TUGAS AKHIR – KM184801

PENGENALAN PLAT NOMOR KENDARAAN BERESOLUSI RENDAH MENGGUNAKAN METODE LEARNING VECTOR QUANTIZATION

Shafira Herlinawati
NRP 0611160000013

Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT

DEPARTEMEN MATEMATIKA
Fakultas Sains Dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – KM184801

PENGENALAN PLAT NOMOR KENDARAAN BERESOLUSI RENDAH MENGGUNAKAN METODE LEARNING VECTOR QUANTIZATION

Shafira Herlinawati
NRP 0611160000013

Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT

DEPARTEMEN MATEMATIKA
Fakultas Sains Dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya 2020

FINAL PROJECT – KM184801

LOW-RESOLUTION VEHICLE LICENSE PLATE RECOGNITION USING LEARNING VECTOR QUANTIZATION METHOD

Shafira Herlinawati
NRP 0611160000013

Supervisor
Dr. Budi Setiyono, S.Si, MT

DEPARTMENT OF MATHEMATICS
Faculty Of Science And Data Analytics
Sepuluh Nopember Institute of Technology
Surabaya 2020

LEMBAR PENGESAHAN

PENGENALAN PLAT NOMOR KENDARAAN BERESOLUSI RENDAH MENGGUNAKAN METODE LEARNING VECTOR QUANTIZATION

LOW-RESOLUTION VEHICLE LICENSE PLATE RECOGNITION USING LEARNING VECTOR QUANTIZATION METHOD

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Matematika
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:
SHAFIRA HERLINAWATI
NRP. 06111640000013

Menyetujui,
Dosen Pembimbing

Dr. Budi Setiyono, S.Si, MT
NIP. 19720207 199702 1 001

Mengetahui
Kepala Departemen Matematika
FSAD ITS

Subchan, Ph.D
NIP. 19710513 199702 1 001
Surabaya, 19 Agustus 2020

PENGENALAN PLAT NOMOR KENDARAAN BERESOLUSI RENDAH MENGGUNAKAN METODE LEARNING VECTOR QUANTIZATION

Nama Mahasiswa	:	Shafira Herlinawati
NRP	:	06111640000013
Departemen	:	Matematika
Pembimbing	:	Dr. Budi Setiyono S.Si, MT

Abstrak

Seiring berjalannya waktu, perkembangan teknologi semakin meningkat. Salah satu yang mengalami perkembangan pesat adalah sistem pengawasan lalu lintas yang semakin canggih. Pengawasan lalu lintas diperlukan mengingat semakin meningkatnya jumlah kecelakaan kendaraan yang disebabkan oleh pelanggaran lalu lintas jalan. Salah satu bagian dari pengawasan lalu lintas adalah pengenalan plat nomor kendaraan yang mana memiliki tahap penting yaitu tahap pengenalan karakter. Dalam tahap pengenalan karakter ini akan dihasilkan suatu keluaran berupa karakter teks. Pada beberapa kasus, citra plat nomor yang akan dikenali memiliki resolusi rendah yang diakibatkan oleh beberapa faktor seperti spesifikasi kamera dan proses deteksi/segmentasi plat nomor. Oleh karena itu dalam tugas akhir ini akan dirancang suatu program yang dapat mengenali karakter plat nomor kendaraan beresolusi rendah dan mengonversinya ke teks dengan menggunakan metode Learning Vector Quantization. Uji coba dilakukan dengan menggunakan 25 *citra plat nomor beresolusi rendah yang terdiri dari* 174 *karakter. Hasil dari pengujian adalah didapatkan akurasi pengenalan karakter sebesar* 96.55% *dan akurasi pengenalan plat nomor sebesar* 88%.

Kata kunci : *learning vector quantization, pengenalan karakter, plat nomor kendaraan beresolusi rendah*

LOW-RESOLUTION VEHICLE LICENSE PLATE RECOGNITION USING LEARNING VECTOR QUANTIZATION METHOD

Name	:	Shafira Herlinawati
NRP	:	06111640000013
Department	:	Mathematics
Supervisor	:	Dr. Budi Setiyono S.Si, MT

Abstract

Over time, technological development has advanced a lot. One that is experiencing rapid development is an increasingly sophisticated traffic control system. Traffic control is needed due to the increasing number of vehicle accidents caused by road traffic violations. One of traffic control system is license plates recognition that have an important steps which is character recognition. The output of this process is a text character. In some cases, license plat image has a low-resolution because of many factors like camera specification and detection/segmentation process. Therefore, in this final project, the writer will design a program that can recognize the characters of a moving vehicle license plate and convert it into text using the Learning Vector Quantization method. Testing was done by using 25 images of low-resolution vehicle license plate that have 174 characters. The accuracy of character recognition is 96.55% *and the accuracy of license plat recognition is* 88%.

Keywords : *character recognition, learning vector quantization, low-resolution vehicle license plate*

KATA PENGANTAR

Puji syukur kehadirat Allah SWT karena atas rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan dengan baik tugas akhir yang berjudul

"PENGENALAN PLAT NOMOR KENDARAAN BERESOLUSI RENDAH MENGGUNAKAN METODE LEARNING VECTOR QUANTIZATION"

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FSAD ITS Surabaya. Tugas akhir ini terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Ayah Drs. H. Mahfud dan Ibu Almh. Siti M. Azizah selaku orangtua yang selalu memberikan semangat, doa, serta dukungan kepada penulis.

2. Bapak Subchan, Ph. D selaku Kepala Departemen Matematika FSAD ITS.

3. Bapak Drs. Suhud Wahyudi, M.Si selaku dosen wali yang telah memberi arahan akademik selama penulis menempuh pendidikan di Departemen Matematika FSAD ITS.

4. Bapak Dr. Budi Setiyono, S.Si, MT selaku dosen pembimbing atas segala bimbingan serta motivasinya kepada penulis dalam mengerjakan tugas akhir ini.

5. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si, MT, Bapak Drs. Daryono Budi Utomo, M.Si, Bapak Dr. Imam Mukhlas, S.Si, MT, selaku dosen penguji atas kritikan dan saran yang diberikan.

6. Bapak dan Ibu dosen serta para staff Departemen Matematika FSAD ITS yang tidak bisa penulis sebutkan satu persatu.

7. Mochammad Habibi Mahfud dan Deswinta selaku saudara penulis dan seluruh keluarga besar yang selalu memberikan motivasi kepada penulis.

8. Aldiansyah Ramadlan yang selalu membantu, memotivasi, mendukung dan menginspirasi dalam pengerjaan tugas akhir ini.

9. Firman, Yuda, Dila, Ario, Dyah, Lutvi, Ferisa, dan Novia sebagai teman seperjuangan dalam penyelesaian tugas akhir ini.

10. Nanda, Ratna, Fikry, Yohana, Aurel, Vira, Clarita dan teman-teman lain yang selalu memberi semangat dan motivasi kepada penulis.

11. Semua pihak yang tidak bisa disebutkan satu persatu. Terimakasih atas doa dan dukungan kepada penulis sampai dengan selesainya tugas akhir ini.

Penulis menyadari bahwa dalam tugas akhir ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan tugas akhir ini. Semoga tugas akhir ini dapat membawa manfaat bagi banyak pihak.

Surabaya, Juni 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL i

HALAMAN JUDUL iii

LEMBAR PENGESAHAN vii

ABSTRAK ix

ABSTRACT xi

KATA PENGANTAR xiii

DAFTAR ISI xv

DAFTAR GAMBAR xix

DAFTAR TABEL xxiii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Permasalahan 4

1.3 Batasan Masalah 5

1.4 Tujuan 5

1.5 Manfaat 5

BAB II TINJAUAN PUSTAKA 7

2.1 Posisi Penelitian 7

2.2 Plat Nomor 12

2.3 Pengolahan Citra Digital 13

2.4 Citra Resolusi Rendah 14

2.5 *Optical Character Recognition* 14

2.6 Jaringan Syaraf Tiruan 15

2.7 *Learning Vector Quantization* 15

BAB III METODE PENELITIAN 19
3.1 Studi literatur 20
3.2 Perancangan Program 20
3.3 Simulasi Program 22
3.4 Pengujian dan Evaluasi Sistem 22
3.5 Penarikan Kesimpulan dan Penyusunan Laporan Tugas Akhir 23

BAB IV ANALISIS DAN PERANCANGAN SISTEM 25
4.1 Analisis Sistem 25
4.1.1 Analisis Fungsional Sistem 25
4.1.2 Analisis non-Fungsional Sistem 25
4.2 Perancangan Sistem 26
4.2.1 Perancangan Data 26
4.2.2 Perancangan Proses 28
4.2.3 Perancangan Antar Muka 55

BAB V UJI COBA DAN PEMBAHASAN 59
5.1 Data Uji Coba 59
5.1.1 Data Uji Coba Proses *Training* 59
5.1.2 Data Uji Coba Proses *Testing* 59
5.2 Hasil Uji Coba 59
5.2.1 Uji Coba Proses *Training* 59
5.2.2 Uji Coba Proses Pengujian 62
5.3 Pembahasan Hasil Uji Coba 67
5.3.1 Pembahasan Tingkat Akurasi Sistem . . 67
5.3.2 Pembahasan Mengenai Tahap Tanpa *Preprocessing* Tertentu 70
5.3.3 Pembahasan Mengenai Pengaruh *maxEpoch* dan *Learning Rate* 72
5.3.4 Pembahasan Mengenai Pengujian Terhadap Plat Nomor Berbeda Warna 78

BAB VI PENUTUP 81
6.1 Kesimpulan 81
6.2 Saran 81
LAMPIRAN 83
DAFTAR PUSTAKA 103
BIODATA PENULIS 107

DAFTAR GAMBAR

Gambar 2.1 Perbandingan citra berdasarkan resolusi, (a) Citra beresolusi tinggi, (b) Citra beresolusi rendah. 14
Gambar 2.2 Arsitektur *Learning Vector Quantization*[13] 17

Gambar 3.1 Blok diagram pengerjaan tugas akhir . . 19
Gambar 3.2 Blok diagram proses *training* 23
Gambar 3.3 Blok diagram proses *testing* 24

Gambar 4.1 Proses *Sharpening*, (a) Citra asli, (b) Citra hasil *sharpening* 30
Gambar 4.2 Citra RGB dengan diambil sebagian pikselnya 31
Gambar 4.3 Hasil proses *grayscalling* 32
Gambar 4.4 Hasil *deblurring*, (a) Citra *grayscale*, (b) Citra hasil *deblurring* 34
Gambar 4.5 Hasil binerisasi citra *deblurring*, (a) Citra biner, (b) Citra biner hasil *deblurring* 35
Gambar 4.6 Citra beserta histogramnya, (a) citra *grayscale*, (b) histogram 37
Gambar 4.7 Eksperimen nilai *threshold*, (a) $T = 100$, (b) $T = 140$, (c) $T = 170$, (d) $T = 200$ 37
Gambar 4.8 Hasil proses binerisasi 38
Gambar 4.9 Contoh proses erosi pada citra biner . . . 39
Gambar 4.10 Contoh proses dilasi pada citra biner . . 40

Gambar 4.11 Proses erosi dan dilasi, (a) Citra biner, (b) Citra hasil erosi, (c) Citra hasil dilasi 40
Gambar 4.12 Proses *draw contour*, (a) Citra hasil operasi morfologi, (b) Citra hasil *draw contour* 43
Gambar 4.13 Hasil *bounding box* objek-objek yang terdeteksi 44
Gambar 4.14 Hasil *bounding box* objek-objek karakter 45
Gambar 4.15 Hasil proses *cropping* menjadi citra per-karakter 45
Gambar 4.16 Interpolasi Bilinear 46
Gambar 4.17 Proses normalisasi karakter, (a) Citra per-karakter hasil segmentasi (b) Citra per-karakter hasil normalisasi karakter . 47
Gambar 4.18 *Preprocessing* karakter, (a) Citra per-karakter hasil *sharpening* (b) Citra per-karakter hasil *grayscalling*, (c) Citra per-karakter hasil binerisasi 48
Gambar 4.19 Proses pengecekan piksel pada ekstraksi ciri 48
Gambar 4.20 Arsitektur *Learning Vector Quantization* 55
Gambar 4.21 Tampilan awal program 56
Gambar 4.22 Tampilan halaman *training* 56
Gambar 4.23 Tampilan halaman *testing* 57

Gambar 5.1 Proses *Training* 62
Gambar 5.2 Tampilan halaman proses *training* 63
Gambar 5.3 Tampilan halaman proses *testing* 67
Gambar 5.4 Proses *Testing* 67
Gambar 5.5 Diagram pengaruh *Learning Rate* terhadap akurasi pengenalan karakter . 74

Gambar 5.6 Diagram pengaruh *Learning Rate* terhadap rata-rata waktu *training* per-citra 75
Gambar 5.7 Diagram pengaruh *MaxEpoch* terhadap akurasi pengenalan karakter . 77
Gambar 5.8 Diagram pengaruh *MaxEpoch* terhadap rata-rata waktu *training* per-citra 77

DAFTAR TABEL

Tabel 2.1 Posisi Penelitian 9

Tabel 4.1 Spesifikasi *software* dan *hardware* yang digunakan. 26
Tabel 4.2 Data proses 27
Tabel 4.3 Hasil uji coba Metode Wiener 34

Tabel 5.1 Hasil uji coba 68
Tabel 5.2 Hasil uji coba tanpa *preprocessing* tertentu 70
Tabel 5.3 Pengaruh *Learning Rate*pada Kondisi A terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra . . . 73
Tabel 5.4 Pengaruh *Learning Rate* pada Kondisi B terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra . . . 74
Tabel 5.5 Pengaruh *MaxEpoch* pada Kondisi A terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra . . . 76
Tabel 5.6 Pengaruh *MaxEpoch* pada Kondisi B terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra . . . 76
Tabel 5.7 Hasil uji coba plat nomor berbeda warna. 78

BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu, perkembangan teknologi semakin meningkat. Salah satu yang mengalami perkembangan pesat adalah sistem pengawasan lalu lintas yang semakin canggih. Pengawasan lalu lintas diperlukan mengingat semakin meningkatnya jumlah kecelakaan kendaraan yang disebabkan oleh pelanggaran lalu lintas jalan. Berdasarkan data dari Polri, terjadi 107.500 kecelakaan lalu lintas pada tahun 2019, jumlah tersebut meningkat 3 persen dari tahun 2018, yaitu sebanyak 103.672 kecelakaan[1]. Oleh karena itu, sistem transportasi cerdas menjadi sangat penting untuk dikembangkan. Pengawasan lalu lintas dapat digunakan untuk memantau lalu lintas jalan, melaporkan kejadian lalu lintas, merekam informasi kendaraan, dan sebagainya. Salah satu bentuk pengawasan lalu lintas adalah pengenalan plat nomor kendaraan. Sistem pengenalan plat nomor kendaraan merupakan suatu sistem yang menggantikan fungsi penglihatan manusia dalam hal mengenali plat nomor kendaraan bermotor.

Salah satu tahap penting dalam pengenalan plat nomor adalah tahap pengenalan karakter. Dalam tahap pengenalan karakter ini akan dihasilkan suatu keluaran berupa karakter teks. Teks hasil pengenalan inilah yang dapat digunakan sebagai *query* untuk mengambil informasi di *database*, sehingga dapat diimplementasikan dalam sistem pengawasan lalu lintas. Pada beberapa kasus, citra plat nomor yang akan dikenali memiliki resolusi rendah yang dapat diakibatkan

oleh beberapa faktor seperti spesifikasi kamera dan proses deteksi/segmentasi plat nomor. Citra yang memiliki resolusi rendah dapat diartikan sebagai citra yang memiliki kerapatan piksel yang rendah[2]. Pada citra plat nomor beresolusi rendah, terdapat beberapa gangguan seperti *noise*, kabur (*blur*), serta beberapa karakter akan terlihat berdekatan (menempel satu sama lain). Selain memiliki resolusi rendah, objek plat nomor yang didapat dari proses hasil deteksi video memiliki efek *motion blur*, karena pergeseran posisi rincian objek pada gambar sehingga tampilan tekstur dan tepian dari objek gambar menjadi tidak jelas[3], sehingga sebelum memasuki tahap pengenalan diperlukan beberapa tahap pengolahan citra digital tertentu seperti *sharpening* dan operasi morfologi.

Berkaitan dengan permasalahan di atas, maka diperlukan suatu penelitian untuk mengenali plat nomor kendaraan beresolusi rendah menjadi sebuah teks sehingga dapat memberikan data yang berguna bagi pihak yang membutuhkan dalam upaya mengurangi tingkat pelanggaran lalu lintas jalan dan kecelakaan di Indonesia. Beberapa penelitian yang telah dilakukan untuk pengenalan plat nomor diantaranya yaitu penelitian yang dilakukan oleh Muhammad Tahir Qadri, dkk pada tahun 2009 dengan menggunakan *Optical Character Recognition* yang bertujuan untuk membuat suatu sistem yang dapat mengenali karakter suatu plat nomor secara otomatis. Sistem *Automatic Number Plate Recognition* (ANPR) bekerja dalam tiga tahap, pertama adalah tahap deteksi dan penangkapan gambar, tahap kedua yaitu deteksi dan ekstraksi dari gambar plat nomor, dan tahap ketiga menggunakan *image segmentation technique* untuk mendapatkan karakter individual dan *correlation method* untuk mengenali karakter individual dengan membantu pengambilan data dalam database untuk setiap alfabet[4].

Lalu penelitian juga dilakukan oleh Edi Sugiarto pada tahun 2013 yang menggunakan metode *Support Vector Machine* (SVM) dalam pengenalan karakter plat nomor. Metode penelitian dalam penelitian ini dibagi menjadi dua bagian, yakni metode pada proses pelatihan (*training*) dan proses pengenalan (*recognition*). Dalam proses pengenalan tahap yang dilakukan meliputi tahap *preprocessing* dengan metode binerisasi citra, segmentasi dengan metode *connected component labeling*, kemudian pada tahap ekstrasi fitur dilakukan pemetaan citra dalam matriks 5×5 dan pada tahap klasifikasi diterapkan menggunakan metode SVM dengan pengujian dan pengukuran akurasinya menggunakan *confusion matrix*[5]. Adapun penelitian pengenalan plat nomor menggunakan metode *Learning Vector Quantization* juga telah beberapa kali dilakukan. Salah satunya yaitu penelitian yang dilakukan oleh Anuja P. Nagare pada tahun 2011 yang bertujuan untuk membandingkan hasil dari pengenalan yang dilakukan dengan metode *Back Propagation* dan *Learning Vector Quantization*. Penelitian dilakukan dengan menggunakan masukan berupa citra tampak depan mobil, kemudian citra tersebut akan memasuki tahap *preprocessing* dan dilakukan ekstraksi, hingga tahap akhir yaitu melakukan pengenalan karakter dengan menggunakan dua metode, yaitu *Back Propagation* dan *Learning Vector Quantization*. Hasil dari penelitian ini adalah pengenalan menggunakan metode *Back Propagation* menghasilkan akurasi sebesar 66,67% dan pengenalam menggunakan metode *Learning Vector Quantization* menghasilkan akurasi sebesar 94,44%[6]. Kemudian penelitian juga dilakukan oleh Achmad Solichin, dkk pada tahun 2015 yang bertujuan untuk merancang suatu aplikasi berbasis *mobile* yang dapat mempermudah penggunanya dalam mengakses informasi kendaraan berdasarkan plat nomor kendaraan tersebut.

Proses identifikasi dilakukan dengan memanfaatkan kamera yang tersedia di perangkat *mobile* dan menggunakan metode pengolahan citra digital. Selanjutnya dilakukan proses segmentasi, ekstraksi ciri dan pengenalan karakter. Untuk mengenali karakter pada nomor kendaraan, dilakukan proses klasifikasi menggunakan metode *Learning Vector Quantization* (LVQ). Hasil dari penelitian ini adalah akurasi rata-rata sebesar 95,32%[7]. Selanjutnya penelitian dilakukan oleh Beryl Labique Ahmadie, dkk pada tahun 2018 dengan menggunakan metode *Learning Vector Quantization* untuk melakukan pengenalan plat nomor mobil. Langkah awal yaitu dengan melakukan deteksi plat nomor dari sebuah citra digital, kemudian segmentasi karakter pada potongan citra plat nomor yang sudah didapat sebelumnya menggunakan algoritma *connected component*. Langkah terakhir yaitu adalah pengenalan karakter dengan menggunakan metode *Learning Vector Quantization*. Berdasarkan hasil pengujian yang dilakukan diperoleh akurasi tertinggi sebesar 94% pada proses deteksi plat nomor dan pada proses pengenalan karakter menggunakan metode LVQ diperoleh akurasi tertinggi sebesar 86,67%[8]. Berdasarkan hasil penelitian tersebut maka penulis memilih untuk menggunakan metode *Learning Vector Quantization* sebagai metode untuk mengenali karakter plat nomor kendaraan beresolusi rendah.

1.2 Rumusan Permasalahan

Berdasarkan latar belakang tersebut disusun rumusan masalah sebagai berikut:

1. Bagaimana melakukan pengenalan plat nomor kendaraan beresolusi rendah dan mengonversinya ke dalam teks menggunakan metode *Learning Vector Quantization*?

2. Bagaimana mengetahui akurasi pengenalan karakter

plat nomor kendaraan beresolusi rendah meggunakan metode *Learning Vector Quantization*?

1.3 Batasan Masalah

Batasan masalah pada Tugas Akhir ini adalah sebagai berikut:

1. Data yang digunakan adalah gambar plat nomor kendaraan beresolusi rendah yang memiliki ukuran panjang sekitar $63 - 86$ piksel dan lebar sekitar $16 - 24$ piksel dari penelitian lain dengan presentase 80% untuk *training* dan 20% untuk *testing*.

2. Plat nomor yang akan dikenali berdasarkan karakter yang dapat dibaca secara visual.

1.4 Tujuan

Tujuan pada Tugas Akhir ini adalah sebagai berikut:

1. Melakukan pengenalan plat nomor kendaraan beresolusi rendah dan mengonversinya ke dalam teks menggunakan metode *Learning Vector Quantization*.

2. Mengetahui akurasi pengenalan karakter plat nomor kendaraan beresolusi rendah dengan meggunakan metode *Learning Vector Quantization*

1.5 Manfaat

Manfaat pada Tugas Akhir ini adalah sebagai berikut :

1. Dihasilkan sebuah program yang bisa mengenali karakter plat nomor kendaraan beresolusi rendah dan mengonversinya menjadi teks.

2. Dapat membantu pihak yang membutuhkan untuk mendapatkan data berupa teks plat nomor kendaraan beresolusi rendah.

BAB II
TINJAUAN PUSTAKA

2.1 Posisi Penelitian

Penelitian tentang pengenalan plat nomor kendaraan sebelumnya telah diteliti oleh beberapa peneliti yaitu Muhammad Tahir Qadri, dkk pada tahun 2009 dengan menggunakan *Optical Character Recognition* yang bertujuan untuk membuat suatu sistem yang dapat mengenali karakter suatu plat nomor secara otomatis. Sistem *Automatic Number Plate Recognition* (ANPR) bekerja dalam tiga tahap, pertama adalah tahap deteksi dan penangkapan gambar, tahap kedua yaitu deteksi dan ekstraksi dari gambar plat nomor. Tahap ketiga menggunakan *image segmentation technique* untuk mendapatkan karakter individual dan *correlation method* untuk mengenali karakter individual dengan membantu pengambilan data dalam database untuk setiap alfabet[4].

Kemudian penelitian juga dilakukan oleh Edi Sugiarto pada tahun 2013 yang menggunakan metode *Support Vector Machine* (SVM) dalam pengenalan karakter plat nomor. Metode penelitian dalam penelitian ini dibagi menjadi dua bagian yakni metode pada proses pelatihan (*training*) dan proses pengenalan (*recognition*), dalam proses pengenalan tahap yang dilakukan meliputi tahap *preprocessing* dengan metode binerisasi citra, segmentasi dengan metode *connected component labeling*, pada tahap ekstrasi fitur dilakukan pemetaan citra dalam matriks 5×5 dan pada tahap klasifikasi diterapkan menggunakan metode SVM dengan pengujian dan pengukuran akurasinya menggunakan *confusion matrix*[5].

Adapun penelitian pengenalan plat nomor

menggunakan metode *Learning Vector Quantization* juga telah beberapa kali dilakukan. Salah satunya yaitu penelitian yang dilakukan oleh Anuja P. Nagare pada tahun 2011 yang bertujuan untuk membandingkan hasil dari pengenalan yang dilakukan dengan metode *Back Propagation* dan *Learning Vector Quantization.* Penelitian dilakukan dengan menggunakan masukan berupa citra tampak depan mobil, kemudian citra tersebut akan memasuki tahap *preprocessing* dan dilakukan ekstraksi, hingga tahap akhir yaitu melakukan pengenalan karakter dengan menggunakan dua metode, yaitu *Back Propagation* dan *Learning Vector Quantization.* Hasil dari penelitian ini adalah pengenalan menggunakan metode *Back Propagation* menghasilkan akurasi sebesar 66,67% dan pengenalam menggunakan metode *Learning Vector Quantization* menghasilkan akurasi sebesar 94,44%[6].

Kemudian penelitian juga dilakukan oleh Achmad Solichin, dkk pada tahun 2015 yang bertujuan untuk merancang suatu aplikasi berbasis mobile yang dapat mempermudah penggunanya dalam mengakses informasi kendaraan berdasarkan plat nomor kendaraan tersebut. Proses identifikasi dilakukan dengan memanfaatkan kamera yang tersedia di perangkat mobile dan menggunakan metode pengolahan citra digital. Selanjutnya dilakukan proses segmentasi, ekstraksi ciri dan pengenalan karakter. Untuk mengenali karakter pada nomor kendaraan, dilakukan proses klasifikasi menggunakan metode *Learning Vector Quantization* (LVQ). Hasil dari penelitian ini adalah akurasi rata-rata sebesar 95,32%[7].

Selanjutnya penelitian dilakukan oleh Beryl Labique Ahmadie, dkk pada tahun 2018 dengan menggunakan metode *Learning Vector Quantization* untuk melakukan pengenalan plat nomor mobil. Langkah awal yaitu dengan melakukan deteksi plat nomor dari sebuah citra digital,

kemudian segmentasi karakter pada potongan citra plat nomor yang sudah didapat sebelumnya menggunakan algoritma *connected component*. Langkah terakhir yaitu adalah pengenalan karakter dengan menggunakan metode *Learning Vector Quantization*. Berdasarkan hasil pengujian yang dilakukan diperoleh akurasi tertinggi sebesar 94% pada proses deteksi plat nomor dan pada proses pengenalan karakter menggunakan metode LVQ diperoleh akurasi tertinggi sebesar 86,67%[8]. Adapun beberapa penelitian di atas apabila dibuat tabulasi dapat dilihat pada Tabel 2.1.

Tabel 2.1: Posisi Penelitian

Penulis	Judul Penelitian	Tahun	Pokok Bahasan	Riset yang Dikerjakan
Muhammad Tahir Qadri, Muhammad Asif	*Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition*	2009	Sistem *Automatic Number Plate Recognition* (ANPR) bekerja dalam tiga tahap, pertama adalah tahap deteksi dan penangkapan gambar, tahap kedua yaitu deteksi dan ekstraksi dari gambar plat nomor. Tahap ketiga menggunakan *image segmentation technique* untuk mendapatkan karakter individual dan *correlation method* untuk mengenali karakter individual dengan membantu pengambilan data dalam database untuk setiap alfabet.	Penelitian ini menggunakan metode OCR dan objek berupa gambar mobil statis yang diambil melalui kamera, sedangkan penelitian yang dikerjakan penulis menggunakan metode LVQ dan objek berupa gambar beresolusi rendah hasil deteksi video.

Lanjutan **Tabel 2.1**

Edi Sugiarto	Pengenalan Karakter pada Plat Nomor Kendaraan Berbasis *Support Vector Machine*	2013	Metode penelitian dalam penelitian ini dibagi menjadi proses pelatihan dan proses pengenalan, dalam proses pengenalan tahap yang dilakukan meliputi tahap *preprocessing* dengan metode binerisasi citra, segmentasi, ekstraksi fitur dan pada tahap klasifikasi diterapkan menggunakan metode SVM.	Penelitian ini menggunakan metode SVM dan objek berupa gambar plat nomor yang diambil dengan kamera, sedangkan penelitian yang dikerjakan penulis menggunakan metode LVQ dan objek berupa gambar beresolusi rendah hasil deteksi video.
Anuja P. Nagare	License Plate Character Recognition System using Neural Network	2011	Penelitian ini bertujuan untuk membandingkan hasil pengenalan yang dilakukan dengan metode *Back Propagation* dan LVQ. Penelitian dilakukan dengan menggunakan masukan berupa citra tampak depan mobil, kemudian memasuki tahap *preprocessing* dan dilakukan ekstraksi, hingga tahap akhir yaitu melakukan pengenalan karakter dengan menggunakan dua metode di atas. Hasil dari penelitian ini adalah akurasi metode BP sebesar 66,67% dan akurasi metode LVQ sebesar 94,44%	Penelitian ini menggunakan metode LVQ dan BP serta objek berupa gambar mobil statis yang diambil melalui kamera, sedangkan penelitian yang dikerjakan penulis menggunakan metode LVQ dan objek berupa gambar beresolusi rendah hasil deteksi video.

Lanjutan **Tabel 2.1**

Achmad Solichin, Zulfikar Rahman	Identifikasi Plat Nomor Kendaraan Berbasis *Mobile* dengan Metode *Learning Vector Quantization*	2015	Proses identifikasi dilakukan dengan memanfaatkan kamera yang tersedia di perangkat *mobile*. Selanjutnya dilakukan proses segmentasi, ekstraksi ciri dan pengenalan karakter. Untuk mengenali karakter pada nomor kendaraan, dilakukan proses klasifikasi menggunakan metode *Learning Vector Quantization* (LVQ). Hasil dari penelitian ini adalah akurasi rata-rata sebesar 95,32%	Penelitian ini menggunakan objek berupa gambar kendaraan statis yang diambil melalui kamera *handphone* dengan resolusi tinggi, sedangkan penelitian yang dikerjakan penulis menggunakan objek berupa gambar beresolusi rendah hasil deteksi video.
Beryl Labique Ahmadie, Agus Wahyu Widodo, Fitri Utaminingrum	Pengenalan Plat Nomor Mobil Menggunakan Metode *Learning Vector Quantization*	2018	Langkah awal yaitu dengan melakukan deteksi plat nomor dari sebuah citra digital, kemudian segmentasi karakter pada potongan citra plat nomor yang sudah didapat sebelumnya. Langkah terakhir yaitu adalah pengenalan karakter dengan menggunakan metode *Learning Vector Quantization*. Berdasarkan hasil pengujian yang dilakukan diperoleh akurasi tertinggi sebesar 94% pada proses deteksi plat nomor dan pada proses pengenalan karakter menggunakan metode LVQ diperoleh akurasi tertinggi sebesar 86,67%	Penelitian ini menggunakan objek berupa gambar mobil statis yang diambil melalui kamera dengan resolusi tinggi, sedangkan penelitian yang dikerjakan penulis menggunakan objek berupa gambar beresolusi rendah hasil deteksi video.

2.2 Plat Nomor

Setiap kendaraan bermotor harus memiliki plat nomor atau yang sering disebut dengan TNKB (Tanda Nomor Kendaraan Bermotor). Berdasarkan pasal 68 dalam UU No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, menjelaskan bahwa kendaraan bermotor wajib menggunakan TNKB yang mmenuhi syarat bentuk, ukuran bahan, warna dan cara pemasangan. Spesifikasi TNKB tersebut adalah sebagai berikut: [9]

1. Berbentuk plat aluminium dengan cetakan tulisan dan baris. Baris pertama menunjukkan: kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf). Baris kedua menunjukkan bulan dan tahun masa berlaku, masing-masing dua digit (misalnya 02.21 berarti berlaku hingga Februari 2021).

2. Bahan baku TNKB adalaha aluminium dengan ketebalan 1 mm. Ukuran TNKB untuk kendaraan bermotr roda 2 dan roda 3 adalah 250 $\times$ 105 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah 450 $\times$ 130 mm.

3. Terdapat garis putih di sekitar TNKB dan tidak ada batas pemisah antara nomor polisi dan masa berlaku (dari tahun 2011).

4. Pada pertengahan 2014 terjadi perubahan tampilan. Plat nomor kini sedikit diperpanjang dari ukuran semula (untuk roda empat). Selain itu, terdapat perubahan posisi lambing Polantas dan tulisan “Korlantas Polri”, yakni, lambing Polantas kini berada di sudut kiri atas dan kanan bawah, sedangkan tulisan “Korlantas Polri” berada pada sudut kiri bawah dan kanan atas.

Selain itu dalam Perkapolri No. 5 tahun 2012 juga disebutkan mengenai warna TNKB, yaitu sebagai berikut:

1. Dasar hitam, tulisan putih untuk kendaraan bermotor perseorangan dan kendaraan bermotor sewa.
2. Dasar kuning, tulisan hitam untuk kendaraan bermotor umum.
3. Dasar merah, tulisan putih untuk kendaraan bermotor pemerintahan.
4. Dasar putih, tulisan biru untuk kendaraan bermotor Korps Diplomatik negara asing.
5. Dasar hijau, tulisan hitam untuk kendaraan bermotordi kawasan perdagangan bebas (*Free Trade Zone*) yang mendapatkan fasilitas pembebasan bea masuk dan berdasarkan Peraturan Menteri Keuangan, bahwa kendaraan bermotor tidak boleh dioperasionalkan atau dimutasikan ke wilayah Indonesia lainnya.

2.3 Pengolahan Citra Digital

Citra sebagai luaran suatu sistem perekaman data dapat bersifat *optic* berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Suatu citra digital adalah fungsi intensitas 2 dimensi $f(x, y)$, dimana x dan y adalah koordinat spasial dan f pada titik (x, y) merupakan tingkat kecerahan (*brightness*) suatu citra pada suatu titik[10].

Pengolahan citra digital adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Pengolahan citra dilakukan untuk memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau mesin (dalam hal ini komputer)[11].

2.4 Citra Resolusi Rendah

Citra merupakan salah satu komponen dari *multimedia* yang memegang peranan penting karena mengandung informasi dalam bentuk visual. Akan tetapi tidak semua citra memiliki kualitas yang bagus, sehingga citra tersebut tidak bisa menghasilkan informasi yang lebih jelas. Hal ini biasanya terjadi pada citra yang memiliki resolusi redah, citra seperti ini memiliki gangguan seperti bintik pada citra (*noise*), kabur (*blur*), dan lain-lain. Citra resolusi rendah dapat diartikan sebagai citra yang memiliki kerapatan piksel yang rendah[2]. Citra yang memiliki resolusi rendah memerlukan tahap *preprocessing* terlebih dahulu agar dapat diinterpretasi dengan baik. Pada Gambar 2.1 terlihat contoh citra plat nomor yang memiliki resolusi tinggi dan citra plat nomor yang memiliki resolusi rendah.

Gambar 2.1: Perbandingan citra berdasarkan resolusi, (a) Citra beresolusi tinggi, (b) Citra beresolusi rendah.

2.5 *Optical Character Recognition*

Optical Character Recognition (OCR) atau pengenalan pola adalah sebuah proses untuk mengonversi dokumen cetak atau tulisan tangan hasil scan menjadi karakter ASCII (karakter yang dapat dibaca oleh mesin)[12] atau dengan kata lain proses untuk mengonversi citra dokumen teks menjadi teks berbentuk digital sehingga dapat dilakukan perubahan (*edit*). Pengenalan pola identik dengan pengelompokan atau

klasifikasi karakter. Komputer menerima masukkan berupa objek citra yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra. Contoh pengenalan pola misalnya pada gambar berupa tulisan tangan yang digunakan sebagai data masukan untuk mengenali karakter 'S'. Dengan menggunakan suatu algoritma atau metode pengenalan pola diharapkan komputer dapat mengenali bahwa karakter tersebut adalah 'S'.

2.6 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) adalah sistem pemrosesan informasi yang memiliki karakteristik unjuk kerja tertentu yang menyerupai jaringan syaraf biologis[13]. JST telah dikembangkan sebagai generalisasi model matematika dari aspek kognitif manusia atau syaraf biologis, yaitu didasarkan pada asumsi-asumsi bahwa:

a. Pemrosesan informasi terjadi pada elemen-elemen yang disebut neuron.

b. Sinyal-sinyal merambat di antara enuron melalui interkoneksi.

c. Setiap interkoneksi memiliki bobot yang bersesuaian yang pada kebanyakan jaringan syaraf berfungsi untuk mengalikan sinyal yang dikirim.

d. Setiap neuron menerapkan fungsi aktivasi (biasanya tidak linear) pada masukan jaringan untuk menentukan sinyal keluarannya.

2.7 *Learning Vector Quantization*

Learning Vector Quantization (LVQ) adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan

secara otomatis belajar untuk mengklarifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika 2 vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vector tersebut ke dalam kelas yang sama[13]. Langkah-langkah dari algoritma ini adalah:

1. Tetapkan Bobot Awal (w) yaitu nilai piksel dari karakter, Maksimum *Epoch* (*MaxEpoch*), dan *Learning Rate* (α).

2. Masukkan:

   a. x yaitu vektor pelatihan $(x_1, \ldots, x_i, \ldots, x_n)$

   b. Target T

3. Tetap kondisi awal $Epoch = 0$

4. Kerjakan jika: ($Epoch < MaxEpoch$)

   a. $Epoch = Epoch + 1$

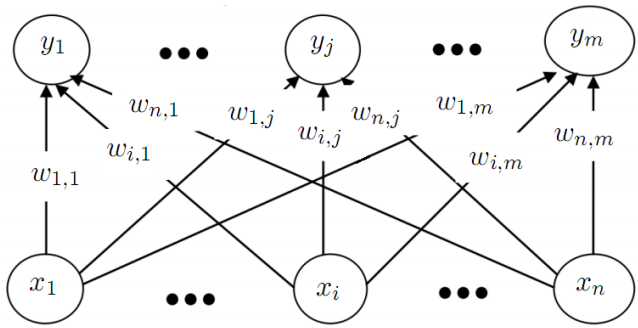
   b. Kerjakan untuk $i = 1$ hingga n

      - Tentukan j sedemikian hingga $d(x_i, w_{i,j})$ minimum (jarak Euclidian)
      - Perbaiki $w_{i,j}$ dengan ketentuan:
        i. Jika $T = y_j$ maka $w_{i,j}(\text{baru}) = w_{i,j}(\text{lama}) + \alpha(x_i - w_{i,j}(\text{lama}))$
        ii. Jika $T \neq y_j$ maka $w_j(\text{baru}) = w_{i,j}(\text{lama}) - \alpha(x_i - w_{i,j}(\text{lama}))$

   c. Kurangi nilai α

5. Selesai.

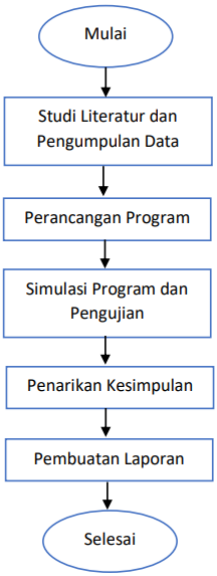
Gambar 2.2 berikut adalah gambar dari arsitektur LVQ:



Gambar 2.2: Arsitektur *Learning Vector Quantization*[13]

BAB III
METODE PENELITIAN

Dalam bab ini akan dijelaskan mengenai metodologi sistem yang digunakan untuk menyelesaikan tugas akhir ini agar dapat mencapai tujuan yang telah ditetapkan sebelumnya. Secara umum pengejaan tugas akhir ini berjalan seperti blok diagram pada Gambar 3.1.



Gambar 3.1: Blok diagram pengerjaan tugas akhir

3.1 Studi literatur

Pada tahap ini dilakukan pengumpulan data yang diperlukan untuk pengerjaan tugas akhir ini. Data yang digunakan dalam tugas akhir ini berupa citra plat nomor kendaraan beresolusi rendah hasil deteksi video. Citra plat nomor yang digunakan adalah citra plat nomor yang diambil dengan kondisi cuaca yang cerah dan dapat dibaca secara visual. Kemudian dilakukan identifikasi permasalahan dan pemahaman teori dengan mencari dan mengupulkan referensi yang menunjang penelitian mengenai *Learning Vector Quantization*. Referensi yang digunakan adalah buku-buku literatur, jurnal ilmiah, tugas akhir yang berkaitan dengan permasalahan yang dibahas serta artikel dari internet yang relevan.

3.2 Perancangan Program

Pada tahap ini akan dilakukan perancangan program. Program dibuat menggunakan bahasa pemograman java. Tahapan perancangan program terdiri atas 3 tahap, yaitu tahap akusisi, tahap *preprocessing*, dan tahap Jaringan Syaraf Tiruan (JST) dengan menggunakan metode *Learning Vector Quantization*. Proses perancangan program disajikan menggunakan blok diagram pada Gambar 3.2 dan Gambar 3.3 serta akan diberikan penjelasan pada setiap tahapan.

1. Tahap Akuisisi
   Tahap akuisisi merupakan tahap awal dari sistem, yaitu memasukkan data masukan ke sistem. Tahap akuisisi pada proses *training* menggunakan data masukan berupa citra per karakter, sedangkan pada proses *testing* menggunakan data masukan berupa citra plat nomor.

2. Tahap *Preprocessing*

   (a) *Sharpening*

Sharpening adalah proses penajaman citra, yang digunakan dalam meningkatkan kontras warna dan cahaya pada suatu citra.

(b) *Grayscalling*
Konversi citra berwarna yang telah menjadi *grayscale*.

(c) *Deblurring*
Deblurring adalah proses penghilangan efek kabur pada citra. Dalam penelitian ini kabur atau *blur* yang akan dihilangkan adalah *motion blur*.

(d) Binerisasi
Binerisasi adalah sebuah proses untuk mengubah citra *grayscale* menjadi citra yang hanya memiliki dua kemungkinan yaitu 0 untuk hitam dan 1 untuk putih.

(e) Operasi Morfologi
Operasi morfologi berguna untuk memisahkan karakter yang terlalu dekat dan menghilangkan *noise*, sehingga setiap karakter nantinya dapat diolah dengan baik di tahap selanjutnya.

(f) Segmentasi dan *cropping*
Setelah gambar diproses dalam operasi morfologi langkah selanjutnya adalah membagi citra menjadi citra per-karakter yang dilakukan pada tahap *testing*.

(g) Normalisasi Karakter
Setelah berhasil melakukan segmentasi, tiap citra karakter diatur dalam satu ukuran sehingga semua citra karakter menjadi satu ukuran yang sama yaitu 7×12.

(h) Ekstraksi ciri
Setelah gambar diubah ukurannya menjadi satu

ukuran, lalu sistem akan mengekstraksi ciri. Ekstraksi ciri dilakukan dengan cara memasukkan nilai piksel gambar ke dalam sebuah *array* satu dimensi. Sebelum dimasukkan ke dalam *array*, piksel diubah terlebih dahulu menjadi nilai 1 dan 0, dimana 0 mewakili warna hitam sedangkan 1 mewakili warna putih pada citra.

3. Tahap JST

   (a) Proses *Training*
   Proses melatih sistem sehingga mampu mengenali apabila diberikan masukan baru.

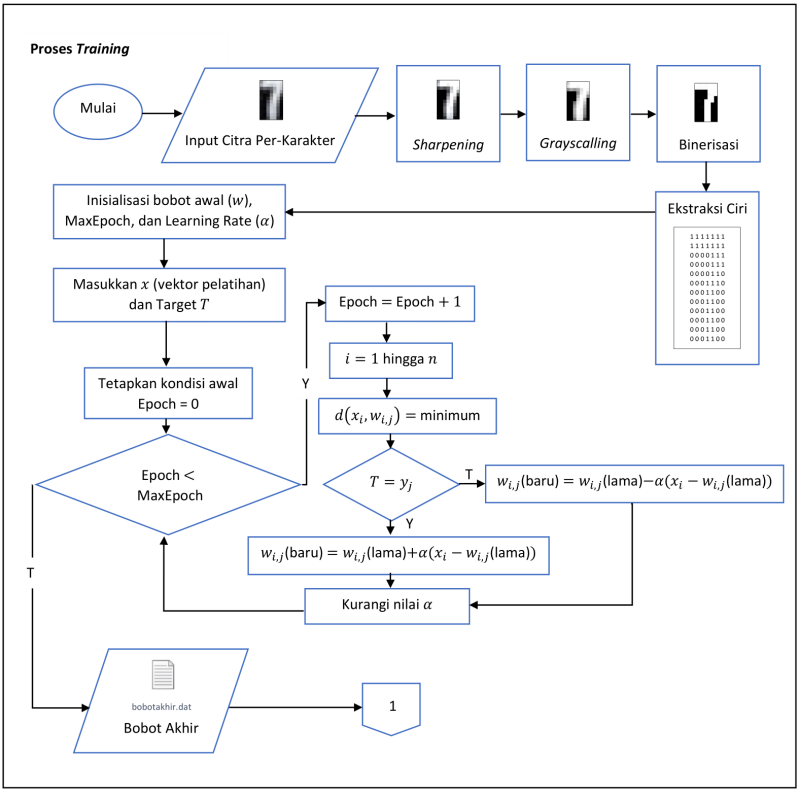
   (b) Proses *Testing*
   Proses pencocokkan ciri suatu masukan baru terhadap ciri referensi yang sebelumnya telah dilatihkan kepada sistem.

3.3 Simulasi Program

Pada tahap ini dilakukan pembuatan implementasi dari tugas akhir ini berupa simulasi program sesuai dengan perancangan program yang telah dilakukan dengan menggunakan bahasa pemrograman Java dengan *library* OpenCV. Pada tahap ini juga akan dilihat apakah program yang dibuat dapat mengenali plat nomor kendaraan melalui gambar dan mengonversinya ke sebuah teks. Apabila terjadi kesalahan, akan dilakukan perbaikan program sehingga didapat hasil yang baik, akurat dan sesuai dengan yang diinginkan.

3.4 Pengujian dan Evaluasi Sistem

Tahap terakhir dari penelitian ini adalah menganalisis hasil program yang telah dibuat. Pengujian dinyatakan berhasil apabila keseluruhan dari fungsi program dapat

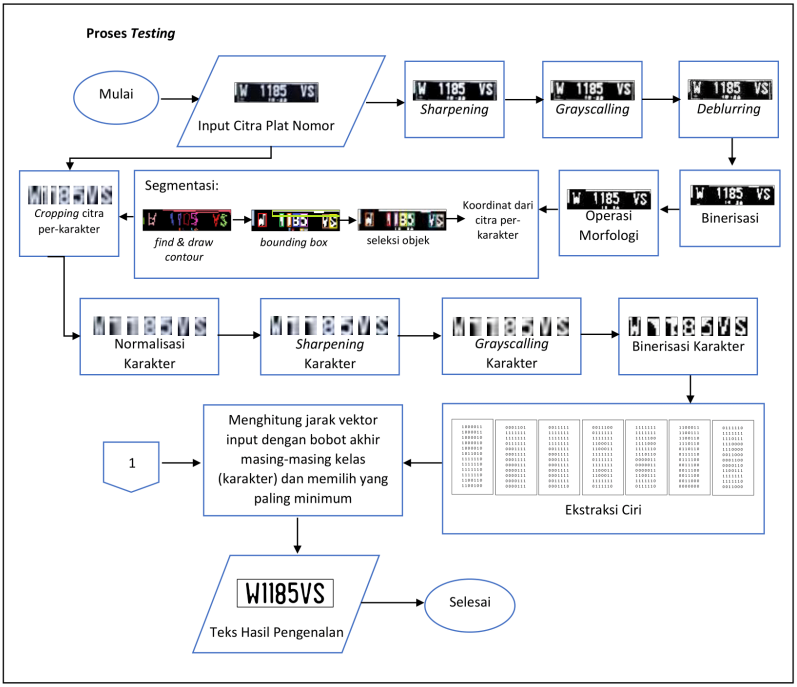


Gambar 3.2: Blok diagram proses *training*

berjalan dengan baik dan sesuai dengan kebutuhan. Apabila terdapat fungsi yang tidak berjalan dengan baik maka dari analisis ini diharapkan diketahui penyebabnya, sehingga kekurangan sistem dapat diperbaiki

3.5 Penarikan Kesimpulan dan Penyusunan Laporan Tugas Akhir

Setelah dilakukan pengujian sistem kemudian dilakukan penarikan kesimpulan dari hasil pengujian yang dilakukan.



Gambar 3.3: Blok diagram proses *testing*

Selain itu, juga akan diberikan saran terkait kekurangan hasil penelitian agar program dapat dikembangkan lebih lanjut. Setelah itu akan dibuat laporan terkait dengan keseluruhan tahapan/proses yang telah dilakukan.

BAB IV
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dibahas mengenai analisis dan perancangan sistem. Untuk tahap perancangan sistem dimulai dari pembahasan tentang perancangan data, gambaran sistem pengenalan plat nomor kendaraan, perancangan *interface* (antarmuka), dan perancangan arsitektur Algoritma *Learning Vector Quantization.*

4.1 Analisis Sistem

Untuk membangun perangkat lunak yang dapat mengenali plat nomor kendaraan beresolusi rendah, maka diperlukan sistem yang mengolah data masukan berupa gambar plat nomor dari kendaraan beresolusi rendah. Pada tugas akhir ini menggunakan data masukan berupa gambar plat nomor kendaraan beresolusi rendah. Dengan menggunakan pengolahan citra digital dan metode *Learning Vector Quantization*, dapat digunakan untuk mengenali plat nomor kendaraan beresolusi rendah sehingga tercipta sistem yang sesuai dengan tujuan tugas akhir ini.

4.1.1 Analisis Fungsional Sistem

Perangkat lunak yang dibangun untuk mendeteksi plat nomor kendaraan ini memiliki kemampuan yaitu mengenali karakter plat nomor kendaraan beresolusi rendah dan menampilkannya dalam bentuk teks.

4.1.2 Analisis non-Fungsional Sistem

Perangkat lunak ini dikembangkan dengan menggunakan *software* dan *hardware* sebagaimana dijelaskan dalam Tabel 4.1.

Tabel 4.1: Spesifikasi *software* dan *hardware* yang digunakan.

Perangkat keras	Prosesor : Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz
	RAM : 6.0 GB
	Sistem Operasi : Windows 10 Pro 64-bit
Perangkat lunak	Netbeans IDE 8.2, dengan bahasa pemograman Java dan library OpenCV 4.1.1

4.2 Perancangan Sistem

Setelah semua kebutuhan sistem dianalisis maka masuk ke tahapan perancangan. Tahapan perancangan meliputi perancangan data sistem, perancangan proses, dan perancangan antarmuka.

4.2.1 Perancangan Data

Data yang digunakan dalam sistem ini dibagi menjadi 3 macam, yaitu data masukan, data proses, dan data luaran.

1. Data Masukan
   Data masukan dalam sistem pengenalan plat nomor ini dibagi menjadi dua. Untuk data masukan proses *training* berupa gambar per-karakter plat nomor kendaraan beresolusi rendah. Sedangkan untuk data masukan proses *testing* adalah gambar utuh plat nomor kendaraan beresolusi rendah.

2. Data Proses
   Gambar yang diperoleh dari data masukan merupakan satu-satunya data yang akan menjadi penentu data-data selanjutnya. Gambar tersebut akan diolah oleh proses

di dalam sistem yang menggiring semua data agar menjadi data yang dibutuhkan oleh Algoritma LVQ. Berikut pada Tabel 4.2 adalah data proses dalam proses *testing* pengenalan plat nomor kendaraan bergerak.

Tabel 4.2: Data proses

No.	Tahapan	Masukan	Keluaran
1.	*Sharpening*	Citra asli	Citra *sharpened*
2.	*Grayscalling*	Citra *sharpened*	Citra *grayscale*
3.	*Deblurring*	Citra *grayscale*	Citra *deblur*
4.	Binerisasi	Citra *deblur*	Citra biner
5.	Operasi Morfologi	Citra biner	Citra biner hasil operasi morfologi
6.	Segmentasi citra	Citra biner hasil operasi morfologi	Citra per-karakter
7.	Normalisasi Karakter	Citra per-karakter	Citra per-karakter dengan ukuran 7×12 piksel
8.	*Preprocessing* Karakter	Citra per-karakter dengan ukuran 7×12 piksel	Citra biner per-karakter dengan ukuran 7×12 piksel
9.	Ekstraksi Ciri	Citra biner per-karakter dengan ukuran 7×12 piksel	*Array* satu dimensi

3. Data Luaran
   Data luaran pada sistem ini berupa teks digital hasil pengenalan oleh Algoritma LVQ terhadap data yang diberikan. Setelah data masukan mengalami proses pengolahan citra digital dan ekstraksi ciri maka didapat data yang merupakan masukan bagi Algoritma LVQ. Selanjutnya data ekstraksi ciri diolah sehingga menghasilkan keputusan bahwa data yang diproses tersebut dikenali sebagai karakter. Keputusan tersebut berdasarkan pada nilai yang didapat dari hasil perbandingan ciri data dengan referensi yang ada dalam sistem tersebut. Luaran akan bergantung pada hasil perbandingan oleh algoritma LVQ terhadap data yang diterima dan data referensi.

4.2.2 Perancangan Proses

Sistem perangkat lunak yang dibangun ini memiliki tiga tahapan yaitu tahap akuisisi, tahap *preprocessing* dan tahap Jaringan Syaraf Tiruan (JST).

1. Tahap Akuisisi Citra
   Tahap akuisisi merupakan tahap awal dari sistem, yaitu memasukkan data masukan ke sistem. Tahap akuisisi pada proses *training* menggunakan data masukan berupa citra per-karakter dengan ukuran 7×12, sedangkan pada proses *testing* menggunakan data masukan berupa citra plat nomor yang jika memiliki lebar lebih dari 25, maka akan dikenai *scalling* menjadi lebar 20 dan panjang yang menyesuaikan dengan aspek rasio.

2. Tahap *Preprocessing*
   Tahap *preprocessing* adalah tahap yang menggunakan pengolahan citra digital untuk mengolah citra masukan menjadi data masukan bagi tahap JST. Proses

ini terdiri dari beberapa operasi yaitu *sharpening*, *grayscalling*, *deblurring*, binerisasi, operasi morfologi, segmentasi, normalisasi karakter, *preprocessing karakter* dan ekstraksi ciri.

(a) *Sharpening*
Sharpening merupakan proses penajaman citra yang digunakan dalam meningkatkan kontras warna dan cahaya pada suatu citra. Proses penajaman citra dilakukan untuk mempermudah pengguna dalam menginterpretasikan objek-objek yang ada pada tampilan citra. Pada Tugas Akhir ini, penajaman citra akan menggunakan Algoritma *Unsharp Mask*[14] yang dapat dituliskan seperti pada Persamaan (4.1).

$$U(x,y) = I(x,y) + k[I(x,y) - G(x,y)] \tag{4.1}$$

dengan $U(x,y)$ adalah citra hasil *sharpening*, $I(x,y)$ adalah citra asli, $G(x,y)$ adalah citra hasil *low-pass filtering*, dan k adalah konstanta dengan $k > 0$. *Low-pass filter* yang digunakan dalam Tugas Akhir ini adalah Gaussian Filter. Bentuk Gaussian Filter dalam citra dua dimensi didefinisikan dengan Persamaan (4.2)

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.2}$$

dimana σ adalah standar deviasi. Sebagai contoh, dengan $\sigma = 0.8$, dapat dibuat sebuah kernel berukuran 3×3 sebagai berikut

$$K = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Setelah didapat kernelnya, maka Gaussian Filter dapat diterapkan pada citra asli dengan menggunakan konvolusi. Sebagai contoh, diasumsikan matriks A adalah sebagian citra pada salah satu karnal dengan ukuran 5×5. Hasil perhitungan setelah dilakukan konvolusi dengan kernel Gaussian akan disimpan dalam matriks B, dengan $*$ adalah operator konvolusi.

$$A = \begin{bmatrix} 0 & 34 & 214 & 248 & 253 \\ 0 & 114 & 252 & 251 & 255 \\ 7 & 46 & 179 & 229 & 233 \\ 9 & 13 & 53 & 202 & 206 \\ 13 & 18 & 58 & 209 & 236 \end{bmatrix}$$

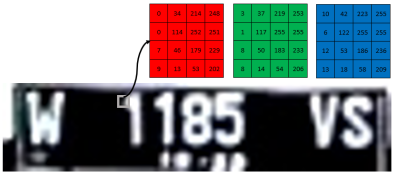
$$B = A * K$$

$$B = \begin{bmatrix} 9 & 41 & 180 & 211 & 255 \\ 7 & 67 & 130 & 178 & 255 \\ 10 & 59 & 109 & 156 & 230 \\ 11 & 32 & 88 & 131 & 216 \\ 8 & 23 & 31 & 195 & 241 \end{bmatrix}$$

Kemudian akan dilakukan proses *Unsharp Masking* menggunakan fungsi *AddWeight* dengan $k = 0.5$. Tahap *sharpening* ini dilakukan pada tahap *training* dan tahap *testing*. Hasil dari proses *sharpening* ditunjukkan pada Gambar 4.1.

Gambar 4.1: Proses *Sharpening*, (a) Citra asli, (b) Citra hasil *sharpening*

(b) *Grayscalling*
Data masukan sistem merupakan citra RGB. Citra RGB (*Red, Green, Blue*) merupakan citra yang nilai intensitas pikselnya tersusun oleh tiga karnal warna, yaitu merah, hijau, dan biru. Citra RGB ini yang kemudian akan diubah ke dalam citra *grayscale*. Proses *grayscalling* ini dilakukan pada tahap *training* dan juga pada tahap *testing*. Untuk mendapat citra *grayscale* dari citra RGB, harus melalui proses pengubahan nilai dari citra RGB tersebut. Sebagai contoh, pertama akan diambil sebagian citra berupa matriks 4×4 seperti pada Gambar 4.2.



Gambar 4.2: Citra RGB dengan diambil sebagian pikselnya

Matriks R, G, B dari citra tersebut seperti pada matriks berikut.

$$R = \begin{bmatrix} 0 & 34 & 214 & 248 \\ 0 & 114 & 252 & 251 \\ 7 & 46 & 179 & 229 \\ 9 & 13 & 53 & 202 \end{bmatrix}, G = \begin{bmatrix} 3 & 37 & 219 & 253 \\ 1 & 117 & 255 & 255 \\ 8 & 50 & 183 & 233 \\ 8 & 14 & 54 & 206 \end{bmatrix},$$

$$B = \begin{bmatrix} 10 & 42 & 223 & 255 \\ 6 & 122 & 255 & 255 \\ 12 & 53 & 186 & 236 \\ 13 & 18 & 58 & 209 \end{bmatrix}$$

Proses *grayscalling* akan dilakukan dengan menggunakan metode Luminosity, yaitu dengan mengalikan nilai RGB yang telah didapat

dengan konstanta yang telah ditentukan. Secara matematis dapat ditulis sebagai Persamaan (4.3).

$$grayscale = (0.21 \times R) + (0.72 \times G) + (0.07 \times B) \quad (4.3)$$

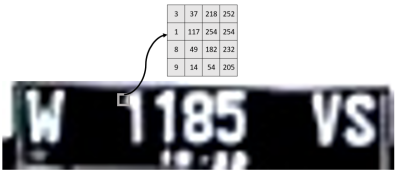
Matriks dari sebagian piksel citra masukan yang akan diubah menjadi citra *grayscale* didapatkan sebagai berikut.

$$grayscale = 0.21 \begin{bmatrix} 0 & 34 & 214 & 248 \\ 0 & 114 & 252 & 251 \\ 7 & 46 & 179 & 229 \\ 9 & 13 & 53 & 202 \end{bmatrix}$$

$$+0.72 \begin{bmatrix} 3 & 37 & 219 & 253 \\ 1 & 117 & 255 & 255 \\ 8 & 50 & 183 & 233 \\ 8 & 14 & 54 & 206 \end{bmatrix} +0.07 \begin{bmatrix} 10 & 42 & 223 & 255 \\ 6 & 122 & 255 & 255 \\ 12 & 53 & 186 & 236 \\ 13 & 18 & 58 & 209 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 37 & 218 & 252 \\ 1 & 117 & 255 & 254 \\ 8 & 49 & 182 & 232 \\ 9 & 14 & 54 & 205 \end{bmatrix}$$

Sehingga citra tersebut telah diubah menjadi citra *grayscale* seperti terlihat pada Gambar 4.3.



Gambar 4.3: Hasil proses *grayscalling*

(c) *Deblurring*

Blur (kesamaran) pada citra merupakan salah satu proses degradasi citra yang direpresentasikan melalui Persamaan (4.4).

$$g(i,j) = (f * h)(i,j) + v(i,j) \quad (4.4)$$

dimana f merepresentasikan citra asli, h adalah PSF (*Point Spread Function*) dan v adalah *additive noise*. Notasi (i, j) adalah koordinat piksel citra dan $*$ merepresentasikan operator konvolusi[15]. Blur atau efek kabur pada citra dapat disebabkan oleh beberapa faktor, yang mana pada penelitian ini efek kabur dalam citra plat nomor disebabkan oleh adanya gerakan dalam proses pengambilan gambar karena citra diambil ketika kendaraan bergerak.

Dalam persamaan model dua dimensi, terlihat bahwa PSF atau operator konvolusi berperan dalam menghasilkan suatu data keluaran. PSF merupakan faktor penting restorasi citra, karena PSF menggambarkan distorsi. Untuk PSF jenis *motion blur*, diperlukan masukan berupa LEN dan TETHA. LEN mewakili panjang piksel dari efek kabur yang akan diterapkan pada citra, sedangkan TETHA mewakili sudut dari efek kabur yang diterapkan pada citra.

Deblurring (penghilangan kesamaran) merupakan proses kebalikan dari *blurring* (proses memberikan efek blur pada citra). Untuk melakukan proses *deblurring* maka diperlukan informasi penyebab dari citra yang terdegradasi yang didapat dari parameter PSF. Terdapat beberapa algoritma *motion deblur*, yang akan dibahas pada pembahasan ini adalah Metode Wiener.

Metode Wiener Filter adalah metode restorasi citra berdasarkan pada *least square*. Metode ini meminimumkan galat restorasi, yaitu selisih antar citra restorasi dengan citra asli. Metode Wiener dapat dinyatakan dalam Persamaan (4.5)

$$F'(u, v) = \left[\frac{1}{H(u, v)} \times \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v) \quad (4.5)$$

yang dalam hal ini, $F'(u, v)$ dan $G(u, v)$

berturut-turut menyatakan citra deblur dan citra terdegradasi, sedangkan $H(u, v)$ menyatakan fungsi degradasi, dan K merupakan konstanta yang dispesifikkan pengguna[16].
Pengujian *motion deblurring* plat nomor kendaraan menggunakan Metode Wiener menghasilkan hasil yang diwakili oleh Tabel 4.3 berikut.

Tabel 4.3: Hasil uji coba Metode Wiener

Citra *Grayscale*	Citra Hasil *Deblurring*	Keterangan
		LEN = 2, TETHA = 20
		LEN = 2, TETHA = 40
		LEN = 2, TETHA = 60
		LEN = 2, TETHA = 80

(a) (b)

Gambar 4.4: Hasil *deblurring*, (a) Citra *grayscale*, (b) Citra hasil *deblurring*

Berdasarkan Tabel 4.3 didapatkan citra hasil *deblurring* yang memiliki kualitas paling baik adalah dengan LEN = 2 dan TETHA = 40.

(a) (b)

Gambar 4.5: Hasil binerisasi citra *deblurring*, (a) Citra biner, (b) Citra biner hasil *deblurring*

Apabila citra hasil *deblurring* tersebut dilihat lebih dalam, maka akan terlihat seperti pada Gambar 4.4 (b). Pada Gambar 4.4 (b) terlihat bahwa citra hasil *deblurring* memunculkan titik-titik *noise*. Hal itu tentu akan berpengaruh pada proses selanjutnya, yaitu binerisasi. Seperti yang terlihat pada Gambar 4.5, citra biner hasil *deblurring* menampilkan titik-titik dan membuat karakter tidak terlihat lebih baik dibanding dengan citra biner yang asli. Hal ini dapat disebabkan karena faktor dimensi dari citra masukan. Metode Wiener sendiri memerlukan parameter berupa PSF yang mana mengandung LEN (panjang piksel) dan TETHA (sudut piksel), maka dimensi dari citra sangat berpengaruh pada proses *deblurring*. Citra masukan dalam penelitian ini memiliki dimensi yang relatif kecil, sehingga menyebabkan proses *deblurring* tidak menjadi baik dan justru menimbulkan *noise*. Maka dari itu proses *deblurring* dengan menggunakan metode Wiener tidak digunakan dalam Tugas Akhir ini.

(d) Binerisasi
Setelah melalui proses *grayscalling*, citra *grayscale* akan diubah menjadi citra biner melalui proses binerisasi. Proses ini dilakukan pada tahap *training*

dan juga tahap *testing*. Proses binerisasi dilakukan untuk mengelompokkan semua piksel pada citra dengan nilai *threshold* menjadi dua bagian yaitu hitam dan putih dengan nilai *gray* level yang telah ditentukan. Pada proses ini kecerahan warna dari setiap piksel dibandingkan ke dalam sebuah nilai *threshold*. Seleksi nilai *threshold* bisa didapat dari histogram.

Jika sebuah gambar terdiri dari dua area, dimana satu daerah gelap dan daerah yang lain terang, maka bisa diasumsikan sebuah histogram mempunyai dua puncak. Dan nilai *threshold* bisa dipilih berdasarkan nilai antara dua puncak histogramnya. Binerisasi dapat didefinisikan dengan Persamaan (4.6).

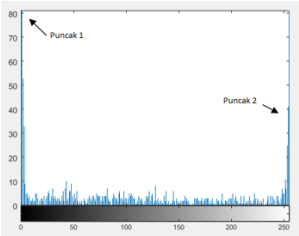
$$g(x,y) = \begin{cases} 1, & \text{jika } f(x,y) \geq T \\ 0, & \text{jika } f(x,y) < T \end{cases} \tag{4.6}$$

dimana

- $g(x,y)$ = citra hasil binerisasi
- $f(x,y)$ = citra *grayscale*
- T = nilai *threshold*

Pada Gambar 4.6 tampak bahwa histogram dari citra *grayscale* memiliki dua puncak, sehingga dipilih titik di antara dua puncaknya. Berdasarkan hasil eksperimen seperti yang terlihat pada Gambar 4.7, maka dipilih $T = 170$, sehingga citra *grayscale* diambil sebagian pikselnya menjadi matriks M, maka matriks *threshold*-nya adalah sebagai berikut.

(a)



(b)

Gambar 4.6: Citra beserta histogramnya, (a) citra *grayscale*, (b) histogram

(a) (b) (c) (d)

Gambar 4.7: Eksperimen nilai *threshold*, (a) $T = 100$, (b) $T = 140$, (c) $T = 170$, (d) $T = 200$

$$\text{Thres}\begin{bmatrix} 3 & 37 & 218 & 252 \\ 1 & 117 & 255 & 254 \\ 8 & 49 & 182 & 232 \\ 9 & 14 & 54 & 205 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hasil proses binerisasi diberikan pada Gambar 4.8. Terlihat bahwa objek karakter ditandai dengan warna putih dan *background* berwarna hitam.

Gambar 4.8: Hasil proses binerisasi

(e) Operasi Morfologi

Operasi Morfologi atau *morphological operation* merupakan bentuk filter yang berguna untuk membersihkan *noise* pada citra biner[17]. Operasi morfologi dilakukan dengan membandingkan objek yang diteliti dengan sebuah matriks yang disebut struktur elemen. Matriks yang digunakan pada struktur elemen dapat menyerupai cakram, garis, lingkaran, dan lain-lain tergantung pada kebutuhan pengguna dalam proses pengolah citra digital. Operasi-operasi pada operasi morfologi dilakukan dengan tujuan untuk identifikasi objek, eliminasi kebisingan citra, dan kebutuhan lainnya[10]. Dalam penelitian ini digunakan dua macam operasi morfologi yaitu erosi dan dilasi yang hanya dilakukan pada tahap *testing*.

Erosi adalah transformasi morfologi yang mengombinasikan dua himpunan dengan menggunakan operasi penghilangan elemen-elemen himpunan. Jika P dan Q adalah himpunan dalam Ruang Euclide dimensi $N(E^N)$, maka erosi P dan Q adalah himpunan semua elemen x dimana $x + q \in P$ untuk setiap $q \in Q$. Erosi dari P dan Q ditulis $P \ominus Q$ dan didefinisikan dengan Persamaan (4.7).

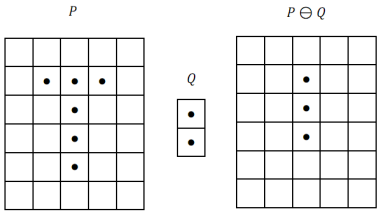
$$P \ominus Q = \{x \in E^N | x + q \in P, \forall q \in Q\} \qquad (4.7)$$

Sebagai contoh dari proses erosi, misalkan P

adalah matriks dari sebuah citra digital biner, dan Q merupakan sebuah matriks struktur elemen yang akan digunakan pada proses erosi.

$$\begin{aligned} P &= \{(1,1),(1,2),(1,3),(2,2),(3,2),(4,2)\} \\ Q &= \{(0,0),(1,0)\} \\ P \ominus Q &= \{(1,2),(2,2),(3,2)\} \end{aligned}$$

Maka proses erosi citra dapat diilustrasikan seperti pada Gambar 4.9.



Gambar 4.9: Contoh proses erosi pada citra biner

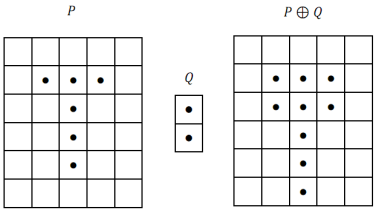
Dilasi adalah transformasi morfologi yang mengombinasikan dua himpunan dengan menggunakan penambahan vektor dari elemen-elemen himpunan. Jika P dan Q adalah himpunan dalam Ruang Euclide dimensi $N(E^N)$ dengan P dan Q masing-masing mempunyai elemen $p = (p_1, p_2, \ldots, p_n)$ dan $q = (q_1, q_2, \ldots, q_n)$, maka dilasi dari P dan Q adalah himpunan dari semua kemungkinan jumlah vektor sepasang elemen dimana satu berasal dari P dan yang lainnya dari Q. Dilasi dari P dan Q ditulis $P \oplus Q$ dan didefinisikan dengan Persamaan (4.8).

$$P \oplus Q = \{r \in E^N | r = p + q, \forall p \in P \text{ dan } q \in Q\} \tag{4.8}$$

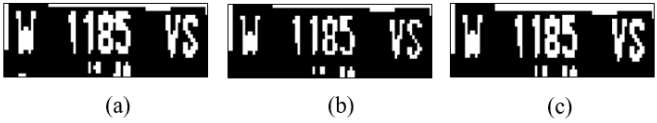
Sebagai contoh dari proses dilasi, misalkan P adalah matriks dari sebuah citra digital biner, dan Q merupakan sebuah matriks struktur elemen yang akan digunakan pada proses dilasi.

$$\begin{aligned} P &= \{(1,1),(1,2),(1,3),(2,2),(3,2),(4,2)\} \\ Q &= \{(0,0),(1,0)\} \\ P \oplus Q &= \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3), \\ &\quad (3,2),(4,2),(5,2)\} \end{aligned}$$

Maka proses dilasi citra dapat diilustrasikan seperti pada Gambar 4.10.



Gambar 4.10: Contoh proses dilasi pada citra biner



Gambar 4.11: Proses erosi dan dilasi, (a) Citra biner, (b) Citra hasil erosi, (c) Citra hasil dilasi

Pada gambar 4.11 (a) yang merupakan citra biner terlihat bahwa karakter "8" dan "5" menyatu, sehingga dilakukan proses erosi dengan kernel $(1,2)$ sehingga terlihat pada gambar (b) bahwa kedua karakter tersebut telah terpisah.

Sedangkan terlihat dari gambar (b) bahwa karakter yang ada menjadi tipis, sehingga untuk mendapatkan hasil proses segmentasi yang optimal pada tahap selanjutnya, maka dilakukan proses dilasi hingga didapatkan citra pada gambar (c). Kernel yang digunakan pada citra hasil proses dilasi adalah kernel $(1, 2)$ dimana citra masukan yang diproses adalah citra hasil proses erosi.

(f) Segmentasi Citra

Segmentasi adalah salah satu metode penting yang digunakan untuk mengubah citra masukan ke dalam citra luaran berdasarkan atribut yang diambil dari citra tersebut. Segmentasi membagi citra ke dalam daerah intensitasnya masing-masing sehingga bisa membedakan antara objek dan *background*-nya.

Dalam penelitian ini, akan dilakukan deteksi tepi dari tiap objek di plat nomor menggunakan *find contour*. Kontur dapat diartikan sebagai garis kurva yang berhubungan dengan titik lainnya yang bersambung sepanjang tepi, yang memiliki warna atau intensitas yang sama. Kontur berguna sebagai *tools* untuk analisis bentuk bangun dan deteksi serta pengenalan objek. Dalam tahap ini, langkah pertama yang harus dilakukan adalah menemukan kontur dari objek di citra plat nomor. Untuk menemukan kontur dari sebuah objek, akan digunakan fungsi *find contour*. Fungsi *find contour* akan mengambil kontur dari suatu citra biner menggunakan algoritma yang dikenalkan oleh S. Suzuki[18]. Langkah-langkah dari algoritma tersebut adalah:

i. Masukan dari algoritma ini adalah suatu

gambar biner, kemudian akan dilakukan analisa pada setiap piksel (i, j). Hentikan analisa jika terdapat suatu piksel (i, j) yang merupakan batas luar atau batas lubang. Jika piksel (i, j) memenuhi kedua kondisi tersebut, maka piksel (i, j) dianggap sebagai titik awal dari batas luar. Tetapkan suatu angka yang unik pada perbatasan yang baru ditemukan, disebut sebagai NBD.

ii. Tentukan *parent* perbatasan dari perbatasan yang baru ditemukan. Selama proses analisa, simpan suatu nomor urut perbatasan dari perbatasan yang baru ditemukan yang disebut sebagai LNBD. LNBD ini merupakan *parent* perbatasan dari perbatasan sebelumnya (NBD). Oleh karena itu, dapat ditentukan urutan nomor *parent* perbatasan (LNBD) dari perbatasan sebelumnya (NBD).

iii. Ikuti perbatasan yang ditemukan dari titik awal dengan menandai piksel. Ketentuan penandaan setiap piksel sebagai berikut:

- Jika perbatasan sekarang adalah di antara 0-komponen yang berisi piksel $(p, q+1)$ dan 1-komponen yang berisi piksel (p, q) maka ubah nilai piksel (p, q) dengan -NBD.
- Jika tidak, tetapkan nilai piksel (p, q) dan piksel $(p, q + 1)$ dengan NBD kecuali piksel berada pada perbatasan yang sudah diikuti.

Penandaan ini mencegah piksel (p, q) menjadi perbatasan titik awal. Jika tanda suatu piksel bernilai positif, maka bisa disebut dengan batas terluar suatu objek.

iv. Lakukan algoritma ini hingga pengamatan piksel mencapai sudut kanan bawah citra.

Setelah mendapatkan kontur dari setiap objek yang ada di plat nomor, agar dapat diilustrasikan maka digambarlah kontur tersebut menggunakan fungsi *draw contour*. Hasil dari *draw contour* dapat dilihat pada Gambar 4.12 berikut.

Gambar 4.12: Proses *draw contour*, (a) Citra hasil operasi morfologi, (b) Citra hasil *draw contour*

Pada Gambar 4.12 (b) dapat dilihat bahwa hasil *find contour* dan *draw contour* dapat mendeteksi objek-objek yang berada di citra plat nomor. Selanjutnya akan dibuat *bounding box* dengan menggunakan fungsi *approPolyDP* berdasarkan kontur-kontur yang telah dideteksi. Fungsi ini digunakan untuk memperkirakan bentuk sebuah kontur dengan bentuk lain yang mana memiliki simpul lebih sedikit sehingga jarak di antara dua kurva tersebut kurang dari atau sama dengan presisi yang ditentukan. Dalam implementasinya, fungsi *approxPolyDP* menggunakan Algoritma Ramer-Douglas-Peucker. Algoritma ini digunakan untuk meminimalisasi jumlah koordinat yang dibutuhkan untuk merepresentasikan sebuah kurva. Algoritma Ramer-Douglas-Peucker bergantung pada jarak metrik epsilon yang akan menentukan apakah sebuah poin dalam kurva

akan diikutkan atau tidak ke dalam poin-poin hasil minimalisasi koordinat. Nilai epsilon merepresentasikan jarak tegak lurus dari segmen garis ke poin lainnya dalam data koordinat yang akan diminimalkan[19]. Berdasarkan eksperimen, pada penelitian ini akan digunakan nilai epsilon yaitu 1. Setelah melalui proses ini, maka akan didapatkan *bounding box* dari tiap-tiap objek yang telah terdeteksi seperti yang terlihat pada Gambar 4.13.

Gambar 4.13: Hasil *bounding box* objek-objek yang terdeteksi

Dapat dilihat pada Gambar 4.13, bahwa objek-objek yang terdeteksi di citra plat nomor tidak hanya karakter saja, juga ada objek-objek lain. Maka untuk mendapatkan hanya objek karakter, maka dilakukan seleksi dengan melihat panjang, lebar dan luas objek. Objek yang memiliki panjang kurang dari 12, lebar kurang dari 12 dan luas lebih dari sama dengan 20, maka akan terdeteksi sebagai objek karakter. Sehingga *bounding box* hasil seleksi akan terlihat seperti pada Gambar 4.14.

Langkah selanjutnya yaitu dengan melakukan *cropping* karakter-karakter yang telah terdeteksi dengan menggunakan data panjang dan lebar dari hasil *bounding box*. Proses *cropping* dikenakan

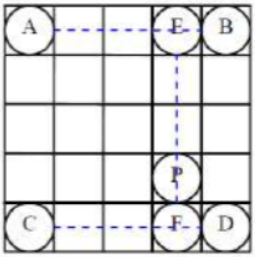
Gambar 4.14: Hasil *bounding box* objek-objek karakter

pada citra asli, sehingga dengan proses ini didapatkan citra per-karakter yang akan menjadi masukan pada tahap selanjutnya. Hasil dari proses *cropping* ditampilkan pada Gambar 4.15 berikut.

Gambar 4.15: Hasil proses *cropping* menjadi citra per-karakter

(g) Normalisasi Karakter
Pada proses normalisasi karakter, citra per-karakter dipetakan pada piksel dengan ukuran tertentu sehingga memberikan representasi dimensi yang tetap. Pada proses ini digunakan teknik interpolasi. Interpolasi adalah sebuah metode untuk meningkatkan (atau mengurangi) jumlah piksel dalam citra digital[10]. Interpolasi yang digunakan dalam penelitian ini adalah Interpolasi Bilinear. Pada implementasinya, proses interpolasi bilinear akan dilakukan pada salah satu sisi dahulu (vertikal atau horizontal), kemudian baru dilanjutkan proses pada sisi yang lain. Harga interpolasi yang diberikan pada suatu titik diperhitungkan dengan memperhatikan distribusi 4 nilai piksel tetangga[20]. Ada beberapa langkah

yang perlu dilakukan untuk menentukan nilai piksel baru. Dicontohkan pada Gambar 4.16,



Gambar 4.16: Interpolasi Bilinear

koordinat piksel A, B, C, dan D adalah $(i, j), (i, j+1), (i+1, j)$, dan $(i+1, j+1)$. Koordinat piksel baru $P(u, v)$. Langkah-langkah metode interpolasi bilinear adalah sebagai berikut.

i. Menghitung pengaruh piksel A dan B, anggap suatu nilai baru, yaitu piksel E.
$f(i, j+v) = [f(i, j+1) - f(i, j)]v + f(i, j)$

ii. Menghitung pengaruh nilai piksel C dan D, anggap sebagai piksel F.
$f(i+1, j+v) = [f(i+1, j+1) - f(i+1, j)]v + f(i+1, j)$

iii. Menghitung pengaruh nilai piksel E dan F, anggap sebagai piksel P.
$f(i+u, j+v) = (1-u)(1-v)f(i, j) + (1-u)vf(i, j+1) + u(1-v)f(i+1, j) + uvf(i+1, j+1)$

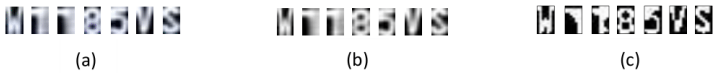
Pada proses normalisasi karakter ini, sistem mengubah ukuran citra menjadi 7×12. Citra yang diubah ukurannya merupakan citra yang

telah dikenai proses segmentasi sebelumnya, sehingga citra yang masuk pada tahap proses normalisasi karakter ini adalah citra per-karakter. Hal ini dilakukan karena ukuran dari tiap-tiap karakter belum tentu sama, padahal sebagai data masukan untuk Algoritma LVQ ukuran *array* harus konsisten. Oleh karena itu perlu dilakukan normalisasi karakter. Ukuran 7×12 digunakan untuk menyesuaikan dimensi citra masukan dengan citra pada proses *training*. Contoh dari proses normalisasi karakter dapat dilihat pada Gambar 4.17.

Gambar 4.17: Proses normalisasi karakter, (a) Citra per-karakter hasil segmentasi (b) Citra per-karakter hasil normalisasi karakter

(h) *Preprocessing* Karakter

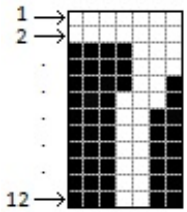
Setelah citra per-karakter memiliki ukuran yang sama, maka selanjutnya citra per-karakter akan dikenakan proses *sharpening*, *grayscalling*, dan binerisasi. Parameter-parameter yang digunakan dalam ketiga proses tersebut sama dengan parameter-parameter yang digunakan sebelumnya. Tahap ini dilakukan agar citra per-karakter yang akan dikenali, melalui tahap yang sama seperti yang dikenakan pada tahap *training*. Hasil dari tahap ini dapat dilihat pada Gambar 4.18.



Gambar 4.18: *Preprocessing* karakter, (a) Citra per-karakter hasil *sharpening* (b) Citra per-karakter hasil *grayscalling*, (c) Citra per-karakter hasil binerisasi

(i) Ekstraksi Ciri

Untuk mendapatkan ciri masing-masing karakter yang akan diproses oleh metode *Learning Vector Quantization*, maka setiap karakter akan dikenai proses ekstraksi ciri. Proses ekstraksi ciri bertujuan untuk mengubah nilai-nilai intensitas koordinat piksel pada citra karakter tersebut menjadi susunan kode-kode nilai pada setiap piksel. Proses ini akan melakukan pengecekan pada setiap piksel dimulai dari ujung atas kiri menuju ke ujung atas kanan, kemudian ke satu piksel dibawah ujung atas kiri menuju ke kanan, dan seterusnya.



Gambar 4.19: Proses pengecekan piksel pada ekstraksi ciri

Piksel karakter (piksel berwarna putih) diberi nilai satu (1) dan untuk piksel *background* (piksel

berwarna hitam) akan diberi nilai nol (0). Contoh ekstraksi ciri dari Gambar 4.19 adalah 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0.

Selanjutnya kode biner akan dimasukkan ke dalam sebuah *array* satu dimensi untuk diolah pada proses JST.

3. Tahap Jaringan Syaraf Tiruan

   Proses JST digunakan untuk melakukan proses *training* dan proses *testing*. Tujuan dari proses *training* dan *testing* adalah untuk mendapatkan keseimbangan antara kemampuan tanggapan yang benar terhadap pola masukan yang dipakai untuk pelatihan jaringan (*memorization*) dan kemampuan memberikan tanggapan yang layak untuk masukan yang sejenis namun tidak identik dengan yang dipakai pada pelatihan (*generalization*).

   (a) Proses *Training*

      i. Menentukan bobot masing-masing kelas acuan, *maxEpoch*, dan *learning rate*

         Jumlah kelas dari sistem ini adalah 36 kelas, yang mewakili 36 karakter (26 huruf dan 10 angka). Setiap karakter akan diambil satu gambar untuk dihitung bobotnya, sehingga akan ada 36 *array* satu dimensi sebagai inisialisasi bobot acuan. Tabel yang menunjukkan vektor bobot acuan akan dilampirkan. Setelah itu juga perlu menentukan *maxEpoch* yaitu sebesar 10, 20, dan 30, serta *learning rate* yaitu sebesar 0.05, 0.1, dan 0.5.

ii. Menghitung jarak Euclidean terdekat antar bobot (C_j)
Untuk menghitung jarak Euclidean terdekat antara bobot yang dilatih dengan bobot acuan digunakan Persamaan (4.9)

$$C_j = \sqrt{(x_i - w_{i,j})^2} \tag{4.9}$$

dimana

- C_j = jarak Euclidean kelas ke-j
- x_i = bobot data latih piksel ke-i
- $w_{i,j}$ = bobot data acuan kelas ke-j piksel ke-i

Sebagai contoh, jika *user* memasukkan citra karakter angka "7" dengan vektor 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 maka perhitungan jarak Euclidean terhadap vektor-vektor acuan adalah sebagai berikut. Epoch ke-1:
C_1 0 = $\sqrt{(1-0)^2 + \cdots + (0-0)^2}$ = 7.07106781186547
C_2 1 = $\sqrt{(1-0)^2 + \cdots + (0-0)^2}$ = 4.12310562561766
C_3 2 = $\sqrt{(1-1)^2 + \cdots + (0-1)^2} = 4.0$
C_4 3 = $\sqrt{(1-1)^2 + \cdots + (0-1)^2} = 5.0$
C_5 4 = $\sqrt{(1-0)^2 + \cdots + (0-1)^2}$ = 5.83095189484530
C_6s 5 = $\sqrt{(1-1)^2 + \cdots + (0-0)^2}$ = 6.24499799839839
C_7 6 = $\sqrt{(1-0)^2 + \cdots + (0-0)^2}$ = 6.32455532033675

C_8 7 $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 2.23606797749979
C_9 8 $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 6.16441400296897
C_{10} 9 $= \sqrt{(1-0)^2 + \cdots + (0-0)^2} =$ 5.47722557505166
C_{11} A $= \sqrt{(1-0)^2 + \cdots + (0-1)^2} =$ 5.6568542494923
C_{12} B $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 5.7445626465380
C_{13} C $= \sqrt{(1-0)^2 + \cdots + (0-1)^2} =$ 6.557438524302
C_{14} D $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 6.2449979983983
C_{15} E $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 5.4772255750516
C_{16} F $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 5.6568542494923
C_{17} G $= \sqrt{(1-0)^2 + \cdots + (0-0)^2} =$ 5.8309518948453
C_{18} H $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 7.2801098892805
C_{19} I $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 4.1231056256176
C_{20} J $= \sqrt{(1-0)^2 + \cdots + (0-0)^2} = 5.0$
C_{21} K $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 5.4772255750516
C_{22} L $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 5.9160797830996
C_{23} M $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 6.7082039324993
C_{24} N $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 7.1414284285428

C_{25} O $= \sqrt{(1-0)^2 + \cdots + (0-0)^2} =$ 5.6568542494923
C_{26} P $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 5.6568542494923
C_{27} Q $= \sqrt{(1-0)^2 + \cdots + (0-0)^2} =$ 5.6568542494923
C_{28} R $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 5.1961524227066
C_{29} S $= \sqrt{(1-0)^2 + \cdots + (0-0)^2} =$ 5.5677643628300
C_{30} T $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 3.7416573867739
C_{31} U $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 6.2449979983983
C_{32} V $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 5.1961524227066
C_{33} W $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 6.708203932499
C_{34} X $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 5.0990195135927
C_{35} Y $= \sqrt{(1-1)^2 + \cdots + (0-0)^2} =$ 4.4721359549995
C_{36} Z $= \sqrt{(1-1)^2 + \cdots + (0-1)^2} =$ 4.4721359549995

Jarak terkecil didapat pada bobot karakter 7 (kelas ke-8) = 2.23606797749979.

iii. Perhitungan bobot baru (w_j)

Setelah mendapat jarak terdekat, akan dihitung bobot yang menjadi bobot baru. Setelah bobot tersebut didapat, maka akan disimpan untuk menjadi bobot acuan di perhitungan selanjutnya. Jika jarak Euclidean terdekat antara bobot yang dilatih dan bobot

acuan sama dengan target, maka perhitungan bobot baru menggunakan Persamaan (4.10)

$$w_{i,j}(\text{baru}) = w_{i,j}(\text{lama}) + \alpha(x_i - w_{i,j}(\text{lama})) \tag{4.10}$$

Sedangkan jika jarak Euclidean terdekat antara bobot yang dilatih dan bobot acuan tidak sama dengan target, maka perhitungan bobot baru menggunakan Persamaan (4.10)

$$w_{i,j}(\text{baru}) = w_{i,j}(\text{lama}) - \alpha(x_i - w_{i,j}(\text{lama})) \tag{4.11}$$

dimana

- $w_{i,j}$ = bobot acuan dari kelas ke-j piksel ke-i
- x_i = bobot yang dilatih piksel ke-i
- α = *learning rate*

Sebagai lanjutan dari contoh sebelumnya, karena jarak Euclidean terdekat antara bobot yang dilatih dan bobot acuan sama dengan target, maka perhitungan bobot baru untuk karakter "7" adalah sebagai berikut.
$w_{1,8}(\text{baru}) = w_{1,8}(\text{lama}) + \alpha(x_1 - w_{1,8}(\text{lama})) = 1 + 0.05(1 - 1) = 1$
$w_{2,8}(\text{baru}) = w_{2,8}(\text{lama}) + \alpha(x_2 - w_{2,8}(\text{lama})) = 1 + 0.05(1 - 1) = 1$
$w_{3,8}(\text{baru}) = w_{3,8}(\text{lama}) + \alpha(x_3 - w_{3,8}(\text{lama})) = 1 + 0.05(1 - 1) = 1$
$w_{4,8}(\text{baru}) = w_{4,8}(\text{lama}) + \alpha(x_4 - w_{4,8}(\text{lama})) = 1 + 0.05(1 - 1) = 1$
$\vdots$
$w_{82,8}(\text{baru}) = w_{82,8}(\text{lama}) + \alpha(x_{82} -$

$w_{82,8}(\text{lama})) = 0 + 0.05(0 - 0) = 0$
$w_{83,8}(\text{baru}) = w_{83,8}(\text{lama}) + \alpha(x_{83} - w_{83,8}(\text{lama})) = 0 + 0.05(0 - 0) = 0$
$w_{84,8}(\text{baru}) = w_{84,8}(\text{lama}) + \alpha(x_{84} - w_{84,8}(\text{lama})) = 0 + 0.05(0 - 0) = 0$
Sehingga bobot acuan baru dari karakter 7 adalah 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0.95 1 1 0 0 0 0 1 1 0.95 0 0 0 1 1 0 0 0 0 0.05 1 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0.05 0 0 0 0 1 1 0 0 0 0 0 1 1 0.05 0 0 0 0 1 1 0 0 0 0 1 1 0.95 0 0 0.

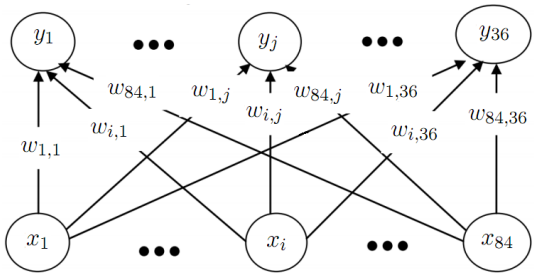
Perhitungan selanjutnya akan dilakukan seperti di atas yaitu menghitung jarak terdekat dari setiap bobot dan menghitung bobot barunya. Perhitungan tersebut akan dijalankan sebanyak *maxEpoch*, dengan ketentuan setiap *Epoch* baru akan menggunakan *learning rate*$\times$0.5.

(b) Proses *Testing*

Pada proses *testing*, setelah menerima *array* satu dimensi dari gambar yang diuji, selanjutnya akan dihitung jarak Euclidean terdekat antara bobot gambar yang diuji dengan bobot hasil *training*. Rumus perhitungan jarak Euclidean untuk proses *testing* ini sama dengan yang terdapat pada proses *training*. Setelah didapat jarak terdekatnya, maka sistem akan melihat kelas dari bobot hasil pelatihan tersebut dan menampilkan karakter-karakter yang berhasil diuji.

Pada penelitian ini, metode Jaringan Syaraf Tiruan yang digunakan adalah metode *Learning Vector Quantization* (LVQ). LVQ merupakan metode klasifikasi pola yang masing-masing unit luaran mewakili kategori atau kelas tertentu. Ada 2 parameter yang diperlukan

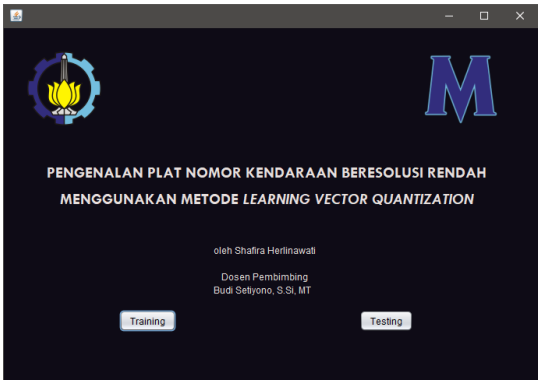
Algoritma LVQ, yaitu *learning rate* dan *epoch* (iterasi). Dalam penelitian ini, jaringan LVQ dirancang dengan komponen 84 neuron pada lapisan masukan yang dinotasikan dengan x_i (karena citra memiliki ukuran 7×12, sehingga jumlah nilai dari *array* yang dimasukkan adalah 84), dan 36 neuron pada lapisan luaran yang dinotasikan dengan y_j (jumlah kelas yang terdiri dari 26 huruf dan 10 angka). Berikut pada Gambar 4.20 adalah arsitektur LVQ dalam penelitian ini.



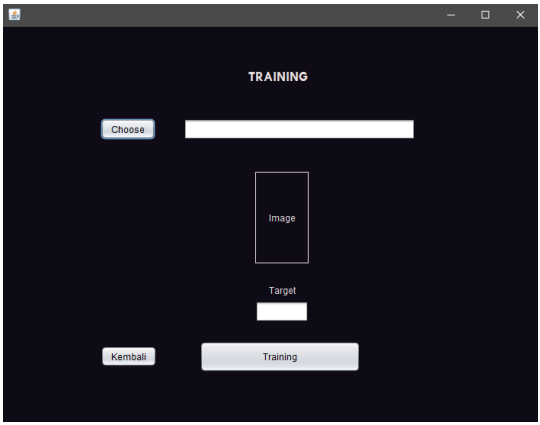
Gambar 4.20: Arsitektur *Learning Vector Quantization*

4.2.3 Perancangan Antar Muka

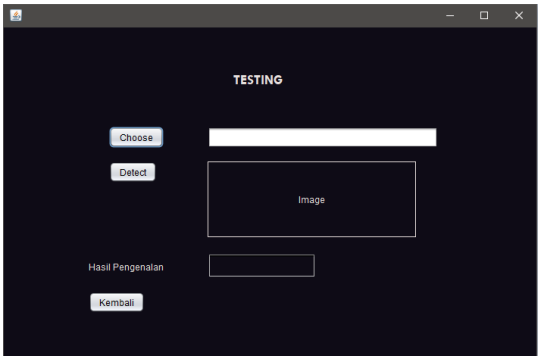
Desain antar muka sistem dibutuhkan agar pengguna dapat dengan mudah mengoperasikan perangkat lunak yang telah dibuat. Antar muka dibagi menjadi tiga bagian, yaitu halaman muka, halaman untuk proses *training*, dan halaman untuk proses *testing*. Halaman awal seperti pada Gambar 4.21 memiliki dua tombol, yaitu untuk menuju ke halaman *training* (Gambar 4.22) dan halaman *testing* (Gambar 4.23).



Gambar 4.21: Tampilan awal program



Gambar 4.22: Tampilan halaman *training*



Gambar 4.23: Tampilan halaman *testing*

BAB V
UJI COBA DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai pengujian program dan pembahasan dari hasil uji coba. Pengujian yang dilakukan adalah pengujian program pada proses *training* dan proses *testing*.

5.1 Data Uji Coba

5.1.1 Data Uji Coba Proses *Training*

Uji coba pada proses *training* dalam Tugas Akhir ini dilakukan terhadap citra per-karakter dari plat nomor kendaraan beresolusi rendah. Citra yang digunakan berukuran 7×12 berjumlah 595 buah yang berasal dari 100 citra plat nomor kendaraan beresolusi rendah.

5.1.2 Data Uji Coba Proses *Testing*

Uji coba pada proses *testing* dalam Tugas Akhir ini dilakukan terhadap citra plat nomor kendaraan beresolusi rendah. Citra memiliki ukuran yang berbeda dengan jumlah 25 buah.

5.2 Hasil Uji Coba

Berikut adalah hasil uji coba yang dilakukan dalam program, terbagi menjadi dua yaitu uji coba pada proses *training* dan uji coba pada proses *testing*.

5.2.1 Uji Coba Proses *Training*

Proses *training* ini bertujuan untuk mencari bobot dari masing-masing huruf. Bobot tersebut yang akan digunakan pada proses *testing*. Bobot paling akhir dari proses *training* adalah bobot yang akan digunakan dalam

proses *testing*. Sebelumnya, sistem telah diatur untuk melakukan proses *training* dengan *maxEpoch* 30 dan *learning rate* 0.05, dan telah diinisiasi bobot awal dari setiap karakter dalam berkas `.dat` bernama `bobotacuan.dat`. Proses yang berkenaan langsung dengan *user* dimulai dengan memilih gambar per-karakter dengan menekan tombol Choose untuk proses akuisisi data. Proses akuisisi ini diimplementasikan oleh *source code* berikut.

```
String file = txtFileName.getText();
Mat src = Imgcodecs.imread(file);
```

Selanjutnya jika gambar sudah dipilih, maka *user* harus menuliskan karakter dari gambar yang dimasukkan di Target, kemudian klik tombol Training. Proses yang terjadi di dalam program selanjutnya yaitu proses *sharpening*, yaitu proses penajaman citra. Proses *sharpening* diimplementasikan dengan *source code* berikut.

```
Mat sharp = new Mat();
Imgproc.GaussianBlur(src, sharp, new Size(0,0), 10);
Core.addWeighted(src, 1.5, sharp, -0.5, 0, sharp);
```

Setelah melalui proses *sharpening*, maka langkah selanjutnya adalah *grayscalling*. Proses *grayscalling* yaitu mengubah citra masukan yang berwarna menjadi citra *grayscale*. Proses *grayscalling* ini diimplementasikan dengan *source code* berikut.

```
Imgproc.cvtColor(sharp, gray, Imgproc.COLOR_BGR2GRAY);
```

Setelah melalui proses *grayscalling*, maka citra *grayscale* akan diubah menjadi citra biner dalam proses binerisasi. Proses binerisasi diimplementasikan dengan *source code* berikut.

```
Imgproc.threshold(gray, bin, 170, 255, Imgproc.
THRESH_BINARY);
```

Citra per-karakter yang telah menjadi citra biner, kemudian dikenai dengan proses ekstraksi ciri. Proses ekstraksi ciri akan mengubah citra biner menjadi sebuah *array* satu dimensi yang berisi kode rantai dari karakter yang diolah dengan ketentuan piksel berwarna hitam bernilai 0, dan piksel berwarna putih bernilai 1. *Array* ini yang akan menjadi data masukan pada tahap selanjutnya. Proses ekstraksi ciri ini diimplementasikan pada kode program berikut.

```
int width = bin.rows();
int height = bin.cols();
int a, b;
double arrayDataSingle[] = new double[width * height];
int arrayDataSingleInt[] = new int[width * height];
int pixelValue = 0;
int iterasi = 0;
for (a = 0; a < width; a++) {
    for (b = 0; b < height; b++) {
        double[] data = bin.get(a, b);
        pixelValue = (int) data[0];
        arrayDataSingle[iterasi] = pixelValue == 0 ? 0 :
1;
        arrayDataSingleInt[iterasi] = (int)
arrayDataSingle[iterasi];
        System.out.print(arrayDataSingleInt[iterasi] + ",
");
        iterasi++;
    }
}
```

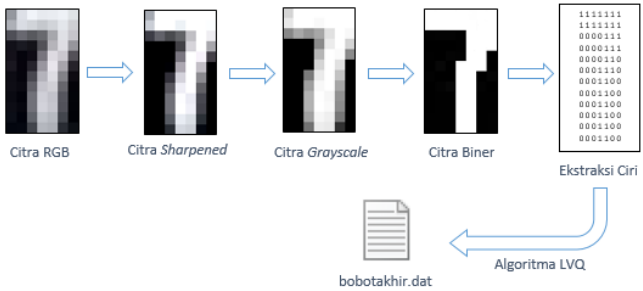
Proses selanjutnya yaitu proses perhitungan dengan Algoritma LVQ. Proses *training* oleh Algoritma LVQ diimplementasikan menjadi sebuah program ke dalam fungsi berikut.

```
public double[] lvqrun(double[] gambaruji, double[][]
gambaracuan, int inputTarget)
```

Algoritma LVQ pada tahap ini dimulai dengan menghitung jarak Euclidean antara bobot acuan dengan bobot yang dilatih. Setelah mendapatkan jarak Euclidean, perhitungan selanjutnya adalah menghitung jarak terdekat

dari setiap bobot dan menghitung bobot barunya. Perhitungan tersebut akan dijalankan sebanyak *maxEpoch*, dengan ketentuan setiap *Epoch* baru akan menggunakan *learning rate*×0.5. Proses *training* ini kemudian diulang kembali dengan menggunakan data *training* yang telah disiapkan.

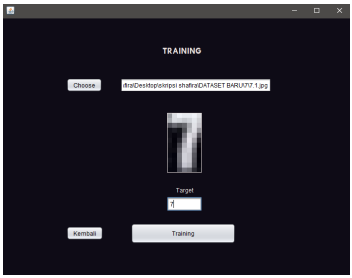
Setelah *user* selesai melakukan *training* pada seluruh data, maka sistem akan secara otomatis memasukkan bobot akhir pelatihan ke suatu berkas bernama `bobotakhir.dat`. Berkas ini yang nantinya akan digunakan pada proses *testing*. Setelah berhasil melakukan *training* dengan *maxEpoch* 30 dan *learning rate* 0.05, selanjutnya juga dilakukan *training* dengan *maxEpoch* 10 dan 20, serta *learning rate* 0.1 dan 0.5. Secara umum, proses *training* dapat digambarkan dalam Gambar 5.1 dan Gambar 5.2.



Gambar 5.1: Proses *Training*

5.2.2 Uji Coba Proses Pengujian

Proses pengujian adalah proses terakhir dari sistem ini. Proses ini dilakukan pada saat proses pelatihan telah dilakukan. Sebelumnya sistem telah mengambil berkas `bobotakhir.dat` sebagai bobot akhir dari setiap karakter.



Gambar 5.2: Tampilan halaman proses *training*

User dapat memulai dengan menekan tombol Choose untuk memilih folder berisi kumpulan citra plat nomor yang akan diuji. Proses akuisisi ini diimplementasikan dengan *source code* berikut.

```
String pathFile = file + "\\" + getFile + ".jpg";
Mat src = Imgcodecs.imread(pathFile);
```

Setelah memilih, maka *user* dapat menekan tombol Testing. Proses yang terjadi di dalam program selanjutnya yaitu proses *scalling* ukuran citra, jika citra memiliki lebar lebih dari 25, maka akan dikenai *scalling* menjadi lebar 20 dan panjang yang menyesuaikan aspek rasio. . Proses yang terjadi di dalam program selanjutnya yaitu proses *sharpening*, yaitu proses penajaman citra. Proses *sharpening* diimplementasikan dengan *source code* berikut.

```
Mat sharp = new Mat();
Imgproc.GaussianBlur(src, sharp, new Size(0,0), 10);
Core.addWeighted(src, 1.5, sharp, -0.5, 0, sharp);
```

Proses selanjutnya yaitu *grayscalling*, yaitu mengubah citra plat nomor yang berwarna menjadi citra *grayscale*. Proses *grayscalling* ini diimplementasikan dengan *source code* berikut.

```
Imgproc.cvtColor(sharp, gray, Imgproc.COLOR_BGR2GRAY);
```

Setelah melalui proses *grayscalling*, maka citra *grayscale* akan diubah menjadi citra biner dalam proses binerisasi. Proses binerisasi diimplementasikan dengan *source code* berikut.

```
Imgproc.threshold(gray, bin, 170, 255, Imgproc.
THRESH_BINARY);
```

Setelah mendapatkan citra biner, maka selanjutnya dilakukan operasi morfologi, yaitu erosi dan dilasi. Proses ini berguna untuk memisahkan karakter yang menempel atau berdekatan pada plat nomor. Proses erosi dan dilasi diimplementasikan dengan *source code* berikut.

```
Mat elementE = Imgproc.getStructuringElement(Imgproc.
CV_SHAPE_RECT, new Size(1, 2), new Point(0, 0));
Imgproc.erode(bin, erosi, elementE);

Mat element = Imgproc.getStructuringElement(Imgproc.
CV_SHAPE_RECT, new Size(1, 2), new Point(0, 0));
Imgproc.dilate(erosi, dilasi, element);
```

Proses selanjutnya adalah segmentasi. Pada proses ini akan dicari kontur dari tiap karakter plat nomor, kemudian membuat *bounding box* untuk setiap karakter. Setelah itu, program akan melakukan seleksi untuk mengambil objek karakter dan melakukan *cropping* pada tiap-tiap karakter citra asli berdasarkan koordinat panjang dan lebar *bounding box* yang telah dibuat, sehingga keluaran dari proses ini adalah citra dari tiap-tiap karakter. Proses pencarian kontur dari tiap karakter diimplementasikan dengan *source code* berikut.

```
List<MatOfPoint> contours = new ArrayList<>();
                        Mat hierarchy = new Mat();
                        Imgproc.findContours(dilasi,
contours, hierarchy, Imgproc.RETR_EXTERNAL, Imgproc.
CHAIN_APPROX_SIMPLE);
```

Setelah mendapatkan citra per-karakter, maka citra akan diubah ukurannya dalam proses normalisasi karakter, menjadi citra berukuran 7×12. Hal ini diperlukan karena ukuran dari citra biner tiap-tiap karakter berbeda, sedangkan ukuran data masukan pada tahap LVQ haruslah sama, sehingga citra per-karakter perlu dikenai proses normalisasi karakter. Proses normalisasi karakter diimplementasikan dengan *source code* berikut.

```
Size sz = new Size(7, 12);
Imgproc.resize(label, resizeimage, sz);
```

Citra per-karakter yang telah memiliki ukuran yang sama kemudian dikenai dengan proses *sharpening, grayscalling,* dan binerisasi. Parameter-parameter yang digunakan dalam ketiga tahap tersebut sama seperti yang digunakan sebelumnya. Tahap ini dilakukan agar citra per-karakter yang akan dikenali, melalui tahap yang sama seperti yang dikenakan pada tahap *training*.

Proses selanjutnya yaitu ekstraksi ciri. Pada proses ini, citra biner per-karakter akan diubah menjadi *array* satu dimensi yang berisi kode rantai dari karakter yang diolah dengan ketentuan piksel berwarna hitam bernilai 0, dan piksel berwarna putih bernilai 1. Proses ekstraksi ciri ini dapat diimplementasikan dengan *source code* berikut.

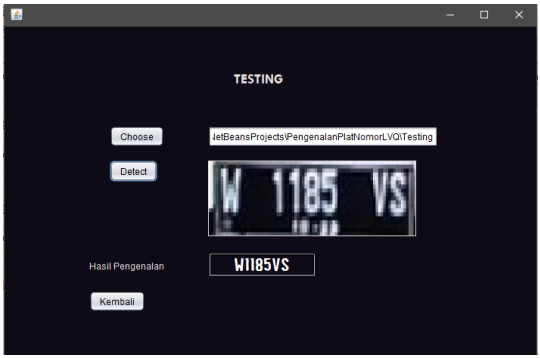
```
int width = resizeimage.rows();
int height = resizeimage.cols();
int a, b;
double arrayDataSingle[] = new double[width * height];
int arrayDataSingleInt[] = new int[width * height];
int pixelValue = 0;
int iterasi = 0;
for (a = 0; a < width; a++) {
    for (b = 0; b < height; b++) {
        double[] data = resizeimage.get(a, b);
        pixelValue = (int) data[0];
        arrayDataSingle[iterasi] = pixelValue == 0 ? 0 :
1;
```

```
        arrayDataSingleInt[iterasi] = (int)
arrayDataSingle[iterasi];
        System.out.print(arrayDataSingleInt[iterasi] + ",
");
        iterasi++;
    }
}
```

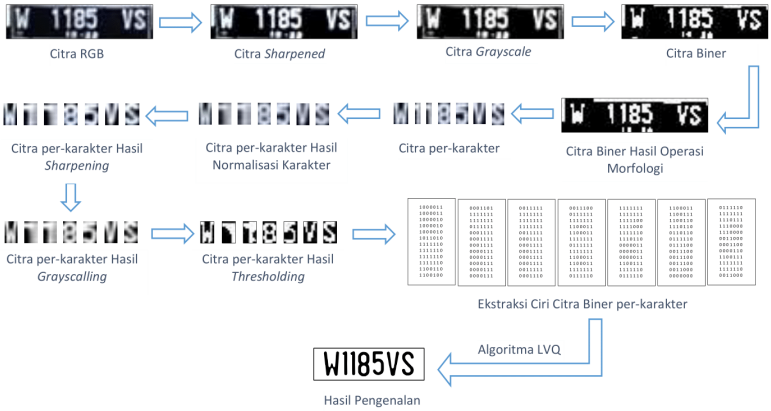
Proses selanjutnya yaitu proses perhitungan dengan Algoritma LVQ. Proses perhitungan Algoritma LVQ dalam proses pengujian ini dimulai dengan menghitung jarak Euclidean antara bobot acuan dengan bobot yang diuji. Bobot acuan yang digunakan adalah bobot akhir hasil *training* yang disimpan dalam berkas `bobotakhir.dat`. Proses *testing* oleh Algoritma LVQ diimplementasikan menjadi sebuah program ke dalam fungsi berikut.

```
public int lvqtest(double[] gambaruji, double[][]
gambaracuan)
```

Setelah berhasil mendapatkan hasil pengenalan dari tiap-tiap karakter, maka karakter-karakter tersebut diurutkan berdasarkan lokasi koordinat *bounding box*-nya, dan ditampilkan dalam kolom Hasil Pengenalan seperti pada Gambar 5.3. Secara umum proses *testing* dapat digambarkan seperti pada Gambar 5.4.



Gambar 5.3: Tampilan halaman proses *testing*



Gambar 5.4: Proses *Testing*

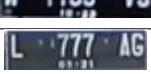
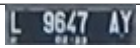
5.3 Pembahasan Hasil Uji Coba

5.3.1 Pembahasan Tingkat Akurasi Sistem

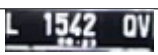
Uji coba dilakukan pada 25 citra plat nomor. Dalam proses pengujian, terlihat bahwa pengenalan citra plat nomor

didasarkan pada apakah karakter berhasil dikenali dengan baik atau tidak. Berdasarkan uji coba, didapatkan hasil yang optimal dengan menggunakan *maxEpoch* 30 dan *learning rate* 0.05 seperti yang ditampilkan pada Tabel 5.1.

Tabel 5.1: Hasil uji coba

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	6	BENAR
3		L1154CO	7	BENAR
4		31936KK0	6	SALAH
5		L1012CN	7	BENAR
6		B2227KKF	8	BENAR
7		L1699SO	7	BENAR
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L9647AY	7	BENAR
11		L1371HG	7	BENAR
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L174Z1N	5	SALAH
15		L1875CK	7	BENAR

Lanjutan **Tabel 5.1**

16		L1994FF	7	BENAR
17		L1163PR	7	BENAR
18		NT766TN	5	SALAH
19		L1792L	6	BENAR
20		L1602MN	7	BENAR
21		N911YF	6	BENAR
22		W1540VK	7	BENAR
23		N1483TN	7	BENAR
24		L1240OF	7	BENAR
25		L1542QV	7	BENAR
Total			168	22

Berdasarkan hasil pengujian terhadap 25 citra plat nomor, maka dapat dihitung tingkat akurasi dari sistem pengenalan plat nomor kendaraan sebagai berikut:

$$\begin{aligned} \text{Akurasi} &= \frac{\text{jumlah plat yang dikenali benar}}{\text{jumlah plat}} \times 100\% \\ &= \frac{22}{25} \times 100\% \\ &= 88\% \end{aligned}$$

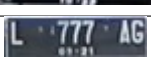
Sedangkan apabila dilihat dari pengujian per-karakter, berdasarkan hasil pengujian terhadap total 174 karakter dari 25 plat nomor, maka dapat dihitung tingkat akurasi dari pengenalan karakter plat nomor kendaraan sebagai berikut:

$$
\begin{aligned}
\text{Akurasi} &= \frac{\text{jumlah karakter yang dikenali benar}}{\text{jumlah karakter}} \times 100\% \\
&= \frac{168}{174} \times 100\% \\
&= 96.55\%
\end{aligned}
$$

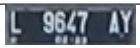
5.3.2 Pembahasan Mengenai Tahap Tanpa *Preprocessing* Tertentu

Data yang digunakan dalam penelitian ini adalah citra plat nomor yang memiliki resolusi rendah, maka sebelum memasuki tahap pengenalan, perlu dilakukan *preprocessing* tertentu sehingga nantinya citra dapat dikenali dengan baik. Dalam penelitian ini terdapat dua tahap *preprocessing* yang sangat berpengaruh yaitu *sharpening* dan operasi morfologi. Tahap *sharpening* merupakan proses penajaman citra yang digunakan dalam meningkatkan kontras dan cahaya pada citra. Sedangkan tahap operasi morfologi digunakan untuk memisahkan karakter yang terlihat berdekatan, agar karakter dapat melalui proses segmentasi dengan baik. Berikut pada Tabel 5.2 ditampilkan hasil dari pengenalan plat nomor tanpa menggunakan tahap *sharpening* dan operasi morfologi.

Tabel 5.2: Hasil uji coba tanpa *preprocessing* tertentu

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W11VS	5	SALAH
2		LTTTAG	3	SALAH
3		L1154CO	7	BENAR

Lanjutan **Tabel 5.2**

4		B1B36K8	5	SALAH
5		L1CN	4	SALAH
6		BZZKKF	4	SALAH
7		L16B330	3	SALAH
8		L1EM33E	4	SALAH
9		IL14Z6OC	6	SALAH
10		L964TAT	5	SALAH
11		L13T1MG	5	SALAH
12		L134TB8	4	SALAH
13		LHK	3	SALAH
14		L1TNM	2	SALAH
15		L18T5CK	6	SALAH
16		L19B4FF	6	SALAH
17		L1T63FR	5	SALAH
18		N1T66TK	4	SALAH
19		L1TBZL	3	SALAH
20		L16U7MH	4	SALAH
21		M9N19TYF	5	SALAH
22		W1546VK	6	SALAH
23		H1493TM	4	SALAH
24		L1Z466F	4	SALAH
25		L1542QV	7	BENAR
Total			114	2

Berdasarkan hasil pengujian terhadap 25 citra plat nomor tanpa menggunakan tahap *sharpening* dan operasi morfologi, terlihat plat yang benar dikenali hanya 2. Hal ini disebabkan karena terdapat beberapa karakter yang menempel, sehingga karakter tidak dapat lolos dalam proses seleksi ukuran di segmentasi. Selain itu operasi morfologi juga dapat menghilangkan *noise*, yang jika tidak dihilangkan dapat memengaruhi segmentasi objek dan menghasilkan pengenalan yang salah. Sedangkan apabila dilihat dari pengujian per-karakter, berdasarkan hasil pengujian terhadap total 174 karakter dari 25 plat nomor tanpa menggunakan tahap *sharpening* dan operasi morfologi, maka dapat dihitung tingkat akurasi dari pengenalan karakter sebagai berikut:

$$\begin{aligned} \text{Akurasi} &= \frac{\text{jumlah karakter yang dikenali benar}}{\text{jumlah karakter}} \times 100\% \\ &= \frac{114}{174} \times 100\% \\ &= 65.52\% \end{aligned}$$

5.3.3 Pembahasan Mengenai Pengaruh *maxEpoch* dan *Learning Rate*

1. Pengaruh *Learning Rate*

   Learning rate dalam algoritma LVQ merupakan konstanta yang digunakan untuk mencari bobot. Rumus untuk mencari bobot pada algoritma LVQ dituliskan pada Persamaan (5.1) berikut.

   $$w_{i,j}(\text{baru}) = w_{i,j}(\text{lama}) \pm \alpha(x_{i,j} - w_{i,j}(\text{lama})) \quad (5.1)$$

   dimana

   - $w_{i,j}$ = bobot acuan dari kelas ke-j piksel ke-i
   - $x_{i,j}$ = bobot yang dilatih ke-j piksel ke-i

- α = *learning rate*

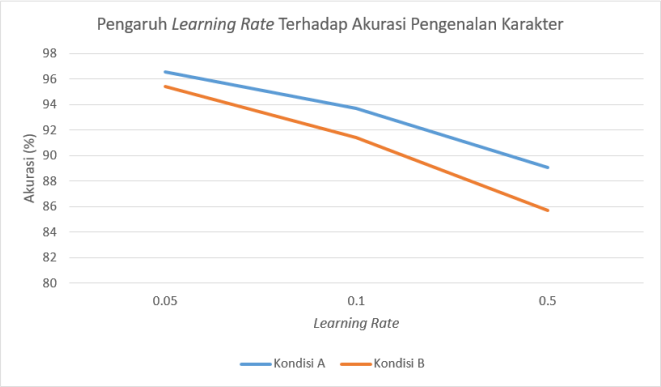
Untuk mengetahui pengaruh *learning rate* terhadap akurasi maka sistem dilatih dengan memasukkan nilai *maxEpoch* sama dan *learning rate* yang berbeda-beda[21]. Nilai *learning rate* yang dipilih sebagai sampel adalah 0.5, 0.1 dan 0.05. Sistem akan diuji dengan dua kondisi yang berbeda. Kondisi pertama (yang selanjutnya akan disebut Kondisi A) adalah ketika pada tahap *training* dan *testing* menggunakan *sharpening*, serta proses *cropping* per-karakter dikenakan pada citra asli. Sedangkan kondisi kedua (yang selanjutnya akan disebut Kondisi B) adalah ketika pada tahap *training* dan *testing* tidak menggunakan *sharpening*, serta proses *cropping* per-karakter dikenakan pada citra biner. Tabel 5.3 dan 5.4 adalah hasil pengujian dengan *learning rate* yang berbeda-beda pada Kondisi A dan Kondisi B.

Tabel 5.3: Pengaruh *Learning Rate*pada Kondisi A terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra

Learning Rate	*maxEpoch*	Akurasi Pengenalan Karakter	Rata-rata Waktu *Training* per-citra
0.05	30	96.55%	33 s
0.1	30	93.68%	28 s
0.5	30	89.08%	26 s

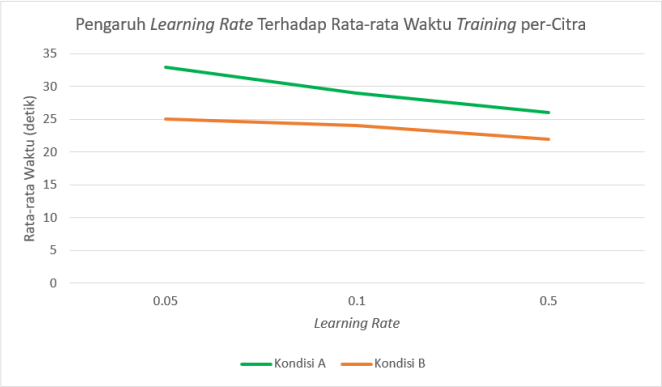
Tabel 5.4: Pengaruh *Learning Rate* pada Kondisi B terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra

Learning Rate	*maxEpoch*	Akurasi Pengenalan Karakter	Rata-rata Waktu *Training* per-citra
0.05	30	95.43%	25 s
0.1	30	91.43%	24 s
0.5	30	85.72%	22 s



Gambar 5.5: Diagram pengaruh *Learning Rate* terhadap akurasi pengenalan karakter

Pada Gambar 5.5 tersebut diketahui bahwa semakin besar *learning rate* mengakibatkan semakin rendah tingkat akurasi pengenalan karakternya. Sedangkan dalam segi kecepatan *training* seperti pada Gambar 5.6, semakin besar nilai *learning rate* maka semakin cepat proses yang dijalankan, namun memiliki perbedaan yang tidak terlalu jauh. Selain itu dapat dilihat bahwa



Gambar 5.6: Diagram pengaruh *Learning Rate* terhadap rata-rata waktu *training* per-citra

hasil akurasi pada Kondisi A lebih baik dibandingkan hasil akurasi pada Kondisi B, namun jika dilihat dari segi rata-rata waktu *training* per-citra, Kondisi A membutuhkan waktu yang lebih lama dibandingkan dengan Kondisi B. Hal ini disebabkan karena adanya penambahan proses *sharpening* dalam sistem. Sehingga untuk penelitian ini digunakan *learning rate* sebesar 0.05.

2. Pengaruh *maxEpoch* (Maksimum Iterasi)
   Untuk mengetahui pengaruh *maxEpoch* terhadap akurasi maka sistem dilatih dengan memasukkan nilai *learning rate* sama dan *maxEpoch* yang berbeda-beda. Nilai *maxEpoch* yang dipilih adalah 10, 20, dan 30. Sistem akan diuji dengan dua kondisi yang berbeda. Kondisi pertama (yang selanjutnya akan disebut Kondisi A) adalah ketika pada tahap *training* dan *testing* menggunakan *sharpening*, serta proses *cropping* per-

karakter dikenakan pada citra asli. Sedangkan kondisi kedua (yang selanjutnya akan disebut Kondisi B) adalah ketika pada tahap *training* dan *testing* tidak menggunakan *sharpening*, serta proses *cropping* per-karakter dikenakan pada citra biner. Tabel 5.5 dan 5.6 adalah hasil pengujian dengan *maxEpoch* yang berbeda-beda pada Kondisi A dan Kondisi B.

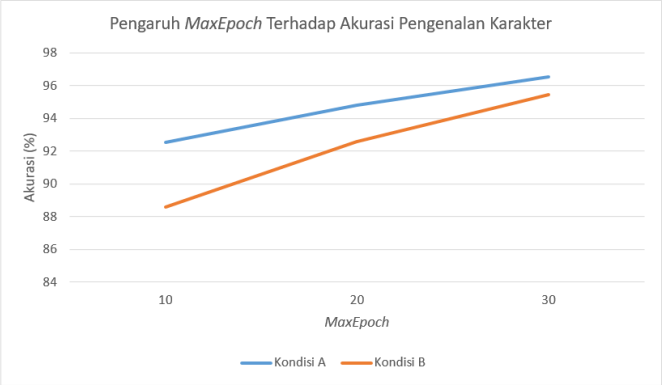
Tabel 5.5: Pengaruh *MaxEpoch* pada Kondisi A terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra

maxEpoch	*Learning Rate*	Akurasi Pengenalan Karakter	Rata-rata Waktu *Training* per-citra
10	0.05	92.53%	29 s
20	0.05	94.83%	32 s
30	0.05	96.55%	33 s

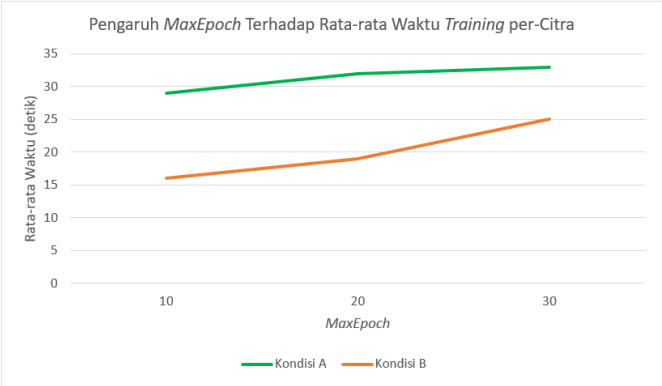
Tabel 5.6: Pengaruh *MaxEpoch* pada Kondisi B terhadap akurasi pengenalan karakter dan rata-rata waktu *training* per-citra

maxEpoch	*Learning Rate*	Akurasi Pengenalan Karakter	Rata-rata Waktu *Training* per-citra
10	0.05	88.57%	16 s
20	0.05	92.57%	19 s
30	0.05	95.43%	25 s

Pada Gambar 5.7 diketahui bahwa semakin besar *maxEpoch* maka semakin tinggi tingkat akurasinya.



Gambar 5.7: Diagram pengaruh *MaxEpoch* terhadap akurasi pengenalan karakter



Gambar 5.8: Diagram pengaruh *MaxEpoch* terhadap rata-rata waktu *training* per-citra

Sedangkan dalam segi kecepatan *training* seperti pada Gambar 5.8, semakin besar nilai *maxEpoch* maka

semakin lambat proses yang dijalankan. Selain itu dapat dilihat bahwa hasil akurasi pada Kondisi A lebih baik dibandingkan hasil akurasi pada Kondisi B, namun jika dilihat dari segi kecepatan *training* per-citra, Kondisi A membutuhkan waktu yang lebih lama dibandingkan dengan Kondisi B. Hal ini disebabkan karena adanya penambahan proses *sharpening* dalam sistem. Sehingga dalam penelitian ini digunakan *maxEpoch* sebesar 30.

5.3.4 Pembahasan Mengenai Pengujian Terhadap Plat Nomor Berbeda Warna

Citra plat nomor yang digunakan dalam penelitian ini adalah plat nomor yang memiliki dasar hitam dengan tulisan putih. Untuk plat nomor dengan warna lain, hasil pengujiannya ditampilkan pada Tabel 5.7.

Tabel 5.7: Hasil uji coba plat nomor berbeda warna

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		K2UYF	2	SALAH
2		-	0	TIDAK TERKENALI

Dapat dilihat bahwa pengujian dengan plat nomor dasar merah dan tulisan putih masih dapat menghasilkan hasil pengenalan walaupun karakter yang dikenali masih ada yang salah. Hal ini dapat disebabkan karena terdapat perbedaan intensitas warna dalam proses *sharpening*, yang akan mempengaruhi tahap-tahap selanjutnya, sehingga hasil pengenalan yang dihasilkan masih kurang akurat. Sedangkan untuk plat dengan dasar kuning dan tulisan hitam tidak

dapat dikenali sama sekali, karena pada proses binerisasi dalam penelitian ini karakter diwakili dengan piksel berwarna putih dan *background* berwarna hitam, sedangkan dalam plat dengan dasar kuning dan tulisan hitam, hasil dari proses binerisasi adalah karakter diwakili oleh piksel berwarna hitam dan *background* berwarna putih. Hal ini menyebabkan tidak terdeteksinya karakter sehingga plat tidak dapat dikenali.

BAB VI
PENUTUP

Pada bab ini berisi kesimpulan yang diperoleh dari pembahasan pada bab sebelumnya serta saran untuk pengembangan penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program dapat diambil kesimpulan sebagai berikut :

1. Dari hasil uji coba pengenalan plat nomor kendaraan bergerak dengan menggunakan tahap *preprocessing sharpening*, *grayscalling*, binerisasi dengan nilai *thresh* 170, operasi morfologi dengan kernel $(1, 2)$, segmentasi, normalisasi karakter, *preprocessing* karakter dan ekstraksi ciri terbukti bahwa metode *Learning Vector Quantization* mampu melakukan pengenalan terhadap plat nomor kendaraan beresolusi rendah dan mengonversinya ke dalam teks.

2. Keakuratan pengenalan plat nomor kendaraan beresolusi rendah dengan metode *Learning Vector Quantization* sebesar 88% untuk pengenalan plat dan 96.55% untuk pengenalan karakter dengan menggunakan *learning rate* sebesar 0.05 dan *maxEpoch* 30.

6.2 Saran

Adapun saran untuk penelitian selanjutnya yaitu agar dapat dilakukan penelitian dengan data plat nomor yang dikenali bukan hanya plat nomor yang memiliki dasar hitam

dan tulisan putih saja, namun juga plat nomor dengan warna lain.

LAMPIRAN

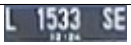
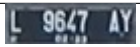
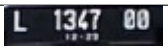
Lampiran 1. Vektor dari Bobot Acuan

No.	Vektor	Karakter
1.	0 1 1 1 1 1 ..... 0 0 1 1 1 0 0	0
2.	0 0 1 1 1 1 ..... 0 0 0 1 1 1 0	1
3.	1 1 1 1 1 1 ..... 1 1 1 1 1 1 1	2
4.	0 0 1 1 0 0 ..... 0 0 1 1 1 1 1	3
5.	0 0 0 0 1 1 ..... 0 0 0 0 1 1 1	4
6.	1 1 1 1 1 1 ..... 0 1 1 1 1 1 1	5
7.	0 1 1 1 1 0 ..... 0 1 1 1 1 1 1	6
8.	1 1 1 1 1 1 ..... 0 0 1 1 1 0 0	7
9.	1 1 1 1 1 1 ..... 1 1 1 1 1 1 1	8
10.	0 1 1 1 1 1 ..... 1 1 1 1 1 1 0	9
11.	0 0 1 1 1 1 ..... 1 1 0 0 1 1 1	A
12.	1 1 1 1 1 1 ..... 1 1 1 1 1 1 1	B
13.	0 1 1 1 1 1 ..... 0 0 1 1 1 1 1	C
14.	1 1 1 1 1 0 ..... 0 1 1 1 1 0 0	D
15.	1 1 1 1 1 1 ..... 1 1 1 1 1 1 1	E
16.	1 1 1 1 1 1 ..... 1 1 1 0 0 0 0	F
17.	0 0 1 1 1 0 ..... 0 1 1 1 1 1 0	G
18.	1 1 0 0 0 1 ..... 1 1 0 0 1 1 1	H
19.	1 1 1 1 1 1 ..... 1 1 1 1 1 1 1	I
20.	0 0 0 0 1 1 ..... 0 1 1 1 1 1 0	J
21.	1 1 1 0 0 0 ..... 1 1 1 0 0 1 1	K
22.	1 1 0 0 0 0 ..... 1 1 1 1 1 1 1	L
23.	1 1 1 0 0 0 ..... 1 0 0 0 0 1 1	M
24.	1 1 0 0 0 1 ..... 1 1 0 0 1 1 1	N
25.	0 0 1 1 0 0 ..... 0 1 1 1 1 1 0	O
26.	1 1 1 1 1 1 ..... 1 1 1 0 0 0 0	P

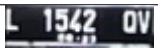
Lanjutan lampiran 1

27.	0 0 1 1 0 0 ..... 0 1 1 1 1 1 1	Q
28.	1 1 1 1 1 0 ..... 1 1 1 0 0 1 1	R
29.	0 1 1 1 0 0 ..... 0 0 1 1 1 0 0	S
30.	1 1 1 1 1 1 ..... 0 0 1 1 1 0 0	T
31.	1 1 0 0 0 1 ..... 0 0 1 1 1 0 0	U
32.	1 1 0 0 0 1 ..... 0 0 0 1 1 0 0	V
33.	1 1 0 0 0 1 ..... 0 1 1 0 1 1 0	W
34.	1 1 0 0 0 1 ..... 1 1 0 0 0 1 1	X
35.	1 1 0 0 0 1 ..... 0 0 1 1 1 0 0	Y
36.	1 1 1 1 1 1 ..... 1 1 1 1 1 1 1	Z

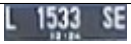
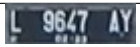
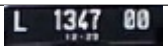
Lampiran 2. Data Hasil Uji Coba Kondisi A dengan *Learning Rate* 0.1 dan *MaxEpoch* 30

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	6	BENAR
3		L1154CO	7	BENAR
4		31936KK0	6	SALAH
5		L101ZCN	6	SALAH
6		B2227KLF	7	SALAH
7		L16996O	6	SALAH
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L9647AY	7	BENAR
11		L1371HG	7	BENAR
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L174ZNN	5	SALAH
15		L18756K	6	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR
18		NT766TN	5	SALAH

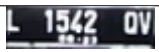
Lanjutan lampiran 2

19		L1792L	6	BENAR
20		L1602HN	6	SALAH
21		N911YF	6	BENAR
22		W1540VK	7	BENAR
23		N1483TN	7	BENAR
24		L1240OF	7	BENAR
25		L1542QV	7	BENAR
Total			163	17

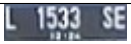
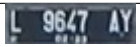
Lampiran 3. Data Hasil Uji Coba Kondisi A dengan *Learning Rate* 0.5 dan *MaxEpoch* 30

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	6	BENAR
3		L1154CO	7	BENAR
4		31936KK0	6	SALAH
5		L101ZCN	6	SALAH
6		BZZZ7KLF	4	SALAH
7		L16996O	6	SALAH
8		L1533SE	7	BENAR
9		L14Z6OC	6	SALAH
10		L9647AY	7	BENAR
11		L1371HG	7	BENAR
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L174ZNN	5	SALAH
15		L18756K	6	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR
18		NT766TN	5	SALAH

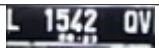
Lanjutan lampiran 3

19		L179ZL	5	SALAH
20		L160ZHN	5	SALAH
21		N911YF	6	BENAR
22		W1540VK	7	BENAR
23		N1483TN	7	BENAR
24		L1Z40OF	6	SALAH
25		L154ZQV	6	SALAH
Total			155	13

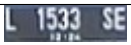
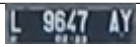
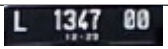
Lampiran 4. Data Hasil Uji Coba Kondisi A dengan *MaxEpoch* 10 dan *Learning Rate* 0.05

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	6	BENAR
3		L1154CO	7	BENAR
4		31936KK0	6	SALAH
5		L101ZCN	6	SALAH
6		B2227KLF	7	SALAH
7		L1699GO	6	SALAH
8		L1533SE	7	BENAR
9		L1426OG	6	SALAH
10		L9B47AY	6	SALAH
11		L1371H0	6	SALAH
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T4Z1N	4	SALAH
15		L1875CK	7	BENAR
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR
18		NT766TN	5	SALAH

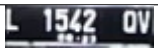
Lanjutan lampiran 4

19		L1792L	6	BENAR
20		L1602MN	7	BENAR
21		N911YF	6	BENAR
22		W1540VK	7	BENAR
23		N1483TN	7	BENAR
24		L1240OF	7	BENAR
25		L1542QV	7	BENAR
Total			161	16

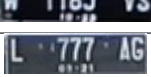
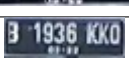
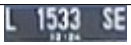
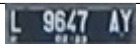
Lampiran 5. Data Hasil Uji Coba Kondisi A dengan *MaxEpoch* 20 dan *Learning Rate* 0.05

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	5	BENAR
3		L1154CO	7	BENAR
4		31936KK0	6	SALAH
5		L101ZCN	6	SALAH
6		B2227KKF	8	BENAR
7		L1699SO	7	BENAR
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L9B47AY	6	SALAH
11		L1371HG	7	BENAR
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T4Z1N	4	SALAH
15		L1875CK	7	BENAR
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR
18		NT766TN	5	SALAH

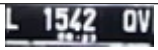
Lanjutan lampiran 5

19		L1792L	7	BENAR
20		L1602MN	7	BENAR
21		N911YF	6	BENAR
22		W1540VK	7	BENAR
23		N1483TN	7	BENAR
24		L1240OF	7	BENAR
25		L1542QV	7	BENAR
Total			165	20

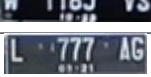
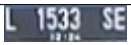
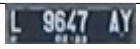
Lampiran 6. Data Hasil Uji Coba Kondisi B dengan *MaxEpoch* 30 dan *Learning Rate* 0.05

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	6	BENAR
3		L1154CO	7	BENAR
4		B1936KK0	7	SALAH
5		L1012CN	7	BENAR
6		B2227KKF	8	BENAR
7		L1699SO	7	BENAR
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L9647AY	7	BENAR
11		L1371HG	7	BENAR
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T42NN	5	SALAH
15		L19T5CK	5	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR

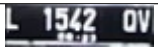
Lanjutan Lampiran 6

18		N17O67N	5	SALAH
19		L1792L	6	BENAR
20		L1602MN	7	BENAR
21		N911YF	6	BENAR
22		W1540VK	7	BENAR
23		N1493TN	6	SALAH
24		L1240OF	7	BENAR
25		L1542QV	7	BENAR
Total			167	20

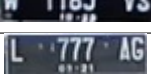
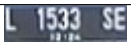
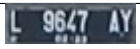
Lampiran 7. Data Hasil Uji Coba Kondisi B dengan *MaxEpoch* 20 dan *Learning Rate* 0.05

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777A6	5	SALAH
3		L1154CO	7	BENAR
4		81936KK0	6	SALAH
5		L1012CN	7	BENAR
6		82227KKF	7	SALAH
7		L1699SO	7	BENAR
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L9647AT	6	SALAH
11		L1371HG	7	BENAR
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T42NN	5	SALAH
15		L19T5CK	5	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR

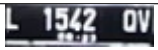
Lanjutan Lampiran 7

18		N17O67N	5	SALAH
19		L1792L	6	BENAR
20		L1602MN	7	BENAR
21		N911TF	5	SALAH
22		W1540VK	7	BENAR
23		N1493TN	6	SALAH
24		L1240OF	7	BENAR
25		L1542QV	7	BENAR
Total			162	16

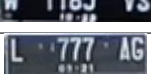
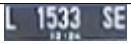
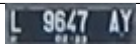
Lampiran 8. Data Hasil Uji Coba Kondisi B dengan *MaxEpoch* 10 dan *Learning Rate* 0.05

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777A6	5	SALAH
3		L1154CO	7	BENAR
4		81938KK0	5	SALAH
5		L1012CN	7	BENAR
6		82227KKF	7	SALAH
7		L1689SO	6	SALAH
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L9647AT	6	SALAH
11		L1371H6	6	SALAH
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T42NN	5	SALAH
15		L19T5CK	5	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR

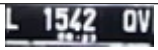
Lanjutan Lampiran 8

18		N17O67N	5	SALAH
19		L1792L	6	BENAR
20		L1802NN	5	SALAH
21		N911TF	5	SALAH
22		W1540VK	7	BENAR
23		N1493TN	6	SALAH
24		L1240OF	7	BENAR
25		L15420V	6	SALAH
Total			155	12

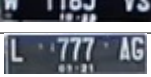
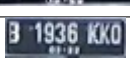
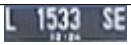
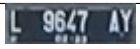
Lampiran 9. Data Hasil Uji Coba Kondisi B dengan *MaxEpoch* 30 dan *Learning Rate* 0.5

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L7T7A6	4	SALAH
3		L1154CO	7	BENAR
4		81936KK0	6	SALAH
5		L1012CN	7	BENAR
6		8222TKKF	6	SALAH
7		L16895O	5	SALAH
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L964TAT	5	SALAH
11		L13T1H6	5	SALAH
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T42NN	5	SALAH
15		L19TSCK	4	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR

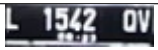
Lanjutan Lampiran 9

18		N17O67N	5	SALAH
19		L1T92L	5	BENAR
20		L1802NN	5	SALAH
21		N811TF	4	SALAH
22		W1540VK	7	BENAR
23		N1493TN	6	SALAH
24		L1240OF	7	BENAR
25		L1542QV	6	BENAR
Total			150	13

Lampiran 10. Data Hasil Uji Coba Kondisi B dengan *MaxEpoch* 30 dan *Learning Rate* 0.1

No.	Citra Plat Nomor	Hasil Pengenalan	Jumlah Karakter Dikenali Benar	Plat Terkenali
1		W1185VS	7	BENAR
2		L777AG	6	BENAR
3		L1154CO	7	BENAR
4		81936KK0	6	SALAH
5		L1012CN	7	BENAR
6		8222TKKF	6	SALAH
7		L1699SO	7	BENAR
8		L1533SE	7	BENAR
9		L1426OC	7	BENAR
10		L964TAY	6	SALAH
11		L13T1H6	5	SALAH
12		L1347OO	7	BENAR
13		L1483HK	7	BENAR
14		L1T42NN	5	SALAH
15		L19TSCK	4	SALAH
16		L1994FF	7	BENAR
17		L1163PR	7	BENAR

Lanjutan Lampiran 10

18		N17O67N	5	SALAH
19		L1792L	6	BENAR
20		L1602MN	7	BENAR
21		N911YF	4	SALAH
22		W1540VK	7	BENAR
23		N1493TN	6	SALAH
24		L1240OF	7	BENAR
25		L1542QV	6	BENAR
Total			160	17

DAFTAR PUSTAKA

[1] Amanda, Gita. (2019). "Jumlah Kecelakaan Lalu Lintas Naik 3 persen pada 2019". RepublikaNews: https://nasional.republika.co.id/berita/q37i3n423/jumlah-kecelakaan-lalu-lintas-naik-3-persen-pada-2019 (diakses 1 Juli 2020)

[2] Abdi, Nailul Mustaqim, Aisyah, Siti, Arnia, Fitri. (2011). "Peningkatan Kualitas Citra Digital Menggunakan Metode Super Resolusi Pada Domain Spasial". Jurnal Rekayasa Elektrika Vol. 9, No. 3.

[3] Gunawan, Irwan Prasetya, Kalembang, Erik, Usman, Koredianto. (2010). "Identifikasi Distorsi Blur pada Gambar Digital". Jurnal Ilmu Komputer dan Informasi, Vol.3, No.1

[4] Qadri M.T, Asif M. (2009). "Automatic Number Plate Recogniton System for Vehicle Identification Using Optical Character Recognition". International Conference onf Education Technology and Computer.

[5] Sugiarto, Edi. (2013). "Pengenalan Karakter pada Plat Nomor Kendaraan Berbasis Support Vector Machine". Techno.com: Vol. 12 No. 1.

[6] Nagare, Anuja P. (2011). "License Plate Character Recognition System using Neural Network". International Journal of Computer Applications Vol. 25 No. 10

[7] Solichin, Achmad, Rahman, Zulfikar. (2015). "Identifikasi Plat Nomor Kendaraan Berbasis Mobile dengan Metode Learning Vector Quantization". TICOM: Vol. 3 No. 3.

[8] Ahamdie Labique, Beryl, Wahyu Widodo, Agus, Utaminingrum, Fitri. (2018). "Pengenalan Plat Nomor Mobil Menggunakan Metode Learning Vector Quantization". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer: Vol.2 No. 3

[9] PMJ, A. (2017). "Agar Tidak Ditilang Karena Masalah Plat Nomor". TribataNews: http://tribratanews.polri.go.id/?p=249778 (diakses 21 November 2019)

[10] Rafael C., Gonzales. (2002). "Digital Image Processing". Prentice Hall: New Jersey.

[11] Putra, D. (2010). "Pengolahan Citra Digital". Yogyakarta: Penerbit Andi

[12] Yadav, D., Sanchez, S., Jorge, M. (2013). "Optical Character Recognition for Hindi Language". Journal of Information Processing Systems, Vol. 9, No.1, 117-140

[13] Fausett, L. (1994). "Fundamentals of Neural Networks: Architectures, Algorithms, and Application". Prentice Hall International.Inc.

[14] Al-Ameen, Zohair, Muttar, Alaa, Al-Badrani, Ghofran. (2019). "Improving the Sharpness of Digital Image Using an Amended Unsharp Mask Filter". International Journal of Image, Graphics, and Signal Processing.

[15] Purwocoroko, Yesaya, Karmilasari, Novrina. (2014). "Komparasi Proses Deblurring Citra Antara Algoritma Lucy-Richardson dan Wiener Filter". Jurnal Ilmiah Komputasi Universitas Gunadarma, Vol. 13, No. 2.

[16] Kusban, Muhammad. (2011). "Restorasi Image dengan metode Wiener dan Lucy-Richardson". Universitas Muhammadiyah Surakarta.

[17] A. M. A. M. Akbar. (2016) "Traffic ip camera for counting four-wheeled vehicle using area of pixels method". Institut Teknologi Sepuluh Nopember.

[18] Suzuki, S., Abe, K. (1985). "Topological Structural Analysis of Digitized Binary Images by Border Following". CVGIP 30 1, pp. 32-46.

[19] Chandra, H. (2013). "Implementasi Algoritma Douglas-Peucker untuk Minimalisasi Data Jalur pada Aplikasi Pelacakan Lokasi". Tangerang: Universitas Multimedia Nusantara.

[20] Christantio, Ricky, Sanubari, Junibakti, Timotius, Ivanna K. (2006). "Peningkatan Resolusi Citra Digital dengan Interpolasi Bilinear". Salatiga: Universitas Kristen Satya Wacana.

[21] Asworo. (2010). "Comparison Between Kohonen Neural Network Method and Learning Vector Quantization in The Online Handwritting Recognition System". Surabaya: Institut Teknologi Sepuluh Nopember.

BIODATA PENULIS

Penulis memiliki nama lengkap Shafira Herlinawati dan lahir di Surabaya, 17 Juli 1998. Pendidikan penulis bermula di SDN Sidoklumpuk Sidoarjo, SMP Negeri 5 Sidoarjo, dan SMA Negeri 1 Sidoarjo. Setelah lulus SMA pada tahun 2016, penulis mengenyam kehidupan mahasiswa di Departemen Matematika, Fakultas Sains dan Analitika Data (FSAD) ITS melalui jalur SNMPTN. Selama proses perkuliahan, penulis menekuni bidang minat Ilmu Komputasi dan Ilmu Komputer. Selain berkuliah, penulis aktif mengikuti kegiatan intra dan ekstra kampus. Untuk organisasi intra kampus, penulis pernah menjabat sebagai Staff Personalia KOPMA dr. Angka ITS XXVIII, Staff Question Maker OMITS 12th-13th, Staff Kaderisasi LDJ Ibnu Muqlah, Sekretaris Departemen Kaderisasi LDJ Ibnu Muqlah, Staff Kaderisasi JMMI ITS, Midle Kaderisasi JMMI ITS, dan Ketua Badan Khusus Kemuslimahan JMMI ITS. Untuk organisasi ekstra kampus, penulis menjadi mentor di Himpunan Alumni SMANISDA dan anggota internal Data Science Indonesia region Jawa Timur.

Apabila ada pertanyaan dan saran, bisa menghubungi penulis via email *shafiralina17@gmail.com* atau LinkedIn *Shafira Herlinawati*.