

***IDENTIFIKASI PLAT NOMOR MOBIL DENGAN
SKELETONISASI MENGGUNAKAN JARINGAN SYARAF
TIRUAN***

SKRIPSI

Oleh :

JAMES

04 03 03 0594



**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 :

IDENTIFIKASI PLAT NOMOR MOBIL DENGAN SKELETONISASI MENGUNAKAN JARINGAN SYARAF TIRUAN

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 mendapatlan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 4 Januari 2008

(James)

NPM 0403030594

PENGESAHAN

Skripsi dengan judul :

IDENTIFIKASI PLAT NOMOR MOBIL DENGAN SKELETONISASI MENGUNAKAN JARINGAN SYARAF TIRUAN

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 skripsi.

Depok, 4 Januari 2008

Dosen pembimbing,

Dr. Ir. Arman Djohan Diponegoro M.Eng

NIP. 131 476 472

UCAPAN TERIMA KASIH

Puji dan syukur kehadiran Tuhan Yesus Kristus yang adalah Allah, Bapa dan sahabat penulis yang selalu ada dan memberi kekuatan serta atas segala berkat ,hikmat, pimpinan dan rahmat-Nya sehingga seminar ini dapat diselesaikan dengan baik. Tak lupa penulis juga mengucapkan terima kasih kepada:

Dr. Ir. Arman Djohan Diponegoro M.Eng

selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan saran, bimbingan, pengarahan, dan kemudahan lain dalam penyelesaian skripsi ini.

Jakarta, 4 Januari 2008

Penulis

James

NPM. 04 03 03 0594

	Dosen Pembimbing I. Dr. Ir. Arman Djohan Diponegoro , M.Eng
	Dosen Pembimbing I. Dr. Ir. Arman Djohan Diponegoro , M.Eng

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	ii
PENGESAHAN iii	
UCAPAN TERIMA KASIH.....	<u>iv</u>
ABSTRAK v	
ABSTRACT <u>vi</u>	
DAFTAR ISI <u>vii</u>	
DAFTAR GAMBAR <u>ix</u>	
DAFTAR TABEL <u>x</u>	
BAB 1 PENDAHULUAN.....	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN PENELITIAN	<u>2</u>
1.3 BATASAN MASALAH.....	<u>2</u>
1.4 SISTEMATIKA PENULISAN.....	3

BAB 2 TINJAUAN PUSTAKA	4
2.1 CITRA (IMAGE)	4
2.2 <i>THRESHOLDING</i>	6
2.3 SKELETONISASI.....	7
2.4 JARINGAN SYARAF TIRUAN.....	9
BAB 3 METODOLOGI PENELITIAN.	16
3.1 LANGKAH Pengerjaan	16
3.2 MENENTUKAN KARAKTERISTIK GAMBAR.....	18
3.3 <i>NEURAL NETWORK</i>	21
3.3.1 proses pelatihan (<i>training</i>).....	22
3.3.2 proses pengenalan.....	23
Bab 4 HASIL UJI COBA DAN ANALISIS	26
4.1 HASIL UJI COBA.....	27
4.2 RANGKUMAN HASIL UJI COBA	28
4.3 ANALISIS.....	29
4.3.1. analisis unjuk kerja berdasarkan ukuran <i>image</i> skeleton dan jumlah <i>training</i> terhadap keseluruhan plat nomor.....	29
4.3.2. analisis terhadap kesalahan karakter yang timbul dalam proses identifikasi.....	30
Bab 5 KESIMPULAN	32
DAFTAR ACUAN.....	33
DAFTAR PUSTAKA.....	34

DAFTAR GAMBAR

Gambar 2.1	Matriks Citra Digital(x,y).....	5
Gambar 2.2	Nilai Ambang Di Antara Dua Nilai Intensitas Yang Berbeda Jauh	6
Gambar 2.3	Piksel Pusat P_i Dengan 8 Tetangganya	8
Gambar 2.4	Hasil Skeletonisasi Langkah 1[2].....	9
Gambar 2.5	Hasil Skeletonisasi Langkah 2[2].....	9
Gambar 2.6	Diagram Blok Jaringan Syaraf Tiruan.....	10
Gambar 2.7	Fungsi Aktivasi <i>Sigmoid</i>	11
Gambar 2.8	Fungsi Aktivasi <i>Tansig</i>	11
Gambar 2.9	Fungsi Aktivasi <i>Purelin</i>	12
Gambar 2.10	Contoh Arsitektur Jaringan <i>Backpropagation</i>	12

Gambar 3.1	Diagram Alir Proses <i>Training</i>	16
Gambar 3.2	Diagram Alir Proses Identifikasi.....	17
Gambar 3.3	Plat Nomor * . jpg.....	17
Gambar 3.4	Plat Nomor * . bmp.....	18
Gambar 3.5	Plat Nomor * . bmp Yang Telah Dipotong 1/3 Bagian.....	18
Gambar 3.6	Pilar-Pilar Indikator Suatu Karakter Secara Vertikal.....	19
Gambar 3.7	Hasil Proses Skeletonisasi.....	20
Gambar 3.8	Diagram Alir Proses <i>Training</i> Jaringan Syaraf Tiruan.....	24
Gambar 3.9	Diagram Alir Proses Identifikasi Jaringan Syaraf Tiruan...	26
Gambar 4.1	Tampilan Hasil Identifikasi Plat Nomor Pada <i>GUI</i>	27
Gambar 4.2	Plat Nomor Dengan Kerusakan Karakter Akibat Baut.....	32

DAFTAR TABEL

Tabel 4.1	Hasil Identifikasi Plat Nomor Dengan 35x <i>Training</i>	28
Tabel 4.2	Hasil Identifikasi Plat Nomor Dengan 25x <i>Training</i>	29
Tabel 4.3	Tingkat Akurasi Sistem Pengenalan Plat Nomor	30
Tabel 4.4	Tampilan <i>Skeleton</i> Dari Karakter Plat Nomor Yang Tidak Dikenali Dengan Benar.....	31

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Di era modern sekarang ini, kemajuan teknologi juga dapat dilihat di banyak tempat termasuk pula ditemukannya kemajuan teknologi yang menghiasi pemandangan di tempat parkir mobil terutama di pusat keramaian. Kecanggihan tersebut terlihat dari penggunaan komputer yang terhubung dengan kamera yang digunakan untuk mengenali setiap plat nomor mobil yang memasuki area perpakiran secara otomatis tanpa perlu campur tangan manusia lagi untuk mencatat satu per satu secara manual. Penggunaan teknologi baru ini muncul karena diakibatkan banyaknya masalah yang timbul dari pencatatan plat nomor yang dilakukan secara manual oleh manusia, dimana dua masalah umum yang paling sering timbul dari sistem manual tersebut terutama berhubungan dengan kecepatan kerja dan *human error* yang dapat terjadi. Pada umumnya pencatatan secara manual yang dilakukan oleh manusia membutuhkan waktu yang agak lama sehingga dapat mengakibatkan antrean yang panjang, di samping itu tingkat ketelitian setiap petugas yang melakukan pencatatan juga berbeda-beda sehingga dapat menyebabkan terjadinya kesalahan dalam melakukan pencatatan.

Teknologi baru berupa sistem pengenalan plat nomor mobil secara otomatis ini diawali dengan cara melakukan proses ekstraksi fitur/karakter plat nomor, yang dapat dilakukan baik dengan menggunakan metode *edge detection* maupun dengan menggunakan metode skeletonisasi. Metode *edge detection* telah dibuat pada tugas akhir sebelumnya dengan judul "Sistem Pengenalan Plat Nomor Mobil Menggunakan Hidden Markov Model" oleh Robby Nelson [1] dan "Pengenalan Plat Mobil Menggunakan

Jaringan Syaraf Tiruan Metode Kohonen” oleh Patardo Marasi Manurung [2] sehingga metode yang digunakan dalam mengekstraksi fitur/karakter pada plat nomor dalam skripsi ini adalah metode skeletonisasi yang akan menghasilkan kerangka/skeleton dari setiap karakter yang ada dan kemudian diikuti oleh langkah selanjutnya yaitu berupa proses identifikasi atau pengenalan karakter yang ada. Proses ini membutuhkan suatu metode pembelajaran yang dapat bertindak seperti layaknya otak manusia sehingga dapat memiliki kemampuan untuk mengenali plat nomor secara otomatis.

Dewasa ini beberapa metode pembelajaran telah diperkenalkan untuk menjawab persoalan di atas, di antaranya dengan menggunakan metode jaringan syaraf tiruan (*Neural network*), *Hidden Markov Model (HMM)*, atau *Fuzzy logic*. Seperti juga pada tugas akhir dengan judul ”Pengenalan Plat Nomor Mobil Dengan Skeletonisasi Dan Menggunakan Metode Pelatihan Hidden Markov Model” oleh Ferry Sugiarto [3], dalam skripsi ini, kerangka/*skeleton* hasil ekstraksi selanjutnya akan diproses namun dengan menggunakan metode pembelajaran jaringan syaraf tiruan (*Neural network*) yang bekerja berdasarkan pembobotan serta dapat melakukan perbaikan bobot secara otomatis pada setiap langkahnya, sehingga akan dihasilkan suatu bobot yang tepat dan dapat digunakan dalam proses pengenalan/identifikasi plat nomor.

1.2 TUJUAN PENULISAN

Penulisan skripsi ini bertujuan untuk mengenali plat nomor mobil dengan mengaplikasikan jaringan syaraf tiruan (*Neural Network*) dengan metode pelatihan *backpropagation* pada sistem identifikasi citra plat nomor mobil dengan menerapkan prinsip-prinsip pengolahan citra pada proses ekstraksi fitur citra dan menganalisa hasil pengenalan plat nomor yang dilakukan dengan memvariasikan parameter-parameter yang ada yaitu berupa jumlah *training* serta ukuran *image* kerangka/*skeleton* yang digunakan sebagai *input* dari jaringan syaraf tiruan.

1.3 BATASAN MASALAH

Pembatasan masalah pada penulisan ini adalah pengidentifikasian jenis plat nomor mobil berwarna hitam putih yang sesuai dengan peraturan DITLANTAS RI berdasarkan pada karakteristik hasil skeletonisasi tiap karakter yang ada pada plat nomor.

Citra yang digunakan terbatas hanya pada plat nomor itu sendiri dan merupakan hasil pengambilan gambar dari arah depan secara tegak lurus. Identifikasi dilakukan dengan memanfaatkan nilai piksel hasil ekstraksi fitur dengan proses skeletonisasi sampel citra plat nomor dengan menggunakan metode pembelajaran jaringan syaraf tiruan.

1.4 SISTEMATIKA PENULISAN

BAB I PENDAHULUAN

Menjelaskan latar belakang, tujuan penulisan, batasan masalah dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Menjelaskan dasar teori mengenai citra, *thresholding*, skeletonisasi dan metode jaringan syaraf tiruan.

BAB III METODOLOGI PENELITIAN

Menjelaskan langkah-langkah perancangan program identifikasi plat nomor mobil dengan pengolahan citra menggunakan metode jaringan syaraf tiruan

BAB IV HASIL UJI COBA DAN ANALISIS

Menjelaskan hasil dari simulasi serta analisis dari hasil tersebut.

BAB V KESIMPULAN

Memberikan kesimpulan dari keseluruhan skripsi ini

BAB II

TINJAUAN PUSTAKA

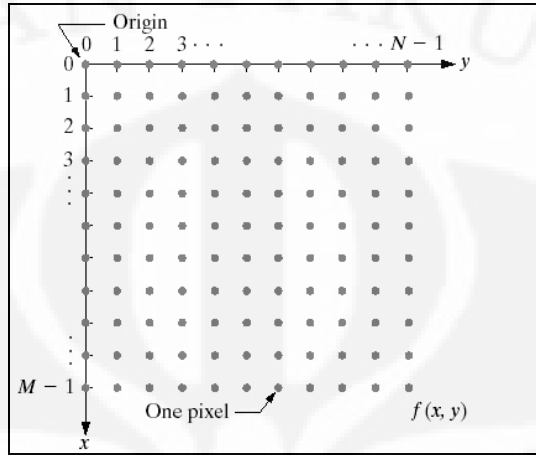
2.1 CITRA (*IMAGE*)

Citra dapat digolongkan berdasarkan terdefinisi atau tidaknya citra tersebut pada setiap titik spasial (x,y) dan terhingga atau tidak terhingganya nilai kecerahan citra. Berdasarkan penggolongan ini, suatu citra dibagi menjadi kontinu-kontinu, kontinu-diskrit, diskrit-kontinu, dan diskrit-diskrit. Citra kontinu-kontinu adalah citra dengan presisi yang tak terhingga, terdefinisi pada setiap titik spasial. Citra ini sering disebut citra analog atau citra kontinu. Citra ini dapat di representasikan dengan presisi yang terhingga untuk menghasilkan citra kontinu diskrit. Citra jenis ini dihasilkan dari proses kuantisasi yang memetakan nilai kecerahan citra yang sebenarnya (tak berhingga) menuju nilai kecerahan yang berhingga yang mampu diakomodasi oleh proses komputer.

Citra juga bisa tetap memiliki tingkat kecerahan yang tak berhingga namun hanya terdefinisi pada kumpulan titik spasial yang terbatas. Citra jenis ini dihasilkan dari proses sampling yang menghasilkan citra diskrit kontinu. Karena komputer digital beroperasi pada presisi yang terbatas, maka komputer digital cocok untuk citra diskrit-diskrit. Pada citra jenis ini koordinat spasial dan tingkat kecerahan dikuantisasi menuju presisi numerik yang mampu ditangani komputer digital. Citra jenis ini biasa disebut citra digital atau citra diskrit.

Teknologi masa kini memungkinkan suatu citra digital dapat diproses dengan mudah menggunakan komputer, untuk melakukan hal itu pertama-tama citra tersebut disimpan di dalam komputer dengan bentuk yang tepat agar citra tersebut dapat dimanipulasi oleh program komputer. Analogi yang paling umum dan praktis dalam melakukan hal ini adalah dengan membagi keseluruhan citra yang ada ke dalam sekumpulan sel-sel diskrit yang menyerupai persegi-persegi kecil yang tampak seperti titik sehingga citra digital dapat direpresentasikan ke dalam suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya disebut sebagai elemen gambar/piksel/*pixel/picture element/pels*. Setiap piksel yang ada

ini selanjutnya akan diberi nilai tertentu yang merepresentasikan warna objek pada bagian itu. Pada gambar 2.1 di bawah ini ditunjukkan gambar matriks citra digital (x,y):



Gambar 2.1. Matriks citra digital (x,y) [4]

Pengolahan citra digital mengacu pada suatu fungsi intensitas cahaya $f(x,y)$, dimana harga x dan y merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (x,y) merupakan tingkat kecerahan (*brightness*) citra pada titik tersebut. Fungsi citra $f(x,y)$ ini harus didigitalisasikan secara amplitudo (intensitas) maupun spasial. Proses digitalisasi dari koordinat spasial (x,y) disebut *Image sampling*, sedangkan digitalisasi amplitudo disebut *gray level quantization*.

Pada dasarnya jenis-jenis citra yang umum digunakan terdiri dari *color image*, *grayscale image*, dan *binary image*. Pada *color image* atau citra berwarna, nilai f merepresentasikan tiga komponen warna pada setiap titik pada koordinat spasial di mana tiga komponen warna tersebut adalah merah, hijau dan biru (RGB), sehingga untuk citra berwarna, f menjadi [4] :

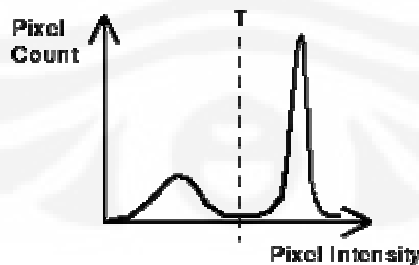
$$f(x,y) = \{ f_{merah}(x,y), f_{hijau}(x,y), f_{biru}(x,y) \} \dots\dots\dots(2.1)$$

Citra berwarna tersebut dapat diinterpretasikan sebagai tumpukan (*stack*) dari tiga citra. Di dalam skripsi ini citra plat nomor yang direkam oleh kamera adalah citra berupa *color image*. Namun citra berwarna ini selanjutnya diubah menjadi *binary image* atau citra hitam putih dengan memberikan suatu nilai ambang batas pada citra awal dengan tujuan untuk memisahkan *background* yang ditandai oleh warna hitam dengan objek citra tersebut (*foreground*) yang ditandai oleh warna putih.

2.2 THRESHOLDING

Thresholding digunakan untuk mengatur derajat keabuan yang ada pada citra, dengan *thresholding* maka derajat keabuan dapat diubah sesuai dengan keinginan. Teknik ini memisahkan bagian gambar yang sesuai dengan objek (*foreground*) dengan latar belakangnya (*background*). Metode ini juga digunakan untuk mengkonversi data *image* menjadi data biner dengan tujuan agar proses selanjutnya menjadi mudah. Masukan dari proses ini dapat berupa citra *grayscale* atau citra RGB dimana implementasi yang paling sederhana akan menghasilkan citra biner atau *binary image*. *Binary image* yang dihasilkan dari proses *thresholding* ini dapat diatur sedemikian rupa sehingga citra ini hanya memiliki dua macam warna saja yaitu warna hitam yang merepresentasikan *background* dan diwakili oleh *level 0* serta warna putih yang merepresentasikan *foreground* dan diwakili oleh *level 1*.

Proses ini dapat diaplikasikan dengan menerapkan suatu parameter berupa nilai ambang batas intensitas piksel sehingga setiap piksel penyusun citra dibandingkan dengan nilai ambang batas yang diaplikasikan, jika intensitas piksel citra lebih tinggi daripada nilai ambang maka piksel *output* akan diset menjadi 1 atau putih tetapi jika intensitas piksel citra lebih rendah daripada nilai ambang maka piksel *output* akan diset menjadi 0 atau hitam, *Thresholding* dapat digunakan dan menghasilkan keluaran yang baik jika citra yang akan diproses memiliki perbedaan yang mencolok antara intensitas piksel *foreground* dengan intensitas piksel *background*-nya seperti ditunjukkan oleh histogram pada gambar 2.2 di bawah ini:



Gambar 2.2. Nilai ambang di antara dua nilai intensitas yang berbeda jauh

2.3 SKELETONISASI

Skeletonisasi merupakan salah satu cara yang dapat digunakan untuk mereduksi objek atau citra awal menjadi suatu representasi yang lebih sederhana dan lebih *compact* namun sifat-sifat objek yang umum seperti topologi maupun karakteristik ukuran dari citra asli masih dapat digambarkan. Kerangka (*skeleton*) mengekspresikan konektivitas struktural dari suatu komponen utama sebuah objek dengan hanya menampilkan citra objek tersebut selebar satu piksel. Teknik seperti ini dapat diterapkan pada banyak bidang, di antaranya dalam penyelesaian masalah pengenalan karakter (*character recognition*) termasuk di dalam skripsi ini dimana skeletonisasi digunakan untuk mengambil informasi-informasi yang penting saja dari tiap karakter yang ada pada sebuah plat nomor mobil untuk mempermudah proses selanjutnya. *Thinning* merupakan metode dasar dalam skeletonisasi. Proses ini bekerja dengan teknik pengulangan, dimana teknik ini mengekstrak kerangka dari sebuah objek sebagai hasilnya. Pada setiap tahapannya, proses *thinning* akan mengikis piksel-piksel yang ada dimulai dari piksel yang terletak pada bagian terluar dari objek dan dilakukan berulang-ulang sampai proses *thinning* tersebut tidak mungkin dapat dilakukan lagi, sehingga yang tertinggal hanyalah kerangka (*skeleton*) dari objek tersebut. Proses penghapusan ini dilakukan dengan hati-hati agar tidak merubah topologi dari objek. Kerangka tersebut yang selanjutnya merepresentasikan bentuk dari sebuah objek dalam jumlah piksel yang kecil. Algoritma *thinning* di bawah ini hanya dapat diterapkan pada *binary image* saja, yaitu citra hitam putih yang hanya terdiri dari piksel/*level* 0 dan 1 atau 0 dan 255. dimana pada umumnya warna hitam (*background*) akan ditunjukkan oleh *level* 0 ,sedangkan objek yang umumnya berwarna putih (*foreground*) ditunjukkan oleh *level* 1 atau 255. Berikut adalah langkah-langkah algoritma skeletonisasi tersebut, yaitu:

1. Tentukan suatu kontour P_1 dan beri label P_2 sampai P_9 untuk ke-8 tetangganya, dimulai dari titik diatas P_1 (yaitu P_2) dan seterusnya sampai P_9 sesuai dengan arah jarum jam, seperti pada gambar 2.3:

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Gambar 2.3. Piksel pusat P_1 dengan 8 tetangganya

Diasumsikan nilai dari kontour adalah 1 dan nilai dari gambar latarnya adalah 0.

2. Tandai kontour P_1 yang akan dihapus jika memenuhi syarat-syarat seperti di bawah ini:

a. $2 \leq N(P_1) \leq 6$ (1)

$N(P_1)$ merupakan jumlah 8 tetangga dari P_1 yang bernilai 1.

$$N(P_1) = P_2 + P_3 + \dots + P_8 + P_9 \quad (2)$$

b. $S(P_1) = 1$ (3)

$S(P_1)$ adalah jumlah transisi 0-1 dengan urutan $P_2, P_3, \dots, P_8, P_9$.

c. $P_2 \cdot P_4 \cdot P_6 = 0$ (4)

d. $P_4 \cdot P_6 \cdot P_8 = 0$ (5)

Jika keempat syarat diatas terpenuhi maka P_1 akan dihapus.

3. Jika salah satu kondisi-kondisi diatas tidak terpenuhi maka tandai kembali kontour P_1 yang nantinya akan dihapus bila memenuhi kondisi-kondisi seperti dibawah ini:

a. $2 \leq N(P_1) \leq 6$ (6)

b. $S(P_1) = 1$ (7)

c. $P_2 \cdot P_4 \cdot P_8 = 0$ (8)

d. $P_2 \cdot P_6 \cdot P_8 = 0$ (9)

Syarat-syarat pada persamaan (1),(3),(4),(5) akan disebut langkah 1 dan syarat-syarat (6),(7),(8),(9) akan disebut langkah 2. Gambar 2.4 dan 2.5 berikut adalah hasil *thinning* dari langkah 1 dan langkah 2 pada karakter huruf H:



Gambar 2.4. Hasil skeletonisasi langkah 1[4]



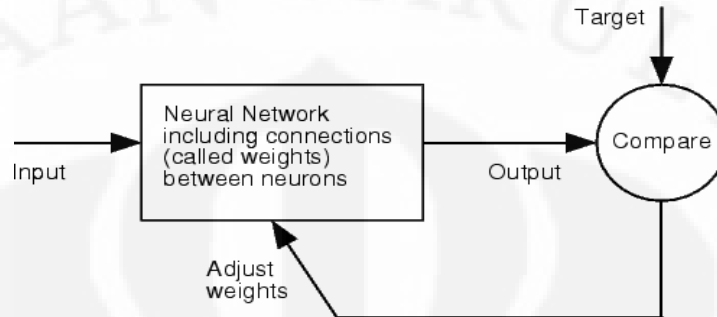
Gambar 2.5. Hasil skeletonisasi langkah 2[4]

2.4 JARINGAN SYARAF TIRUAN

Jaringan syaraf tiruan atau *Neural network* merupakan teknik yang digunakan untuk membangun program yang cerdas untuk memproses informasi dengan pemodelan yang mensimulasikan cara kerja jaringan syaraf seperti pada otak manusia. Sehingga dapat dikatakan bahwa jaringan syaraf tiruan menggunakan konsep kerja dari syaraf otak manusia untuk menyelesaikan perhitungan pada komputer di mana performa jaringan ini bergantung kepada arsitektur, algoritma pembelajaran dan fungsi aktivasi yang digunakan. Seperti otak manusia, fungsi dari jaringan ditentukan oleh hubungan antara *neuron*. Hubungan antara *neuron* ini disebut bobot (*weight*). Untuk mendapatkan fungsi tertentu dapat dilakukan dengan melakukan pelatihan (*training*) dengan menyesuaikan nilai bobot dari masing-masing *neuron*.

Pada umumnya jaringan syaraf tiruan dilatih (*trained*) agar *input* mengarah ke *output target* yang spesifik. Jadi jaringan dilatih terus menerus hingga mencapai kondisi dimana *input* sesuai dengan *target* yang telah ditentukan. Pelatihan dimana setiap *input*

diasosiasikan dengan *target* yang telah ditentukan disebut pelatihan terarah (*Supervised learning*).



Gambar 2.6. Diagram blok jaringan syaraf tiruan[5]

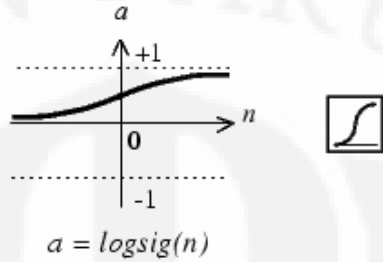
Saat ini yang sering digunakan pada jaringan syaraf tiruan adalah pelatihan terarah (*Supervised learning*). Prinsip dasar pelatihan ini adalah membandingkan antara *input* dengan *target* untuk menyesuaikan nilai bobot pada setiap kali *training*, Hal ini dilakukan agar pada pelatihan selanjutnya, nilai bobot yang didapat akan semakin mendekati dan akhirnya mencapai kondisi dimana *input* telah sesuai dengan *target output*. Selain pelatihan terarah, ada juga pelatihan tidak terarah dimana tidak ditentukan *target output*-nya namun metode pelatihan ini jarang digunakan.

Salah satu algoritma yang digunakan dalam suatu pelatihan jaringan syaraf tiruan adalah algoritma *backpropagation* yang merupakan salah satu algoritma pelatihan terarah. Algoritma *backpropagation* biasa digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* tersebut, tahap perambatan maju (*forward propagation*) harus dilakukan terlebih dahulu. Pada perambatan maju *neuron-neuron* akan diaktifkan dengan menggunakan fungsi aktivasi yang dapat didiferensiasikan di antaranya :

1. *Sigmoid* [6] :

$$y = f(x) = \frac{1}{1 + e^{-ax}} \dots\dots\dots(2.2)$$

dimana: $f'(x) = af(x)[1 - f(x)]$ dan fungsinya dapat dilihat pada gambar 2.7:



$a = \text{logsig}(n)$
Log-Sigmoid Transfer Function

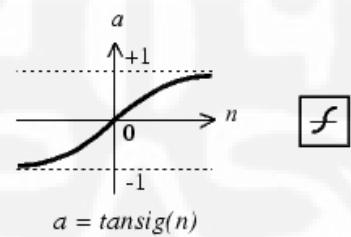
Gambar 2.7. Fungsi aktivasi *sigmoid*

2. *Tansig* [6] :

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \dots\dots\dots(2.3)$$

$$\text{atau } y = f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \dots\dots\dots(2.4)$$

dimana : $f'(x) = [1+f(x)][1-f(x)]$ dan fungsinya dapat dilihat pada gambar 2.8:



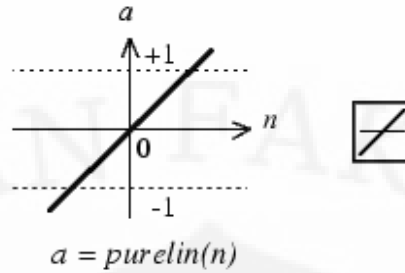
$a = \text{tansig}(n)$
Tan-Sigmoid Transfer Function

Gambar 2.8. Fungsi aktivasi *tansig*

3. *Purelin* [6] :

$$y = f(x) = x \dots\dots\dots(2.5)$$

dimana $f'(x) = 1$ dan fungsinya dapat dilihat pada gambar 2.9:



$a = \text{purelin}(n)$
Linear Transfer Function

Gambar 2.9. Fungsi aktivasi *purelin*

Pada gambar 2.10 dapat dilihat arsitektur jaringan *backpropagation* yang terdiri dari 3 unit (*neuron*) pada lapisan *input* yaitu x_1 , x_2 , dan x_3 ; 1 lapisan tersembunyi dengan 2 *neuron* yaitu z_1 dan z_2 ; serta 1 unit pada lapisan *output*, yaitu y . Bobot yang menghubungkan x_1 , x_2 , dan x_3 dengan *neuron* pertama pada lapisan tersembunyi adalah v_{11} , v_{21} dan v_{31} . (v_{ij} ; bobot yang menghubungkan *neuron input* ke- j pada suatu lapisan ke *neuron* ke- i pada lapisan sesudahnya) Bobot bias yang menuju ke *neuron* pertama dan kedua pada lapisan tersembunyi adalah b_{11} dan b_{12} Bobot yang menghubungkan bobot z_1 dan z_2 dengan *neuron* lapisan *output* adalah w_1 dan w_2 . Bobot bias b_2 menghubungkan lapisan tersembunyi dengan lapisan *output*. Fungsi aktivasi digunakan antar lapisan *input* dengan lapisan tersembunyi dan lapisan tersembunyi dengan lapisan *output*.

Gambar 2.10. Contoh arsitektur jaringan *backpropagation* [6]

Algoritma *backpropagation* adalah [6] :

- a. Inisialisasi bobot (ambil bobot awal dengan nilai *random* yang cukup kecil)
- b. Tetapkan : Maksimum Epoch, *Target error*, dan *learning rate* (α)
- c. Inisialisasi : Epoch = 0, MSE = 1.
- d. Kerjakan langkah-langkah berikut selama (Epoch < Maksimum Epoch) dan (MSE > Target Error) :

1. Epoch = Epoch + 1

2. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan :

Feedforward :

- a. Tiap-tiap unit *input* ($x_i = 1, 2, 3, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Tiap-tiap unit pada lapisan tersembunyi ($Z_j, j=1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal input berbobot :

$$z_in_j = b1_j + \sum_{i=1}^n x_i v_{ij} \dots \dots \dots (2.6)$$

dimana $b1_j$ merupakan bias pada *hidden layer*

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya :

$$z_j = f(z_in_j) \dots \dots \dots (2.7)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

- c. Tiap-tiap unit *output* ($Y_k, k=1, 2, 3, \dots, m$) menjumlahkan sinyal-sinyal *input* terboboti.

$$y_in_k = b2_k + \sum_{j=1}^p z_j w_{jk} \dots \dots \dots (2.8)$$

dimana $b2_k$ merupakan bias pada *output layer*

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya :

$$y_k = f(y_in_k) \dots \dots \dots (2.9)$$

Dan kirimkan sinyal *output* tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Langkah (b) dilakukan sebanyak jumlah lapisan tersembunyi.

- d. tiap-tiap unit *output* ($Y_k = 1,2,3,\dots,m$) menerima *target* pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*-nya :

$$\delta 2_k = (t_k - y_k) f'(y - in_k) \dots\dots\dots(2.10)$$

$$\varphi 2_{jk} = \delta_k z_j \dots\dots\dots(2.11)$$

$$\beta 2_k = \delta_k \dots\dots\dots(2.12)$$

Dimana t_k merupakan target

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk menghitung nilai w_{jk}):

$$\Delta w_{jk} = \alpha \varphi 2_{jk} \dots\dots\dots(2.13)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai b_{2k}):

$$\Delta b_{2k} = \alpha \beta 2_k \dots\dots\dots(2.14)$$

Langkah(d) ini juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi *error* dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

- e. tiap-tiap unit tersembunyi ($Z_j, j=1,2,3,\dots,p$) menjumlahkan delta *input*nya (dan unit-unit yang berada pada lapisan yang ada di atasnya):

$$\delta - in_j = \sum_{k=1}^m \delta 2_k w_{jk} \dots\dots\dots(2.15)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error* :

$$\delta 1_j = \delta - in_j f'(z - in_j) \dots\dots\dots(2.16)$$

$$\varphi 1_{ij} = \delta 1_j x_j \dots\dots\dots(2.17)$$

$$\beta 1_j = \delta 1_j \dots\dots\dots(2.18)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{ij} = \alpha \varphi 1_{ij} \dots\dots\dots(2.19)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai b_{1j})

$$\Delta b_{1j} = \alpha \beta_{1j} \dots \dots \dots (2.20)$$

Tiap-tiap unit *output* (Y_k , $k = 1, 2, 3, \dots, m$) memperbaiki bias dan bobotnya ($j=0, 1, 2, \dots, p$):

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk} \dots \dots \dots (2.21)$$

$$b_{2k} \text{ (baru)} = b_{2k} \text{ (lama)} + \Delta b_{2k} \dots \dots \dots (2.22)$$

f. Tiap-tiap unit tersembunyi ($Z_j = j=1, 2, 3, \dots, p$) memperbaiki bias dan bobotnya ($i=0, 1, 2, \dots, n$):

$$v_{ij} \text{ (baru)} = v_{ij} \text{ (lama)} + \Delta v_{ij} \dots \dots \dots (2.23)$$

$$b_{1j} \text{ (baru)} = b_{1j} \text{ (lama)} + \Delta b_{1j} \dots \dots \dots (2.24)$$

3. Hitung Mean Square Error

Setelah dilakukan algoritma tersebut pada jaringan maka kita akan mendapatkan jaringan yang sudah *ditraining*. Sehingga untuk melakukan indentifikasi, dapat dilakukan dengan langsung memberikan *input* dan jaringan akan mengklasifikasikannya sesuai dengan bobot-bobot yang diperoleh dari proses *training* sebelumnya. Dimana besar *mean square error* ditunjukkan persamaan berikut:

$$E = \frac{1}{2} \sum_{j=1}^m [t_j(k) - y_j(k)]^2 \dots \dots \dots (2.25)$$

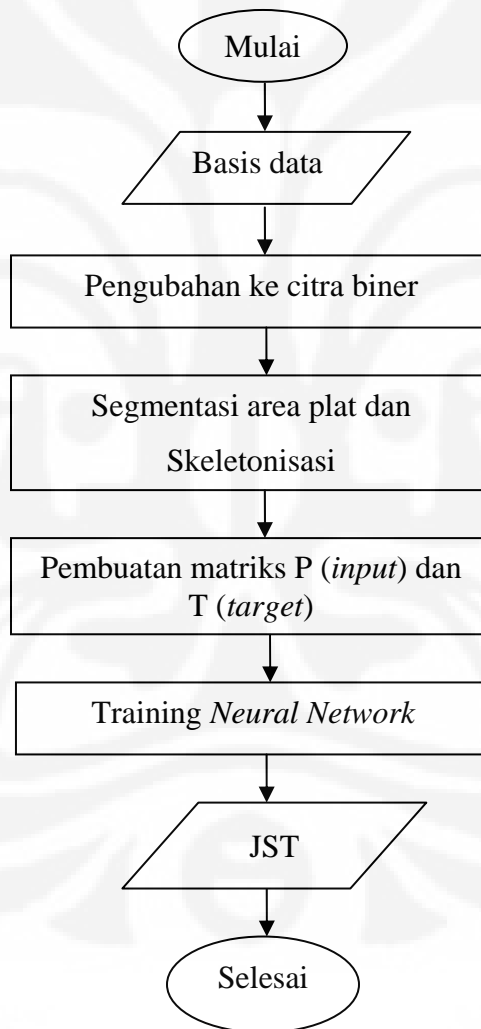
$$= \frac{1}{2} \sum_{j=1}^m e_j^2(k) \dots \dots \dots (2.26)$$

BAB III

METODOLOGI PENELITIAN

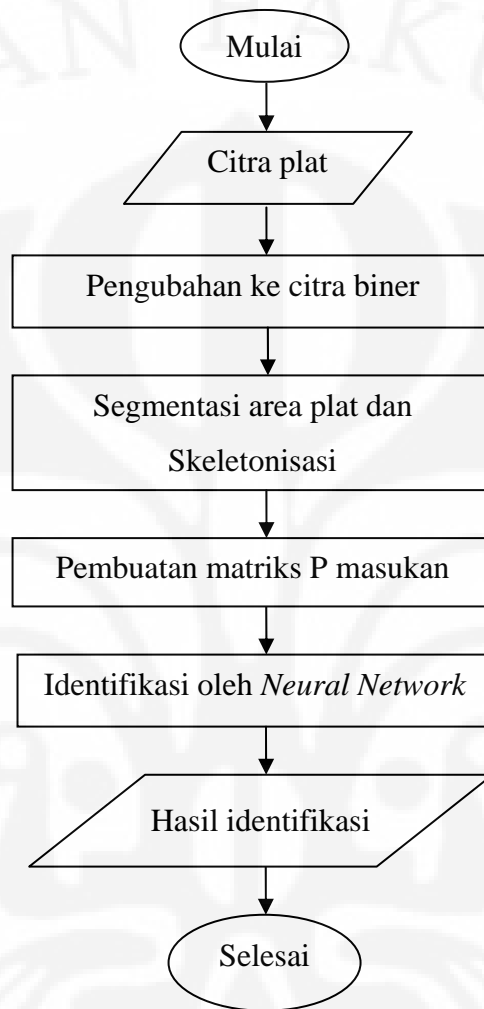
3.1 LANGKAH Pengerjaan

Proses ini terdiri dari 2 tahap yaitu proses pelatihan (*training*) dan proses pengenalan (identifikasi). Diagram alir untuk *training Neural Network* (jaringan syaraf tiruan) ditunjukkan pada Gambar 3.1:



Gambar 3.1. Diagram alir proses *training*

Dan diagram alir untuk proses pengenalan/identifikasi ditunjukkan pada Gambar 3.2:



Gambar 3.2. Diagram alir proses identifikasi

Pada skripsi ini digunakan data-data berupa citra plat nomor mobil yang didapatkan dari hasil pemotretan yang dilakukan di beberapa tempat parkir, namun citra yang digunakan hanya sebatas plat nomornya saja yang berjumlah 30 buah citra plat nomor. Pada Gambar 3.3 berikut diperlihatkan 2 contoh plat nomor yang digunakan pada skripsi ini:



Gambar 3.3. Plat nomor * .jpg

3.2 MENENTUKAN KARAKTERISTIK GAMBAR

Citra plat nomor yang masih dalam format *.jpg pada awalnya direpresentasikan ke dalam bentuk matriks dengan menggunakan fungsi *MatLab*TM *imread*, kemudian diubah menjadi citra plat hitam putih (dalam format *.bmp) dengan menggunakan fungsi *im2bw* sehingga pada citra tersebut hanya terdiri dari dua macam nilai piksel/level saja sebagai penyusunnya yaitu *level* 0 dan 1 atau *level* 0 dan 255. Hal ini bertujuan agar nilai-nilai informasi yang cukup disediakan oleh piksel-piksel citra plat lebih sedikit sehingga meringankan proses kerja yang perlu dilakukan berikutnya seperti ditunjukkan oleh Gambar 3.4 berikut:



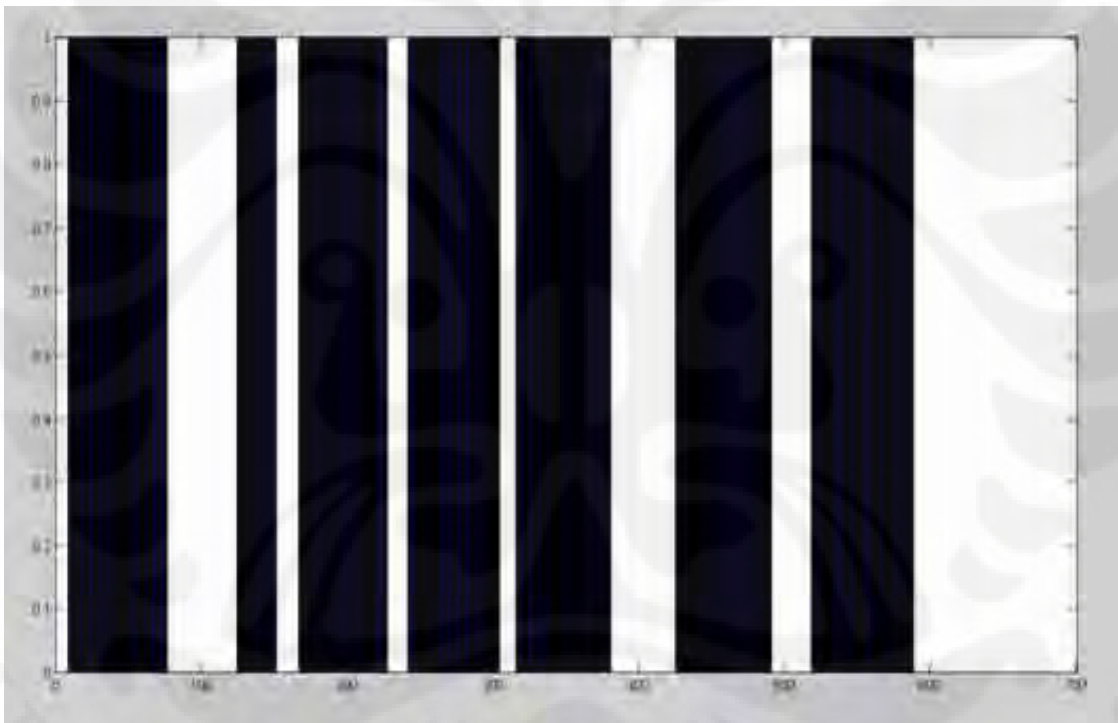
Gambar 3.4. Plat nomor * .bmp

Setelah didapatkan gambar hitam putih atau citra biner, langkah berikutnya yang dilakukan adalah penentuan batas bawah dari plat tersebut, di mana penentuan tersebut dilakukan dengan tujuan mempersempit ruang lingkup dari citra tersebut agar dapat mempermudah pelaksanaan proses selanjutnya. Penetapan batas bawah dilakukan ini dengan cara memotong kurang lebih sepertiga bagian bawah plat nomor tersebut atau dengan kata lain hanya dua per tiga bagian dari tinggi plat tersebut yang diambil untuk diproses. Dimana hasil pemotongan citra plat tersebut ditunjukkan oleh Gambar 3.5 di bawah ini:



Gambar 3.5. Plat nomor * .bmp yang telah dipotong 1/3 bagian

Selanjutnya dilakukan proses *scanning* secara vertikal terhadap piksel-piksel yang ada pada citra tersebut yang dilakukan baik terhadap piksel/*level* 1 yang menunjukkan adanya suatu karakter maupun *level* 0 yang menunjukkan tidak terdapatnya karakter apapun pada bagian tersebut (*background*). Jika terdapat sekumpulan piksel 1 dalam satu kolom dengan panjang kurang dari 0.08 kali tinggi citra plat maka titik tersebut akan dianggap sebagai noda sehingga keseluruhan kolom tersebut akan dijadikan kolom yang berwarna putih, jika sebaliknya, maka keseluruhan kolom tersebut akan dijadikan kolom yang berwarna hitam sebagai *foreground* atau tanda bahwa di kolom tersebut terdapat suatu karakter sehingga akan didapatkan suatu gambar pada *workspace* yang menyerupai pilar-pilar vertikal seperti ditunjukkan Gambar 3.6 di bawah ini:



Gambar 3.6. Pilar-pilar indikator suatu karakter secara vertikal

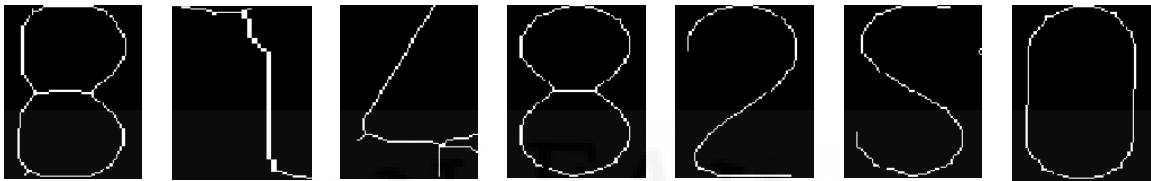
Berdasarkan pilar-pilar hitam dan putih tersebut kemudian ditentukan posisi awal dan akhir dari pilar hitam yang ada dengan cara membandingkan keadaan antara kolom yang satu dengan satu kolom sebelumnya ataupun antara kolom yang satu dengan satu kolom sesudahnya sehingga untuk keadaan suatu kolom berisi suatu karakter sedangkan kolom sebelumnya tidak terdapat karakter apapun maka akan didapatkan suatu nilai

posisi awal, tetapi jika didapatkan keadaan sebaliknya maka yang akan didapatkan adalah suatu nilai berupa posisi akhir dari pilar atau karakter tersebut yang akan digunakan nantinya untuk mendapatkan lebar suatu karakter. Lebar tiap karakter tersebut kemudian digunakan pada proses selanjutnya yaitu proses segmentasi dimana setiap karakter beserta *noise* yang satu dengan yang lainnya di dalam satu plat yang sama selanjutnya dipisahkan. Setelah disegmentasi, dilakukan pemisahan antara karakter dengan *noise* dengan cara melakukan pengecekan terhadap kondisi tertentu yaitu terhadap tinggi dan lebar tiap-tiap hasil segmentasi yang didapatkan dimana jika ternyata tinggi minimal hasil segmentasi tersebut 0.5 kali tinggi plat nomor dengan lebar 0.03 kali lebar keseluruhan plat nomor, maka hasil segmentasi tersebut akan dianggap sebagai karakter dan bukan *noise*.

Di samping itu, untuk setiap hasil segmentasi yang terbentuk di atas, dilakukan proses *scanning* kembali namun kali ini dilakukan secara horizontal guna mendapatkan hal yang serupa seperti sebelumnya yaitu jumlah atau banyaknya piksel 1 yang menandakan suatu karakter. Jika jumlah 1 yang ada bernilai nol atau dengan kata lain pada baris tersebut tidak terdapat karakter sedikitpun maka baris tersebut secara keseluruhan akan dijadikan baris berwarna putih namun jika sebaliknya, keseluruhan baris tersebut akan dihitamkan.

Setelah didapatkan blok-blok hitam horizontal dari proses di atas kemudian dilakukan proses penentuan posisi awal ataupun akhir dari blok-blok hitam tersebut mirip seperti penentuan posisi awal dan akhir kolom yaitu dengan membandingkan pula keadaan antara baris yang satu dengan satu baris sebelumnya ataupun antara suatu baris dengan baris sesudahnya.

Setelah semua proses di atas dilakukan, terhadap karakter-karakter yang ada pada citra plat nomor tersebut berikutnya dilakukan proses skeletonisasi dengan menggunakan fungsi yang sudah terintegrasi dalam *MatLab*TM sehingga kemudian dihasilkan hasil berupa karakter-karakter plat nomor yang tipis dan hanya merupakan sumbu pusat karakter yang menyerupai kerangka/*skeleton* dari suatu karakter aslinya seperti ditunjukkan oleh Gambar 3.7 berikut:



Gambar 3.7. Hasil proses skeletonisasi

3.3 NEURAL NETWORK

Pada bagian ini akan dilakukan 2 bagian penting yaitu proses pelatihan (*training*) dan proses pengenalan (identifikasi). Pada proses *training* dibutuhkan sebuah parameter karakteristik dari citra untuk dijadikan *input* bagi jaringan syaraf tiruan. Hal ini dilakukan dengan menggunakan nilai skeletonisasi tiap karakter dari masing-masing citra plat. Jadi jaringan syaraf tiruan ini menggunakan 25 dan 35 *input training* yang masing-masingnya terdiri dari 36 buah gambar *skeleton* hasil skeletonisasi tiap karakter yang diambil dari sampel citra plat. Sampel-sampel *skeleton* yang diambil tersebut harus dapat merepresentasikan pola dari masing-masing karakter karena nantinya akan dijadikan acuan dalam melakukan proses identifikasi.

Model jaringan syaraf tiruan yang digunakan adalah *backpropagation*. Jaringan ini menggunakan 2 buah lapisan/*layer* dengan jumlah *neuron* pada layer pertama (*hidden layer*) sebanyak 5 buah, sedangkan pada *layer* kedua (*output*) sebanyak 1 buah. Pada *layer* pertama digunakan fungsi aktivasi *tansig*, sedangkan untuk lapisan *output*-nya digunakan fungsi aktivasi *purelin*. Di samping itu jaringan ini menggunakan sistem *training Supervised training* dan fungsi *training* yang digunakan adalah *traingd* (*gradient Descent*).

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

Proses *training* :

1. Menentukan *Input* untuk training.

$P = [\text{nilai piksel dari sampel yang dipetakan ke dalam matriks berukuran luas gambar} \times \text{jenis sampel}] ;$

2. Menentukan *target set* dari jaringan

$T_{\text{huruf}} = [1 \ 2 \ 3 \dots 26] ;$ sebanyak sampel training yang ada
 $T_{\text{angka}} = [1 \ 2 \ 3 \dots 10] ;$ sebanyak sampel training yang ada

3. Membangun jaringan dan menetapkan banyaknya *neuron* tiap lapisan dan fungsi-fungsi aktivasi yang digunakan :

```
net = newff(minmax(P),[5 1],{'tansig' 'purelin'},'traingd');
```

4. Selanjutnya menentukan *maximum epoch*, *goal*, *learning rate*, *show step*.

```
net.trainParam.epochs = 2000;  
net.trainParam.goal = 1e-5;  
net.trainParam.lr = 0.01;  
net.trainParam.show = 10;
```

5. melakukan pembelajaran (*training*)

```
net = train(net,P,T);
```

Proses pengenalan/*recognition* :

1. Memasukan *file* nilai piksel *skeleton* matriks sampel citra yang akan diidentifikasi

```
load(nama file.mat');
```

2. Mensimulasikan *file* tersebut kedalam jaringan yang telah *training* untuk mendapatkan *output*.

```
output = sim(net,P);
```

3. Menentukan hasil identifikasi dengan membulatkan nilai *output* jaringan terlebih dahulu dan menyesuaikan dengan *target*.

```
huruf = 'ABCDE....'  
angka = '0123.....'  
output = round(output);  
hasil =huruf(output); untuk huruf  
hasil =angka(output); untuk angka
```

3.3.1 Proses Pelatihan (*Training*)

Selain membutuhkan *training set* yang dapat merepresentasikan pola karakter dari masing-masing citra sebagai *input*, untuk melakukan proses *training* juga harus ditentukan *target set* yang nantinya akan dituju oleh *input* dari jaringan syaraf tiruan. Setiap *input* memiliki pasangan *output* masing-masing, dan *target* bersifat sebagai pemilih terhadap *input* untuk proses klasifikasi. Jaringan syaraf tiruan bertugas mengarahkan *input* menuju *target* dengan menyesuaikan nilai bobot dari masing-masing *nodenya*. Setelah proses *training* dilakukan, *input* akan langsung diklasifikasikan menuju *target* yang sesuai berdasarkan bobot yang didapatkan pada saat *training*.

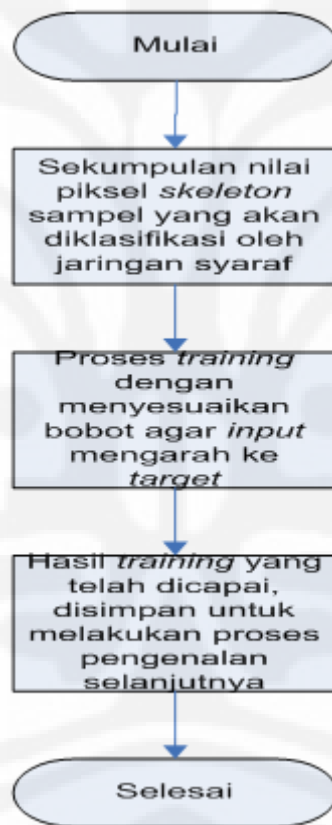
Target set adalah matriks dengan ukuran 1x26 (sebanyak jumlah folder *training*) untuk huruf dan 1x10 (sebanyak jumlah folder *training*) untuk angka. Matriks tersebut adalah :

Target huruf = [1 2 3.....26]

Target angka = [1 2 3.....10]

elemen matriks inilah yang akan dituju oleh *input* sesuai dengan bobot jaringan untuk proses klasifikasi.

Gambar 3.8 berikut menunjukkan diagram alir proses *training* jaringan syaraf tiruan untuk menentukan nilai-nilai bobot yang akan diaplikasi pada proses selanjutnya.



Gambar 3.8. Diagram alir proses *training* jaringan syaraf tiruan

3.3.2 Proses Pengenalan

Pada proses pengenalan sistem akan berusaha mengenali nilai-nilai piksel *skeleton* sampel citra yang dijadikan *input* pada sistem. Pada proses ini sistem berfungsi sebagai alat klasifikasi dari sampel citra yang dijadikan *input*. Setiap *input* memiliki

pasangan *target* masing-masing dan sistem akan mengarahkan *input* tersebut ke *target* yang paling sesuai.

Proses pengenalan pada jaringan syaraf tiruan dilakukan dengan mengklasifikasikan *input* menuju *target* yang sesuai dengan proses *training*. Pengklasifikasian ini berdasarkan bobot tiap *node* yang diperoleh pada proses *training*. Jadi pada proses klasifikasi tidak terjadi perubahan atau penyesuaian bobot. Proses klasifikasi inilah yang dijadikan dasar dalam menentukan jenis karakter menggunakan parameter karakteristik dari citra yaitu nilai-nilai piksel *skeleton* sebagai *input* bagi sistem.

Nilai piksel *skeleton* sampel citra yang dimasukkan kedalam sistem akan diarahkan ke *target* yang sesuai. Pada sistem ini *target set* yang digunakan adalah matriks dengan ukuran $1 \times (26 \times \text{jumlah folder})$ untuk huruf dan $1 \times (10 \times \text{jumlah folder})$ untuk angka sehingga *input* yang dimasukkan ke dalam sistem akan diarahkan ke salah satu elemen matriks yang telah dijadikan *target*. Dan elemen matriks *target* tersebut yang menjadi dasar pengenalan huruf dan angka dari sampel yang dimasukkan.

Huruf:

Huruf A = [1]

Huruf B = [2]

Huruf C = [3]

Huruf D = [4]

Huruf E = [5]

Huruf F = [6]

Huruf G = [7]

Huruf H = [8]

Huruf I = [9]

Huruf J = [10]

Huruf K = [11]

Huruf L = [12]

Huruf M = [13]

Huruf N = [14]

Huruf O = [15]

Huruf P = [16]

Huruf Q = [17]

Huruf R = [18]

Huruf S = [19]

Huruf T = [20]

Huruf U = [21]

Huruf V = [22]

Huruf W = [23]

Huruf X = [24]

Huruf Y = [25]

Huruf Z = [26]

Angka:

Angka 0 = [1]

Angka 1 = [2]

Angka 2 = [3]

Angka 3 = [4]

Angka 4 = [5]

Angka 5 = [6]

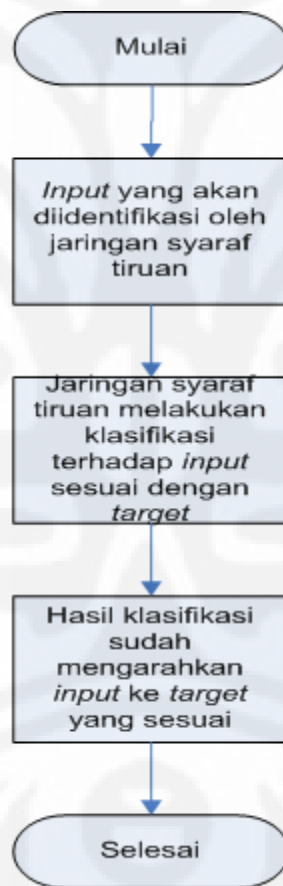
Angka 6 = [7]

Angka 7 = [8]

Angka 8 = [9]

Angka 9 = [10]

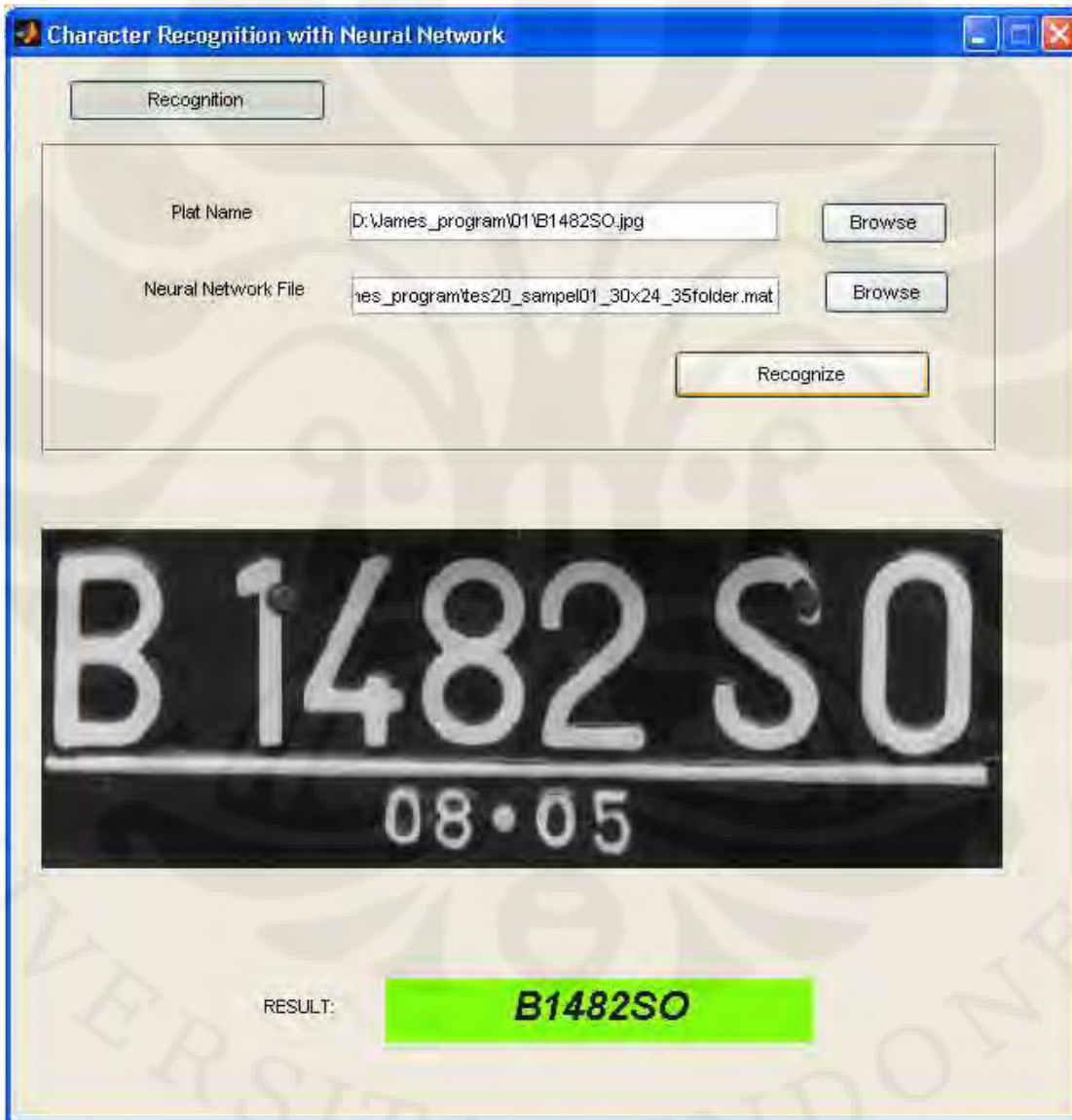
Pada Gambar 3.9 berikut ini ditunjukkan diagram alir dari proses identifikasi plat nomor menggunakan jaringan syaraf tiruan :



Gambar 3.9. Diagram alir proses identifikasi jaringan syaraf tiruan

BAB IV HASIL UJI COBA DAN ANALISIS

Uji coba yang dilakukan dalam skripsi ini menggunakan 30 buah citra plat nomor mobil yang didapatkan dari hasil pemotretan. Semua citra plat nomor tersebut akan dicoba untuk dikenali oleh jaringan syaraf tiruan dan hasilnya akan ditampilkan pada *GUI (Graphic User Interface)* yang ada. Pada Gambar 4.1 berikut ditunjukkan tampilan hasil identifikasi salah satu citra plat nomor oleh *GUI MatLab™*:



Gambar 4.1. Tampilan hasil identifikasi plat nomor pada *GUI*

Citra plat nomor yang hendak digunakan untuk diujicoba oleh sistem dipilih dengan menekan tombol *browse* pada bagian *plat name* layar tampilan sistem, langkah selanjutnya yang harus dilakukan adalah memilih hasil *setting* bobot-bobot yang dihasilkan dari proses *training* sebelumnya yang disimpan dengan menekan tombol *browse* pada bagian *Neural Network file*. Setelah semua parameter yang dipilih lengkap maka proses identifikasi akan dijalankan dengan menekan tombol *recognize*. Hasil identifikasi kemudian ditampilkan pada bagian *result*.

4.1 HASIL UJI COBA

Uji coba terhadap keseluruhan citra plat nomor yang ada dilakukan berdasarkan dua parameter yang ada di antaranya berupa parameter *training* serta parameter ukuran *image skeleton* yang digunakan baik sebagai basis data maupun dalam proses identifikasi. Dari uji coba yang dilakukan maka diperoleh hasil unjuk kerja sistem yang ditunjukkan oleh Tabel 4.1 dan 4.2 berikut :

Tabel 4.1. Hasil Identifikasi Plat Nomor Dengan 35x Training

No.	Plat nomor	ukuran 30x24	ukuran 100x50	ukuran 100x80
1	AB 7055 PE	AB 7055 PE	AB 7055 PE	AB 7055 PD
2	B 20 EH	B 20 EH	B 20 EH	B 20 EH
3	B 138 LK	B 138 LK	B 138 LK	B 138 KK
4	B 168 JF	B 168 JF	B 168 JF	B 168 JE
5	B 1044 JY	B 1044 JY	B 1044 JY	B 1044 JY
6	B 1395 NW	B 1395 NW	B 1395 NW	B 1395 MW
7	B 1411 NV	B 1411 NV	B 1411 NV	B 1411 NU
8	B 1482 SO	B 1482 SO	B 1482 SO	B 1482 SO
9	B 1670 HC	B 1670 HC	B 1670 HC	B 1670 HC
10	B 2295 QC	B 2295 QC	B 2295 QC	B 2295 QC
11	B2752 QY	B2752 QY	B2752 QY	B2752 QY
12	B 2908 BB	B 2908 BB	B 2908 BB	B 2908 BB
13	B 7925 GC	B 7925 GC	B 7925 GC	B 7925 GC
14	B 8048 SK	B 8048 SK	B 8048 SK	B 8048 SK
15	B 8079 JY	B 8079 JY	B 8079 JY	B 8079 JY
16	B 8192 SU	B 8192 SU	B 8192 SU	B 8192 SU
17	B 8198 MS	B 8198 MS	B 8198 MS	B 8198 MS
18	B 8419 RX	B 8419 RX	B 8419 RX	B 8419 RX
19	B 8615 MR	B 8615 MR	B 8615 MR	B 8615 MR
20	B 8661 KZ	B 8661 KZ	B 8661 KZ	B 8661 KZ
21	B 8912 RV	B 8912 RV	B 8912 RV	B 8912 RV
22	B 8953 RU	B 8953 RU	B 8953 RU	B 8953 RU
23	B 8966 GZ	B 8966 GZ	B 8966 GZ	B 8966 GZ
24	B 8906 VH	B 8906 VH	H 6783 EA	B 8906 VH

25	B 8903 PH	B 8913 PH	B 8903 PH	B 8903 PH
26	B 342 XV	A 342 XV	A 342 XV	B 342 XV
27	B 1264 SS	B 1064 SS	I 2464 SS	I 2274 SS
28	B 8887 LY	B 8287 LX	B 8387 JY	B 8587 LY
29	B 8456 VC	K 8456 VC	H 8456 VC	I 8456 VC
30	B 2123 VC	L 2124 VC	B 2121 VC	B 2103 VC
	Total plat yang dikenali dengan benar	24 plat	24 plat	21 plat

Tabel 4.2. Hasil Identifikasi Plat Nomor Dengan 25x *Training*

No.	Plat nomor	ukuran 30x24	ukuran 100x50	ukuran 100x80
1	AB 7055 PE	AB 7055 PE	AB 7055 PE	AB 7055 PF
2	B 20 EH	B 20 EH	B 20 EH	B 20 FH
3	B 138 LK	B 138 LK	B 138 LK	B 138 MK
4	B 168 JF	B 168 JF	B 168 JF	B 168 JG
5	B 1044 JY	B 1044 JY	B 1044 JY	B 1044 JY
6	B 1395 NW	B 1395 NW	B 1395 NW	B 1395 OW
7	B 1411 NV	B 1411 NV	B 1411 NV	B 1411 OW
8	B 1482 SO	B 1482 SO	B 1482 SO	B 1482 SP
9	B 1670 HC	B 1670 HC	B 1670 HC	B 1670 HC
10	B 2295 QC	B 2295 QC	B 2295 QC	B 2295 QC
11	B2752 QY	B2752 QY	B2752 QY	B2752 QY
12	B 2908 BB	B 2908 BB	B 2908 BB	B 2908 BB
13	B 7925 GC	B 7925 GC	B 7925 GC	B 7925 GC
14	B 8048 SK	B 8048 SK	B 8048 SK	B 8048 SK
15	B 8079 JY	B 8079 JY	B 8079 JY	B 8079 JY
16	B 8192 SU	B 8192 SU	B 8192 SU	B 8192 SU
17	B 8198 MS	B 8198 MS	B 8098 MS	B 8098 NS
18	B 8419 RX	B 8419 RX	B 8419 RX	B 8419 RX
19	B 8615 MR	B 8615 MR	B 8615 MR	B 8615 NR
20	B 8661 KZ	B 8661 KZ	B 8661 KZ	B 8661 KY
21	B 8912 RV	B 8912 RV	B 8912 RV	B 8912 RV
22	B 8953 RU	B 8953 RU	B 8953 RU	B 8953 RV
23	B 8966 GZ	B 8966 GZ	B 8966 GZ	B 8966 GY
24	B 8906 VH	A 6538 PL	A 6807 MK	E 6828 PL
25	B 8903 PH	F 7550 EF	B 7905 LI	H 6914 LH
26	B 342 XV	C 033 VY	F 313 OX	D 434 WU
27	B 1264 SS	F 3084 RU	I 1250 KI	H 2463 NL
28	B 8887 LY	O 2366 KL	G 3143 JR	E 6363 JQ
29	B 8456 VC	G 0538 UM	G 4326 PC	N 7357 QI
30	B 2123 VC	A 2305 BA	G 3257 GN	D 2043 OK
	Total plat yang dikenali dengan benar	23 plat	22 plat	11 plat

4.2 RANGKUMAN HASIL UJI COBA

Dari hasil uji coba yang dilakukan maka dapat dibandingkan tingkat akurasi dari sistem berdasarkan ukuran *image skeleton* serta jumlah *training* yang dilakukan terhadap jaringan syaraf tiruan seperti ditunjukkan pada Tabel 4.3 :

Tabel 4.3. Tingkat Akurasi Sistem Pengenalan Plat Nomor

	Ukuran 30x24	Ukuran 100x50	Ukuran 100x80
35x <i>training</i>	80%	80%	70%
25x <i>training</i>	76.67%	73.33%	36.67%

4.3 ANALISIS

Analisis yang dilakukan pada program ini dapat mencakup 2 hal yakni :

1. Analisis unjuk kerja berdasarkan ukuran *image skeleton* dan jumlah *training* terhadap keseluruhan plat nomor yang diujicoba.
2. Analisis kesalahan karakter/*skeleton* pada sistem identifikasi yang menggunakan ukuran *image* sebesar 30x24 piksel dengan jumlah *training* sebanyak 35x .

4.3.1. Analisis unjuk kerja berdasarkan ukuran *image skeleton* dan jumlah *training* terhadap keseluruhan plat nomor.

Berdasarkan Tabel 4.3 di atas, tampak bahwa tingkat akurasi uji coba terhadap plat nomor secara utuh meningkat seiring dengan penambahan jumlah *training* yang dilakukan terhadap jaringan syaraf tiruan dimana ditunjukkan hasil identifikasi menggunakan 35x *training* lebih baik dibandingkan dengan sistem pengenalan yang hanya menggunakan 25x *training*. Hal ini dikarenakan semakin banyak jumlah *training* yang dilakukan, maka jaringan syaraf tiruan akan memiliki kesempatan lebih besar untuk mencari bobot-bobot yang lebih tepat untuk diaplikasikan pada setiap interkoneksi *node* yang ada.




Di samping itu, parameter lain yang cukup berperan dalam peningkatan akurasi sistem ini adalah ukuran *image skeleton* yang digunakan dari tiap-tiap karakter yang ada baik sebagai basis data maupun yang digunakan dalam proses identifikasi, namun efek yang ditimbulkan oleh parameter ini berbanding terbalik terhadap efek yang ditimbulkan oleh parameter jumlah *training*. Dari Tabel 4.3 tersebut, sekaligus dapat ditunjukkan bahwa dengan kenaikan ukuran *image skeleton* akan disertai dengan penurunan tingkat akurasi yang tercapai. Hal tersebut disebabkan semakin besar ukuran *image* yang digunakan maka semakin banyak pula kemiripan ciri antara *image skeleton* yang satu

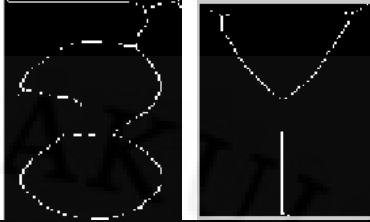
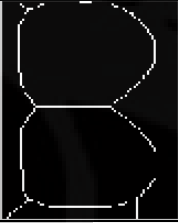
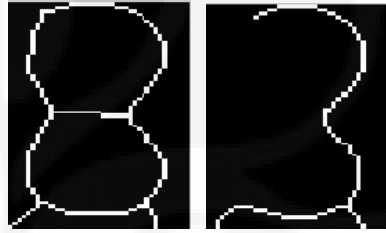
dengan yang lainnya sehingga dapat mengacaukan proses identifikasi yang dilakukan. Sebaliknya, semakin kecil ukuran *image* maka nilai informasi yang disediakan oleh piksel-piksel penyusunnya akan semakin unik sehingga perbedaan ciri antara *skeleton* yang satu dengan yang lainnya akan semakin jelas dan memudahkan jaringan syaraf tiruan dalam melakukan proses pengenalan. Oleh sebab itu, tingkat akurasi tertinggi yaitu sebesar 80% didapatkan oleh sistem yang menggunakan ukuran *image skeleton* sebesar 30x24 piksel dengan jumlah *training* sebanyak 35x.

4.3.2. Analisis terhadap kesalahan karakter yang timbul dalam proses identifikasi.

Dari Tabel 4.3, tingkat akurasi tertinggi yang dapat dicapai hanya sebesar 80% dari total plat nomor sebanyak 30 buah, hal ini menandakan masih ada sekitar 6 buah plat nomor dimana komponen-komponen karakter dalam tiap-tiap plat nomor tersebut tidak seluruhnya dapat dikenali dengan benar. Pada Tabel 4.4 ini contoh dari 6 buah plat nomor yang tidak dapat dikenali dengan benar:

Tabel 4.4. Tampilan Skeleton Dari Karakter Plat Nomor Yang Tidak Dikenali Dengan Benar

No.	Plat nomor	Dikenali sebagai	Hasil skeletonisasi
1	B 8903 PH	B 8913 PH	
2	B 342 XV	A 342 XV	
3	B 1264 SS	B 1064 SS	

4	B 8887 LY	B 8287 LX	
5	B 8456 VC	K 8456 VC	
6	B 2123 VC	L 2124 VC	

Dari Tabel 4.4 tersebut, dapat dilihat bahwa kesalahan yang terdapat pada hampir keseluruhan plat nomor yang memiliki salah satu atau beberapa karakternya tidak terdeteksi dengan benar disebabkan oleh karakteristik *skeleton* yang dihasilkan dari proses skeletonisasi tidak memiliki kemiripan dengan *skeleton- skeleton* yang terdapat dalam basis data sehingga *output* hasil jaringan saraf tiruan harus mengalami proses pembulatan dan dapat menyebabkan terjadinya kesalahan terhadap hasil identifikasi karakter yang dilakukan dan terbaca sebagai karakter lain. Hal itu juga diperparah oleh hasil *skeleton* yang memiliki intensitas atau ketebalan yang tidak merata sehingga tampak ada bagian yang terang dan gelap di sepanjang hasil *skeleton* yang ada. Faktor ini dapat disebabkan oleh cacat yang ada pada plat seperti adanya bagian cat yang terkelupas atau terdapatnya baut yang letaknya berdekatan atau bahkan menempel pada suatu karakter yang ada pada plat nomor seperti yang terdapat pada karakter angka 8 kedua pada plat nomor B 8887 LY, seperti ditunjukkan Gambar 4.2 berikut:



Gambar 4.2. Plat nomor dengan kerusakan karakter akibat baut

BAB V

KESIMPULAN

1. Pada pengenalan nomor mobil dengan skeletonisasi menggunakan jaringan syaraf tiruan, tingkat akurasi terbaik yaitu sebesar 80% didapatkan pada uji coba dengan jumlah *training* sebanyak 35x, dan ukuran *image* sebesar 30x24 piksel.
2. Peningkatan persen akurasi terjadi dengan menambah jumlah *training* dan mengurangi ukuran *image skeleton* yang digunakan dalam proses training dan pengenalan.
3. Proses pembulatan terhadap *output* jaringan syaraf tiruan dapat memungkinkan terjadinya kesalahan identifikasi karakter yang dilakukan.
4. Plat nomor dengan kondisi yang kurang baik seperti pencahayaan yang tidak merata, cat plat yang terkelupas, ataupun adanya baut yang mengenai suatu karakter akan menyebabkan plat nomor tersebut tidak dapat dikenali dengan baik oleh sistem.
5. Bentuk *skeleton* dari suatu jenis karakter yang sama cukup bervariasi sehingga dapat menimbulkan perbedaan antara *skeleton* yang satu dengan yang lainnya.

DAFTAR ACUAN

- [1] Nelson, Robby, “*Sistem Pengenalan Plat Mobil Menggunakan Hidden Markov Model*”, Universitas Indonesia, 2007
- [2] Marasi Manurung, Patardo, “*Pengenalan Plat Mobil Menggunakan Jaringan Saraf Tiruan Metode Kohonen*”, Universitas Indonesia, 2007
- [3] Sugiarto, Ferry, ”*Pengenalan Plat Nomor Mobil Dengan Skeletonisasi Dan Menggunakan Metode Pelatihan Hidden Markov Model*”, Universitas Indonesia, 2007
- [4] Gonzalez, Rafael, Woods, Richard, “*Digital Image Processing*”, Addison Wesley Publishing co., USA, 1992
- [5] Matlab help, image processing toolbox, mathworks inc.
- [6] Fausett, Laurence., “*Fundamental of Neural Network (Architectures, Algorithms, and Application)* “, New Jersey : Prentice-Hall, 1994

DAFTAR PUSTAKA

1. Dayhoff, Judith E. *Neural Network Architectures (An Introduction)*. USA: Van Nostrand Reinhold. 1990.
2. Gonzalez, Rafael, Woods, Richard, “*Digital Image Processing*”, Addison Wesley Publishing co., USA, 1992.
3. Haykin, Simon. *Neural Networks (A Comprehensive Foundation)*. New York: Macmillan College Publishing Company.
4. Matlab help, image processing toolbox, mathworks inc.
5. Matlab help, neural network toolbox, mathworks inc.
6. Rich, Elaine dan Kevin Knight, *Artificial Intelligence*. New York: McGraw-hill Inc. 1991.
7. S. Kusumadewi, “Membangun Jaringan Syaraf Tiruan” , Penerbit Graha Ilmu, Yogyakarta, 2004.