PENGENALAN NOMOR PLAT KENDARAAN MENGGUNAKAN JARINGAN SYARAF TIRUAN (JST) *BACKPROPAGATION*

SKRIPSI

Oleh:

LIA TRIWAHYUNI

(04109603030-96)

PROGRAM STUDI ILMU KOMPUTER

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS BRAWIJAYA

MALANG

2011

PENGENALAN NOMOR PLAT KENDARAAN MENGGUNAKAN JARINGAN SYARAF TIRUAN (JST) *BACKPROPAGATION*

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

Oleh:
LIA TRIWAHYUNI
(0410963030-96)

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG

2011

LEMBAR PENGESAHAN SKRIPSI

PENGENALAN NOMOR PLAT KENDARAAN MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN (JST) *BACKPROPAGATION*

Oleh:
Lia Triwahyuni
0410963030-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 16 Agustus 2011

dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,	Pembimbing II,
Drs. Marji, MT	Bayu Rahayudi, ST., MT
NIP. 196708011992031001	NIP. 197407122006041001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc
NIP.196709071992031001


UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	:	Lia Triwahyuni
NIM	:	0410963030-96
Jurusan	:	Matematika
Program Studi	:	Ilmu Komputer
Penulis Tugas Akhir berjudul	:	Pengenalan Nomor Plat Kendaraan Menggunakan Metode Jaringan Syaraf Tiruan (JST) *Backpropagation*

Dengan ini menyatakan bahwa :

1. **Isi dari tugas akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.**
2. **Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.**

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 16 Agustus 2011
Yang menyatakan,

Lia Triwahyuni
NIM. 0410963030-96

PENGENALAN NOMOR PLAT KENDARAAN MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN (JST) *BACKPROPAGATION*

ABSTRAK

Teknik pengenalan huruf terus dikembangkan dalam berbagai metode pengenalan. Salah satunya dapat dimanfaatkan untuk pengenalan angka pada plat nomor. Dewasa ini banyak penelitian tentang pengenalan plat nomor dengan menggunakan berbagai metode. Pada umumnya metode yang digunakan dalam pengenalan huruf dalam hal ini pengenalan plat nomor kendaraan yaitu menggunakan metode pencocokan citra dan pendekatan statistika. (Kusumoputro, dkk. 1999).

Untuk mendapatkan struktur JST yang terbaik, dilakukan pelatihan dengan beberapa parameter diantaranya jumlah neuron pada *hidden layer* dan *learning rate*. Untuk mengetahui keberhasilan sistem pengenalan nomor plat ini yaitu dengan cara mengitung jarak *error* terdekat dan tingkat keakuratan dalam mengenali nomor plat. Dari penelitian ini diperoleh struktur JST dengan jumlah *neuron* pada *input layer* sebanyak 540 unit, jumlah *neuron* pada *hidden layer* sebanyak 100 unit, jumlah neuron pada *output layer* sebanyak 15 unit, nilai *learning rate* sebesar 0,9 dan maksimal *epoch* sebesar 10000 dengan nilai MSE terkecil sebesar 0,000975482.

JST yang terbentuk mampu mengenali karakter nomor plat dengan nilai keakuratan rata-rata untuk data yang sudah pernah dilakukan pelatihan sebesar 100% dan rata-rata keakuratan hasil pengenalan untuk data yang belum pernah dilakukan pembelajaran sebesar 52%.

VEHICLE PLATE IDENTIFICATION USING BACKPROPAGATION ARTIFICIAL NEURAL NETWORK (ANN)

ABSTRACT

Character identification techniques were continually developed in various identification methods. One of them could be exploited to identify numbers on licence plates. A great number of studies about licence numbers with different kinds of methods were executed nowadays. In general methods used in character identification in this matter vehicle licence plates identification was image compatibility and statistical approach (Kusumoputro, dkk. 1999).

To obtain the best ANN structure, training was conducted with several parameters including the number of neurons in hidden layer and learning rate. To determine the success of this licence number identification system was by counting the nearest distance error and accuracy level in identifying license number. This study obtained an ANN structure with the number of neuron in input layer of 540 units, the number of neuron in hidden layer of 100 units, the number of neuron in output layer of 15 units, learning rate value of 0.9 and the maximum epoch of 10000 with the smallest MSE value of 0.000975482.

The developed ANN was able to identify licence number characters with average accuracy value of 100% for data that has done training and average identification result accuracy of 25% for data that has never done learning.

KATA PENGANTAR

Alhamdulillahi Robbil alamin, puji syukur penulis panjatkan kepada Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer.

Skripsi ini bertujuan untuk melakukan perancangan dan implementasi pengenalan nomor plat kendaraan menggunakan metode jaringan syaraf tiruan (JST) *backpropagation*. Dengan mengetahui hasil pengenalan dari metode ini, diharapkan dapat dijadikan sebagai sebuah masukan untuk membuat aplikasi pengenalan nomor plat dengan metode yang lebih baik.

Pada penyusunan skripsi ini, penulis ingin mengucapkan terima kasih kepada :

1. Drs. Marji MT., selaku pembimbing utama penulisan skripsi sekaligus Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
2. Bayu Rahayudi, ST. MT., selaku pembimbing pendamping dalam penulisan skripsi.
3. Kasyful Amron, ST., selaku pembimbing akademik.
4. Dr. Abdul Rouf Alghofari, M.Sc., selaku Ketua Jurusan Matematika, FMIPA Universitas Brawijaya.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
7. Mama, Apa, Kakak-kakak: Teh Ning dan Mas Tutug, A'Deni, Teh Evi, dan keluarga penulis terima kasih atas dukungan, do'a restu dan semangat yang tiada henti kepada penulis.
8. Sahabat Terbaik : Uwa Nita, Bi Lina, Iko-iko, Uwie, Anne, Dzois, terima kasih atas do'a dan semangat untuk penulis.
9. Sahabat LoeLie : Oelpha dan Litha,Nudy, Yayo, Rizka, Ika, serta sahabat ilkomer's 2004. Terima kasih atas do'a dan dukungan.
10. A'Adam yang senantiasa memberikan dukungan dan do'a.
11. Saudara Putra dan Rifa'i yang telah membantu atas terselesaikannya skripsi ini.

12. Rekan-rekan di Program Studi Ilmu Komputer FMIPA Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan skripsi ini.
13. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat penulis sebutkan satu per satu.

Penulis sadari bahwa masih banyak kekurangan dalam laporan ini, oleh karena itu penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi skripsi ini untuk kelanjutan penelitian serupa di masa mendatang.

Semoga laporan skripsi ini dapat bermanfaat. Amin.

Malang, 16 Agustus 2011

Penulis

DAFTAR ISI

Halaman Judul i
Halaman Pengesahan iii
Lembar Pernyataan v
Abstrak..... vii
Abstract..... ix
Kata Pengantar xi
Daftar Isi xiii
Daftar Gambar xv
Daftar Tabel..... xvii

BAB I PENDAHULUAN

1.1 Latar Belakang..... 1
1.2 Rumusan Masalah..... 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian..... 3
1.6 Metodologi 3
1.7 Sistematika Penulisan..... 4

BAB II TINJAUAN PUSTAKA

2.1 Citra 5
2.1.1 Pengolahan Citra 5
2.1.1.1 *Edge Detection* 6
2.1.1.2 *Thresholding* 7
2.1.1.3 Segmentasi Citra..... 7
2.2 Plat Nomor Kendaraan Bermotor..... 7
2.3 Jaringan Syaraf Tiruan..... 8
2.3.1 Definisi Jaringan Syaraf Tiruan 8
2.3.2 Proses Pembelajaran 9
2.3.3 Arsitektur Jaringan Syaraf Tiruan 10
2.3.4 Algoritma *Backpropagation* 12
2.3.5 Inisialisasi Bobot Awal Secara Random 16
2.3.6 Inisialisasi Bobot Awal dengan Nguyen-Widrow 17
2.3.7 Jumlah Unit Tersembunyi..... 17
2.3.8 Fungsi Aktivasi..... 18

BAB III METODOLOGI DAN PERANCANGAN SISTEM

3.1 Deskripsi Umum Sistem 19
3.2 Batasan Sistem 19
3.3 Data Uji 19
3.4 Perancangan Sistem 20
3.4.1 Pengolahan Citra 21
3.4.2 Proses Jaringan Syaraf Tiruan 22
3.4.2.1 Struktur Jaringan Syaraf Tiruan 22
3.4.2.2 Proses Jaringan Syaraf Tiruan 23
3.4.2.3 Proses Pelatihan 24
3.4.2.4 Inisialisasi Bobot dan Bias Awal 26
3.4.2.5 *Feedforward* 27
3.4.2.6 *Backpropagation* 28
3.4.2.7 Perbaharui Bobot dan Bias 30
3.4.2.8 Proses Pengenalan 32
3.5 Contoh Perhitungan Manual 34
3.5.1 Pelatihan Menggunakan *Backpropagation* 35
3.5.2 Pengujian Menggunakan *Backpropagation* 46
3.6 Rancangan Uji Coba 48
3.6.1 Rancangan Uji Coba Terhadap *Neuron Hidden* 48
3.6.2 Rancangan Uji Coba Terhadap *Learning Rate* 48
3.6.3 Rancangan Pengujian Sistem 49
3.6.3.1 Rancangan Uji Segmentasi 49
3.6.3.2 Rancangan Uji Pengenalan 49
3.7 Evaluasi 49
3.8 Tampilan Antarmuka 50
3.8.1 *Form* Pelatihan 50
3.8.2 *Form* Pengenalan 51

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi 53
4.1.1 Lingkungan Perangkat Keras 53
4.1.2 Lingkungan Perangkat Lunak 53
4.2 Implementasi Program 53
4.2.1 Struktur Data 54
4.2.2 Inisialisasi Bobot dan Bias Awal 56
4.2.3 Fungsi Aktivasi 57
4.2.4 Turunan Fungsi Aktivasi 57

4.2.5 *Feedforward* 57
4.2.7 Hitung *Error* 58
4.2.8 *Backpropagation* 59
4.2.9 Proses Perbaharui Bobot dan Bias 60
4.2.10 Proses Peengujian 60
4.3 Implementasi Antarmuka 61
4.4 Hasil dan Pembahasan 65
4.4.1 Hasil Percobaan 65
4.4.1.1 Pengaruh Jumlah *Hidden Layer* 65
4.4.1.2 Pengaruh Nilai *Learning Rate* 67
4.5 Hasil Pengujian 68
4.5.1 Hasil Pengujian Segmentasi Karakter 68
4.5.2 Hasil Pengujian Data Latih 68
4.5.3 Hasil Pengenalan 69
4.6 Analisa Hasil 70
4.6.1 Analisa Hasil Segmentasi 70
4.6.2 Analisa Hasil Pengenalan 71

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan 73
5.2 Saran 73

DAFTAR PUSTAKA 75
LAMPIRAN 77

DAFTAR GAMBAR

Gambar 2.1 *Single-Layer Network* 9
Gambar 2.2 *Multi-Layer Network*..... 9
Gambar 2.3 *Reccurrent Network*..... 10
Gambar 3.1 Contoh data untuk pelatihan 17
Gambar 3.2 Contoh data untuk pengujian 17
Gambar 3.3 Diagram Alir Sistem 19
Gambar 3.4 *Flowchart* Proses Pengolahan Citra 20
Gambar 3.5 Arsitektur Jaringan Syaraf Tiruan 21
Gambar 3.6 *Flowchart* Proses JST 22
Gambar 3.7 Diagram Alir Proses Pelatihan 23
Gambar 3.8 *Flowchart* Inisialisasi Bobot dan Bias Awal 25
Gambar 3.9 *Flowchart* Proses *Feedforward*..... 26
Gambar 3.10 *Flowchart* Proses *Backpropagation*..... 28
Gambar 3.11 *Flowchart* Proses *Update* Bobot dan Bias 29
Gambar 3.12 *Flowchart* Proses Pengenalan 31
Gambar 3.13 Pola Karakter..... 32
Gambar 3.14 Representasi Pola Karakter..... 32
Gambar 3.15 *Form* Pelatihan 48
Gambar 3.16 *Form* Pengenalan..... 49
Gambar 4.1 Struktur Data *Backpropagation* 52
Gambar 4.2 Prosedur Inisialisasi Bobot 55
Gambar 4.3 *Source code* Fungsi Aktivasi 55
Gambar 4.4 *Source code* Turunan Fungsi Aktivasi 55
Gambar 4.5 Prosedur *Feedforward*..... 56
Gambar 4.6 Prosedur Hitung *Error* 57
Gambar 4.7 Prosedur *Backward*..... 58
Gambar 4.8 Prosedur *Update* Bobot..... 58
Gambar 4.9 Prosedur Pengujian 59
Gambar 4.10 *Form* Utama..... 60
Gambar 4.11 *Form Input Database* Karakter 60
Gambar 4.12 *Form* Segmentasi Karakter..... 61
Gambar 4.13 *Form* Pelatihan 62
Gambar 4.14 *Form* Pengenalan..... 63
Gambar 4.15 Grafik Pengaruh *Hidden Layer* 64
Gambar 4.16 Grafik Pengaruh *Learning Rate*..... 65
Gambar 4.17 Contoh Kesalahan Segmentasi..... 68
Gambar 4.18 Contoh Kesalahan Pengenalan..... 69

DAFTAR TABEL

Tabel 2.1 Contoh input dan target pembelajaran terawasi 7
Tabel 3.1 Data Pertama 32
Tabel 3.2 Target untuk Data Pertama 33
Tabel 3.3 Data Kedua 33
Tabel 3.4 Target untuk Data Kedua 33
Tabel 3.5 Bobot dari *Input* ke *hidden layer* 33
Tabel 3.6 Inisialisasi NIlai Vij 34
Tabel 3.7 Nilai Vij (baru) 35
Tabel 3.8 Nilai Bias (V0j) 36
Tabel 3.9 Nilai Bobot ke*output layer* (Wjk) 36
Tabel 3.10 Nilai Bias (W0k) 36
Tabel 3.11 Hasil operasi pada *hidden* (Z_in) 37
Tabel 3.12 Hasil Aktivasi Z_in 37
Tabel 3.13 Hasil operasi pada *output* (Y_in) 38
Tabel 3.14 Hasil Aktivasi Y_in 38
Tabel 3.15 Nilai Kemelesetan 38
Tabel 3.16 Nilai Perubahan Bobot (ΔWjk) 39
Tabel 3.17 Perbaharui Bobot Bias (ΔW0k) 39
Tabel 3.18 Faktor penimbang di unit *hidden* 39
Tabel 3.19 Aktivasi faktor kesalahan di unit *Hidden* 40
Tabel 3.20 Nilai perubahan bobot (ΔVij)..... 40
Tabel 3.21 Hasil Nilai Koreksi bobot bias (ΔV0j)..... 41
Tabel 3.22 Nilai bobot baru (Vij)..... 41
Tabel 3.23 Nilai V0j baru..... 42
Tabel 3.24 Perubahan Bobot baru Wjk (baru) 42
Tabel 3.25 Nilai Bias W0k (baru) 42
Tabel 3.26 Nilai Bobot baru di unit *hidden*..... 43
Tabel 3.27 Nilai bias *input* ke *hidden layer* baru..... 43
Tabel 3.28 Perubahan Bobot Wjk (baru)..... 43
Tabel 3.29 Nilai Bias W0k (baru)..... 44
Tabel 3.30 Operasi pada *Hidden*..... 44
Tabel 3.31 Hasil Aktivasi Operasi pada *Hidden*..... 44
Tabel 3.32 Operasi pada *output Layer* (Y_in_k) 45

Tabel 3.33 Hasil Aktivasi Y_in_k 45
Tabel 3.34 Pengaruh Jumlah *Neuron Hidden* 46
Tabel 3.35 Pengaruh Jumlah *Learning Rate* 46
Tabel 3.36 Hasil Uji Segmentasi 47
Tabel 3.36 Hasil Uji Pengenalan 47
Tabel 4.1 Hasil Percobaan pengaruh *hidden layer* 64
Tabel 4.2 Hasil Percobaan pengaruh *Learning rate* 65
Tabel 4.3 Hasil Pengujian Segmentasi Karakter 66
Tabel 4.4 Hasil pengujian citra Latih 67
Tabel 4.5 Hasil pengenalan 67

BAB I
PENDAHULUAN

1.1. Latar Belakang

Dengan meningkatnya teknologi komputer, maka semakin meningkat pula tuntutan akan aplikasi komputer yang mampu mendukung kinerja manusia. Salah satunya adalah kebutuhan akan aplikasi yang mampu mengenali nomor plat kendaraan untuk sistem kontrol dan informasi pada area parkir karena sekarang ini sistem pada area parkir dilakukan secara manual. Maka dari itu, dibutuhkan suatu aplikasi yang memudahkan pengontrolan sistem keamanan parkir dengan memanfaatkan teknologi komputer. Dalam aplikasi pengenalan nomor plat kendaraan ini, kamera digunakan sebagai sensor yang menangkap gambar dari plat nomor kendaraan. Kemudian gambar tersebut selanjutnya akan diolah dengan menggunakan *image processing* (pengolahan citra).

Dengan adanya teknologi pengolahan citra, maka data berupa gambar yang mengandung gambar suatu karakter, dapat diambil informasinya dan dikonversikan ke dalam bentuk teks. *P*roses pengenalan karakter menggunakan metode jaringan syaraf tiruan (JST) *Backpropagation*. Dengan menggunakan metode ini diharapkan dapat melakukan pengenalan karakter nomor plat dengan baik. JST merupakan sebuah sistem pembelajaran terhadap penerimaan informasi yang memiliki kinerja layaknya sebuah jaringan syaraf pada manusia. JST diimplementasikan dengan menggunakan program komputer sehingga mampu menyelesaikan sejumlah proses perhitungan untuk pengenalan pola. Sistem pengenalan pola merupakan komponen penting dalam proses peniruan cara kerja sistem manusia

Telah banyak dilakukan penelitian mengenai pengenalan nomor plat kendaraan dengan berbagai macam metode dan pendekatan. Banyak teknik yang digunakan untuk melakukan pengenalan nomor plat, misalnya penggabungan dari teknologi logika fuzzy dan jaringan syaraf (M. Syamsa Ardisasmita, 2000), dan menggunakan metode *mean shift spectral clustering* (Afwan Badru Naim, 2009).

Dalam aplikasi pengenalan nomor plat kendaraan, terdapat dua buah proses utama, yaitu segmentasi plat nomor dan pengenalan hasil segmentasi. Kesulitan yang dihadapi dalam proses pengenalan

plat nomor kendaraan di Indonesia adalah terdapat warna latar belakang plat nomor yang berbeda. Oleh karena itu, dalam penelitian ini dilakukan pembatasan, yaitu hanya pada plat nomor kendaraan pribadi dengan warna dasar hitam dan tulisan nomor plat berwarna putih.

Berdasarkan latar belakang yang telah dikemukakan maka Skripsi ini diberi judul **"Pengenalan Nomor Plat Kendaraan Menggunakan Metode Jaringan Syaraf Tiruan (JST) *Backpropagation* ".**

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang dalam penyusunan skripsi ini adalah :

1. Bagaimana arsitektur Jaringan Syaraf Tiruan (JST) *backpropagation* terbaik dalam mengenali nomor plat kendaraan.
2. Bagaimana menganalisis hasil pengujian segmentasi dan tingkat keakuratan pada pengenalan nomor plat kendaraan menggunakan Jaringan Syaraf Tiruan (JST) *backpropagation*.

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, maka batasan masalah pada skripsi ini adalah :

1. Mengingat bervariasinya ukuran, letak, dan jenis huruf pada plat nomor, maka kasus ini dibatasi hanya pengenalan plat nomor kendaraan pribadi dengan warna dasar hitam dan tulisan nomor plat warna putih.
2. Pengambilan gambar pada posisi sejajar antara kamera dan plat nomor.
3. Jarak pengambilan gambar $\pm$ 100-200 cm dari kamera.
4. Citra masukan menggunakan citra dengan ekstensi *Bitmap* (.bmp).
5. Citra masukan berupa citra plat nomor yang telah di-*cropping* secara manual dari citra kendaraan.
6. Pengenalan dibatasi hanya mengenali nomor plat saja tidak termasuk bulan dan tahun masa berlaku kendaraan.

1.4 Tujuan

Berdasarkan rumusan masalah yang telah dikemukakan, maka tujuan dari skripsi ini adalah :

1. Mengimplementasikan metode Jaringan Syaraf Tiruan *backpropagation* untuk pengenalan nomor plat.
2. Menganalisis hasil uji penggunaan Jaringan Syaraf Tiruan *backpropagation* untuk mengenali nomor plat.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini antara lain :

1. Tersedianya perangkat lunak untuk melakukan pengenalan nomor plat kendaraan dengan metode JST *backpropagation*.
2. Tersedianya referensi sebagai acuan pengembangan untuk pengenalan nomor plat kendaraan
3. Tersedianya informasi tentang keakuratan pengenalan pola menggunakan metode JST *backpropagation*.
4. Tersdianya perangkat lunak yang memudahkan pengontrolan sistem keamanan parkir

1.6 Metodologi

Metodologi yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut :

1. Studi literatur
   Studi ini dilakukan dengan cara mencari sekaligus mempelajari beberapa literatur dan artikel mengenai pengenalan plat nomor dan Jaringan Syaraf Tiruan *backpropagation*.
2. Pendefinisian dan Analisis Masalah
   Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan implementasi perangkat lunak
   Membuat perancangan model perangkat lunak dan mengimplementasikan hasil rancangan tersebut yaitu membuat perangkat lunak untuk pengenalan nomor plat menggunakan metode Jaringan Syaraf Tiruan *backpropagation*.
4. Uji coba dan analisa hasil implementasi
   Menguji perangkat lunak, kemudian menganalisa hasil dari implementasi pengenalan plat nomor sesuai dengan tujuan yang

dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Pembuatan Skripsi ini dilakukan dengan pembagian bab sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan skripsi.

BAB II : TINJAUAN PUSTAKA

Bab ini mencantumkan beberapa tinjauan pustaka yang berkaitan dengan penelitian ini, diantaranya : konsep dasar dari citra, serta beberapa macam representasinya, pengertian jaringan syaraf dan jaringan syaraf tiruan, metode belajar pada jaringan syaraf tiruan, dan algoritma propagasi balik (*backpropagation*) pada jaringan syaraf tiruan.

BAB III : METODOLOGI DAN PERANCANGAN

Bab ini menjelaskan mengenai metode dan perancangan yang akan dilakukan dalam membangun tahapan-tahapan dalam proses pengenalan plat nomor kendaraan.

BAB IV : IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa sistem aplikasi yang dibangun, yaitu apakah hasil uji coba menunjukkan tingkat keberhasilan yang memuaskan dan apakah sistem ini cukup prospektif untuk digunakan sebagai salah satu sistem kontrol dan sekuriti pada area parkir.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan, dan saran untuk pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II
TINJAUAN PUSTAKA

2.1 Citra

Citra terdiri dari 2 macam: citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog, seperti mata. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinu. Beberapa sistem optik dilengkapi dengan fungsi digitalisasi sehingga mampu menghasilkan citra diskrit, misalnya kamera digital. Citra diskrit inilah yang disebut citra digital.Citra yang merupakan gambar pada bidang dua dimensi, secara matematis, adalah fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Citra digital biasanya berbentuk persegi panjang dengan dimensi ukuran yang dinyatakan sebagai panjang (*M*) x lebar (*N*) dengan derajat keabuan f(x,y), dapat dinyatakan dalam matriks yang ditunjukkan pada Persamaan 2.1. (Iswanto,Noerdityo, dkk.2010)

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,M-1) \\ f(1,0) & f(1,1) & f(1,M-1) \\ \vdots & & \\ f(N-1,0) & f(N-1,1) & f(N-1,M-1) \end{matrix} \quad (2.1)$$

2.1.1Pengolahan Citra

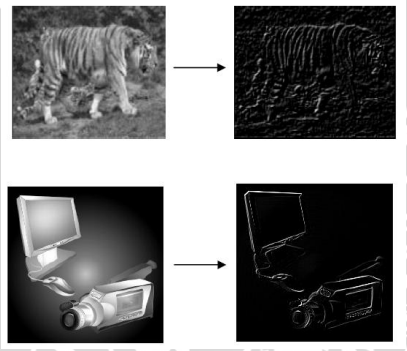
Pengolahan citra merupakan suatu metode atau teknik pemrograman yang digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapat informasi tertentu. Dalam prosesnya terdapat dua buah cara untuk melakukan proses sampling dan kuantisasi, yaitu secara ruang 2 dimensi (*spatial quant*) dan secara tingkat keabuan (*grayscale*). Pada kuantitas secara 2 dimensi digunakan untuk menentukan resolusi citra hasil dan kuantisasi secara tingkat keabuan digunakan untuk menentukan resolusi intensitas citra yang dihasilkan. Hasil akhir dimodelkan dengan matriks N x M dimana elemen-elemen dari matriks ini adalah perwakilan dari beberapa pixel pada gambar (*sampling*) dengan nilai bit biner (0 atau 1).
(Balza, Achmad dan Firdausy Kartika, 2005)

2.1.1.1 *Edge Detection*

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah untuk menandai bagian yang menjadi detail citra dan untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra.

Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

Perhatikan hasil deteksi dari beberapa citra menggunakan model differensial pada gambar 2.1.



Gambar 2.2 Hasil Deteksi Tepi

Pada gambar 2.2. terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi.

Macam-macam metode untuk proses deteksi tepi ini, antara lain:

1. Metode Robert
2. Metode Prewitt
3. Metode Sobel

Metode yang banyak digunakan untuk proses deteksi tepi adalah metode Robert, Prewitt dan Sobel.(Gonzales, Rafael C and Wintz, Paul.1987)

2.1.1.2 *Thresholding*

Thresholding digunakan untuk mengubah gambar berwarna menjadi gambar hitam putih (biner) dengan tingkat kontras yang tinggi. Semua piksel yang lebih terang dari *threshold* akan diubah menjadi putih. Sebaliknya, semua piksel yang lebih gelap akan diubah menjadi hitam. Proses *threshold* sangat berguna untuk menentukan daerah yang terterang dan tergelap dari sebuah citra. (Resmana Liem,dkk, 2003).

2.1.1.3 Segmentasi Citra

Segmentasi gambar adalah pemisahan objek yang satu dengan objek yang lain dalam suatu gambar. Ada 2 macam segmentasi, yaitu *full segmentation* dan *partial segmentation*. *Full segmentation* adalah pemisahan suatu object secara individu dari *background* dan diberi ID (label) pada tiap-tiap segmen. *Partial segmentation* adalah pemisahan sejumlah data dari *background* dimana data yang disimpan hanya data yang dipisahkan saja untuk mempercepat proses selanjutnya.

Ada 3 tipe dari segmentasi yaitu:

1. *classification-based*: segmentasi berdasarkan kesamaan suatu ukuran dari nilai *pixel*. Salah satu cara paling mudah adalah *thresholding*. *Thresholding* ada 2 macam yaitu global dan lokal. Pada *thresholding* global, segmentasi berdasarkan pada sejenis histogram. Pada *thresholding* lokal, segmentasi dilakukan berdasarkan posisi pada gambar, gambar dibagi menjadi bagianbagian yang saling melengkapi, jadi sifatnya dinamis.
2. *edge-based*: mencari garis yang ada pada gambar dan garis tersebut digunakan sebagai pembatas dari tiap segmen.
3. *region-based*: segmentasi dilakukan berdasarkan kumpulan *pixel* yang memiliki kesamaan (tekstur, warna atau tingkat warna abu-abu) dimulai dari suatu titik ke titik-titik lain yang ada disekitarnya. (Adipranata, Rudy, dkk.2005)

2.2 Plat Nomor Kendaraan Bermotor

Bahan baku Tanda nomor Kendaraan Bermotor (TNKB) resmi adalah aluminium dengan ketebalan 1 mm [1]. Ukuran TNKB untuk kendaraan bermotor roda 2 dan roda 3 adalah 250x105 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah

395x135 mm. Baris pertama pada TNKB menunjukkan kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf) sedangkan baris kedua menunjukkan bulan dan tahun masa berlaku. Terdapat cetakan garis lurus pembatas lebar 5 mm diantara ruang nomor polisi dengan ruang angka masa berlaku. (Liliana dan Gregorius Satia Budhi, Hendra.2010)

Warna tanda nomor kendaraan bermotor resmi ditetapkan sebagai berikut:

1. Kendaraan bermotor bukan umum dan kendaraan bermotor sewa dengan warna dasar hitam dengan tulisan berwarna putih.
2. Kendaraan bermotor umum dengan warna dasar kuning dengan tulisan berwarna hitam.
3. Kendaraan bermotor milik pemerintah dengan warna dasar merah dengan tulisan berwarna putih..
4. Kendaraan bermotor Corps Diplomatik Negara Asing dengan warna dasar putih dengan tulisan berwarna hitam
5. Kendaraan bermotor untuk transportasi dealer (pengiriman dari perakitan ke dealer, atau dealer ke dealer) dengan warna dasar putih dengan tulisan berwarna merah.

2.3 Jaringan Syaraf Tiruan

2.3.1 Definisi Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (*artificial neural network*) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologi di dalam otak. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi nonlinier, klasifikasi data, cluster, dan regresi non parametik atau sebagai sebuah simulasi dari koleksi model syaraf biologi.

Model syaraf ditunjukkan dengan kemampuannya dalam emulsi, analisa, prediksi dan asosiasi. Berdasarkan kemampuan yang dimiliki, JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh, untuk menghasilkan output yang sempurna dari contoh atau input yang dimasukkan dan membuat prediksi tentang kemungkinan output yang akan muncul atau menyimpan karakteristik dari input yang disimpan kepadanya (Kristanto, Andri.2004).

2.3.2 Proses Pembelajaran

Karakteristik dari JST adalah kemampuannya untuk belajar. Seperti halnya manusia untuk mengenali suatu benda, sistem juga perlu diberi pembelajaran. Namun kemampuan belajar JST bersifat terbatas, sehingga JST tidak dapat melakukan segalanya. Contohnya manusia dapat mempelajari sesuatu, menghafalnya, dan menalarkan. Berbeda dengan JST, yang dilakukan tidak sampai pada penalaran. Sehingga dalam pembelajarannya dibutuhkan pengawasan atas kebenaran yang diajarkan. Kebenaran pengetahuan yang disimpan dalam sistem ditentukan oleh pengajarnya (manusia). (Kusumadewi, 2003).

Pembelajaran JST merupakan proses pencarian konfigurasi bobot yang sesuai dengan cara perkalian, penjumlahan dan aktivasi bobot dan input. Menurut Kosko, 1990 JST dikelompokkan menjadi 2 bagian. Pertama bagaimana JST menyimpan pengetahuan atau *encode*, yang kedua bagaimana JST menanggapi dan memproses data yang masuk atau *decode*. Berdasarkan *encode* proses pembelajaran JST dibagi atas metode pembelajaran dibimbing atau *Supervised* dan tidak dibimbing atau *Unsupervised*. Sedangkan berdasarkan *decode* dibedakan menjadi umpan maju atau *feedforward* dan umpan balik atau *feedback*.

a. Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Contoh : andaikan kita memiliki jaringan syaraf yang akan digunakan untuk mengenali pasangan pola, misalkan pada operasi AND. *Input* dan targetnya terlihat pada Tabel 2.1.

Tabel 2.1 Contoh *input* dan target pembelajaran terawasi

Input		Target
0	0	0
0	1	0
1	0	0
1	1	1

Pada proses pembelajaran, suatu pola *input* akan diberikan ke satu neuron pada lapisan *input*. Pola ini akan dirambatkan di sepanjang

jaringan syaraf hingga sampai ke neuron pada lapisan *output*. Lapisan *output* ini akan membangkitkan pola *output* yang nantinya akan dicocokkan dengan pola *output* targetnya. Apabila terjadi perbedaan antara pola *output* hasil pembelajaran dengan pola target, maka disini akan terjadi *error*. Apabila nilai *error* ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi. **(**Kristanto,Andri. 2004**)**

b. Pembelajaran tak terawasi (*unsupervised learning*)

Pada metode pembelajaran yang tak terawasi ini tidak memerlukan target *output*. Pada metode ini tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola. (Kristanto,Andri. 2004)

2.3.3 Arsitektur Jaringan Syaraf Tiruan

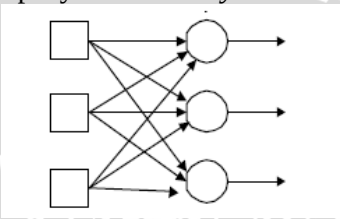
Suatu jaringan syaraf minimum tersusun atas *input layer* dan *output layer*. Dalam beberapa tipe jaringan diantara *input layer* dan *output layer* terdapat *hidden layer*. *Input layer* merupakan aktifitas unit-unit lapisan masukan yang menunjukkan informasi dasar yang kemudian digunakan dalam jaringan syaraf tiruan. *Hidden layer* merupakan aktifitas setiap unit-unit lapisan tersembunyi ditentukan oleh aktifitas dari unit-unit masukan dan bobot dari koneksi antara unit-unit masukan dan unit-unit lapisan tersembunyi. Sedangkan *output layer* merupakan karakteristik dari unit-unit keluaran tergantung dari aktifitas unit-unit lapisan tersembunyi dan bobot antara unit-unit lapisan tersembunyi dan unit-unit keluaran.

Hal ini berarti bahwa semua neuron pada *input layer* akan berhubungan ke semua neuron dalam *hidden layer* yang selanjutnya setiap *neuron* dalam *hidden layer* nantinya akan dihubungkan ke semua *neuron* di *output layer*. Pada setiap *layer* biasanya *neuron* mempunyai fungsi aktivasi serta pola hubungan ke *neuron* lain yang sama.

Jaringan syaraf tiruan dapat dibagi menjadi beberapa kelompok. Menurut Siang J.J. (2005) berdasarkan jumlah *layernya* jaringan syaraf tiruan dibagi menjadi:

a. *Single-Layer Network*

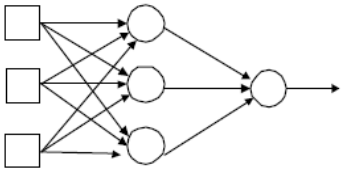
Single layer network ini pertama kali dirancang oleh Widrow dan Holf pada tahun 1960. *Neural network* jenis ini memiliki koneksi pada *inputnya* secara langsung ke jaringan *outputnya*. Pada jaringan ini, semua unit input dihubungkan langsung ke unit *output*, meskipun dengan bobot yang berbeda-beda. *Single layer network* ini tidak mempunyai *hidden layer*.



Gambar 2.4 *Single layer network*

b. *Multi layer network*

Jaringan ini merupakan perluasan dari *single layer network*. Pada jaringan ini selain terdapat unit *input* dan *output* juga terdapat unit lain yang biasanya disebut dengan *hidden layer*.

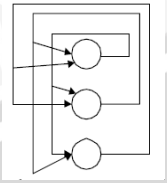


Gambar 2.5 *Multi layer network*

Multi layer network ini dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan *single layer network*, meskipun kadang proses pelatihan lebih kompleks dan lama.

c. *Reccurent network*

Model jaringan ini sama seperti *single layer network* maupun *multi layer network*. Hanya saja ada *neuron output* yang memberikan sinyal pada unit *input* (*feedback loop*).



Gambar 2.6 *Reccurent network*

Algoritma pelatihan *Backpropagation* atau propagasi balik pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhat dan McClelland untuk dipakai pada JST, dan selanjutnya algoritma ini biasa disingkat dengan BP. Algoritma ini termasuk metode pelatihan *supervised* dan didesign untuk operasi pada jaringan *feedforward* multi lapis.

Metode BP ini banyak diaplikasikan secara luas. Bahkan, sekitar 90% lebih BP telah berhasil diaplikasikan di berbagai bidang, diantaranya diterapkan di bidang financial, pengenalan pola tulisan tangan, pengenalan pola suara, sistem kendali, pengolahan citra medika dan masih banyak lagi keberhasilan BP sebagai salah satu metode komputasi yang handal.

Pelatihan jaringan menggunakan *backpropagation* dibagi menjadi tiga tahap, yaitu *feedforward* dari input pola latihan, dan *backpropagation* dari *error* yang berhubungan, dan pengaturan bobot berdasarkan bobot sebelumnya. Setelah dilatih, aplikasi jaringan yang terdapat pada tahap komputasi dari *feedforward*. Meskipun latihan dilakukan lambat, jaringan yang telah dilatih dapat mrnghasilkan outputnya dengan sangat cepat.

Struktur jaringan syaraf yang dipakai memiliki 75 neuron pada lapisan input, satu lapisantersembunyi dengan jumlah neuron, dan 10 neuron pada lapisan output. (Kristanto, Andri.2004)

2.3.4 Algoritma *Backpropagation*

Algoritma *Backpropagation* mempunyai 2 tahap utama, yaitu *learning* dan *testing*. Untuk *learning,* proses bekerja secara *forward*, dan kemudian *backward*. *Forward* maksudnya perhitungan dihitung mulai dari *layer* yang paling depan, yaitu *input layer,* ke belakang sampai *output layer*. Pada tahap *forward* ini dilakukan perhitungan *output* dari *network*. Setelah itu ada tahap *backward,* yang dilakukan dari *output layer* sampai ke *input layer* yang bertujuan meng- *update* bobot dalam setiap *network*.

Berikut adalah algoritma dari *Backpropagation Network* (algoritma untuk mentraining *network* sehingga menghasilkan *output* yang diinginkan):

1. Inisialisasi Bobot
2. Tentukan semua bobot pada setiap koneksi ke suatu bilangan *random* yang kecil.
3. Perhitungan aktivasi

Input : data pelatihan, jumlah neuron *input*, jumlah neuron *hidden layer*, *learning rate*, maksimum *epoch*, target *error*.
Output : model JST yang siap mengolah data baru.
Algoritma:

Langkah 0:

- Pemberian inisialisasi bobot (diberi nilai yang cukup kecil secara acak)

Langkah 1:

- Ulangi langkah 2 hingga 9 sampai kondisi akhir iterasi dipenuhi

Langkah 2:

- Untuk masing-masing pasangan data pelatihan (training data) lakukan langkah 3 hingga 8

Propagasi maju(*feedforward*)

Langkah 3:

- Masing-masing masukan X_i, $i = 1, \dots n$ menerima sinyal masukan X_i dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit-unit lapisan tersembunyi)

Langkah 4:

- Masing-masing unit dilapisan tersembunyi (Z_j, j= 1,2,3,…,n) menjumlahkan sinyal-sinyal input terbobot:

$$Z\_in_j = V_{0j} + \sum_{i=1}^{n} x_i v_{ij} \qquad (2.2)$$

Dimana:

- z_in_j = *Input* jaringan ke z_j
- v0j = Bias pada lapisan tersembunyi j
- x_i = Unit *input* i
- V_{ij} = nilai penimbang sambungan dari unit X_i ke unit Z_i

Kemudian gunakan fungsi aktifasi untuk menghitung sinyal *output*-nya dengan rumus:

$$Z_j = f(Z\_in_j) \qquad (2.3)$$

Dimana : Z_j = Unit ke-j pada lapisan tersembunyi

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

Langkah 5 :

- Masing-masing unit *output* (y_k, k=l,2,3...m) menjumlahkan sinyal-sinyal input terbobot :

$$Y\_in_k = W_{ok} + \sum_{j=1}^{p} Zj W_{jk} \qquad (2.4)$$

Dimana :

- Y_in_k = net masukan untuk unit Y_k
- W_{0k} = nilai penimbang sambungan pada bias untuk unit Y_k
- W_{jk} = nilai penimbang sambungan dari Z_{ij} ke unit Y_k

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya

$$y_k = f(y\_in_k) \qquad (2.5)$$

Dimana : Y_k = unit ke -k pada lapisan keluaran

BackPropagasi dan Galatnya
Langkah 6:

- Masing-masing unit keluaran (Y_k, k=l,2,3,...,m) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*-nya:

$$\delta_k = (t_k - y_k) f'(y\_in_k) \qquad (2.6)$$

Dimana: δ_k = faktor pengaturan nilai penimbang sambungan pada lapisan keluaran

sebagaimana *input training* data, *output training* data t_k juga telah diskalakan menurut fungsi aktivasi yang digunakan.

Faktor δ_k ini digunakan untuk menghitung koreksi *error* (δw_{jk}) yang nantinya akan dipakai untuk memperbaharui w_{jk}, dimana:

$$\Delta W_{jk} = \alpha.\partial_k.Z_j \quad (2.7)$$

Dimana : α = konstanta laju pelatihan (learning rate)

$$0<\alpha<1$$

selain itu juga dihitung koreksi bias ΔW_{0k} yang nantinya akan digunakan untuk memperbaharui W_{0k}, dimana:

$$\Delta W_{ok} = \alpha.\partial_k \quad (2.8)$$

Faktor δ_k ini kemudian dikirimkan ke layer yang berada pada unit selanjutnya.

Langkah 7 :

- Setiap *hidden* unit (Z_j ,j=1..,p) menjumlahkan input delta yang sudah terbobot.

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k W_{jk} \quad (2.9)$$

Dimana: δ_in_j = Keluaran unit δ_j

Selanjutnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi δ_j, dimana:

$$\delta_j = \delta\_in_j f'(y\_in_j) \quad (2.10)$$

Faktor δ_j ini digunakan untuk menghitung koreksi *error* (ΔV_{ij}) yang nantinya akan digunakan untuk memperbaiki V_{ij}, dimana:

$$\Delta V_{ij} = \alpha\, \delta_j X_i \quad (2.11)$$

Selain itu juga dihitung koreksi bias ΔV_{oj} yang nantinya akan digunakan untuk memperbaiki v_{0j}, dimana:

$$\Delta V_{oj} = \alpha\delta_j \quad (2.12)$$

Memperbaiki penimbang dan bias

Langkah 8:

- Masing-masing unit *output* (y_k, k=1,..,m) akan memperbaharui bias dan bobotnya dari setiap *hidden* unit (j=0,...,p),

$$W_{jk}\ (baru) = W_{jk}\ (lama) + \Delta W_{jk} \quad (2.13)$$

Dimana: ΔW_{kj} = selisih antara $W_{kj}(t)$ dengan$W_{kj}(t+1)$

Demikian pula untuk setiap *hidden* unit (Z_j, j: 1,2,3,...,p) akan memperbaharui bias dan bobotnya dari setiap unit *input* (j=0,1,2,...,n).

$$V_{ij}\ (baru) = V_{ij}\ (lama) + \Delta V_{ij} \quad (2.14)$$

Dimana: ΔV_{ij} = selisih antara $V_{0j}(t)$ dengan$V_{ij}(t+1)$

Langkah 9 :

Memeriksa *stoppping condition*.
Jika *stop condition* telah terpenuhi, maka pelatihan jaringan dapat dihentikan.

Cara memeriksa *stoppping condition* dengan *Mean Square Error* (MSE) adalah sebagai berikut:

- Dengan bobot yang telah didapat, lakukan langkah *feedforward* (langkah 3 sampai langkah 5).
- Kemudian dicari selisih antara target output (t_k) dengan output jaringan (y_k) dan diimplementasikan pada persamaan *Mean Square Error* (MSE). Jika terdapat m training, maka

$$MSE = E = 0.5 \text{ x } (t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \ldots + (t_{km} - y_{km})^2 \quad (2.15)$$

2.3.5 Inisialisasi Bobot Awal Secara Random

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global (atau mungkin hanya local saja) terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Apabila nilai bobot awal terlalu besar, maka input ke setiap lapisan tersembunyi atau lapisan output akan jatuh pada daerah dimana turunan fungsi sigmoidnya akan sangat kecil. Sebaiknya, apabila nilai boboy awal terlalu kecil, maka input ke setiap lapisan tersembunyi atau lapisan output akan sangat kecil,

yang akan menyebabkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara random dengan nilai -0.5 sampai 0.5 (atau -1 sampai 1, atau interval yang lainnya)

2.3.6 Inisialisasi Bobot Awal dengan Metode Nguyen-Widrow

Metode Nguyen dan Widrow akan menginisialisasi bobot-bobot lapisan dengan nilai antara -0.5 sampai 0.5. Sedangkan bobot-bobot dari lapisan input ke lapisan tersembunyi dirancang sedemikian rupa sehingga dapat meningkatkan kemampuan lapisan tersembunyi dalam melakukan proses pembelajaran. Metode Nguyen-Widrow secara sederhana dapat diimplementasikan dengan prosedur sebagai berikut:

Langkah 0. Tetapkan : n = jumlah unit masukan

P = jumlah unit tersembunyi

β = faktor skala = $0.7\sqrt[n]{p}$ (2.16)

Langkah 1. Kerjakan untuk setiap unit lapisan tersembunyi (j= 1,2,3,..., p):

a. Inisialisasi semua bobot dari lapisan input ke lapisan tersembunyi:
   v_{ij} = bilangan random antara -0.5 sampai 0.5.
b. Hitung inisialisasi bobot menggunakan persamaan 2.17

$$\left\| v_j \right\| = \sqrt{v^2_{1j} + v^2_{2j} + ... + v^2_{nj}} \quad (2.17)$$

c. Inisialisasi ulang bobot-bobot

$$v_{ij} = \frac{\beta v_{ij}}{\left\| v_j \right\|} \quad (2.18)$$

d. Bias yang dipakai sebagai inisialisasi v_j = bilangan acak antara $-\beta$ dan β.

2.3.7 Jumlah Unit Tersembunyi

Menentukan jumlah lapis pada *hidden layer* dan menentukan jumlah umit per layernya sangat sulit. Hasil teoritis yang didapat menunjukkan bahwa jaringan dengan sebuah *hidden layer* sudah cukup bagi backpropagation untuk mengenali sembarang

perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Tetapi penambahan jumlah *hidden* layer kadangkala membuat pelatihan menjadi lebih mudah (Siang, J.J, 2005). Jumlah unit per *layer* yang optimal dapat ditentukan dengan beberapa percobaan.

2.3.8 Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan saraf tiruan (Kusumadewi, Sri. 2003), tetapi yang digunakan adalah fungsi Sigmoid Biner.

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki range (0,1).

Fungsi ini digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan saraf yang nilai *output* nya 0 atau 1.
Fungsi sigmoid biner dapat dilihat pada persamaan 2.19 :

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} \qquad (2.19)$$

Dimana: $f(x)$ = Fungsi Sigmoid
e = eksponensial
σ = konstanta sigmoid

dengan turunan pertama dari persamaan 2.20 :

$$f'(x) = \sigma f(x)[1 - f(x)] \qquad (2.20)$$

BAB III
METODOLOGI DAN PERANCANGAN SISTEM

3.1 Deskripsi Umum Sistem

Sistem pengenalan nomor plat merupakan sebuah sistem yang dapat melakukan pengenalan terhadap citra plat nomor yang berisi huruf dan angka. Pada dasarnya, sistem yang akan dibuat terdiri dari dua tahap yaitu, tahap pelatihan (*training*) dan tahap pengenalan.

Tahap pelatihan atau pembelajaran merupakan tahapan pertama sebelum dilakukan tahap pengenalan, dimana dalam tahapan ini sistem akan melakukan pembelajaran terhadap huruf dan angka dari nomor plat yang dimasukkan ke dalam sistem. Huruf dan angka yang dimasukkan ke dalam sistem berupa citra seperti contoh yang terlihat pada Gambar 3.1 untuk data latih dan Gambar 3.2 untuk data yang akan diujikan.

Pada tahapan pengenalan, sistem akan melakukan pengenalan terhadap citra yang dimasukkan oleh *user* yaitu berupa citra plat nomor (Gambar 3.2). Dalam tahapan ini, citra akan dicocokkan dengan data pelatihan hasil dari tahapan sebelumnya yaitu tahap pelatihan.

3.2 Batasan Sistem

Batasan dari perangkat lunak yang akan dikembangkan adalah sebagai berikut :

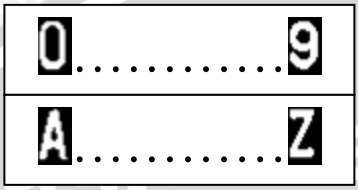
- Perangkat lunak yang dibangun hanya dapat memproses citra dengan format *Bitmap* (.bmp).
- Citra masukan berupa citra plat nomor saja.
- Sistem hanya dapat mengenali citra berupa plat nomor pribadi yang dikeluarkan oleh pihak kepolisian dengan warna dasar hitam dan warna huruf plat putih.

3.3 Data Uji

Pada penelitian ini data yang digunakan yaitu citra plat nomor yang diambil menggunakan kamera digital dalam file JPEG dan diubah ke dalam file Bitmap. Dan file Bitmap ini yang digunakan untuk data uji.

Data yang terkumpul akan digunakan untuk proses pelatihan dan proses pengujian sistem. Untuk pelatihan, data yang digunakan

merupakan citra karakter dari nomor plat berupa huruf alphabet A-Z dan angka 0-9 dapat dilihat pada Gambar 3.1 dan Gambar 3.2 menunjukkan data untuk pengujian berupa citra plat nomor.



Gambar 3.1 Contoh data untuk pelatihan

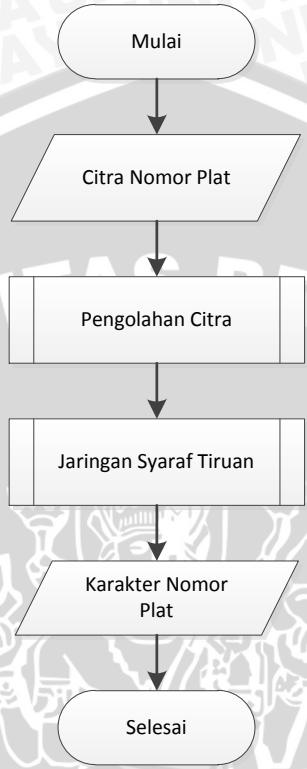
Gambar 3.2 Contoh data untuk pengujian

3.4 Perancangan Sistem

Pada perancangan sistem pengenalan nomor plat ini terdapat beberapa tahapan yang perlu dilakukan untuk dapat membangun sistem pengenalan nomor plat kendaraan dengan metode JST *Backpropagation*. Secara umum proses dalam sistem ini terdiri dari proses pengolahan citra dan proses jaringan syaraf tiruan. Masukkan yang digunakan dalam sistem ini yaitu berupa citra plat nomor dan keluaran sistem berupa karakter nomorplat kendaraan.

Dari gambar diagram alir sistem (Gambar 3.3) dengan penjelasan langkah-langkah sebagai berikut:

1. Masukkan citra plat nomor
2. Proses pengolahan citra (Gambar 3.4)
3. Proses Jaringan Syaraf Tiruan (Gambar 3.5)
4. Keluaran pada sistem pengenalan ini berupa karakter nomor plat kendaraan



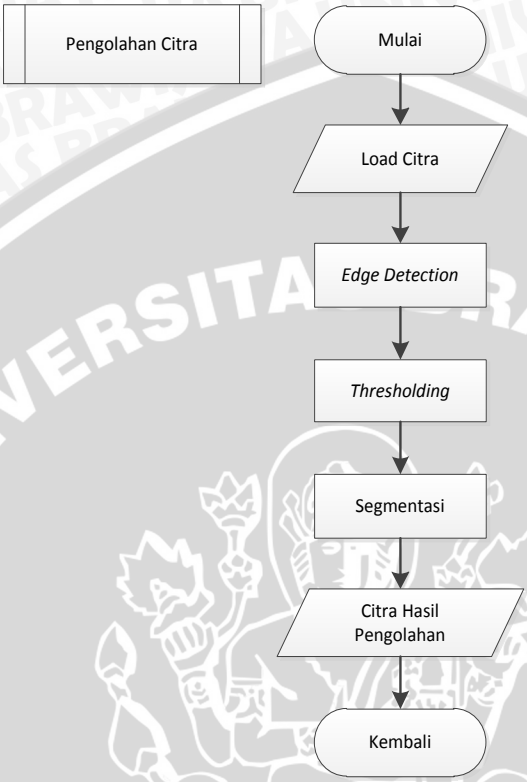
Gambar 3.3 Diagram Alir Sistem

3.4.1 Pengolahan Citra

Pengolahan citra merupakan satu proses yang sangat berpengaruh terhadap hasil pengenalan maupun pembelajaran pada karakter nomor plat.

Secara garis besar, proses pengolahan citra dapat digambarkan seperti pada Gambar 3.4 dengan langkah-langkah sebagai berikut:

1. Load citra
2. Proses *edge detection* atau pendeteksian tepi menggunakan metode *Sobel*.
3. Proses *thresholding* untuk mendapatkan citra benilai biner.
4. Proses segmentasi yaitu proses pemenggalan nomor plat sehingga dihasilkan citra huruf dan angka dari nomor plat.



Gambar 3.4 *Flowchart* Proses Pengolahan Citra

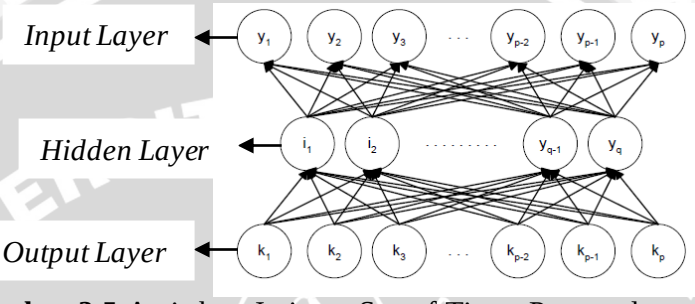
3.4.2 Proses Jaringan Syaraf Tiruan (JST)

Proses jaringan syaraf tiruan dilakukan setelah proses pengolahan citra selesai dilakukan. Proses ini dapat dibagi menjadi proses membuat struktur jaringan, proses pelatihan, dan proses pengenalan.

3.4.2.1 Struktur Jaringan Syaraf Tiruan

Jaringan yang akan dibangun meliputi tiga buah *layer*, yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* terdiri dari M x N buah neuron yang merupakan hasil pada saat normalisasi. Misalkan nilai M x N adalah 18 x 30, maka terdapat 540 buah neuron pada *input layer*. Dan untuk *output* terdapat 15 *output layer*, yaitu berupa vektor untuk mengenali karakter angka atau huruf dari nomor plat yang dikenali. Sedangkan jumlah neuron pada *hidden layer*

adalah n, dimana nilai n nantinya akan diujicobakan beberapa nilai. Hal ini dilakukan untuk mendapatkan arsitektur jaringan yang sesuai untuk melakukan pengenalan. Arsitektur jaringan syaraf tiruan untuk pengenalan nomor plat dapat dilihat pada Gambar 3.5.

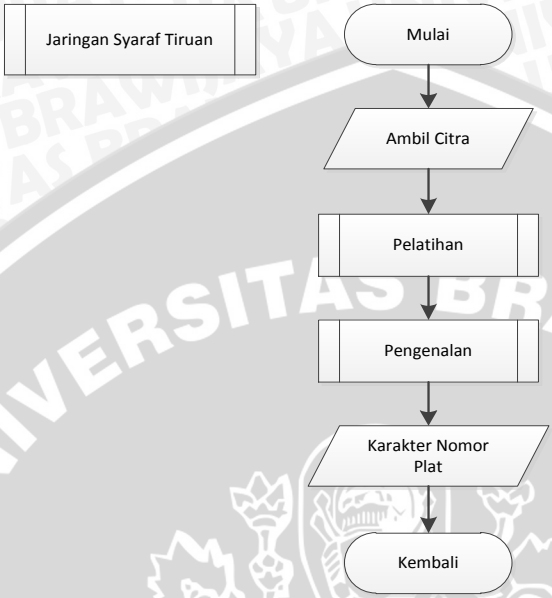


Gambar 3.5 Arsitektur Jaringan Syaraf Tiruan Pengenalan Nomor Plat

3.4.2.2 Proses Jaringan Syaraf Tiruan (JST)

Tahapan dalam proses JST ini terdiri dari dua proses utama yaitu proses pelatihan dan proses pengenalan. Metode JST yang digunakan dalam sistem ini yaitu menggunakan algoritma *backpropagation*. Secara keseluruhan langkah-langkah pada proses JST untuk pengenalan nomor plat kendaraan menggunakan algoritma *backropagation* diilustrasikan pada Gambar 3.6 dengan penjelasan sebagai berikut:

1. Mulai
2. Ambil citra
3. Proses pelatihan (Gambar)
4. Proses pengenalan
5. Keluaran dari proses ini berupa karakter nomor plat
6. Kembali



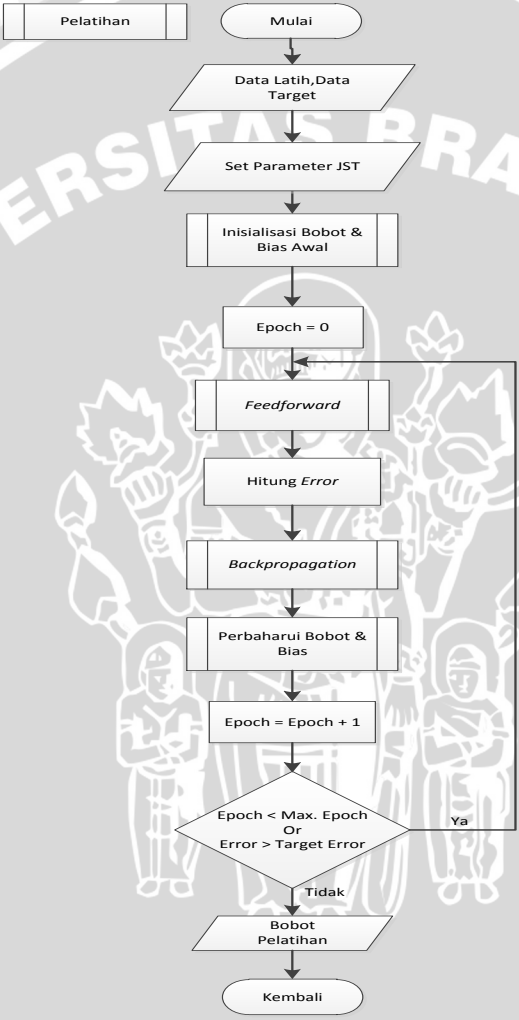
Gambar 3.6 *Flowchart* Proses JST

3.4.2.3 Proses Pelatihan

Algoritma pelatihan *backpropagation* pada dasarnya terbagi menjadi 3 langkah, yaitu: langkah maju (*feedforward),* propagasi balik (*backpropagation)* dan perubahan bobot. Secara keseluruhan langkah-langkah pelatihan algoritma *backropagation* dapat dilihat pada Gambar 3.7 dengan langkah-langkah sebagai berikut:

1. Mulai
2. Masukkan data latih dan data target
3. *Set* parameter JST
4. Proses inisialisasi bobot dan bias secara *random* yang berkisar antara -0.05 sampai 0.05
5. *Setting* parameter iterasi awal (*epoch*) sama dengan 0.
6. Selama *epoch* kurang dari maksimal iterasi maka lakukan proses *feedforward*
7. Lakukan perhitungan kesalahan antara pola *output* JST dan pola target.
8. Periksa apakah kesalahan *output* lebih besar dari kesalahan yang ditargetkan? Jika ya, maka lakukan langkah 9, jika tidak, maka lakukan langkah 10.

9. Lakukan langkah *backpropagation*.
10. Proses berhenti, bobot dan bias akhir pelatihan disimpan pada dataset, data siap digunakan untuk proses pengujian.
11. Selesai.

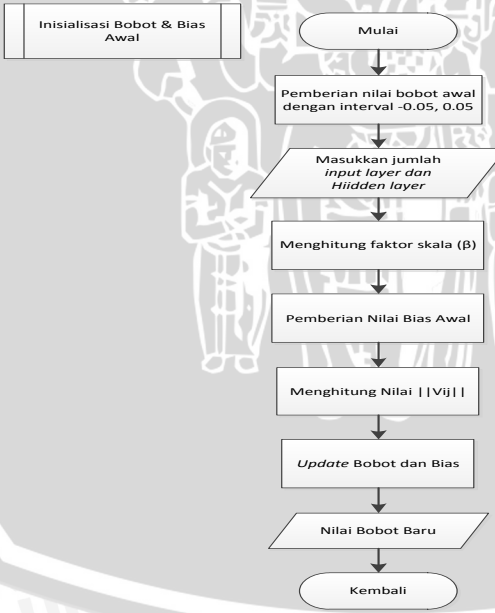


Gambar 3.7 Diagram Alir Proses Pelatihan

3.4.2.4 Inisialisasi Bobot dan Bias Awal

Inisialisasi Bobot dan Bias Awal Nguyen Widrow digunakan untuk menentukan nilai bobot dan bias awal dari unit masukan ke unit tersembunyi. Untuk proses Inisialisasi Bobot dan Bias Awal Nguyen Widrow dapat dilihat pada Gambar 3.8 dan langkah-langkahnya adalah sebagai berikut:

1. Mulai
2. Pemberian nilai awal pada semua bobot dari unit masukan ke unit tersembunyi dengan bilangan acak dalam interval -0,05 sampai 0,05
3. Tentukan jumlah unit masukan dan unit tersembunyi.
4. Kemudian hitung faktor skala menggunakan rumus $\beta = 0.7\sqrt[n]{p}$
5. Setelah diperoleh hasil perhitungan faktor skala, selanjutnya adalah Pemberian nilai bias awal dari unit masukan ke unit tersembunyi
6. Hitung || Vj || menggunakan persamaan 2.17
7. Selanjutnya, pemberian nilai bobot kembali dari unit masukan ke unit tersembunyi menggunakan persamaan 2.18.
8. Selesai

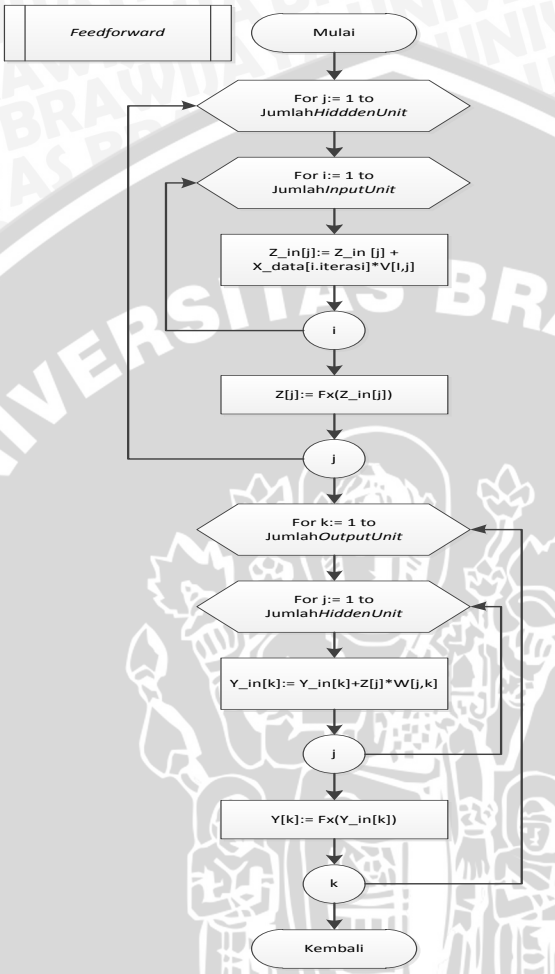


Gambar 3.8 *Flowchart* Inisialisasi Bobot dan Bias Awal

3.4.2.5 *Feedfordward*

Proses yang dilakukan dalam fase *feedforward* adalah menjumlahkan perkalian antara masukan dengan bobot yang ada dan menghitung nilai aktivasinya untuk kemudian hasil dari perhitungan tersebut dijadikan masukan oleh lapisan yang berada di atasnya. Untuk proses *feedforward* dapat dilihat pada Gambar 3.9 dan langkah-langkah *feedforward* adalah sebagai berikut:

1. Mulai
2. Kalikan seluruh data *input* pada *neuron input* dengan bobot random pada masing-masing bobot koneksi bobot *Input* yang terhubung dengan *neuron input*. Kemudian jumlahkan seluruh vektor bobot yang menuju neuron *hidden* yang sama.
3. Lakukan aktivasi hasil penjumlahan tersebut pada masing-masing *neuron* di lapisan tersembunyi, sehingga *ouput* pada lapis ini berada pada kisaran 0 dan 1.
4. Kalikan seluruh data hasil aktivasi masing-masing neuron lapis *hidden* pada *neuron hidden* dengan bobot pada masing-masing koneksi bobot *output* yang terhubung dengan *neuron* pada lapis *hidden*. Kemudian jumlahkan seluruh vektor bobot yang menuju *neuron output* yang sama.
5. Lakukan aktivasi hasil penjumlahan tersebut pada masing masing neuron di lapisan *output*, sehingga *ouput* pada lapis ini berada pada kisaran 0 dan 1.
6. Selesai.

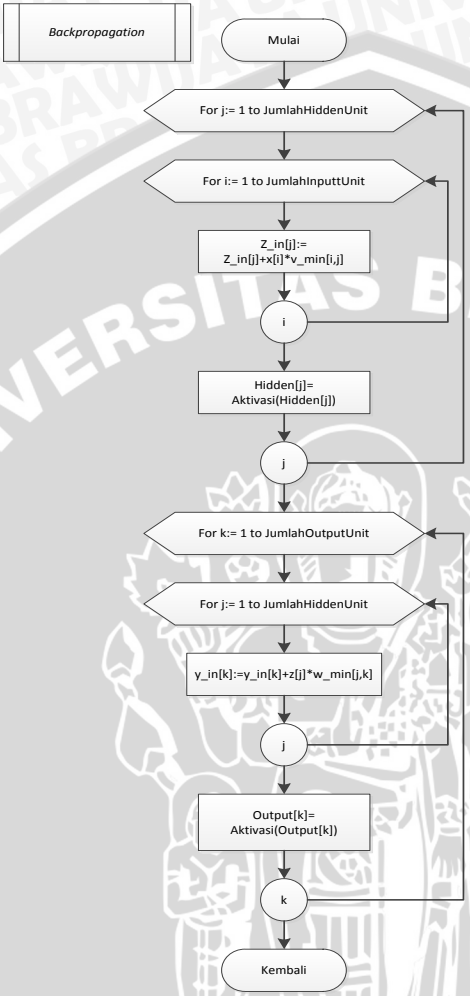


Gambar 3.9 *Flowchart* Proses *Feedforward*

3.4.2.6 *Backpropagation*

Proses *backpropagation* adalah perhitungan informasi kesalahan pada tiap *neuron* pada masing-masing lapisan dimulai dari kesalahan pada *output layer* hingga *hidden layer* terdekat dengan *input layer*. Informasi kesalahan berguna untuk menghitung faktor peubah bobot yang akan digunakan untuk perbaikan bobot lama. Algoritma *backpropagation* diperlihatkan pada Gambar 3.10 dan langkah-langkah proses *backpropagation* adalah sebagai berikut:

1. Mulai
2. Pada *output layer*. Pertama hitung selisih antara target pelatihan dengan *output*. Selisih ini disebut sebagai *Error*. Kalikan selisih ini dengan *output* yang telah diaktivasi dengan fungsi turunan aktivasi. Hasil perkalian merupakan faktor kesalahan pada *output layer* dan akan digunakan untuk menghitung faktor kesalahan pada *hidden layer* dan untuk menghitung faktor peubah bobot pada vektor bobot menuju *output*.
3. Hitung besar faktor peubah bobot baru pada vektor yang menuju *output layer* dengan cara mengalikan *Learning Rate* dengan *hidden layer* dan *error* pada *output layer*.
4. Pada *hidden layer*. Untuk menghitung faktor kesalahan masing-masing *neuron hidden layer* lakukan untuk masing-masing faktor kesalahan pada *output* kalikan dengan bobot lama yang terkoneksi dengan *neuron output layer*. Kemudian hasil perkalian pada seluruh koneksi yang terhubung dengan masing-masing *neuron* pada *hidden layer* akan dijumlahkan. Faktor kesalahan pada *neuron hidden layer* akan digunakan untuk menghitung peubah bobot pada koneksi dari *input layer* menuju *hidden layer*.
5. Hitung nilai faktor peubah bobot baru pada tiap vektor yang menuju *hidden layer* dengan cara mengalikan *learning rate* dengan input layer dan *error* di *hidden layer*.
6. Selesai.

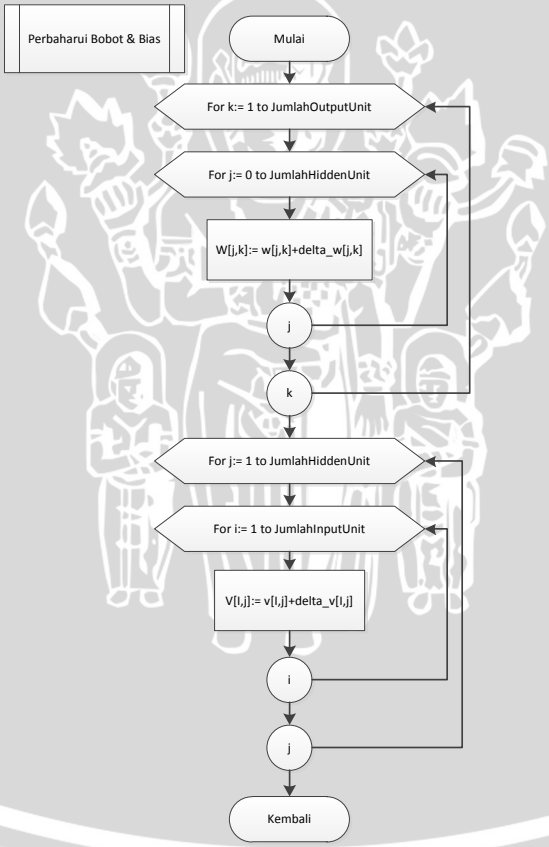


Gambar 3.10 *Flowchart* Proses *Backpropagation*

3.4.2.7 Perbaharui Bobot dan Bias

Hasil dari proses *backpropagation* adalah nilai faktor peubah bobot yang digunakan untuk melakukan perubahan bobot. Proses perubahan bobot dilakukan untuk mendapatkan nilai bobot baru. Proses ini digambarkan dengan diagram alir pada Gambar 3.11.
Langkah-langkah yang dilakukan dalam proses perubahan bobot ini adalah :

1. Mulai
2. Perbaiki nilai bobot untuk setiap koneksi yang menuju ke *output layer* dengan cara menjumlahkan nilai bobot lama dengan suku peubah bobot yang telah dihitung pada proses *backpropagation*.
3. Perbaiki nilai bobot untuk setiap koneksi yang menuju ke *hidden layer* dengan cara menjumlahkan nilai bobot lama dengan suku peubah bobot yang telah dihitung pada proses *backpropagation*.
4. Selesai. Bobot baru akan digunakan untuk proses *feedforward* kedua dan seterusnya hingga bobot optimal didapatkan.

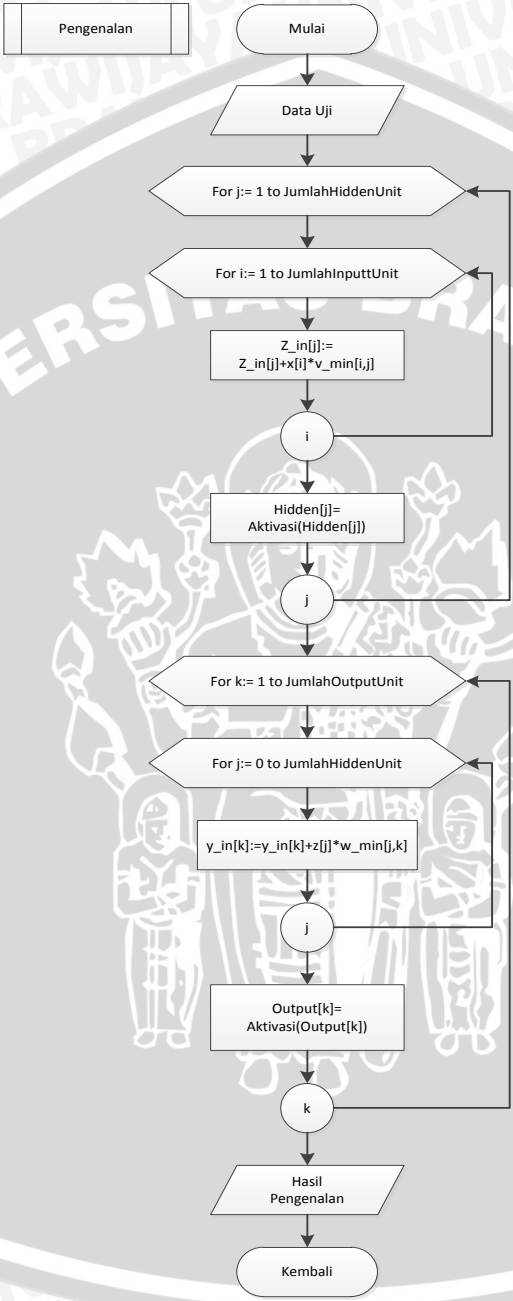


Gambar 3.11 *Flowchart* Proses Perbaharui Bobot dan Bias

3.4.2.8 Proses Pengenalan

Tahap pengujian jaringan syaraf tiruan *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju (*feed fordward*) dari algoritma pelatihan *backpropagation*. Di dalam proses pengenalan, keluaran yang dihasilkan oleh jaringan tidak akan diproses lagi menuju ke *backpropagation*. Secara keseluruhan langkah-langkah pengujian algoritma *backropagation* diilustrasikan pada Gambar 3.12 dan langkah-langkahnya adalah sebagai berikut:

1. Mulai
2. Inisialisasi bobot-bobot yang diperoleh dari proses pelatihan.
3. Masukkan data pengujian.
4. Kalikan seluruh data *input* pada *neuron input* dengan bobot *random* pada masing-masing bobot koneksi bobot *input* yang terhubung dengan *neuron input*. Kemudian jumlahkan seluruh vektor bobot yang menuju *neuron hidden* yang sama.
5. Lakukan aktivasi hasil penjumlahan tersebut pada masing-masing *neuron* di *hidden layer*, sehingga *output* pada lapisan ini berada pada kisaran 0 dan 1.
6. Kalikan seluruh data hasil aktivasi masing-masing *neuron hidden layer* pada *neuron hidden* dengan bobot pada masing-masing koneksi bobot *output* yang terhubung dengan *neuron* pada *hidden layer*. Kemudian jumlahkan seluruh vektor bobot yang menuju *neuron output* yang sama.
7. Lakukan aktivasi hasil penjumlahan tersebut pada masing masing *neuron* di *output layer*, sehingga *ouput* pada lapisan ini berada pada kisaran 0 dan 1.
8. Diperoleh keluaran yang berupa hasil pengenalan karakter nomor plat.
9. Selesai.

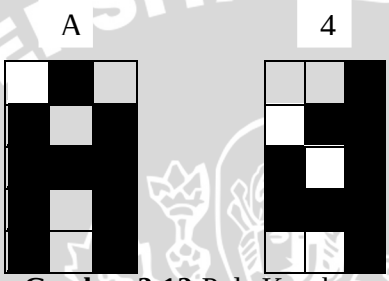


Gambar 3.12 *Flowchart* Proses Pengenalan

3.5 Contoh Perhitungan Manual

Input data

Diberikan pola huruf A dan angka 4 yang direpresentasikan menggunakan kode 0 dan 1 pada matriks berukuran 3x5. Pola karakter huruf A dan angka 4 ditunjukkan pada Gambar 12. Sedangkan Gambar 3.13 merupakan representasi dari pola karakter (Gambar 3.12) menggunakan kode 0 dan 1 yang digunakan untuk input pada contoh perhitungan.



Gambar 3.12 Pola Karakter

A

0	1	0
1	0	1
1	1	1
1	0	1
1	0	1

4

0	0	1
0	1	1
1	0	1
1	1	1
0	0	1

Gambar 3.13 Representasi Pola Karakter

Selanjutnya data inputan berupa matrik berukuran 3x5 diubah menjadi vektor seperti pada Tabel 3.1 yang digunakan untuk data masukan pertama yang akan dimasukkan pada iterasi pertama dengan targetnya yang dapat dilihat pada tabel 3.2. Sedangkan Tabel 3.3 untuk data masukan kedua pada iterasi kedua dengan targetnya yang ditunjukkan pada tabel 3.4.

Tabel 3.1 Data Pertama

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1

Tabel 3.2 Target untuk Data Pertama

t1	t2
1	0

Tabel 3.3 Data Kedua

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15
0	0	1	0	1	1	1	0	1	1	1	1	0	0	1

Tabel 3.4 Target untuk Data Kedua

t1	t2
0	1

3.5.1 Pelatihan Menggunakan *Bckpropagation*

Iterasi 1

Data masukan pertama pada iterasi pertama seperti dapat dilihat pada Gambar 3.1 dan targetnya ditunjukkan pada tabel 3.2.

Inisialisasi Bobot

Inisialisasi bobot-bobot dari *input layer* ke *hidden layer* mula-mula diberi nilai acak antara -1 hingga 1. Nilai bobot awal (Vij) dapat dilihat pada 3.5.

Tabel 3.5 Bobot dari *Input* ke *hidden layer* (Vij)

Vij	z1	z2	z3	z4	z5
x1	-1	-0.2	-0.5	0.8	-0.9
x2	0.6	0.8	-0.4	-0.7	-0.8
x3	-0.8	-0.3	-0.2	-0.1	0.5
x4	0.4	1	0.9	0	0.4
x5	-0.1	-0.5	-0.8	-1	-0.4
x6	-0.5	0.2	0.5	0.8	-0.8
x7	-0.3	-0.5	0.4	-0.1	1
x8	-0.4	-0.9	-0.9	0.1	-0.9
x9	0.6	1	0.5	0.3	0.7
x10	-0.3	-0.2	-0.1	-0.5	-0.4
x11	0.6	-0.6	0.4	1	0.3
x12	-0.5	-0.7	-1	0.8	0.4

x13	-0.8	-0.4	0.6	0.9	0.9
x14	-0.2	0.4	0.6	-0.2	-0.7
x15	-0.1	-1	0.5	-0.3	-1

Inisialisasi bobot baru dan bias ke *hidden layer* menggunakan metode Nguyen Widrow: $\beta = 0.7 \sqrt[n]{p}$

Jumlah *unit Input* (n) = 15

Jumlah *unit hidden* (p) = 5

Faktor skala = $\beta = 0{,}7\sqrt[15]{5} = 0.233333$. Bias awal yang dipakai adalah bilangan acak antara -0.233333 hingga 0.233333.

Hitung nilai Vj berdasarkan persamaan 2.17

$\|V_1\| = \sqrt{(-1)^2 + (0.6)^2 + (-0.8)^2 + (0.4)^2 + (-0.1)^2 + (-0.5)^2 + (-0.3)^2 + (-0.4)^2 + (0.6)^2 + (-0.3)^2 + (0.6)^2 + (-0.5)^2 + (-0.8)^2 + (-0.2)^2 + (-0.1)^2} = 2.10238$

.

.

$\|V_{15}\| = \sqrt{(-0.9)^2 + (-0.8)^2 + (0.5)^2 + (0.4)^2 + (-0.4)^2 + (-0.8)^2 + (1)^2 + (-0.9)^2 + (0.7)^2 + (-0.4)^2 + (0.3)^2 + (0.4)^2 + (0.9)^2 + (-0.7)^2 + (-1)^2} = 2.769476$

Nilai Vj dapat dilihat pada Tabel 3.6

Tabel 3.6 Inisialisasi Nilai Vj

j	Vj
1	2.10238
2	2.515949
3	2.347339
4	2.391652
5	2.769476

Dengan menggunakan persamaan 2.18 , diperoleh nilai bobot yang digunakan untuk inisialisasi Vij (baru) :

$V_{1,1}(\text{baru}) = \frac{\beta * v_{11(lama)}}{|v_1|} = \frac{0.233333 * (-1)}{2.10238} = -0.11099$

.

.

$V_{15,5}(\text{baru}) = \frac{\beta * v_{11(lama)}}{|v_1|} = \frac{0.233333 * (-1)}{2.769476} = -0.08425$

Nilai bobot baru yang digunakan untuk inisialisasi *input layer* ke *hidden layer* (V_{ij}) dapat dilihat pada tabel 3.7

Tabel 3.7 Nilai Vij (baru)

	z1	z2	z3	z4	z5
x1	-0.11099	-0.01855	-0.0497	0.078049	-0.07583
x2	0.066591	0.074193	-0.03976	-0.06829	-0.0674
x3	-0.08879	-0.02782	-0.01988	-0.00976	0.042126
x4	0.044394	0.092742	0.089463	0	0.033701
x5	-0.0111	-0.04637	-0.07952	-0.09756	-0.0337
x6	-0.05549	0.018548	0.049702	0.078049	-0.0674
x7	-0.0333	-0.04637	0.039761	-0.00976	0.084252
x8	-0.04439	-0.08347	-0.08946	0.009756	-0.07583
x9	0.066591	0.092742	0.049702	0.029268	0.058976
x10	-0.0333	-0.01855	-0.00994	-0.04878	-0.0337
x11	0.066591	-0.05565	0.039761	0.097562	0.025276
x12	-0.05549	-0.06492	-0.0994	0.078049	0.033701
x13	-0.08879	-0.0371	0.059642	0.087805	0.075827
x14	-0.0222	0.037097	0.059642	-0.01951	-0.05898
x15	-0.0111	-0.09274	0.049702	-0.02927	-0.08425

Nilai bias V0j yang dipakai adalah bilangan acak antara –β hingga β yaitu antara nilai -0.233333 hingga 0.233333. Nilai V_{0j} dapat dilihat pada tabel 3.8.

Tabel 3.8 Nilai Bias (V0j)

	V0j
1	-0.1903
2	-0.2297
3	0.0727
4	-0.056
5	0.1802

Nilai bobot awal dari *hidden layer* ke *output layer* (Wjk) diperoleh dari nilai acak antara -1 sampai 1, inisialisasi nilai Wjk dapat dilihat pada Tabel 3.9

Tabel 3.9 Nilai bobot ke *output layer* (Wjk)

Wjk	Y1	Y2
z1	-0.9	-0.8
z2	0	0.4
z3	-0.5	0.1
z4	0.8	0.8
z5	0.6	0.3

Nilai bias W0k juga diperoleh dari nilai acak antara -1 sampai 1, inisialisasi nilai bias W0k dapat dilihat pada Tabel 3.10

Tabel 3.10 Nilai Bias W0k

	w0k
1	-0.4
2	1

Proses *Feedforward*

Hitung keluaran di unit *hidden* (Zj) menggunakan persamaan 2.1. Hasil nilai operasi *hidden* Z_in_j dapat dilihat pada Tabel 3.11.

Z_in_1 = -0.1903+ [(0*-0.11099)+(1*0.066591)+(0*-0.08879)+(1*0.044394)+(0*-0.0111)+(1*-0.05549)+(1*-0.0333)+(1*-0.04439)+(1*0.066591)+(1*-0.0333)+(0*0.066591)+(1*-0.05549)+(1*-0.08879)+(0*-0.0222)+(1*-0.0111)] = -0.33458

.

.

Z_in_5 = 0.1802 + [(0*-0.07583)+(1*-0.0674)+(0*0.042126)+(1*0.033701)+(0*-0.0337)+(1*-0.0674)+(1*0.084252)+(1*-0.07583)+(1*0.058976)+(1*-0.0337)+(0*0.025276)+(1*0.033701)+(1*0.075827)+(0*-0.05898)+(1*-0.08425)] = 0.138074

Tabel 3.11 Hasil Operasi pada *hidden* (Z_in)

j	Z_in
1	-0.33458
2	-0.29462
3	0.172103
4	0.07083
5	0.138074

Menghitung aktivasinya menggunakan persamaan 2.3. Hasil aktivasi unit *hidden* (Z_in) dapat dilihat pada Tabel 3.12

Tabel 3.12 Hasil Aktivasi Z_in

j	Z
1	2.397355
2	2.342615
3	1.841892
4	1.93162
5	1.871034

Selanjutnya yaitu menghitung keluaran unit *output* (Yk) menggunakan persamaan 2.4. Hasil Operasi pada unit *output* dapat dilihat pada tabel 3.13

Y_in_1= -0.4 + ((2.397355*-0.9)+(2.342615*0)+(1.841892*-0.5)+(1.93162*0.8)+(1.871034*0.6)) = -0.81065

Tabel 3.13 Hasil Operasi pada *output* (Y_in)

k	Y_in
1	-0.81065
2	2.309958

Kemudian, hitung aktivasi Y_in. Hasil aktivasi Y_in dapat dilihat pada Tabel 3.14

Tabel 3.14 Hasil aktivasi Y_in

j	Y
1	3.24937
2	1.099265

Proses *Backpropagation*

Hitung nilai faktor δ di unit *output* menggunakan persamaan 2.6. Nilai faktor δ di unit *output* dapat dilihat pada tabel 3.15.

Tabel 3.15 Nilai Kemelesetan Yk

k	δ
1	-0.06556
2	0.007418

Hitung nilai perubahan bobot dengan α=0.2 dengan menggunakan persamaan 2.8. Hasil perhitungan nilai perubahan bobot (Δw_{jk}) dapat dilihat pada tabel 3.16.

Tabel 3.16 Nilai Perubahan bobot (ΔWjk)

ΔWjk	w1	w2
z1	-0.00547	0.000619
z2	-0.0056	0.000633
z3	-0.00712	0.000805
z4	-0.00679	0.000768
z5	-0.00701	0.000793

Mengkoreksi bobot bias ΔW0k menggunakan persamaan 2.9. Hasil koreksi bias pada unit *output* dapat dilihat pada Tabel 3.17

Tabel 3.17 Koreksi Nilai Bias Output (W0k)

k	w0
1	-0.01311
2	0.001484

Hitung penjumlahan kesalahan di unit *hidden* menggunakan persamaan 2.10. Masing-masing penimbang yang menghubungkan unit-unit lapisan keluaran dengan unit-unit pada *hidden layer* (Z_j ,j=1..,p) dikalikan delta dan dijumlahkan sebagai masukan ke unit-unit lapisan berikutnya. Hasil perhitungan penimbang di unit *hidden* dapat dilihat pada tabel 3.18.

Tabel 3.18 Faktor penimbang di unit *hidden*

j	δ_in_j
1	0.053073
2	0.002967
3	0.033524
4	-0.04652
5	-0.03711

Kemudian hitung aktivasi faktor kesalahan δ di unit *hidden*, menggunakan persamaan 2.11. Hasil perhitungan aktivasi faktor kesalahan di unit hidden, dapat dilihat pada tabel 3.19.

Tabel 3.19 Aktivasi faktor kesalahan di unit *hidden*

j	δ
1	0.012904
2	0.000726
3	0.008319
4	-0.01161
5	-0.00923

Suku perubahan bobot ke unit *hidden* ΔV_{ij} dihitung dengan persamaan 2.12. Nilai perbaikan bobot penimbang ke unit *hidden* dapat dilihat pada tabel 3.20.

Tabel 3.20 Nilai perubahan bobot (ΔVij)

ΔVij	ΔV1	ΔV2	ΔV3	ΔV4	ΔV5
x1	0	0	0	0	0
x2	0.002581	0.000145	0.001664	-0.00232	-0.00185
x3	0	0	0	0	0
x4	0.002581	0.000145	0.001664	-0.00232	-0.00185
x5	0	0	0	0	0
x6	0.002581	0.000145	0.001664	-0.00232	-0.00185
x7	0.002581	0.000145	0.001664	-0.00232	-0.00185
x8	0.002581	0.000145	0.001664	-0.00232	-0.00185
x9	0.002581	0.000145	0.001664	-0.00232	-0.00185
x10	0.002581	0.000145	0.001664	-0.00232	-0.00185
x11	0	0	0	0	0
x12	0.002581	0.000145	0.001664	-0.00232	-0.00185
x13	0.002581	0.000145	0.001664	-0.00232	-0.00185
x14	0	0	0	0	0
x15	0.002581	0.000145	0.001664	-0.00232	-0.00185

Kemudian perbaiki nilai bias $\Delta V_{0j,}$ menggunakan persamaan 2.13. Nilai bias ΔV_{0j} baru dapat dilihat pada tabel 3.21.

Tabel 3.21 Hasil Nilai Koreksi Bobot Bias ΔV_{0j}

	ΔV_{0j}
z1	0.002581
z2	0.000145
z3	0.001664
z4	-0.00232
z5	-0.00185

Perbaharui bobot di unit *hidden* menggunakan persamaan 2.15. Bobot baru di unit *hidden* dapat dilihat pada tabel 3.22.

Tabel 3.22 Bobot baru (Vij)

Vij	z1	z2	z3	z4	z5
x1	-0.11099	-0.01855	-0.0497	0.078049	-0.07583
x2	0.069172	0.074339	-0.0381	-0.07062	-0.06925
x3	-0.08879	-0.02782	-0.01988	-0.00976	0.042126
x4	0.046975	0.092887	0.091127	-0.00232	0.031854
x5	-0.0111	-0.04637	-0.07952	-0.09756	-0.0337
x6	-0.05291	0.018694	0.051366	0.075726	-0.06925
x7	-0.03071	-0.04623	0.041425	-0.01208	0.082405
x8	-0.04181	-0.08332	-0.0878	0.007433	-0.07767
x9	0.069172	0.092887	0.051366	0.026946	0.057129
x10	-0.03071	-0.0184	-0.00828	-0.0511	-0.03555
x11	0.066591	-0.05565	0.039761	0.097562	0.025276
x12	-0.05291	-0.06477	-0.09774	0.075726	0.031854
x13	-0.08621	-0.03695	0.061306	0.085482	0.07398
x14	-0.0222	0.037097	0.059642	-0.01951	-0.05898
x15	-0.00852	-0.0926	0.051366	-0.03159	-0.0861

Kemudian perbaiki nilai bias V0j. Nilai V0j baru dapat dilihat pada tabel 3.23

Tabel 3.23 Nilai V0j baru

j	V0j
1	-0.18772
2	-0.22955
3	0.074364
4	-0.05832
5	0.178353

Perbaharui bobot di unit *output*, menggunakan persamaan 2.14. Nilai bobot baru di unit *hidden* ke *output* dapat dilihat pada tabel 3.24.

Tabel 3.24 Perubahan Bobot Wjk(baru)

Wjk	Y1	Y2
z1	-0.90547	-0.79938
z2	-0.0056	0.400633
z3	-0.50712	0.100805
z4	0.793212	0.800768
z5	0.592992	0.300793

Kemudian perbaiki nilai bias W_{0k}.Nilai bias W_{0k} baru dapat dilihat pada tabel 3.25

Tabel 3.25 Nilai Bias W0k(baru)

k	W0
1	-0.41311
2	1.001484

Iterasi 2

Data masukan pada iterasi kedua dapat dilihat pada tabel 3.3 dengan targetnya yang ditunjukkan pada tabel 3.4. Pada iterasi kedua, didapat bobot bias baru seperti terlihat pada tabel 3.26

Tabel 3.26 Nilai Bobot Baru di unit *Hidden*

Vij	z1	z2	z3	z4	z5
x1	-0.11099	-0.01855	-0.0497	0.078049	-0.07583
x2	0.069172	0.074339	-0.0381	-0.07062	-0.06925
x3	-0.08879	-0.02782	-0.01988	-0.00976	0.042126
x4	0.046975	0.092887	0.091127	-0.00232	0.031854
x5	-0.0111	-0.04637	-0.07952	-0.09756	-0.0337
x6	-0.05291	0.018694	0.051366	0.075726	-0.06925
x7	-0.03071	-0.04623	0.041425	-0.01208	0.082405
x8	-0.04181	-0.08332	-0.0878	0.007433	-0.07767
x9	0.069172	0.092887	0.051366	0.026946	0.057129
x10	-0.03071	-0.0184	-0.00828	-0.0511	-0.03555
x11	0.066591	-0.05565	0.039761	0.097562	0.025276
x12	-0.05291	-0.06477	-0.09774	0.075726	0.031854
x13	-0.08621	-0.03695	0.061306	0.085482	0.07398
x14	-0.0222	0.037097	0.059642	-0.01951	-0.05898
x15	-0.00852	-0.0926	0.051366	-0.03159	-0.0861

Tabel 3.27 Nilai bias *input* ke *hidden layer* baru ($V_{0j\ baru}$)

j	V0j
1	-0.18772
2	-0.22955
3	0.074364
4	-0.05832
5	0.178353

Tabel 3.28 Perubahan Bobot Wjk(baru)

Wjk	Y1	Y2
z1	-0.90547	-0.79938
z2	-0.0056	0.400633
z3	-0.50712	0.100805
z4	0.793212	0.800768
z5	0.592992	0.300793

Kemudian perbaiki nilai bias W_{0k}.Nilai bias W_{0k} baru dapat dilihat pada tabel 3.29

Tabel 3.29 Nilai Bias W0k(baru)

K	W0
1	-0.41311
2	1.001484

Setelah dilakukan perbaikan bobot sampai didapatkan *error* minimal atau maksimum *epoch*, maka hasil perubahan bobot akan disimpan untuk digunakan pada saat ada data baru yang akan dilakukan proses pengenalan.

3.5.2 Pengujian Menggunakan *Backpropagation*

Langkah 1: Inisialisasi bobot yang diperoleh dari proses pelatihan
Langkah 2: Hitung keluaran unit *hidden* (Zj)
Masing-masing unit *hidden* menjumlahkan bobot sinyalnya, pada langkah ini menggunakan persamaan 2.16. Hasil operasi pada *hidden* ditunjukkan pada Tabel 3.30.

Tabel 3.30 Operasi pada *hidden*

J	Z_in
1	0.672393
2	0.530191
3	1.104228
4	1.01555
5	1.192543

Kemudian hitung aktivasinya dengan menggunakan persamaan 2.21. Hasil aktivasi unit *hidden* dapat dilihat pada tabel 3.31.

Tabel 3.31 Hasil Aktivasi Operasi *hidden*

j	Z
1	0.662039
2	0.629528
3	0.751051
4	0.734105
5	0.767196

Langkah 3: Hitung keluaran unit Yk

Masing-masing unit *output* menjumlahkan bobot sinyal *input*. Untuk menghitung nilai keluaran menggunakan persamaan 2.22. Hasil operasi pada *output neuron* dapat dilihat pada tabel 3.32.

Tabel 3.32 Operasi pada *output layer* (y_in_k)

k	Y_in
1	-0.35972
2	1.618797

Kemudian hitung aktivasinya menggunakan persamaan 2.23. Hasil aktivasinya dapat dilihat pada tabel 3.33.

Tabel 3.33 Hasil aktivasi y_in

k	Y
1	0.411027
2	0.834629

Misalkan *threshold* = 0,5; jika nilai $y \geq 0{,}5$ maka *output* yang diberikan adalah 1, dan jika nilai $y \leq 0{,}5$ maka *output* yang diberikan adalah 0. Dengan demikian *output* dari data kedua (Gambar 3.4) adalah t1 = 0 (karena $0.411027 < 0{,}5$) dan t2 = 1 (karena $0.834629 > 0{,}5$), sesuai dengan target yang diharapkan.

3.6 Rancangan Uji Coba

3.6.1 Rancangan Uji Coba Pengaruh Jumlah *Neuron Hidden*

Untuk mengetahui pengaruh *neuron* terhadap nilai kesalahan minimal, dilakukan pengujian dengan menentukan jumlah *neuron* yang diujikan. Jumlah iterasi maksimal ditetapkan 10000 iterasi. Rancangan uji coba ini akan disajikan seperti terlihat pada Tabel 3.34

Tabel 3.34 Pengaruh Jumlah *Neuron Hidden*

Percobaan	*Hidden Layer*	Max. Iterasi	MSE
1	60		
2	70		
3	80		
4	90		
5	100		

3.6.2 Rancangan Uji Coba Pengaruh Nilai *Learning Rate*

Uji coba ini dilakukan dengan menggunakan nilai *learning rate* yang berbeda yaitu 0.1, 0.3, 0.5, 0.7, dan 0.9. Kombinasi *learning rate* dilakukan untuk mengetahui pengaruh nilai *learning rate* terhadap kesalahan minimal. Rancangan uji coba ini akan disajikan seperti terlihat pada tabel 3.35

Tabel 3.35 Pengaruh Nilai *Learning Rate*

Percobaan	*Learning Rate*	Max. Iterasi	MSE
1	60		
2	70		
3	80		
4	90		
5	100		

3.6.3 Rancangan Pengujian Sistem

3.6.3.1 Rancangan Uji Segmentasi

Uji coba ini dilakukan untuk mengetahui kemampuan sistem dalam melakukan proses pemenggalan (segmentasi) karakter pada sistem pengenalan nomor plat. Dari hasil uji coba ini akan dilakukan perhitungan tingkat akurasi keberhasilan system dalam proses segmentasi. Rancangan uji segmentasi akan disajikan seperti pada tabel 3.36

Tabel 3.36 Hasil Uji Segmentasi

No	Nama File (.bmp)	Benar	Salah

3.6.3.2 Rancangan Uji Pengenalan

Uji pengenalan merupakan proses pengujian terakhir untuk mengetahui tingkat keberhasilan system untuk melakukan pengenalan. Hasil pengenalan disajikan seperti pada tabel 3.37.

Tabel 3.37 Hasil Uji Pengenalan

Nama File (.bmp)	Jumlah Karakter	Benar	Salah	Keakuratan

3.7 Evaluasi

Tahap evaluasi merupakan tahapan akhir yang nantinya akan dilakukan pada hasil uji coba terhadap sistem yang telah dibangun dengan melakukan proses pembelajaran dan proses pengenalan terhadap citra nomor plat.

Ada beberapa parameter yang digunakan untuk mengetahui seberapa besar keberhasilan sistem terhadap pengenalan citra nomor plat. Parameter yang dimaksud adalah besarnya nilai MSE (*Mean Square Error*) yang dihasilkan pada saat pelatihan, serta berapa persentase keberhasilan pengenalan nomor plat. Nilai keberhasilan dihitung dengan cara :

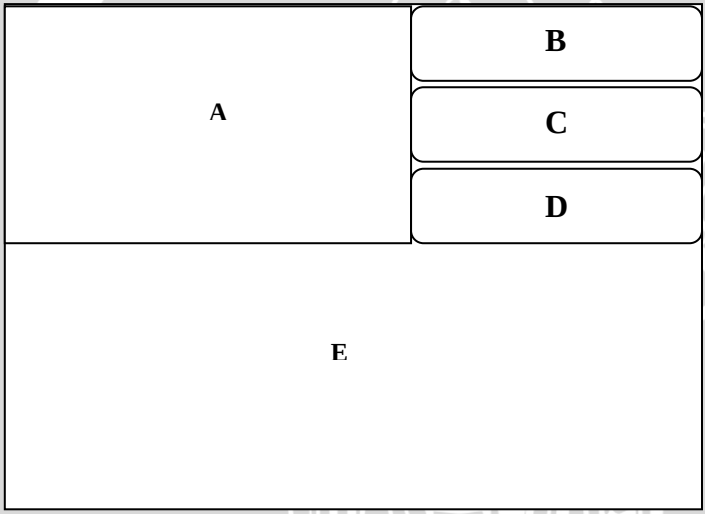
$$\text{Keberhasilan} = \frac{\text{Jumlah karakter dikenali dengan benar}}{\text{Jumlah karakter yang diujikan}} \times 100\%$$

Dengan menggunakan perhitungan di atas, dapat diketahui seberapa besar tingkat keberhasilan sistem dalam memberikan keluaran (Arifin, Saiful 2008).

3.8 Tampilan Antar Muka

3.8.1 *Form* Pelatihan

Antarmuka *form* pelatihan akan digunakan oleh *user* untuk memasukkan parameter dalam proses pelatihan. *Form* pelatihan ditunjukkan pada gambar 3.11.



Gambar 3.14 *Form* Pelatihan

Form Pelatihan terdiri dari 5 bagian, yaitu:

A : Menu parameter, dimana *user* dapat memasukkan nilai parameter *backpropagation* yang diantara adalah *Input layer, Hidden Layer, Output layer, Learning rate,* dan Maksimum iterasi.

B : Tombol untuk proses pelatihan

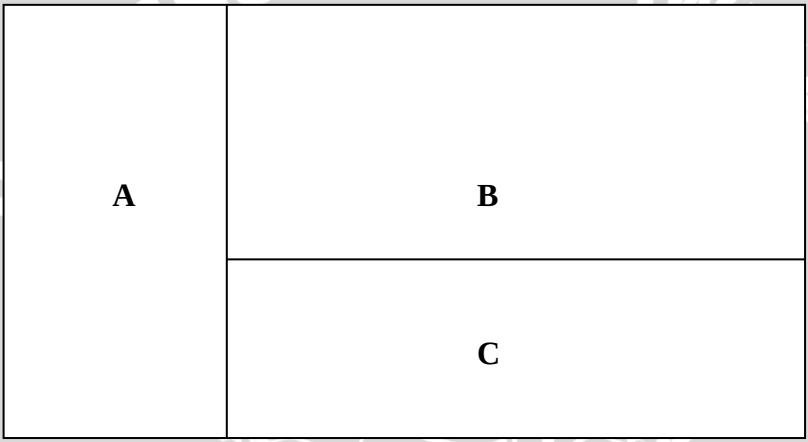
C : Tombol untuk menyimpan bobot asil pelatihan

D : Tombol untuk melakukan pengujian data yang sudah dilakukan pelatihan

E: Untuk menampilkan hasil proses pelatihan

3.8.2 *Form* Pengenalan

Form pengenalan digunakan untuk proses pengenalan yang dilakukan setelah dilakukan pelatihan pada *form* pelatihan. Form *Form* Pengenalan. *Form* pengenalan akan disajikan seperti terlihat pada gambar 3.15.



Gambar 3.15 *Form* Pengenalan

Form pengenalan terdiri dari 3 bagian, yaitu:

A : Menu untuk mengambil citra yang akan dikenali

B : Menampilakan citra yang akan dikenali

C : Menampilkan hasil pengenalan

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan untuk pengembangan sistem pengenalan nomor plat kendaraan.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem pengenalan nomor plat menggunakan Jaringan Syaraf Tiruan (JST) *Backpropagation* adalah

1. Prosesor Intel Core i3 2.27 GHz
2. Memori 2 GB
3. Harddisk dengan kapasitas 160 GB
4. Monitor 14”
5. Keyboard
6. Mouse
7. Digital Camera

4.1.2 Lingkungan Perangkat Lunak

Perangkat keras yang digunakan dalam pengembangan sistem pengenalan nomor plat menggunakan Jaringan Syaraf Tiruan (JST) *Backpropagation* ini adalah

1. Sistem Operasi Windows 7
2. Borland Delphi 7
3. Text editor Notepad
4. Ms. Access 2007

4.2 Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi dari program pengenalan nomor plat kendaraan menggunakan JST *Backpropagation*.

4.2.1 Struktur Data

Untuk memudahkan dalam implementasi jaringan syaraf tiruan, penentuan struktur data harus dilakukan dengan baik. Struktur data yang digunakan dalam sistem merupakan array 1 dimensi dan array 2 dimensi. Gambar 4.1 adalah struktur data yang digunakan dalam proses jaringan syaraf tiruan. Struktur data yang digunakan dalam metode *backpropagation* tampak pada gambar 4.6

```
const
  MaxIterasi = 63000;
  MaxUnitInput  = 20*35;
  MaxUnitHidden = 100;
  MaxUnitOutput = 15;
  MaxJumlahData = 10000;
  NTarget = 15;

var
X : array[1..MaxUnitInput ] of  Extended;
X_Data:array[1..MaxUnitInput,1..MaxJumlahData]
      of double;
T : array[1..MaxUnitOutput ] of  Extended;
T_ : array[1..MaxUnitOutput ] of  Integer;
T_Data:array[1..MaxUnitOutput,1..MaxJumlahData]
      of double;
Y_hitung:array[1..MaxUnitOutput,1..MaxJumlahData]
      Of double;
w,delta_w,w_min:array[0..MaxUnitHidden,1..MaxUnit
      Output]of double;
v,delta_v,v_min:array[0..MaxUnitInput,1..MaxUnit
      Hidden]of double;
std_min : real;
z,z_in  : array[1..MaxUnitHidden] of double;
y_in,y  : array[1..MaxUnitOutput] of double;
delta   : array[1..MaxUnitOutput] of double;
delta_in: array[1..MaxUnitHidden] of double;
deltaHidden: array[1..MaxUnitHidden] of double;
```

Gambar 4.1 Struktur data *Backpropagation*

Berikut adalah penjelasan dari struktur data yang terdapat dalam Gambar 4.1:

- MaxIterasi merupakan jumlah maksimal iterasi. Berupa ketetapan yang bernilai 63000 karena maksimal jumlah iterasi yang dilakukan pada proses pelatihan yaitu maksimal 63000.
- MaxUnitInput merupakan jumlah maksimal *neuron input*. Berupa ketetapan yang bernilai 20*35 dikarenakan maksimal jumlah input yang akan diinputkan ke dalam sistem berjumlah 20*35 neuron sesuai dengan ukuran maksimal data latih.
- MaxUnitHidden adalah jumlah maksimal *neuron hidden*. Berupa ketetapan yang bernilai 100 dikarenakan jumlah hidden neuron yang akan diinputkan maksimal 100 neuron.
- MaxUnitOutput merupakan jumlah maksimal *neuron output*. Berupa ketetapan yang bernilai 15 dikarenakan jumlah keluaran yang akan dihasilkan oleh sistem sebanyak 15 neuron.
- MaxJumlahData merupakan jumlah maksimal data yang akan diinputkan. Berupa ketetapan nilai 10000 karena maksimal jumlah data input sebanyak 10000.
- X berisi informasi mengenai nilai dari unit masukan, berupa array 1 dimensi.
- X_Data berisi informasi mengenai nilai dari unit masukan ke maksimal jumlah data, berupa array 2 dimensi.
- T dan T_ berisi informasi mengenai nilai unit keluaran, berupa array 1 dimensi.
- T_Data, Y_Hitung berisi informasi mengenai nilai dari lapisan keluaran ke maksimal jumlah data.
- w, delta_w, w_min berisi informasi mengenai nilai bobot, nilai perubahan bobot, dan bias dari lapisan tersembunyi ke lapisan keluaran.
- v,delta_v,v_min berisi informasi mengenai nilai bobot, nilai perubahan bobot dan bias dari lapisan masukan ke lapisan tersembunyi.

- z, z_in berisi informasi mengenai keluaran di lapisan tersembunyi, berupa array 1 dimensi.
- y_in, y berisi informasi mengenai keluaran di *output layer*, berupa array 1 dimensi.
- delta berisi informasi mengenai faktor kesalahan di unit keluaran.
- delta_in berisi informasi mengenai faktor kesalahan di unit tersembunyi.
- deltaHidden berisi informasi mengenai nilai aktivasi faktor kesalahan di unit tersembunyi.

4.2.3 Inisialisasi Bobot dan Bias Awal

Prosedur *InitBobot* merupakan prosedur yang bertugas untuk menginisialisasi nilai awal bobot dari *unit-unit* yang berada pada *layer input* ke *unit-unit* yang berada pada *layer hidden*, demikian juga dengan bobot dari *unit-unit* yang berada pada *layer hidden* ke *unit-unit* yang berada pada *layer output*. Bobot diinisialisasi secara acak dengan nilai berkisar antara -0,05 sampai 0,05. Prosedurnya ditunjukkan pada Gambar 4.2

```
initBobot();
for i:= 0 to JumlahInputUnit-1 do
begin
   temp := 0;
   for j:= 0 to JumlahHiddenUnit-1 do
   begin
      temp :=  temp + sqr(Zj[i,j]) ;
   end;
   Vj[i] := roundVal( sqrt(temp)) ;
end;

for i:= 0 to JumlahInputUnit-1 do
for j:= 0 to JumlahHiddenUnit-1 do
begin
    V[i,j]  := ( (Beta * Zj[i,j])/Vj[i] );
end;

for i:= 0 to JumlahHiddenUnit-1  do
for j:= 1 to JumlahOutputUnit-1 do
```

```
begin
    Randomize;
    temp :=  RandomRange(-5,5)/100;
    W[i,j] :=  (temp);
end;
```

Gambar 4.2 *Source Code* Inisialisasi Bobot dan Bias Awal

4.2.4 Fungsi Aktivasi

Fungsi aktivasi yang digunakan adalah *Binary Sigmoid* dimana merupakan fungsi yang bertugas untuk mengaktivasi tiap-tiap *unit* pada *hidden layer* dan *output layer*. *Source code* fungsi aktivasi dapat dilihat pada Gambar 4.3.

```
function Fx(x:real):real;
begin
  Fx := 1/(1+exp(-x));
end;
```

Gambar 4.3 *Source Code* Fungsi Aktivasi

4.2.5 Turunan Fungsi Aktivasi

Turunan fungsi aktivasi *Binary Sigmoid* digunakan ketika perhitungan pada proses *backward*. *Source code* turunan fungsi aktivasi dapat dilihat pada Gambar 4.4.

```
function Fx_Aksen(x:real):real;
begin
  Fx_Aksen := Fx(x)*(1-Fx(x));
end;
```

Gambar 4.4 *Source Code* Turunan Fungsi Aktivasi

4.2.6 *Feedforward*

Feedforward merupakan proses yang bertugas untuk mengaktifkan unit-unit yang berada pada layer hidden dan layer output dengan menggunakan fungsi aktivasi pada persamaan 2.2. Prosedurnya dapat dilihat pada Gambar 4.5.

```
  for j:=1 to JumlahHiddenUnit do
  begin
     z_in[j] :=v[0,j];
     for i:= 1 to JumlahInputUnit do
     begin
       Application.ProcessMessages;
       z_in[j]:=z_in[j]+x_data[i,iterasi]*v[i,j];
     end;
     z[j] := fx(z_in[j]);
  end;

  for k:=1 to JumlahOutputUnit do
  begin
    y_in[k] :=w[0,k];
    for j:=1 to JumlahHiddenUnit do
    begin
      Application.ProcessMessages;
      y_in[k]:=y_in[k]+z[j]*w[j,k];
    end;
    y_hitung[k,iterasi]:=y_in[k];
    y[k]:=fx(y_in[k]);
  end;
```

Gambar 4.5 *Source Code Feedforward*

4.2.7 Hitung *Error*

Nilai output (y) yang diperoleh dari proses *feedforward*, digunakan untuk menghitung informasi *error* pada *output layer*. Gambar 4.6 menunjukkan fungsi untuk menghitung error pada *output layer*. Nilai *error* ini digunakan untuk mendapatkan nilai koreksi bobot pada *output layer* yang nantinya digunakan untuk menghitung nilai bobot baru pada *output layer*.

```
function HitungError(Threshold:real) : boolean;
var jkw,std    : real;
    i,j        : integer;
    retval     : Boolean;
    ei         : real;
    n_1        : real;
begin
  jkw :=0.0;
```

```
for j:=1 to JumlahOutputUnit do
for i:=1 to JumlahData do
begin
  ei  := MaxOutput*(T_data[j,i]-y_hitung[j,i]);
  jkw := jkw+ei*ei;
end;
n_1 :=JumlahOutputUnit*JumlahData;
std :=(jkw/n_1);
```

Gambar 4.6 *Source Code* Hitung *Error*

4.2.8 *Backpropagation*

Proses *backward* terdiri dari perhitungan faktor kesalahan di unit keluaran, perhitungan nilai perubahan bobot lapisan tersembunyi ke lapisan keluaran, perhitungan perubahan bias lapisan tersembunyi ke lapisan keluaran, perhitungan faktor kesalahan di unit tersembunyi, perhitungan nilai perubahan bobot lapisan masukan ke lapisan tersembunyi, dan perhitungan perubahan bias lapisan masukan ke lapisan tersembunyi. Prosedur *backward* dapat dilihat pada Gambar 4.7

```
for k:=1 to JumlahOutputUnit do
  begin
    Application.ProcessMessages;
    t[k] := fx(t_data[k,iterasi]);
    delta[k] :=(t[k]-y[k])*Fx_Aksen(y_in[k]);
  end;
  for j:=1 to JumlahHiddenUnit do
  begin
    for k:=1 to JumlahOutputUnit do
    begin
      Application.ProcessMessages;
      delta_w[j,k]:=alpha*delta[k]*z[j];
    end;
  end;

  for k:=1 to JumlahOutputUnit do
  begin
    Application.ProcessMessages;
```

```
    delta_w[0,k]:=alpha*delta[k];
  end;

  for j:=1 to JumlahHiddenUnit do
  begin
    delta_in[j] :=0;
    for k:=1 to JumlahOutputUnit do
      delta_in[j]:=delta_in[j]+delta[k]*w[j,k];
    Application.ProcessMessages;
    deltaHidden[j]                                    :=
delta_in[j]*Fx_Aksen(z_in[j]);
  end;
```

Gambar 4.7 *Source Code Backward*

4.2.9 Proses Perbaharui Bobot dan Bias

Proses update bobot digunakan untuk menghitung bobot baru dari unit-unit yang berada pada *layer input* ke unit-unit yang berada pada layer hidden, dan juga bobot baru dari unit-unit yang berada pada layer hidden ke unit-unit yang berada pada layer output. Prosedur perbarui bobot dan bias dapat dilihat pada Gambar 4.8.

```
for k:=1 to JumlahOutputUnit do
  for j:=0 to JumlahHiddenUnit do
  begin
    w[j,k]:=w[j,k]+delta_w[j,k];
    Application.ProcessMessages;
  end;

for j:=1 to JumlahHiddenUnit do
  for i:=0 to JumlahInputUnit do
  begin
    v[i,j]:=v[i,j]+delta_v[i,j];
    Application.ProcessMessages;
  end;
```

Gambar 4.8 *Source Code* Perbaharui Bobot dan Bias

4.2.10 Proses Pengujian

Proses pengujian digunakan untuk menguji data pasien. Proses pengujian ini dilakukan menggunakan proses *feedforward*. *Source code* proses pengujian dapat dilihat pada Gambar 4.9.

```
for j:=1 to JumlahHiddenUnit do begin
     z_in[j] :=v_min[0,j];
     for i:= 1 to JumlahInputUnit do begin
       z_in[j]:=z_in[j]+x[i]*v_min[i,j];
     end;
     z[j] := fx(z_in[j]);
  end;

  for k:=1 to JumlahOutputUnit do begin
    y_in[k] :=w_min[0,k];
    for j:=1 to JumlahHiddenUnit do begin
      y_in[k]:=y_in[k]+z[j]*w_min[j,k];
    end;
  end;
```

Gambar 4.9 *Source Code* Pengujian

4.3 Implementasi Antarmuka

Berdasarkan rancangan antarmuka pada sub bab 3.5 maka dihasilkan antarmuka sebagai berikut:

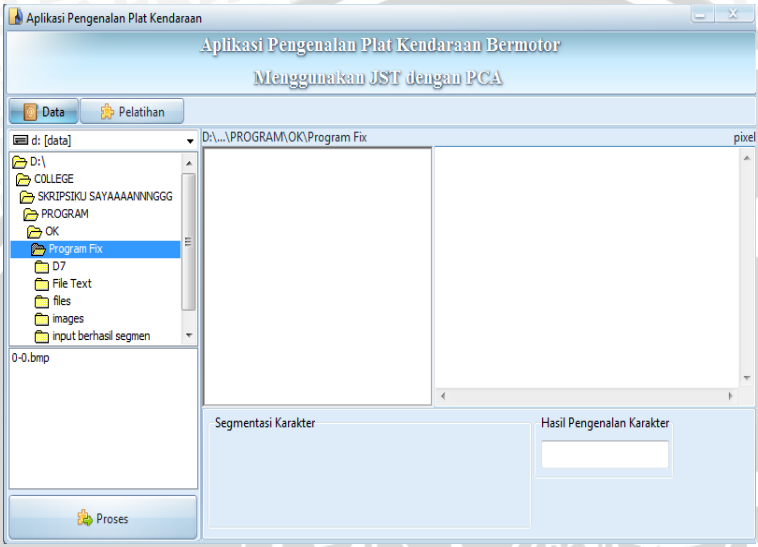
1. *Form* Utama

   Form utama terdiri dari beberapa menu yaitu *Button* Data berfungsi untuk melihat tabel data latih karakter dan table target. Pada form ini juga tersedia *button* untuk mengedit, tambah, hapus data, dan segmentasi. *Button* Pelatihan berisi *form* untuk melakukan proses pembelajaran JST. *Button* Proses berfungsi untuk proses pengenalan nomor plat dari citra plat nomor yang ingin dilakukan pengenalan. *Form* Utama dapat dilihat pada Gambar 4.10

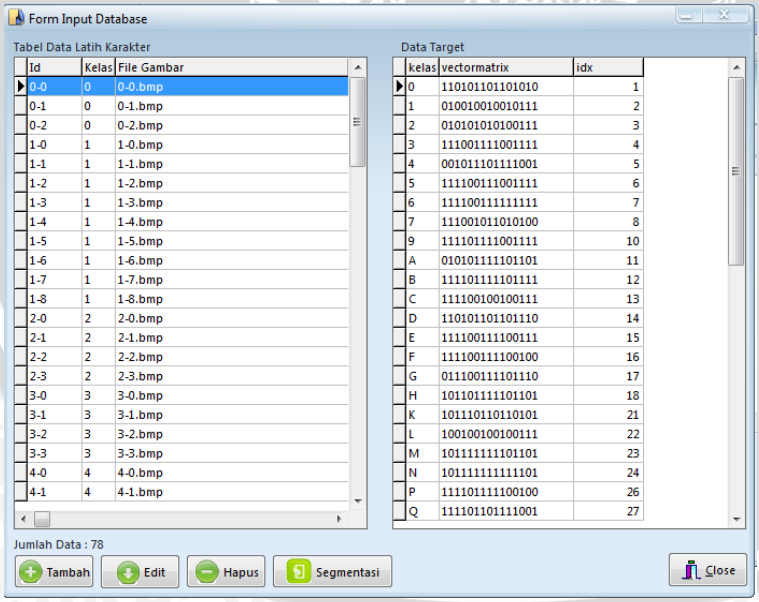
2. *Form* Input Database Karakter

   Form ini memanggil dari *file database* yang berisi tabel data latih karakter dan tabel data target. Pada *form* ini disediakan menu tambah, edit dan hapus apabila *user* ingin mengedit data dan menu segmentasi yaitu untuk melakukan segmentasi jika ada data latih

baru yang akan dilakukan pelatihan. *Form input database* dapat dilihat pada Gambar 4.11



Gambar 4.10 *Form* Utama



Gambar 4.11 *Form Input Database* Karakter

3. *Form* Segmentasi

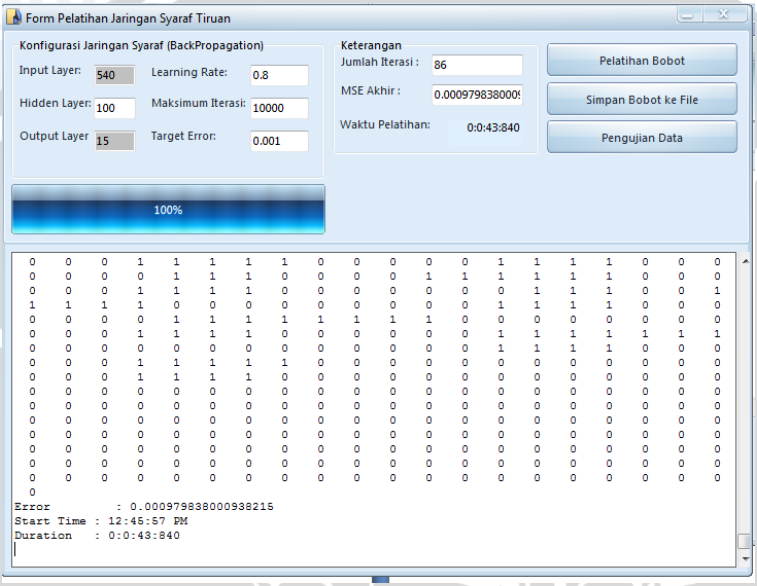
Form segmentasi yaitu berfungsi untuk memasukkan data latih baru yang akan dilakukan pelatihan. Terdapat menu *Browse Image* yang berfungsi mengambil citra yang ingin dilakukan segmentasi, citra yang dipilih akan muncul pada menu *Source Image* dan menu *Segmentation* untuk proses segmentasi dimana hasil segmentasi tersebut akan muncul pada menu *Result*. Hasil segmentasi dapat disimpan dengan meng-*klik* pada karakter hasil segmentasi untuk digunakan sebagai data latih yang kemudian dapat dimasukkan kedalam *database*. *Form* Segmentasi Karakter dapat dilihat pada Gmbar 4.12

Gambar 4.12*Form* Segmentasi Karakter

4. *Form* Pelatihan JST

Form pelatihan digunakan untuk melakukan proses *training/*pelatihan bobot menggunakan JST *backpropagation* dengan menentukan parameter-parameter seperti, *hidden layer*, *learning rate*, dan maksimum iterasi, dan target *error* yang diinginkan. *Input layer* dan *output layer* yang digunakan dalam program ini ditentukan dengan nilai 540 untuk *input layer*, nilai ini sesuaikan dengan ukuran citra data latih yang terdapat dalam *database* dan untuk *output layer* ditentukan 15 sesuai dengan panjang target dalam *database*. Setelah semua parameter terisi, maka tekan tombol Pelatihan Bobot, maka dapat dilihat bobot-bobot baru yang terbentuk dan nilai MSE akhir setelah pelatihan. Kemudian simpan bobot ke dalam file, untuk

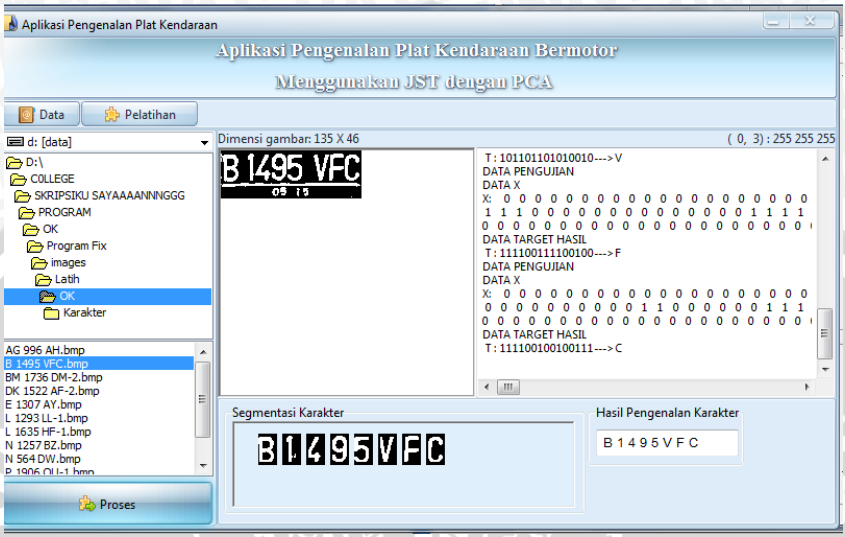
menguji data latih tekan tombol Pengujian Data lalu pilih data latih yang ingin diujikan untuk mengetahui apakah data latih tersebut bias dikenali atau tidak setelah dilakukan pelatihan. Selanjutnya lakukan proses pengenalan pada *form* Utama. *Form* Pelatihan dapat dilihat pada Gambar 4.13.



Gambar 4.13 *Form* Pelatihan

5. *Form* Pengenalan Plat

Form pengenalan merupakan *form* utama. Pada form ini *user* menginputkan citra yang ingin dikenali yang berupa citra plat nomor. Setelah memilih citra yang akan dilakukan proses pengenalan pada *file DirectoryListBox* lalu pilih citra uji yang ingin dikenali. Citra uji akan muncul pada *ScrollBox*. Setelah memilih citra uji maka tekan tombol Proses untuk proses pengenalan. Setelah menekan tombol Proses maka akan muncul data pengujian yang berupa nilai biner beserta target pengenalan pada *Memo*. Pada menu Segmentasi Karakter akan muncul citra hasil segmentasi dari citra uji. Hasil pengenalan akan muncul pada menu Hasil Pengenalan Karakter sehingga dapat diketahui citra uji tersebut dapat dikenali atau tidak pada program pengenalan ini. *Form* Pengenalan Plat dapat dilihat pada Gambar 4.14



Gambar 4.14 *Form* Pengenalan

4.4 Hasil dan Pembahasan

Untuk memperoleh struktur jaringan syaraf tiruan yang terbaik yang digunakan untuk mengenali nomor plat kendaraan maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan cara melatih jaringan syaraf tiruan dengan parameter-parameter yang berbeda, yang nantinya akan diambil satu struktur jaringan yang terbaik yang digunakan untuk melakukan pengenalan.

4.4.1 Hasil Percobaan

Percobaan dilakukan guna menentukan jumlah *hidden neuron*, *learning rate* dan target kesalahan agar didapatkan struktur jaringan syaraf tiruan yang terbaik.

4.4.1.1 Pengaruh Jumlah *Neuron Hidden Layer* Terhadap MSE

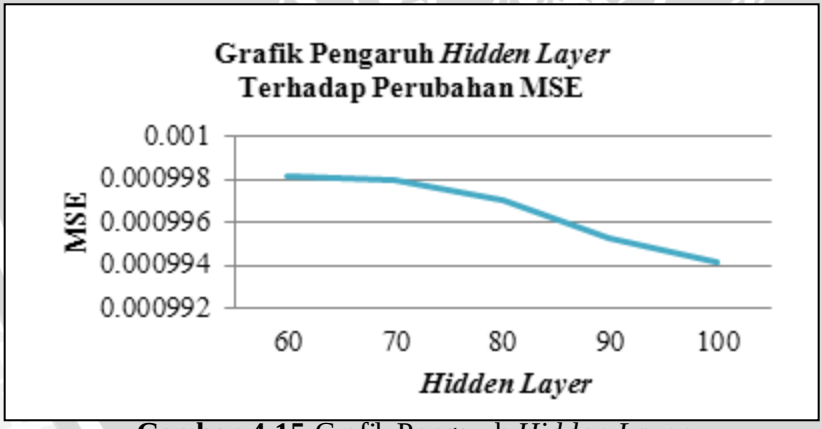
Untuk menentukan jumlah *hidden neuron* pada sistem dilakukan melalui proses *trial and error*. Dalam percobaan ini akan dilakukan percobaan dengan *learning rate* 0,1. Kemudian jumlah neuron yang diujikan yaitu 60, 70, 80, 90 dan 100. Data hasil percobaan untuk

mengetahui pengaruh jumlah unit neuron pada *hidden layer* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil percobaan pengaruh *hidden layer*

Percobaan	*Hidden Layer*	Max. Iterasi	MSE
1	60	10000	0.000998112
2	70	10000	0.000997987
3	80	10000	0.000997037
4	90	10000	0.000995246
5	100	10000	0.000994103

Berdasarkan tabel 4.1 dapat dilihat bahwa dapat dilihat bahwa nilai MSE yang dihasilkan pada masing-masing *hidden neuron* untuk tiap pelatihan selalu berubah-ubah. Dari percobaan ini diperoleh MSE yang terkecil terjadi pada percobaan ke-5, yaitu dengan jumlah *hidden layer* sebanyak 100 buah. Sehingga untuk percobaan berikutnya akan menggunakan *hidden layer* sebanyak 100 buah. Pada Gambar 4.15 disajikan grafik pengaruh *hidden layer* terhadap perubahan MSE.



Gambar 4.15 Grafik Pengaruh *Hidden Layer* Terhadap Perubahan MSE

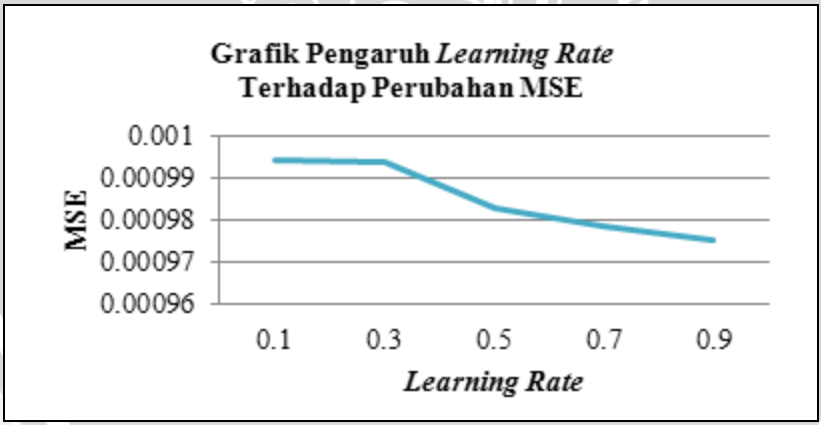
4.4.1.2 Pengaruh Nilai *Learning Rate* Terhadap MSE

Pada percobaan ini akan diuji coba beberapa nilai *learning rate* yaitu 0,1; 0,3; 0,5; 0,7 dan 0,9. Jumlah neuron *hidden* yang digunakan adalah sebanyak 100 *neuron*. Data hasil percobaan untuk mengetahui pengaruh *learning rate* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Percobaan Pengaruh *Learning Rate*

Percobaan	*Learning Rate*	Max. Iterasi	MSE
1	0.1	10000	0.000994103
2	0.3	10000	0.000993986
3	0.5	10000	0.0009829
4	0.7	10000	0.000978376
5	0.9	10000	0.000975482

Berdasarkan tabel 4.2 dapat dilihat bahwa MSE terkecil yang dihasilkan pada masing-masing nilai *learning rate* untuk tiap pelatihan selalu berubah-ubah. Dari percobaan terhadap nilai *learning rate* diperoleh nilai MSE paling kecil terdapat pada percobaan ke-5. Pada Gambar 4.16 disajikan grafik pengaruh *learning rate* terhadap perubahan MSE.



Gambar 4.16 Grafik Pengaruh *Learning Rate* Terhadap Perubahan MSE

4.5 Hasil Pengujian

4.5.1 Hasil Pengujian Segmentasi Karakter

Sebelum melakukan proses pengenalan, terlebih dahulu dilakukan proses pemisahan (segmentasi) karakter. Tujuan dari proses segmentasi ini adalah mendapatkan citra karakter dari nomor plat. Kondisi pemisahan benar merupakan kondisi dimana semua karakter pada nomor plat dapat disegmentasi dengan benar.

Tabel 4.3 Hasil Pengujian Segmentasi Karakter

Jumlah Citra Nomor Plat	Benar	Salah	Keakuratan
100	63	37	63%

Dari tabel 4.3 didapatkan hasil keakuratan proses segmentasi karakter pada 100 citra nomor plat yaitu sebesar 63 % yang didapat dari pengujian sebanyak 63 citra nomor plat yang berhasil disegmentasi dengan benar dan 37 citra nomor plat tidak dapat disegmentasi dengan benar.

4.5.2 Hasil Pengujian Data Latih

Untuk memperoleh struktur jaringan syaraf tiruan yang terbaik yang digunakan untuk mengenali nomor plat maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan cara melatih jaringan syaraf tiruan dengan parameter-parameter yang berbeda.

Pelatihan jaringan syaraf tiruan ini menggunakan 162 citra latih dari hasil segmentasi karakter nomor plat. Kondisi benar merupakan kondisi dimana pengujian citra latih ini dapat dikenali dengan benar sesuai dengan citra karakter nomor plat yang diujikan, sebaliknya kondisi salah jika proses tidak bisa mengenali citra karakter nomor plat yang diujikan dari citra latih tersebut. Tabel 4.4 berikut merupakan tabel hasil pengujian terhadap data latih.

Tabel 4.4 Hasil Pengujian Citra
yang pernah dilakukan Pembelajaran

Jumlah Karakter	Benar	Salah	Keakuratan
162	162	0	100%

Berdasarkan tabel 4.4 dapat dilihat bahwa hasil pengujian terhadap citra karakter nomor plat yang sudah pernah dilakukan pembelajaran, sistem mampu mengenali dengan benar semua citra yang diujikan dengan tingkat keakuratan sebesar 100%. Sedangkan untuk citra karakter nomor plat yang belum dilakukan pembelajaran dapat dilihat pada tabel 4.5.

4.5.2 Hasil Pengenalan

Berikut ini dilakukan pengujian pengenalan citra karakter nomor plat terhadap citra yang belum pernah dilakukan pembelajaran menggunakan arsitektur JST yang menghasilkan MSE terkecil yaitu dengan parameter nilai *learning rate* 0.9 dan *hidden layer* sebanyak 100 *neuron*. Citra karakter nomor plat yang digunakan untuk data uji terdiri dari 50 citra nomor plat dengan jumlah karakter nomor plat sebanyak 347 karakter. Hasil pengujian data uji ini dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil pengujian citra
(*hidden layer* = 100, *learning rate* = 0.9)

Jumlah Karakter	Benar	Salah	Keakuratan
347	181	166	52%

Dapat dilihat pada tabel 4.5 hasil pengujian citra yang belum pernah dilakukan pembelajaran menghasilkan tingkat akurasi sebesar 44.36%. Dari citra yang diujikan sebanyak 408 citra karakter nomor plat, yang berhasil dikenali dengan benar yaitu sebanyak 181 citra sedangkan 277 citra lainnya tidak dapat dikenali.

4.6 Analisa Hasil

4.6.1 Analisa Hasil Segmentasi

Dari hasil pengujian segmentasi karakter pada tabel 4.3, dapat dilihat bahwa hasil keakuratan sistem dalam melakukan proses segmentasi yaitu sebesar 63%. Pada proses segmentasi ini terjadi kesalahan sebanyak 37 citra dari 100 citra percobaan. Kesalahan pada proses segmentasi ini disebabkan oleh beberapa hal. Salah satu penyebab kesalahan pada proses segmentasi ini yaitu kesalahan pada pemotongan daerah yang bukan termasuk karakter, misalnya terdapat *noise* sehingga *noise* tersebut dinggap sebagai karakter karena memiliki nilai *threshold* yang lebih kecil dari nilai putihnya. Contoh kesalahan segmentasi ini dapat dilihat pada gambar 4.17

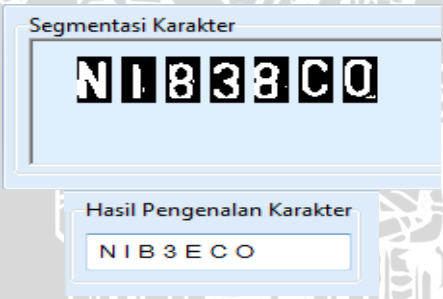
Gambar 4.17 Contoh Kesalahan Segmentasi yang disebabkan oleh pemotongan daerah yang bukan termasuk karakter

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa proses segmentasi berhasil dilakukan tetapi belum optimal. Oleh karena itu, diperlukan perbaikan pada proses segmentasi dengan menambah atau menggabungkan metode lain yang mungkin dapat memperbaiki proses segmentasi pada karakter nomor plat sehingga menghasilkan tingkat keberhasilan yang optimal.

4.6.2 Analisa Hasil Pengenalan Berdasarkan Struktur Jaringan *Backpropagation*

Berdasarkan hasil pelatihan struktur jaringan yang optimal dengan nilai *error* terkecil yaitu dengan parameter *hidden layer* sebanyak 100 *neuron* dan *learning rate* 0.9. Selanjutnya parameter tersebut digunakan untuk menguji kemampuan sistem dalam mengenali karakter nomor plat kendaraan. Kumpulan data uji terdiri dari 50 citra nomor plat kendaraan.

Hasil pengujian sistem dalam mengenali karakter nomor plat terhadap data latih adalah sebesar 100%. Sedangkan hasil pengenalan sistem terhadap data baru yaitu sebesar 52%. Kegagalan sistem dalam mengenali keseluruhan data uji baru disebabkan oleh beberapa hal. Salah satu penyebab kesalahan pada pengenalan karakter ini yaitu adanya karakter yang memiliki kemiripan bentuk antar karakter. Misalkan karakter angka 1 dan I, 8 dan B dimana angka 1 dikenali sebagai I atau sebaliknya dan 8 dikenali sebagai B dan sebagainya. Contoh kesalahan pengenalan karena kemiripan bentuk karakter dapat dilihat pada gambar 4.18



Gambar 4.18 Contoh kesalahan pengenalan karena kemiripan bentuk karakter

Kesalahan pada pengenalan karakter juga dapat terjadi akibat citra karakter yang tidak jelas atau adanya bagian karakter yang hilang ataupun karena adanya *noise* pada citra. Hal ini disebabkan nilai kedekatan antara karakter bobot lain dengan karakter yang sesungguhnya sehingga menyebabkan kesalahan pada pengenalan karakter. Kesalahan pengenalan juga dipengaruhi oleh kondisi dari nomor plat dengan garis karakter yang tipis sehingga karakter sulit

dikenali. Selain itu, faktor pencahayaan ketika melakukan pengambilan citra juga berpengaruh sehingga pada proses pengolahan citra yang mengakibatkan citra tidak jelas ketika proses segmentasi karakter. Segmentasi karakter juga berpengaruh terhadap tingkat keberhasilan pengenalan karakter sehingga semakin jelas gambar karakter hasil segmentasi maka hasil pengenalan karakter akan lebih baik.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan percobaan dan analisa pada sistem, maka dapat disimpulkan sebagai berikut:

1. Arsitektur jaringan syaraf tiruan terbaik yang terbentuk terdiri dari 540 unit *neuron input layer*, 100 unit *neuron hidden layer*, 15 unit *neuron output layer*, nilai *learning rate* sebesar 0,9 dengan maksimal iterasi sebanyak 10000 menghasilkan MSE terkecil yaitu 0.000975482.

2. Didapatkan hasil pengujian segmentasi sebesar 63% dari 100 citra uji dan tingkat keakuratan rata-rata untuk pengenalan nomor plat sebesar 52% dari 50 citra yang berhasil di segmentasi.

5.2. Saran

Saran yang dapat diberikan setelah melakukan pengujian pada sistem yang telah dibuat adalah :

1. Pada penelitian selanjutnya dapat ditambahkan citra latih dengan kelas yang lebih bervariasi sehingga sistem dapat melakukan pengenalan karakter dengan lebih baik.
2. Metode segmentasi karakter dapat diperbaiki sehingga dapat melakukan segmentasi karakter dengan benar.
3. Penggabungan metode dapat diterapkan untuk menghasilkan pengenalan nomor plat dengan tingkat keakuratan yang lebih tinggi.


UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Adipranata, Rudy., dkk.2005.” Perancangan dan Pembuatan Aplikasi Segmentasi Gambar dengan Menggunakan Metode Morphological Watershed”.Surabaya: Universitas Kristen Petra. [PDF]
Tanggal Akses: 17 Juni 2011

Ardisasmita, M. Syamsa. 2000.”Metoda Segmentasi Citra Resonansi Magnetik Otak Menggunakan Sistem Pengkodean Neuronfuzzy”. [PDF]
http://www.batan.go.id/ppin/lokakarya/LKSTN_10/Syamsa.pdf
Tanggal Akses: 18 Agustus 2011

Arifin, Saiful. 2008. “Pengenalan Tulisan Tangan Menggunakan Algoritma Jaringan Syaraf Tiruan Propagasi Balik (Backpropagation)”**.** Malang : Universitas Brawijaya

Balza, Achmad dan Kartika Firdausy. 2005.”Teknik Pengolahan Citra Digital Menggunakan Delphi”. Yogyakarta : PT. Mitra Angkasa Mulia

Gonzales, Rafael C and Wintz, Paul.1987.”Digital Image Processing Second Edition”.Addison-Wesley Publishing

Iswanto,Noerdityo., dkk.2010. “Desain dan Implmentasi *Color Code* untuk Verifikasi Nomor Kendaraan Bermotor pada Sistem Parkir”.Bandung : Institut Teknologi Telkom. [PDF]
http://openstorage.gunadarma.ac.id/~mwiryana/KOMMIT/per-artikel/01-03-006-Desain%5Biswanto%5D.pdf
Tanggal Akses: 15 Juni 2011

Kartika, Gunadi dan Sonny Reinard Pongsitanan.2000.”Pembuatan Perangkat Lunak Pengenalan Wajah Menggunakan *Principal Components Analysis”*.[PDF]
http://puslit.petra.ac.id/journals/informatics/
Tanggal akses : 19 Oktober 2008

Kristanto, Andri.2004."Jaringan Syaraf Tiruan: Teori dan Aplikasi".Yogyakarta:Andi

Kusumadewi, S. 2003. "Artificial Intelligence (Teknik dan Aplikasi) ".Yogyakarta : Graha Ilmu

Kusumoputro, B., dkk. "Pengenalan Huruf Tulisan Tangan Menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan".[PDF]
http://www.std.ryu.titech.ac.jp/~indonesia/tokodai/zoa/pdf/zoawidyanto.pdf.
Tanggal Akses: 19 Oktober 2008

Naim, Afwan Badru. 2009. "Segmentasi Citra Dengan Menggunakan Metode Mean Shift Spectral Clustering".Surabaya : Institut Teknologi Sepuluh November.[PDF]
Skripsi
Tanggal Akses : 18 Agustus 2011

Nugroho, A.P. dan A.B. Mutiara.2002."Metode Ekstraksi Data untuk Pengenalan Huruf dan Angka Tulisan dengan manggunakan Jaringan Syaraf Buatan Propagasi Balik".[PDF]
http://www.gunadarma.ac.id
Tanggal akses : 29 Oktober 2008

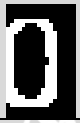
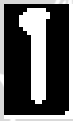
Resmana, Liem,dkk,2003."Sistem Pengenalan Plat Nomor Kendaraan menggunakan Metode Principal Component Analysis".[PDF]
http://puslit.petra.ac.id/journals/electrical/
Tanggal Akses: 23 September 2008

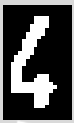
Siang, JJ.2005.Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab.Yogyakarta:Andi

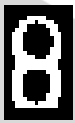
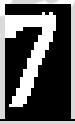
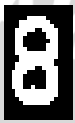
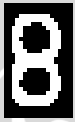
Widyadi Setiawan dan Sri Andriati Asri.2005."Aplikasi Jaringan Syaraf Tiruan Perambatan Balik pada Pengenalan Angka Tulisan Tangan".[PDF]
http://ejoumal.unud.ac.id/abstrak/widiadi%202.pdf
Tanggal Akses : 02 Desember 2008

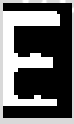
LAMPIRAN

Lampiran 1. Data Latih

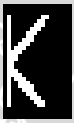
Id Karakter	Gambar Karakter		Id Karakter	Gambar Karakter
0-1			1-2	
0-2			1-3	
0-3			1-4	
0-4			1-5	
0-5			2-1	
1-1			2-2	
2-3			3-5	

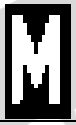
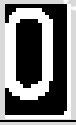
2-4			**4-1**	
2-5			**4-2**	
3-1			**4-3**	
3-2			**4-4**	
3-3			**4-5**	
3-4			**5-1**	
6-1			**7-3**	
6-2			**7-4**	
6-3			**7-5**	

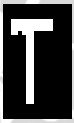
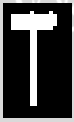
6-4			**8-1**	
6-5			**8-2**	
7-1			**8-3**	
7-2			**8-4**	
8-5			**A-2**	
9-1			**A-3**	
9-2			**A-4**	
9-3			**A-5**	
9-4			**B-1**	

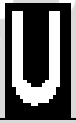
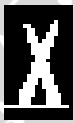
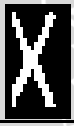
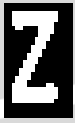
9-5			**B-2**	
A-1			**B-3**	
B-4			**D-1**	
B-5			**D-2**	
C-1			**D-3**	
C-2			**D-4**	
C-3			**D-5**	
C-4			**E-1**	
C-5			**E-2**	

E-3			**F-5**	
E-4			**G-1**	
E-5			**G-2**	
F-1			**G-3**	
F-2			**H-1**	
F-3			**H-2**	
F-4			**H-3**	
H-4			**J-3**	
H-5			**J-4**	

I-1			**K-1**	
I-2			**K-2**	
I-3			**K-3**	
J-1			**K-4**	
J-2			**K-5**	
L-1			**M-3**	
L-2			**M-4**	
L-3			**M-5**	
L-4			**N-1**	

L-5			**N-2**	
M-1			**N-3**	
M-2			**N-4**	
N-5			**P-2**	
O-1			**P-3**	
O-2			**P-4**	
O-3			**Q-1**	
O-4			**Q-2**	
O-5			**Q3**	

P-1			**Q-4**	
Q-5			**S-2**	
R-1			**S-3**	
R-2			**S-4**	
R-3			**S-5**	
R-4			**T-1**	
R-5			**T-2**	
S-1			**T-3**	
U-1			**V-3**	

U-2			**W-1**	
U-3			**W-2**	
U-4			**W-3**	
U-5			**W-4**	
V-1			**W-5**	
V-2			**X-1**	
X-2			**Z-1**	
Y-1			**Z-2**	
Y-2			**Z-3**	

Lampiran 2. Data Uji

No.	Nama File	Jumlah Karakter
1	AB 1088 PB.bmp	8
2	AB 8287 SK.bmp	8
3	AD 9430 QB.bmp	8
4	AG 996 AH.bmp	7
5	B 207 DX.bmp	6
6	B 503 UNI-1.bmp	7
7	B 503 UNI-2.bmp	7
8	B 503 UNI-3.bmp	7
9	B 1057 TKI.bmp	8
10	B 1362 SFB.bmp	8
11	B 1495 VFC.bmp	8
12	B 8044 NR.bmp	7
13	B 8070 LJ.bmp	7
14	B 8103 DU.bmp	7
15	B 8498 ND.bmp	7
16	BM 1736 DM-1.bmp	8
17	BM 1736 DM-2.bmp	8
18	DK 1522 AF-1.bmp	8
19	DK 1522 AF-2.bmp	8
20	DK 1522 AF-3.bmp	8
21	E 1142 E.bmp	6
22	E 1194 AJ.bmp	7
23	E 1307 AY.bmp	7
24	E 1565 AO-1.bmp	7
25	E 1565 AO-2.bmp	7
26	E 1711 KL.bmp	7
27	E 1876 AO.bmp	7
28	E 69 HY.bmp	5
29	L 1293 LL-1.bmp	7

30	L 1293 LL-2.bmp	7
31	L 1635 HF-1.bmp	7
32	L 1635 HF-2.bmp	7
33	L 1635 HF-3.bmp	7
34	L 1662 BE.bmp	7
35	L 1739 AA.bmp	7
36	N 1037 BQ.bmp	7
37	N 1066 XC.bmp	7
38	N 1083 CU-1.bmp	7
39	N 1083 CU-2.bmp	7
40	N 1083 CU-3.bmp	7
41	N 1084 AH.bmp	7
42	N 1091 CT.bmp	7
43	N 1128 BN .bmp	7
44	N 1156 CM.bmp	7
45	N 1174 CG-1.bmp	7
46	N 1174 CG-2.bmp	7
47	N 1190 CR.bmp	7
48	N 1199 CG.bmp	7
49	N 1204 CO.bmp	7
50	N 1211 CF-1.bmp	7
51	N 1211 CF-2.bmp	7
52	N 1216 BO.bmp	7
53	N 1235 DN-1.bmp	7
54	N 1235 DN-2.bmp	7
55	N 1257 BZ.bmp	7
56	N 1344 AC-1.bmp	7
57	N 1344 AC-2.bmp	7
58	N 1344 AC-3.bmp	7
59	N 1365 KB.bmp	7
60	N 1368 CI-1.bmp	7
61	N 1368 CI-2.bmp	7

62	N 1604 CH.bmp	7
63	N 1628 AO.bmp	7
64	N 1709 AF-1.bmp	7
65	N 1709 AF-2.bmp	7
66	N 1730 GC.bmp	7
67	N 1792 CT.bmp	7
68	N 1822 CW-1.bmp	7
69	N 1822 CW-2.bmp	7
70	N 1838 CO-1.bmp	7
71	N 1838 CO-2.bmp	7
72	N 1838 CO-3.bmp	7
73	N 1884 CS.bmp	7
74	N 1886 BO-1.bmp	7
75	N 1886 BO-2.bmp	7
76	N 1889 AN.bmp	7
77	N 1990 CR-1.bmp	7
78	N 1990 CR-2.bmp	7
79	N 439 CE.bmp	6
80	N 481 CZ-1.bmp	6
81	N 481 CZ-2.bmp	6
82	N 502 AK-1.bmp	6
83	N 502 AK-2.bmp	6
84	N 502 AK-3.bmp	6
85	N 503 CV.bmp	6
86	N 544 DO.bmp	6
87	N 564 DW.bmp	6
88	N 722 AL-1.bmp	6
89	N 722 AL-2.bmp	6
90	N 722 AL-3.bmp	6
91	N 732 AQ.bmp	6
92	N 840 KD.bmp	6
93	N 8570 CJ.bmp	7

94	P 1906 QU-1.bmp	7
95	P 1906 QU-2.bmp	7
96	S 1056 P.bmp	6
97	S 1200 WC.bmp	7
98	S 329 S.bmp	5
99	S 610 WC.bmp	6
100	W 1239 NQ.bmp	7

Lampiran 3. Hasil Uji Segmentasi

No	Nama File	Benar (B)	Salah (B)
1	AB 1088 PB.bmp	B	
2	AB 8287 SK.bmp	B	
3	AD 9430 QB.bmp		S
4	AG 996 AH.bmp		S
5	B 207 DX.bmp	B	
6	B 503 UNI-1.bmp		S
7	B 503 UNI-2.bmp		S
8	B 503 UNI-3.bmp		S
9	B 1057 TKI.bmp		S
10	B 1362 SFB.bmp	B	
11	B 1495 VFC.bmp	B	
12	B 8044 NR.bmp	B	
13	B 8070 LJ.bmp	B	
14	B 8103 DU.bmp	B	
15	B 8498 ND.bmp		S
16	BM 1736 DM-1.bmp	B	
17	BM 1736 DM-2.bmp	B	
18	DK 1522 AF-1.bmp	B	
19	DK 1522 AF-2.bmp	B	
20	DK 1522 AF-3.bmp		S
21	E 1142 E.bmp	B	
22	E 1194 AJ.bmp		S
23	E 1307 AY.bmp	B	
24	E 1565 AO-1.bmp	B	
25	E 1565 AO-2.bmp	B	
26	E 1711 KL.bmp		S
27	E 1876 AO.bmp	B	
28	E 69 HY.bmp	B	
29	L 1293 LL-1.bmp	B	

30	L 1293 LL-2.bmp	B	
31	L 1635 HF-1.bmp	B	
32	L 1635 HF-2.bmp	B	
33	L 1635 HF-3.bmp		S
34	L 1662 BE.bmp	B	
35	L 1739 AA.bmp		S
36	N 1037 BQ.bmp	B	
37	N 1066 XC.bmp	B	
38	N 1083 CU-1.bmp		S
39	N 1083 CU-2.bmp	B	
40	N 1083 CU-3.bmp	B	
41	N 1084 AH.bmp	B	
42	N 1091 CT.bmp		S
43	N 1128 BN .bmp		S
44	N 1156 CM.bmp		S
45	N 1174 CG-1.bmp		S
46	N 1174 CG-2.bmp		S
47	N 1190 CR.bmp		S
48	N 1199 CG.bmp	B	
49	N 1204 CO.bmp	B	
50	N 1211 CF-1.bmp		S
51	N 1211 CF-2.bmp		S
52	N 1216 BO.bmp	B	
53	N 1235 DN-1.bmp		S
54	N 1235 DN-2.bmp		S
55	N 1257 BZ.bmp	B	
56	N 1344 AC-1.bmp	B	
57	N 1344 AC-2.bmp	B	
58	N 1344 AC-3.bmp		S
59	N 1365 KB.bmp		S
60	N 1368 CI-1.bmp	B	
61	N 1368 CI-2.bmp	B	

62	N 1604 CH.bmp	B	
63	N 1628 AO.bmp		S
64	N 1709 AF-1.bmp		S
65	N 1709 AF-2.bmp		S
66	N 1730 GC.bmp		S
67	N 1792 CT.bmp		S
68	N 1822 CW-1.bmp	B	
69	N 1822 CW-2.bmp	B	
70	N 1838 CO-1.bmp	B	
71	N 1838 CO-2.bmp	B	
72	N 1838 CO-3.bmp	B	
73	N 1884 CS.bmp	B	
74	N 1886 BO-1.bmp	B	
75	N 1886 BO-2.bmp	B	
76	N 1889 AN.bmp	B	
77	N 1990 CR-1.bmp	B	
78	N 1990 CR-2.bmp	B	
79	N 439 CE.bmp	B	
80	N 481 CZ-1.bmp		S
81	N 481 CZ-2.bmp	B	
82	N 502 AK-1.bmp		S
83	N 502 AK-2.bmp		S
84	N 502 AK-3.bmp	B	
85	N 503 CV.bmp	B	
86	N 544 DO.bmp	B	
87	N 564 DW.bmp	B	
88	N 722 AL-1.bmp	B	
89	N 722 AL-2.bmp		S
90	N 722 AL-3.bmp		S
91	N 732 AQ.bmp		S
92	N 840 KD.bmp	B	
93	N 8570 CJ.bmp	B	

94	P 1906 QU-1.bmp	B	
95	P 1906 QU-2.bmp	B	
96	S 1056 P.bmp		S
97	S 1200 WC.bmp	B	
98	S 329 S.bmp	B	
99	S 610 WC.bmp	B	
100	W 1239 NQ.bmp	B	
	TOTAL	**63**	**37**

Lampiran 4 Hasil Pengenalan (*Learning Rate* = 0.9, *Hidden Layer* =100)

Nama File	Jumlah Karakter	Benar	Salah	Keakuratan
AB 1088 PB.bmp	8	4	4	50%
AB 8287 SK.bmp	8	5	3	63%
B 207 DX.bmp	6	2	4	33%
B 1362 SFB.bmp	8	8	0	100%
B 1495 VFC.bmp	8	8	0	100%
B 8044 NR.bmp	7	4	3	57%
B 8070 LJ.bmp	7	3	4	43%
B 8103 DU.bmp	7	3	4	43%
BM 1736 DM-1.bmp	8	5	3	63%
BM 1736 DM-2.bmp	8	6	2	75%
DK 1522 AF-1.bmp	8	4	4	50%
DK 1522 AF-2.bmp	8	7	1	88%
E 1142 E.bmp	6	5	1	83%
E 1307 AY.bmp	7	3	4	43%
E 1876 AO.bmp	7	2	5	29%
E 69 HY.bmp	5	3	2	60%
L 1293 LL-1.bmp	7	5	2	71%
L 1293 LL-2.bmp	7	5	2	71%
L 1635 HF-1.bmp	7	3	4	43%
L 1635 HF-2.bmp	7	2	5	29%
L 1635 HF-3.bmp	7	3	4	43%
L 1662 BE.bmp	7	3	4	43%
N 1066 XC.bmp	7	2	5	29%
N 1083 CU-2.bmp	7	5	2	71%
N 1083 CU-3.bmp	7	3	4	43%
N 1084 AH.bmp	7	2	5	29%
N 1091 CT.bmp	7	2	5	29%
N 1199 CG.bmp	7	2	5	29%

N 1204 CO.bmp	7	2	5	29%
N 1216 BO.bmp	7	3	4	43%
N 1257 BZ.bmp	7	4	3	57%
N 1344 AC-2.bmp	7	2	5	29%
N 1838 CO-1.bmp	7	4	3	57%
N 1838 CO-2.bmp	7	3	4	43%
N 1838 CO-3.bmp	7	4	3	57%
N 1884 CS.bmp	7	2	5	29%
N 1886 BO-1.bmp	7	2	5	29%
N 1889 AN.bmp	7	4	3	57%
N 439 CE.bmp	6	1	5	17%
N 502 AK-3.bmp	6	3	3	50%
N 503 CV.bmp	6	3	3	50%
N 544 DO.bmp	6	5	1	83%
N 564 DW.bmp	6	4	2	67%
N 722 AL-1.bmp	6	3	3	50%
N 8570 CJ.bmp	7	4	3	57%
P 1906 QU-1.bmp	7	6	1	86%
P 1906 QU-2.bmp	7	5	2	71%
S 1200 WC.bmp	7	2	5	29%
S 610 WC.bmp	6	3	3	50%
W 1239 NQ.bmp	7	3	4	43%
TOTAL	**347**	**181**	**166**	**52%**