

**PENGENALAN PLAT KENDARAAN BERMOTOR MENGGUNAKAN  
METODE GRADIEN KARAKTER DAN BPNN  
(BACKPROPAGATION NEURAL NETWORK)**

Tugas Akhir

Untuk memenuhi sebagian persyaratan

Mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh:

**Muhammad Ridwan Ali**

**F1D 013 064**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MATARAM  
2020**

**TUGAS AKHIR**  
**PENGENALAN PLAT KENDARAAN BERMOTOR MENGGUNAKAN**  
**METODE GRADIEN KARAKTER DAN BPNN**  
**(BACKPROPAGATION NEURAL NETWORK)**

Oleh:

**Muhammad Ridwan Ali**

**F1D 013 064**

Telah diperiksa dan disetujui oleh:

1. Pembimbing Utama



Prof. Dr. Eng I Gede Pasek Suta Wijaya, S.T., M.T.

Tanggal: 28 Februari 2020

NIP: 19731130 200003 1 001

2. Pembimbing Pendamping



Ario Yudo Husodo, S.T., M.T.

Tanggal: 28 Februari 2020

NIP: 19901218 201212 1 002

Mengetahui

Ketua Program Studi Teknik Informatika

Fakultas Teknik

Universitas Mataram



Prof. Dr. Eng I Gede Pasek Suta Wijaya, S.T., M.T.

NIP: 19731130 200003 1 001

## TUGAS AKHIR

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


Oleh:

**Muhammad Ridwan Ali**  
F1D 013 064

Telah dipertahankan di depan dewan Penguji  
Pada tanggal 10 Februari 2020  
dan dinyatakan telah memenuhi syarat mencapai derajat sarjana S-1  
Program Studi Teknik Informatika


Susunan Tim Penguji

1. Penguji 1

  
Fitri Bimantoro, ST., M.Kom.  
NIP: 19860622 201504 1 002

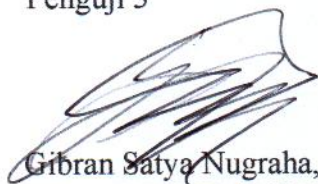
Tanggal: 28 Februari 2020

2. Penguji 2

  
Royana Afwani, ST., MT.  
NIP: 19850707 201404 2 001

Tanggal: 28 Februari 2020

3. Penguji 3

  
Gibran Satya Nugraha, S.Kom., M.Eng.  
NIP: 19920323 201903 1 012

Tanggal: 28 Februari 2020

Mataram, 28 Februari 2020

Dekan Fakultas Teknik  
Universitas Mataram



Akmaluddin, ST, M.Sc, (Eng), Ph.D.  
NIP: 19681231 199412 1 001

## **HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Saya yang bertanda tangan di bawah ini menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Mataram,    Februari 2020

Muhammad Ridwan Ali

## PRAKATA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat bimbingan, dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir dengan judul “*Pengenalan Plat Kendaraan Bermotor Menggunakan Metode Gradien Karakter dan BPNN (Backpropagation Neural Network)*”.

Tugas Akhir ini dilaksanakan di Fakultas Teknik Universitas Mataram. Tujuan dari pembuatan Tugas Akhir ini adalah untuk melakukan pengenalan plat kendaraan menggunakan metode gradien karakter di mana pengenalan citra dilakukan dengan menghitung nilai dan arah *magnitude* dari citra uji, kemudian dilakukan pengklasifikasian data menggunakan metode BPNN (*Backpropagation Neural Network*).

Akhir kata semoga tidaklah terlampau berlebihan, bila penulis berharap agar karya ini dapat bermanfaat bagi pembaca.

Mataram,            Februari 2020

Penulis

## UCAPAN TERIMA KASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materiil dari berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada :

1. Bapak Mujitahid, Ibu Nurhidayah, Saudara Hidayaturrehman Ali, Saudara Yazid Bustami Ali, Saudari Nasya Hikmatun Nisa, serta keluarga besar selaku pemberi dukungan utama yang selalu memberikan doa dan dukungan baik moral maupun materiil yang tidak putus-putus kepada penulis sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik.
2. Prof. Dr.Eng. I Gede Pasek Suta Wijaya, ST., MT, selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama menyusun Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
3. Bapak Ario Yudo Husodo, ST., MT, selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan selama menyusun Tugas Akhir ini.
4. Bapak Fitri Bimantoro, ST.,M.Kom, selaku dosen penguji 1 yang telah memberikan kritik dan saran yang bersifat membangun dalam menyelesaikan Tugas Akhir ini.
5. Ibu Royana Afwani, ST., MT, selaku dosen penguji 2 yang telah memberikan kritik dan saran yang bersifat membangun dalam menyelesaikan Tugas Akhir ini.
6. Bapak Gibran Satya Nugraha, S.Kom., M.Eng, selaku dosen penguji 3 yang telah memberikan kritik dan saran yang bersifat membangun dalam menyelesaikan Tugas Akhir ini.
7. Riska Anwar, Rival Biasrori, serta bapak/ibu dan teman-teman dari Program Studi Teknik Informatika, yang telah membantu memberikan bimbingan dan dukungan dalam penyelesaian Tugas Akhir ini. Dan semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini.

Jazaakumullahu khairan, semoga Allah SWT memberikan balasan kebaikan atas bantuan yang diberikan kepada penulis.

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	iv
<b>LEMBAR PENGESAHAN</b> .....	iv
<b>HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	iv
<b>PRAKATA</b> .....	v
<b>UCAPAN TERIMA KASIH</b> .....	vi
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	ix
<b>DAFTAR TABEL</b> .....	xi
<b>INTISARI</b> .....	xii
<b>ABSTRACT</b> .....	xiii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Manfaat .....	2
1.6 Sistematika Penulisan Laporan .....	3
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI</b>	
2.1 Tinjauan Pustaka .....	4
2.2 Dasar Teori .....	6
2.2.1 Citra .....	6
2.2.2 Pengolahan Citra .....	8
2.2.3 Pengenalan Pola .....	8
2.2.4 Gradient Feature Extraction .....	9
2.2.5 Neural Network .....	10
2.2.6 Backpropagation .....	11
2.2.7 MATLAB .....	14
<b>BAB III METODE PENELITIAN</b>	
3.1 Alat dan Bahan .....	16
3.1.1 Alat .....	16
3.1.2 Bahan .....	16

3.2	Rencana Penelitian .....	16
3.3	Teknik Pengujian.....	33
3.4	Jadwal Penelitian.....	35
<b>BAB IV HASIL DAN PEMBAHASAN</b>		
4.1	Pengumpulan Data.....	36
4.2	Pengujian .....	37
4.2.1	Pengujian Skenario Pertama .....	38
4.2.1.1	Tingkat Akurasi.....	38
4.2.1.2	Tingkat Presisi.....	39
4.2.1.3	<i>True Postitive Rate (TPR), True Negative Rate (TNR), False Negative Rate (FNR), dan False Postive Rate (TPR)</i> .....	40
4.2.2	Pengujian Skenario Kedua .....	43
<b>BAB V PENUTUP</b>		
5.1	Kesimpulan.....	52
5.2	Saran.....	53
<b>DAFTAR PUSTAKA</b>		



## DAFTAR GAMBAR

Gambar 2.1 Koordinat citra digital .....	6
Gambar 2.2 Ilustrasi digitalisasi citra.....	7
Gambar 2.3 Contoh citra <i>grayscale</i> .....	7
Gambar 2.4 Proses pengolahan citra.....	8
Gambar 2.5 Sobel masks untuk gradien.....	9
Gambar 2.6 8 arah chain-code .....	10
Gambar 3.1 Diagram alur proses pembuatan sistem.....	17
Gambar 3.2 Citra pelatihan dengan ukuran 50x50 pixel.....	18
Gambar 3.3 Blok diagram sistem.....	19
Gambar 3.4 Citra matriks <i>grayscale</i> alpabet A.....	20
Gambar 3.5 Citra matriks biner alpabet A .....	20
Gambar 3.6 Arah ekstraksi fitur .....	21
Gambar 3.7 Matriks gradien horizontal ( $G_x$ ) .....	22
Gambar 3.8 Matriks gradien vertikal ( $G_y$ ).....	22
Gambar 3.9 Matriks gradien <i>magnitude</i> ( $G$ ).....	23
Gambar 3.10 Matriks arah sudut gradien.....	23
Gambar 3.11 Matriks normalisasi arah sudut gradien .....	24
Gambar 3.12 Contoh hasil <i>single frame</i> dari video data <i>testing</i> .....	29
Gambar 3.13 Hasil <i>cropping</i> data <i>testing</i> .....	30
Gambar 3.14 Citra <i>input</i> .....	30
Gambar 3.15 Citra <i>grayscale</i> .....	30
Gambar 3.16 Citra biner.....	31
Gambar 3.17 Citra setelah <i>noise filtering</i> .....	31
Gambar 3.18 Citra segmentasi karakter .....	31
Gambar 3.19 Contoh segmentasi karakter 'R' .....	32
Gambar 3.20 Citra hasil <i>resizing</i> 50x50 pixel.....	32
Gambar 3.21 Citra hasil <i>thinning</i> .....	33
Gambar 4.1 Data Latih.....	36
Gambar 4.2 Data Uji .....	37
Gambar 4.3 Perbandingan tingkat akurasi berdasarkan jumlah fitur.....	38

Gambar 4.4 Persentase tingkat presisi berdasarkan jumlah <i>hidden layer</i> .....	39
Gambar 4.5 Persentase TPR berdasarkan jumlah <i>hidden layer</i> .....	40
Gambar 4.6 Persentase TNR berdasarkan jumlah <i>hidden layer</i> .....	41
Gambar 4.7 Persentase FNR berdasarkan jumlah <i>hidden layer</i> .....	41
Gambar 4.8 Persentase FPR berdasarkan jumlah <i>hidden layer</i> .....	42
Gambar 4.9 Nilai rata-rata presisi fitur berdasarkan jumlah <i>hidden layer</i> .....	49
Gambar 4.10 Waktu rata-rata pengolahan citra per fitur .....	51

## DAFTAR TABEL

Tabel 3.1 Nilai bobot $V_{ij}$ secara acak.....	25
Tabel 3.2 Nilai bobot $W_{ij}$ secara acak.....	25
Tabel 3.3 Nilai suku perubahan bobot .....	27
Tabel 3.4 Nilai bobot pada unit tersembunyi yang baru .....	27
Tabel 3.5 Nilai bobot $V_{ij}$ hasil 1 kali pelatihan .....	27
Tabel 3.6 Nilai bobot $W_{ij}$ hasil 1 kali pelatihan .....	28
Tabel 3.7 <i>Confusion matrix</i> untuk teknik pengujian.....	34
Tabel 3.8 Estimasi jadwal penelitian .....	35
Tabel 4.1 <i>Output</i> berdasarkan jumlah <i>hidden layer</i> pada masing-masing fitur .....	43
Tabel 4.2 Hasil Presisi berdasarkan jumlah <i>hidden layer</i> masing-masing fitur.....	45

## INTISARI

Plat kendaraan adalah ciri yang unik yang digunakan untuk mengidentifikasi sebuah kendaraan dalam bentuk kombinasi huruf dan angka. Ekstraksi fitur diperlukan untuk mengidentifikasi huruf dan angka pada citra digital. Terdapat beberapa metode pada ekstraksi fitur salah satunya yaitu metode gradien karakter. Pada penelitian ini, program aplikasi yang digunakan untuk mengidentifikasi plat kendaraan adalah metode gradien karakter dan BPNN (*backpropagation neural network*). Langkah pertama, pada citra digital dilakukan pemotongan untuk mendapatkan plat kendaraan, kemudian dilakukan segmentasi untuk mendapatkan masing-masing karakter. Langkah selanjutnya yaitu ekstraksi fitur menggunakan metode gradien karakter untuk mendapatkan ciri dari masing-masing karakter. BPNN digunakan untuk klasifikasi. Pada penelitian ini dilakukan dua jenis pengujian yaitu analisis performa berdasarkan jumlah hidden layer dan jumlah fitur pada data uji latih dan data uji. Berdasarkan hasil pengujian, dapat disimpulkan bahwa jumlah fitur mempengaruhi performa sistem. Performa tertinggi pada skenario pertama yaitu pada fitur 48 dengan jumlah hidden layer 60, dan pada skenario kedua nilai tertinggi terdapat pada fitur 108 dengan jumlah hidden layer 60. Nilai performa terendah terdapat pada fitur 12 dengan jumlah hidden layer 20.

**Keywords:** Plat kendaraan, metode gradien karakter, backpropagation neural network, feature, fitur, hidden layer.

## 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 consist 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.

**Keywords:** License plate, character gradient method, backpropagation neural network, feature, hidden layer.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

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 yang citra karakter lain. Dengan pengenalan pola dilakukan klasifikasi dari hasil ekstraksi ciri dengan menggunakan metode *Backpropagation Neural Network* (BPNN).

Berdasarkan penelitian Dileep Gaurav [1], penelitian Dileep Gaurav [1] dilakukan pada karakter alpabet dari tulisan tangan, penelitian yang diajukan dalam skripsi ini merupakan implementasi dari penelitian Dileep Gaurav [1] 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 penelitian Wang [2] pada plat kendaraan.

## **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, rumusan masalah yang dapat dibuat berhubungan dengan penelitian ini antara lain:

1. Bagaimana mengubah data citra plat kendaraan bermotor menjadi teks menggunakan metode *character gradient* dan BPNN?
2. Bagaimana menganalisis hasil pengenalan plat kendaraan bermotor yang telah didapat untuk melihat tingkat akurasi dan kecepatan proses pengolahan data?

## **1.3 Batasan Masalah**

Dalam penelitian ini terdapat pembatasan terhadap masalah yang akan dibahas, yaitu jenis tulisan (*font*) pada plat kendaraan bermotor yang digunakan untuk penelitian hanya jenis ukuran dan warna tulisan standar yang digunakan pada nomor kendaraan bermotor di Indonesia.

## **1.4 Tujuan**

Tujuan dari penelitian ini yaitu:

1. Untuk membuat aplikasi pengenalan plat kendaraan bermotor menggunakan metode *character gradient* dan BPNN.
2. Untuk mengetahui potensi metode pengolahan citra yang diusulkan dan metode pengenalan pola yang diusulkan dalam pemecahan masalah pengenalan plat kendaraan bermotor.

## **1.5 Manfaat**

Hasil dari penelitian ini diharapkan dapat memberikan manfaat dalam memperbaiki serta meningkatkan efektivitas kinerja dari sistem pengenalan plat kendaraan bermotor dengan mengoptimalkan waktu pengenalan dan mengurangi kesalahan pencatatan akibat *human error* serta mengatasi lamanya proses pencatatan secara manual.

## **1.6 Sistematika Penulisan Laporan**

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

### **1. Bab I Pendahuluan**

Bab ini mengemukakan latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi serta sistematika penulisan yang digunakan untuk menyusun laporan tugas akhir.

### **2. Bab II Tinjauan Pustaka dan Dasar Teori**

Bab ini berisi dasar teori yang digunakan dalam melakukan analisis, perancangan dan implementasi tugas akhir yang dilakukan pada bab-bab selanjutnya.

### **3. Bab III Metode Penelitian**

Bab ini berisi tentang pelaksanaan, alat, bahan, jalannya perencanaan dan hasil yang diharapkan.

### **4. Bab IV Hasil dan Pembahasan**

Bab ini memuat tentang analisa dan pembahasan hasil penelitian yang telah dilakukan.

### **5. Bab V Kesimpulan dan saran**

Bab ini berisi kesimpulan dan saran-saran berdasarkan hasil pembahasan yang diperoleh.



## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Penelitian terdahulu yang terkait dengan penelitian ini antara lain; penelitian yang berjudul *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 yang berjudul *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 [3].

Penelitian yang berjudul *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 [4].

Penelitian yang berjudul Sistem Deteksi dan Pengenalan Karakter pada Plat Nomor Kendaraan Dengan Metode Backpropagation. Dalam penelitian ini 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 [5].

Penelitian yang berjudul Realisasi Pengenalan Plat Nomor Kendaraan Dengan Metode Histogram Citra dan Jaringan Syaraf Tiruan Backpropagation. Dalam penelitian ini, dilakukan 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% [6].

Berdasarkan penelitian-penelitian terdahulu, metode BPNN memiliki kemampuan untuk menghasilkan hasil klasifikasi yang sangat baik. Berdasarkan penelitian Dileep Gaurav [1], penelitian Dileep Gaurav [1] tersebut dilakukan pada karakter alfabet dari tulisan tangan, selanjutnya pada penelitian yang diajukan

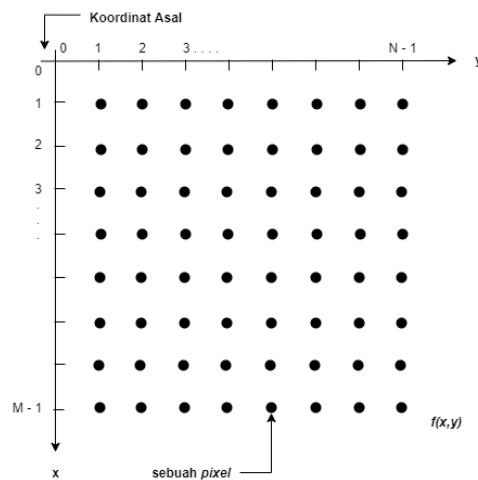
dalam skripsi ini merupakan implementasi dari penelitian Dileep Gaurav [1] tersebut dengan menambahkan karakter numerik dengan studi kasus plat kendaraan. Penelitian ini bertujuan untuk mengetahui hasil pengenalan karakter yang didapatkan menggunakan metode klasifikasi BPNN dengan ekstraksi fitur gradien karakter berdasarkan penelitian Wang [2] pada plat kendaraan.

## 2.2 Dasar Teori

Teori-teori yang dijadikan dasar pada penelitian ini yaitu tentang konsep-konsep yang perlu diketahui dan akan digunakan dalam perancangan dan pembuatan aplikasi pengenalan plat kendaraan bermotor. Adapun yang akan dibahas yaitu mulai dari konsep terkait citra, proses pengolahan citra, pengenalan pola yang akan digunakan untuk mengklasifikasikan atau menggambarkan citra berdasarkan fitur (ciri) yang dimiliki citra, kemudian metode ekstraksi fitur yang digunakan yaitu *gradient feature extraction* dan *backpropagation neural network* (BPNN), dan *tools* yang akan digunakan dalam membuat aplikasi yaitu MATLAB.

### 2.2.1 Citra

Menurut Putra [7], suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada citra tersebut. Apabila nilai x, y, dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar 2.1 menunjukkan posisi koordinat citra digital.

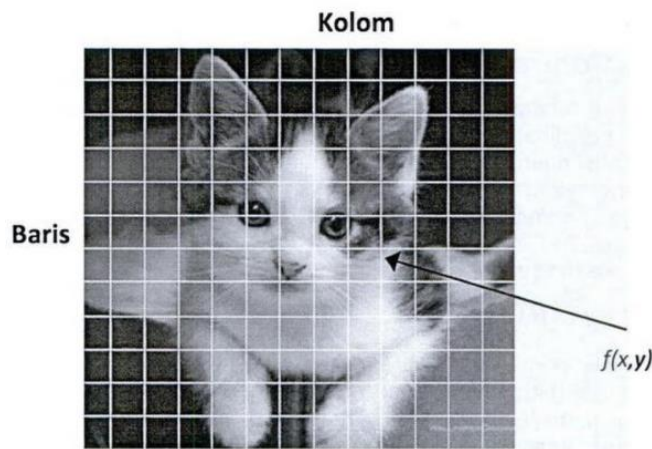


Gambar 2.1 Koordinat citra digital.

Citra digital dapat ditulis dalam matriks seperti Persamaan (2-1) berikut.

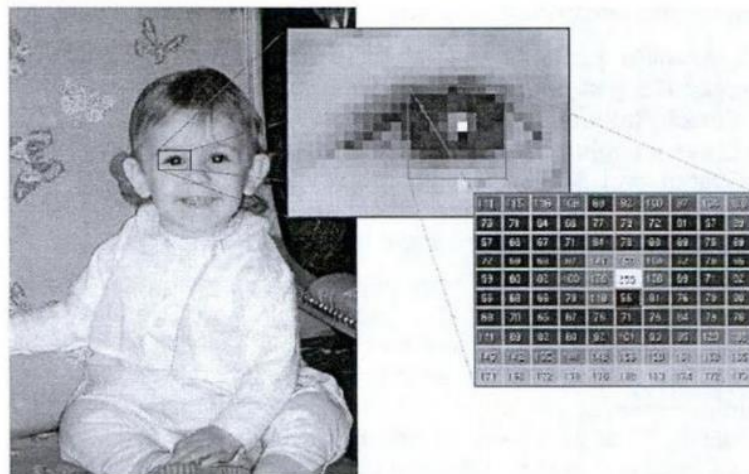
$$(x,y)= \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2-1)$$

Nilai pada suatu irisan antara baris dan kolom (pada posisi  $x,y$ ) disebut sebagai *picture elements*, *image elements*, *pels*, atau *pixels*. Istilah terakhir (*pixel*) paling sering digunakan pada citra digital. Gambar 2.2 menunjukkan ilustrasi digitalisasi citra dengan  $M = 16$  dan baris  $N = 16$  kolom.



Gambar 2.2 Ilustrasi digitalisasi citra (pixel pada koordinat  $x = 10$ ,  $y = 3$  memiliki nilai 110).

Gambar 2.3 menyajikan contoh lain dari suatu citra digital (citra *grayscale*), dengan nilai intensitas pada citra pada area tertentu.

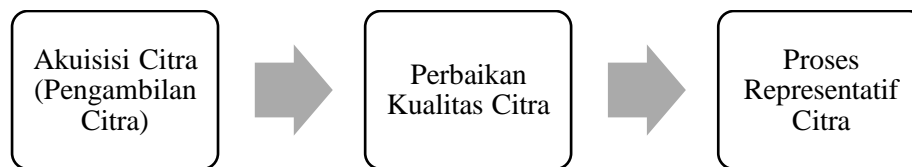


Gambar 2.3 Contoh citra *grayscale*, cuplikan (*cropping*) pada area tertentu beserta nilai intensitasnya.

### 2.2.2 Pengolahan Citra

Pengolahan citra (*image processing*) merupakan proses mengolah *pixel* di dalam citra digital untuk tujuan tertentu. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra [8].

Proses pengolahan citra secara diagram proses dimulai dari pengambilan citra, perbaikan kualitas citra, sampai dengan pernyataan representatif citra yang dicitrakan sebagai berikut:



Gambar 2.4 Proses pengolahan citra.

Dalam perkembangan lebih lanjut, *image processing* dan *computer vision* digunakan sebagai mata manusia, dengan perangkat *input image capture* seperti kamera dan *scanner* dijadikan sebagai mata dan mesin komputer (dengan program komputasinya) dijadikan sebagai otak yang mengolah informasi. Sehingga muncul beberapa pecahan bidang yang menjadi penting dalam *computer vision*, antara lain: *pattern recognition* (pengenalan pola), *biometric* (pengenalan identifikasi manusia berdasarkan ciri-ciri biologis yang tampak pada badan manusia), *content based image and video retrieval* (mendapatkan kembali citra atau video dengan informasi tertentu), *video editing*, dan lain-lain [9].

### 2.2.3 Pengenalan Pola

Perkembangan teknologi baru akan didominasi oleh sistem dan mesin-mesin dengan kecerdasan buatan (*machine intelligence*). Teknik pengenalan pola merupakan salah satu komponen penting dari mesin atau sistem cerdas tersebut yang digunakan baik untuk mengolah data maupun dalam pengambilan keputusan.

Secara umum pengenalan pola (*pattern recognition*) adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu obyek. Pola sendiri adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Sidik jari adalah suatu contoh pola. Pola bisa merupakan kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor atau matriks [7].

Pengenalan pola adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu obyek. Pengenalan pola merupakan teknik yang bertujuan untuk mengklasifikasikan citra yang telah diolah sebelumnya berdasarkan kesamaan atau kemiripan ciri yang dimilikinya. Bagian terpenting dari teknik pengenalan pola adalah bagaimana memperoleh informasi atau ciri penting yang terdapat dalam sinyal [10].

#### 2.2.4 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 2.5.

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

Horizontal Component

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

Vertical Component

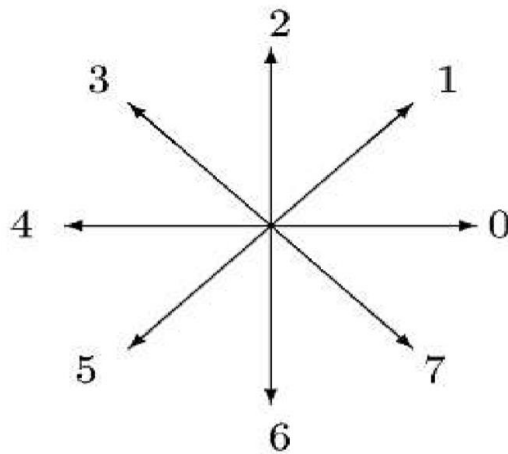
Gambar 2.5 Sobel masks untuk gradien.

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 (2-2) dan (2-3) merepresentasikan nilai  $S_x$  dan  $S_y$  secara berurutan.

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

$$S_{(i,j)} = I_{(i-1,j-1)} + 2 \times I_{(i-1,j)} + I_{(i-1,j+1)} - I_{(i+1,j-1)} - 2 \times I_{(i+1,j)} - I_{(i+1,j+1)} \quad (2-3)$$

Di mana  $(i, j)$  merupakan *range* dari citra  $I$  dengan baris ( $D1$ ) dan kolom ( $D2$ ). Gradien *magnitude* dan *direction* dapat dihitung dari gradien vektor  $[S_x, S_y]$ . Setelah mendapatkan gradien vektor dari setiap pixel, citra gradien didekomposisi menjadi 4 *orientation planes* atau 8 *orientation planes* (chain-code directions) seperti pada Gambar 2.6.



Gambar 2.6 8 arah chain-code.

### 2.2.5 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 Jong Jek Siang [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;

4. 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.

Dalam bukunya, Jong Jek Siang [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).

### 2.2.6 Backpropagation

Pelatihan propagasi balik (backpropagation) meliputi tiga fase yaitu:

1. Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan ( $x_i$ ) dipropagasikan ke lapisan tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit lapisan tersembunyi ( $z_j$ ) tersebut selanjutnya dipropagasikan maju lagi ke lapisan tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ( $y_k$ ). Berikutnya, keluaran jaringan ( $y_k$ ) dibandingkan dengan *goal* yang harus dicapai ( $t_k$ ). Selisih  $t_k - y_k$  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.

2. Fase II : Propagasi mundur

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_j$  ( $k = 1, 2, \dots, m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor  $\delta_j$  di setiap unit di lapisan tersembunyi sebagai dasar perubahan semua bobot yang berasal dari unit tersembunyi di lapisan di bawahnya. Demikian seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.



### 3. Fase III : Perubahan bobot

Setelah semua faktor  $\delta$  dihitung, semua bobot akan dimodifikasi bersamaan. Perubahan suatu bobot didasarkan atas faktor  $\delta$  neuron di lapisan atasnya. Sebagai contoh, perubahan bobot yang menuju ke lapisan keluaran didasarkan atas  $\delta_k$  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:

1. Inisialisasi bobot (ambil bobot awal dengan nilai acak yang cukup kecil), tetapkan maksimum iterasi, target eror dan laju pembelajaran.
2. Jika iterasi < maksimum iterasi dan MSE (*Mean Square Error*) > target eror maka proses perhitungan akan terus berlanjut.
3. Tiap-tiap unit *input* ( $X_i, i=1,2,\dots,n$ ) menerima sinyal  $X_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
4. Tiap unit pada suatu lapisan tersembunyi ( $Z_j, j=0,1,2,\dots,p$ ) menjumlahkan sinyal-sinyal *input* terbobot menggunakan Persamaan (2-4).

$$Z_{netj} = V_{0j} + \sum_{i=1}^p X_i V_{ij} \quad (2-4)$$

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 (2-5).

$$Z_j = f(Z_{netj}) \quad (2-5)$$

Di mana  $Z$  adalah *Output* sinyal dari *hidden layer*.

Kemudian mengirim sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *Output*).

5. Tiap-tiap unit *Output*  $Y_k$  ( $k=0,1,2,\dots,m$ ) menjumlahkan sinyal-sinyal *input* terbobot menggunakan Persamaan (2-6).

$$Y_{\_net\ k} = W_{0j} + \sum_{k=0}^m Z_j W_{jk} \quad (2-6)$$

Menggunakan fungsi aktivasi untuk menghitung sinyal *Output* menggunakan Persamaan (2-7).

$$Y_k = f(Y_{\_net\ k}) \quad (2-7)$$

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*.

6. 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 (2-8).

$$\delta_k = (t_k - Y_k) f'(Y_{\_net\ k}) \quad (2-8)$$

Kemudian menghitung suku perubahan bobot yang akan digunakan untuk mengubah bobot  $W_{jk}$  dengan laju pembelajaran  $\alpha$  menggunakan Persamaan (2-9).

$$\Delta w_{jk} = \alpha \delta_k Z_j \quad (2-9)$$

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.

7. Menghitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $Z_j$  ( $j=0,1,2,\dots,p$ ) menggunakan Persamaan (2-10).

$$\delta_{net\ j} = \sum_{k=0}^m \delta_k W_{jk} \quad (2-10)$$

Mengalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi eror menggunakan Persamaan (2-11).

$$\delta_j = \delta_{net\ j} f'(z_{net\ j}) \quad (2-11)$$

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 (2-12).

$$\delta_j = \delta_{net\ j} f'(z_{net\ j}) \quad (2-12)$$

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

8. Menghitung semua perubahan bobot ( $k=0,1,2,\dots m$ ) memperbaiki bias dan bobot ( $j=0,1,2\dots p$ ) menggunakan Persamaan (2-13).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2-13)$$

Tiap-tiap unit tersembunyi  $Z_j$  ( $j=0,1,2,\dots p$ ) memperbaiki bias dan bobotnya ( $i=0,1,2,\dots n$ ) menggunakan Persamaan (2-14).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2-14)$$

9. Menghitung MSE (*Mean Square Error*) menggunakan Persamaan (2-15).

$$MSE = \sum (t - Y_k)^2 / n \quad (2-15)$$

Di mana  $n$  merupakan jumlah dari unit *input*.

### 2.2.7 MATLAB

Matlab merupakan suatu program komputer (*software*) yang dapat membantu memecahkan berbagai masalah matematis yang terdapat dalam kehidupan sehari-hari lebih tepatnya dalam bidang teknis. Kemampuan MATLAB dapat digunakan untuk menemukan solusi dari berbagai masalah numerik secara cepat dan tepat, mulai dari hal yang paling dasar, misalkan sistem 2 persamaan dengan 2 variabel: (1)  $x - 2y = 32$ ; (2)  $12x + 5y = 12$ , hingga yang kompleks, seperti mencari akar-akar polinomial, interpolasi dari sejumlah data, perhitungan dengan matriks, pengolahan sinyal, dan metode numerik [12].

Matlab merupakan kepanjangan dari *Matrix Laboratory*. Sesuai dengan namanya, struktur data yang terdapat dalam Matlab menggunakan matriks atau array berdimensi dua (*double*). Oleh karenanya penguasaan teori matriks mutlak diperlukan bagi pengguna pemula Matlab agar mudah dalam mempelajari dan memahami operasi-operasi yang ada di Matlab [12].

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Alat dan Bahan**

Penelitian yang dilakukan untuk sistem pengenalan plat kendaraan bermotor menggunakan *character geometri* dan BPNN, dibutuhkan alat dan bahan sebagai berikut.

##### **3.1.1 Alat**

Alat yang digunakan untuk pengenalan plat kendaraan menggunakan *character geometri* dan BPNN antara lain.

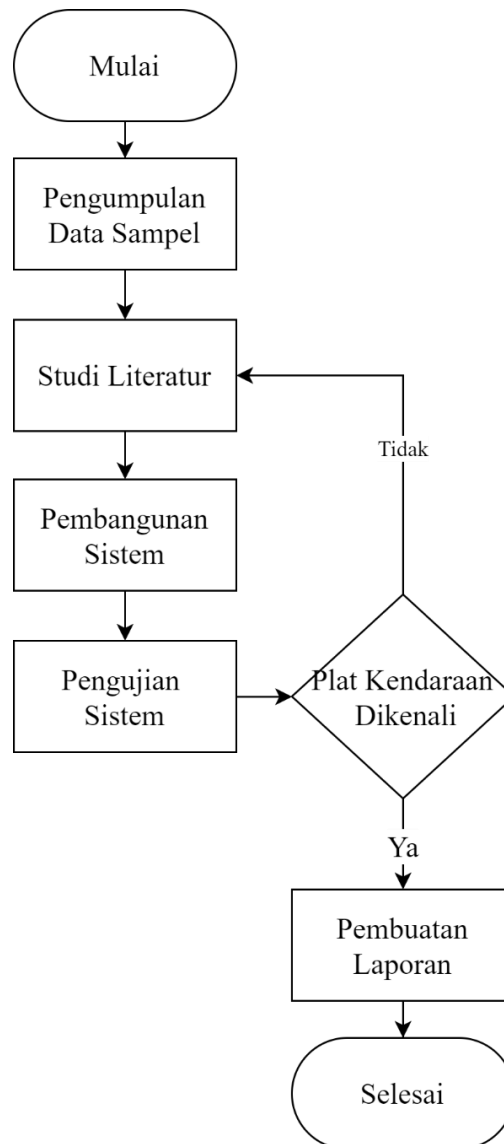
1. Laptop Processor Intel i5-3337U @1.80GHz, RAM 8 GB, HDD 500 GB
2. *Operating System* Microsoft Windows 10 64-bit OS
3. *Software* MATLAB R2017a
4. Microsoft Paint
5. *Camera Smartphone* Sony Z2 Global 20MP

##### **3.1.2 Bahan**

Bahan yang digunakan untuk pelatihan merupakan data dengan format jpg yang dibuat menggunakan *Microsoft Paint*. Sedangkan bahan yang digunakan untuk pengujian adalah citra plat kendaraan.

#### **3.2 Rencana Penelitian**

Gambar 3.1 menunjukkan merupakan diagram alur dan proses pembuatan sistem dari mulai pengumpulan data hingga pembuatan laporan dari sistem yang akan dibuat.



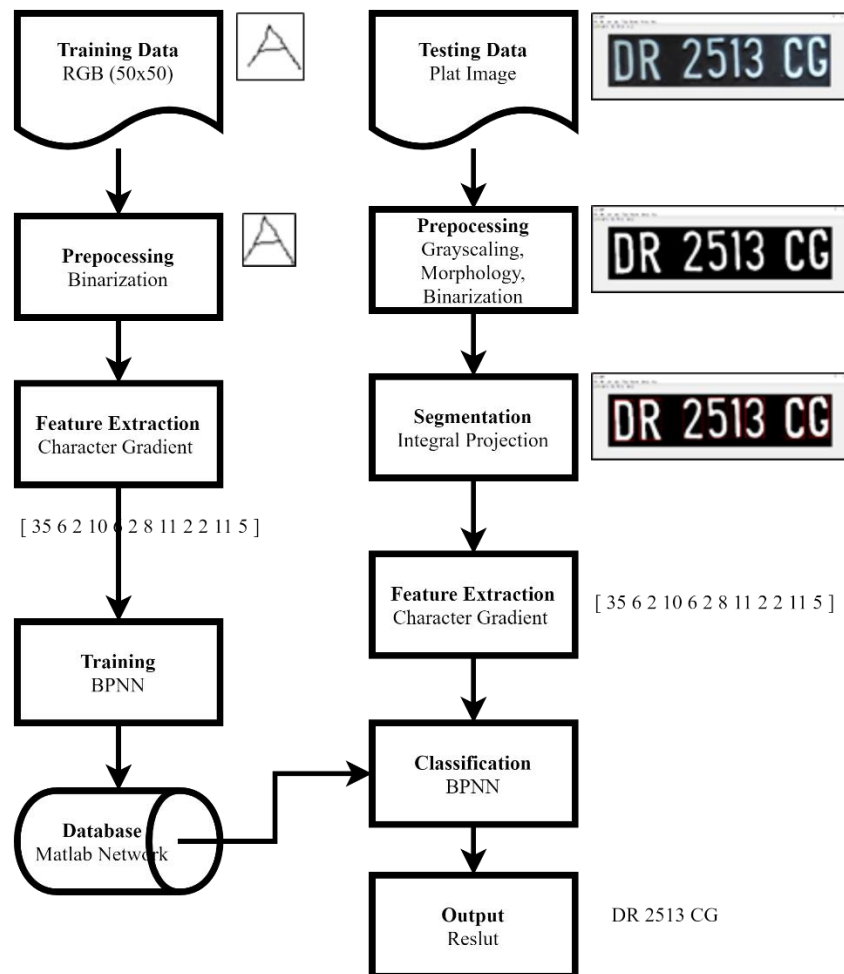
Gambar 3.1 Diagram alur proses pembuatan sistem.

Berdasarkan Gambar 3.1 terdapat tahapan-tahapan yang dilalui dalam pembuatan sistem mulai dari pengumpulan data hingga pembuatan laporan. Tahap pertama yaitu mencari pengumpulan data sampel yang terdiri dari data pelatihan dan data pengujian. Data pelatihan dibuat menggunakan Microsoft Paint dengan cara menggambar karakter menggunakan *tools* Pencil pada *canvas* dengan ukuran 50x50 pixel.

Data pengujian merupakan citra plat kendaraan dari gambar yang diambil menggunakan kamera. Data pengujian diambil di tempat parkir Fakultas Teknik Universitas Mataram.

Tahap selanjutnya yaitu melakukan studi literatur untuk mempelajari cara pembangunan sistem dan prinsip kerja dari metode yang digunakan yaitu *character gradient* dan BPNN. Kemudian melakukan pembangunan sistem dengan metode yang digunakan. Selanjutnya dilakukan pengujian terhadap sistem yang telah dibangun, apabila sistem tidak berjalan dengan baik maka akan dilakukan studi literatur kembali dan apabila sistem berjalan dengan baik maka akan dilanjutkan ke tahap pembuatan laporan.

Rencana penelitian yang akan diterapkan pada penelitian terdiri dari 2 tahap yaitu tahap pelatihan dan tahap pengujian. Tahap pelatihan merupakan tahap untuk membangun basis data acuan untuk melakukan klasifikasi yang dilakukan pada tahap pengujian. Tahap pengujian merupakan tahap klasifikasi plat kendaraan. Berikut diagram alur penelitian sistem pengenalan plat kendaraan menggunakan *character gradient* dan BPNN.



Gambar 3.2 Blok diagram sistem.

Berdasarkan blog diagram sistem pada Gambar 3.2 tersebut maka terdapat 2 tahap utama yaitu tahap pelatihan dan tahap pengujian seperti yang telah dijelaskan sebelumnya.

#### 1. Tahap Pelatihan

Tahap pelatihan terdiri dari beberapa bagian di antaranya *training data*, *preprocessing*, *feature extraction*, *training*, *database*. Berikut penjabaran dari masing-masing bagian tersebut.

##### a) *Training Data*

Data *training* dibuat menggunakan *Microsoft Paint* yang merupakan editor grafis sederhana yang telah disertakan dengan semua versi *Microsoft Windows* dan mudah untuk digunakan. Data *training* 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, citra hasil gambar tersebut disimpan dalam format .jpg seperti yang telah dijelaskan pada diagram alur proses pembuatan sistem pada bagian pengumpulan data sampel. Data *training* yang digunakan berjumlah 900 data yang terdiri dari 26 karakter alpabet dan 10 karakter numerik dan masing-masing karakter tersebut berjumlah 25 citra *training*. Pada tahapan-tahapan selanjutnya akan digunakan sebuah citra *training* alpabet 'A' seperti pada Gambar 3.3 berikut.

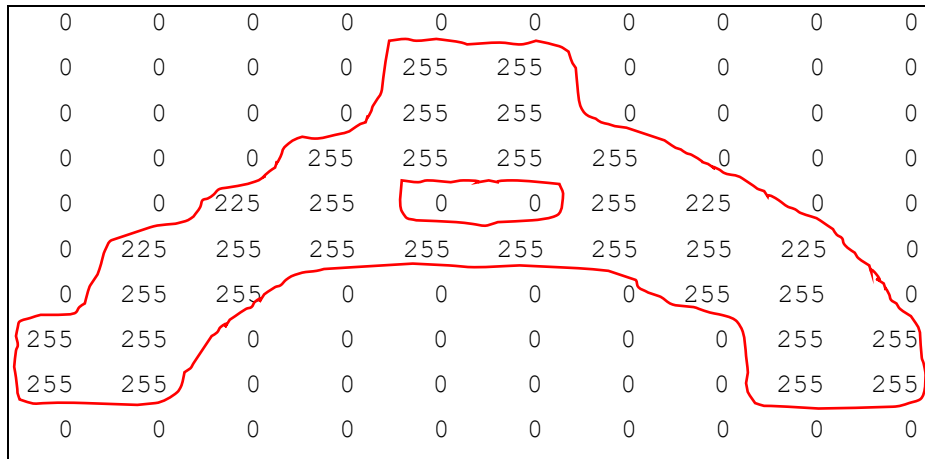


Gambar 3.3 Citra pelatihan alpabet 'A'.

##### b) *Preprocessing*

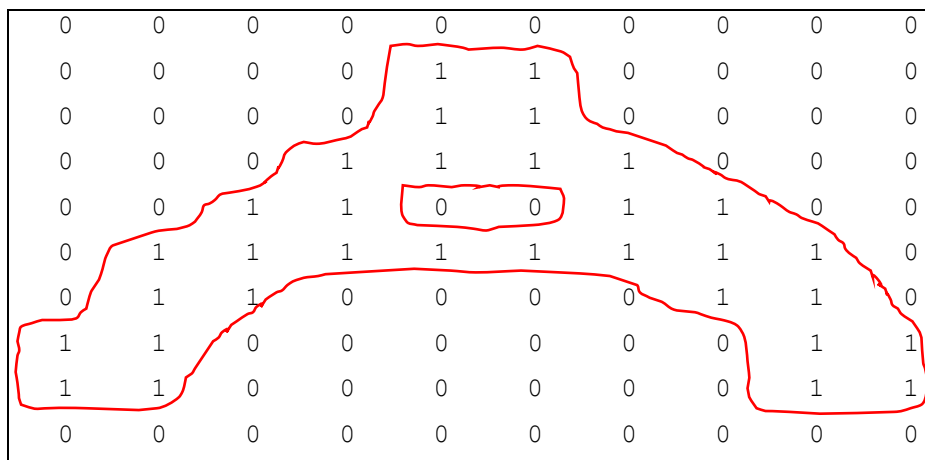
*Preprocessing* yang dilakukan pada tahap pelatihan yaitu *Binarization*. *Binarization* merupakan proses mengubah citra matriks dalam bentuk desimal menjadi citra matriks dalam bentuk biner. Misalkan terdapat citra data pelatihan alpabet 'A' dengan ukuran 10x10 pixel seperti berikut.





Gambar 3.4 Citra matriks *grayscale* alpabet A.

Matriks di atas merupakan representasi dari citra data pelatihan alpabet ‘A’ yang telah melalui proses *grayscale* menggunakan fungsi `rgb2gray` pada Matlab. *Grayscale* digunakan untuk mengubah citra RGB menjadi citra *grayscale*. Selanjutnya dilakukan binarization untuk mempermudah proses ekstraksi fitur, di mana pada proses binarization setiap pixel pada citra matriks alpabet ‘A’ tersebut diubah menjadi bentuk biner seperti matriks berikut.



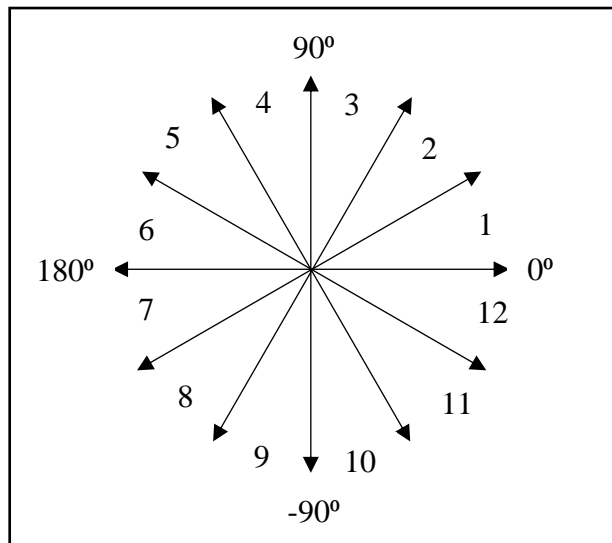
Gambar 3.5 Citra matriks biner alpabet A.

Matriks di atas merupakan representasi dari citra data pelatihan alpabet ‘A’ yang telah melalui yang telah melalui proses *binarization* menggunakan fungsi `imbinarize` pada Matlab. Fungsi `imbinarize` pada Matlab digunakan untuk mengubah citra *grayscale* menjadi citra biner.

c) *Feature Extraction*

Ekstraksi fitur dilakukan dengan menggunakan metode *character gradient* yang terdiri dari fitur vektor sebagai berikut.

1. *Number of direction between*  $(0^\circ - 30^\circ)$ .
2. *Number of direction between*  $(30^\circ - 60^\circ)$ .
3. *Number of direction between*  $(60^\circ - 90^\circ)$ .
4. *Number of direction between*  $(90^\circ - 120^\circ)$ .
5. *Number of direction between*  $(120^\circ - 150^\circ)$ .
6. *Number of direction between*  $(150^\circ - 180^\circ)$ .
7. *Number of direction between*  $(-180^\circ - -150^\circ)$ .
8. *Number of direction between*  $(-150^\circ - -120^\circ)$ .
9. *Number of direction between*  $(-120^\circ - -90^\circ)$ .
10. *Number of direction between*  $(-90^\circ - -60^\circ)$ .
11. *Number of direction between*  $(-60^\circ - -30^\circ)$ .
12. *Number of direction between*  $(-30^\circ - 0^\circ)$ .



Gambar 3.6 Arah ekstraksi fitur.

Untuk mendapatkan arah sudut dari tiap pixel pada citra *training* melalui beberapa tahap. Tahap awal, citra biner hasil *preprocessing* dilakukan konvolusi dengan operator Sobel seperti pada Gambar 2.5 dan dilakukan perhitungan dengan Persamaan (2-2) dan (2-3) untuk mendapatkan gradien horizontal ( $G_x$ ) dan gradien vertikal ( $G_y$ ). Model *edge detection* Sobel digunakan karena memiliki kelebihan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi.

Berdasarkan Persamaan (2-2) untuk menghitung gradien horizontal ( $G_x$ ) dari citra *training* didapatkan hasil seperti pada Gambar 3.7 berikut.

0	0	0	1	3	3	1	0	0	0
0	0	0	1	3	3	1	0	0	0
0	0	1	2	1	1	2	1	0	0
0	1	3	2	-2	-2	2	3	1	0
1	3	3	1	0	0	1	3	3	1
1	2	0	-2	-1	-1	-2	0	2	1
3	0	-3	-4	-4	-4	-4	-3	0	3
3	0	-2	-1	0	0	-1	-2	0	3
-4	-3	-1	0	0	0	0	-1	-3	-4
-4	-3	-1	0	0	0	0	-1	-3	-4

Gambar 3.7 Matriks gradien horizontal ( $G_x$ ).

Berdasarkan persamaan 2.3 untuk menghitung gradien vertikal ( $G_y$ ) dari citra *training* didapatkan hasil seperti pada Gambar 3.8 berikut.

0	0	0	1	3	3	1	0	0	0
0	0	0	1	3	3	1	0	0	0
0	0	1	2	1	1	2	1	0	0
0	1	3	2	-2	-2	2	3	1	0
1	3	3	1	0	0	1	3	3	1
1	2	0	-2	-1	-1	-2	0	2	1
3	0	-3	-4	-4	-4	-4	-3	0	3
3	0	-2	-1	0	0	-1	-2	0	3
-4	-3	-1	0	0	0	0	-1	-3	-4
-4	-3	-1	0	0	0	0	-1	-3	-4

Gambar 3.8 Matriks gradien vertikal ( $G_y$ ).

Setelah mendapatkan nilai  $G_x$  dan  $G_y$ , selanjutnya menghitung gradien *magnitude* dengan Persamaan (3-1) berikut.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3-1)$$

Berdasarkan Persamaan (3-1), didapatkan hasil gradien *magnitude* seperti pada Gambar 3.9 berikut.

0	0	0	1.4142	3.1623	3.1623	1.4142	0	0	0
0	0	0	3.1623	4.2426	4.2426	3.1623	0	0	0
0	0	1.4142	4.4721	3.1623	3.1623	4.4721	1.4142	0	0
0	1.4142	4.2426	2.8284	2.0000	2.0000	2.8284	4.2426	1.4142	0
1.4142	4.2426	4.2426	1.4142	2.0000	2.0000	1.4142	4.2426	4.2426	1.4142
3.1623	4.4721	0	2.8284	1.4142	1.4142	2.8284	0	4.4721	3.1623
4.2426	2.0000	4.2426	4.4721	4.0000	4.0000	4.4721	4.2426	2.0000	4.2426
3.1623	2.0000	4.4721	1.4142	0	0	1.4142	4.4721	2.0000	3.1623
4.0000	4.2426	3.1623	0	0	0	0	3.1623	4.2426	4.0000
4.0000	3.1623	1.4142	0	0	0	0	1.4142	3.1623	4.0000

Gambar 3.9 Matriks gradien *magnitude* (G).

Gambar 3.9 merupakan gradien *magnitude* yang merupakan hasil konvolusi matriks dengan menggunakan operator Sobel. Perhitungan dilakukan dengan menggunakan fungsi `imgradient(I, 'sobel')` pada Matlab di mana `I` merupakan citra *input* yaitu citra biner pada tahap *preprocessing*.

Tahap selanjutnya yaitu menghitung arah sudut & gradien dari citra *preprocessing* yang didapatkan dari Persamaan (3-2) berikut.

$$f(G_x, G_y) = \begin{cases} \arctan\left(\frac{G_y}{G_x}\right), & G_x > 0 \\ \arctan\left(\frac{G_y}{G_x}\right) - 180^\circ, & G_x < 0 \\ 90^\circ, & G_x = 0 \text{ dan } G_y > 0 \\ -90^\circ, & G_x = 0 \text{ dan } G_y < 0 \\ 0^\circ, & G_x = 0 \text{ dan } G_y = 0 \end{cases} \quad (3-2)$$

Berdasarkan Persamaan (3-2), didapatkan arah sudut gradien seperti pada Gambar 3.10 berikut.

0.00	0.00	0.00	-45.00	-71.57	-108.43	-135.00	0.00	0.00	0.00
0.00	0.00	0.00	-18.43	-45.00	-135.00	-161.57	0.00	0.00	0.00
0.00	0.00	-45.00	-26.57	-18.43	-161.57	-153.43	-135.00	0.00	0.00
0.00	-45.00	-45.00	-45.00	90.00	90.00	-135.00	-135.00	-135.00	0.00
-45.00	-45.00	-45.00	-135.00	-180.00	0.00	-45.00	-135.00	-135.00	-135.00
-18.43	-26.57	0.00	135.00	135.00	45.00	45.00	0.00	-153.43	-161.57
-45.00	0.00	135.00	116.57	90.00	90.00	63.43	45.00	-180.00	-135.00
-71.57	-180.00	153.43	135.00	0.00	0.00	45.00	26.57	0.00	-108.43
90.00	135.00	161.57	0.00	0.00	0.00	0.00	18.43	45.00	90.00
90.00	108.43	135.00	0.00	0.00	0.00	0.00	45.00	71.57	90.00

Gambar 3.10 Matriks arah sudut gradien.

Gambar 3.10 merupakan arah sudut gradien di mana perhitungan dilakukan dengan menggunakan fungsi  $[Gmag, Gdir] = \text{imgradient}(I, 'sobel')$  pada Matlab di mana  $I$  merupakan citra *input* yaitu citra biner pada tahap *preprocessing* dan  $Gdir$  merupakan arah sudut gradien. Selanjutnya yaitu normalisasi dengan Persamaan (3-3). Normalisasi dilakukan untuk mempermudah perhitungan pada saat melakukan klasifikasi menggunakan JST backpropagation.

$$f(x) = \begin{cases} 0, & 0 \leq x < 30 \\ 1, & 30 \leq x < 60 \\ 2, & 60 \leq x < 90 \\ 3, & 90 \leq x < 120 \\ 4, & 120 \leq x < 150 \\ 5, & 150 \leq x < 180 \\ 6, & -180 \leq x < -150 \\ 7, & -150 \leq x < -120 \\ 8, & -120 \leq x < -90 \\ 9, & -90 \leq x < -60 \\ 10, & -60 \leq x < -30 \\ 11, & -30 \leq x < 0 \end{cases} \quad (3-3)$$

Berdasarkan persamaan 3.3, didapatkan hasil normalisasi arah sudut gradien seperti pada Gambar 3.11 berikut.

0	0	0	10	9	8	7	0	0	0
0	0	0	11	10	7	6	0	0	0
0	0	10	11	11	6	6	7	0	0
0	10	10	10	3	3	7	7	7	0
10	10	10	7	6	0	10	7	7	7
11	11	0	4	4	1	1	0	6	6
10	0	4	3	3	3	2	1	6	7
9	6	5	4	0	0	1	0	0	8
3	4	5	0	0	0	0	0	1	3
3	3	4	0	0	0	0	1	2	3

Gambar 3.11 Matriks normalisasi arah sudut gradien.

Tahap terakhir yaitu menghitung jumlah normalisasi arah sudut gradien dari citra *preprocessing*. Dan didapatkan fitur vektor [35 6 2 10 6 2 8 11 2 2 11 5] yang didapatkan dari jumlah nilai 0 hingga 11 pada matriks normalisasi arah sudut gradien.

d) *Training*

*Training* dilakukan dengan menggunakan metode JST backpropagation. Misalkan didapatkan hasil dari ekstraksi fitur dengan fitur vektor [1 1 0 2] dengan laju pembelajaran 0.2 dan target *Output* 0.

Arsitektur jaringan yang digunakan untuk soal tersebut yaitu 4 *input*, 1 *hidden layer* dan 1 *Output*. Dalam *hidden layer* terdapat 3 neuron.

1) Pelatihan pada iterasi pertama

Mula-mula bobot diberi nilai acak yang kecil (*range* [-1,1]). Misalkan bobot  $V_{ij}$  seperti tampak pada tabel berikut.

Tabel 3.1 Nilai bobot  $V_{ij}$  secara acak

	$Z_0$	$Z_1$	$Z_2$
$X_0$	0.1	0.2	-0.4
$X_1$	-0.1	0.1	0.1
$X_2$	0.2	0.2	0.3
$X_3$	0.3	0.1	0.2
$X_4$	0.2	0.3	-0.2

Tabel 3.2 Nilai bobot  $W_{ij}$  secara acak

	Y
$Z_0$	-0.3
$Z_1$	0.4
$Z_2$	0.5
W	-0.1

Menghitung keluaran unit tersembunyi  $Z_{netj}$  dengan Persamaan (2-4).

$$Z_{net0} = 0,1+1(-0,1)+1(0,2)+0(0,3)+2(0,2) = 0,6$$

$$Z_{net1} = 0,2+1(0,1)+1(0,2)+0(0,1)+2(0,3) = 1,1$$

$$Z_{net2} = -0,4+1(0,1)+1(0,2)+0(0,2)+2(-0,2) = -0,4$$

Menghitung keluaran unit tersembunyi  $Z_j$  dengan Persamaan (2-5).

$$Z_0 = f(Z_{\text{net } 0}) = \frac{1}{1+e^{-0,6}} = 0,64$$

$$Z_1 = f(Z_{\text{net } 1}) = \frac{1}{1+e^{-1,1}} = 0,75$$

$$Z_2 = f(Z_{\text{net } 2}) = \frac{1}{1+e^{0,4}} = 0,4$$

Menghitung  $Y_{\text{net } k}$  dengan Persamaan (2-6).

$$Y_{\text{net}} = -0,1 + 0,64(-0,3) + 0,75(0,4) + 0,4(0,5) = 0,2$$

Menghitung  $Y_{\text{net } k}$  dengan Persamaan (2-7).

$$Y_k = \frac{1}{1+e^{-0,2}} = 0,55$$

Menghitung MSE dengan Persamaan (2-15).

$$\text{MSE} = \frac{(0-0,55)^2}{4} = 0,075$$

Menghitung faktor  $\delta$  di unit keluaran  $Y_k$  dengan Persamaan (2-8).

$$\delta_k = (t-Y) Y(1-Y) = (0-0,55)(0,55)(1-0,55) = -0,1$$

Menghitung suku perubahan bobot  $W_{jk}$  dengan Persamaan (2-9).

$$\Delta W_0 = 0,2 (-0,1) = -0,02$$

$$\Delta W_1 = 0,2 (-0,1) (0,64) = -0,01$$

$$\Delta W_2 = 0,2 (-0,1) (0,75) = -0,02$$

$$\Delta W_3 = 0,2 (-0,1) (0,4) = -0,01$$

Menghitung jumlah kesalahan unit tersembunyi dengan Persamaan (2-10).

$$\delta_{\text{net } 0} = (-0,1)(-0,3) = 0,03$$

$$\delta_{\text{net } 1} = (-0,1)(0,4) = -0,04$$

$$\delta_{\text{net } 2} = (-0,1)(0,5) = -0,05$$

Menghitung faktor kesalahan  $\delta$  di unit tersembunyi dengan Persamaan (2-11).

$$\delta_1 = 0,03(0,6)(1-0,6) = 0,007$$

$$\delta_2 = -0,04(1,1)(1-1,1) = 0,002$$

$$\delta_3 = -0,05(-0,4)(1-(-0,4)) = 0,028$$

Suku perubahan bobot  $\Delta V_{ij}$  dengan Persamaan (2-12).

Tabel 3.3 Nilai suku perubahan bobot

	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>
X <sub>0</sub>	$\Delta V_{00}=(0,2)(0,0072)\approx 0$	$\Delta V_{01}=(0,2)(0,0016)\approx 0$	$\Delta V_{02}=(0,2)(0,028)\approx 0$
X <sub>1</sub>	$\Delta V_{10}=(0,2)(0,0072)(1)\approx 0$	$\Delta V_{11}=(0,2)(0,0016)(1)\approx 0$	$\Delta V_{12}=(0,2)(0,028)(1)\approx 0$
X <sub>2</sub>	$\Delta V_{20}=(0,2)(0,0072)(1)\approx 0$	$\Delta V_{21}=(0,2)(0,0016)(1)\approx 0$	$\Delta V_{22}=(0,2)(0,028)(1)\approx 0$
X <sub>3</sub>	$\Delta V_{30}=(0,2)(0,0072)(0)\approx 0$	$\Delta V_{31}=(0,2)(0,0016)(0)\approx 0$	$\Delta V_{32}=(0,2)(0,028)(0)\approx 0$
X <sub>4</sub>	$\Delta V_{40}=(0,2)(0,0072)(2)\approx 0$	$\Delta V_{41}=(0,2)(0,0016)(2)\approx 0$	$\Delta V_{43}=(0,2)(0,028)(2)\approx 0$

Perubahan bobot unit keluaran dengan menggunakan Persamaan (2-12).

$$W_0(\text{baru}) = -0,1 - 0,02 = 0,12$$

$$W_1(\text{baru}) = -0,3 - 0,01 = -0,31$$

$$W_2(\text{baru}) = 0,4 - 0,02 = 0,38$$

$$W_3(\text{baru}) = 0,5 - 0,01 = 0,49$$

Perubahan bobot unit tersembunyi dengan menggunakan persamaan (2-17).

Tabel 3.4 Nilai bobot pada unit tersembunyi yang baru

	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>
X <sub>0</sub>	0,1	0,2	-0,4
X <sub>1</sub>	-0,1	0,1	0,1
X <sub>2</sub>	0,2	0,2	0,3
X <sub>3</sub>	0,3	0,1	0,2
X <sub>4</sub>	0,2	0,3	-0,2

2) Uji coba hasil dari pelatihan pertama

Tabel 3.5 Nilai bobot V<sub>ij</sub> hasil 1 kali pelatihan

	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>
X <sub>0</sub>	0.1	0.2	-0.4
X <sub>1</sub>	-0.1	0.1	0.1
X <sub>2</sub>	0.2	0.2	0.3
X <sub>3</sub>	0.3	0.1	0.2
X <sub>4</sub>	0.2	0.3	-0.2



Tabel 3.6 Nilai bobot  $W_{ij}$  hasil 1 kali pelatihan

	Y
$Z_0$	-0.3
$Z_1$	0.4
$Z_2$	0.5
W	-0.1

Menghitung keluaran unit tersembunyi  $Z_{\text{net } j}$  dengan Persamaan (2-4).

$$Z_{\text{net } 0} = 0,1+1(-0,1)+1(0,2)+0(0,3)+2(0,2) = 0,6$$

$$Z_{\text{net } 1} = 0,2+1(0,1)+1(0,2)+0(0,1)+2(0,3) = 1,1$$

$$Z_{\text{net } 2} = -0,4+1(0,1)+1(0,2)+0(0,2)+2(-0,2) = -0,4$$

Menghitung keluaran unit tersembunyi  $Z_j$  dengan Persamaan (2-5).

$$Z_0 = f(Z_{\text{net } 0}) = \frac{1}{1+e^{-0,6}} = 0,64$$

$$Z_1 = f(Z_{\text{net } 1}) = \frac{1}{1+e^{-1,1}} = 0,75$$

$$Z_2 = f(Z_{\text{net } 2}) = \frac{1}{1+e^{0,4}} = 0,4$$

Menghitung  $Y_{\text{net } k}$  dengan Persamaan (2-6).

$$Y_{\text{net}} = -0,1 + 0,64(-0,31) + 0,75(0,38) + 0,4(0,49) = 0.16$$

Menghitung  $Y_{\text{net } k}$  dengan persamaan (2-7).

$$Y_k = \frac{1}{1+e^{-0,16}} = 0,53$$

Menghitung MSE dengan persamaan (2-15).

$$\text{MSE} = \frac{(0-0,53)^2}{4} = 0,070$$

Jadi, setelah melakukan pelatihan sebanyak 1 kali terhadap data dan dilakukan uji coba didapatkan nilai MSE yang sebelumnya 0,075 mengalami penurunan nilai menjadi 0,070. Pelatihan akan terus dilakukan hingga nilai MSE lebih kecil dari nilai target eror yang diinginkan. Sehingga dapat disimpulkan

bahwa semakin banyak data latih maka semakin kecil nilai dari MSE yang dihasilkan.

e) *Database*

Hasil dari *training* akan disimpan dalam *database* dalam bentuk *file* matlab MAT-files (\*.mat) sehingga tidak perlu melakukan *training* setiap kali akan melakukan *testing*.

2. Tahap Pengujian

a) *Testing Data*

Data *testing* yang digunakan merupakan citra plat kendaraan yang diambil menggunakan kamera seperti yang telah dijelaskan pada diagram alur proses pembuatan sistem pada bagian pengumpulan data sampel.



Gambar 3.12 Contoh gambar data *testing*.

Gambar 3.12 Merupakan contoh gambar data *testing* yang akan digunakan untuk pengujian. Sebelum digunakan untuk pengujian, dilakukan proses *cropping* pada Gambar 3.12 secara manual menggunakan Microsoft Paint. Selanjutnya akan didapatkan hasil seperti Gambar 3.12 berikut.



Gambar 3.13 Hasil *cropping* data *testing*.

b) *Preprocessing*

*Preprocessing* pada tahap pengujian berbeda dengan tahap pelatihan karena *input* pada tahap pelatihan berupa citra sebuah karakter seperti Gambar 3.2 sedangkan *input* pada tahap pengujian berupa plat kendaraan seperti Gambar 3.13. *Preprocessing* pada tahap pengujian terdiri dari *grayscale*, *binarization*, dan *noise filtering*. Untuk proses *grayscale* dan *binarization* sama dengan proses *grayscale* dan *binarization* pada tahap pelatihan. Setelah didapatkan citra biner dilakukan *noise filtering* untuk menghilangkan objek-objek kecil yang akan mengganggu proses segmentasi. Berikut merupakan contoh hasil tiap-tiap proses dari tahap *preprocessing*.



Gambar 3.14 Citra *input*.



Gambar 3.15 Citra *grayscale*.



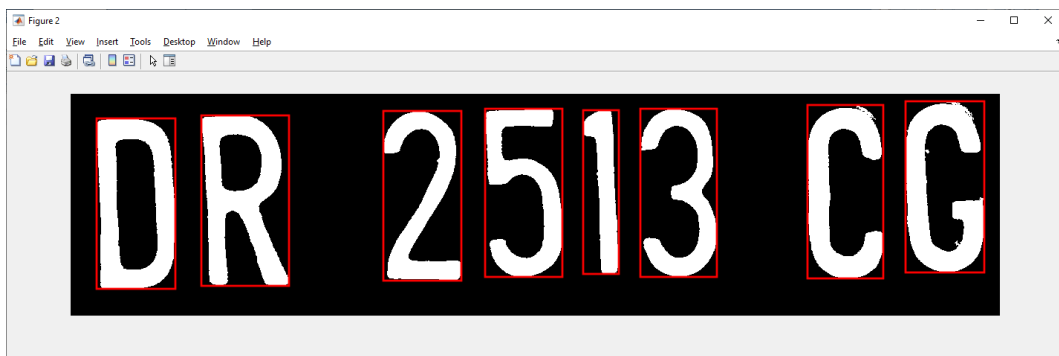
Gambar 3.16 Citra biner.



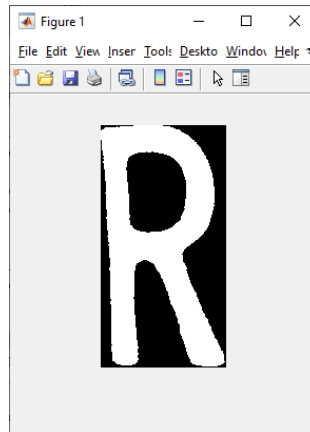
Gambar 3.17 Citra setelah *noise filtering*.

c) *Segmentation*

Segmentasi karakter bertujuan untuk memisahkan setiap karakter yang ada pada citra. Segmentasi dilakukan dengan menggunakan fungsi `regionprops(I, 'BoundingBox')` pada Matlab di mana `I` merupakan citra *input* yaitu citra setelah *noise filtering* seperti pada Gambar 3.16, dan `'BoundingBox'` merupakan *properties* dari fungsi `regionprops`. Citra hasil dari segmentasi karakter dapat dilihat pada Gambar 3.18 berikut.

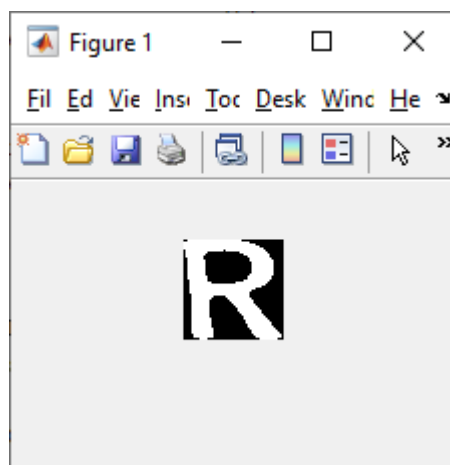


Gambar 3.18 Citra segmentasi karakter.



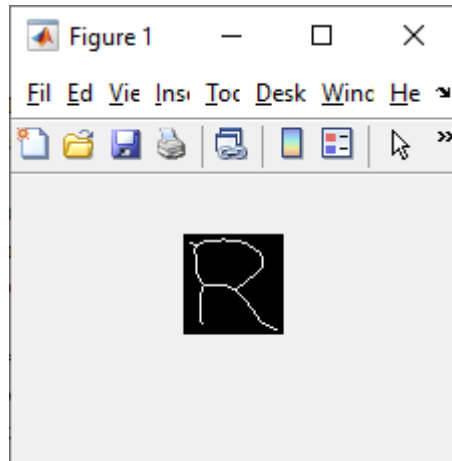
Gambar 3.19 Contoh segmentasi karakter 'R'.

Setelah segmentasi tiap-tiap karakter dilakukan, maka proses selanjutnya yaitu melakukan *resizing* dan *thinning*. *Resizing* dilakukan dengan menggunakan fungsi `imresize(I, [50 50])` pada Matlab di mana  $I$  merupakan citra *input* yaitu citra hasil segmentasi seperti pada Gambar 3.18, dan  $[50\ 50]$  merupakan ukuran citra *Output* dari hasil *resize*. Citra hasil *resizing* karakter dapat dilihat pada Gambar 3.20 berikut.



Gambar 3.20 Citra hasil *resizing* 50x50 pixel.

Selanjutnya yaitu melakukan proses *thinning*. *Thinning* dilakukan dengan menggunakan metode Zhang-Suen. Citra hasil *thinning* karakter dapat dilihat pada Gambar 3.21 berikut.



Gambar 3.21 Citra hasil *thinning*.

Proses *resizing* dan *thinning* digunakan untuk menyelaraskan *input* pada tahap pengujian dengan hasil *preprocessing* pada tahap pelatihan, sehingga data yang masuk pada ekstraksi fitur tahap pengujian identik dengan data yang masuk pada ekstraksi fitur tahap pelatihan.

d) *Feature Extraction*

Ekstraksi fitur dilakukan dengan menggunakan metode *character gradient*. Hasil ekstraksi fitur pada tahap pelatihan ini sepenuhnya sama dengan hasil ekstraksi fitur pada tahap pelatihan.

e) *Classification*

Klasifikasi dilakukan dengan menggunakan metode BPNN. Hasil klasifikasi pada tahap pelatihan ini sepenuhnya sama dengan hasil hitungan pada tahap pelatihan bagian *Training* bagian 2 (uji coba hasil dari pelatihan).

f) *Output Result*

Hasil *Output* merupakan pengolahan dari hasil klasifikasi menjadi karakter. Misalkan hasil klasifikasi yang didapatkan adalah 35, maka akan diolah menjadi karakter 'Z'.

### 3.3 Teknik Pengujian

Teknik pengujian yang akan diterapkan pada penelitian ini yaitu menghitung tingkat akurasi pada percobaan sistem. Data yang digunakan untuk pelatihan sebanyak 900 data karakter dan data pengujian sebanyak 100 data plat kendaraan. Pada percobaan ini akan dilihat apakah sistem dapat mengklasifikasikan karakter

sesuai dengan yang diinginkan. Keberhasilan sistem mengklasifikasikan karakter dilihat dari kemampuan sistem mengenali karakter dengan benar.

Untuk menghitung tingkat akurasi dari sistem pada tugas akhir ini akan menggunakan *confusion matrix*. *Confusion matrix* merupakan alat berbentuk tabel yang digunakan untuk menghitung tingkat akurasi dari suatu sistem yang memanfaatkan *machine learning*.

Tabel 3.7 *Confusion matrix* untuk teknik pengujian

		Prediksi Kondisi	
		<i>Positive</i>	<i>Negative</i>
Kondisi TRUE	Total Populasi		
	Kondisi <i>Positive</i>	<b><i>True Positive</i></b>	<i>False Negative</i>
	Kondisi <i>Negative</i>	<i>False Positive</i>	<b><i>True Negative</i></b>

Pada Tabel 3.1 terdapat 4 parameter penilaian dari sistem yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. *True Positive* merupakan kondisi di mana suatu data yang sudah benar didapatkan hasil benar pada sistem. *True Negative* merupakan kondisi di mana suatu data yang sudah dianggap salah oleh sistem. *False Positive* merupakan kondisi eror sistem di mana suatu data yang salah dianggap benar pada sistem. *False Negative* merupakan kondisi eror sistem di mana suatu data yang benar dianggap salah pada sistem.

$$Accuracy = \frac{\sum True\ Positive + \sum True\ Negative}{\sum Total\ Populasi} \quad (3-4)$$

Persamaan (3-4) merupakan persamaan yang digunakan untuk menghitung tingkat akurasi dari sistem. Tingkat akurasi merupakan hasil dari jumlah kali pengujian dengan kondisi *True Positive* ditambah jumlah hasil pengujian dengan kondisi *True Negative* kemudian dibagi dengan seluruh jumlah data yang digunakan untuk proses pengujian. Selain parameter tingkat akurasi, parameter lain yang digunakan pada pengujian sistem ini yaitu TPR (*True Positive Rate*), FPR (*False Positive Rate*), FNR (*False Negative Rate*), dan TNR (*True Negative Rate*). Persamaan dari parameter tersebut ditunjukkan seperti pada persamaan berikut:

$$TPR = \frac{\Sigma \text{True Positif}}{\Sigma \text{Condition Positive}} \quad (3-5)$$

$$FPR = \frac{\Sigma \text{False Positif}}{\Sigma \text{Condition Negative}} \quad (3-6)$$

$$FNR = \frac{\Sigma \text{False Negative}}{\Sigma \text{Condition Positive}} \quad (3-7)$$

$$TNR = \frac{\Sigma \text{True Negative}}{\Sigma \text{Condition Negative}} \quad (3-8)$$

### 3.4 Jadwal Penelitian

Untuk melakukan penelitian dengan judul di atas, maka akan dilaksanakan berdasarkan estimasi jadwal seperti pada tabel berikut.

Tabel 3.8 Estimasi jadwal penelitian

No.	Kegiatan	Bulan					
		1	2	3	4	5	6
1.	Studi literatur						
2.	Survei citra						
3.	Desain sistem						
4.	Pembuatan program						
5.	Implementasi						
6.	Pengujian						
7.	Pembuatan Laporan						

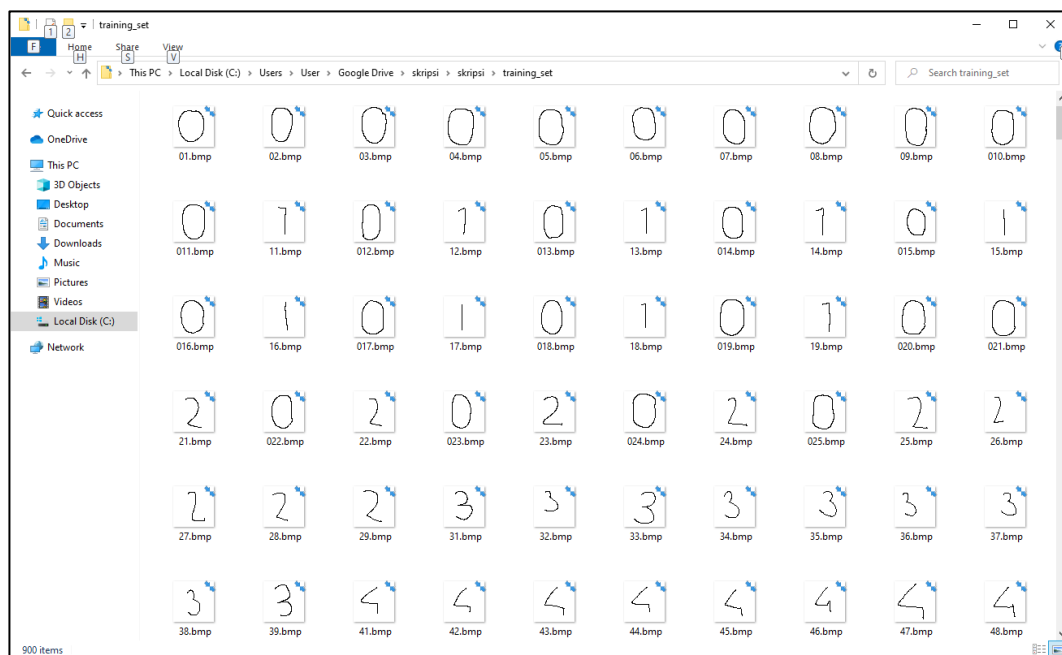


## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 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* Pensil 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 4.1.



Gambar 4. 1 Data Latih

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 4.2.



Gambar 4.2 Data Uji

## 4.2 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.

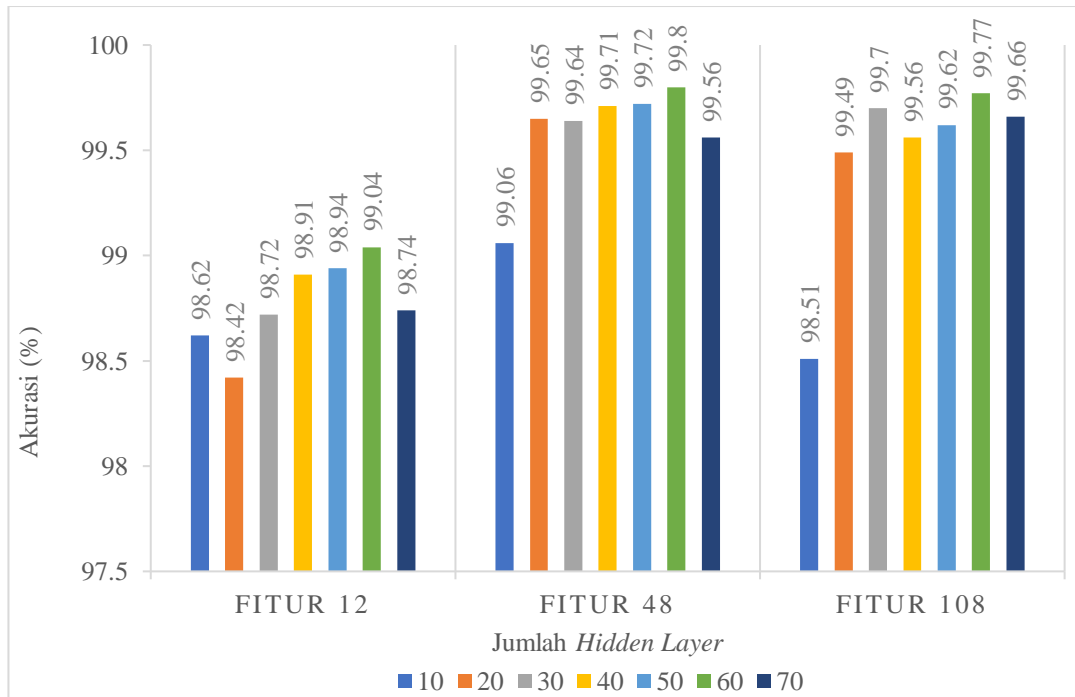
### 4.2.1 Pengujian Skenario Pertama

Pengujian skenario pertama dilakukan untuk mengetahui pengaruh jumlah fitur terhadap performa sistem, dilakukan manipulasi jumlah *hidden layer* dimana *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 :

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

#### 4.2.1.1 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 BAB III. Hasil persentase akurasi berdasarkan jumlah fitur dapat dilihat pada Gambar 4.3.



Gambar 4.3 Perbandingan tingkat akurasi berdasarkan jumlah fitur

Berdasarkan Gambar 4.3 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 4.3 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*.

#### 4.2.1.2 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 4.4.



Gambar 4.4 Persentase tingkat presisi berdasarkan jumlah *hidden layer*

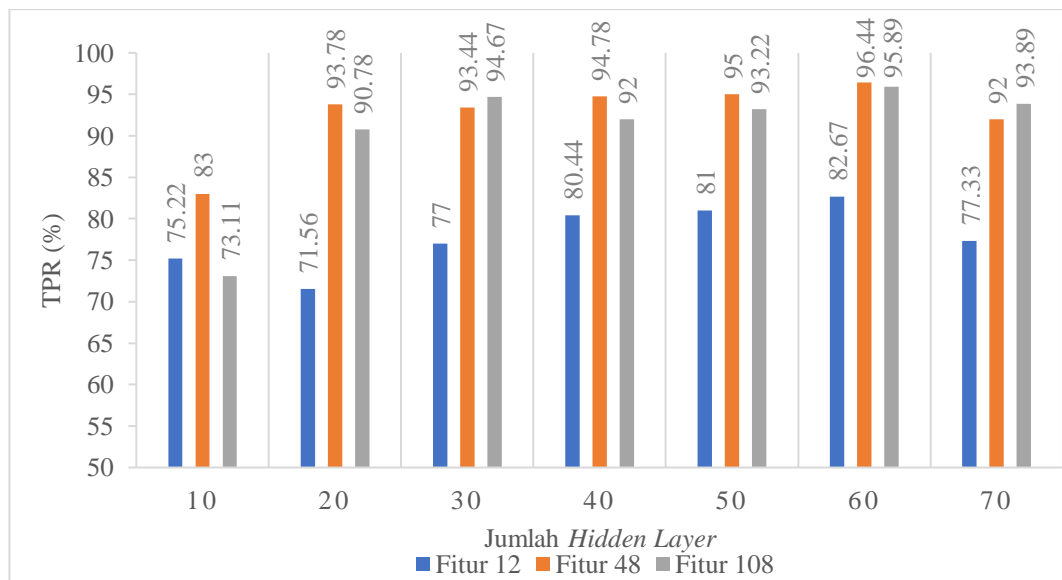
Persentase tingkat presisi pada Gambar 4.4 menunjukkan bahwa tingkat presisi paling tinggi terdapat pada fitur 48 dengan nilai presisi yaitu 96,60% dengan jumlah *hidden layer* 60. Jumlah fitur mempengaruhi nilai presisi yang didapatkan, semakin meningkat jumlah fitur, maka tingkat presisi yang didapatkan meningkat pula. Seperti terlihat pada Gambar 4.4 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.

#### 4.2.1.3 True Postitive Rate (TPR), True Negative Rate (TNR), False Negative Rate (FNR), dan False Postive Rate (TPR)

##### 1. True Postive Rate (TPR)

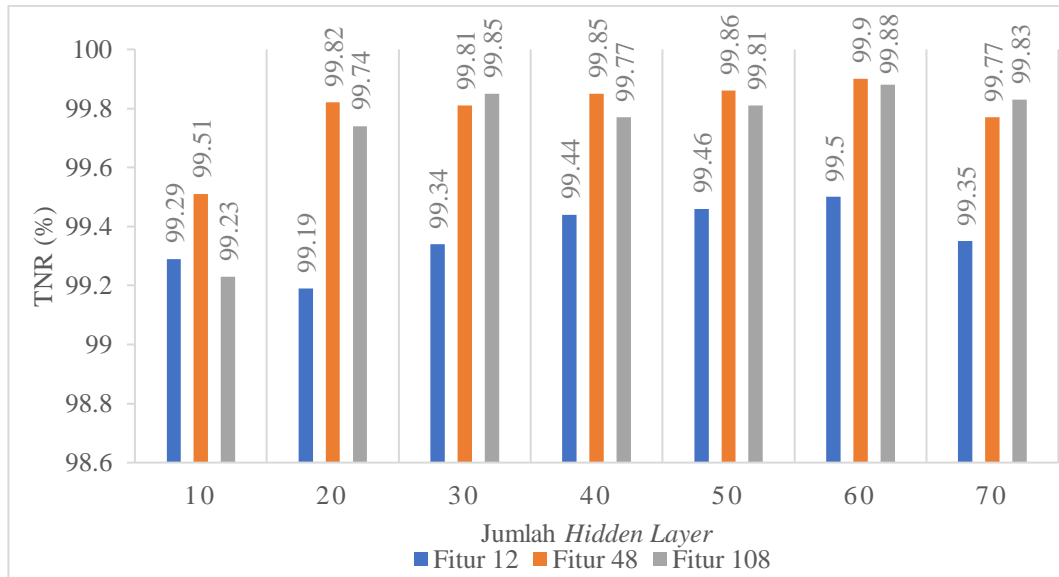
TPR menunjukkan kemampuan sistem dalam memprediksi hasil dengan benar, dengan kata lain TPR menunjukkan tingkat deteksi dari sistem, pada Gambar 4.5 peningkatan nilai TPR juga dipengaruhi dengan bertambahnya jumlah fitur yang digunakan, hal ini dapat dilihat pada data dengan jumlah *hidden layer* 30 dan 70 dengan nilai TPR fitur 12, 48, dan 108 yaitu 77%, 93,44%, dan 94,67%.



Gambar 4.5 Persentase TPR berdasarkan jumlah *hidden layer*

##### 2. True Negative Rate (TNR)

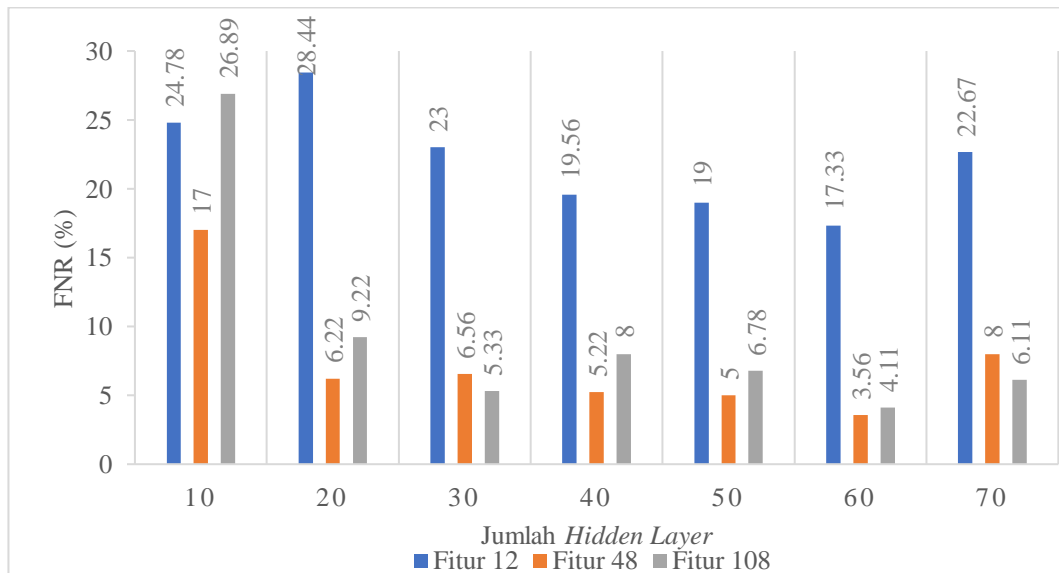
TNR menunjukkan nilai data salah yang berhasil diidentifikasi salah oleh sistem, nilai TNR dapat diukur dengan menggunakan formulasi (3 – 8) pada Bab III. Gambar 4.6 menunjukkan nilai TNR terendah terdapat pada fitur 12 dengan jumlah *hidden layer* 20 dan TNR tertinggi terdapat pada fitur 48 dengan jumlah *hidden layer* 60, Pada Gambar 4.6 menampilkan data TNR dengan nilai di atas 99%, dan nilai TNR tertinggi yaitu 99,90% yang terdapat pada fitur 48 dengan jumlah *hidden layer* 60.



Gambar 4.6 Persentase TNR berdasarkan jumlah *hidden layer*

### 3. *False Negative Rate (FNR)*

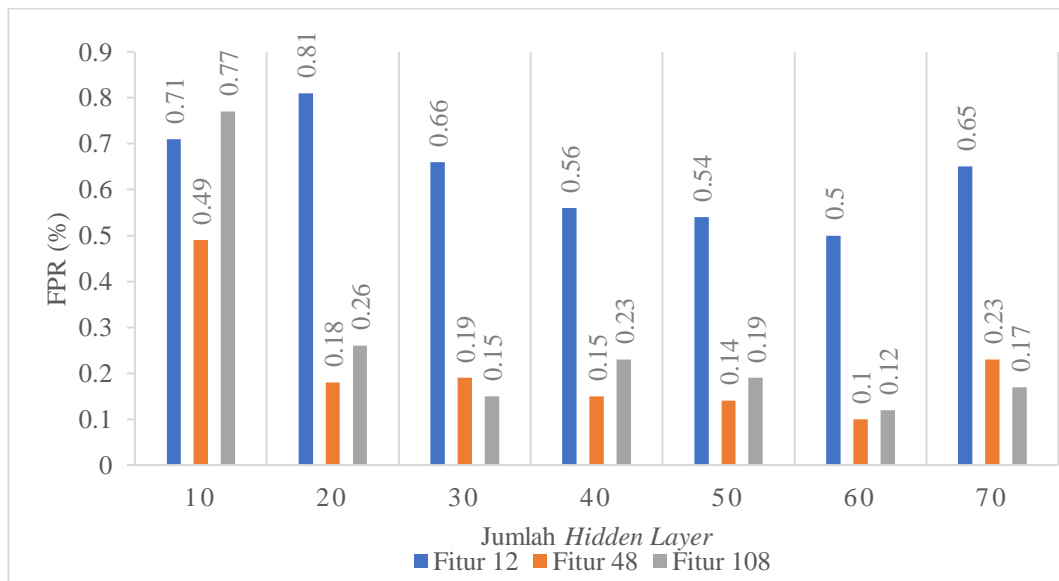
FNR atau disebut pula tingkat kesalahan sistem dalam memprediksi hasil dari *input* yang masuk ke dalam sistem, pada Gambar 4.7 tingkat kesalahan terendah dimiliki oleh fitur 48 pada *hidden layer* 60 dengan persentase 3,56%, sedangkan untuk tingkat kesalahan tertinggi dimiliki oleh fitur 12 pada *hidden layer* 20 dengan persentase mencapai 28,44%.



Gambar 4.7 Persentase FNR berdasarkan jumlah *hidden layer*

#### 4. *False Postive Rate (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 4.8 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 4.8 Persentase FPR berdasarkan jumlah *hidden layer*



#### 4.2.2 Pengujian Skenario Kedua

Pengujian skenario kedua dilakukan dengan menggunakan data citra plat kendaraan, terdapat 10 citra yang akan dilakukan pengujian untuk melihat hasil *output*, tingkat presisi, dan waktu yang dibutuhkan untuk mengolah data dengan menggunakan fitur 12, 48, dan 108 dengan jumlah *hidden layer* yaitu 10, 20, 30, 40, 50, 60, dan 70 *hidden layer*. Hasil pengujian ditunjukkan pada tabel-tabel berikut:

Tabel 4.1 *Output* berdasarkan jumlah *hidden layer* pada masing-masing fitur

Citra	Fitur	Jumlah Hidden Layer						
		10	20	30	40	50	60	70
DR 4150 CR	12	YY4Y3YYY	CC4ICYQ	YDYYPYYX	YY4J3YJ0	JY4Y3YYX	YY4Y3YYQ	YYYY3YYY
	48	NR31SO1Q	OR41GO4R	AQ4A9AAQ	4Q4Y4OO4	YQ4Y3O1Q	HR4Y1O0R	OR415OOR
	108	DK2KVMCK	OR413OCR	NR4TBOCP	OP4D5OCR	OQ4T8OCQ	OR4T5OOR	OR4D5OOR
DR 3629 YD	12	YYYO4OYY	YCYY4YYY	YYY444YY	YYYX3XYY	YYY34YYY	YYY24XYY	YYYY4YYY
	48	9R3QXVY9	OR7O24YO	DRJOGJYD	OR3O44YO	DRQO4QYD	DR4P2QYD	DR3O2QMD
	108	DRU629YD	ORFKKOYO	DRHOP9VP	DRO8P9VO	OQS6P9YO	OR56Z9YO	OR36ZOYO
DR 3704 SM	12	YYY4Y4YY	YY4IYYXY	YX4YYYYV	YY34YYYV	YY34YYYV	YY44Y7YV	YY34YYYV
	48	VS31QGSV	OX31OGG7	HA7A07VV	OX34O437	OY74O43V	DY31O4YV	DA31O43V
	108	DK9FWANT	BPFOOQ3W	DX3ZO43K	DR3ZO43M	CX3ZOH3M	DQ3FO43W	OQ3ZO4OW
DK 5212 OC	12	YY34Y4YY	CY44J4YY	4Y44YYYY	JY33J3YJ	JY34J4YY	JY44Y4YY	JY34Y4YY
	48	13SX1X11	O4GOJ2GC	T497T7AA	OX44TYOO	QX4212QQ	ORS212OU	ORO212OO
	108	DKHDQDCC	OKFZQZOC	DKH922OC	OK8ZQDOC	OK59T2OC	OK5ZTZOO	OK5ZDZOO



Citra	Fitur	Jumlah Hidden Layer						
		10	20	30	40	50	60	70
<b>DR 2513 CG</b>	12	YYO3YYYY	CY431XYY	YYY4Y4YY	JY33J3JY	JY43V3YY	PY43Y2YY	RY43Y3YY
	48	1R7S131Q	OR2JJGO3	HR7913A1	4R24T3O3	DR231QOQ	HR21Y3O3	DR2513OQ
	108	DRDS3TCJ	ORZSAKCK	NRZBT3OQ	OQD5Q3OO	OQ95T3CC	ORZ5T3OO	ORZ5D3C6
<b>DR 6744 HM</b>	12	YYY4Y4CY	YCYIYYIY	YX4YYYYY	YYXYYYMY	YX3Y442Y	YY27444Y	YYYYYY4Y
	48	1A37334V	7PY771HY	HAH744HM	4R33444X	DRQ7444W	HR87424U	JRQ7444W
	108	JNJ7CKKY	OR8144HM	0Q87O4HM	D887M4MM	DX8244HM	DR8744AM	OR87O4HM
<b>DR 5190 UA</b>	12	YY3YOYCY	CCC3YIV	YY4YXYIV	CY3J3YYV	JY3Y3Y2Y	YY3Y2YUV	2Y3IOY2V
	48	1R3A9OUA	ORJ19O7A	DRVT9O1A	OR449OUA	DRQ19OUA	DR519OUA	DR51OOHA
	108	DRVC9DUC	ORWQOOUA	PRHT9PUA	DR818OUA	OR5T9OUA	DR5T9OUA	OR5O9OUA
<b>DR 2058 HP</b>	12	YY4Y3OCC	YY4Y4YIC	YY4Y44IY	JY3Y33CC	JY4Y3322	RY4Y3O74	RY4Y3O24
	48	1QXQSX71	OR2GGGH4	TR4T9HHA	4R4O44HP	DR2Q38HY	HR2O14HI	OR2O5QH1
	108	MR2MSJHR	ORZO38MP	PQ9OS8HP	JQZO58HP	OQZOS8HP	ORZOS8HP	ORZOS8HP
<b>DR 6944 YC</b>	12	YYOO44YY	YYOY44YY	YXX4YYYY	YYXX4YYY	2XXY44YY	2YOO44YY	2YOOYYYY
	48	1AX94WYO	OAG441YD	HAAY44YO	OP4444YO	OAQY44YO	HRQ442YO	DPQO44MO
	108	DR6VLLYC	ORKO4LYO	PR8OZZYO	DP88Z4YO	OQ68Z4YO	DR6944YO	OR6944YO
<b>DR 6658 L</b>	12	YYOO3OC	CQ304QI	YXX44XI	UX333X1	JX333XY	JQ2O3X4	RY3O3XI
	48	1RQQ3XI	ORQGGOI	TRQAJFL	ORQ434I	DRQQ38F	HRQQ38L	DRQQ5OU
	108	DR6AVJL	OAKK3KI	PR8ON8L	OP8888L	OQ66S8L	DR6658L	DR66S8L

Tabel 4.2 Hasil Presisi berdasarkan jumlah *hidden layer* masing-masing fitur

Citra	Fitur	Jumlah <i>Hidden Layer</i>						
		10	20	30	40	50	60	70
DR 4150 CR	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%
DR3629YD	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%
DR 3704 SM	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%
DK 5212 OC	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%
DR 2513 CG	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%
DR 6744 HM	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%

Citra	Fitur	Jumlah <i>Hidden Layer</i>						
		10	20	30	40	50	60	70
	108	12.50%	62.50%	62.50%	54.17%	62.50%	75.00%	75.00%
DR 5190 UA	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%
DR 2058 HP	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%
DR 6944 YC	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%
DR 6658 L	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%

Tabel 4.3 Waktu Pengolahan Data berdasar jumlah *hidden layer* masing-masing fitur dalam detik

Citra	Fitur	Jumlah <i>Hidden Layer</i>						
		10	20	30	40	50	60	70
DR 4150 CR	12	0,007950975	0,009313613	0,007218363	0,007251938	0,007209675	0,007414888	0,007390888
	48	0,01035745	0,007082875	0,00708025	0,007268225	0,007579225	0,007343938	0,008115638
	108	0,007945413	0,007387763	0,007620363	0,00708815	0,00748675	0,007762213	0,007720925
DR 3629 YD	12	0,007052038	0,007798263	0,00819375	0,008027888	0,0071802	0,008124738	0,007018063
	48	0,00736275	0,0071727	0,007279038	0,007246488	0,007108213	0,007253488	0,007706688
	108	0,007331988	0,007583525	0,007715075	0,0073822	0,0070376	0,007997338	0,00736255
DR 3704 SM	12	0,007187963	0,0072904	0,0087356	0,0070292	0,007406888	0,007407363	0,007266288
	48	0,007134438	0,007137738	0,007179375	0,007202063	0,007159863	0,007240963	0,007877188
	108	0,007356575	0,00722695	0,007848675	0,007242813	0,007497275	0,007624238	0,0072528
DK 5212 OC	12	0,007246025	0,0070143	0,008891738	0,007312988	0,007313175	0,007119175	0,007150575
	48	0,007305038	0,007227063	0,006984238	0,00701095	0,00785875	0,00703235	0,00727565
	108	0,0070592	0,007212575	0,007221563	0,007439538	0,009566725	0,007290275	0,00749095
DR 2513 CG	12	0,007301875	0,0074379	0,008832138	0,007364613	0,007223388	0,007064525	0,007231463
	48	0,00687755	0,007352113	0,011275425	0,007008463	0,0071727	0,007401863	0,007582063
	108	0,007301888	0,0073479	0,006879225	0,007291638	0,008910013	0,010902313	0,007974388

Citra	Fitur	Jumlah <i>Hidden Layer</i>						
		10	20	30	40	50	60	70
<b>DR 6744 HM</b>	12	0,007027475	0,007221438	0,008016363	0,0079413	0,007108213	0,006945863	0,007101688
	48	0,00699705	0,007258825	0,009607963	0,007512388	0,006983138	0,007207125	0,0076041
	108	0,009377075	0,00717255	0,0075143	0,007498763	0,008981913	0,00895665	0,00731725
<b>DR 5190 UA</b>	12	0,0096482	0,007273913	0,007168113	0,0070216	0,007197788	0,00711475	0,0071098
	48	0,009803413	0,007254563	0,009579688	0,007043413	0,007289363	0,00716815	0,011193825
	108	0,006964688	0,007305263	0,007457425	0,008799963	0,007611338	0,016197988	0,00755105
<b>DR 2058 HP</b>	12	0,008360875	0,007202175	0,007468525	0,007711688	0,00743555	0,007400538	0,006972125
	48	0,00717585	0,007002075	0,00774255	0,007285488	0,007892938	0,007420125	0,009219163
	108	0,007011738	0,007163525	0,007208313	0,007986038	0,007274975	0,009597238	0,007250263
<b>DR 6944 YC</b>	12	0,007217613	0,008995188	0,006883638	0,00706345	0,00718475	0,007166588	0,007190125
	48	0,006933288	0,007150125	0,007040675	0,007165213	0,008741125	0,007424663	0,008603413
	108	0,007825513	0,0072058	0,007501263	0,007388025	0,01007015	0,0112951	0,007539338
<b>DR 6658 L</b>	12	0,007258857	0,008566814	0,007195029	0,007146657	0,007233629	0,007397557	0,007366371
	48	0,007082443	0,006904557	0,007207871	0,007314886	0,008049086	0,007269329	0,007263343
	108	0,008131814	0,007188586	0,0071341	0,008986843	0,007947357	0,009499129	0,0076968

Berdasarkan data Tabel 4.2 yang menampilkan hasil presisi pengujian masing-masing citra, diperoleh presisi rata-rata dari masing-masing filter sesuai dengan jumlah *hidden layer*-nya adalah seperti pada Gambar 4.9 berikut:



Gambar 4.9 Nilai rata-rata presisi fitur berdasarkan jumlah *hidden layer*

Rata-rata nilai presisi pada Gambar 4.8 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, 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 grayscaling, 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:

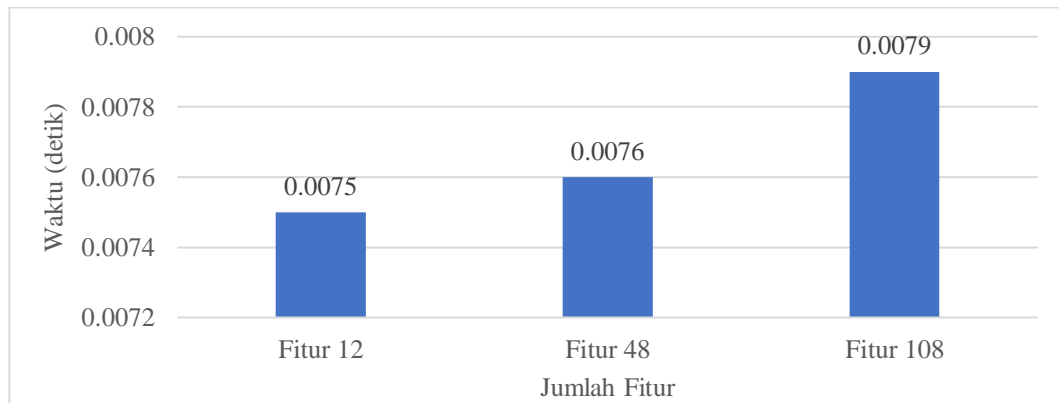
- Jumlah maksimum jangka waktu *training* data (1000)
- Performance goal (0)
- Jumlah maksimum validasi yang gagal (6)
- Jumlah minimum performa gradien (1e-07)
- Nilai awal  $\mu$  (0.001)
- Nilai penurunan  $\mu$  (0.1)
- Nilai kenaikan  $\mu$  (10)
- Nilai maksimum  $\mu$  (10000000000)
- Jangka waktu 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 (`trainrbp`)
- 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 (`traingdx`)

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 4.10 berikut:



Gambar 4.10 Waktu rata-rata pengolahan citra per fitur

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.



## BAB V PENUTUP

### 5.1 Kesimpulan

Berdasarkan paparan yang telah disampaikan pada bab-bab 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 pada 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. Waktu pengolahan data yang dibutuhkan sebanding dengan jumlah fitur yang digunakan, semakin banyak jumlah fitur maka jumlah waktu pengolahan data yang dibutuhkan semakin banyak pula.
6. Jumlah fitur mempengaruhi tingkat performa sistem, semakin tinggi jumlah fitur maka performa sistem semakin meningkat.
7. 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.

## 5.2 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.

## DAFTAR PUSTAKA

- [1] Dileep Gaurav, D., and Ramesh, Renu, 2012, *A Feature Extraction Technique Based on Character Geometry for Character Recognition*, New York: Cornell University Library.
- [2] Wang, Sheng, 2011, *A Review of Gradient-based and Edge-based Feature Extraction Methods for Object Detection*, IEEE International Conference on Computer and Information Technology, p277-282.
- [3] Prasad, K., C. Nigam, D., Lakhotiya, A., and Umre, D, 2013, *Character Recognition Using Matlab's Neural Network Toolbox*, International Journal of u- and e- Service, Science and Technology, IJUNESST, Vol. 6, No. 1, p.13-20.
- [4] M. Hanmandlu, K. R. Murali Mohan and H. Kumar, 1999, *Neural Based Handwritten Character Recognition*, Document Analysis and Recognition, ICDAR '99, Proceedings of the Fifth International Conference on, Bangalore, p.241-244.
- [5] Sri Widodo, Agus, 2014, *Sistem Deteksi Dan Pengenalan Karakter pada Plat Nomor Kendaraan Dengan Metode Backpropagation*, Yogyakarta.
- [6] Syuhada, Muhammad, 2015, *Realisasi Pengenalan Plat Nomor Kendaraan dengan Metode Histogram Citra dan Jaringan Syaraf Tiruan Backpropagation*, Bandar Lampung.
- [7] Putra, Darma, 2010, *Pengolahan Citra Digital*, p.19-20 p.303, C.V ANDI OFFSET, Yogyakarta.
- [8] Riadi, Muchlisin, 2016, *Pengolahan Citra Digital*, tersedia di [www.kajianpustaka.com/2016/04/pengolahan-citra-digital.html](http://www.kajianpustaka.com/2016/04/pengolahan-citra-digital.html), diakses 10 Desember 2016.
- [9] Basuki, Achmad, 2005, *Metode Numerik dan Algoritma Komputasi*, Yogyakarta.
- [10] Theodoridis, S., Koutroumbas, K., 2003, *Pattern Recognition*, 2nd Edition, Academic Press, New York, USA.

- [11] Jong Jek Siang, 2009, *Jaringan Syaraf Tiruan dan Pemrograman menggunakan MATLAB*, C.V ANDI OFFSET, Yogyakarta.
- [12] Widiarsono, Teguh. 2005. *Tutorial Praktis Belajar MATLAB*. Jakarta.