marquardt (trainlm)*, pada fungsi ini terdapat beberapa parameter yang dapat disesuaikan dengan kebutuhan pengolahan data, adapun parameter yang digunakan yaitu:

- Jumlah maksimum jangka waktu training data (1000)
- Performance goal (0)
- Jumlah maksimum validasi yang gagal (6)
- Jumlah minimum performa gradien ( $1e-07$ )
- Nilai awal mu (0.001)
- Nilai penurunan mu (0.1)
- Nilai kenaikan mu (10)
- Nilai maksimum mu (10000000000)
- Jangka waktu setiap penampilan data (25)
- Waktu maksimal yang dibutuhkan untuk training data (Inf)

Training data menggunakan fungsi *Levenberg-Marquardt* pada penelitian ini menghasilkan nilai presisi maksimum yaitu 58,13%, sehingga untuk penelitian selanjutnya dapat dilakukan percobaan dengan data yang sama menggunakan fungsi training lainnya pada BPNN, antara lain yaitu:

- BFGS Quasi-Newton (*trainbfg*)
- Resilient Backpropagation (*trainrpr*)
- Scaled Conjugate Gradient (*trainscg*)
- Conjugate Gradient with Powell/Beale Restarts (*traincgb*)
- Fletcher-Powell Conjugate Gradient (*traincgf*)
- Polak-Ribière Conjugate Gradient (*traincgp*)
- One Step Secant (*trainoss*)
- Variable Learning Rate Backpropagation (*trainidx*)

Waktu yang dibutuhkan untuk mengelola masing-masing citra dapat dilihat pada Tabel 4.3, berdasarkan data tersebut dapat ditentukan waktu rata-rata yang dibutuhkan untuk mengolah data citra dapat dilihat pada Gambar 12. Berdasarkan data tersebut diketahui bahwa semakin meningkat jumlah fitur maka semakin banyak waktu pengolahan citra yang diperlukan, hal ini disebabkan karena dengan meningkatnya jumlah fitur maka semakin banyak jumlah data matriks yang akan dikelola, sehingga waktu pengolahan data yang diperlukan semakin banyak.

TABEL I. OUTPUT BERDASARKAN JUMLAH HIDDEN LAYER PADA MASING-MASING FITUR

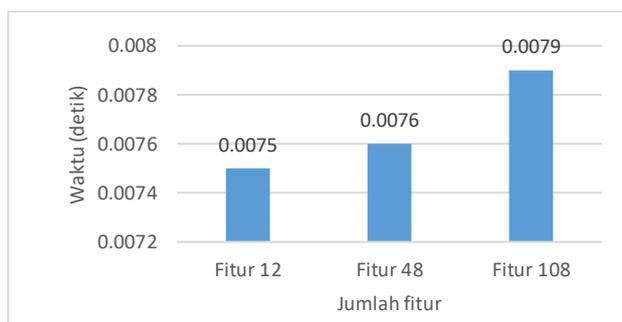
Citra	Fitur	Jumlah Hidden Layer						
		10	20	30	40	50	60	70
	12	YY4Y3YY	CC4ICYYQ	YDYYPYYX	YY4J3YJ0	JY4Y3YYX	YY4Y3YYQ	YYYY3YYY
	48	NR31S01Q	OR41GO4R	AQ4A9AAQ	4Q4Y4004	YQ4Y301Q	HR4Y100R	OR41500R
	108	DK2KVMCK	OR413OCR	NR4TBOCP	OP4D5OCR	OQ4T8OCQ	OR4T500R	OR4D500R
	12	YYO4OYY	YCY4YYY	YYY444YY	YYX3XYY	YYY34YYY	YY24XY	YYYY4YYY
	48	9R3QXVY9	OR7024YO	DRJOGJYD	OR3044YO	DRQO4QYD	DR4P2QYD	DR302QMD
	108	DRU629YD	ORFKKOYO	DRHOP9VP	DRO8P9VO	OQS6P9YO	OR56Z9YO	OR36Z0YO
	12	YYY4Y4YY	YY4IYYX	YX4YYYYV	YY34YYYV	YY34YYYV	YY4Y7YV	YY34YYYV
	48	VS31QGSV	OX31OGG7	HA7A07VV	OX340437	OY74043V	DY3104YV	DA31043V
	108	DK9FWANT	BPFOOQ3W	DX3ZO43K	DR3ZO43M	CX3ZO43M	DQ3FO43W	OQ3ZO40W
	12	YY34Y4YY	CY44J4YY	4Y44YYY	JY33J3YJ	JY34J4YY	JY44Y4YY	JY34Y4YY
	48	13S1X11	O4GOJ2GC	T497T7AA	OX44TYOO	QX4212QQ	OR52120Y	ORO2120Y
	108	DKHDQDCC	OKFZQZOC	DKH922OC	OK8ZQDOC	OK59T2OC	OK5ZT20O	OK5ZD20O
	12	YYO3YYY	CY431XY	YYY4Y4YY	JY33J3YJ	JY43V3YY	PY43Y2YY	RY43Y3YY
	48	1R7S131Q	OR2JIG03	HR7913A1	4R24T3O3	DR231QOQ	HR21Y3O3	DR2513OQ
	108	DRD53TCJ	ORZSAKCK	NRZBT3OQ	OQD5Q3OO	OQ95T3CC	ORZ5T3OO	ORZ5D3C6
	12	YYY4Y4CY	YCYIYYIY	YX4YYYY	YYXYYMY	YX3Y44Y2	YY27444Y	YYYYYY4Y
	48	1A37334V	7PY771HY	HAH744HM	4R33444X	DRQ7444W	HR87424U	JRQ7444W
	108	JNJ7CKKY	OR8144HM	OQ87O4HM	D887M4MM	DX8244HM	DR8744AM	OR87O4HM
	12	YY3YOYCY	CCCI3YIV	YY4YXYIV	CY3J3YV	JY3Y2Y2Y	YY3Y2YUV	2Y3I0Y2V
	48	1R3A90UA	ORJ1907A	DRV7901A	OR4490UA	DRQ190UA	DR5190UA	DR5100HA
	108	DRVC9DUC	ORWQO0UA	PRHT9PUA	DR8180UA	OR5T90UA	DR5T90UA	OR5O90UA
	12	YY4Y3OCC	YY4Y4YIC	YY4Y44IY	JY3Y33CC	JY4Y3322	RY4Y3074	RY4Y3024
	48	1QXQSX71	OR2GGGH4	TR4T9HHA	4R4044HP	DR2Q38HY	HR2014HI	OR205QH1
	108	MR2MSJHR	ORZ038MP	PQ90S8HP	JQZ0S8HP	OQZ0S8HP	ORZ0S8HP	ORZ0S8HP
	12	YYO044YY	YYOY44YY	YXX4YYY	YXX4YYY	2XXY44YY	2Y0044YY	2Y00YYY
	48	1AX94WYO	OAG441YD	HAAY44YO	OP4444YO	OAQY44YO	HRQ442YO	DPQO44MO
	108	DR6VLLYC	ORK04LYO	PR8OZZYO	DP88Z4YO	OQ68Z4YO	DR6944YO	OR6944YO
	12	YYO03OC	CQ304QI	YXX44XI	UX333X1	JX333XY	JQ203X4	RY303XI
	48	1RQQ3XI	ORQGGOI	TRQAJFL	ORQ434I	DRQQ38F	HRQQ38L	DRQQ50U
	108	DR6AVJL	OAKK3KI	PR8ON8L	OP8888L	OQ6658L	DR6658L	DR6658L

TABEL II. OUTPUT PRESISI BERDASARKAN JUMLAH HIDDEN LAYER PADA MASING-MASING FITUR

Citra	Fitur	Jumlah Hidden Layer						
		10	20	30	40	50	60	70
	12	12.50%	12.50%	0.00%	12.50%	12.50%	12.50%	0.00%
	48	31.25%	43.75%	12.50%	3.13%	12.50%	37.50%	62.50%
	108	25.00%	62.50%	50.00%	62.50%	25.00%	50.00%	50.00%
	12	2.50%	2.08%	2.50%	2.50%	2.08%	2.50%	1.79%
	48	37.50%	37.50%	50.00%	37.50%	50.00%	62.50%	62.50%
	108	87.50%	25.00%	50.00%	50.00%	37.50%	50.00%	50.00%
	12	6.25%	0.00%	0.00%	12.50%	12.50%	0.00%	12.50%
	48	18.75%	12.50%	0.00%	12.50%	6.25%	37.50%	31.25%
	108	12.50%	0.00%	31.25%	56.25%	18.75%	31.25%	25.00%
	12	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	48	3.13%	37.50%	0.00%	0.00%	37.50%	37.50%	37.50%
	108	22.92%	25.00%	50.00%	25.00%	62.50%	25.00%	25.00%
	12	0.00%	12.50%	0.00%	4.17%	6.25%	0.00%	6.25%
	48	29.17%	25.00%	31.25%	31.25%	50.00%	31.25%	75.00%
	108	31.25%	25.00%	25.00%	25.00%	31.25%	37.50%	50.00%
	12	12.50%	0.00%	0.00%	0.00%	25.00%	29.17%	0.00%
	48	12.50%	16.67%	54.17%	25.00%	54.17%	37.50%	41.67%
	108	12.50%	62.50%	62.50%	54.17%	62.50%	75.00%	75.00%
	12	0.00%	0.00%	0.00%	0.00%	0.00%	12.50%	0.00%
	48	43.75%	50.00%	50.00%	50.00%	75.00%	87.50%	62.50%
	108	43.75%	37.50%	50.00%	62.50%	62.50%	75.00%	62.50%
	12	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	48	0.00%	37.50%	18.75%	37.50%	62.50%	31.25%	50.00%
	108	31.25%	37.50%	31.25%	50.00%	37.50%	50.00%	50.00%
	12	28.13%	27.50%	2.50%	27.50%	29.17%	29.17%	2.50%
	48	50.00%	25.00%	31.25%	25.00%	31.25%	37.50%	37.50%
	108	62.50%	50.00%	25.00%	50.00%	50.00%	87.50%	75.00%
	12	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	48	14.29%	14.29%	28.57%	14.29%	42.86%	42.86%	42.86%
	108	71.43%	0.00%	35.71%	17.86%	57.14%	100.00%	85.71%



Gambar 11. Nilai rata-rata presisi fitur berdasarkan jumlah hidden layer.



Gambar 12. Waktu rata-rata pengolahan citra per fitur

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan penelitian pada bagian sebelumnya, maka dapat disimpulkan bahwa:

1. Penelitian dilakukan terhadap beberapa fitur yaitu 12, 48, dan 108 dengan jumlah hidden layer 10, 20, 30, 40, 50, 60, dan 70 hidden layer.
2. Pada skenario pengujian pertama diperoleh performa terbaik yaitu fitur 48 dengan jumlah hidden layer 60. Nilai akurasi, presisi, TPR, TNR, FNR, dan FPR yang didapatkan pada fitur ini yaitu masing-masing 99,80%, 96,60%, 96,44%, 99,90%, 3,56%, dan 0,10%.
3. Performa terendah pada pengujian skenario pertama terdapat pada fitur 12 dengan jumlah hidden layer 20. Nilai akurasi, presisi, TPR, TNR, FNR, dan FPR yang diperoleh dari pengujian yaitu masing-masing 98,42%, 69,19%, 71,56%, 99,19%, 28,44%, dan 0,81%.
4. Skenario kedua diperoleh fitur terbaik yaitu pada fitur 108 dengan nilai rata-rata presisi mencapai 58,13% dengan jumlah hidden layer 60.
5. Jumlah fitur mempengaruhi tingkat performa sistem dikur, semakin tinggi jumlah fitur maka performa sistem semakin meningkat.
6. Nilai presisi yang didapatkan pada pengujian skenario kedua menggunakan citra plat kendaraan menunjukkan bahwa metode gradient feature

extraction hanya mampu menghasilkan nilai presisi tertinggi di bawah 60% yang menandakan bahwa teknik yang digunakan belum mampu menghasilkan output yang maksimal.

### B. Saran

Jika dilakukan penelitian lebih lanjut pada kasus ini dapat mempertimbangkan saran-saran dan perubahan sebagai berikut:

1. Metode preprocessing data lainnya yang mampu menghasilkan data yang lebih baik untuk dijadikan sebagai data input.
2. Metode ekstraksi fitur lainnya yang mampu memberikan fitur yang lebih baik dari metode Gradient Feature Extraction, salah satunya yaitu menggunakan metode Template Matching.
3. Pengujian untuk menentukan performa sistem dapat ditambahkan sehingga tingkat performa sistem akan lebih terukur.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu/terlibat dalam penelitian ini, antara lain Rival Biasrori yang telah membantu dalam pengumpulan data uji, dan semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan bimbingan kepada penulis dalam menyelesaikan Penelitian ini.

## DAFTAR PUSTAKA

- [1] D. D. Gaurav and R. Ramesh, "A feature extraction technique based on character geometry for character recognition," *arXiv Prepr. arXiv1202.3884*, 2012.
- [2] S. Wang, "A review of gradient-based and edge-based feature extraction methods for object detection," in *2011 IEEE 11th International Conference on Computer and Information Technology*, 2011, pp. 277–282.
- [3] I. G. P. S. Wijaya, A. Y. Husodo, and A. H. Jatmika, "Real time face recognition engine using compact features for electronics key," in *Proceeding - 2016 International Seminar on Intelligent Technology and Its Application, ISITIA 2016: Recent Trends in Intelligent Computational Technologies for Sustainable Energy*, 2017, doi: 10.1109/ISITIA.2016.7828650.
- [4] A. Y. Husodo and I. G. P. S. Wijaya, "Porn picture files scanning method based on the number of picture files in a directory," in *Proceedings of 2015 International Conference on Information and Communication Technology and Systems, ICTS 2015*, 2016, doi: 10.1109/ICTS.2015.7379866.
- [5] I. G. P. S. Wijaya, A. Y. Husodo, and I. W. A. Arimbawa, "Real time face recognition based on face descriptor and its application," *Telkonnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 2, 2018, doi: 10.12928/telkonnika.v16.i2.7418.
- [6] I. G. P. S. Wijaya, I. B. K. Widiartha, K. Uchimura, M. S. Iqbal, and A. Y. Husodo, "Fast pornographic image recognition using compact holistic features and multi-layer

- neural network,” *Int. J. Adv. Intell. Informatics*, vol. 5, no. 2, pp. 89–100, 2019.
- [7] K. Prasad, D. C. Nigam, A. Lakhotiya, and D. Umre, “Character recognition using matlab’s neural network toolbox,” *Int. J. u-and e-Service, Sci. Technol.*, vol. 6, no. 1, pp. 13–20, 2013.
- [8] M. Hanmandlu, K. R. M. Mohan, and H. Kumar, “Neural based handwritten character recognition,” in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No. PR00318)*, 1999, pp. 241–244.
- [9] A. S. Widodo, “Sistem Deteksi Dan Pengenalan Karakter pada Plat Nomor Kendaraan Dengan Metode Backpropagation,” Yogyakarta., 2014.
- [10] M. Syuhada, “Realisasi pengenalan plat nomor kendaraan dengan metode histogram citra dan jaringan syaraf tiruan backpropagation,” 2015.
- [11] S. J. Jek, “Jaringan Syaraf Tiruan dan Pemrograman Menggunakan Matlab.” Andi Offset, Yogyakarta, Cetakan pertama, 2005.