

**PERANGKAT LUNAK PENGENALAN PLAT NOMOR
MOBIL MENGGUNAKAN JARINGAN KOMPETITIF
DAN JARINGAN KOHONEN**

SKRIPSI

Oleh

PATARDO MARASI MANURUNG

04 03 03 711 4



**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GANJIL 2007/2008**

**PERANGKAT LUNAK PENGENALAN PLAT NOMOR
MOBIL MENGGUNAKAN JARINGAN KOMPETITIF
DAN JARINGAN KOHONEN**

SKRIPSI

Oleh

PATARDO MARASI MANURUNG

04 03 03 711 4



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GANJIL 2007/2008**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

PERANGKAT LUNAK PENGENALAN PLAT NOMOR MOBIL MENGUNAKAN JARINGAN KOMPETITIF DAN JARINGAN KOHONEN

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 14 Desember 2007

Patardo Marasi Manurung

NPM. 04 03 03 711 4

PENGESAHAN

Skripsi dengan judul :

PERANGKAT LUNAK PENGENALAN PLAT NOMOR MOBIL MENGUNAKAN JARINGAN KOMPETITIF DAN JARINGAN KOHONEN

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian skripsi. Skripsi ini telah diujikan pada sidang ujian skripsi pada tanggal 2 Januari 2008 dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 7 Januari 2008

Dosen Pembimbing 1,

Dosen Pembimbing 2,

Dr. Ir. Dodi Sudiana M.Eng

Dr. Ir. Arman Djohan Diponegoro M.Eng

NIP. 131 944 413

NIP. 131 476 472

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

Dr.Ir. Dodi Sudiana , M.Eng

Dr. Ir. Arman Djohan Diponegoro, M.Eng

sebagai dosen pembimbing I dan II yang telah memberikan bimbingan, arahan, dan persetujuan sehingga skripsi ini dapat diselesaikan dengan baik.

Patardo Marasi Manurung
NPM 04 03 03 711 4
Departemen Teknik Elektro

Dosen Pembimbing
I. Dr. Ir. Dodi Sudiana M.Eng
II. Dr. Ir. Arman Djohan Diponegoro M.Eng

**PERANGKAT LUNAK PENGENALAN PLAT NOMOR MOBIL MENGGUNAKAN
JARINGAN KOMPETITIF DAN JARINGAN KOHONEN**

ABSTRAK

Skripsi ini dibuat untuk merancang dan mengimplementasikan perangkat lunak yang dapat mengenali karakter (berupa angka dan huruf) pada plat nomor mobil pribadi di Indonesia dari citra hasil pemotretan kamera digital dengan menggunakan jaringan syaraf tiruan tipe *Kohonen* dan kompetitif.sebagai metode pengenalan

Terdapat 3 tahapan proses yang digunakan dalam pembuatan perangkat lunak yaitu tahap *pre-processing*, tahap pelatihan jaringan syaraf tiruan tipe *kohonen* dan kompetitif dan tahap pengenalan. Pada tahap *pre-processing* bertujuan untuk mendeteksi lokasi dan ekstraksi plat nomor lalu mengekstaksi karakter huruf dan angka dari plat nomor. Untuk mendeteksi plat nomor digunakan *assymetric filter (rank filter)* terhadap citra yang telah dilakukan proses *vertical edge detection*. Untuk mengekstraksi karakter angka dan huruf dilakukan proses *background equalization* terlebih dahulu. Pada tahap pelatihan jaringan bertujuan untuk pembentukan suatu database angka dan database huruf yang akan dibutuhkan pada tahap pengenalan. Perbedaan pola dari masing-masing angka dan huruf merupakan suatu ciri yang akan digunakan untuk data masukan jaringan syaraf tiruan. Tahap pengenalan merupakan suatu *pattern recognition* untuk mengenali angka dan huruf pada plat nomor. Jaringan syaraf tiruan tipe *Kohonen* digunakan untuk pengenalan huruf dan tipe kompetitif untuk pengenalan angka. Analisis yang dilakukan bertujuan untuk menentukan pengaruh jumlah sampel pada tahap pelatihan jaringan, pengaruh tipe jaringan syaraf tiruan, dan penyebab kesalahan pada tahap *pre-processing* dan tahap pengenalan.

Berdasarkan hasil simulasi, perangkat lunak pengenalan plat nomor yang dibuat telah berhasil mengenali pola huruf dengan tingkat akurasi 88,89% dan pola angka dengan tingkat akurasi 98.3% dan pengenalan plat dengan tingkat akurasi 60%

Kata Kunci: *Pre-Processing* , Pengenalan Plat Nomor Mobil, Jaringan Syaraf Tiruan, Kohonen, Kompetitif

Patardo Marasi Manurung
NPM 04 03 03 711 4
Departemen Teknik Elektro

Dosen Pembimbing
I. Dr. Ir. Dodi Sudiana M.Eng
II. Dr. Ir. Arman Djohan Diponegoro M.Eng

**CAR LICENSE PLATE NUMBER RECOGNITION SOFTWARE USING
COMPETITIVE AND KOHONEN NETWORK**

ABSTRACT

This final project is created to design and to implement software which can recognize Indonesian license plate number from digital camera image using competitive and *Kohonen* Neural Network as recognition method.

There are 3 process which are used in making software which are: pre-processing phase, training phase of kohonen and competitive neural network and recognition phase. The purposes of pre-processing phase are to be able to localize and to extract license plate then to extract number and letter from the license plate. The license plate is detected by applying *asymmetric rank filter* to the image that has been *vertical edge detected*. Background equalization process is needed first for the License Plate Character Segmentation. The purposes of training phase are able to make number database and letter database that will be needed in recognition phase. Pattern difference from each number and letter is a characteristic that will be used for input data of neural network. Recognition phase is a pattern recognition to recognize letter and number from license plate. Kohonen neural network is used to recognize letter and competitive neural network is used to recognize number. The final project analyzed the influence from a number of sample, the influence of neural network type, and cause of error in pre-processing phase and recognition phase.

From the simulation output, the software can achieve 88,89% accuracy in recognize the letter, 98.3% accuracy in recognize the number, and 60% accuracy in recognize plate

Key Word: Pre-Processing, License Plate Recognition, Neural Network, Kohonen, Competitive

DAFTAR ISI

	halaman
PERNYATAAN KEASLIAN SKRIPSI.....	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH	2
1.3 TUJUAN.....	3
1.4 METODOLOGI PENELITIAN.....	3
1.5 BATASAN MASALAH.....	3
1.6 SISTEMATIKA PENELITIAN.....	4
BAB II LANDASAN TEORI.....	6
2.1 PERATURAN PENOMORAN PLAT NOMOR MOBIL.....	6
2.1.1 Plat Nomor	6
2.1.1.1 Peraturan Penomoran Plat Nomor Mobil Pribadi.....	7
2.1.1.2 Desain Plat Nomor Pribadi	7
2.1.1.3 Lokasi Plat Nomor Pribadi.....	8
2.2 PEMROSESAN CITRA	9
2.2.1 Sampling dan Graylevel Quantization	9

2.2.2 Leveling dan Reshaping.....	10
2.2.3 Nilai Rata-Rata <i>Graylevel</i>	11
2.2.4 <i>Filtering</i>	11
2.2.5 <i>Edge Detection</i>	11
2.3 SISTEM PENGENALAN POLA.....	12
2.4 JARINGAN SARAF TIRUAN.....	16
2.5 JARINGAN SARAF TIRUAN <i>KOHONEN</i>	20
2.5.1 Arsitektur jaringan <i>Kohonen</i>	20
2.6 JARINGAN KOMPETITIF.....	23
BAB III METODOLOGI PENELITIAN.....	24
3.1 LANGKAH PEMECAHAN MASALAH.....	24
3.2 <i>PRE PROCESSING CITRA</i>	28
3.2.1 Pendeteksian dan Ekstraksi Plat Nomor.....	28
3.2.1.1 <i>Sampling</i>	28
3.2.1.3 <i>Grayscaleing Citra Mobil</i>	30
3.2.1.4 <i>Vertical Edge Detection</i>	30
3.2.1.5 <i>Cleaning Phase</i>	31
3.2.1.6 <i>Horizontal Rank Filtering</i>	32
3.2.1.7 <i>Area terbesar (High Density Of Bright Edges)</i>	32
3.2.1.8 <i>Ekstraksi Plat Nomor</i>	33
3.2.2 Background Equalization.....	33
3.2.2.1 <i>Imopen Background</i>	34
3.2.2.2 <i>Imsubtract Background</i>	34
3.2.2.3 <i>Grayscaleing</i>	35
3.2.2.4 <i>Contrasting</i>	35
3.2.2.5 <i>Proses Graytresh dan Binary</i>	35
3.2.2.6 <i>Proses filter</i>	36
3.2.2.7 <i>Imclearborder</i>	37
3.2.2.8 <i>Ekstraksi Karakter</i>	37
3.3 LABELISASI KARAKTER.....	38

3.4 PROSES PELATIHAN KARAKTER PLAT NOMOR DENGAN JARINGAN SYARAF TIRUAN KOHONEN	40
3.4.1 Pelatihan Terhadap Angka	40
3.4.2 Pelatihan Terhadap Huruf	41
3.5 PROSES PENGENALAN KARAKTER PLAT NOMOR	43
BAB IV HASIL UJI COBA DAN ANALISA	46
4.1 HASIL UJI COBA	46
4.1.1 Hasil Uji Coba Dengan Sampel Training 15 Buah	47
4.1.1.1 Pengenalan Dilakukan Dengan Jaringan Kompetitif.....	47
4.1.1.2 Pengenalan Dilakukan Dengan Jaringan Kohonen	48
4.1.2 Hasil Uji Coba Dengan Sampel Training 20 Buah	49
4.1.2.1 Pengenalan Dilakukan Dengan Jaringan Kompetitif.....	49
4.1.2.2 Pengenalan Dilakukan Dengan Jaringan Kohonen	50
4.1.2.3 Gabungan jaringan kompetitif dan jaringan Kohonen.....	51
4.2 PENGOLAHAN HASIL UJI COBA.....	52
4.3 ANALISA SISTEM.....	53
4.3.1 Pengaruh Jumlah Sampel <i>Training</i>	53
4.3.2 Pengaruh Metode Yang Digunakan Dalam Pengenalan	54
4.3.3 Analisa Kesalahan Pada Tahap <i>Pre-Processing</i>	55
4.3.4 Analisa Kesalahan Pada Tahap Identifikasi.....	55
BAB V KESIMPULAN.....	59
DAFTAR ACUAN	60
DAFTAR PUSTAKA	62
LAMPIRAN.....	64

DAFTAR GAMBAR

	halaman
Gambar 2.1 Citra plat nomor	8
Gambar 2.2 Citra dengan <i>grayscale level 2</i> (nilai 0 sampai 1).....	10
Gambar 2.3 <i>Leveling</i> citra menjadi hanya 2 tingkat (nilai 0 dan 1).....	10
Gambar 2.4 Grafik fungsi kerapatan dari ciri diameter jeruk dan apel.....	13
Gambar 2.5 Sistem pengenalan pola dengan pendekatan statistik.....	14
Gambar 2.6 Contoh pembagian kelas pola	15
Gambar 2.7 Sistem pengenalan pola dengan pendekatan sintatik	16
Gambar 2.8 Struktur unit jaringan syaraf tiruan	18
Gambar 2.9 Arsitektur jaringan kohonen.....	21
Gambar 2.10 w berupa vektor 1 dimensi	21
Gambar 2.11 w berupa vektor 2 dimensi	22
Gambar 2.12 Arsitektur jaringan kompetitif.....	23
Gambar 3.1 Blok Diagram Proses Sistem.....	24
Gambar 3.2 Diagram alir <i>pre-processing</i>	25
Gambar 3.3 Diagram alir proses <i>training</i>	26
Gambar 3.4 Diagram alir proses identifikasi	27
Gambar 3.5 Hasil sampling citra.....	29
Gambar 3.6 Hasil <i>resizing</i> citra.....	29
Gambar 3.7 Hasil <i>grayscale</i> citra	30
Gambar 3.8 Vertical edge	31
Gambar 3.9 <i>Range</i> besar Piksel	31
Gambar 3.10 Rank filter.....	32
Gambar 3.11 <i>Edge</i> maksimum.....	33
Gambar 3.12 Citra plat nomor	33

Gambar 3.13 Imopen background.....	34
Gambar 3.14 Imsubstract background	34
Gambar 3.15 Grayscaleing citra plat nomor	35
Gambar 3.16 Contrasting	35
Gambar 3.17 Citra <i>binary</i>	36
Gambar 3.18 Hasil filter.....	36
Gambar 3.20 Ekstraksi karakter.....	38
Gambar 3.21 Labelisasi karakter	38
Gambar 3.22 Ekstraksi karakter.....	43
Gambar 4.1 Contoh plat yang tidak diedentifikasi dengan benar	56
Gambar 4.2 Perbandingan huruf K	56
Gambar 4.3 Perbandingan huruf B.....	57

DAFTAR TABEL

	halaman
Tabel IV.1. Training 15 Buah (Jaringan Kompetitif)	47
Tabel IV.2. Training 15 Buah (Jaringan <i>Kohonen</i>).....	48
Tabel IV.3. Training 20 Buah (Jaringan Kompetitif)	49
Tabel IV.4. Training 20 Buah (Jaringan Kohonen)	50
Tabel IV.5. Training 20 Buah (gabungan).....	51
Tabel IV.6. Persentase Akurasi Training 15 Buah.....	52
Tabel IV.7. Persentase Akurasi Training 20 Buah.....	52
Tabel IV.8. Persentase Akurasi Training 20 Buah (jaringan kompetitif dan kohonen).....	53
Tabel V.1. <i>Database</i> Kelas Angka.....	65
Tabel V.2. <i>Database</i> Kelas Huruf.....	68

DAFTAR LAMPIRAN

	halaman
Lampiran 1 <i>Database</i> Angka	65
Lampiran 2 <i>Database</i> Huruf	67

DAFTAR SINGKATAN

SPPNM	Sistem Pengenalan Plat Nomor Mobil
JST	Jaringan Syaraf Tiruan
WIB	Waktu Indonesia Barat
RGB	Red Green Blue

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Berbagai bidang dalam ilmu sains dan teknologi telah banyak dikembangkan oleh para ilmuwan untuk dapat diterapkan bagi kehidupan manusia. Disiplin ilmu yang ada tersebut bila digunakan secara tepat dapat membantu manusia dalam memecahkan masalah manusia di kehidupan sehari-hari. Diantara disiplin ilmu yang ada tersebut adalah pengenalan pola serta JST.

Salah satu metode yang digunakan dalam bidang pengenalan pola adalah metode pendeteksian tepi yang digunakan untuk mencari batas-batas pola tertentu pada suatu citra. JST merupakan salah satu disiplin ilmu yang telah dikembangkan sejak tahun 1943 [1]. JST berusaha memodelkan cara kerja jaringan syaraf manusia sebagai model matematis berdasarkan pemahaman manusia dibidang biologi tentang saraf otak manusia dengan menggunakan mekanisme pemrosesan paralel [1], yaitu neuron sebagai elemen-elemen pemroses yang saling berhubungan. Keunggulan metode ini dibanding metode lain adalah kemampuannya belajar dan memecahkan hubungan yang rumit, yang sulit untuk di deskripsikan antara data masukan dan data keluaran. Suatu sistem JST pada dasarnya akan mempelajari pola perubahan nilai dari suatu matriks masukan, oleh karena itulah system JST juga dikenal sebagai Sistem Pengenal Pola (*Pattern Recognition*).

Semenjak munculnya Sistem Pengenalan Plat Nomor Mobil (SPPNM) dalam era tahun 80an [2] dan berkembang lebih jauh lagi sekitar era 90an menjadi SPPNM yang dapat di komersialisasikan, telah banyak SPPNM yang tersedia dengan metode-metode yang berbeda pula. Sampai saat ini penelitian untuk mengembangkan SPPNM baik dalam lokalisasi plat maupun dalam pengenalan karakter mobil masih

terus berlanjut, hal ini karena SPPNM harus lah bersifat adaptif untuk segala kondisi *image* yang diambil [2]. Misalnya ketika *image* memiliki noise karena adanya cahaya yang berlebihan dalam pengambilan *image* plat nomor. Selain itu SPPNM terus dikembangkan karena aplikasi dari SPPNM terus meningkat contohnya dalam mengatasi kemacetan ditempat-tempat penting seperti parkir suatu tempat perbelanjaan, jalan tol, atau tempat-tempat lain yang sejenisnya dibutuhkan SPPNM yang dapat mengenali plat nomor dengan cepat. SPPNM juga dibutuhkan untuk mengenali daerah (kota) asal mobil itu berada contohnya B untuk Jakarta dan BD untuk Bengkulu. SPPNM juga dibutuhkan untuk penghitungan lamanya waktu parkir suatu mobil ditempat parkir dan untuk menghitung berapa nominal uang yang harus dibayar oleh si pemilik mobil ketika akan keluar dari tempat parkir. SPPNM juga sudah pernah diteliti oleh Darundriyo Djoko [3] menggunakan metode pencocokan pola dan JST

Berdasarkan penelitian mengenai metode pengenalan pola menggunakan pendeteksian tepi dan JST sebagai *pattern recognition* serta aplikasi dari SPPNM yang terus meningkat, penulis merasa bahwa metode pendeteksian tepi dan JST dapat digunakan untuk membantu mengembangkan sebuah perangkat lunak untuk pengenalan plat nomor mobil. Kedua metode tadi diharapkan dapat menyelesaikan permasalahan dasar dalam pengenalan sebuah plat nomor yaitu pencarian lokasi suatu plat nomor serta pengenalan karakter pada plat nomor

1.2 PERUMUSAN MASALAH

Terdapat beberapa masalah yang dihadapi dan akan dipecahkan dalam penelitian, antara lain :

1. penentuan lokasi dan ekstraksi plat nomor dari citra sebuah mobil
2. penyamaan *background* dari citra plat nomor. Yang dimaksudkan dengan *background* disini adalah bentuk dan warna yang ada pada plat nomor selain karakter plat nomor.
3. ekstraksi masing-masing karakter huruf dan angka yang terdapat pada citra plat nomor

4. pengenalan angka dan huruf yang telah diekstraksi

1.3 TUJUAN

Tujuan penelitian pada skripsi ini yaitu merancang dan mengimplementasikan sistem pengenalan plat nomor mobil (SPPNM) dengan memanfaatkan metode pendeteksian tepi tipe verical dan rank filter untuk menentukan batas-batas lokasi plat nomor serta jaringan kompetitif dan kohonen untuk mengenali karakter yang ada di dalam plat nomor berupa huruf dan angka

1.4 METODOLOGI PENELITIAN

Pada metodologi penelitian akan dilakukan beberapa tahap untuk memecahkan masalah-masalah yang dihadapi seperti yang telah disebutkan pada perumusan masalah. Untuk memecahkan masalah 1-3 digunakan tahap *pre processing* citra dan untuk masalah 4 digunakan tahap identifikasi menggunakan jaringan kohonen dan jaringan kompetitif

1.5 BATASAN MASALAH

Dalam skripsi ini, ruang lingkup masalah dibatasi pada hal-hal berikut:

1. Plat nomor yang ingin dikenali adalah plat nomor kendaraan pribadi yang dikeluarkan oleh instansi yang berwenang di Indonesia
2. Pemrosesan citra dilakukan secara offline dimana citra yang akan diproses telah di akuisisi terlebih dahulu menggunakan kamera digital
3. Posisi plat nomor yang ingin dikenali sejajar dengan garis horizontal
4. Masukan sistem berupa citra tampak depan atau belakang mobil
5. Sampel data plat yang diambil memiliki 7 karakter
6. Plat nomor tidak mengalami modifikasi (plat nomor standar).

1.6 SISTEMATIKA PENELITIAN

Untuk memberikan gambaran mengenai apa saja yang ditulis dalam skripsi ini maka penulis pada bagian ini akan menguraikan secara garis besar sistematika/cara penulisan yang terdiri dari:

BAB I PENDAHULUAN

Pada bab ini penulis menjelaskan secara singkat mengenai latar belakang, perumusan masalah, tujuan, metodologi penelitian, batasan masalah dan sistematika penelitian, dengan maksud memberikan gambaran kepada pembaca tentang isi skripsi yang penulis buat.

BAB II LANDASAN TEORI

Pada bab ini penulis menjelaskan secara singkat tentang teori-teori yang akan berhubungan dengan penelitian yang akan dilakukan. Bab ini akan menjelaskan tentang sistem pengenalan plat nomor mobil, *image recognition*, sistem pengenalan pola (pattern recognition system), jaringan saraf tiruan *kohonen* dan jaringan kompetitif

BAB III METODOLOGI PENELITIAN

Pada bab ini penulis menjelaskan perancangan SPPNM dengan jaringan kohonen dan kompetitif

BAB IV HASIL UJI COBA DAN ANALISIS

Menjelaskan hasil dari simulasi serta analisis dari hasil simulasi

BAB V KESIMPULAN

Memberikan kesimpulan dari keseluruhan skripsi ini

DAFTAR ACUAN

DAFTAR PUSTAKA

LAMPIRAN



BAB II

LANDASAN TEORI

2.1 PERATURAN PENOMORAN PLAT NOMOR MOBIL [3]

Setiap mobil di belahan dunia manapun yang memiliki izin untuk berjalan di jalan raya pasti memiliki sebuah plat nomor unik yang membedakannya dengan mobil lainnya di negara tempat mobil tersebut berada. Aturan atau regulasi mengenai plat nomor mobil tersebut dapat berbeda-beda dari satu negara ke negara lainnya. Di Amerika misalnya, untuk membedakan mobil di suatu negara bagian dengan negara bagian lainnya digunakan warna dasar plat nomor yang berbeda. Begitu juga dengan karakter yang ada di dalam plat nomor tersebut, dapat dipisah sesuai keinginan pemilik mobil yang bersangkutan atau mendapatkannya dari instansi yang berwenang.

Berbeda dengan peraturan plat nomor di negara Amerika, Indonesia memiliki peraturan yang lebih sederhana. Warna dasar plat nomor di Indonesia digunakan untuk membedakan jenis kendaraan, seperti warna hitam untuk kendaraan pribadi, warna kuning untuk kendaraan umum dan seterusnya. Dan untuk membedakan area asal kendaraan tersebut digunakan huruf yang diletakkan pada bagian depan plat nomor, seperti B untuk Jakarta dan seterusnya. Secara hukum pemilik kendaraan di Indonesia tidak dapat memesan plat nomor sesuai dengan keinginannya tanpa mengikuti peraturan penomoran plat yang berlaku.

2.1.1 Plat Nomor

Seperti telah dikemukakan pada batasan masalah, sistem yang akan dibuat pada skripsi ini hanya terbatas untuk mengenali plat nomor pribadi di wilayah Indonesia, oleh karena itu sebelum melangkah lebih jauh, kita perlu mengetahui

peraturan plat nomor mobil pribadi yang berlaku di Indonesia, bentuk umum dari plat nomor serta lokasi tempat plat nomor tersebut di mobil.

2.1.1.1 Peraturan Penomoran Plat Nomor Mobil Pribadi

Di Indonesia, sebuah plat nomor pribadi memiliki warna dasar hitam dengan karakter berwarna putih, sedangkan pola karakter di dalamnya memiliki kesamaan dengan pola kendaraan umum, dengan pola sebagai berikut di mana :

KA: Kode area mobil tersebut berupa huruf dengan maksimum jumlah digit 2 dan minimum 1, seperti B untuk Jakarta dan sekitarnya, F untuk Bogor, AB untuk Yogyakarta, dan lain-lain.

NP: Nomor plat mobil berupa angka dengan batas maksimum jumlah digit 4 dan minimum 1, seperti 1, 1092, 9211, dan lain-lain.

KT: Karakter tambahan yang bisa ada atau tidak (optional), berupa huruf dengan batas maksimum jumlah digit 2 dan minimum 0, seperti A, BS, XY , dan lain-lain.

Pada bagian ini terdapat suatu pengecualian, yaitu huruf pertama di bagian ini tidak boleh huruf "O".

Contoh plat nomor pribadi dengan mengikuti peraturan di atas adalah B1, B 10 LA , AD 9922 CY , dan seterusnya. Sedangkan contoh plat nomor mobil pribadi yang tidak mengikuti peraturan di atas adalah B 10288 AC , BBB 2G, dan seterusnya.

2.1.1.2 Desain Plat Nomor Pribadi

Berbeda dengan plat nomor kendaraan militer atau kendaraan dinas kedutaan besar, plat nomor pribadi memiliki desain tersendiri yang dikhususkan untuk kendaraan pribadi dan kendaraan umum baik kendaraan roda 2,3,4. Desain plat nomor tersebut dapat digambar sebagai berikut:



Gambar 2.1 Citra plat nomor

Dari gambar di atas, plat nomor mobil pribadi dapat dipisah menjadi 2 bagian utama, yang dipisahkan oleh suatu garis lurus, yaitu:

1. B 2629 VM merupakan informasi plat nomor mobil yang mengikuti peraturan penomoran seperti yang telah dijelaskan sebelumnya
2. 05.10 merupakan informasi masa berlaku plat nomor yang berisi informasi bulan dan tahun terakhir STNK (Surat Tanda Nomor Kendaraan) tersebut berakhir.

2.1.1.3 Lokasi Plat Nomor Pribadi

Peraturan mengenai di mana plat nomor suatu kendaraan harus ditempatkan dapat berbeda-beda dari satu negara dengan negara lain bahkan untuk setiap negara bagian pun dapat berbeda. Di Amerika, setiap kendaraan diwajibkan untuk meletakkan plat nomor kendaraannya di bagian belakang dan untuk beberapa negara bagian diharuskan pula untuk meletakkan satu plat nomor lagi di bagian depan mobil. Dengan kata lain, plat nomor untuk 1 kendaraan di suatu negara bagian dapat berjumlah 1 buah di bagian belakang dan di beberapa negara bagian lain berjumlah 2 buah untuk bagian depan dan belakang kendaraan.

Peraturan lokasi plat nomor yang berlaku di Indonesia mewajibkan setiap kendaraan meletakkan plat nomor mobilnya di bagian depan dan belakang kendaraan tersebut. Peraturan ini berlaku untuk setiap propinsi di Indonesia tanpa kecuali. Pada bagian depan, plat nomor biasanya diletakkan pada bumper depan mobil dan untuk bagian belakang, plat nomor biasanya diletakkan pada bumper belakang atau badan belakang mobil (seperti pintu belakang).

Dari penjelasan di atas, kini kita telah memiliki informasi mengenai plat nomor beserta informasi untuk mendapatkan plat nomor tersebut sehingga kita dapat

menyusun sistem yang dapat dipergunakan untuk mengenali plat nomor kendaraan pribadi. Untuk mempersingkat penulisan, mulai saat ini plat nomor pribadi akan ditulis sebagai plat nomor saja.

2.2 PEMROSESAN CITRA

Pemrosesan citra yang dilakukan terdiri dari 4 tahap, yaitu:

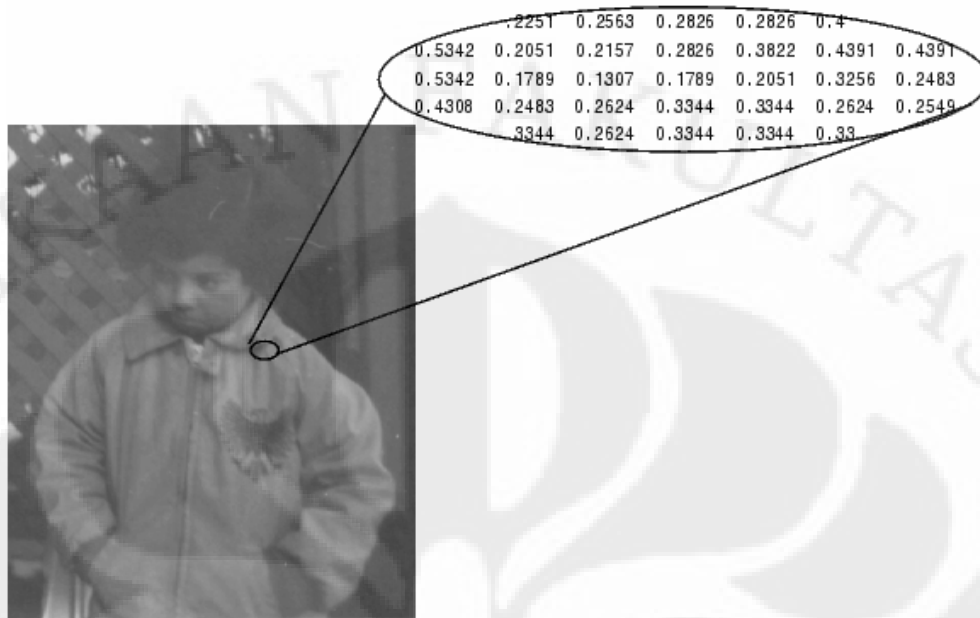
1. *sampling* dan *grayleveling*
2. *leveling* dan *reshaping*
3. *filtering*
4. *edge detection*

2.2.1 Sampling dan Graylevel Quantization

Pada proses MATLAB, fungsi citra $f(x,y)$ harus didijitaskan secara amplitudo (intensitas) maupun spasial. Proses dijitasi dari kordinat spasial (x,y) disebut *Image sampling* [4], sedangkan dijitasi amplitudo disebut *gray level quantization* [4]

Jika suatu citra $f(x,y)$ disampling dan menjadi $N \times M$ array maka setiap elemen dari *array* merupakan kuantitas diskrit seperti dapat dilihat pada Gambar 2.2. Setiap elemen dari array dapat disebut *image element* atau *picture element* atau *pixel*. Untuk sebuah *image digital*, setiap *pixel* memiliki nilai integer yakni *gray level* yang menunjukkan amplitudo atau intensitas dari *pixel* tersebut. Citra merupakan fungsi dua dimensi yang kedua variabelnya yaitu nilai amplitudo dan koordinatnya merupakan nilai integer.

Gambar 2.2 memperlihatkan sebuah citra dengan *grayscale level 2* dimana *pixel* yang bernilai 0 merepresentasikan warna hitam sempurna dan 1 merepresentasikan warna putih sempurna.



Gambar 2.2 Citra dengan *grayscale level 2* (nilai 0 sampai 1) [5]

2.2.2 Leveling dan Reshaping [5]

Leveling merupakan pelevelan nilai *graylevel* citra dari suatu *array* atau matriks ke suatu *range* nilai diskrit tertentu. Pada gambar 2.3 dapat dilihat pelevelan citra menjadi hanya 2 level yaitu 0 yang merepresentasikan warna hitam dan 1 yang merepresentasikan warna putih. Sedangkan *reshaping* adalah proses untuk mengubah ukuran matriks namun dengan jumlah *array* yang tetap sama, misalkan matriks $N \times M$ menjadi matriks $M \times N$.



Gambar 2.3 *Leveling* citra menjadi hanya 2 tingkat (nilai 0 dan 1) [5]

2.2.3 Nilai Rata-Rata *Graylevel* [5]

Untuk memproses sebuah citra, dibutuhkan sebuah parameter yang dapat dijadikan representasi karakteristik dari citra tersebut. Salah satu pendekatan yang dapat dijadikan parameter karakteristik dari sebuah citra adalah nilai rata-rata matriks citra. Nilai rata-rata *graylevel* didapatkan dari penjumlahan nilai *graylevel* dari suatu citra dibagi jumlah *pixel*.

2.2.4 *Filtering* [4]

Pada pemrosesan citra, proses *filtering* suatu citra adalah proses yang sangat penting karena berbagai kegunaannya dalam memanipulasi citra. Proses *filtering* suatu citra adalah proses konvolusi antara matriks citra dengan matriks filter untuk mendapatkan suatu efek khusus dari citra, seperti mendeteksi tepi, mengurangi *noise*, meningkatkan ketajaman gambar, dsb. *Filtering* pada pemrosesan citra terbagi 2, yaitu:

- filter pada domain spasial

 - Terbagi 2 yaitu : - filter linear ($m \times m$)

 - filter non-linear (*rank filter*) ($m \times n$)

- filter pada domain frekuensi

 - Terbagi 2 yaitu: - *Low Pass Filter*

 - *High Pass Filter*

2.2.5 *Edge Detection* [4]

Pendeteksian tepi adalah pendekatan yang paling umum digunakan untuk mendeteksi diskontinuitas pada nilai intensitas citra. Diskontinuitas itu biasanya dideteksi dengan mengaplikasikan suatu filter spasial khusus yang merupakan turunan pertama atau gradien dari citra $f(x,y)$, yaitu:

$$\nabla f = G_x + G_y$$

Di mana ∇f bernilai nol pada area dengan intensitas konstan dan berubah proporsional terhadap perubahan intensitas pada area dengan nilai piksel variabel.

Faktor penting dari pendeteksian tepi ini adalah memperkirakan turunan G_x dan G_y .

Berbagai metode deteksi tepi dengan menggunakan turunan pertama telah ditemukan, seperti:

1. Sobel
2. Prewitt
3. Roberts

Sobel :

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Prewitt :

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Roberts :

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

2.3 SISTEM PENGENALAN POLA

Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain. Terdapat dua pendekatan yang

dilakukan dalam pengenalan pola: pendekatan pola secara statistik dan pendekatan pola secara sintatik atau struktural.

(a) Pengenalan Pola Secara Statisik [6]

Pendekatan ini menggunakan teori-teori ilmu peluang dan statistik. Ciri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya. Pola yang berbeda memiliki distribusi yang berbeda pula. Dengan menggunakan teori keputusan di dalam statistik, kita menggunakan distribusi ciri untuk mengkasifikasikan pola.

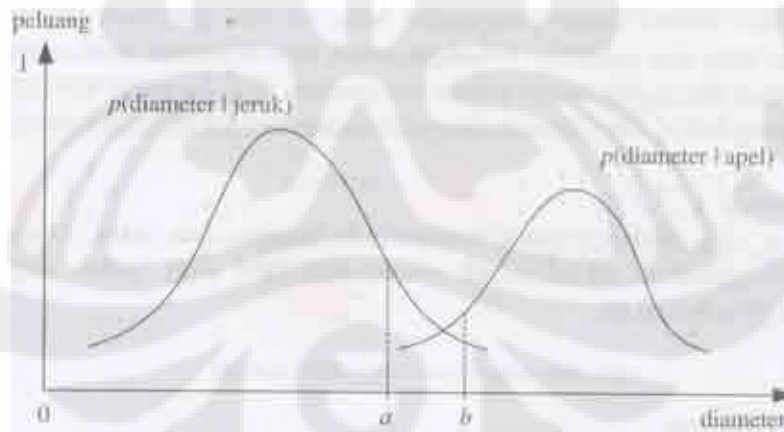
Contoh teori keputusan:

Misalkan ada N pola yang dikenali, yaitu w_1, w_2, \dots, w_n dan fungsi peluang atau kerapatan dari ciri-ciri pada pola diketahui. Jika \bar{x} merupakan hasil pengukuran ciri-ciri, maka

$$p(\bar{x} | w_i), \quad i = 1, 2, \dots, n \dots \dots \dots (2.1)$$

dapat dihitung.

Sebagai contoh, misalkan diketahui fungsi kerapatan dari diameter buah jeruk dan apel yang diperlihatkan pada Gambar 2.4.



Gambar 2.4 Grafik fungsi kerapatan dari ciri diameter jeruk dan apel. [6]

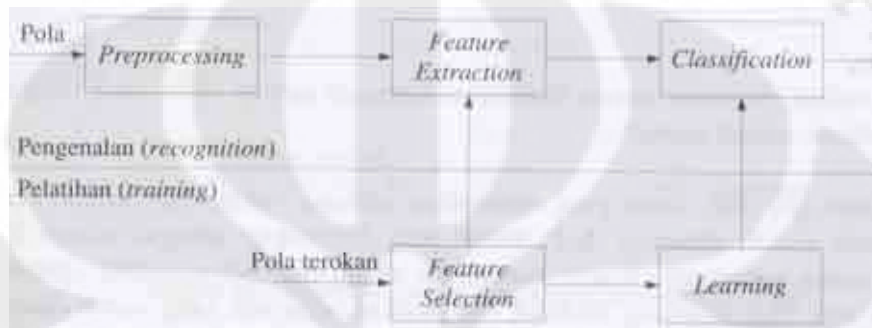
Jika sebuah objek diukur dan diperoleh diameternya adalah a cm, maka kita mengklasifikasikan objek tersebut sebagai “jeruk”, karena

$$p(a | jeruk) > p(a | apel)$$

dan jika hasil pengukuran diameter adalah b cm, kita mengklasifikasikan objek tersebut sebagai “apel”, karena

$$p(a | jeruk) < p(a | apel)$$

Sistem pengenalan pola dengan pendekatan statistik ditunjukkan oleh diagram pada Gambar 2.5.



Gambar 2.5 Sistem pengenalan pola dengan pendekatan statistik. [6]

Pada Gambar 2.5 terdapat istilah pola terakan, yang dimaksud dengan pola terakan yaitu suatu pola citra hasil dari proses *sampling*. Ada dua fase dalam sistem pengenalan pola: (i) fase pelatihan dan (ii) fase pengenalan. Pada fase pelatihan, beberapa contoh citra dipelajari untuk menentukan ciri yang akan digunakan dalam proses pengenalan serta prosedur klasifikasinya. Pada fase pengenalan, citra diambil cirinya kemudian ditentukan kelas kelompoknya.

a. Pre-processing

Proses awal yang digunakan memperbaiki kualitas citra (*edge enhancement*) dengan menggunakan teknik-teknik pengolahan citra.

b. Feature extraction

Proses pengambilan ciri-ciri yang terdapat pada objek di dalam citra. Pada proses ini objek di dalam citra mungkin perlu dideteksi seluruh tepinya, lalu menghitung properti-properti objek yang berkaitan dengan ciri. Beberapa proses ekstraksi ciri mungkin perlu mengubah citra masukan sebagai masukan citra biner, melakukan penipisan pola, dan sebagainya.

c. Classification

Proses ini mengelompokkan objek ke dalam kelas yang sesuai.

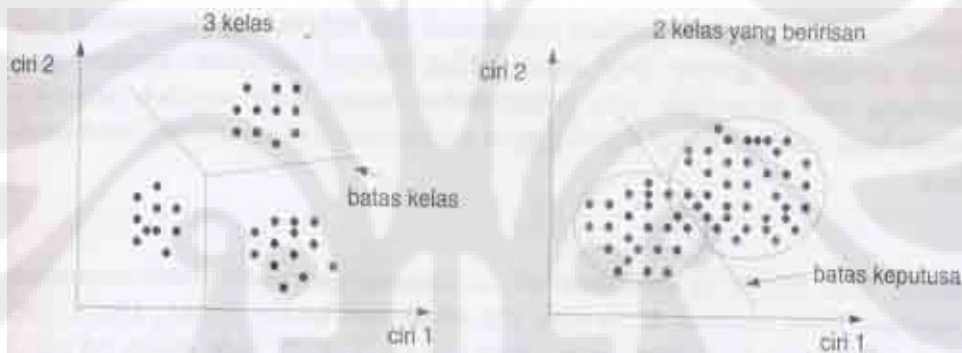
d. *Feature selection*

Proses memilih ciri pada suatu objek agar diperoleh ciri yang optimum, yaitu ciri yang dapat digunakan untuk membedakan suatu objek dengan objek lainnya.

e. *Learning*

Proses belajar membuat aturan klasifikasi sehingga jumlah kelas yang tumpang tindih dibuat sekecil mungkin

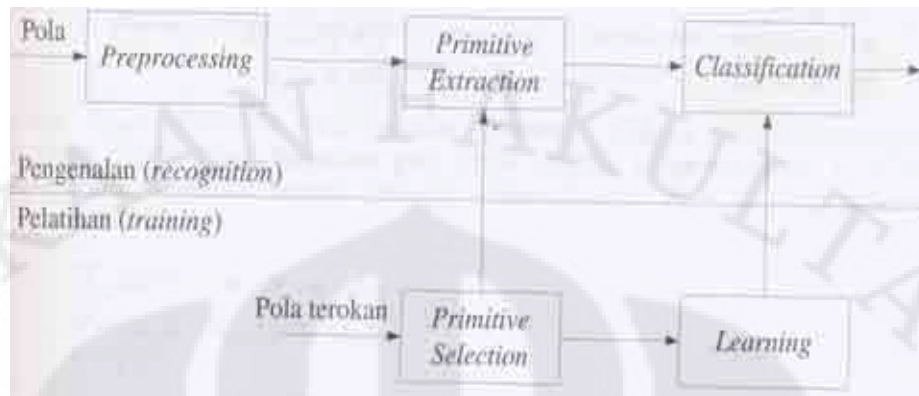
Kumpulan ciri dari suatu pola dinyatakan sebagai vektor ciri dalam ruang bahumatra (multi dimensi). Jadi, setiap pola dinyatakan sebagai sebuah titik dalam ruang bahumatra. Ruang bahumatra dibagi menjadi sejumlah uparuang (sub-ruang). Tiap uparuang dibentuk berdasarkan pola-pola yang sudah dikenali kategori dan ciri-cirinya (melalui fase pelatihan). Lihat Gambar 2.6.



Gambar 2.6 Contoh pembagian kelas pola [6]

(b) Pengenalan Pola secara Sintatik [6]

Pendekatan ini menggunakan teori bahasa formal. Ciri-ciri yang terdapat pada suatu pola ditentukan primitif dan hubungan struktural antara primitif kemudian menyusun tata bahasanya. Dari aturan produksi pada tata bahasa tersebut kita dapat menentukan kelompok pola. Gambar 2.7 memperlihatkan sistem pengenalan pola dengan pendekatan sintaktik. Pengenalan pola secara sintatik lebih dekat ke strategi pengenalan pola yang dilakukan manusia, namun secara praktek penerapannya relatif sulit dibandingkan pengenalan pola secara statistik.



Gambar 2.7 Sistem pengenalan pola dengan pendekatan sintatik [6]

Pada Gambar 2.7 terdapat istilah pola terokan, yang dimaksud dengan pola terokan yaitu suatu pola citra hasil dari proses *sampling*. Pendekatan yang digunakan dalam membuat tata bahasa untuk mengenali pola adalah mengikuti kontur (tepi batas) objek dengan sejumlah segmen garis terhubung satu sama lain, lalu mengkodekan setiap garis tersebut (misalnya dengan kode rantai). Setiap segmen garis merepresentasikan primitif pembentuk objek.

Contoh aplikasi sistem pengenalan pola:

1. Sistem pengenalan plat nomor mobil (seperti yang penulis buat).
2. Sistem pengenalan wajah (*face recognition system*).
3. Sistem pengenalan tanda tangan (*signature recognition system*).
4. Sistem pengenalan angka dan huruf yang digunakan sebagai prinsip kerja dari mesin scanner.

2.4 JARINGAN SARAF TIRUAN

Jaringan saraf tiruan didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan saraf manusia [1]. Jaringan syaraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia yang didasarkan asumsi sebagai berikut: [7]

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron.
2. Isyarat mengalir di antara sel saraf/neuron melalui suatu sambungan penghubung.

3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan/mengalikan isyarat yang dikirim melaluinya.
4. Setiap sel syaraf akan menerapkan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.

Model saraf dalam jaringan saraf tiruan ditunjukkan dengan kemampuan dalam emulasi, analisis, prediksi, dan asosiasi. Kemampuan yang dimiliki oleh jaringan saraf tiruan dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau *input* yang dimasukkan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan karakteristik dari *input* yang disimpan kepadanya.

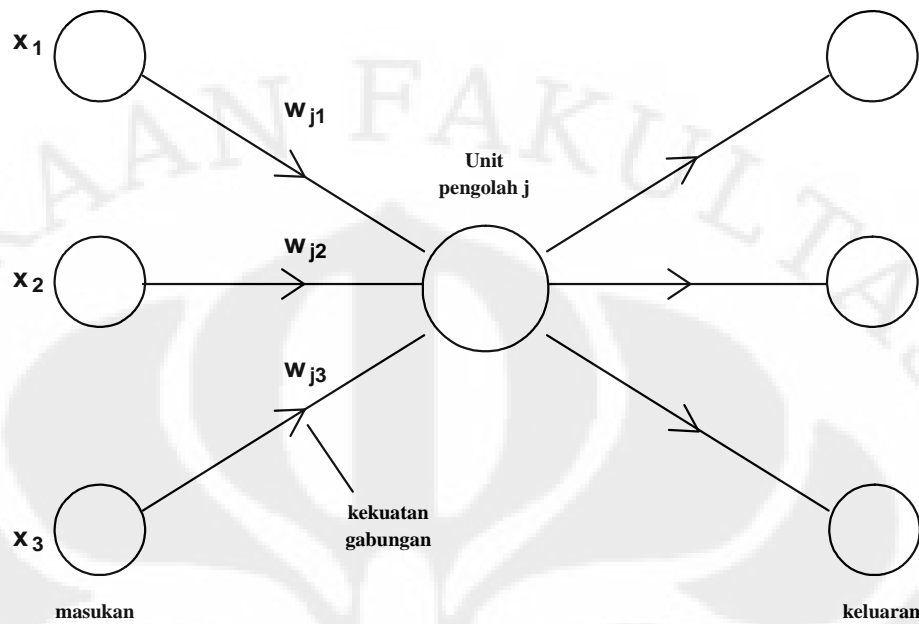
Jaringan syaraf tiruan merupakan suatu bentuk arsitektur terdistribusi paralel dengan sejumlah besar *node* dan hubungan antar *node* tersebut. Tiap titik hubungan dari suatu *node* ke *node* yang lain mempunyai harga yang diasosiasikan dengan bobot. Setiap *node* memiliki suatu nilai yang diasosiasikan sebagai nilai aktivasi *node*.

Karakteristik jaringan saraf tiruan ditentukan oleh: [7]

1. Pola hubungan antar neuron (disebut dengan arsitektur jaringan)
2. Metode penentuan bobot-bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan),
3. Fungsi aktivasi.

Neuron dimodelkan dari penyederhanaan sel syaraf manusia yang sebenarnya.

Gambar 2.8 merupakan struktur unit jaringan syaraf tiruan.



Gambar 2.8 Struktur unit jaringan syaraf tiruan [7]

Pada sisi sebelah kiri Gambar 2.8 terlihat beberapa masukan yang menuju ke unit pengolah yang masing-masing datang dari unit-unit yang berbeda $X(n)$. Setiap sambungan mempunyai kekuatan hubungan terkait (bobot) yang disimbolkan dengan $w(n)$. Unit pengolah akan membentuk penjumlahan berbobot dari tiap masukannya dan menggunakan fungsi ambang non-linier (fungsi aktivasi) untuk menghitung keluarannya. Hasil perhitungan akan dikirimkan melalui hubungan keluaran seperti tampak pada gambar sisi sebelah kanan.

Masing-masing sambungan antar unit pengolah berperan sebagai sebuah penghubung. Nilai-nilai numerik dilewatkan sepanjang sambungan dari satu unit pengolah ke unit pengolah lainnya. Ketika unit pengolah melakukan perhitungan, nilai-nilai ini diberi bobot berdasarkan kekuatan hubungan. Kekuatan hubungan pada tiap sambungan akan disesuaikan selama pelatihan sehingga pada akhir pelatihan dihasilkan jaringan saraf dengan bobot yang mantap.

Jaringan saraf tiruan dirancang dengan menggunakan suatu aturan yang bersifat menyeluruh di mana seluruh model jaringan memiliki konsep dasar yang sama. Arsitektur sebuah jaringan akan menentukan keberhasilan target yang akan

dicapai karena tidak semua permasalahan dapat diselesaikan dengan arsitektur yang sama. Ada banyak model jaringan yang dapat digunakan sebagai jaringan saraf tiruan.

Sebagian besar jaringan saraf melakukan penyesuaian bobot-bobotnya selama menjalani prosedur latihan. Pelatihan dapat berupa pelatihan terbimbing dan pelatihan tak terbimbing. Pada pelatihan terbimbing diperlukan pasangan masukan sasaran untuk tiap pola yang dilatihkan. Sedangkan pelatihan tak terbimbing, penyesuaian bobot (sebagai tanggapan terhadap masukan), tak perlu disertai sasaran. Dalam pelatihan tak terbimbing, jaringan mengklasifikasikan pola-pola yang ada berdasarkan kategori kesamaan.

Fungsi aktivasi merupakan bagian penting dalam tahapan perhitungan keluaran dari suatu algoritma. Beberapa fungsi aktivasi yang digunakan dalam jaringan saraf tiruan adalah:

1. Fungsi identitas

$$f(x)=x, \text{ untuk semua } x \dots\dots\dots (2.2)$$

2. Fungsi undak biner(dengan batas ambang)

$$f(x) = \begin{cases} 1 & \text{untuk } x \geq \theta \\ 0 & \text{untuk } x < \theta \end{cases} \dots\dots\dots (2.3)$$

3. Fungsi Sigmoid

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \dots\dots\dots (2.4)$$

$$f'(x) = \sigma f(x)[1 - f(x)] \dots\dots\dots (2.5)$$

Dengan : σ = konstanta

4. Fungsi sigmoid bipolar

$$g(x) = 2f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)} \dots\dots\dots (2.6)$$

$$= \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)}$$

$$g'(x) = \frac{\sigma}{2} [1 + g(x)][1 - g(x)] \dots\dots\dots (2.7)$$

Dengan : σ = konstanta

Jaringan syaraf tiruan *Kohonen* merupakan salah satu jenis algoritma jaringan syaraf tiruan yang banyak digunakan, penjelasan mengenai jaringan syaraf tiruan *Kohonen* lebih lengkapnya akan dijelaskan pada sub bab 2.5.

2.5 JARINGAN SARAF TIRUAN KOHONEN

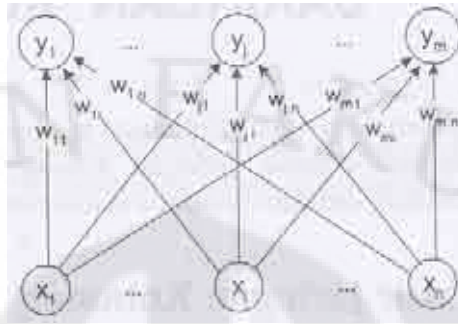
Jaringan yang ditemukan oleh Kohonen merupakan salah satu jaringan yang banyak dipakai. Pada jaringan ini, suatu lapisan yang berisi neuron-neuron akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu kelompok yang dikenal dengan istilah *cluster*. Selama proses penyusunan diri, *cluster* yang memiliki vektor bobot yang paling cocok dengan pola *input* (memiliki jarak yang paling dekat) akan terpilih sebagai pemenang [8].

2.5.1 Arsitektur jaringan Kohonen

Misalkan masukan berupa vektor yang terdiri dari n komponen (tuple) yang akan dikelompokkan dalam maksimum m buah kelompok (disebut vektor contoh). Keluaran jaringan adalah kelompok yang paling dekat dengan masukan yang diberikan. Ada beberapa ukuran kedekatan yang dapat dipakai. Ukuran yang sering dipakai adalah jarak euclidean yang paling minimum.

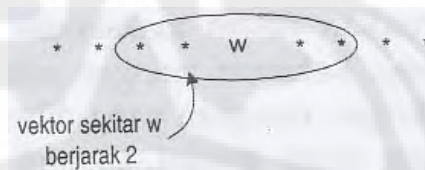
Bobot vektor-vektor contoh berfungsi sebagai penentu kedekatan vektor contoh tersebut dengan masukan yang diberikan. Selama proses pengaturan, vektor contoh yang pada saat itu paling dekat dengan masukan akan muncul sebagai pemenang. Vektor pemenang (dan vektor-vektor sekitarnya) akan dimodifikasi bobotnya.

Arsitektur jaringan kohonen [9] tampak dalam gambar 2.9.

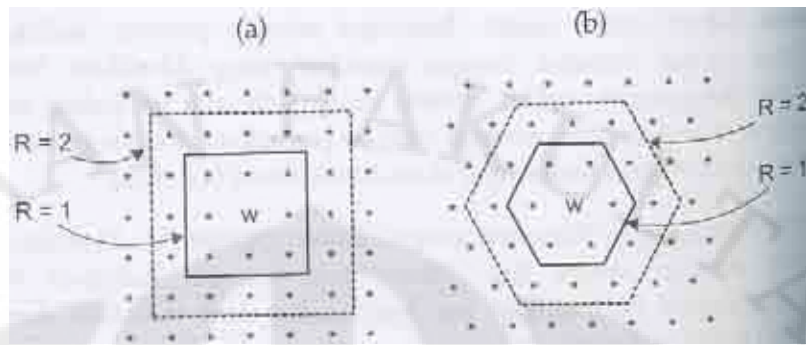


Gambar 2.9 Arsitektur jaringan kohonen [9]

Arsitektur ini mirip dengan model lain. Hanya saja jaringan Kohonen tidak menggunakan perhitungan net (hasil kali masukan dengan bobot) maupun fungsi aktivasi. Misalkan pada suatu iterasi tertentu, vektor contoh w menjadi pemenang. Pada iterasi berikutnya, vektor w dan vektor-vektor sekitarnya akan dimodifikasi bobotnya. Gambar 2.10 menunjukkan kasus untuk vektor w berupa vektor 1 dimensi (dengan jarak $R=2$), sedangkan gambar 2.11 a dan b menunjukan vektor sekitar w jika w direpresentasikan dalam 2 dimensi dengan $R=1$ dan $R=2$. Jika menggunakan bentuk bujur sangkar dengan jarak $R=1$, ada 8 vektor disekitar vektor w (gambar 2.11(a)). Tetapi jika menggunakan bentuk heksagonal ada 6 vektor disekitar vektor w (gambar 2.11(b)).



Gambar 2.10 w berupa vektor 1 dimensi [9]



Gambar 2.11 w berupa vektor 2 dimensi [9]

Algoritma Jaringan Kohonen [9]

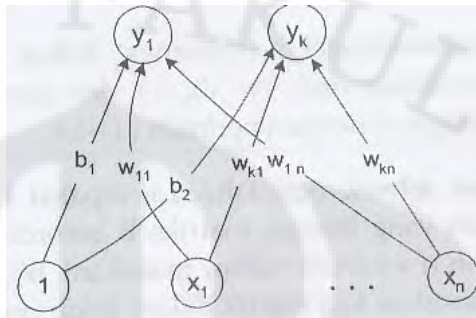
Algoritma pengelompokan pola jaringan Kohonen adalah sebagai berikut :

1. Inisialisasi
 - a. Bobot w_{ji} (acak).
 - b. Laju pemahaman awal dan faktor penurunannya
 - c. Bentuk dan jari-jari (=R) topologi sekitarnya.
2. Selama kondisi penghentian bernilai salah, lakukan langkah 2 sampai langkah 7
3. Untuk setiap vektor masukan x , lakukan langkah 3 sampai langkah 5
4. Hitung $D(j) = \sum_i (w_{ji} - x_i)^2$ untuk semua j (2.8)
5. Tentukan indeks J sedemikian sehingga $D(J)$ minimum
6. Untuk setiap unit j disekitar J modifikasi bobot :
7. $W_{ji}^{baru} = W_{ji}^{lama} + \alpha(x_i - W_{ji}^{lama})$ (2.9)
8. Modifikasi laju pemahaman
9. Uji kondisi penghentian

Kondisi penghentian iterasi adalah selisih antara w_{ji} saat itu dengan w_{ji} pada iterasi sebelumnya. Apabila semua w_{ji} hanya berubah sedikit saja, berarti iterasi sudah mencapai konvergensi sehingga dapat dihentikan.

2.6 JARINGAN KOMPETITIF

Arsitektur jaringan kompetitif tampak pada Gambar 2.8.



Gambar 2.12 Arsitektur jaringan kompetitif [9]

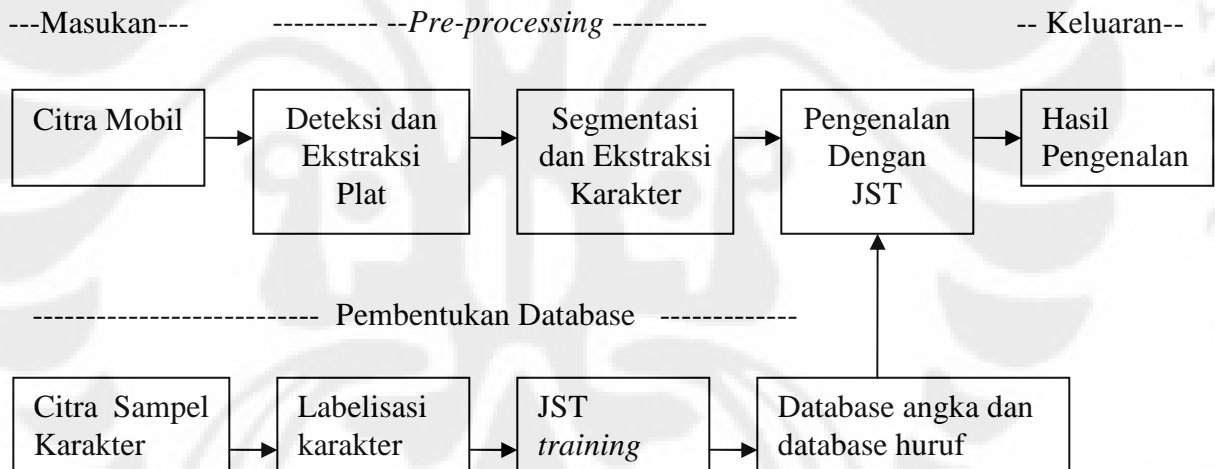
Arsitektur tersebut mirip dengan jaringan ADALINE [9]. Terdapat bias yang terhubung ke setiap neuron keluaran. Dalam iterasinya, Matlab menggunakan aturan Kohonen dalam mengubah bobot-bobotnya. Neuron yang bobotnya paling mendekati vektor masukan akan diperbaiki dan dibuat lebih dekat lagi. Dengan bertambahnya vektor masukan yang diberikan, maka setiap neuron yang terdekat ke suatu kelompok neuron akan mengubah bobotnya ke arah vektor masukan tersebut. Ini berarti bahwa lama kelamaan vektor masukan akan terbagi menjadi beberapa kelompok. Vektor-vektor masukan yang saling berdekatan akan membentuk sebuah kelompok.

BAB III

METODOLOGI PENELITIAN

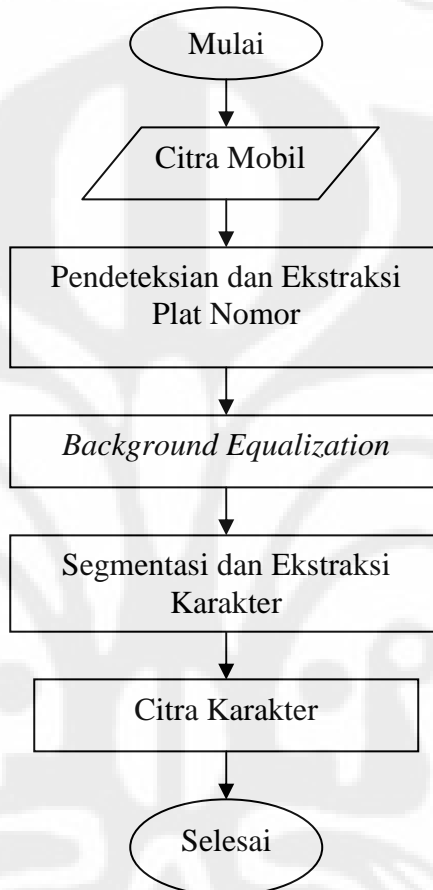
3.1 LANGKAH PEMECAHAN MASALAH

Perangkat yang dibuat terdiri dari 3 tahap yaitu *pre-processing*, proses *training* dan proses *Identifikasi*. Secara garis besar, proses sistem yang dirancang adalah sebagai berikut:

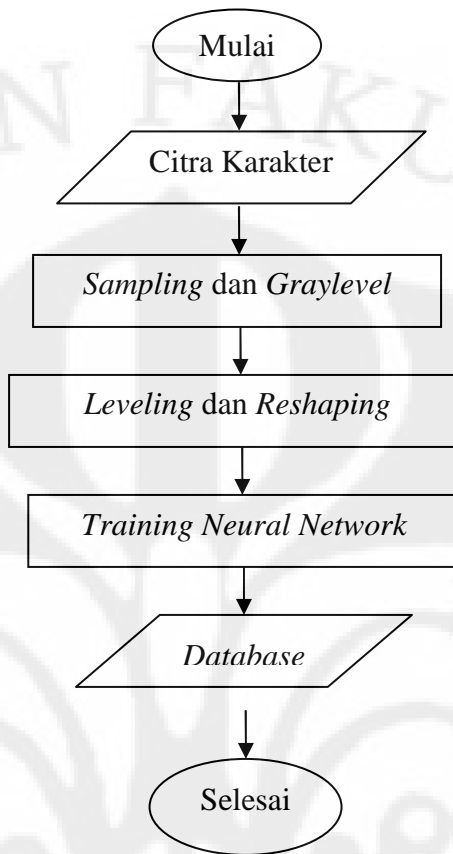


Gambar 3.1 Blok Diagram Proses Sistem

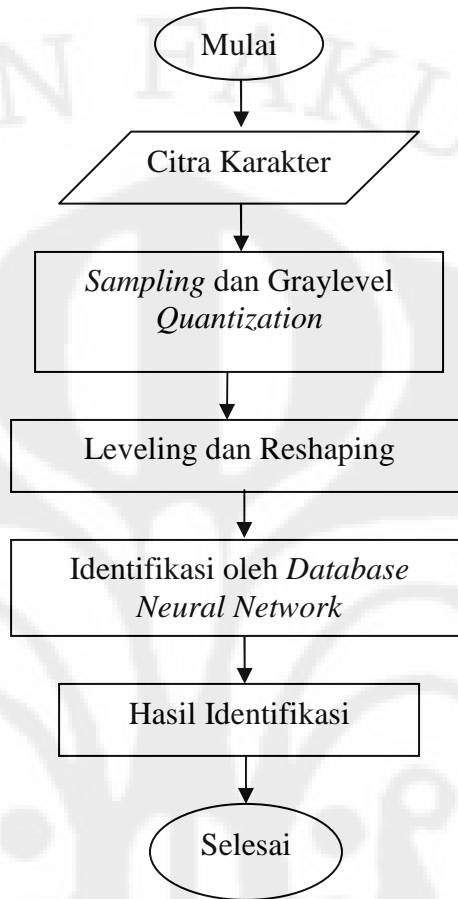
Berikut ini merupakan diagram alir untuk *pre-processing*, proses *training* dan proses identifikasi



Gambar 3.2 Diagram alir *pre-processing*



Gambar 3.3 Diagram alir proses *training*



Gambar 3.4 Diagram alir proses identifikasi

Perangkat lunak Pengenalan Plat Nomor dirancang dengan menggunakan menggunakan komputer dengan spesifikasi sebagai berikut :

<i>Sistem Operasi</i>	: <i>Windows XP Professional</i>
<i>Processor</i>	: <i>Pentium 3.0 GHz</i>
<i>VGA Card</i>	: <i>NVIDIA GeForce4 MX 440</i>
<i>Software</i>	: <i>Program Simulasi</i>
<i>Memory</i>	: <i>512 MB</i>

3.2 PRE PROCESSING CITRA

Terdiri dari 3 proses antara lain:

1. Pendeteksian dan Ekstraksi Plat Nomor
2. *Background Equalization*
3. Ekstraksi Karakter

3.2.1 Pendeteksian dan Ekstraksi Plat Nomor

Pada proses ini terdiri dari 8 tahap, antara lain:

1. *Sampling*
2. *Resizing*
3. *Grayscale* citra mobil
4. *Vertical Edge Detection*
5. *Cleaning phase*
6. *Horizontal Rank Filtering*
7. Area Terbesar (*High Density Of Bright Edges*)
8. Ekstraksi Plat Nomor

3.2.1.1 *Sampling* [4]

Citra mobil berupa fungsi $f(x,y)$, pada tahap ini dilakukan proses *sampling* pada citra mobil. Seperti yang telah dijelaskan pada dasar teori bahwa pada proses *sampling* terjadi proses digitasi pada domain spasial. Hasil dari proses *sampling* ini yaitu suatu array (matriks) berukuran $N \times M$ dan setiap elemen pada matrik ini merupakan kuantitas diskrit. Proses *sampling* citra pada Matlab menggunakan sintaks *imread* [10]. Hasil dari proses ini *sampling* ditunjukkan pada Gambar 3.5.



Gambar 3.5 Hasil sampling citra

3.2.1.2 Resizing

Proses *resizing* berfungsi untuk mengubah ukuran dari array yang telah diperoleh sebelumnya dari proses *sampling*. Proses *resizing* citra pada Matlab menggunakan sintaks *imresize* [10]. Tujuan dari proses ini adalah untuk mempermudah proses pengolahan citra pada tahap selanjutnya seperti pengurangan noise dan proses-proses lainnya. Hasil dari proses *resizing* ditunjukkan pada Gambar 3.6.



Gambar 3.6 Hasil *resizing* citra

3.2.1.3 Grayscale Citra Mobil [10]

Citra mobil yang masih dalam bentuk RGB (red green blue) diubah terlebih dahulu menjadi citra mobil dalam bentuk *grayscale* dengan level 2^8 atau 256



Gambar 3.7 Hasil *grayscale* citra

Berdasarkan Gambar 3.7 terdapat 2 ciri penting yang dapat digunakan untuk memperoleh informasi mengenai lokasi plat nomor yaitu:

1. plat nomor berada dalam posisi horizontal
2. antara karakter dan *background* plat nomor memiliki perbedaan kontras warna yang sangat tinggi.

Berdasarkan dua ciri utama ini deteksi plat nomor dapat dilakukan dengan melakukan proses *vertical edge detection* untuk mendeteksi unsur vertikal plat dan proses *rank filter* untuk mengarsir secara horizontal hasil dari *vertical edge detection*.

3.2.1.4 Vertical Edge Detection

Untuk *vertical edge detection* digunakan operator Prewitt [10] dengan tipe vertikal untuk memperoleh *vertical edge*. Operator Prewitt dipilih karena berdasarkan pengamatan penulis menunjukkan hasil terbaik.

$$\begin{matrix} -1 & 0 & 1 \end{matrix}$$

Filter *vertical Prewitt* adalah : $\begin{matrix} -1 & 0 & 1 \end{matrix}$

$$\begin{matrix} -1 & 0 & 1 \end{matrix}$$



Gambar 3.8 Vertical edge

Pada proses ini bertujuan untuk melihat jumlah *vertical edge* dari citra mobil. Dari Gambar 3.8, dapat dilihat bahwa *vertical edge* lebih banyak berkonsentrasi pada daerah sekitar plat nomor.

3.2.1.5 Cleaning Phase

Pada proses ini kita memfilter piksel-piksel yang tidak dibutuhkan (*noise*). Untuk mengurangi *noise* diberi batasan jumlah piksel dari *vertical edge* yang ada menggunakan perintah *bwareaopen* [10]. hasilnya seperti ditunjukkan pada Gambar 3.9.



Gambar 3.9 Range besar Piksel

Dari Gambar 3.9 dapat dilihat jumlah *edge* yang ada semakin sedikit, sehingga *noise* semakin berkurang.

3.2.1.6 Horizontal Rank Filtering

Seperti ditunjukkan Gambar 3.9, bahwa *vertical edge* lebih banyak berkonsentrasi di daerah sekitar plat, sehingga untuk mendeteksi koordinat lokasi plat nomor digunakan rank filter dengan orientasi horizontal dengan ukuran $M \times N$ di mana $M \gg N$ [2].

Rank Filter adalah bentuk khusus dari filter median [10]. Perbedaannya adalah jika pada filter median, setiap piksel dari citra diganti dengan median dari tetangga piksel tersebut, sedangkan pada *rank filter*, setiap piksel dari citra diganti dengan orde (rank) tertentu dari area yang dilingkupi filter. Tujuan dari penggunaan *rank filter secara horizontal* ini adalah untuk mengarsir secara horisontal daerah sekitar *vertical edge*. Penulis menggunakan *rank filter* ukuran 50×1 bernilai 1 dengan orde 44 atau dengan kata lain setiap piksel citra diganti dengan 85% dari intensitas piksel tetangga yang dilingkupi filter. Hasil dari proses ini ditunjukkan pada Gambar 3.10.



Gambar 3.10 Rank filter

3.2.1.7 Area terbesar (High Density Of Bright Edges)

Dari Gambar 3.10 terdapat beberapa *edge*, *edge* yang paling besar menunjukkan koordinat dari plat nomor. Sehingga dilakukan proses *filtering* untuk menghilangkan *noise*. Hasil dari proses ini ditunjukkan pada Gambar 3.11



Gambar 3.11 Edge maksimum

3.2.1.8 Ekstraksi Plat Nomor

Untuk mencari koordinat kotak dari lokasi plat nomor, digunakan program sebagai berikut:

```
L = bwlabeln (citra);  
S = regionprops (L, 'boundingbox');  
Koordinat = cat (1, S.BoundingBox);
```

Citra plat nomor dapat di ekstraksi dengan menggunakan fungsi *imcrop* pada matlab [10] dengan berdasarkan nilai koordinat plat nomor. Hasil ekstraksi plat nomor ditunjukkan pada Gambar 3.12.



Gambar 3.12 Citra plat nomor

3.2.2 Background Equalization

Pada proses ini terdiri dari 7 tahap, antara lain:

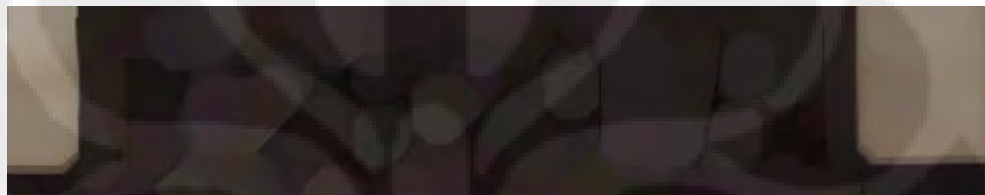
1. *Imopen background*
2. *Imsubsubtract background*
3. *Grayscaleing* citra plat nomor
4. *Imadjust* citra plat nomor
5. *Graytresh dan Binary background*

6. *Range besar piksel dari objek*

7. *Imclear border*

3.2.2.1 *Imopen Background*

Proses ini bertujuan untuk mengestimasi nilai piksel-piksel yang terdapat pada background. Pada proses ini digunakan fungsi *imopen* pada Matlab [10]. Hasil dari proses ini ditunjukkan pada Gambar 3.13.



Gambar 3.13 *Imopen background*

Fungsi *imopen* yaitu merupakan proses *erosion* yang diikuti dengan proses *dilation* [4].

3.2.2.2 *Imsubstract Background*

Proses ini bertujuan untuk menghasilkan suatu keseragaman nilai piksel pada background, hal ini dapat dilakukan dengan menggunakan fungsi *imsubstract* pada Matlab [10]. Hasil dari proses ini ditunjukkan pada Gambar 3.14.



Gambar 3.14 *Imsubstract background*

3.2.2.3 Grayscale

Pada Gambar 3.14 citra plat nomor masih dalam bentuk RGB. Citra ini harus diubah ke dalam bentuk *grayscale* supaya dapat dilakukan proses *Imadjust* pada proses selanjutnya. Untuk *Grayscale* citra digunakan fungsi *rgb2gray* [10]. Hasil dari proses ini ditunjukkan pada Gambar 3.15.



Gambar 3.15 Grayscale citra plat nomor

3.2.2.4 Contrasting

Pada Gambar 3.15 citra plat nomor masih terlihat terlalu gelap sehingga diperlukan suatu proses *contrasting* warna. Tujuan dari proses ini yaitu untuk memperjelas nilai putih dari karakter, hal ini dapat dilakukan dengan fungsi *imadjust* [10]. Hasil dari proses ini ditunjukkan pada Gambar 3.16.



Gambar 3.16 Contrasting

3.2.2.5 Proses Graythresh dan Binary

Pada Gambar 3.16 antara *background* dan karakter bisa kita bedakan dimana warna putih untuk karakter dan warna hitam untuk background. Akan tetapi warna putih pada karakter belum begitu sempurna (belum 100% putih), begitu juga dengan warna hitam pada background. Untuk mengatasi hal ini dapat dilakukan proses pengubahan jenis citra menjadi citra *binary* [2], hal ini dapat dilakukan dengan menggunakan fungsi *graythresh* [10] untuk memperoleh nilai *thresh* [10] antara

hitam dan putih kemudian untuk membuat citra *binary* menggunakan fungsi *im2bw* [10]. Hasil dari proses ini ditunjukkan pada Gambar 3.17.



Gambar 3.17 Citra *binary*

3.2.2.6 Proses filter

Pada gambar 3.17 dapat dilihat masih terdapat beberapa piksel yang bukan termasuk karakter dan background atau dengan kata lain *noise*, untuk menghilangkan *noise* ini dapat dilakukan suatu proses filter dengan memberi batasan besar piksel. Contohnya: apabila jumlah piksel dibawah 500 dan diatas 40000 (satuan piksel) maka piksel tersebut harus dihilangkan. Berikut ini *syntaks* program untuk proses filter.

```
L = bwlabeln(bw);  
S = regionprops(L, 'Area');  
im5 = ismember(L, find([S.Area] <= 40000));  
figure, imshow(im5);  
L = bwlabeln(im5);  
S = regionprops(L, 'Area');  
im6 = ismember(L, find([S.Area] >= 500));  
figure, imshow(im6);
```

Hasil dari proses ini ditunjukkan pada Gambar 3.18.



Gambar 3.18 Hasil filter

3.2.2.7 *Imclearborder*

Pada gambar 3.18 masih terdapat satu *noise* lagi yaitu piksel garis putih yang terdapat pada batas bawah dari citra. Piksel ini tidak dapat dihilangkan pada proses filter karena besar piksel yang dimiliki termasuk dalam rentang besar piksel pada karakter, untuk mengatasi hal ini yaitu dengan menggunakan fungsi *imclearborder* [10]. Fungsi ini akan menghilangkan setiap piksel yang terletak pada batas pinggir (*border*) dari citra. Hasil dari proses ini ditunjukkan pada Gambar 3.19.



Gambar 3.19 *Imclearborder*

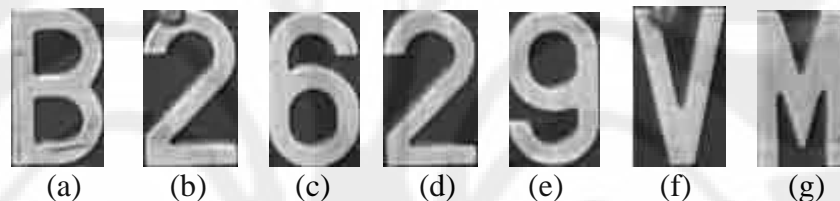
3.2.2.8 *Ekstraksi Karakter*

Pada Gambar 3.19 sudah terlihat jelas antara karakter dan background. Proses selanjutnya yaitu ekstraksi karakter. Berikut ini program untuk melakukan proses Ekstraksi masing-masing karakter dari Gambar 3.19

```
[kanan bawah]=size(im7);  
x1=1;  
while (sum(im7(:,x1))==0)  
    x1=x1+1;  
end  
x2=x1;  
while (sum(im7(:,x2))~=0)  
    x2=x2+1;  
end  
im8 = imcrop(im7,[x1 0 (x2-x1) bawah]);  
im8a= imcrop(g1,[x1 0 (x2-x1) bawah]);  
L = bwlabeln (im8);  
S = regionprops (L,'BoundingBox');  
v = cat (1, S.BoundingBox) ;  
im8 = imcrop (im8a , v);
```

Program diatas berfungsi untuk mengekstraksi karakter dengan prinsip menjumlahkan setiap piksel dari kolom ke-1 sampai kolom ke-n. Pertama-tama dijumlahkan nilai piksel pada kolom ke-1 dari Gambar 3.19, apabila jumlah piksel

nya nol (tidak terdapat piksel putih) maka akan terus dilakukan *looping* sampai diperoleh pada kolom berapa jumlah piksel tidak sama dengan nol (terdapat piksel putih). Ketika *looping* berhenti, akan diperoleh nilai kolom yang merupakan batas kiri dari karakter B (Gambar 3.19). Nilai kolom ini juga digunakan sebagai acuan untuk melakukan *looping* lagi untuk memperoleh batas kanan dari karakter B (gambar 3.19), program akan melakukan *looping*, dan akan berhenti apabila jumlah kolom sama dengan nol (tidak terdapat piksel putih). Ketika *looping* berhenti akan diperoleh nilai kolom sebagai batas kanan dari karakter B. hasil ekstraksi karakter B dari Gambar 3.19 ditunjukkan oleh Gambar 3.20. Begitu juga proses ekstraksi dilakukan berulang untuk mengekstraksi keseluruhan karakter.



Gambar 3.20 Ekstraksi karakter

3.3 LABELISASI KARAKTER

Proses labelisasi yaitu proses leveling dan reshaping dari citra karakter. Proses labelisasi sangat diperlukan untuk menentukan inputan pada proses pelatihan jaringan maupun proses pengenalan karakter. contoh labelisasi karakter B (Gambar 3.21).



Gambar 3.21 Labelisasi karakter

```
B = imread ( 'b.jpg' );
```

Fungsi `imread` untuk mengubah karakter B (Gambar 3.21) menjadi suatu matriks

```
B = rgb2gray (B);
```

Fungsi `rgb2gray` untuk menghasilkan citra *grayscale*

Citra B dari perintah diatas masih berupa citra grayscale dengan level 2^8 (256 nilai), jadi 0 untuk menyatakan hitam dan 255 untuk menyatakan putih. Citra grayscale dengan level 2^8 terlebih dahulu diubah menjadi citra *graylevel* dengan level 2^2 (4 nilai), hal ini dapat dilakukan seperti pada list program berikut:

```
B=imresize(B,[80 40]); % citra b terlebih dahulu diubah ukurannya menjadi 80 x 40
for i= 1:size(z1,1)
for j= 1:size(z1,2)
    if z1(i,j)>0 & z1(i,j)<=64
        k(i,j)=0;
    elseif z1(i,j)>64 & z1(i,j)<=128
        k(i,j)=1;
    elseif z1(i,j)>128 & z1(i,j)<=192
        k(i,j)=2;
    elseif z1(i,j)>192 & z1(i,j)<=256
        k(i,j)=3;
    end
end
end
```

Citra yang telah dilevelkan kemudian dibuat menjadi 1 kolom dengan fungsi *reshape*. Hasil citra ini kemudian masuk kedalam fungsi *make_labels* yang dipanggil dengan fungsi matlab:

```
make_labels(repetisi)
```

Besarnya repetisi merupakan jumlah *training* yang dilakukan. Dalam pembuatan *database* ini dilakukan *training* dengan memasukan beberapa contoh karakter yang berbeda tiap karakternya. Fungsi ini akan membuat 10 label angka + 26 label huruf (label1, ..., label26) yang masing - masing disimpan dalam ekstensi mat.

```
Pb= reshape(k,[],1); % Pb berukuran 3200 x 1
```

Pb ini lah yang akan menjadi inputan untuk training huruf B. untuk satu huruf B, Pb akan berukuran 3200×1 dan apabila huruf B dilatih dengan 20 sampel maka matriks Pb akan berukuran 3200×20 . Begitu juga untuk tahap pengenalan, hasil segmentasi dari plat nomor terlebih dahulu di labelisasi

3.4 PROSES PELATIHAN KARAKTER PLAT NOMOR DENGAN JARINGAN SYARAF TIRUAN KOHONEN

Proses pelatihan menggunakan JST dengan tipe *Kohonen*. Jaringan kompetitif dalam matlab digunakan untuk mengelompokkan data sedemikian hingga data yang berdekatan satu sama lain akan berada pada kelompok yang sama. Pengaturan mandiri merupakan generalisasi dari jaringan kompetitif. Keduanya merupakan jaringan tanpa supervisi (bimbingan). Neuron akan berkompetisi untuk menjadi pemenang terhadap vektor masukan yang diberikan. Bobot ke neuron pemenang akan diperbaiki pada iterasi berikutnya. Pelatihan dilakukan 2 tahap yaitu :

1. pelatihan terhadap angka
2. pelatihan terhadap huruf dengan topologi *Hextop*

Pelatihan membutuhkan sampel sebanyak 20 buah untuk setiap masing-masing huruf maupun masing-masing angka. Contohnya untuk huruf A dilatih dengan 20 buah huruf A yang bentuknya berbeda-beda dan untuk angka 1 dilatih dengan 20 buah angka 1 yang bentuknya berbeda-beda.

Untuk Simulasi dari sistem ini digunakan *Neural Network Toolbox* pada *MATLAB*. Algoritma penggunaan *toolbox* tersebut adalah :

3.4.1 Pelatihan Terhadap Angka

Proses pelatihan terhadap angka terdiri dari 3 proses.

1. Input jaringan

Pangka = [P0 P1 P2 P3 P4 P5 P6 P7 P8 P9]

Ket = P0 merupakan matriks yang berukuran 3200×20, matriks ini memiliki baris sebanyak 3200 karena hasil labelisasi dari citra angka 0 yang berukuran 80×40, dan matriks ini memiliki kolom sebanyak 20 karena sampel yang digunakan untuk melatih angka 0 sebanyak 20 buah. Hal yang serupa berlaku untuk P1 P2 dan seterusnya.

2. Proses Pembentukan Jaringan

Dalam Matlab, jaringan kompetitif dibentuk dengan perintah `newc` yang formatnya adalah sebagai berikut:

```
net = newc (PR, S, KLR, CLR);
```

ket:

PR: Matriks ukuran $R \times 2$ yang berisi nilai minimum dan maksimum masing-masing nilai vektor masukan. Pada pelatihan terhadap angka, PR berupa matriks 3200×2 dengan nilai minimum 0 dan nilai maksimum 3.

s: Jumlah neuron target. Pada pelatihan terhadap angka, s bernilai 64 yang berarti akan ada 64 kelompok untuk pengklasifikasian angka

KLR: Laju Pemahaman Kohonen (default=0.01), pada pelatihan angka KLR dipilih default

CLR: Laju Pemahaman Consience (default=0.001), CLR juga dipilih default

Bobot awal yang dibentuk adalah titik tengah matriks PR

semua parameter di set bernilai default pada pelatihan jaringan untuk angka.

3. Proses Training Jaringan

Untuk melatih jaringan digunakan perintah train

```
net = train (net,Pangka);
```

4. Proses Simulasi Jaringan

Untuk simulasi jaringan digunakan perintah sim

```
databaseangka =sim (net, pangka);  
databaseangka=full(databaseangka);  
[databaseangka k]=find(database==1);  
databaseangka=reshape(databaseangka,20,10);
```

3.4.2 Pelatihan Terhadap Huruf

Secara garis besar pelatihan jaringan pada huruf hampir sama seperti pelatihan pada angka. Perbedaannya terletak pada jumlah masuan training dan topologi pada jaringan kohonen

1. Input Jaringan

```
Phuruf = [pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pq pr  
ps pt pu pw px y pz]
```


Ket:

p_a merupakan matriks yang berukuran 3200×20 , matriks ini memiliki baris sebanyak 3200 karena hasil labelisasi dari citra huruf a yang berukuran 80×40 , dan matriks ini memiliki kolom sebanyak 20 karena sampel yang digunakan untuk melatih huruf a sebanyak 20 buah. Hal yang serupa untuk p_b p_c dan seterusnya.

2. Proses Pembentukan Jaringan

Karena topologi yang digunakan adalah topologi *Hex*top, jaringan dibentuk dengan perintah

```
net = newsom (PR, [D1,D2,...], TFCN, DFCN, OLR, OSTEPS, TLR, TND)
```

dengan :

PR: Matriks $R \times 2$ yang berisi nilai minimum dan maksimum R buah elemen masukan,

Di: Ukuran topologi layar (default = [5 8])

TFCN: Fungsi topologi (default = 'hex')
(default = 'hex')

DFCN: Fungsi jarak (default = 'linkdist')

OLR: Laju pemahaman fase pengaturanan (*ordering*). Default = 0.9

OSTEPS: Jumlah iterasi langkah pengaturan. Default = 1000

TLR : Laju Pemahaman fase penyempurnaan (*tuning*). Default =0.02

TND : Jarak sekitar neuron pemenang pada fase penyempurnaan. Default = 1

semua parameter di set bernilai default pada pelatihan jaringan untuk huruf.

3. Proses Training Jaringan

Untuk melatih jaringan digunakan perintah `train`

```
net = train (net,Phuruf);
```

4. Proses Simulasi Jaringan

Untuk simulasi jaringan digunakan perintah `sim`

```
databasehuruf =sim (net,Phuruf);
```

```

databasehuruf=full(databasehuruf);
[databasehuruf k]=find(databasehuruf==1);
databasehuruf=reshape(databasehuruf,20,26);

```

3.5 PROSES PENGENALAN KARAKTER PLAT NOMOR

Pada Proses training diperoleh *database* angka dan *database* huruf yang akan digunakan untuk proses identifikasi

1. Memasukan file *database* angka dan *database* huruf hasil dari proses training


```

load databaseangka.mat;
load databasehuruf.mat;

```
3. Memasukan citra karakter hasil dari segmentasi dan ekstraksi pada pre-processing



Gambar 3.22 Ekstraksi karakter

4. Labelisasi citra karakter

Seperti yang telah dijelaskan pada Sub Bab 3.3
5. Citra huruf di simulasikan ke *database* huruf dan citra angka disimulasikan ke *database* angka. *Database* angka dan *database* huruf merupakan kelompok kelas untuk mengklasifikasikan angka dan huruf kedalam kelasnya masing-masing. *Database* angka dan *database* huruf dapat dilihat pada lampiran 1.

Contoh pengenalan huruf B (Gambar 3.21) yang telah dilabelisasi menjadi pb

```

output=sim(net,pb);
output=full(output);
output=find(output==1);
output=output==databasehuruf;
jum=sum(output);
kolom=find(jum==max(jum));
hasil=tampilkanhuruf(kolom);
disp(hasil)

```

Contoh pengenalan angka2 yang telah dilabelisasi menjadi p2

```

output=sim(net,p2);
output=full(output);

```

```

output=find(output==1);
output=output==databaseangka;
jum=sum(output);
kolom=find(jum==max(jum));
hasil=tampilkanangka(kolom);
disp(hasil)

```

Proses pengenalan pada huruf dan angka hampir sama, perbedaannya terletak pada *database* yang digunakan dan fungsi tampilan yang berbeda, fungsi *tampilkanangka* untuk menampilkan angka dan fungsi *tampilkanhuruf* untuk menampilkan huruf. Berikut ini adalah fungsi *tampilkanangka* dan *tampilkanhuruf*: Keuntungan membuat program dalam format fungsi adalah kemudahannya untuk digunakan kembali pada program yang lain. [10]

```

function angka=tampilkanangka(input)
hasilangka=input;
if hasilangka==1
    angka='0';
elseif hasilangka==2
    angka=('1');
elseif hasilangka==3
    angka=('2');
elseif hasilangka==4
    angka=('3');
elseif hasilangka==5
    angka=('4');
elseif hasilangka==6
    angka=('5');
elseif hasilangka==7
    angka=('6');
elseif hasilangka==8
    angka=('7');
elseif hasilangka==9
    angka=('8');
elseif hasilangka==10
    angka=('9');
end

```

```

function huruf=tampilkan(input)
hasil=input;
if hasil==1
    huruf='A';
elseif hasil==2
    huruf=('B');
elseif hasil==3
    huruf=('C');
elseif hasil==4
    huruf=('D');

```

```
elseif hasil==5
    huruf=('E');
elseif hasil==6
    huruf=('F');
elseif hasil==7
    huruf=('G');
elseif hasil==8
    huruf=('H');
elseif hasil==9
    huruf=('I');
elseif hasil==10
    huruf=('J');
elseif hasil==11
    huruf=('K');
elseif hasil==12
    huruf=('L');
elseif hasil==13
    huruf=('M');
elseif hasil==14
    huruf=('N');
elseif hasil==15
    huruf=('O');
elseif hasil==16
    huruf=('P');
elseif hasil==17
    huruf=('Q');
elseif hasil==18
    huruf=('R');
elseif hasil==19
    huruf=('S');
elseif hasil==20
    huruf=('T');
elseif hasil==21
    huruf=('U');
elseif hasil==22
    huruf=('V');
elseif hasil==23
    huruf=('W');
elseif hasil==24
    huruf=('X');
elseif hasil==25
    huruf=('Y');
elseif hasil==26
    huruf=('Z');
end
```

BAB IV

HASIL UJI COBA DAN ANALISA

4.1 HASIL UJI COBA

Untuk menguji sistem yang dibuat dilakukan uji coba dengan plat nomor mobil sebanyak 15 buah. Uji coba pengenalan plat nomor terdiri dari 2 bagian:

1. ujicoba dengan menggunakan data training sebanyak 15 sampel
 - a. pengenalan dilakukan dengan jaringan kompetitif
 - b. pengenalan dilakukan dengan jaringan *Kohonen*
2. ujicoba dengan menggunakan data training sebanyak 20 sampel
 - a. pengenalan dilakukan dengan jaringan kompetitif
 - b. pengenalan dilakukan dengan jaringan *Kohonen*

Sistem yang dibuat menggunakan jumlah *training* yang sama untuk tiap karakter. Pengambilan citra mobil dilakukan pada siang hari dan sore hari berkisar dari pukul 10.00-16.00 WIB dengan kondisi cuaca cerah. Pengambilan gambar mobil dilakukan dengan kamera Canon *powershoota410* dengan resolusi 3.2 MegaPiksel.

4.1.1 Hasil Uji Coba Dengan Sampel Training 15 Buah

4.1.1.1 Pengenalan Dilakukan Dengan Jaringan Kompetitif

Pada uji coba ini digunakan sampel sebanyak 15 buah untuk masing-masing angka dan huruf dan pengenalan dilakukan dengan jaringan kompetitif. Pengambilan data dilakukan pada siang hari dan sore hari berkisar dari pukul 10.00-16.00 WIB dengan kondisi cuaca cerah. Pengambilan data dilakukan dengan kamera Canon *powershoota410* dengan resolusi 3.2 MegaPiksel. Hasil uji coba ditunjukkan pada Tabel IV.1

Tabel IV.1. Training 15 Buah (Jaringan Kompetitif)

Data Plat	huruf1	angka1	angka2	angka3	angka4	huruf2	huruf3	Hasil Pengenalan	% keberhasilan
B 8679 CB	benar	benar	salah	benar	benar	salah	benar	B 8079 JB	71.43%
B 8463 JV	benar	benar	benar	salah	benar	benar	salah	B 8403 JZ	71.43%
B 8347 PZ	benar	salah	benar	benar	benar	benar	benar	B 0347 PZ	85.71%
B 1884 MT	benar	benar	salah	salah	benar	salah	salah	B 1004 NZ	42.86%
B 1767 EM	benar	benar	salah	salah	salah	salah	salah	B 1181 BN	28.57%
B 1169 JL	benar	benar	salah	benar	benar	benar	benar	B 1069 JL	85.71%
B 1625 AB	benar	benar	salah	benar	benar	benar	benar	B 1025 AB	85.71%
B 2629 VM	benar	benar	salah	benar	benar	salah	salah	B 2029 ZH	57.14%
B 2855 MT	benar	benar	salah	salah	salah	salah	salah	B 2088 HY	28.57%
B 2976 OZ	benar	benar	benar	benar	salah	salah	benar	B 2970 GZ	71.43%
B 8652 RV	benar	salah	salah	benar	benar	salah	salah	B 0052 BZ	42.86%
B 8629 JN	benar	salah	salah	benar	salah	benar	benar	B 0020 JN	57.14%
B 2187 KU	benar	benar	benar	benar	benar	salah	salah	B 2187 WC	71.43%
B 2378 XJ	benar	benar	benar	benar	salah	salah	benar	B 2370 ZJ	71.43%
B 1781 KF	benar	benar	benar	benar	benar	benar	salah	B 1781 KB	85.71%

4.1.12 Pengenalan Dilakukan Dengan Jaringan Kohonen

Pada uji coba ini digunakan sampel sebanyak 15 buah untuk masing-masing angka dan huruf dan pengenalan dilakukan dengan jaringan *Kohonen*. Pengambilan data dilakukan pada siang hari dan sore hari berkisar dari pukul 10.00-16.00 WIB dengan kondisi cuaca cerah. Pengambilan data dilakukan dengan kamera Canon *powershoota410* dengan resolusi 3.2 MegaPiksel. Hasil uji coba ditunjukkan pada Tabel IV.2

Tabel IV.2. Training 15 Buah (Jaringan *Kohonen*)

Data Plat	huruf1	angka1	angka2	angka3	angka4	huruf2	huruf3	Hasil Pengenalan	% keberhasilan
B 8679 CB	salah	benar	benar	benar	benar	salah	salah	H 8679 ZH	57.14%
B 8463 JV	salah	benar	benar	salah	benar	benar	benar	H 8483 JV	71.43%
B 8347 PZ	salah	benar	benar	benar	salah	benar	salah	C 8340 PX	57.14%
B 1884 MT	benar	benar	benar	benar	benar	salah	benar	B 1884 HT	85.71%
B 1767 EM	salah	benar	benar	benar	benar	benar	salah	C 1767 EH	71.43%
B 1169 JL	salah	benar	benar	benar	benar	benar	benar	C 1169 JL	85.71%
B 1625 AB	salah	benar	benar	benar	benar	benar	salah	C 1625 AC	71.43%
B 2629 VM	salah	benar	benar	benar	benar	benar	salah	C 2629 VH	71.43%
B 2855 MT	salah	benar	benar	salah	salah	salah	benar	C 2888 HT	42.86%
B 2976 OZ	salah	benar	benar	benar	benar	salah	benar	C 2976 CZ	71.43%
B 8652 RV	salah	benar	benar	benar	benar	salah	benar	C 8652 HV	71.43%
B 8629 JN	salah	benar	benar	benar	benar	benar	benar	C 8629 JN	85.71%
B 2187 KU	benar	benar	benar	salah	benar	benar	salah	B 2107 KC	71.43%
B 2378 XJ	salah	benar	benar	benar	benar	benar	salah	C 2378 XC	71.43%
B 1781 KF	benar	benar	benar	salah	benar	benar	salah	B 1701 KH	71.43%

4.1.2 Hasil Uji Coba Dengan Sampel Training 20 Buah

4.1.2.1 Pengenalan Dilakukan Dengan Jaringan Kompetitif

Pada uji coba ini digunakan sampel sebanyak 20 buah untuk masing-masing angka dan huruf dan pengenalan dilakukan dengan jaringan kompetitif. Pengambilan data dilakukan pada siang hari dan sore hari berkisar dari pukul 10.00-16.00 WIB dengan kondisi cuaca cerah. Pengambilan data dilakukan dengan kamera Canon powershoota410 dengan resolusi 3.2 MegaPiksel. Hasil uji coba ditunjukkan pada Tabel IV.3

Tabel IV.3. Training 20 Buah (Jaringan Kompetitif)

Data Plat	huruf1	angka1	angka2	angka3	angka4	huruf2	huruf3	Hasil Pengenalan	% keberhasilan
B 8679 CB	salah	benar	benar	benar	benar	salah	salah	U 8679 UU	57.14%
B 8463 JV	salah	benar	benar	benar	benar	benar	salah	U 8463 JY	71.43%
B 8347 PZ	salah	benar	benar	benar	benar	benar	benar	U 8347 PZ	85.71%
B 1884 MT	salah	benar	benar	benar	benar	salah	salah	U 1884 NZ	57.14%
B 1767 EM	salah	benar	benar	benar	benar	salah	salah	U 1767 RN	57.14%
B 1169 JL	salah	benar	benar	benar	benar	benar	salah	U 1169 JS	71.43%
B 1625 AB	salah	benar	benar	benar	benar	benar	salah	U 1625 AU	71.43%
B 2629 VM	salah	benar	benar	benar	benar	salah	salah	U 2629 YR	57.14%
B 2855 MT	salah	benar	benar	benar	benar	salah	salah	U 2855 NY	57.14%
B 2976 OZ	salah	benar	benar	benar	benar	salah	benar	U 2976 UZ	71.43%
B 8652 RV	salah	benar	benar	benar	benar	salah	salah	U 8652 PY	57.14%
B 8629 JN	salah	benar	benar	benar	benar	benar	benar	U 8629 JN	85.71%
B 2187 KU	benar	benar	benar	benar	benar	benar	benar	B 2187 KU	85.71%
B 2378 XJ	salah	benar	benar	benar	salah	salah	salah	R 2379 ZS	42.86%
B 1781 KF	salah	benar	benar	benar	benar	benar	salah	0 1781 KP	71.43%

4.1.2.2 Pengenalan Dilakukan Dengan Jaringan Kohonen

Pada uji coba ini digunakan sampel sebanyak 20 buah untuk masing-masing angka dan huruf dan pengenalan dilakukan dengan jaringan *Kohonen*. Pengambilan data dilakukan pada siang hari dan sore hari berkisar dari pukul 10.00-16.00 WIB dengan kondisi cuaca cerah. Pengambilan data dilakukan dengan kamera Canon *powershoota410* dengan resolusi 3.2 MegaPiksel. Hasil uji coba ditunjukkan pada Tabel IV.4

Tabel IV.4. Training 20 Buah (Jaringan Kohonen)

Data Plat	huruf1	angka1	angka2	angka3	angka4	huruf2	huruf3	Hasil Pengenalan	% Keberhasilan
B 8679 CB	benar	benar	benar	benar	benar	benar	benar	B 8679 CB	100%
B 8463 JV	benar	benar	benar	benar	benar	benar	benar	B 8463 JV	100%
B 8347 PZ	benar	benar	benar	benar	benar	benar	benar	B 8437 PZ	100%
B 1884 MT	benar	benar	benar	salah	benar	benar	benar	B 1894 MT	85.71%
B 1767 EM	benar	benar	benar	benar	benar	benar	benar	B 1767 EM	100%
B 1169 JL	benar	benar	benar	benar	benar	benar	benar	B 1169 JL	100%
B 1625 AB	benar	benar	benar	benar	benar	benar	benar	B 1625 AB	100%
B 2629 VM	benar	benar	benar	benar	benar	benar	benar	B 2629 VM	100%
B 2855 MT	benar	benar	benar	salah	salah	benar	benar	B 2866 MT	71.43%
B 2976 OZ	benar	benar	benar	benar	benar	salah	benar	B 2976 QZ	85.71%
B 8652 RV	benar	benar	benar	salah	benar	salah	benar	B 8662 KV	71.43%
B 8629 JN	benar	benar	salah	benar	benar	benar	salah	B 8529 JM	71.43%
B 2187 KU	benar	benar	benar	benar	benar	benar	salah	B 2187 KF	85.71%
B 2378 XJ	benar	benar	benar	benar	benar	benar	benar	B 2378 XJ	100%
B 1781 KF	benar	benar	benar	benar	benar	benar	salah	B 1781 KE	85.71%

4.1.2.3 Gabungan jaringan kompetitif dan jaringan Kohonen

Pada uji coba ini digunakan sampel sebanyak 20 buah untuk masing-masing angka dan huruf dan pengenalan dilakukan dengan jaringan kompetitif untuk angka dan pengenalan dilakukan dengan jaringan kohonen untuk huruf. Pengambilan data dilakukan pada siang hari dan sore hari berkisar dari pukul 10.00-16.00 WIB dengan kondisi cuaca cerah. Pengambilan data dilakukan dengan kamera Canon *powershoota410* dengan resolusi 3.2 MegaPiksel. Hasil uji coba ditunjukkan pada Tabel IV.4

Tabel IV.5. Training 20 Buah (gabungan)

Data Plat	huruf1	angka1	angka2	angka3	angka4	huruf2	huruf3	Hasil Pengenalan	% Keberhasilan
B 8679 CB	benar	benar	benar	benar	benar	benar	benar	B 8679 CB	100%
B 8463 JV	benar	benar	benar	benar	benar	benar	benar	B 8463 JV	100%
B 8347 PZ	benar	benar	benar	benar	benar	benar	benar	B 8347 PZ	100%
B 1884 MT	benar	benar	benar	benar	benar	benar	benar	B 1884 MT	100%
B 1767 EM	benar	benar	benar	benar	benar	benar	benar	B 1767 EM	100%
B 1169 JL	benar	benar	benar	benar	benar	benar	benar	B 1169 JL	100%
B 1625 AB	benar	benar	benar	benar	benar	benar	benar	B 1625 AB	100%
B 2629 VM	benar	benar	benar	benar	benar	benar	benar	B 2629 VM	100%
B 2855 MT	benar	benar	benar	benar	benar	benar	benar	B 2855 MT	100%
B 2976 OZ	benar	benar	benar	benar	benar	salah	benar	B 2976 QZ	85.71%
B 8652 RV	benar	benar	benar	benar	benar	benar	salah	B 8652 KV	85.71%
B 8629 JN	benar	benar	benar	benar	benar	benar	salah	B 8629 JM	85.71%
B 2187 KU	benar	benar	benar	benar	benar	benar	salah	B 2187 KF	85.71%
B 2378 XJ	benar	benar	benar	benar	salah	benar	benar	B 2379 XJ	85.71%
B 1781 KF	benar	benar	benar	benar	benar	benar	salah	B 1781 KE	85.71%

4.2 PENGOLAHAN HASIL UJI COBA

Dari Tabel IV.1 sampai dengan Tabel IV.5 dapat diperoleh beberapa hal yang dapat dibandingkan, antara lain:

1. Perbandingan persentase keberhasilan dalam pengenalan pola huruf dan angka
2. Perbandingan persentase plat nomor yang berhasil diidentifikasi dengan benar.

Perbandingan dilakukan terhadap perbedaan jumlah sampel training

1. Training menggunakan 15 sampel

Dari Tabel IV.1 dan Tabel IV.2 dapat diperoleh jumlah kesalahan dalam pengenalan baik pada angka maupun pada huruf, setiap kesalahan dalam pengenalan karakter dijumlahkan untuk memperoleh persentase keberhasilan dalam pengenalan karakter. Hasil pengolahan data ditunjukkan pada Tabel IV.6.

Tabel IV.6. Persentase Akurasi Training 15 Buah

Metode Pengenalan	Σ huruf salah	Σ angka salah	% pengenalan huruf	% pengenalan angka	% pengenalan plat tepat benar
jaringan kompetitif	17	21	62.22	65%	0%
jaringan kohonen	20	6	44.44	90%	0%

2. Training menggunakan 20 sampel

Dari Tabel IV.3 dan Tabel IV.4 dapat diperoleh jumlah kesalahan dalam pengenalan baik pada angka maupun pada huruf, setiap kesalahan dalam pengenalan karakter dijumlahkan untuk memperoleh persentase keberhasilan dalam pengenalan karakter. Hasil pengolahan data ditunjukkan pada Tabel IV.7

Tabel IV.7. Persentase Akurasi Training 20 Buah

Metode Pengenalan	Σ huruf salah	Σ angka salah	% pengenalan huruf	% pengenalan angka	% pengenalan plat tepat benar
jaringan kompetitif	34	1	24.44%	98.33%	0%
jaringan kohonen	5	5	88.89%	91.67%	53.33%

Pada Tabel IV.5 pengenalan dilakukan dengan menggunakan jaringan kompetitif dan jaringan *Kohonen* dan diperoleh jumlah kesalahan dalam pengenalan karakter sehingga dapat dihitung persentase keberhasilan dalam pengenalan karakter seperti yang ditunjukkan pada Tabel IV.8.

Tabel IV.8. Persentase Akurasi Training 20 Buah (jaringan kompetitif dan kohonen)

Metode Pengenalan	Σ huruf salah	Σ angka salah	% pengenalan huruf	% pengenalan angka	% pengenalan plat tepat benar
gabungan jaringan kompetitif dan kohonen	5	1	88.89%	98.33%	60%

4.3 ANALISA SISTEM

Analisa sistem dibutuhkan untuk menentukan parameter apa saja yang dibutuhkan agar dihasilkan suatu perangkat lunak pengenalan plat nomor yang baik. Analisa dilakukan dengan memperhatikan pengaruh jumlah sampel *training* terhadap persentase keberhasilan dalam pengenalan karakter dan pengaruh metode yang digunakan terhadap persentase keberhasilan pengenalan karakter, apakah menggunakan metode jaringan kompetitif atau jaringan *Kohonen* atau menggunakan gabungan antara jaringan kompetitif dan jaringan *Kohonen*. Analisa juga dilakukan untuk mengetahui parameter-parameter yang dapat menyebabkan kegagalan pada sistem baik pada tahap *pre-processing* maupun pada tahap identifikasi

4.3.1 Pengaruh Jumlah Sampel *Training*

Jumlah training sangat berpengaruh didalam pengenalan pola angka maupun huruf, dapat dilihat dari Tabel IV.6 dan Tabel IV.7 diperoleh bahwa pada sistem yang menggunakan training sebanyak 15 sampel tidak ada satu pun plat yang berhasil diidentifikasi dengan benar, namun apabila jumlah sampel ditambah menjadi 20 sampel, terdapat 8 plat yang berhasil diidentifikasi dari total plat sebanyak 15 plat sehingga diperoleh persentase pengenalan plat dengan benar sebesar 53 %. Semakin banyak sampel yang digunakan dalam training, maka akan dihasilkan suatu jaringan *Kohonen* yang memiliki daya klasifikasi pola yang lebih baik. Hal ini bisa terjadi

karena vektor masukan pola pada pelatihan jaringan kohonen akan memiliki kemiripan ciri yang semakin jelas. Kemiripan ciri sangatlah dipentingkan di dalam pembentukan suatu kelompok pola. Berdasarkan teori dan hasil ujicoba telah ditunjukkan bahwa sistem pengenalan menggunakan sampel training 20 buah akan menghasilkan klasifikasi pola yang lebih baik.

4.3.2 Pengaruh Metode Yang Digunakan Dalam Pengenalan

Dari Tabel IV.7 dapat terlihat bahwa persentase pengenalan huruf lebih besar pada metode yang menggunakan jaringan *Kohonen* dan persentase pengenalan angka lebih besar pada metode yang menggunakan jaringan kompetitif. Sehingga apabila sistem yang dipilih menggunakan jaringan kohonen pada pengenalan angka dan jaringan kompetitif pada pengenalan huruf akan meningkatkan persentase pengenalan plat seperti yang ditunjukkan pada tabel IV.5, diperoleh 9 plat nomor yang berhasil diidentifikasi dengan benar dari total 15 plat nomor sehingga persentase pengenalan plat menjadi 60%. Pengenalan huruf lebih baik menggunakan metode jaringan kohonen karena pada iterasi jaringan kohonen bobot neuron yang diubah tidak hanya bobot garis yang terhubung ke neuron pemenang saja tapi juga bobot ke neuron-neuron sekitarnya. Jaringan *Kohonen* cocok untuk mengenali huruf karena huruf memiliki banyak pola yaitu 26 pola, karena banyak pola maka vektor masukan juga semakin beragam sehingga untuk membedakan ciri antar pola semakin susah, sehingga untuk mempermudah pengelompokkan pola, bobot neuron yang diubah tidak hanya bobot garis yang terhubung ke neuron pemenang saja tapi juga ke neuron sekitarnya. Berbeda halnya dengan pengenalan huruf yang hanya memiliki 10 pola. Pada pengenalan huruf cukup dengan menggunakan jaringan kompetitif, hal ini disebabkan pola huruf hanya sedikit sehingga ciri yang dimiliki tidak terlalu beragam sehingga untuk pengklasifikasiannya cukup dengan mengubah bobot neuron yang terhubung ke neuron pemenang saja.

4.3.3 Analisa Kesalahan Pada Tahap *Pre-Processing*

SPPNM yang penulis rancang memiliki dua tahap utama untuk *pre-processing* yang sangat menentukan keberhasilan sistem, yaitu:

1. Tahap Deteksi dan Ekstraksi Plat Nomor
2. Tahap Ekstraksi Karakter

1. Tahap Deteksi dan Ekstraksi Karakter

Tahap Deteksi dan Ekstraksi Plat Nomor adalah tahap yang paling vital. Jika tahap ini gagal, maka tahap-tahap selanjutnya gagal juga. Proses deteksi plat nomor sangat tergantung kepada ciri khusus plat nomor, yaitu banyak memiliki garis vertikal yang berdekatan. Sehingga kegagalan deteksi plat disebabkan oleh dua faktor, yaitu:

1. Terdapat garis-garis vertikal berdekatan lain yang lebih banyak dari plat nomor, misalnya bentuk ventilasi udara kap mesin pada beberapa mobil.
2. plat nomor memiliki sedikit garis vertikal berdekatan, misalnya B 1, dll.

Akan tetapi kegagalan jenis ini sangat jarang terjadi. Umumnya sistem dapat mendeteksi plat dengan baik.

2. Tahap Ekstraksi Karakter

Ekstraksi karakter akan mengalami kegagalan apabila jumlah karakter kurang dari 7 karakter, hal ini sesuai dengan batasan masalah yang penulis buat yaitu sampel plat nomor yang digunakan terdiri dari 7 karakter sehingga penulis membuat program ekstraksi yang membagi citra plat nomor seperti ditunjukkan pada Gambar 3.20 menjadi 7 bagian.

4.3.4 Analisa Kesalahan Pada Tahap Identifikasi

Analisa kesalahan dilakukan untuk mengetahui parameter apa saja yang menyebabkan sistem menjadi kurang handal didalam pengenalan pola. Berikut ini contoh plat nomor yang tidak diidentifikasi dengan benar. Sistem mengenali plat nomor yang ada pada citra yaitu B 2976 QZ yang seharusnya B 2976 OZ. Hal ini dikarenakan pola huruf O dan pola huruf Q mirip, sehingga untuk membedakan pola

yang mirip diperlukan training yang lebih banyak tidak cukup hanya dengan 20 sampel training.



Gambar 4.1 Contoh plat yang tidak diidentifikasi dengan benar

kesalahan juga bisa terjadi pada pembentukan *database* huruf maupun angka. Contohnya untuk melatih pola huruf k diperlukan 20 sampel training. 20 sampel ini mestilah terdiri dari 20 buah huruf K yang berbeda bentuknya namun masih memiliki ciri yang sama. Berikut ini adalah contoh huruf k yang tidak terlalu mirip



(a)



(b)

Gambar 4.2 Perbandingan huruf K

Antara huruf K pada Gambar 4.2(a) dan Gambar 4.2(b) kemiripannya tidak terlalu proporsional baik ditinjau dari ukuran maupun nilai intensitas, sehingga pada saat pengklasifikasian pola, kedua huruf diatas bisa saja tidak dalam satu kelompok. Berbeda halnya dengan perbandingan huruf B seperti yang ditunjukkan pada Gambar 4.3



Gambar 4.3 Perbandingan huruf B

antara huruf B pada Gambar 4.3(a) dan Gambar 4.3(b) memiliki perbedaan tapi perbedaannya tidak terlalu mencolok. Sehingga pada saat pengklasifikasian pola, kedua huruf diatas tentunya akan menjadi satu kelompok yang sama. Dari hasil uji coba terlihat bahwa pola huruf B lah yang pengenalan nya lebih baik dibandingkan dengan pengenalan pola huruf yang lain, hal ini dikarenakan data citra mobil diambil di daerah Jakarta yang memiliki nomor seri awal B. Sehingga penulis memiliki pola B yang lebih banyak dibandingkan dengan pola huruf yang lain.

Proses perubahan tingkat *graylevel quantization* yang dilakukan juga memiliki pengaruh yang signifikan. Penggunaan tingkat *graylevel quantization* yang semakin bertingkat-tingkat membuat perbedaan nilai intensitas dari setiap *pixel* semakin beragam dan pola nilai rata-rata yang akan dihasilkan juga semakin unik. Perubahan tingkat *grayscale quantization* menjadi level 2^2 (4 nilai) seperti yang dilakukan pada sistem ini sudah cukup baik, namun apabila level *graylevel quantization* lebih tinggi maka hasil yang didapat juga akan semakin baik.

Kesalahan juga bisa terjadi dikarenakan hasil ekstraksi karakter hanya di normalisasi ukuran dan labelisasi tanpa mengalami pengolahan citra yang lebih jauh sebelum dikenali seperti pembersihan citra, dilatasi dan sebagainya. Tentunya dengan pengolahan citra yang lebih baik akan mempermudah didalam pengenalan pola nantinya.



BAB V

KESIMPULAN

1. Perangkat lunak pengenalan plat nomor yang dibuat telah berhasil mengenali pola huruf dengan tingkat akurasi 88,89%, pola angka dengan tingkat akurasi 98.3%, dan pengenalan plat dengan tingkat akurasi 60%
2. Tingkat akurasi jaringan yang dibentuk didalam pengenalan pola tergantung dari jumlah sampel yang digunakan. Semakin banyak sampel yang digunakan dalam proses training maka jaringan yang dibentuk akan memiliki daya klasifikasi pola yang lebih handal
3. Jaringan kohonen sangat cocok digunakan untuk membedakan beberapa pola yang cukup banyak seperti pada pola huruf dan untuk membedakan pola yang tidak terlalu banyak seperti pola angka cukup dengan menggunakan jaringan kompetitif.
4. Untuk lebih meningkatkan akurasi pengenalan keseluruhan plat mobil, khususnya untuk pengenalan jenis huruf, diperlukan jumlah sampel data lebih banyak dan sama jumlahnya untuk seluruh huruf yang mungkin muncul.

DAFTAR ACUAN

- [1] Simon Haykin, *Neural Networks A comprehensive Foundation* (New York: Macmillan College Publishing Company., 1994.)
- [2] Vladimir Shapiro, Dimo Dimov, Stefan Bonchev, Veselin Velichop, Georgi Gluchev, “ The Automatic Plate Localization Component of a Car License Plate Recognition System.”, *Jurnal Teknologi*, III (2003) hal.1-5
- [3] Darundriyo Djoko. “Sistem Pengenalan Plat Nomor Mobil Menggunakan pencocokan pola dan JST.” Skripsi, Program Sarjana Fakultas Ilmu Komputer Universitas Indonesia, Depok, 1999, hal. 12-15
- [4] Rafael C. Gonzalez, Richard E Woods, Steven L. Edddins, *Digital Image Processing* (USA : Addison Wesley Publishing Company., 1992.)
- [5] Matlab help, *Image Processing Toolbox*, (mathworks inc.,2004)
- [6] Rinaldi Munir, *Pengolahan Citra Digital Dengan Pendekatan Algoritmik* (Bandung : Informatika Bandung.,2004)
- [7] Matlab help, *Neural Network Toolbox* (mathworks inc.,2004)

- [8] Sri Kusumadewi, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK* (Yogyakarta: Penerbit GRAHA ILMU., 2005), hal. 257
- [9] Jong Jek Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab* (Yogyakarta: Penerbit ANDI., 2004), hal. 141
- [10] Rafael C. Gonzalez, Richard E Woods, Steven L. Eddins, *Digital Image Processing using Matlab* (USA : Addison Wesley Publishing co., 1992.)
- [11] Gunaidi Abdia Away, *Matlab Programing* (Bandung: Informatika Bandung., 2006)

DAFTAR PUSTAKA

- Nugroho, Radityo A. “ Identifikasi Pola Schooling Ikan Dengan Analisis Echogram Fish Finder Menggunakan Jaringan Syaraf Tiruan.” Skripsi, Program Sarjana Fakultas Teknik Universitas Indonesia, Depok, 2006
- Hutapea, Martin B. “Identifikasi Tanda Tangan (Signature Recogniton) Menggunakan Jaringan Syaraf Tiruan (Artificial Neural Network).” Skripsi, Program Sarjana Fakultas Teknik Universitas Indonesia, Depok, 2007
- Rafael C. Gonzalez, Richard E Woods, Steven L. Edddins, *Digital Image Processing using Matlab* (USA : Addison Wesley Publishing co., 1992.)
- Away, Gunaidi Abdia, *Maltab Programing* (Bandung: Informatika Bandung., 2006)
- Rafael C. Gonzalez, Richard E Woods, Steven L. Edddins, *Digital Image Processing* (USA : Addison Wesley Publishing Company., 1992.)
- Matlab help, *Image Processing Toolbox*, (mathworks inc.,2004)
- Munir, Rinaldi, *Pengolahan Citra Digital Dengan Pendekatan Algoritmik* (Bandung : Informatika Bandung.,2004)
- Matlab help, *Neural Network Toolbox* (mathworks inc.,2004)

Kusumadewi, Sri, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK* (Yogyakarta: Penerbit GRAHA ILMU., 2005)

Ifeachor, Emmanuel. Barrie Jervis, " *Digital Signal Processing*". Prentice Hall, 2001

Kwaśnicka, Halina., " *License plate localization and recognition in camera pictures*", Jurnal, *Faculty Division of Computer Science, Wrocław University of Technology*,

Zhang, Yungang., Changsui Zhang, ' *A New Algorithm For Character Segmentation of License Plate*', Jurnal, Dept. of Automation. Tsinghua University, China.

Brad, Remus, " *License Plate Recognition System*", jurnal, Computer Science Department, "Lucian Blaga" University, Sibiu, Romania.

LAMPIRAN



Lampiran 1 Database Angka

Lampiran ini menunjukkan *database* kelas untuk angka. Dari Sub Bab 3.5 diperoleh list program sebagai berikut :

Contoh pengenalan angka2 yang telah dilabelisasi menjadi p2

```
output=sim(net,p2);% variable p2 disimulasikan ke jaringan kompetitif yang telah dilatih
output=full(output); % untuk mengConvert sparse matrix menjadi full matrix
output=find(output==1); % untuk memperoleh kelas dari P2 (angka 2)
output=output==databaseangka; % berdasarkan kelas yang diperoleh disimulasikan ke database,
                                % untuk menentukan angka 2 masuk ke dalam kelompok mana,
                                % apakah kedalam kelompok angka dua atau kelompok angka
                                % adalah 43 maka kelas 43 kebanyakan berkumpul di kelompok
                                % dua
jum=sum(output); % setiap nilai kolom dijumlahkan untuk menentukan kolom mana yang paling
                                % banyak kelas 43 nya
kolom=find(jum==max(jum)); % diperoleh kolom mana yang banyak terdapat kelas 43
hasil=tampilkanangka(kolom); % diperoleh kolom 2
disp(hasil); % ditampilkan lah angka 2
```

Database angka ditunjukkan pada Tabel 5.1

Tabel V.1. *Database* Kelas Angka

nol	satu	dua	tiga	empat	lima	enam	tujuh	delapan	sembilan
28	40	32	37	2	52	50	37	6	34
49	25	43	50	29	52	44	37	6	34
60	40	3	55	58	33	42	37	53	13
60	54	3	37	39	33	44	37	50	34
60	54	3	13	39	42	53	37	30	13
49	37	32	62	29	52	42	10	15	60
60	54	43	62	24	44	61	37	30	47
61	54	43	62	58	14	60	10	61	13
61	54	43	55	24	44	17	37	6	47



28	5	32	55	24	14	17	59	20	13
13	5	37	55	2	14	4	59	30	34
28	25	32	55	58	44	22	10	20	13
61	25	11	62	29	42	64	59	13	47
28	25	43	57	58	7	17	59	30	47
49	16	32	55	58	45	42	59	27	13
28	54	32	55	38	52	22	59	6	34
28	54	32	55	58	45	22	59	56	13
61	16	32	63	38	45	4	10	15	13
61	37	32	55	2	44	15	10	56	34
28	54	32	57	58	14	22	12	15	34

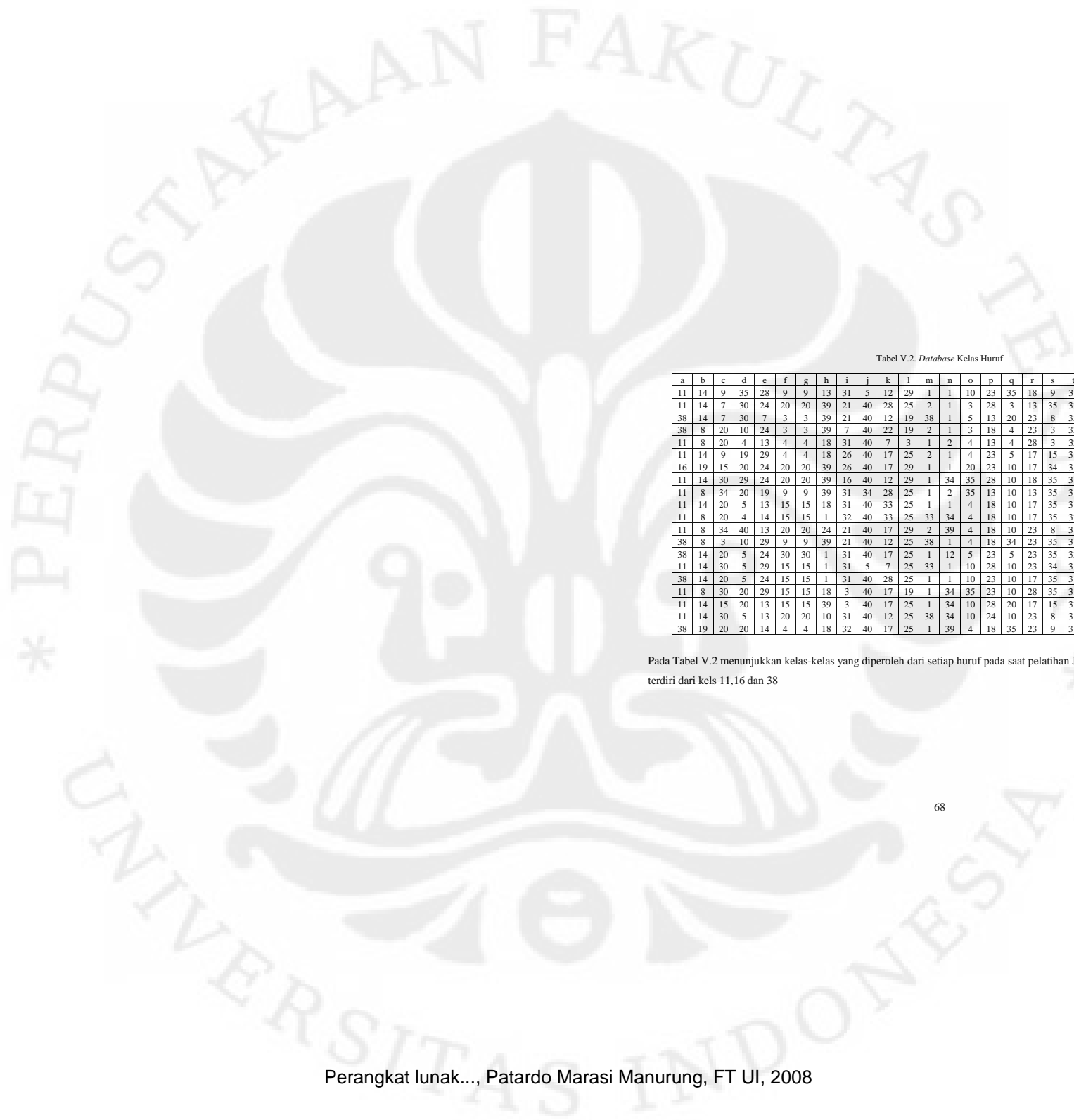
Pada Tabel V.1 menunjukkan kelas-kelas yang diperoleh dari setiap angka pada saat pelatihan JST, contohnya untuk 20 huruf a terdiri dari kelas 13, 28, 49, 60, 61

Lampiran 2 Database Huruf

Lampiran ini menunjukkan *database* kelas untuk huruf. Dari Sub Bab 3.5 diperoleh list program sebagai berikut :
Contoh pengenalan huruf B (Gambar 3.20) yang telah dilabelisasi menjadi pb

```
output=sim(net,pb);% variable pb disimulasikan ke jaringan kohonen yang telah dilatih
output=full(output); % untuk mengConvert sparse matrix menjadi full matrix
output=find(output==1); % untuk memperoleh kelas dari pb (huruf b)
output=output--databahuruf; % berdasarkan kelas yang diperoleh disimulasikan ke database,
    untuk menentukan huruf b masuk ke dalam kelompok mana,
    apakah kedalam kelompok huruf b atau kelompok huruf
    lain, mislakan dari hasil simulasi diperoleh kelas huruf b
    adalah 5 maka kelas 5 kebanyakan berkumpul di kelompok
    b
jum=sum(output); % setiap nilai kolom dijumlahkan untuk menentukan kolom mana yang paling
    banyak kelas 5 nya
kolom=find(jum==max(jum)); diperoleh kolom mana yang banyak terdapat kelas 5
hasil=tampilkanangka(kolom); diperoleh kolom b
disp(hasil); ditampikan lah angka b
```

Database huruf ditunjukkan pada Tabel 5.2



Tabel V.2. Database Kelas Huruf

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
11	14	9	35	28	9	9	13	31	5	12	29	1	1	10	23	35	18	9	31	5	36	38	21	7	26
11	14	7	30	24	20	20	39	21	40	28	25	2	1	3	28	3	13	35	33	5	37	23	21	36	26
38	14	7	30	7	3	3	39	21	40	12	19	38	1	5	13	20	23	8	33	3	7	2	21	32	21
38	8	20	10	24	3	3	39	7	40	22	19	2	1	3	18	4	23	3	33	4	36	23	21	32	27
11	8	20	4	13	4	4	18	31	40	7	3	1	2	4	13	4	28	3	32	40	36	17	21	36	21
11	14	9	19	29	4	4	18	26	40	17	25	2	1	4	23	5	17	15	31	20	36	11	21	32	27
16	19	15	20	24	20	20	39	26	40	17	29	1	1	20	23	10	17	34	31	10	39	11	26	32	21
11	14	30	29	24	20	20	39	16	40	12	29	1	34	35	28	10	18	35	31	10	33	6	22	36	26
11	8	34	20	19	9	9	39	31	34	28	25	1	2	35	13	10	13	35	31	40	37	39	16	33	21
11	14	20	5	13	15	15	18	31	40	33	25	1	1	4	18	10	17	35	31	40	37	39	21	33	26
11	8	20	4	14	15	15	1	32	40	33	25	33	34	4	18	10	17	35	32	10	36	6	21	33	21
11	8	34	40	13	20	20	24	21	40	17	29	2	39	4	18	10	23	8	31	5	37	6	33	33	26
38	8	3	10	29	9	9	39	21	40	12	25	38	1	4	18	34	23	35	31	40	37	11	7	37	26
38	14	20	5	24	30	30	1	31	40	17	25	1	12	5	23	5	23	35	33	10	37	11	16	36	26
11	14	30	5	29	15	15	1	31	5	7	25	33	1	10	28	10	23	34	31	5	7	6	21	36	21
38	14	20	5	24	15	15	1	31	40	28	25	1	1	10	23	10	17	35	31	10	36	11	26	36	26
11	8	30	20	29	15	15	18	3	40	17	19	1	34	35	23	10	28	35	31	10	12	39	21	36	26
11	14	15	20	13	15	15	39	3	40	17	25	1	34	10	28	20	17	15	33	4	36	39	21	36	26
11	14	30	5	13	20	20	10	31	40	12	25	38	34	10	24	10	23	8	31	10	36	12	21	36	26
38	19	20	20	14	4	4	18	32	40	17	25	1	39	4	18	35	23	9	31	40	36	7	21	37	21

Pada Tabel V.2 menunjukkan kelas-kelas yang diperoleh dari setiap huruf pada saat pelatihan JST, contohnya untuk 20 huruf a terdiri dari kelas 11,16 dan 38