

**Pengenalan Tulisan Tangan Bersambung
Menggunakan Metode *Optical Backpropagation***

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

IKHWAN EKO SETIAWAN

NIM. 0810960049

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG**

2013

LEMBAR PERSETUJUAN
Pengenalan Tulisan Tangan Bersambung
Menggunakan Metode *Optical Backpropagation*

SKRIPSI



Disusun oleh :
IKHWAN EKO SETIAWAN
NIM. 0810960049

Telah diperiksa dan disetujui oleh :

Pembimbing I,

Pembimbing II,

Drs. Achmad Ridok, M.Kom
NIP. 196808251994031002

Drs. Muh. Arif Rahman, M.Kom
NIP. 196604231991111001

LEMBAR PENGESAHAN
Pengenalan Tulisan Tangan Bersambung
Menggunakan Metode *Optical Backpropagation*

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :
IKHWAN EKO SETIAWAN
NIM. 0810960049

Skripsi ini telah diuji dan dinyatakan lulus tanggal 22 Januari 2013

Penguji I

Penguji II

Drs. Marji, M.T.
NIP. 196708011992031001

Wijaya Kurniawan, S.T., M.T.

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom

Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, MT.
NIP. 196708011992031001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, Januari 2013

Mahasiswa,

Ikhwan Eko Setiawan

NIM 0810960049

KATA PENGANTAR

Puji syukur Penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul: **“Pengenalan Tulisan Tangan Bersambung Menggunakan Metode *Optical Backpropagation*”**.

Skripsi ini diajukan sebagai syarat ujian seminar skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi & Ilmu Komputer (PTIIK), Program Studi Teknik Informatika, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Drs. Achmad Ridok, M.Kom. selaku Dosen Pembimbing Skripsi I.
2. Drs. Muh. Arif Rahman, M.Kom. selaku Dosen Pembimbing Skripsi II.
3. Candra Dewi, S.Kom., M.Sc. selaku Dosen Penasehat Akademik.
4. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
5. Drs. Marji, MT. selaku Ketua Program Studi Informatika / Ilmu Komputer Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
7. Segenap staf dan karyawan di Program Studi Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Orang tua Penulis, adek Luqman, dan adek Hilal atas segala dukungan materi dan doa restunya kepada Penulis.
9. Seluruh Civitas Akademika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

10. Fitria Handayani Pamungkas, A.Md. Keb. beserta keluarga, yang telah memberi dukungan baik dalam bentuk material maupun non material demi terselesainya skripsi ini.
11. Segenap keluarga besar PPTI UB yang telah banyak memberikan semangat dan doa.
12. Teman-teman Ilmu Komputer angkatan 2008 tercinta yang telah banyak memberikan bantuan, semangat, dan doa, serta pengalaman selama menjadi mahasiswa di Universitas Brawijaya.
13. Teman-teman KKN Desa Krisik 2012 tercinta yang telah banyak memberikan semangat dan doa.
14. Semua pihak yang telah membantu terselesainya skripsi ini yang tidak dapat kami sebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari sempurna, karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Maka, saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Semoga skripsi ini dapat bermanfaat dan berguna bagi semua pihak, baik penulis maupun pembaca, dan semoga Allah SWT meridhoi dan dicatat sebagai ibadah. Amin.

Malang, Januari 2013

Ikhwan Eko Setiawan
Penulis

ABSTRAK

Ikhwan Eko Setiawan. 2013. Pengenalan Tulisan Tangan Bersambung Menggunakan Metode *Optical Backpropagation*.

Dosen Pembimbing : Drs. Achmad Ridok, M.Kom. dan Drs. Muh. Arif Rahman, M.Kom.

Jaringan syaraf tiruan telah banyak digunakan dalam berbagai permasalahan, seperti klasifikasi, peramalan, pengenalan, dan lain sebagainya. Dalam penelitian ini menggunakan metode dalam jaringan syaraf tiruan, yaitu *Optical Backpropagation* yang diterapkan pada kasus pengenalan tulisan tangan bersambung. Segmentasi karakter memegang peranan penting dalam pengenalan tulisan tangan bersambung. Jaringan syaraf tiruan digunakan untuk proses pengenalan yang sekaligus membantu dalam penentuan segmentasi karakter yang sesuai. Adapun *pre-processing* yang dilakukan yaitu *binarization*, *thinning*, dan *core-zone estimation*. Dari sejumlah percobaan yang dilakukan didapatkan arsitektur jaringan dengan jumlah *hidden unit* 90 dan *learning rate* 0.3 yang menghasilkan error pelatihan sebesar 4.68%. Adapun hasil pengenalan didapatkan rata-rata hasil segmentasi sebesar 61.52% dan rata-rata hasil pengenalan sebesar 52.50%.

Kata Kunci : Jaringan Syaraf Tiruan, *Optical Backpropagation*, Tulisan tangan bersambung

ABSTRACT

Ikhwan Eko Setiawan. 2013. *Cursive Handwriting Recognition Using Optical Backpropagation*.

Advisor : Drs. Achmad Ridok, M.Kom. dan Drs. Muh. Arif Rahman, M.Kom.

Artificial Neural networks have been widely used in a variety of issues, such as classification, forecasting, recognition, and so forth. This study uses the method in artificial neural networks, the Optical Backpropagation that applied to the case of cursive handwriting recognition. Segmentation of characters plays an important role in cursive handwriting recognition. Neural network is used for recognition process that also helps in determining the appropriate character segmentation. The used pre-processings are binarization, thinning, and the core-zone estimation. From a number of experiments, this study found that network architectures contained 90 hidden units and 0.3 learning rate that generated 4.68% training error. The recognition results obtained 61.52% segmentation average and 52.50% recognition average.

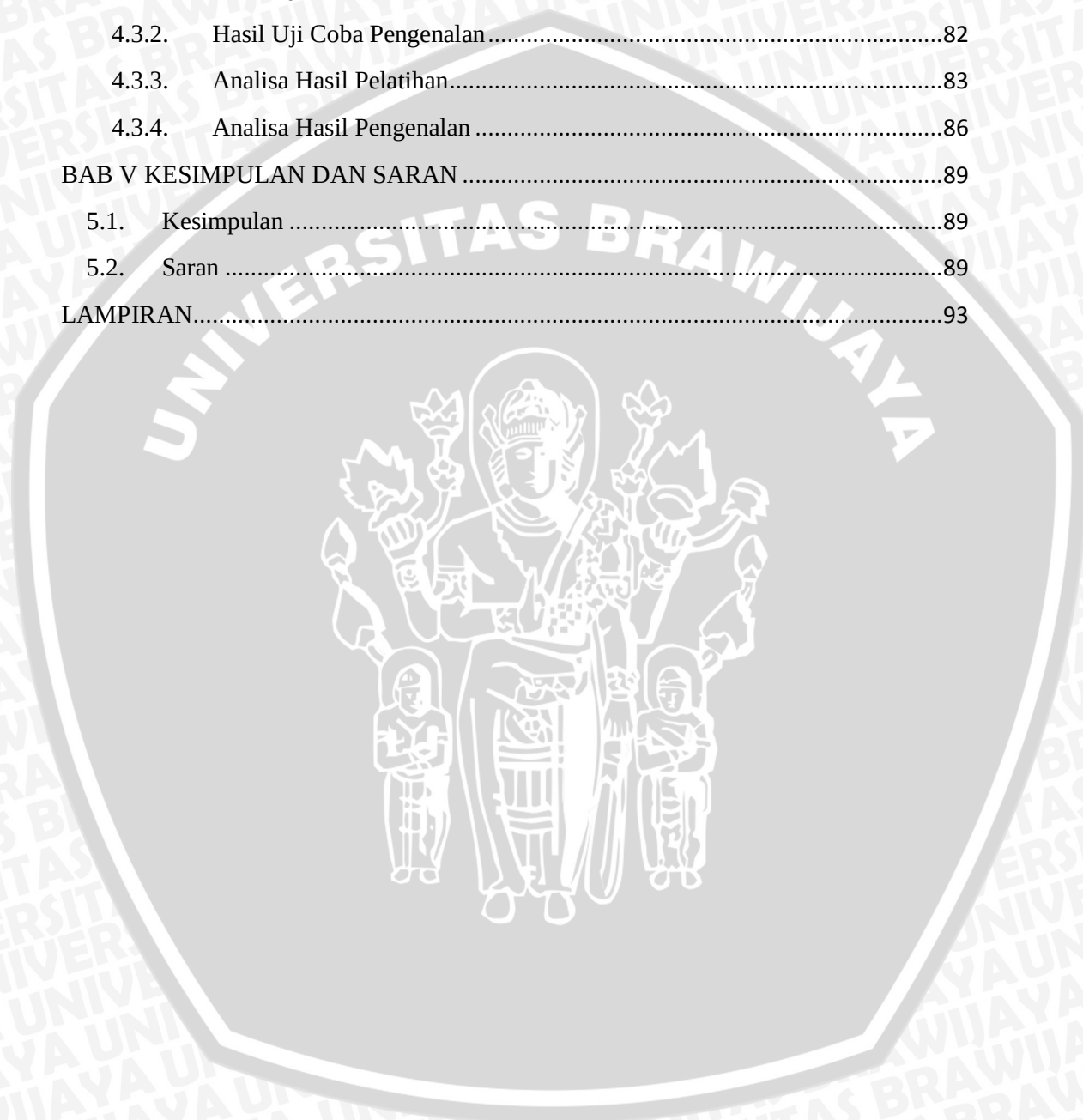
Keywords : Artificial Neural Network, Optical Backpropagation, Cursive Handwriting

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS SKRIPSI.....	iii
KATA PENGANTAR	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
DAFTAR SOURCECODE	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan	3
1.4. Batasan Masalah.....	3
1.5. Manfaat	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Citra	6
2.1.1. Gambar Biner	6
2.1.2. Pengolahan Citra	6
2.2. Jaringan Syaraf Tiruan	11
2.2.1. Definisi Jaringan Syaraf Tiruan.....	11
2.2.2. Proses Pembelajaran.....	11
2.2.3. Arsitektur Jaringan Syaraf Tiruan.....	13
2.2.4. Algoritma Backpropagation	15
2.2.5. Algoritma Optical Backpropagation	20

2.2.6.	Inisialisasi Bobot Awal Secara Random.....	20
2.2.7.	Inisialisasi Bobot Awal dengan Metode Nguyen-Widrow.....	20
2.2.8.	Jumlah Unit Tersembunyi	21
2.2.9.	Fungsi Aktivasi	21
BAB III PERANCANGAN DAN DESAIN SISTEM.....		23
3.1.	Studi Literatur	23
3.2.	Data Latih.....	23
3.3.	Data Uji.....	24
3.4.	Deskripsi Umum Sistem.....	24
3.5.	Perancangan Proses	25
3.5.1.	Pelatihan Jaringan.....	27
3.5.2.	Inisialisasi Bobot dan Bias Awal	28
3.5.3.	<i>Feedforward</i>	30
3.5.4.	Backpropagation.....	31
3.5.5.	Perbaharui Bobot dan Bias	33
3.5.6.	Pengenalan Tulisan Tangan.....	35
3.6.	Perancangan Uji Coba	44
3.6.1.	Rancangan Uji Coba Pengaruh Nilai Learning Rate dan Jumlah Hidden Unit Terhadap Error	44
3.6.2.	Rancangan Uji Coba Pengaruh Nilai Learning Rate dan Jumlah Hidden Unit Terhadap Waktu Pelatihan	45
3.6.3.	Rancangan Pengujian Sistem.....	46
3.7.	Perhitungan Manual	46
3.7.1.	Pelatihan Jaringan.....	48
3.7.2.	Pengujian Jaringan	56
3.8.	Perancangan Antarmuka.....	56
3.8.1.	Form Pelatihan	56
3.8.2.	Form Pengenalan	57
BAB IV IMPLEMENTASI DAN PEMBAHASAN		58
4.1.	Lingkungan Implementasi	58
4.1.1.	Lingkungan Perangkat Keras.....	58
4.1.2.	Lingkungan Perangkat Lunak.....	58
4.2.	Implementasi Program	58

4.2.1.	Implementasi Kelas	59
4.2.2.	Implementasi Antarmuka	75
4.3.	Implementasi Uji Coba.....	76
4.3.1.	Hasil Uji Coba Pelatihan	77
4.3.2.	Hasil Uji Coba Pengenalan.....	82
4.3.3.	Analisa Hasil Pelatihan.....	83
4.3.4.	Analisa Hasil Pengenalan	86
BAB V KESIMPULAN DAN SARAN		89
5.1.	Kesimpulan	89
5.2.	Saran	89
LAMPIRAN.....		93



DAFTAR TABEL

Tabel 2. 1 8-neighbours.....	8
Tabel 2. 2 Contoh input dan target pembelajaran terawasi	12
Tabel 3. 1 Pengaruh nilai learning rate dan jumlah hidden unit terhadap error pelatihan	45
Tabel 3. 2 Pengaruh nilai learning rate dan jumlah hidden unit terhadap lama waktu pelatihan	45
Tabel 3. 3 Hasil Uji Segmentasi.....	46
Tabel 3. 4 Hasil Uji Pengenalan.....	46
Tabel 3. 5 Data pertama	47
Tabel 3. 6 Target untuk data pertama.....	47
Tabel 3. 7 Data kedua.....	47
Tabel 3. 8 Target untuk data kedua.....	48
Tabel 3. 9 Bobot dari input ke hidden layer (V_{ij}).....	48
Tabel 3. 10 Inisialisasi nilai V_j	49
Tabel 3. 11 Nilai V_{ij} yang baru	49
Tabel 3. 12 Nilai bias (V_{0j}).....	50
Tabel 3. 13 Nilai bobot ke output layer (W_{jk})	50
Tabel 3. 14 Nilai bias W_{0k}	50
Tabel 3. 15 Hasil operasi pada hidden (Z_{in})	51
Tabel 3. 16 Hasil aktivasi Z_{in}	51
Tabel 3. 17 Hasil operasi pada output (Y_{in})	52
Tabel 3. 18 Hasil aktivasi Y_{in}	52
Tabel 3. 19 Nilai error Y_k	52
Tabel 3. 20 Nilai Perubahan bobot (ΔW_{jk}).....	52
Tabel 3. 21 Perubahan bobot bias ΔW_{0k}	53
Tabel 3. 22 Faktor penimbang di unit hidden	53
Tabel 3. 23 Aktivasi faktor kesalahan δ	53
Tabel 3. 24 Perubahan bobot ke unit hidden ΔV_{ij}	53

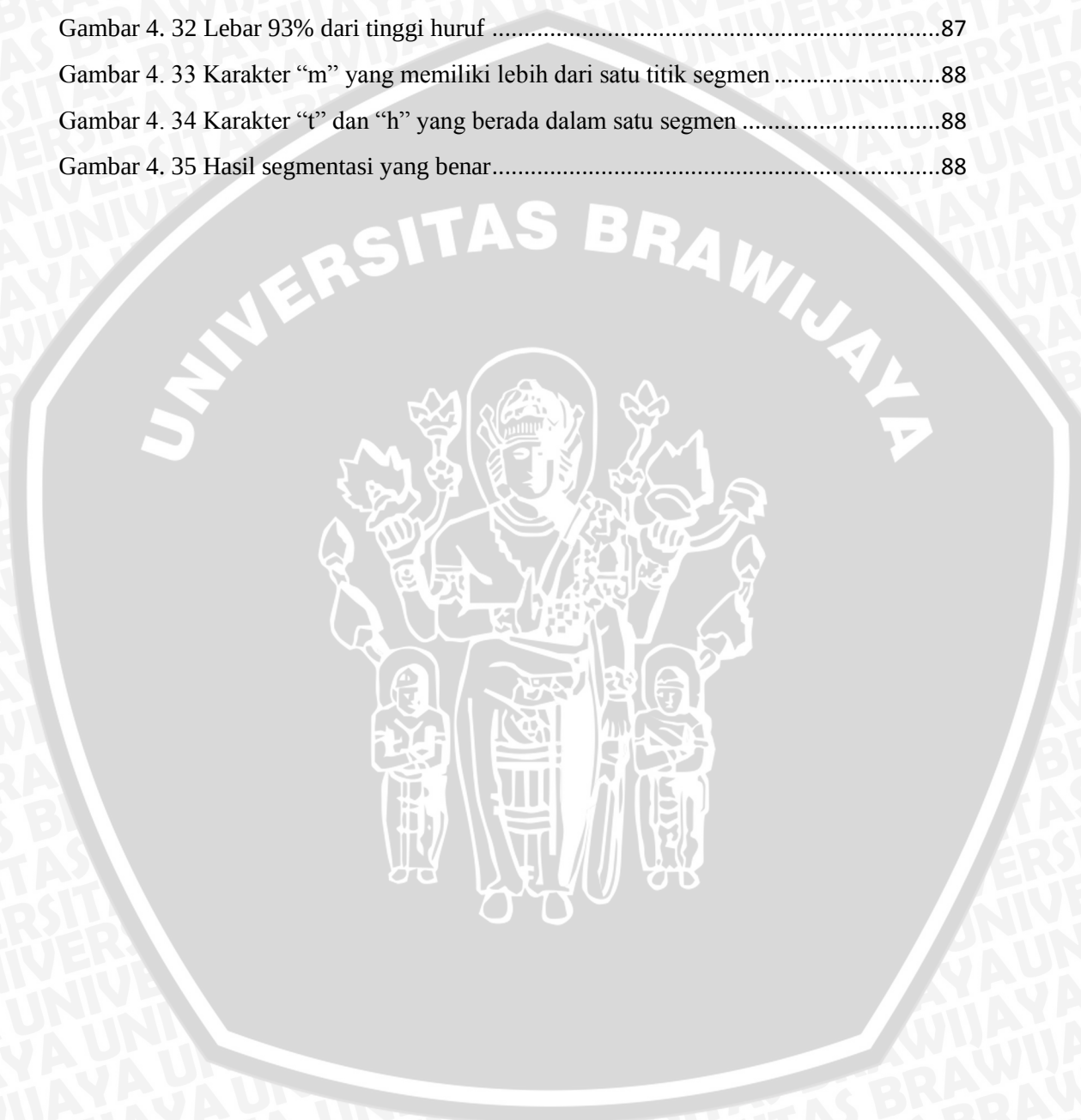
Tabel 3. 25 perubahan bobot bias ΔV_{0j}	54
Tabel 3. 26 Bobot baru (V_{ij}).....	54
Tabel 3. 27 Nilai V_{0j} baru.....	55
Tabel 3. 28 Perubahan bobot W_{jk}	55
Tabel 3. 29 Nilai bobot bias W_{0k} baru	55
Tabel 4. 1 Kelas-kelas yang terdapat pada package gui.....	59
Tabel 4. 2 Method – method yang terdapat pada kelas ImageContainer	61
Tabel 4. 3 Kelas-kelas yang terdapat pada package image manipulation.....	62
Tabel 4. 4 Method-method yang terdapat pada kelas Segmentation	65
Tabel 4. 5 Kelas-kelas yang terdapat pada package neuralnetwork.....	67
Tabel 4. 6 Variabel-variabel yang terdapat pada kelas DataNeural	68
Tabel 4. 7 Penjelasan fungsi method – method yang terdapat pada kelas FileOperation.....	69
Tabel 4. 8 Method-method yang terdapat pada kelas Network	72
Tabel 4. 9 nilai error (RMSE) untuk setiap perubahan hidden unit dan learning rate pada pelatihan.....	79
Tabel 4. 10 waktu yang dibutuhkan untuk pelatihan untuk setiap hidden unit dan learning rate.....	81

DAFTAR GAMBAR

Gambar 2. 1 Contoh hasil thinning	9
Gambar 2. 2 Contoh Word Skew	9
Gambar 2. 3 Proses penentuan baseline	10
Gambar 2. 4 Hasil koreksi Word Skew dan Baseline	11
Gambar 2. 5 Single Layer Network	14
Gambar 2. 6 Multi Layer Network.....	14
Gambar 2. 7 Reccurent Network.....	15
Gambar 3. 1 Data latih untuk karakter a, d, o, e	23
Gambar 3. 2 Contoh IAM Handwriting Character Database	24
Gambar 3. 3 Contoh gambar kata yang terdapat slant	24
Gambar 3. 4 Contoh gambar kata yang terdapat slop	24
Gambar 3. 5 Diagram proses pengenalan tulisan tangan	26
Gambar 3. 6 Arsitektur jaringan syaraf tiruan	26
Gambar 3. 7 Diagram alir proses pelatihan jaringan.....	28
Gambar 3. 8 Diagram alir proses Inisialisasi Bobot	29
Gambar 3. 9 Diagram alir proses Feedforward	31
Gambar 3. 10 Diagram alir proses Backpropagation.....	33
Gambar 3. 11 Diagram alir proses Perbaharui Bobot dan Bias	34
Gambar 3. 12 Diagram alir pengenalan tulisan tangan.....	35
Gambar 3. 13 Diagram alir proses Image Binarization.....	36
Gambar 3. 14 Diagram alir proses Thinning.....	37
Gambar 3. 15 Diagram alir Proses menghapus piksel	39
Gambar 3. 16 Diagram alir proses deteksi word skew dan baseline.....	41
Gambar 3. 17 Diagram alir proses segmentasi dan pengenalan.....	42
Gambar 3. 18 Diagram alir proses segmentasi karakter	43
Gambar 3. 19 Diagram alir proses pengenalan karakter	44
Gambar 3. 20 Pola karakter.....	47
Gambar 3. 21 Representasi pola karakter	47

Gambar 3. 22 Form pelatihan.....	56
Gambar 3. 23 Form Pengenalan.....	57
Gambar 4. 1 Class Diagram aplikasi pengenalan tulisan tangan.....	59
Gambar 4. 2 Kelas ImageContainer	60
Gambar 4. 3 Kelas IFTraining.....	61
Gambar 4. 4 Kelas IFRecognition	62
Gambar 4. 5 Kelas Binarization	63
Gambar 4. 6 Kelas DataImage	64
Gambar 4. 7 Kelas Thinning	64
Gambar 4. 8 Kelas Segmentation.....	65
Gambar 4. 9 Kelas SkewDetection.....	66
Gambar 4. 10 Kelas Histogram	66
Gambar 4. 11 Kelas DataNeural.....	68
Gambar 4. 12 Kelas FileOperation.....	69
Gambar 4. 13 Kelas LookUp.....	70
Gambar 4. 14 Kelas MasterLookUp.....	71
Gambar 4. 15 Kelas Network.....	72
Gambar 4. 16 Kelas Recognition	74
Gambar 4. 17 Kelas Tools.....	74
Gambar 4. 18 Form Training.....	75
Gambar 4. 19 Form Recognition.....	76
Gambar 4. 20 Threshold #222222.....	77
Gambar 4. 21Threshold #777777.....	78
Gambar 4. 22 Threshold #0f0f0f	78
Gambar 4. 23 Threshold #dedcdc.....	78
Gambar 4. 24 Grafik pengaruh learning rate terhadap error (RMSE) yang dihasilkan.....	79
Gambar 4. 25 Grafik pengaruh hidden unit terhadap perubahan error (RMSE)	80
Gambar 4. 26 Grafik pengaruh learning rate terhadap waktu pelatihan.....	81
Gambar 4. 27 Grafik pengaruh hidden unit terhadap waktu yang dibutuhkan untuk pelatihan.....	82
Gambar 4. 28 Grafik perubahan error (RMSE)	83

Gambar 4. 29 Grafik pengaruh hidden unit terhadap training error (hidden unit mulai dari 1 - 110).....	85
Gambar 4. 30 Lebar 63% dari tinggi huruf	86
Gambar 4. 31 Lebar 83% dari tinggi huruf	87
Gambar 4. 32 Lebar 93% dari tinggi huruf	87
Gambar 4. 33 Karakter “m” yang memiliki lebih dari satu titik segmen	88
Gambar 4. 34 Karakter “t” dan “h” yang berada dalam satu segmen	88
Gambar 4. 35 Hasil segmentasi yang benar.....	88



DAFTAR SOURCECODE

Sourcecode 4. 1 fungsi binarization	63
Sourcecode 4. 2 kelas DataImage	64
Sourcecode 4. 3 fungsi penentuan baseline.....	66
Sourcecode 4. 4 memetakan dari kode ke karakter aslinya	71
Sourcecode 4. 5 memetakan karakter asli ke kode	71
Sourcecode 4. 6 kelas MasterLookUp	72
Sourcecode 4. 7 fungsi untuk pelatihan	73
Sourcecode 4. 8 fungsi Sigmoid.....	74
Sourcecode 4. 9 fungsi turunan sigmoid	74
Sourcecode 4. 10 fungsi Mean Square Error (MSE).....	74



DAFTAR LAMPIRAN

Lampiran 1 Data Hasil Segmentasi	93
Lampiran 2 Data Hasil Pengenalan	99
Lampiran 3 Data Kode Karakter	109



BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi komputer saat ini telah berkembang sangat pesat dengan beragam kemampuan yang semakin mempermudah kerja manusia. Tidak hanya itu, teknologi itu pun sekarang telah dikemas dalam bentuk yang lebih kecil dan lebih murah, seperti *smartphone*, komputer tablet, dan perangkat-perangkat yang lainnya yang semakin nyaman untuk dibawa kemana saja.

Sekarang dikenal adanya teknologi *touchscreen*, yang mana teknologi tersebut dimungkinkan untuk memberikan suatu masukan atau aksi pada suatu program hanya dengan sentuhan jari atau alat khusus (*stylus*) pada layar perangkat. Perangkat-perangkat yang menggunakan teknologi *touchscreen* memaksimalkan ukuran layar sebagai tampilan aplikasinya, yang kebanyakan tanpa menggunakan *keyboard* atau *keypad*, adapun hanya *keyboard virtual* atau *keypad virtual*. Tulisan tangan merupakan metode yang dapat digunakan untuk memasukkan suatu data pada perangkat-perangkat tersebut, seperti dalam menuliskan suatu pesan atau catatan yang kemudian dari tulisan tangan tersebut dapat diubah menjadi karakter yang dapat dikenali komputer.

Secara umum, terdapat beberapa tahapan yang dilakukan dalam pengenalan tulisan tangan, yang secara terurut yaitu *pre-processing*, segmentasi dan kemudian pengenalan. Tahapan *pre-processing* difokuskan pada pengubahan masukan tulisan tangan menjadi sesuai dengan standar yang dikehendaki pada proses selanjutnya. Segmentasi adalah proses menemukan batasan-batasan karakter. Dan akhirnya pengenalan yang mengklasifikan karakter yang terbentuk pada tahapan sebelumnya ke dalam karakter yang merepresentasikannya [LEE-09].

Segmentasi karakter memegang peranan penting dalam pengenalan tulisan tangan, terutama tulisan tangan yang antar hurufnya bersinggungan. Pada tulisan tangan yang antar hurufnya bersambung, rawan menimbulkan error pada tahapan segmentasi yang kemudian juga mengarah pada kesalahan pengenalan karakter [KUR-11].

Penelitian tentang pengenalan karakter pada tulisan tangan bersambung sebelumnya pernah dilakukan dengan judul “*Binary Segmentation with Neural Validation for Cursive Handwriting Recognition*” [LEE-09]. Dalam penelitian tersebut dilakukan segmentasi karakter yang kemudian menggunakan metode dalam Jaringan Syaraf Tiruan yaitu Backpropagation sebagai validator apakah kandidat segmentasi tersebut karakter atau bukan, sedangkan *pre-processing* yang digunakan adalah menentukan *baseline*.

Jaringan Syaraf Tiruan (JST) adalah sebuah mekanisme pemrosesan informasi yang karakteristik pemrosesannya didasarkan pada prinsip kerja jaringan syaraf pada makhluk hidup. Jaringan Syaraf Tiruan dapat dikarakteristikkan berdasarkan 3 hal, yaitu berdasarkan arsitektur jaringannya (hubungan antar neuron), berdasarkan metode dalam menentukan bobot setiap neuron, dan berdasarkan fungsi aktivasi [FAU-93].

Backpropagation merupakan salah satu metode dalam Jaringan Syaraf Tiruan (JST) yang termasuk dalam jenis pembelajaran dengan pengawasan (*supervised*), yang mana pembelajaran dilakukan dengan diberikan output yang spesifik [FAU-93].

Pada skripsi ini akan digunakan metode dalam Jaringan Syaraf Tiruan yaitu Optical Backpropagation untuk mengenali tulisan tangan sesuai dengan karakter yang merepresentasikannya, yang sekaligus juga sebagai validator apakah kandidat segmentasi tersebut termasuk karakter atau bukan, dan dengan beberapa *pre-processing* yang akan dilakukan seperti *Binarization*, *Thinning*, dan *Core-Zone Estimation*. Adapun Optical Backpropagation merupakan metode Backpropagation yang telah dimodifikasi pada tahapan penghitungan *error*.

Data (gambar tulisan tangan) yang digunakan pada skripsi ini adalah tulisan tangan yang antar hurufnya bersambung, yang mana tulisan tangan tersebut direpresentasikan dalam suatu gambar *grayscale*.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut maka rumusan masalah dalam skripsi yang akan dikerjakan adalah:

1. Bagaimana menerapkan metode Optical Backpropagation untuk pengenalan karakter pada tulisan tangan bersambung?
2. Bagaimana pengaruh jumlah *neuron* dan *learning rate* terhadap keakuratan dan kecepatan proses metode Optical Backpropagation untuk pengenalan karakter pada tulisan tangan bersambung?

1.3. Tujuan

Tujuan yang akan dicapai pada skripsi ini adalah:

1. Mengimplementasikan metode Optical Backpropagation untuk pengenalan karakter pada tulisan tangan bersambung.
2. Mengetahui pengaruh jumlah *neuron* dan *learning rate* terhadap keakuratan dan kecepatan proses metode Optical Backpropagation untuk pengenalan karakter pada tulisan tangan bersambung.

1.4. Batasan Masalah

Hal-hal yang dibatasi pada skripsi ini adalah sebagai berikut:

1. Tulisan tangan yang dikenali adalah tulisan tangan alfabet bersambung.
2. Tulisan tangan yang dikenali direpresentasikan dalam suatu gambar *grayscale*.
3. Tulisan yang dikenali hanya perkata

1.5. Manfaat

Manfaat yang ingin dicapai dari skripsi ini adalah metode ini dapat diimplementasikan dalam suatu aplikasi yang *user friendly*, sehingga nantinya bisa diterapkan pada perangkat-perangkat yang membutuhkan inputan berupa tulisan tangan.

1.6. Metodologi Penelitian

1. Studi Literatur

Mempelajari tentang pengolahan citra digital dan metode Optical Backpropagation melalui berbagai macam media, antara lain melalui internet, jurnal-jurnal dan buku yang berhubungan dengan materi tersebut.

2. Perancangan Sistem

Melakukan perancangan sistem dengan menguji algoritma yang digunakan terhadap data-data yang ada dan melakukan perhitungan manual apakah telah sesuai dengan yang diharapkan.

3. Implementasi

Pembuatan aplikasi pengenalan tulisan tangan berdasarkan perancangan yang telah dibuat sebelumnya ke dalam program komputer.

4. Uji coba produk dan evaluasi

Melakukan uji coba program yang telah dibuat. Kemudian melakukan evaluasi terhadap kekurangan program dan membuat laporannya.

1.7. Sistematika Penulisan

Skripsi ini disusun dengan sistematika penulisan, sebagai berikut:

1. BAB I PENDAHULUAN

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

2. BAB II TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai pustaka yang digunakan dalam pengerjaan skripsi. Teori-teori yang terdapat pada bab ini mencakup pengolahan citra digital, jaringan syaraf tiruan secara umum dan Optical Backpropagation.

3. BAB III PERANCANGAN DAN DESAIN SISTEM

Pada bab ini dibahas mengenai urutan langkah-langkah pengerjaan untuk pengenalan tulisan tangan bersambung, perancangan *user interface* dan disertai dengan perhitungan manual menggunakan algoritma Optical Backpropagation.

4. BAB IV IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini dibahas tentang implementasi metode yang digunakan dalam hal ini adalah algoritma Optical Backpropagation dalam pengenalan tulisan tangan dan uji coba terhadap program yang telah dibuat.

5. BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan yang didapat dari pembuatan skripsi ini dan saran-saran yang mungkin dapat berguna dalam penelitian lebih lanjut.

UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

2.1. Citra

Citra yang merupakan gambar pada bidang dua dimensi, secara matematis, adalah fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, seperti mata pada manusia, kamera, pemindai (scanner), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam. Citra digital biasanya direpresentasikan sebagai matriks dengan dimensi ukuran yang dinyatakan sebagai panjang (x) dan lebar (y) dengan derajat keabuan $f(x,y)$ [MUN-04].

2.1.1. Gambar Biner

Gambar biner (binary image) adalah gambar dimana piksel-piksennya hanya memiliki dua buah nilai intensitas yang biasanya dikodekan dengan 0 dan 1, yang mana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta/objek (*foreground*) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih.

2.1.2. Pengolahan 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, frequency*) dan secara tingkat keabuan. 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 $N \times M$ dimana elemen-elemen dari matriks ini adalah perwakilan dari beberapa pixel pada gambar (sampling) dengan nilai bit biner (0 atau 1) [MUN-04].

2.1.2.1. *Image Binarization*

Image Binarization digunakan untuk mengubah citra *grayscale* menjadi citra 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 *image binarization* sangat berguna untuk menentukan daerah yang terterang dan tergelap dari sebuah citra [MUN-04].

2.1.2.2. *Thinning*

Thinning adalah proses penipisan garis pada suatu objek citra biner (*binary image*). Oleh karena itu jika citra tersebut merupakan suatu citra *grayscale*, biasanya dilakukan *binarization* terlebih dahulu sedemikian rupa sehingga citra tersebut menjadi citra biner. Algoritma *thinning* secara iteratif menghapus piksel-piksel pada citra sampai terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu piksel) terhubung menjadi suatu garis.

Thinning dengan algoritma Zhang-Suen adalah salah satu algoritma *thinning* yang cukup populer dan telah digunakan sebagai suatu basis perbandingan untuk *thinning*. Setiap iterasi dari metode ini terdiri dari dua sub-iterasi yang berurutan yang dilakukan terhadap *contour points* dari wilayah citra [ZUR-09].

a. *Connectivity Number*

Connectivity Number adalah nilai yang menyatakan jumlah berapa banyak objek yang tersambung dengan piksel tertentu. Adapun rumusan yang digunakan untuk menghitung *Connectivity Number* adalah

$$C_n = \sum_{k \in S} N_k - (N_k * N_{k+1} * N_{k+2}) \quad (2.1)$$

C_n adalah nilai *Connectivity* pada piksel ke- n . N_k adalah warna dari 8-*neighbours* dari piksel yang dianalisis. N_0 adalah piksel pusat. N_1 adalah nilai warna dari piksel sebelah kanan dari piksel pusat, dan seterusnya mengikuti arah jarum jam.

Tabel 2. 1 8-neighbours

N_6	N_7	N_8
N_5	N_0	N_1
N_4	N_3	N_2

b. Algoritma Zhang-Suen

Algoritma Zhang-Suen adalah metode paralel yang berarti nilai baru yang diperoleh hanya tergantung pada nilai iterasi sebelum-nya. Metode ini cepat dan sederhana untuk diimplementasikan. Algoritma ini menggunakan dua sub-iterasi, pada sub-iterasi yang pertama, sebuah piksel $P(i,j)$ akan dihapus jika kondisi berikut terpenuhi:

1. Nilai *Connectivity Number* adalah 1
2. Sekurang-kurangnya terdapat 2 tetangga piksel yang bernilai 1 dan tidak lebih dari 6
3. Sekurang-kurangnya 1 dari $P(i,j+1)$, $P(i-1,j)$, $P(i,j-1)$ adalah 0
4. Sekurang-kurangnya 1 dari $P(i-1,j)$, $P(i+1,j)$, $P(i,j-1)$ adalah 0

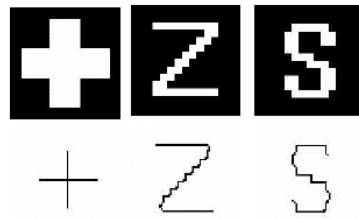
Pada sub-iterasi yang kedua, kondisi pada point 3 dan 4 diganti, sehingga menjadi:

1. Nilai *Connectivity Number* adalah 1
2. Sekurang-kurangnya terdapat 2 tetangga piksel yang bernilai 1 dan tidak lebih dari 6
3. Sekurang-kurangnya 1 dari $P(i-1,j)$, $P(i,j+1)$, $P(i+1,j)$ adalah 0
4. Sekurang-kurangnya 1 dari $P(i,j+1)$, $P(i+1,j)$, $P(i,j-1)$ adalah 0

Ket :

$P(i,j)$ = nilai piksel pada titik (i, j) dan pada akhir, piksel yang memenuhi kondisi-kondisi tersebut akan dihapus. Jika pada akhir sub-iterasi, baik itu sub-iterasi yang pertama maupun yang kedua, tidak ada piksel yang dihapus maka iterasi dihentikan dan proses selesai [ZUR-09].

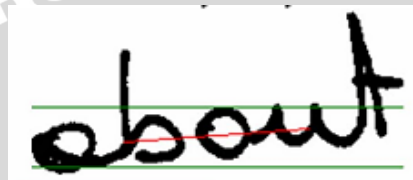
Adapun contoh hasil dari algoritma thinning ini adalah



Gambar 2. 1 Contoh hasil thinning

2.1.2.3. Word Skew dan Baseline Detection

Word Skew adalah kondisi dimana kata yang dituliskan tidak sejajar dengan garis horizontal (*Baseline*), seperti gambar



Gambar 2. 2 Contoh Word Skew

Adapun algoritma yang digunakan untuk menentukan *word skew* dan *baseline* menurut Gatos, dkk [GAT-06] adalah,

1. Sebuah array yang merepresentasikan sebuah gambar $im(x,y)$ yang mempunyai nilai 1 untuk foreground dan 0 untuk background piksel, x_{max} dan y_{max} adalah lebar dan tinggi dari gambar kata.
2. Hitung proyeksi horizontal kiri dan kanan (LP dan RP), menggunakan rumusan :

$$LP(y) = \sum_{x=0}^{\frac{x_{max}}{2}} im(x,y) \quad (2.2) \quad RP(y) = \sum_{x=\frac{x_{max}}{2}}^{x_{max}} im(x,y) \quad (2.3)$$

$LP(y)$ = nilai *density histogram* bagian kiri pada titik y

$RP(y)$ = nilai *density histogram* bagian kanan pada titik y

3. Hitung global maksimum dari LP dan RP, $y=y_L$ dan $y=y_R$
4. Hitung nilai dari y_{L1} , y_{L2} dan y_{R1} , y_{R2} yang sesuai dengan nilai-nilai y terdekat dari kedua sisi y_L dan y_R , sehingga memenuhi

$$LP(y) < 0.2LP(y_L) \quad \text{dan} \quad RP(y) < 0.2RP(y_R) \quad (2.4)$$

$$y_{L1} = y: (LP(y) < 0.2LP(y_L) \& y = \max(y_i)), y_i \in [0, y_L]$$

$$y_{L2} = y: (LP(y) < 0.2LP(y_L) \& y = \max(y_i)), y_i \in [y_L, y_{\max}]$$

$$y_{R1} = y: (RP(y) < 0.2RP(y_R) \& y = \max(y_i)), y_i \in [0, y_R]$$

$$y_{R2} = y: (RP(y) < 0.2RP(y_R) \& y = \max(y_i)), y_i \in [y_R, y_{\max}]$$

y_{L1} = titik batas atas untuk bagian kiri

y_{L2} = titik batas bawah untuk bagian kiri

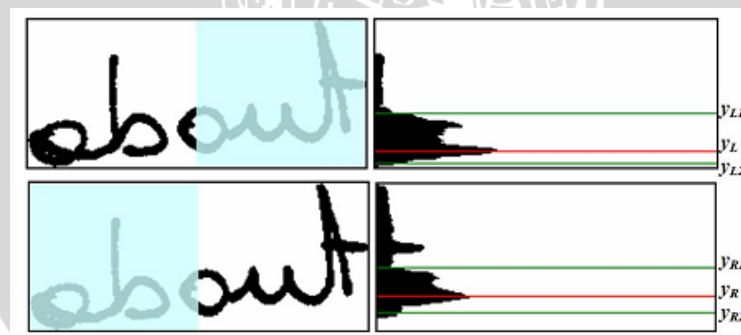
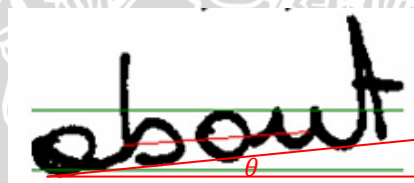
y_{R1} = titik batas atas untuk bagian kanan

y_{R2} = titik batas bawah untuk bagian kanan

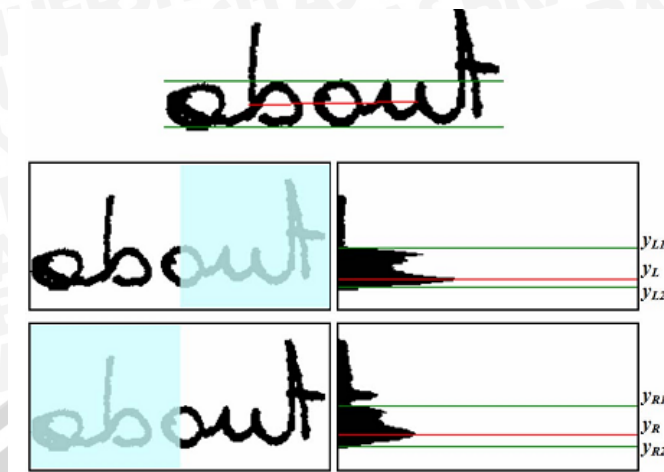
5. Besarnya sudut word skew dapat ditentukan dengan rumusan,

$$\theta = \tan^{-1} \left(\frac{y_{R1} + y_{R2} - y_{L1} - y_{L2}}{x_{\max}} \right) \quad (2.5)$$

θ = sudut yang terbentuk antara tulisan tangan dengan garis horizontal
(baseline)



Gambar 2. 3 Proses penentuan baseline



Gambar 2. 4 Hasil koreksi Word Skew dan Baseline

2.2. Jaringan Syaraf Tiruan

2.2.1. 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 [KRI-04].

2.2.2. 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, menghafal-

nya, dan menalar. 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) [KUS-03].

Pembelajaran JST merupakan proses pencarian konfigurasi bobot yang sesuai dengan cara perkalian, penjumlahan dan aktivasi bobot dan input. 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 *Backpropagation* [FAU-93].

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 2.2.

Tabel 2. 2 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 [KRI-04].

b. Pembelajaran Tidak Terawasi (*Unsupervised Larning*)

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 [KRI-04].

2.2.3. Arsitektur Jaringan Syaraf Tiruan

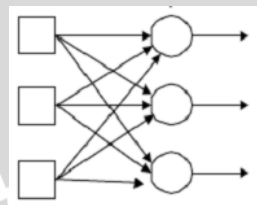
Suatu jaringan syaraf minimum tersusun atas *input layer* dan *output layer*. Dalam beberapa tipe jaringan diantara *input layer* dan *output layer* terdapat *hidden layer*. *Input layer* merupakan aktifitas unit-unit lapisan masukan yang menunjukkan informasi dasar yang kemudian digunakan dalam jaringan 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 kesemua *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. [SIA-05] berdasarkan jumlah layernya jaringan syaraf tiruan dibagi menjadi:

a. Single Layer Network

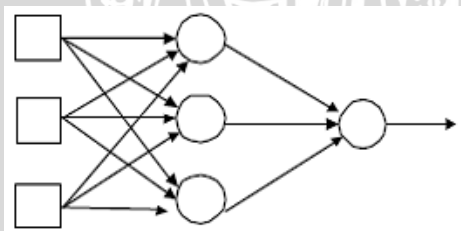
Single layer network ini pertama kali dirancang oleh Widrow dan Holf pada tahun 1960. Neural network jenis ini memiliki koneksi pada inputnya secara langsung ke jaringan outputnya. Pada jaringan ini, semua unit input dihubungkan langsung ke unit output, meskipun dengan bobot yang berbeda-beda. *Single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



Gambar 2. 5 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*.

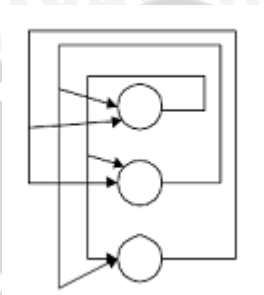


Gambar 2. 6 Multi Layer Network

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

c. *Recurrent Network*

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



Gambar 2. 7 Reccurent Network

Algoritma pelatihan Backpropagation atau propagasi balik pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhat dan McClelland untuk dipakai pada JST. Algoritma ini termasuk metode pelatihan *supervised* dan untuk operasi pada jaringan *feedforward* multi lapis.

Metode Backpropagation ini banyak diaplikasikan secara luas. Bahkan, sekitar 90% lebih Backpropagation 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 Backpropagation 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 [KRI-04].

2.2.4. 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 memperbaharui bobot dalam setiap *network*.

Berikut adalah algoritma dari Backpropagation Network (algoritma untuk melatih *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 , $i = 1, 2, \dots, n$ menerima sinyal masukan x_i dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit lapisan tersembunyi).

Langkah 4 :

Masing-masing unit dilapisan tersembunyi (Z_j , $j=1,2,3, \dots, n$) menjumlahkan sinyal-sinyal input berbobot:

$$Z_{in_j} = V_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.6)$$

Dimana:

Z_{in_j} = Input jaringan ke Z_j

V_{0j} = Bias pada lapisan tersembunyi j

x_i = Unit input i

v_{ij} = Nilai penimbang sambungan dari unit x_i ke unit Z_j

Kemudian gunakan fungsi aktivasi untuk menghitung sinyal keluarannya dengan rumus :

$$Z_j = f(Z_{in_j}) = \frac{1}{1 + \exp(-Z_{in_j})} \quad (2.7)$$

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 keluaran (y_k , $k = 1,2,3, \dots, m$) menjumlahkan sinyal-sinyal input berbobot :

$$y_{in_k} = W_{ok} + \sum_{j=1}^p Z_j W_{jk} \quad (2.8)$$

Dimana :

y_{in_k} = net masukan untuk unit y_k

W_{ok} = nilai penimbang sambungan pada bias untuk unit y_k

W_{jk} = nilai penimbang sambungan dari Z_j ke unit y_k

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran-nya

$$y_k = f(y_{in_k}) = \frac{1}{1 + \exp(-y_{in_k})} \quad (2.9)$$

Dimana : y_k = Unit ke- k pada lapisan tersembunyi

Backpropagasi dan Galatnya

Langkah 6 :

Masing-masing unit keluaran (y_k , $k = 1, 2, 3, \dots, m$) menerima target pola yang berhubungan dengan pola masukan pembelajaran, hitung informasi error-nya:

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (2.10)$$

Dimana :

δ_k = faktor pengaturan nilai penimbang sambungan pada lapisan keluaran.

t_k = pola target ke- k .

Sebagaimana input training data, target training data t_k juga telah diskalakan menurut fungsi aktivasi yang digunakan. Faktor δ_k ini digunakan untuk menghitung koreksi error ($\delta_{w_{jk}}$) yang nantinya akan dipakai untuk memperbaharui W_{jk} , dimana :

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (2.11)$$

Dimana :

α = konstanta laju pelatihan (learning rate) dengan $0 < \alpha < 1$

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

$$\Delta W_{0k} = \alpha \delta_k \quad (2.12)$$

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

Langkah 7 :

Setiap hidden unit (Z_j , $j=1, \dots, p$) menjumlahkan input delta yang sudah terbobot.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (2.13)$$

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 :

$$\delta_j = \delta_{in_j} f'(Z_{in_j}) \quad (2.14)$$

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

$$\Delta V_{ij} = \alpha \delta_j x_i \quad (2.15)$$

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

$$\Delta V_{0j} = \alpha \delta_j \quad (2.16)$$

Memperbaiki bobot dan bias

Langkah 8 :

Masing-masing unit output (y_k , $k = 1, 2, 3, \dots, m$) akan memperbaharui bias dan bobotnya dari setiap hidden unit ($j=0, \dots, p$),

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.17)$$

Demikian pula untuk setiap hidden unit (Z_j , $j=1, \dots, p$) akan diperbaharui bias dan bobotnya dari setiap unit input ($j=0, 1, 2, 3, \dots, n$)

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (2.18)$$

Langkah 9 :

Memeriksa *stopping condition*

Terdapat dua parameter untuk memeriksa *stopping condition*, yang pertama yaitu maksimum epoch dan minimum *error*. Jika salah satu dari *stop condition* tersebut telah terpenuhi, maka pelatihan jaringan dapat dihentikan. Untuk mengetahui error pelatihan digunakan metode *Mean Square Error* (MSE) yang langkah-langkahnya sebagai berikut :

1. Dengan bobot yang telah didapat, lakukan langkah *feedforward* (langkah 3 sampai langkah 5).
2. 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

$$MSE = \frac{1}{2} \sum_{j=1}^m (t_{kj} - y_{kj})^2 \quad (2.19)$$

2.2.5. Algoritma Optical Backpropagation

Algoritma Optical Backpropagation tidak jauh beda dengan algoritma Backpropagation, yang membedakan hanyalah rumusan untuk menghitung informasi errornya (langkah 6). Kecepatan konvergensi dari proses pelatihan dapat ditingkatkan secara signifikan oleh OBP melalui memaksimalkan sinyal error, yang akan ditransmisikan ke belakang dari lapisan output untuk setiap unit dilapisan tengah (*hidden layer*) [SAL-05].

Dalam Backpropagation rumus yang digunakan untuk menghitung informasi error adalah :

$$\delta_k = (t_k - y_k)f'(y_{in_k})$$

Sedangkan dalam Optical Backpropagation rumusan yang digunakan adalah :

$$\text{New}\delta_k = (1 + e^{(t_k - y_k)^2})f'(y_{in_k}), \quad \text{Jika } (t_k - y_k) \geq 0 \quad (2.20)$$

$$\text{New}\delta_k = -(1 + e^{(t_k - y_k)^2})f'(y_{in_k}), \quad \text{Jika } (t_k - y_k) < 0 \quad (2.21)$$

2.2.6. Inisialisasi Bobot Awal Secara Random

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global (atau mungkin hanya lokal 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. Sebaliknya, apabila nilai bobot 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).

2.2.7. 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 = $0.7\sqrt{P}$

Langkah 1 : Kerjakan untuk setiap unit lapisan tersembunyi ($j = 1, 2, 3, \dots, P$):

- Inisialisasi semua bobot dari lapisan input ke lapisan tersembunyi:
 V_{ij} = bilangan random antara -0.5 sampai 0.5
- Hitung inisialisasi bobot menggunakan persamaan 2.22

$$\|V_j\| = \sqrt{V_{1j}^2 + V_{2j}^2 + \dots + V_{nj}^2} \quad (2.22)$$

- Inisialisasi ulang bobot-bobot

$$V_{ij} = \frac{\beta V_{ij}}{\|V_j\|} \quad (2.23)$$

- Bias yang dipakai sebagai inisialisasi V_{0j} = bilangan acak antara $-\beta$ dan β .

2.2.8. Jumlah Unit Tersembunyi

Menentukan jumlah lapis pada hidden layer dan menentukan jumlah unit 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 [SIA-05]. Jumlah unit per layer yang optimal dapat ditentukan dengan beberapa percobaan.

2.2.9. Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan saraf tiruan [KUS-03], tetapi yang digunakan adalah fungsi Sigmoid Biner.

Dalam back propagation, 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 outputnya 0 atau 1. Fungsi sigmoid biner dapat dilihat pada persamaan 2.24 :

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} \quad (2.24)$$

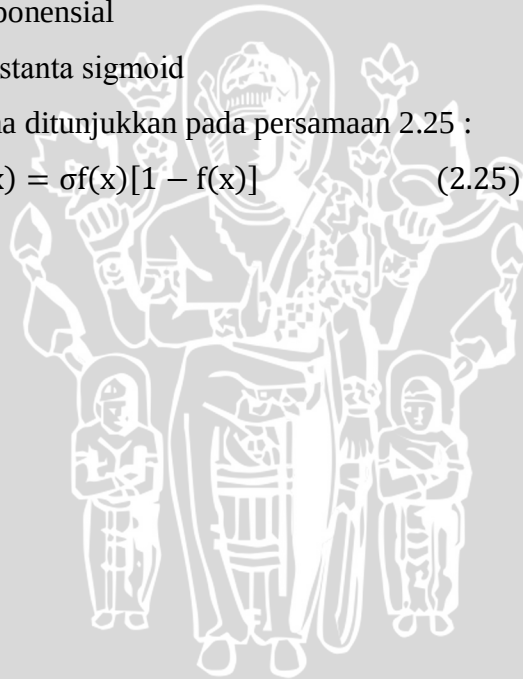
Dimana : $f(x)$ = Fungsi Sigmoid

e = eksponensial

σ = konstanta sigmoid

dengan turunan pertama ditunjukkan pada persamaan 2.25 :

$$f'(x) = \sigma f(x)[1 - f(x)] \quad (2.25)$$



BAB III

PERANCANGAN DAN DESAIN SISTEM

Dalam bab ini akan dijelaskan mengenai analisis sistem dan perancangan sistem untuk pengenalan tulisan tangan bersambung menggunakan jaringan syaraf tiruan dengan metode Optical Backpropagation.

Penelitian dilakukan dengan tahapan-tahapan sebagai berikut:

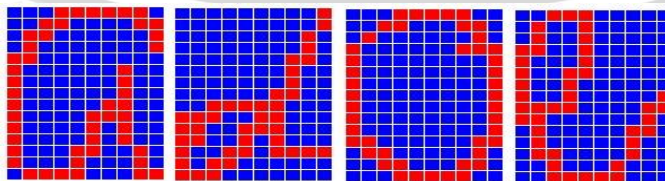
1. Mempelajari literatur yang berhubungan dengan pengolahan citra digital dan jaringan syaraf tiruan Optical Backpropagation.
2. Menganalisa sistem dan melakukan perancangan sistem menggunakan jaringan syaraf tiruan Optical Backpropagation.
3. Mengimplementasikan sistem.
4. Melakukan uji coba sistem dengan memasukkan data yang berbeda dengan data training ke dalam sistem.
5. Mengevaluasi hasil pengujian.

3.1. Studi Literatur

Studi Literatur digunakan untuk memperdalam pengetahuan yang berhubungan dengan permasalahan pada skripsi ini. Teori-teori mengenai pengolahan citra digital dan jaringan syaraf tiruan yang digunakan sebagai dasar skripsi ini didapat dari buku, jurnal dan artikel-artikel dari internet.

3.2. Data Latih

Data latih yang digunakan untuk pelatihan jaringan syaraf tiruan dalam penelitian ini adalah pattern yang merepresentasikan citra setiap karakter. Adapun contoh data latih ditunjukkan pada gambar 3.1.



Gambar 3. 1 Data latih untuk karakter a, d, o, e

3.3. Data Uji

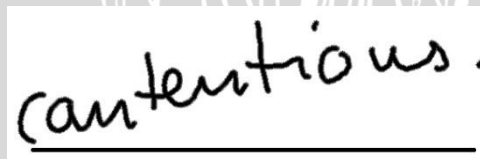
Data uji yang digunakan dalam penelitian ini diperoleh dari IAM Handwriting Character Database (<http://www.iam.unibe.ch/fki/databases/iam-handwriting-database>). Database tersebut berisi citra tulisan tangan dengan berbagai bentuk tulisan tangan yang berbeda. Dari database tersebut, tidak semua data akan dijadikan sebagai data uji, data tersebut dipilih gambar kata yang tidak terdapat slop maupun slant. Adapun contoh IAM Handwriting Character Database ditunjukkan pada gambar 3.2, sedangkan contoh gambar kata yang terdapat *slant* dan *slop* ditunjukkan pada gambar 3.3 dan 3.4.

A sample of handwritten text from the IAM Handwriting Character Database. The word "tomorrow." is written in a clear, legible cursive script on a white background.

Gambar 3. 2 Contoh IAM Handwriting Character Database

A sample of handwritten text showing a slant. The word "strong" is written in a cursive script, tilted to the right, on a white background.

Gambar 3. 3 Contoh gambar kata yang terdapat *slant*

A sample of handwritten text showing slop. The word "cautentious." is written in a cursive script, with irregular spacing and alignment, on a white background.

Gambar 3. 4 Contoh gambar kata yang terdapat *slop*

3.4. Deskripsi Umum Sistem

Secara umum sistem yang dibangun adalah perangkat lunak untuk mengenali tulisan tangan yang mengimplementasikan jaringan syaraf tiruan Optical Backpropagation. Jaringan syaraf tiruan berfungsi sebagai penentuan

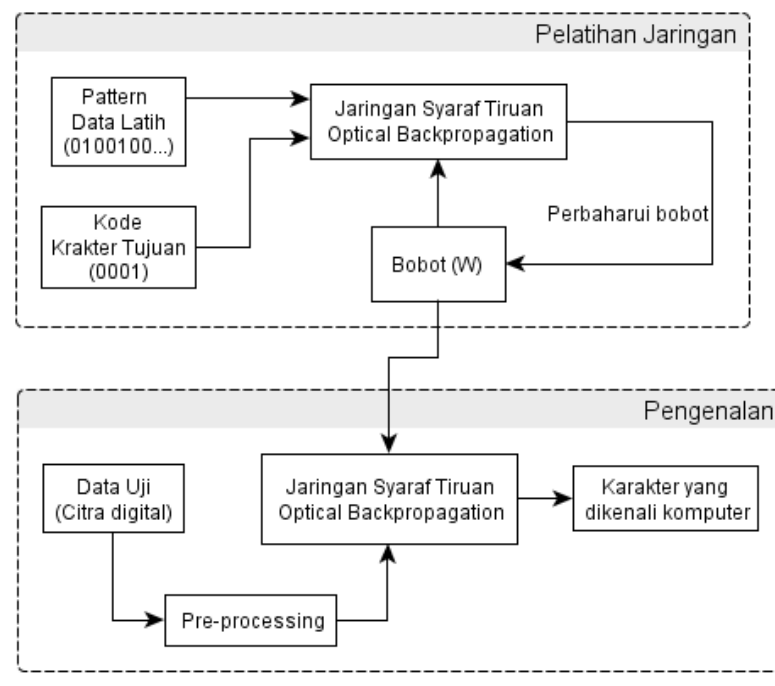
segmentasi karakter yang sekaligus mengenali karakter tersebut. Sistem bertujuan untuk merubah tulisan tangan menjadi karakter-karakter yang dikenali komputer, sehingga nantinya dapat dimanfaatkan pada perangkat-perangkat yang membutuhkan masukan tulisan tangan.

3.5. Perancangan Proses

Tahapan proses dalam sistem dibagi menjadi dua bagian yaitu pelatihan jaringan dan pengenalan tulisan tangan. Pelatihan jaringan adalah suatu proses yang dimaksudkan untuk melatih jaringan agar mempunyai pengetahuan sehingga dapat mengenali karakter. Proses tersebut akan menghasilkan bobot yang sesuai yang nantinya digunakan untuk pengenalan karakter pada jaringan. Proses pelatihan jaringan membutuhkan beberapa masukan, yaitu pola (*pattern*) data latih, kode karakter tujuan dan inisialisasi bobot awal. Pola (*pattern*) data latih merupakan pola yang didapatkan dari piksel tiap karakter. Kode karakter tujuan merupakan kode yang merepresentasikan setiap karakter tujuan. Sedangkan inisialisasi bobot awal merupakan inisialisasi bobot dengan nilai random. Proses pelatihan ini akan memperbaharui bobot sampai bobot tersebut sudah sesuai (mencapai error minimum), sehingga bobot tersebut telah siap untuk digunakan pada proses pengenalan.

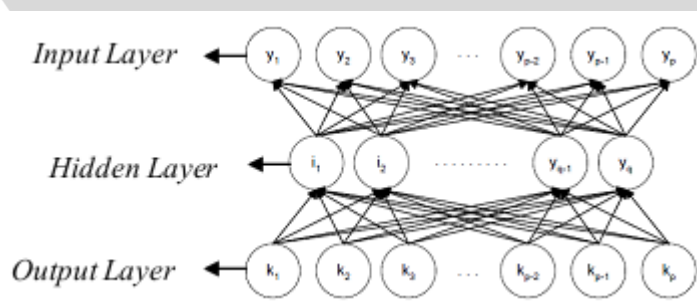
Adapun proses pengenalan tulisan tangan terbagi menjadi dua proses utama yaitu pre-processing dan pengenalan karakter. Proses pre-processing yaitu proses yang dimaksudkan untuk mengubah masukan agar sesuai dengan masukan yang dibutuhkan oleh jaringan. Proses pengenalan mempunyai masukan berupa gambar dari kata yang akan dikenali oleh jaringan, yang kemudian masukan tersebut dilakukan pre-processing untuk menyesuaikan masukan yang dibutuhkan jaringan. Hasil dari proses pre-processing akan diproses pada jaringan menggunakan bobot yang dihasilkan pada proses pelatihan. Hasil proses pengenalan tersebut adalah karakter-karakter yang merepresentasikan tiap karakter gambar masukan yang dapat dikenali komputer.

Diagram proses pengenalan tulisan tangan dapat dilihat pada gambar 3.5.



Gambar 3. 5 Diagram proses pengenalan tulisan tangan

Adapun arsitektur jaringan syaraf tiruan yang digunakan terdiri dari tiga buah layer, yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* terdiri dari $H \times W$ buah *neuron* yang merupakan array piksel tiap data latih. Misalkan $H \times W$ adalah 15×10 , maka terdapat 150 buah neuron pada input layer. Sedangkan jumlah neuron pada *hidden layer* adalah n , dimana nilai n akan diujicobakan beberapa nilai. Hal ini dilakukan untuk mendapatkan arsitektur jaringan yang sesuai untuk melakukan pengenalan. Sedangkan neuron pada output layer berjumlah 6. Arsitektur jaringan syaraf tiruan untuk pengenalan tulisan tangan bersambung dapat dilihat pada gambar 3.6.

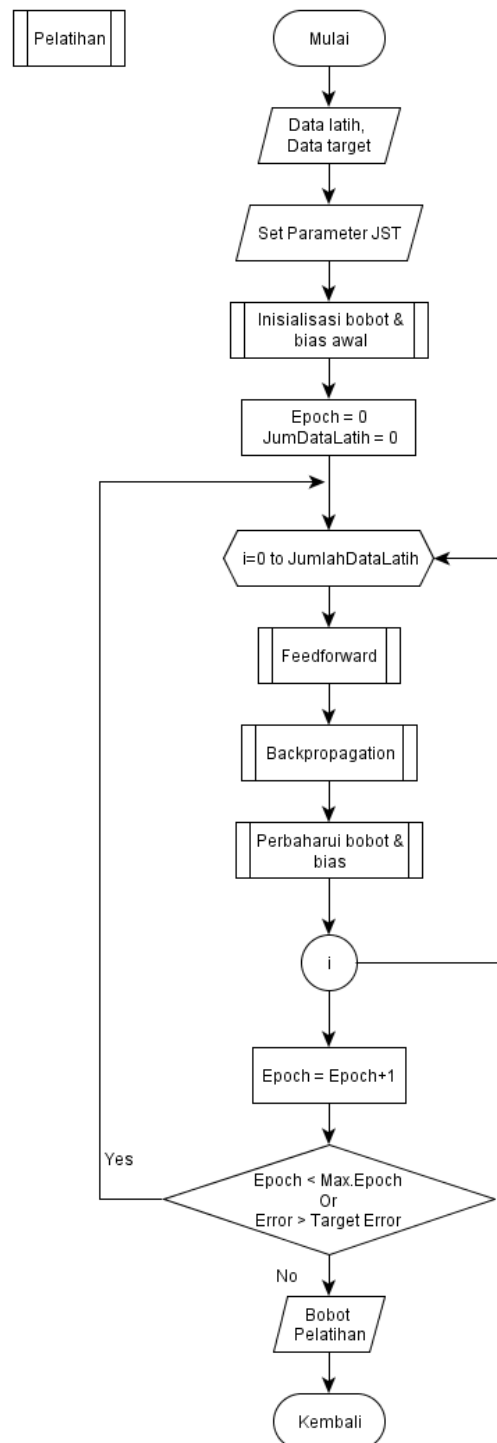


Gambar 3. 6 Arsitektur jaringan syaraf tiruan

3.5.1. Pelatihan Jaringan

Algoritma pelatihan Optical Backpropagation tidak jauh berbeda dengan pelatihan Backpropagation yang pada dasarnya terbagi menjadi 3 tahapan, yaitu langkah maju (*feedforward*), propagasi balik (*backpropagation*), dan perubahan bobot. Secara keseluruhan langkah-langkah pelatihan algoritma optical backpropagation 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 inialisasi bobot dan bias
5. Setting parameter iterasi awal (epoch) sama dengan 0
6. Proses *Feedforward*
7. Proses Backpropagation
8. Proses perbaharui bobot dan bias
9. Jika epoch kurang dari maksimum epoch atau error lebih dari target error maka lakukan langkah 6-8
10. bobot hasil pelatihan disimpan, data siap digunakan untuk proses pengujian (pengenalan).
11. Selesai



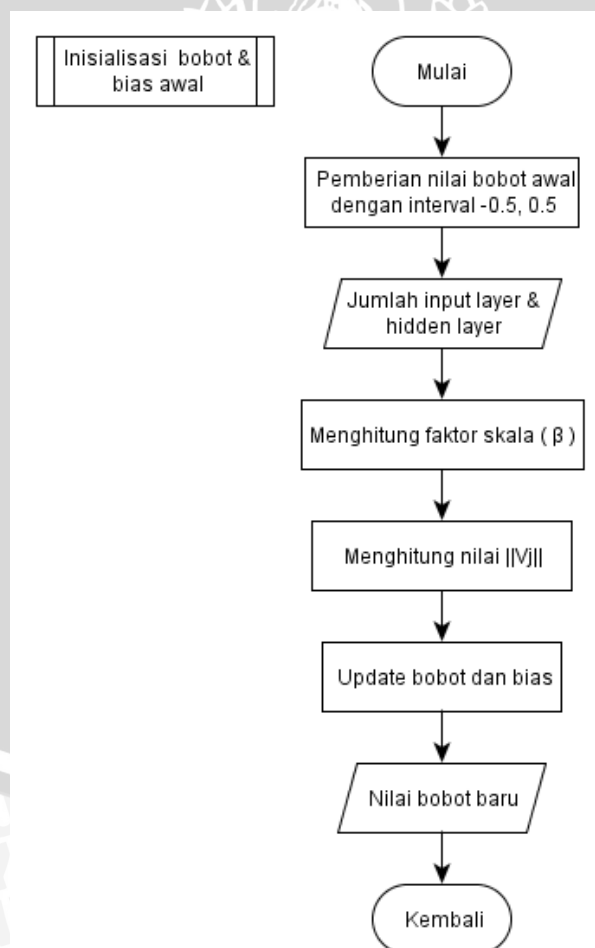
Gambar 3. 7 Diagram alir proses pelatihan jaringan

3.5.2. 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. Hitung faktor skala menggunakan rumus $\beta = 0.7\sqrt{P}$
5. Hitung nilai $\|V_j\|$ menggunakan persamaan 2.22
6. Perbaharui bobot (V_{ij}) menggunakan persamaan 2.23
7. Perbaharui bias dengan menggunakan bilangan acak antara $-\beta$ dan β .
8. Selesai.

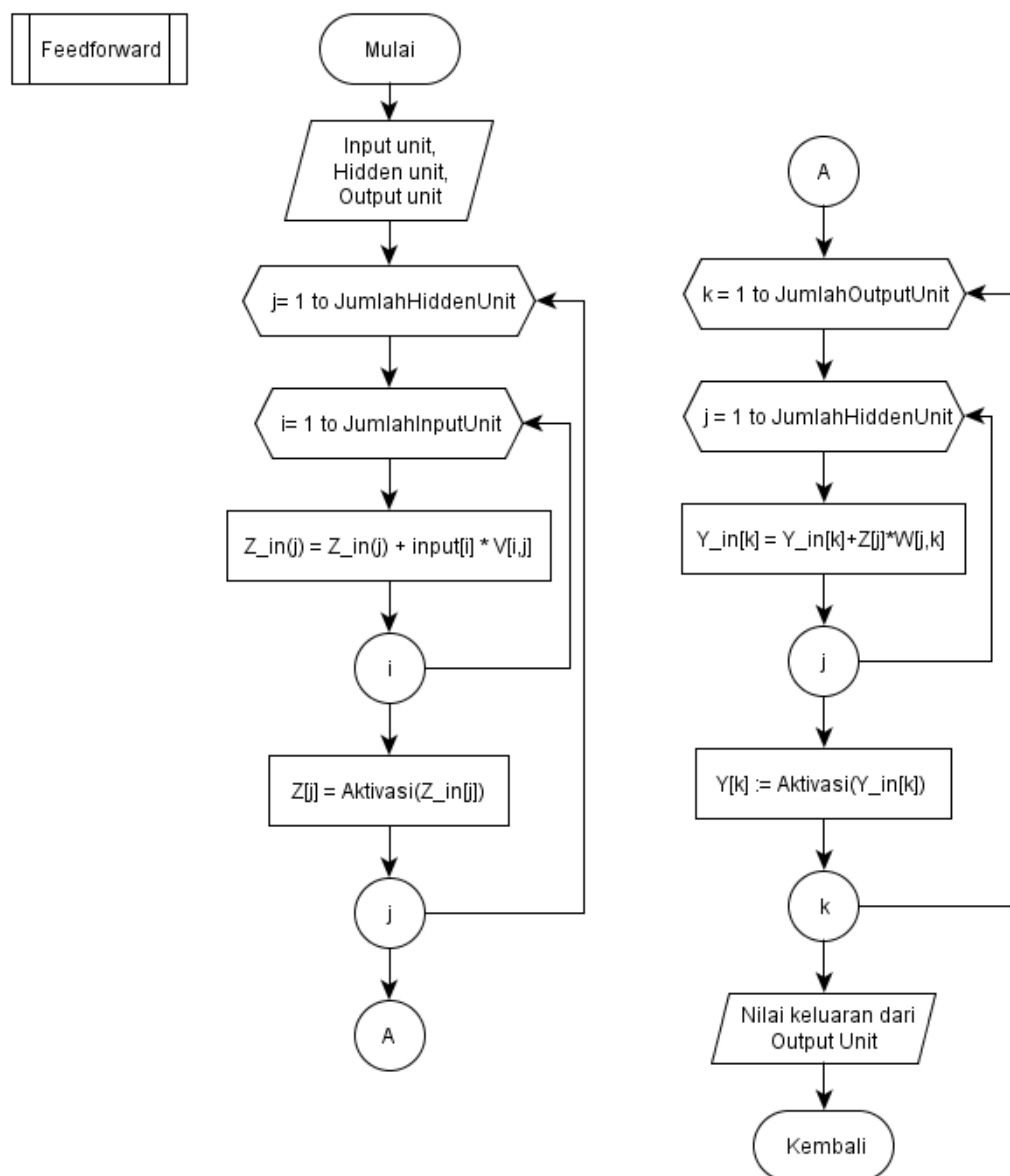


Gambar 3. 8 Diagram alir proses Inisialisasi Bobot

3.5.3. *Feedforward*

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 dengan 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 output pada lapis ini berada pada kisaran 0 dan 1.
4. Kalikan seluruh data hasil aktivasi masing-masing neuron hidden unit pada hidden layer 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 output pada lapis ini berada pada kisaran 0 dan 1.
6. Selesai



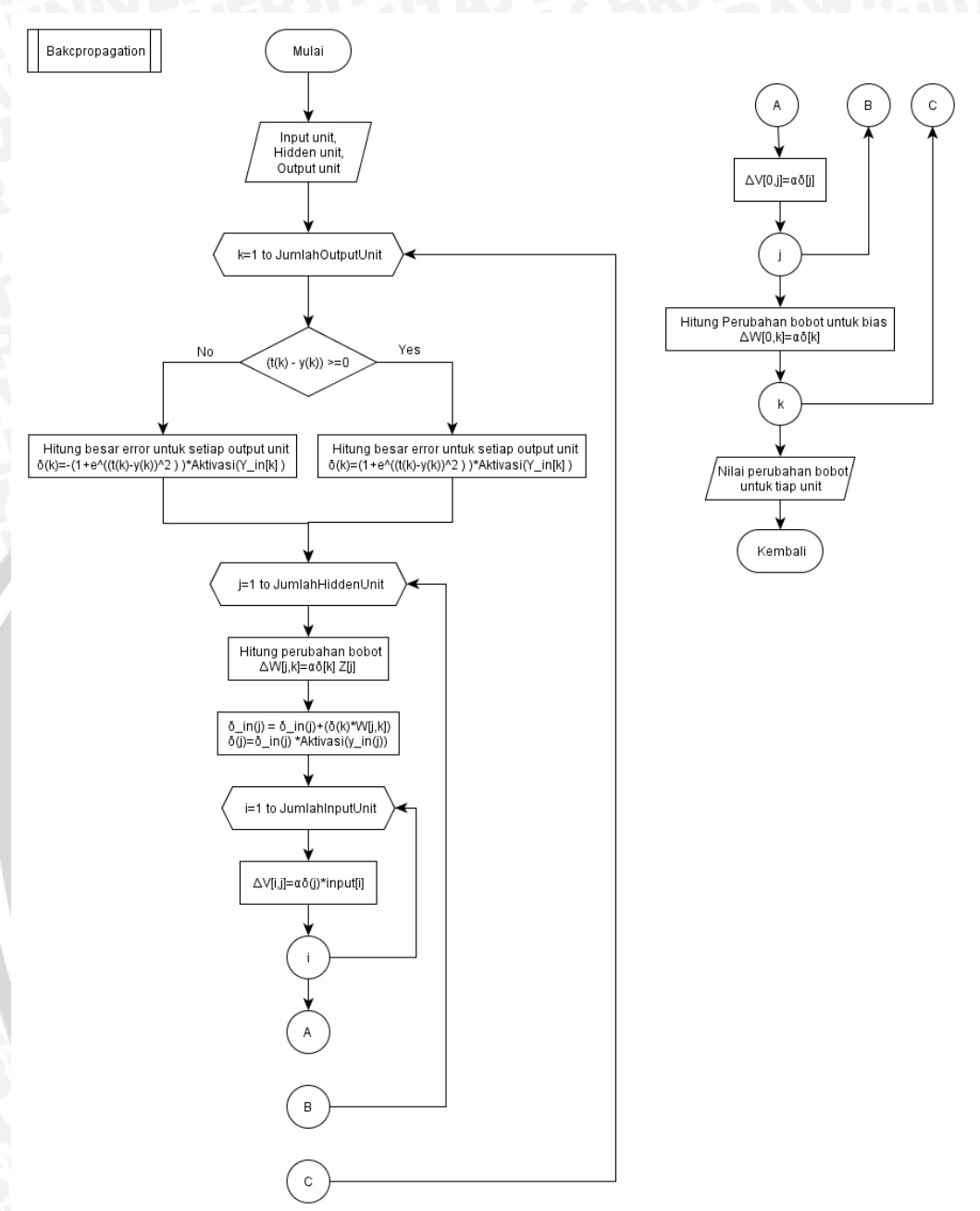
Gambar 3. 9 Diagram alir proses Feedforward

3.5.4. 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 back propagation adalah sebagai berikut:

1. Mulai
2. Pada output layer. Pertama hitung selisih antara target pelatihan dengan output. jika selisih tersebut bernilai positif maka gunakan persamaan 2.20 untuk menghitung informasi errornya, sedangkan jika bernilai negative maka gunakan persamaan 2.21. Hasil perhitungan tersebut 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.

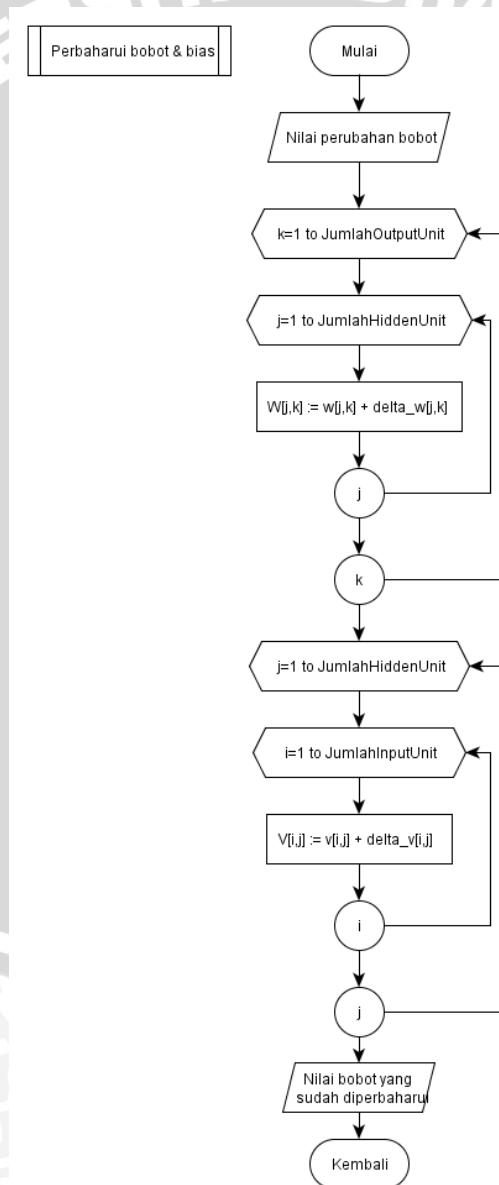


Gambar 3. 10 Diagram alir proses Backpropagation

3.5.5. 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 Perbaharui bobot dan bias diperlihatkan 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 Diagram alir proses Perbaharui Bobot dan Bias

3.5.6. Pengenalan Tulisan Tangan

Proses pengenalan tulisan tangan menggunakan jaringan syaraf tiruan dilakukan sesudah dilakukan pelatihan pada jaringan. akan tetapi sebelum dimasukkan ke dalam jaringan, data uji harus dinormalisasi dengan beberapa proses *pre-processing*, yaitu *Thresholding*, *Thinning*, Deteksi *Word Skew* dan *Baseline*. Secara keseluruhan langkah-langkah pengenalan tulisan tangan dapat dilihat pada gambar 3.12 dengan langkah-langkah sebagai berikut :

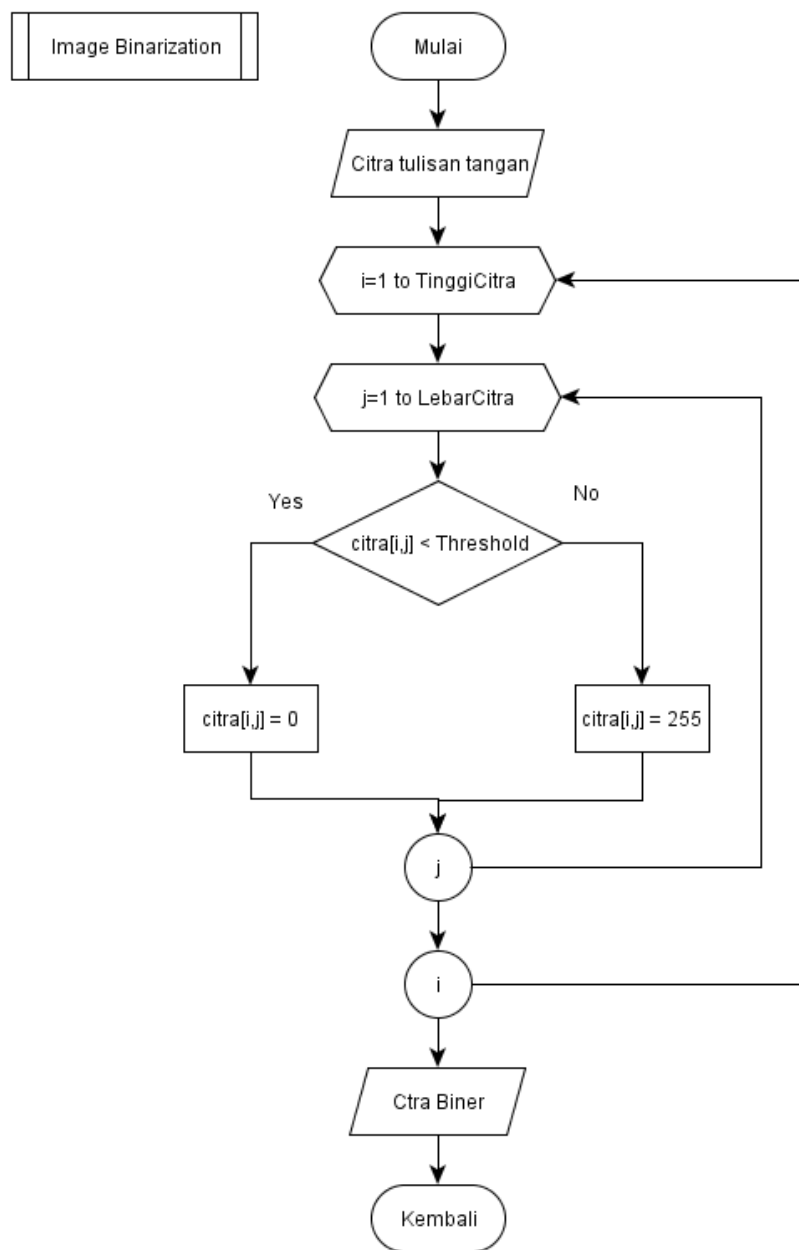
1. Mulai
2. Masukkan data uji berupa citra tulisan tangan
3. Proses *Thresholding*
4. Proses *Thinning*
5. Proses Deteksi *Word Skew* dan *Baseline*
6. Normalisasi ukuran citra
7. Segmentasi karakter dan pengenalan
8. Tampilkan hasil pengenalan
9. Selesai



Gambar 3. 12 Diagram alir pengenalan tulisan tangan

3.5.6.1. Image Binarization

Algoritma *Image Binarization* digunakan untuk merubah citra menjadi citra biner (hanya terdiri dari hitam (0) dan putih (255)). Pixel yang mendekati warna hitam akan di set menjadi hitam dan pixel yang mendekati warna putih akan di set menjadi putih sesuai dengan *threshold* tertentu yang telah ditentukan. Secara keseluruhan langkah-langkah proses *Image Binarization* dapat dilihat pada gambar 3.13.

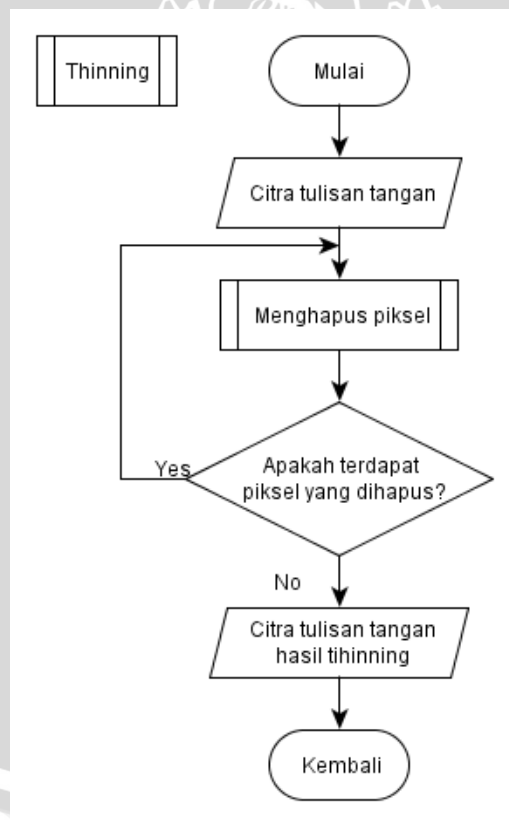


Gambar 3. 13 Diagram alir proses Image Binarization

3.5.6.2. *Thinning*

Algoritma *Thinning* digunakan untuk menipiskan garis (stroke) pada citra. Algoritma Thinning yang digunakan adalah Algoritma Thinning Zhang-suen yang prosesnya melibatkan dua sub iterasi. Secara keseluruhan langkah-langkah algoritma thinning dapat dilihat pada gambar 3.14 dengan langkah-langkah sebagai berikut:

1. Mulai
2. Masukkan citra tulisan tangan
3. Menghapus piksel
4. Jika masih terdapat piksel yang dihapus maka ulangi langkah 3-4
5. Simpan citra hasil thinning
6. Kembali

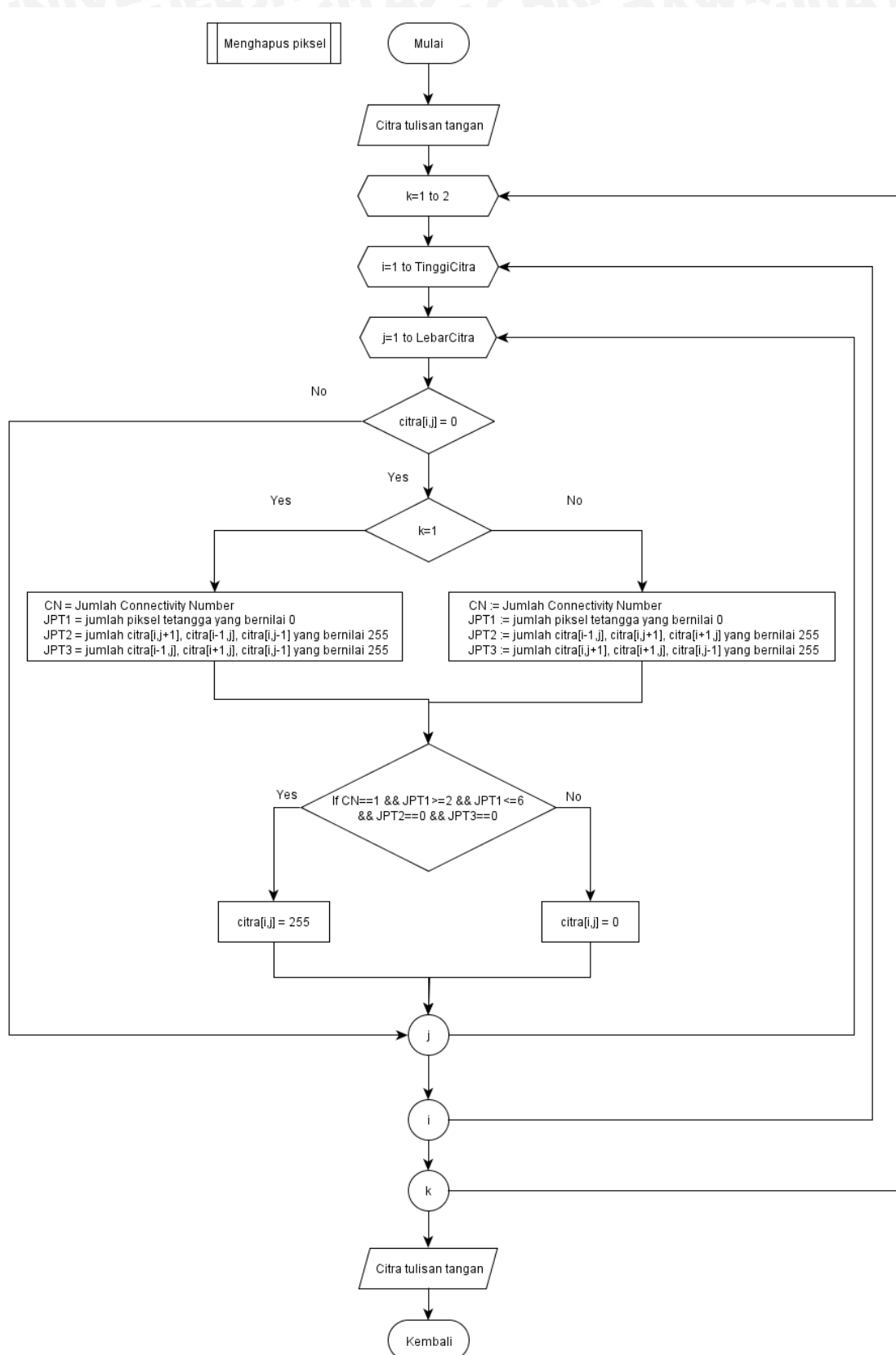


Gambar 3. 14 Diagram alir proses Thinning

3.5.6.3. Menghapus Piksel

Proses menghapus piksel dapat dilihat pada gambar 3.15 dengan langkah-langkah sebagai berikut :

1. Mulai
2. Sub-iterasi pertama
3. Jika piksel pada titik i,j ($citra[i,j]$) berwarna hitam (0)
4. Hitung jumlah connectivity number (CN)
5. Hitung jumlah piksel tetangga yang berwarna hitam (0) (JPT1).
6. Hitung $citra[i,j+1]$, $citra[i-1,j]$, $citra[i,j-1]$ yang berwarna putih (255) (JPT2).
7. Hitung $citra[i-1,j]$, $citra[i+1,j]$, $citra[i,j-1]$ yang berwarna putih (255) (JPT3).
8. Jika $CN=1$ dan $2 \leq JPT1 \leq 6$ dan $JPT2=0$ dan $JPT3=0$ maka set $citra[i,j]$ menjadi putih, jika tidak maka set menjadi hitam.
9. Sub-iterasi kedua
10. Jika piksel pada titik i,j ($citra[i,j]$) berwarna hitam (0)
11. Hitung jumlah connectivity number (CN)
12. Hitung jumlah piksel tetangga yang berwarna hitam (0) (JPT1).
13. Hitung $citra[i-1,j]$, $citra[i,j+1]$, $citra[i+1,j]$ yang berwarna putih (255) (JPT2).
14. Hitung $citra[i,j+1]$, $citra[i+1,j]$, $citra[i,j-1]$ yang berwarna putih (255) (JPT3).
15. Jika $CN=1$ dan $2 \leq JPT1 \leq 6$ dan $JPT2=0$ dan $JPT3=0$ maka set $citra[i,j]$ menjadi putih, jika tidak maka set menjadi hitam.
16. Kembali.

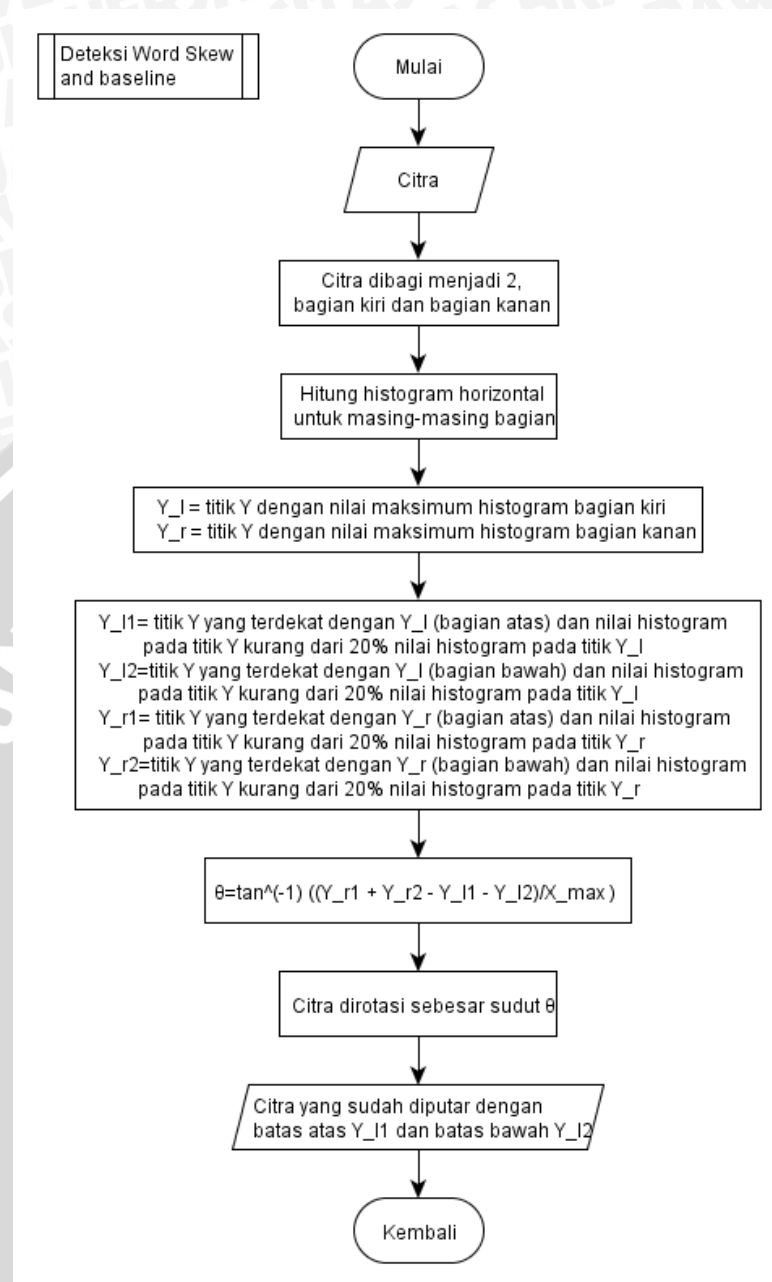


Gambar 3. 15 Diagram alir Proses menghapus piksel

3.5.6.4. Deteksi Word Skew dan Baseline

Deteksi *Word Skew* digunakan untuk mendeteksi kemiringan tulisan terhadap *baseline*. Dengan adanya proses ini didapatkan sudut kemiringan tulisan, dan kemudian citra tulisan dapat dirotasi sebesar sudut tersebut, sehingga citra tulisan sejajar dengan *baseline*. Secara keseluruhan langkah-langkah proses deteksi *Word Skew* dan *Baseline* dapat dilihat pada gambar 3.16 dengan langkah-langkah sebagai berikut:

1. Mulai
2. Masukkan citra tulisan tangan
3. Citra dibagi menjadi 2 bagian menurut garis horizontal, yaitu $0 - (X_{\max}/2)$ dan $(X_{\max}/2) - X_{\max}$
4. Hitung Horizontal Density Histogram untuk masing-masing bagian
5. Tentukan titik Y yang mempunyai nilai density tertinggi dari masing-masing bagian (Y_l dan Y_r)
6. Tentukan titik Y_{l1} yang terdekat dengan titik Y_l dari interval $[0, Y_l]$, dan nilai histogram titik tersebut kurang dari 20% nilai histogram pada titik Y_l .
7. Tentukan titik Y_{l2} yang terdekat dengan titik Y_l dari interval $[Y_l, Y_{\max}]$, dan nilai density titik tersebut kurang dari 20% nilai density pada titik Y_l .
8. Lakukan langkah 6 dan 7 untuk bagian yang kanan.
9. Hitung besar sudut kemiringan.
10. Citra dirotasi sebesar sudut yang didapatkan pada langkah 9
11. Citra sudah sejajar dengan baseline, dengan batas baseline bawah $Y_{l2} \approx Y_{r2}$ dan batas baseline atas $Y_{l1} \approx Y_{r1}$.
12. Kembali.

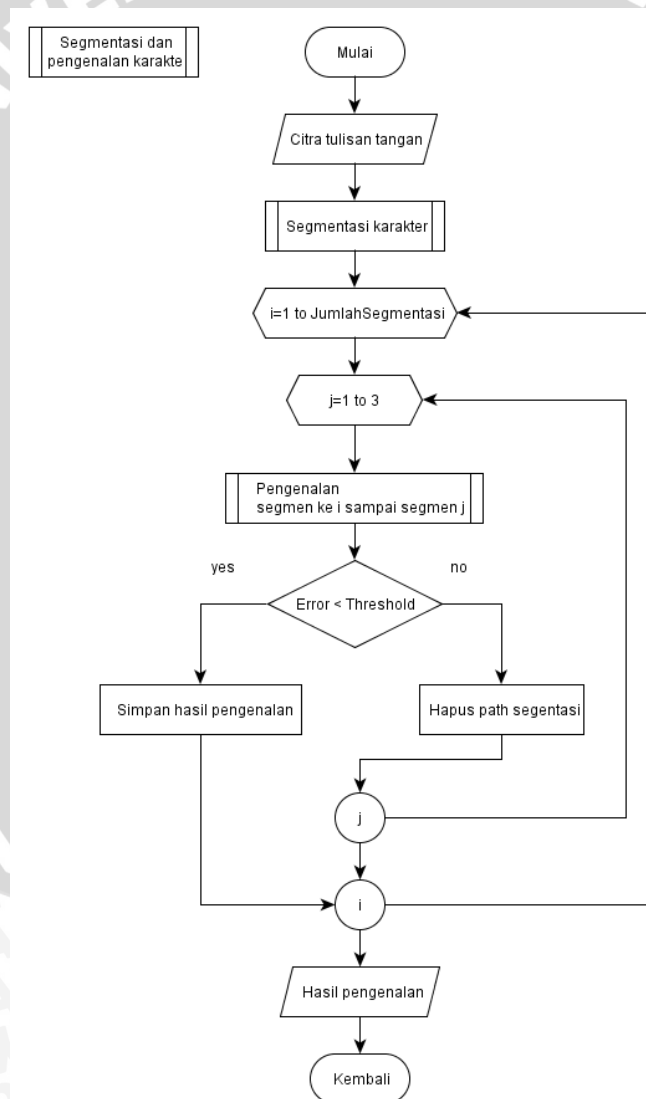


Gambar 3. 16 Diagram alir proses deteksi word skew dan baseline

3.5.6.5. Segmentasi Karakter dan Pengenalan

Proses Segmentasi karakter dan pengenalan dilakukan untuk menentukan titik segmetasi yang merepresentasikan karakter yang menyusun tulisan tangan tersebut, sekaligus melakukan pengenalan bahwa bagian segmentasi tersebut termasuk huruf apa. Proses segmentasi karakter dan pengenalan dapat dilihat pada gambar 3.17 dengan langkah-langkah sebagai berikut :

1. Mulai
2. Masukkan citra tulisan tangan
3. Proses segmentasi karakter
4. Lakukan pengenalan untuk segmentasi ke i.
5. Jika error pengenalan kurang dari threshold, maka segmentasi tersebut disimpan dan berlanjut ke langkah 6, jika tidak maka ulangi langkah 4-5 untuk segmentasi ke i dan (i+1), kemudian segmentasi ke i, (i+1), dan (i+2)
6. Lakukan langkah 4-5 sampai akhir segmentasi.
7. Kembali

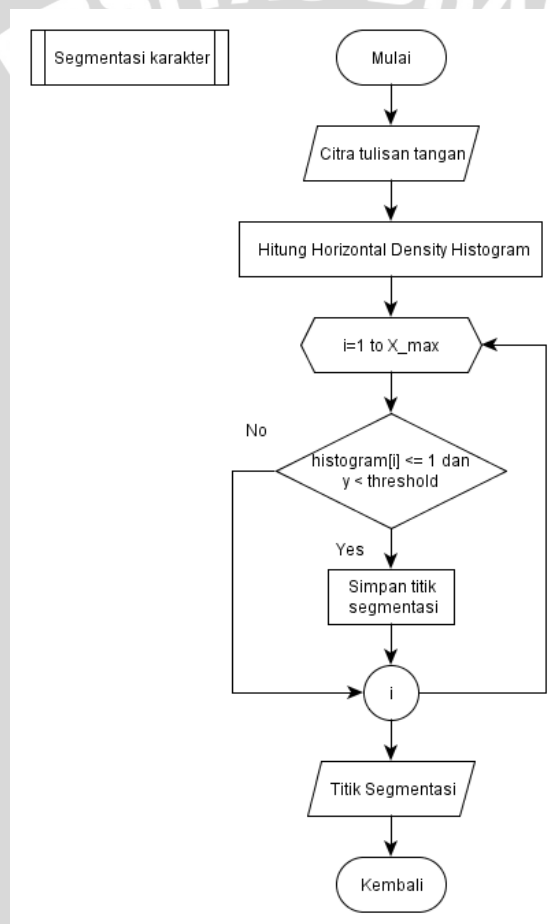


Gambar 3. 17 Diagram alir proses segmentasi dan pengenalan

3.5.6.6. Segmentasi Karakter

Proses segmentasi karakter dapat dilihat pada gambar 3.18 dengan langkah-langkah sebagai berikut:

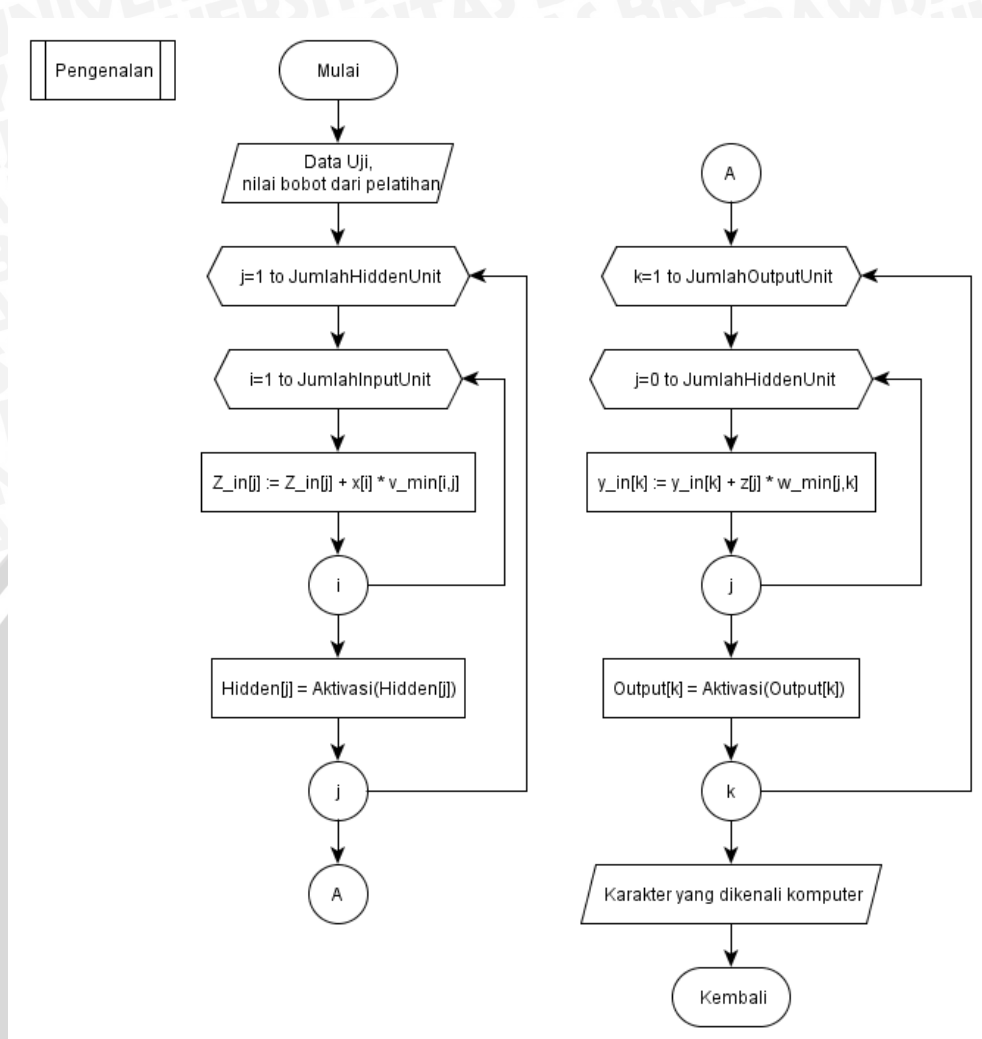
1. Mulai
2. Hitung horizontal density histogram
3. Jika nilai density pada titik $x \leq 1$ dan titik $y < \text{threshold}$ maka simpan titik segmentasi
4. Lakukan langkah 3 sampai x_{maks}
5. Kembali



Gambar 3. 18 Diagram alir proses segmentasi karakter

3.5.6.7. Pengenalan

Proses segmentasi karakter dapat dilihat pada gambar 3.19 yang mana proses pengenalan ini sama dengan proses *feedforward* yang dilakukan pada waktu pelatihan:



Gambar 3. 19 Diagram alir proses pengenalan karakter

3.6. Perancangan Uji Coba

3.6.1. Rancangan Uji Coba Pengaruh Nilai Learning Rate dan Jumlah Hidden Unit Terhadap Error

Uji coba ini dilakukan dengan menggunakan nilai *learning rate* yang berbeda, yaitu 0.1, 0.3, 0.5, 0.7, dan 0.9, dengan jumlah *hidden unit* yang beragam pula, yaitu 30, 50, 70, 90, 110. Kombinasi *learning rate* dan jumlah *hidden unit* dilakukan untuk mengetahui pengaruh nilai *learning rate* dan jumlah *hidden unit* terhadap error pelatihan. Rancangan uji coba ini akan disajikan seperti terlihat pada tabel 3.1

Tabel 3. 1 Pengaruh nilai learning rate dan jumlah hidden unit terhadap error pelatihan

LEARNING RATE	JUMLAH HIDDEN UNIT				
	30	50	70	90	110
0.1					
0.3					
0.5					
0.7					
0.9					

3.6.2. Rancangan Uji Coba Pengaruh Nilai Learning Rate dan Jumlah Hidden Unit Terhadap Waktu Pelatihan

Uji coba ini dilakukan dengan menggunakan nilai *learning rate* yang berbeda, yaitu 0.1, 0.3, 0.5, 0.7, dan 0.9, dengan jumlah *hidden unit* yang beragam pula, yaitu 30, 50, 70, 90, 110. Kombinasi *learning rate* dan jumlah *hidden unit* dilakukan untuk mengetahui pengaruh nilai *learning rate* dan jumlah *hidden unit* terhadap lama waktu pelatihan. Rancangan uji coba ini akan disajikan seperti terlihat pada tabel 3.2.

Tabel 3. 2 Pengaruh nilai learning rate dan jumlah hidden unit terhadap lama waktu pelatihan

LEARNING RATE	JUMLAH HIDDEN UNIT				
	30	50	70	90	110
0.1					
0.3					
0.5					
0.7					
0.9					

3.6.3. Rancangan Pengujian Sistem

3.6.3.1. Rancangan Uji Segmentasi

Uji coba ini dilakukan untuk mengetahui kemampuan sistem dalam melakukan proses segmentasi karakter pada citra tulisan tangan bersambung. Rancangan uji segmentasi akan disajikan seperti pada tabel 3.3.

Tabel 3. 3 Hasil Uji Segmentasi

No	Nama File	Text Asli	Segmentasi Benar	Persentase (%)

3.6.3.2. Rancangan Uji Pengenalan

Uji pengenalan merupakan proses pengujian terakhir untuk mengetahui tingkat keberhasilan sistem untuk melakukan pengenalan. Hasil pengenalan disajikan seperti pada tabel 3.4

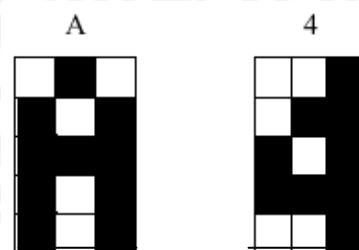
Tabel 3. 4 Hasil Uji Pengenalan

No	Nama File	Text Asli	Text yang dikenali	Pengenalan Benar	Persentase (%)

3.7. 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 3.20 Sedangkan Gambar 3.21 merupakan representasi dari pola karakter menggunakan kode 0 dan 1 yang digunakan untuk input pada contoh perhitungan.



Gambar 3. 20 Pola karakter

A			4		
0	1	0	0	0	1
1	0	1	0	1	1
1	1	1	1	0	1
1	0	1	1	1	1
1	0	1	0	0	1

Gambar 3. 21 Representasi pola karakter

Selanjutnya data inputan berupa matrik berukuran 3x5 diubah menjadi vektor seperti pada Tabel 3.5 yang digunakan untuk data masukan pertama yang akan dimasukkan pada iterasi pertama dengan targetnya yang dapat dilihat pada tabel 3.6. Sedangkan Tabel 3.7 untuk data masukan kedua pada iterasi kedua dengan targetnya yang ditunjukkan pada tabel 3.8.

Tabel 3. 5 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. 6 Target untuk data pertama

t1	t2
1	0

Tabel 3. 7 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. 8 Target untuk data kedua

t1	t2
0	1

3.7.1. Pelatihan Jaringan

Iterasi I

Data masukan pertama pada iterasi pertama seperti dapat dilihat pada Gambar 3.14 Dan targetnya ditunjukkan pada tabel 3.6

Inisialisasi Bobot

Inisialisasi bobot-bobot dari input layer ke hidden layer mula-mula diberi nilai acak antara -0.5 hingga 0.5. Nilai bobot awal (V_{ij}) dapat dilihat pada tabel 3.9.

Tabel 3. 9 Bobot dari input ke hidden layer (V_{ij})

V_{ij}	z_1	z_2	z_3	z_4	z_5
x_1	-0.1	0.2	-0.4	0.5	0.4
x_2	-0.3	0.1	0.3	0.2	0.3
x_3	-0.1	0.5	-0.2	-0.3	-0.3
x_4	0.3	-0.3	-0.4	-0.2	0.3
x_5	0.3	-0.2	-0.3	-0.1	-0.1
x_6	-0.3	0.1	0.3	-0.3	-0.1
x_7	0.1	0.3	0.3	-0.4	-0.2
x_8	-0.1	0.1	0.2	0.4	-0.4
x_9	-0.4	0	-0.5	0.1	0.2
x_{10}	-0.4	-0.1	-0.4	-0.1	-0.4
x_{11}	-0.2	-0.3	0.3	0.4	-0.4
x_{12}	-0.5	0.4	-0.2	0.2	0.1
x_{13}	-0.4	-0.3	-0.4	0.1	0.4
x_{14}	-0.4	0	0	-0.4	0.1
x_{15}	-0.1	-0.2	0.3	0	-0.1

Inisialisasi bobot baru dan bias ke hidden layer menggunakan metode Nguyen

$$\text{Widrow} : \beta = 0.7^{\frac{1}{\sqrt{P}}}$$

Jumlah input unit (n) = 15

Jumlah hidden unit (P) = 5

faktor skala = $\beta = 0.7^{\frac{1}{\sqrt{5}}} = 0.78$. Bias awal yang dipakai adalah bilangan acaka antara -0.78 hingga 0.78

Hitung nilai $\|v_j\|$ berdasarkan persamaan 2.22

$$\|v_1\| = \sqrt{(-0.1)^2 + (-0.3)^2 + (-0.1)^2 + (0.3)^2 + (0.3)^2 + (-0.3)^2 + (0.1)^2 + (-0.1)^2 + (-0.4)^2 + (-0.4)^2 + (-0.2)^2 + (-0.5)^2 + (-0.4)^2 + (-0.4)^2 + (-0.1)^2}$$

$$= 1.157584$$

dengan cara yang sama untuk menghitung $\|v_2\|, \|v_3\|, \|v_4\|, \|v_5\|$

Nilai V_j dapat dilihat pada tabel 3.10

Tabel 3. 10 Inisialisasi nilai V_j

j	V_j
1	1.157584
2	0.96436
3	1.24499
4	1.109054
5	1.095445

Dengan menggunakan persamaan 2.23, diperoleh nilai bobot yang digunakan untuk inisialisasi V_{ij} (baru) :

$$v_{1,1} = \frac{\beta * v_{1,1}(\text{lama})}{\|v_1\|} = \frac{0.78 * (-0.1)}{1.157584} = -0.067320$$

dengan cara yang sama untuk menghitung $v_{1,2}, v_{1,3}, \dots, v_{15,4}, v_{15,5}$

Nilai bobot baru yang digunakan untuk inisialisasi input layer ke hidden layer (V_{ij}) dapat dilihat pada tabel 3.11

Tabel 3. 11 Nilai V_{ij} yang baru

V_{ij}	z_1	z_2	z_3	z_4	z_5
x1	-0.067320	0.161616	-0.250375	0.351329	0.284554
x2	-0.201960	0.080808	0.187781	0.140531	0.213416
x3	-0.067320	0.404040	-0.125187	-0.210797	-0.213416
x4	0.201960	-0.242424	-0.250375	-0.140531	0.213416
x5	0.201960	-0.161616	-0.187781	-0.070266	-0.071139
x6	-0.201960	0.080808	0.187781	-0.210797	-0.071139
x7	0.067320	0.242424	0.187781	-0.281063	-0.142277
x8	-0.067320	0.080808	0.125187	0.281063	-0.284554
x9	-0.269280	0.000000	-0.312968	0.070266	0.142277
x10	-0.269280	-0.080808	-0.250375	-0.070266	-0.284554
x11	-0.134640	-0.242424	0.187781	0.281063	-0.284554

x12	-0.336600	0.323232	-0.125187	0.140531	0.071139
x13	-0.269280	-0.242424	-0.250375	0.070266	0.284554
x14	-0.269280	0.000000	0.000000	-0.281063	0.071139
x15	-0.067320	-0.161616	0.187781	0.000000	-0.071139

Nilai bias V_{0j} yang dipakai adalah bilangan acak antara $-\beta$ hingga β yaitu antara nilai -0.78 hingga 0.78. Nilai V_{0j} dapat dilihat pada tabel 3.12.

Tabel 3. 12 Nilai bias (V_{0j})

	V_{0j}
1	0.5874
2	-0.7022
3	-0.2315
4	0.4899
5	-0.5924

Nilai bobot awal dari hidden layer ke output layer (W_{jk}) diperoleh dari nilai acak antara -0.5 sampai 0.5, inisialisasi nilai W_{jk} dapat dilihat pada tabel 3.13

Tabel 3. 13 Nilai bobot ke output layer (W_{jk})

W_{jk}	Y_1	Y_2
z_1	0.4	-0.5
z_2	-0.5	0.3
z_3	0.2	-0.2
z_4	0.2	-0.1
z_5	0.4	-0.4

Nilai bias W_{0k} juga diperoleh dari nilai acak antara -0.5 hingga 0.5, inisialisasi nilai bias W_{0k} dapat dilihat pada tabel 3.14

Tabel 3. 14 Nilai bias W_{0k}

	W_{0k}
1	-0.4
2	0.2

Proses Feedforward

Hitung keluaran di unit hidden (Z_j) menggunakan persamaan 2.6

Hasil nilai operasi hidden Z_{in} dapat dilihat pada tabel 3.15.

$$Z_{in1} = 0.5874 + [(0 * -0.067320) + (1 * -0.201960) + (0 * -0.067320) + (1 * 0.201960) + (0 * 0.201960) + (1 * -0.201960) + (1 * 0.067320) + \dots + (0 * -0.269280) + (1 * -0.067320)] = -0.8263$$

dengan cara yang sama untuk menghitung Z_{in2} , Z_{in3} , Z_{in4} dan Z_{in5}

Tabel 3. 15 Hasil operasi pada hidden (Z_{in})

j	Z_{in}
1	-0.8263
2	-0.6214
3	-0.5445
4	0.4899
5	-0.5213

Menghitung aktivasinya menggunakan persamaan 2.7. Hasil aktivasi unit hidden (Z_{in}) dapat dilihat pada tabel 3.16

Tabel 3. 16 Hasil aktivasi Z_{in}

j	Z
1	0.3044
2	0.3495
3	0.3671
4	0.6201
5	0.3726

Selanjutnya yaitu menghitung keluaran unit output (Y_k) dengan menggunakan persamaan 2.8. Hasil operasi pada unit output dapat dilihat pada tabel 3.17

$$Y_{in1} = -0.4 + [(0.3044 * 0.4) + (0.3495 * -0.5) + (0.3671 * 0.2) + (0.6201 * 0.2) + (0.3726 * 0.4)] = -0.10649$$

Tabel 3. 17 Hasil operasi pada output (Y_{in})

k	Y_{in}
1	-0.10649
2	-0.13183

Kemudian hitung nilai aktivasi Y_{in} . Hasil aktivasi Y_{in} dapat dilihat pada tabel 3.18

Tabel 3. 18 Hasil aktivasi Y_{in}

k	Y
1	0.4734
2	0.4671

Proses Backpropagation

Hitung nilai faktor δ di unit output menggunakan persamaan 2.10. Nilai faktor δ di unit output dapat dilihat pada tabel 3.19

Tabel 3. 19 Nilai error Y_k

k	δ
1	0.5486
2	-0.5314

Hitung nilai perubahan bobot dengan $\alpha = 0.2$ dengan menggunakan persamaan 2.11. Hasil perhitungan nilai perubahan bobot (ΔW_{jk}) dapat dilihat pada tabel 3.20

Tabel 3. 20 Nilai Perubahan bobot (ΔW_{jk})

ΔW_{jk}	w_1	w_2
z_1	0.0334	-0.0324
z_2	0.0383	-0.0371
z_3	0.0403	-0.0390
z_4	0.0680	-0.0659
z_5	0.0409	-0.0396

Hitung juga perubahan bobot bias ΔW_{0k} menggunakan persamaa 3.12. Hasil perhitungan nilai perubahan bobot ΔW_{0k} dapat dilihat pada tabel 3.21

Tabel 3. 21 Perubahan bobot bias ΔW_{0k}

k	w_0
1	0.1097
2	-0.1063

Hitung penjumlahan kesalahan di unit hidden menggunakan persamaan 2.13.

Hasil perhitungan penimbang di unit hidden dapat dilihat pada tabel 3.22

Tabel 3. 22 Faktor penimbang di unit hidden

j	δ_{in}
1	0.4851
2	-0.4337
3	0.2160
4	0.1629
5	0.4320

Kemudia hitung aktivasi faktor kesalahan δ di unit hidden menggunakan persamaan 2.14. Hasil perhitungan aktivasi faktor dapat dilihat pada tabel 3.23

Tabel 3. 23 Aktivasi faktor kesalahan δ

j	δ
1	0.1185
2	-0.1052
3	0.0522
4	0.0370
5	0.1043

Hitung perubahan bobot ke unit hidden ΔV_{ij} dengan menggunakan rumus 2.15.

Hasil perhitungan dapat dilihat pada tabel 3.24

Tabel 3. 24 Perubahan bobot ke unit hidden ΔV_{ij}

ΔV_{ij}	ΔV_1	ΔV_2	ΔV_3	ΔV_4	ΔV_5
x_1	0.000000	0.000000	0.000000	0.000000	0.000000
x_2	0.023704	-0.021037	0.010444	0.007408	0.020868
x_3	0.000000	0.000000	0.000000	0.000000	0.000000
x_4	0.023704	-0.021037	0.010444	0.007408	0.020868
x_5	0.000000	0.000000	0.000000	0.000000	0.000000
x_6	0.023704	-0.021037	0.010444	0.007408	0.020868

X ₇	0.023704	-0.021037	0.010444	0.007408	0.020868
X ₈	0.023704	-0.021037	0.010444	0.007408	0.020868
X ₉	0.023704	-0.021037	0.010444	0.007408	0.020868
X ₁₀	0.023704	-0.021037	0.010444	0.007408	0.020868
X ₁₁	0.000000	0.000000	0.000000	0.000000	0.000000
X ₁₂	0.023704	-0.021037	0.010444	0.007408	0.020868
X ₁₃	0.023704	-0.021037	0.010444	0.007408	0.020868
X ₁₄	0.000000	0.000000	0.000000	0.000000	0.000000
X ₁₅	0.023704	-0.021037	0.010444	0.007408	0.020868

Hitung perubahan bobot bias ΔV_{0j} dengan menggunakan rumus 2.16. Hasil perhitungan dapat dilihat pada tabel 3.25

Tabel 3. 25 perubahan bobot bias ΔV_{0j}

	ΔV_{0j}
Z ₁	0.023703972
Z ₂	-0.021037114
Z ₃	0.010444186
Z ₄	0.007407767
Z ₅	0.020867726

Perbaharui bobot di unit hidden menggunakan persamaan 2.18. Hasil perhitungan dapat dilihat pada tabel 3.26

Tabel 3. 26 Bobot baru (V_{ij})

V_{ij}	z ₁	z ₂	z ₃	z ₄	z ₅
x ₁	-0.067320	0.161616	-0.250375	0.351329	0.284554
x ₂	-0.178256	0.059771	0.198225	0.147939	0.234284
x ₃	-0.067320	0.404040	-0.125187	-0.210797	-0.213416
x ₄	0.225664	-0.263461	-0.239930	-0.133124	0.234284
x ₅	0.201960	-0.161616	-0.187781	-0.070266	-0.071139
x ₆	-0.178256	0.059771	0.198225	-0.203389	-0.050271
x ₇	0.091024	0.221387	0.198225	-0.273655	-0.121410
x ₈	-0.043616	0.059771	0.135631	0.288471	-0.263687
x ₉	-0.245576	-0.021037	-0.302524	0.077673	0.163145
x ₁₀	-0.245576	-0.101845	-0.239930	-0.062858	-0.263687
x ₁₁	-0.134640	-0.242424	0.187781	0.281063	-0.284554
x ₁₂	-0.312896	0.302195	-0.114743	0.147939	0.092006
x ₁₃	-0.245576	-0.263461	-0.239930	0.077673	0.305422
x ₁₄	-0.269280	0.000000	0.000000	-0.281063	0.071139
x ₁₅	-0.043616	-0.182653	0.198225	0.007408	-0.050271

Kemudian perbaiki bias V_{0j} . Nilai V_{0j} dapat dilihat pada tabel 3.27.

Tabel 3. 27 Nilai V_{0j} baru

j	V_{0j}
1	0.611103972
2	-0.723237114
3	-0.221055814
4	0.497307767
5	-0.571532274

Perbaharui bobot di unit output, dengan menggunakan persamaan 2.17. Hasil perhitungan dapat dilihat pada tabel 3.28

Tabel 3. 28 Perubahan bobot W_{jk}

W_{jk}	Y_1	Y_2
z_1	0.433400114	-0.532356229
z_2	-0.461658193	0.262856525
z_3	0.240282003	-0.239023032
z_4	0.268032856	-0.165906562
z_5	0.440875406	-0.439597888

Kemudian perbaiki nilai bobot bias W_{0k} . Hasil perhitungan dapat dilihat pada tabel 3.29

Tabel 3. 29 Nilai bobot bias W_{0k} baru

k	W_{0k}
1	-0.290284253
2	0.093713301

Pada iterasi ke-dua lakukan hal yang sama seperti pada iterasi pertama dengan data yang dipakai adalah data hasil perhitungan pada iterasi pertama. Lakukan proses tersebut sampai mencapai maksimum epoch atau minimum error yang telah ditentukan di awal.

3.7.2. Pengujian Jaringan

Pada pengujian jaringan, proses yang dipakai hanya *feedforward*, yang langkah-langkah prosesnya seperti pada pelatihan jaringan, yang kemudian hasil output disamakan dengan kode output sebenarnya untuk mengetahui hasil karakter yang dikenali.

3.8. Perancangan Antarmuka

3.8.1. Form Pelatihan

Form pelatihan digunakan untuk proses pelatihan jaringan syaraf tiruan, optical backpropagation. Pelatihan jaringan ini akan menghasilkan bobot yang nantinya digunakan pada proses pengenalan. Form pelatihan akan disajikan seperti terlihat pada gambar 3.22

Gambar 3. 22 Form pelatihan

Keterangan:

A: Menu parameter, dimana pengguna dapat memasukkan parameter-parameter untuk proses pelatihan jaringan, seperti maksimum epoch, minimum error, jumlah hidden unit, dan besar learning rate.

B: Tombol untuk memulai proses pelatihan

C: Tombol untuk menyimpan hasil pelatihan

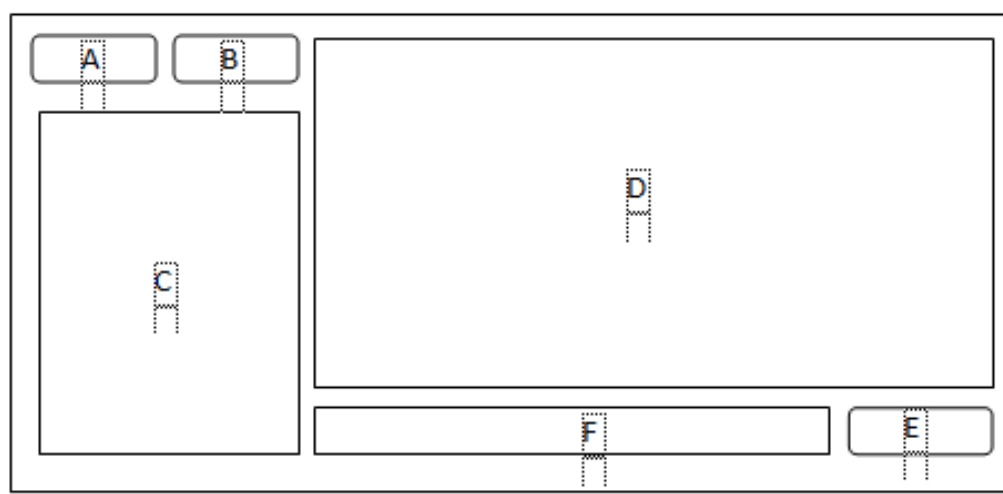
D: Tempat untuk menampilkan data pelatihan

E: Tempat untuk alamat file data pelatihan

F: Tombol untuk memasukkan data pelatihan

3.8.2. Form Pengenalan

Form pengenalan digunakan untuk proses pengenalan yang dilakukan setelah dilakukan pelatihan pada form pelatihan. Form pengenalan akan disajikan seperti terlihat pada gambar 3.23



Gambar 3. 23 Form Pengenalan

Keterangan:

A : Tombol untuk memasukkan data bobot hasil pelatihan

B : Tombol untuk memasukkan data pengujian

C : Tempat untuk menampilkan data pengujian

D: Tombol untuk menampilkan gambar tulisan tangan

E : Tombol untuk proses pengenalan

F : Tempat untuk hasil pengenalan

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1. Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalam sub bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1. Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan perangkat lunak ini adalah :

1. Intel® Core™ 2 Duo CPU E7500 @ 2.93GHz
2. Memori 3 GB Harddisk 258 GB
3. Monitor 14.6 inches
4. Keyboard
5. Mouse

4.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan aplikasi pengenalan tulisan tangan bersambung ini adalah :

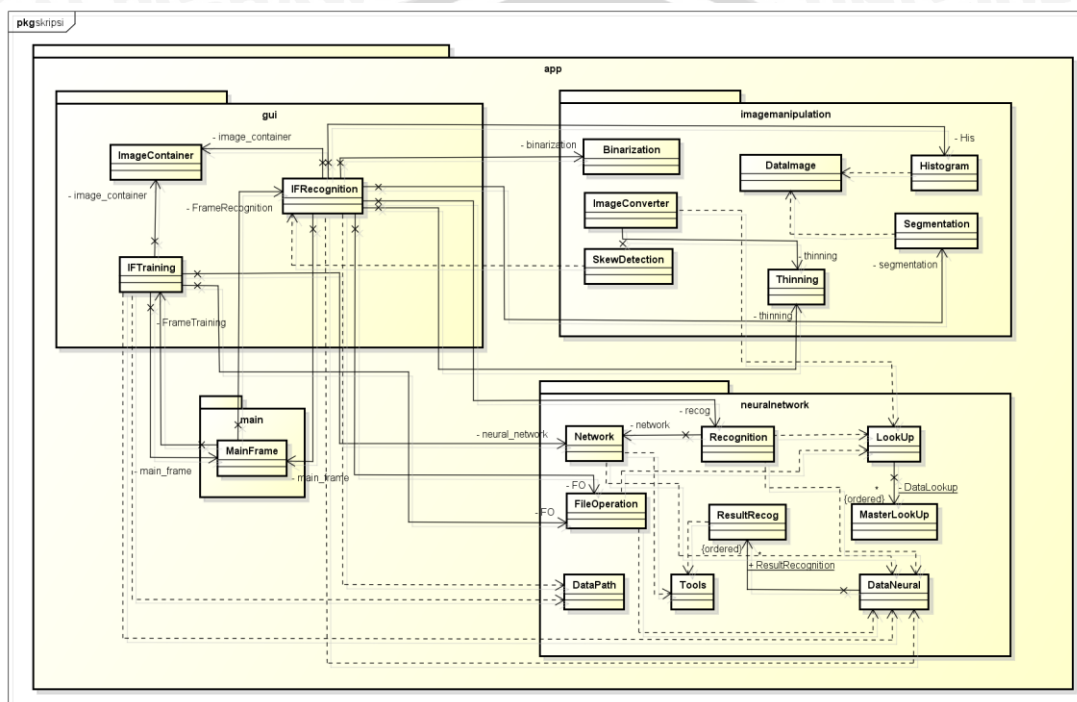
1. Sistem operasi windows 7 Professional 32 bit
2. Netbeans 7.2
3. Java Development Kit (JDK) 1.7.0_03
4. GIMP 2.6

4.2. Implementasi Program

Pada subbab implementasi program ini akan dijelaskan mengenai implementasi dari rancangan perangkat lunak yang meliputi implementasi kelas dan implementasi antarmuka.

4.2.1. Implementasi Kelas

Implementasi kelas yang digunakan, ditulis dengan menggunakan bahasa pemrograman Java. Kelas - kelas yang dibuat diletakkan terpisah menurut kategori fungsinya ke dalam package – package. Adapun package-package yang dibuat adalah package gui, imagemanipulation, dan neuralnetwork.



Gambar 4. 1 Class Diagram aplikasi pengenalan tulisan tangan

4.2.1.1. Package gui

Package gui ini berfungsi sebagai tempat menyimpan kelas-kelas yang berhubungan dengan graphical user interface (GUI), seperti form, frame, panel, table, dan lain sebagainya. Adapun kelas-kelas utama yang berada pada package gui dapat dilihat pada tabel 4.1.

Tabel 4. 1 Kelas-kelas yang terdapat pada package gui

No	Nama Kelas	Keterangan
1	ImageContainer	Kelas ImageContainer merupakan turunan dari kelas javax.swing. JPanel, yang mana kelas ini berfungsi sebagai tempat untuk menampilkan gambar dan operasi-operasi yang berhubungan dengan gambar tersebut.
2	IFTraining	Kelas IFTraining merupakan turunan dari kelas

		javax.swing.JFrame, yang mana kelas ini berfungsi sebagai tempat form-form untuk proses pelatihan.
3	IFRecognition	Kelas IFRecognition merupakan turunan dari kelas javax.swing.JFrame, yang mana kelas ini berfungsi sebagai tempat form-form untuk proses pengenalan tulisan tangan.

1. Kelas ImageContainer

Kelas ImageContainer merupakan turunan dari kelas javax.swing.JPanel. kelas javax.swing.JPanel sendiri merupakan kelas yang sudah disediakan oleh Java yang berfungsi sebagai container untuk komponen-komponen yang lain, seperti button, textfield, dan lain sebagainya. Kelas ini berfungsi sebagai tempat untuk menampilkan gambar beserta menangani proses-proses yang bersangkutan dengan gambar tersebut seperti menentukan batas kata, dan lain sebagainya.

Adapun method-method yang terdapat pada kelas ImageContainer dapat dilihat pada gambar 4.2, sedangkan untuk penjelasan fungsi method-method utama dapat dilihat pada tabel 4.2.

ImageContainer
- X_KIRI : int = -1 - X_KANAN : int = -1 - Y_ATAS : int = -1 - Y_BAWAH : int = -1 + ImageContainer() + setImage(Uri : String) : void + setImage(width : int, height : int) : void + repaintSource() : void + CleanCanvas() : void + getWorkImage() : BufferedImage + RemoveWorkImage() : void + setWorkImage() : void + setWorkImage(inImage : BufferedImage) : BufferedImage - DetermineWorkImage() : void + DetermineWorkImage(inImage : BufferedImage) : int[] + DrawWorkImage() : void + DrawBoundary() : void + getBound() : int[] + getImage() : BufferedImage + getPartImage(fullImage : BufferedImage, FirstX : int, LastX : int) : BufferedImage + getPartImage(fullImage : BufferedImage, PositionX : int[]) : BufferedImage[] + DrawImage(inImage : BufferedImage[]) : void + DrawPreviewTrain(DataTrain : String[], split : int) : void + paintComponent(g : Graphics) : void

Gambar 4. 2 Kelas ImageContainer

Tabel 4. 2 Method – method yang terdapat pada kelas ImageContainer

No	Nama Method	Keterangan
1	Public ImageContainer()	Method ImageContainer merupakan constructor dari kelas ImageContainer, yang mana nama method-nya sama dengan nama kelasnya.
2	Public void setImage(String url)	Method ini berfungsi untuk menempatkan gambar pada ImageContainer sesuai dengan alamat gambar yang diberikan.
3	private void DetermineWorkImage ()	Method ini berfungsi untuk menentukan batasan kata, sehingga area yang harus diproses tidak seluruh panel, tetapi hanya pada gambar kata tersebut.
4	public BufferedImage getPartImage(BufferedImage fullImage, int FirstX, int LastX)	Method ini berfungsi untuk mengambil sebagian gambar menurut titik x awal (FirstX) sampai titik x akhir (LastX).
5	public void DrawPreviewTrain(String [] DataTrain, int split)	Method ini berfungsi untuk menggambar data training, sehingga data training dapat divisualisasi menjadi sebuah karakter.

2. Kelas IFTraining

Kelas IFTraining merupakan turunan dari kelas javax.swing.JInternalFrame. Kelas javax.swing.JInternalFrame sendiri merupakan kelas yang sudah disediakan oleh Java yang digunakan untuk membuat internal frame, yaitu frame yang berada di dalamnya frame. Method-method yang berada pada kelas IFTraining merupakan aksi dari komponen yang ada pada frame tersebut, seperti aksi pada tombol, aksi mouse click, dll. kelas ini digunakan untuk menampilkan form-form yang ditujukan untuk proses pelatihan. Adapun method-method yang terdapat pada kelas IFTraining dapat dilihat pada gambar 4.3.

IFTraining
- Input : ArrayList<String[]> - Target : ArrayList<String[]>
+ IFTraining() - initComponents() : void - JTFOpenFileMouseClicked(evt : MouseEvent) : void - JBLoadActionPerformed(evt : ActionEvent) : void - JTDataTrainingMouseClicked(evt : MouseEvent) : void - JBStartActionPerformed(evt : ActionEvent) : void - JBSaveActionPerformed(evt : ActionEvent) : void - JBCancelActionPerformed(evt : ActionEvent) : void

Gambar 4. 3 Kelas IFTraining

3. Kelas IFRecognition

Kelas IFRecognition seperti pada kelas IFTraining, akan tetapi kelas IFRecognition digunakan untuk menampilkan form-form yang ditujukan untuk proses pengujian (pengenalan tulisan tangan). Adapun method-method yang terdapat pada kelas IFRecognition dapat dilihat pada gambar 4.4.

IFRecognition
+ IFRecognition() - initComponents() : void - initPopupMenu() : void - JBLoadWeightActionPerformed(evt : ActionEvent) : void - JBLoadTrialActionPerformed(evt : ActionEvent) : void - formComponentResized(evt : ComponentEvent) : void - JTDDataTrialValueChanged(evt : TreeSelectionEvent) : void - JBRognitionActionPerformed(evt : ActionEvent) : void - JMIgetPartImageActionPerformed(evt : ActionEvent) : void - JMISplitImageActionPerformed(evt : ActionEvent) : void

Gambar 4. 4 Kelas IFRecognition

4.2.1.2. Package imagemanipulation

Package imagemanipulation ini berfungsi sebagai tempat menyimpan kelas-kelas yang berhubungan dengan pengolahan citra digital, seperti proses Thinning, Binarization, dan lain sebagainya. Adapun kelas-kelas utama yang berada pada package imagemanipulation dapat dilihat pada tabel 4.3.

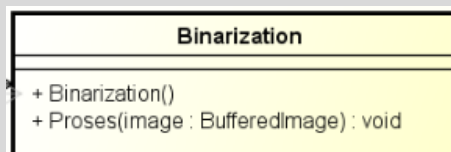
Tabel 4. 3 Kelas-kelas yang terdapat pada package imagemanipulation

No	Nama Kelas	Keterangan
1	Binarization	Kelas Binarization merupakan kelas yang berfungsi untuk merubah suatu gambar <i>grayscale</i> menjadi gambar biner.
2	DataImage	Kelas DataImage merupakan kelas yang digunakan untuk menyimpan data-data tentang gambar, seperti data histogram, <i>baseline</i> .
3	Thinning	Kelas Thinning merupakan kelas yang digunakan untuk melakukan proses <i>thinning</i> pada gambar biner.
4	Segmentation	Kelas Segmentation merupakan kelas yang digunakan untuk melakukan proses segmentasi gambar kata menjadi gambar huruf-huruf.
5	SkewDetection	Kelas SkewDetection merupakan kelas yang digunakan untuk melakukan proses penentuan <i>baseline</i> dari suatu kata pada gambar.

6	Histogram	Kelas Histogram merupakan kelas yang digunakan untuk melakukan proses perhitungan histogram dari suatu gambar, baik <i>horizontal histogram</i> maupun <i>vertical histogram</i> .
---	-----------	--

1. Kelas Binarization

Kelas Binarization digunakan untuk menangani proses binarization yaitu proses merubah gambar grayscale menjadi gambar biner. Pada kelas Binarization hanya terdapat satu method yaitu method Proses, yang mana method inilah yang memuat proses binarization. Method ini memiliki satu parameter yaitu image yang bertipe BufferedImage, yang mana parameter tersebut sebagai input juga sebagai output.



Gambar 4. 5 Kelas Binarization

```

1  public void Proses(BufferedImage image){
2      int height = image.getHeight();
3      int width = image.getWidth();
4
5      for(int i=0;i<height;i++){
6          for(int j=0;j<width;j++){
7              if(image.getRGB(j, i) <= 0xff2e2d2d){
8                  image.setRGB(j, i, Color.BLACK.getRGB());
9              }else{
10                 image.setRGB(j, i, Color.WHITE.getRGB());
11             }
12         }
13     }
14 }
  
```

Sourcecode 4. 1 fungsi binarization

2. Kelas DataImage

Kelas DataImage digunakan untuk menyimpan data-data tentang gambar, diantaranya data density histogram dan data baseline. Kelas DataImage hanya berisi inisialisasi variable-variabel yang digunakan untuk menyimpan data yang betipe public dan static, sehingga variable tersebut dapat diakses di kelas manapun tanpa harus membuat instance dari kelas DataImage.

Datallmage
+ DensityHistogramVertical : ArrayList<int[]>
+ LeftBaseline : int[]
+ RightBaseline : int[]

Gambar 4. 6 Kelas DataImage

```

1 public class DataImage {
2     public static ArrayList<int []> DensityHistogramVertical;
3     public static int [] LeftBaseline;
4     public static int [] RightBaseline;
5 }

```

Sourcecode 4. 2 kelas DataImage

3. Kelas Thinning

Kelas Thinning digunakan untuk melakukan proses penipisan garis (thinning) pada gambar biner. Kelas Thinning ini memiliki dua method yaitu method getNeighbors dan Zhang_Suen.

Method getNeighbors merupakan method yang digunakan untuk mencari nilai piksel tetangga dari piksel tertentu. Method getNeighbors mengembalikan suatu nilai yaitu berupa array integer(int[]) yang berisi nilai piksel tetangga. Method Zhang_Suen digunakan untuk melakukan proses penipisan garis dengan menggunakan algoritma Zhang Suen, yang dalam prosesnya membutuhkan method getNeighbors untuk mendapatkan nilai piksel tetangga dari piksel yang sedang dievaluasi. Method Zhang_Suen memiliki satu parameter yaitu image yang bertipe BufferedImage, yang mana parameter tersebut sebagai masukan dan juga sebagai keluaran dari method tersebut.

Thinning
- CONSTANT_BLACK : int = 0xff000000
- CONSTANT_WHITE : int = 0xffffffff
+ Zhang_Suen(image : BufferedImage): void
- getNeighbors(imgval: int[], x: int, y: int, w: int, h: int): int[]

Gambar 4. 7 Kelas Thinning

4. Kelas Segmentation

Kelas Segmentation digunakan untuk melakukan proses segmentasi gambar kata menjadi gambar-gambar huruf yang menyusun kata tersebut. Adapun method-method yang terdapat pada kelas Segmentation dapat dilihat pada gambar 4.8, sedangkan penjelasan fungsi-fungsinya dapat dilihat pada tabel 4.4.

Segmentation
- PreSegmented : ArrayList<Integer[]>
+ Segmentation()
+ Generate(image : BufferedImage) : void
+ DrawSegmentation(image : BufferedImage, bound : int[]) : void
+ DrawSegmentation(image : BufferedImage, true_segmen : ArrayList<Integer>, bound : int[]) : void
- Sort(Data : ArrayList<Integer[]>) : void
+ getPreSegmented() : ArrayList<Integer[]>

Gambar 4. 8 Kelas Segmentation

Tabel 4. 4 Method-method yang terdapat pada kelas Segmentation

No	Nama Method	Keterangan
1	public Segmentation()	Method Segmentation merupa-kan constructor dari kelas Segmentation, yang mana nama method-nya sama dengan nama kelasnya.
2	Public void Generate(BufferedImage image)	Method ini berfungsi untuk menghasilkan titik-titik segmentasi, yang mana mempunyai masukan berupa gambar : BufferedImage.
3	private void DrawSegmentation (BufferedImage image, int[]bound)	Method ini digunakan untuk menggambarkan garis segmentasi pada gambar sesuai dengan titik-titik segmentasi yang telah dihasilkan pada method Generate
4	private void Sort(ArrayList <Integer[]> Data)	Method ini digunakan untuk mengurutkan data titik segmentasi sesuai dengan titik index terurut dari index terkecil sampai terbesar.
5	public ArrayList <Integer[]> getPreSegmented()	Method ini digunakan untuk mengambil nilai titik-titik segmentasi.

5. Kelas SkewDetection

Kelas SkewDetection digunakan untuk menentukan garis dasar (baseline) dari suatu gambar kata. Method yang dimiliki oleh kelas SkewDetection adalah Baseline, yang mana method tersebut berisikan proses untuk menentukan garis

dasar (baseline). Method Baseline menghasilkan suatu nilai berupa array integer (int[]) yang mana berisi nilai garis dasar (batas atas dan batas bawah). Method ini juga membutuhkan masukan berupa data density vertical histogram dari gambar.

SkewDetection
+ Baseline(Density: int[], max: int): int[] + AngleSkew(Left: int[], Right: int[], x_max: int): double

Gambar 4. 9 Kelas SkewDetection

```

1 public int [] Baseline(int [] Density, int max){
2     int [] y_atas_bawah = {0, 0};
3     for(int i=max;i>=0;i--){
4         if(Density[i] < (0.2*Density[max])){
5             y_atas_bawah[0] = i;
6             break;
7         }
8     }
9
10    for(int i=max;i<Density.length;i++){
11        if(Density[i] < (0.2*Density[max])){
12            y_atas_bawah[1] = i;
13            break;
14        }
15    }
16
17    return y_atas_bawah;
18 }
    
```

Sourcecode 4. 3 fungsi penentuan baseline

6. Kelas Histogram

Kelas Histogram digunakan untuk menghitung density histogram dari suatu gambar. Kelas ini memiliki dua method yaitu VerticalHistogram dan HorizontalHistogram, kedua method tersebut sama-sama berfungsi untuk menghitung density histogram, akan tetapi method VerticalHistogram menghitung density secara vertikal sedangkan method HorizontalHistogram menghitung density secara horisontal.

Histogram
+ VerticalHistogram(image : BufferedImage) : void + HorizontalHistogram(image : BufferedImage) : void

Gambar 4. 10 Kelas Histogram

4.2.1.3. Package neuralnetwork

Package neuralnetwork ini berfungsi sebagai tempat menyimpan kelas-kelas yang berhubungan dengan proses neural network, seperti proses pelatihan, penentuan bobot awal, pengujian, dan lain sebagainya. Adapun kelas-kelas utama yang berada pada package neuralnetwork dapat dilihat pada tabel 4.5.

Tabel 4. 5 Kelas-kelas yang terdapat pada package neuralnetwork

No	Nama Kelas	Keterangan
1	DataNeural	Kelas DataNeural merupakan kelas yang digunakan untuk menyimpan data-data <i>optical backpropagation neural network</i> , seperti data bobot, hasil aktivasi, data training, data target, dan lain sebagainya.
2	FileOperation	Kelas FileOperation merupakan kelas yang digunakan untuk melakukan proses-proses yang berhubungan dengan file, baik itu proses penyimpanan ke file atau proses pembacaan dari file.
3	LookUp	Kelas LookUp merupakan kelas yang digunakan untuk melakukan proses-proses untuk pencocokan hasil jaringan dengan target yang sesuai.
4	MasterLookUp	Kelas MasterLookUp merupakan kelas yang digunakan untuk menyimpan data target beserta kodenya, yang nantinya digunakan pada proses pencocokan dengan hasil jaringan.
5	Network	Kelas Network merupakan kelas yang digunakan untuk melakukan proses-proses yang berhubungan dengan neural network, seperti proses pelatihan dan proses pengujian. Kelas network ini <i>implements</i> kelas Runnable, sehingga kelas ini bias dijalankan dalam bentuk <i>multi thread</i> .
6	Recognition	Kelas Recognition merupakan kelas yang <i>implements</i> Runnable, yang digunakan untuk melakukan proses pengenalan tulisan tangan. Dikarenakan kelas ini <i>implements</i> Runnable, sehingga kelas ini dapat dijalankan secara <i>multi thread</i> .
7	Tools	Kelas Tools merupakan kelas yang digunakan untuk melakukan proses-proses perhitungan rumus, seperti perhitungan sigmoid, turunan sigmoid, dan <i>mean square error</i> (MSE).

1. Kelas DataNeural

Kelas DataNeural merupakan kelas yang berfungsi untuk menyimpan data-data yang nantinya akan digunakan dalam proses jaringan syaraf tiruan optical

backpropagation beserta mengelola proses-proses untuk akses ke data tersebut. Adapun data-data utama yang disimpan dalam kelas tersebut dapat dilihat pada gambar 4.11, sedangkan penjelasan mengenai fungsi-fungsinya dapat dilihat pada tabel 4.6.

DataNeural
<pre> + NET_MODE : int + INPUT_UNIT : int + HIDDEN_UNIT : int + OUTPUT_UNIT : int + MAX_EPOCH : int + DATA_DIR : String = "D:/Kuliah/SKRIPSI/PROGRESS_REPORT/Aplikasi/Data" + DATA_TRIAL_DIR : String = "" + LEARNING_RATE : double + MINIMUM_ERROR : double + DATA_TRAINING : int + DataTrainingInput : int[] + DataTrainingTarget : int[] + WeighInputHid : double[] + WeighHiddenOut : double[] + AktivasiHidden : double[] + AktivasiOutput : double[] + ErrorOutput : double[] + ErrorHidden : double[] + Init(Input : ArrayList<String[]>, Target : ArrayList<String[]>) : void + Init(input : int, hidden : int, output : int, jum_dat_train : int, max_epoch : int, learn_rate : double, min_error : double) : void + ConvertData(Input : ArrayList<String[]>, Target : ArrayList<String[]>) : void </pre>

Gambar 4. 11 Kelas DataNeural

Tabel 4. 6 Variabel-variabel yang terdapat pada kelas DataNeural

No	Nama Variabel	Keterangan
1	public static int [][] DataTrainingInput	Variabel ini digunakan untuk menyimpan data pelatihan, yaitu berupa <i>pattern</i> tiap karakter.
2	public static int [][] DataTrainingTarget	Variable ini digunakan untuk menyimpan pola data target.
3	public static double [][] WeighInputHid	Variabel ini digunakan untuk menyimpan data bobot dari <i>input layer</i> ke <i>hidden layer</i> .
4	public static double [][] WeighHiddenOut	Variabel ini digunakan untuk menyimpan data bobot dari <i>hidden layer</i> ke <i>output layer</i> .
5	public static double [] AktivasiHidden	Variabel ini digunakan untuk menyimpan data sinyal output dari <i>hidden layer</i> .
6	public static double [] AktivasiOutput	Variabel ini digunakan untuk menyimpan data sinyal output dari <i>output layer</i> .
7	public static double [] ErrorOutput	Variabel ini digunakan untuk menyimpan data sinyal error dari <i>output layer</i> .

8	public static double [] ErrorHidden	Variabel ini digunakan untuk menyimpan data sinyal error dari <i>hidden layer</i> .
---	--	---

2. Kelas FileOperation

Kelas FileOperation adalah kelas yang digunakan untuk melakukan proses akses file, baik itu pembacaan atau penulisan ke dalam file. Adapun method-method yang terdapat pada kelas tersebut dapat dilihat pada gambar 4.12, sedangkan penjelasan mengenai fungsi-fungsinya dapat dilihat pada tabel 4.7.

FileOperation
~ Input : ArrayList<String[]> = new ArrayList<>() ~ Target : ArrayList<String[]> = new ArrayList<>() - BaseReadOperation(filename : String) : BufferedReader - BaseWriteOperation(filename : String) : BufferedWriter + StandarSaveOperation(Content : StringBuffer, filename : String) : void + ReadDataTraining(mode : int, FileName : String) : void + ReadDataTraining(FileName : String) : void + getInpu() : ArrayList<String[]> + getTarget() : ArrayList<String[]> + SaveWeight(weight : double[], FileName : String) : void + SaveWeight(weightInpuHid : double[], weightHiddenOut : double[], filename : String) : void + ReadWeight(mode : int, FileName : String) : void + ReadWeight(FileName : String) : String + getListFiles(Dir : String) : HashMap<String,File[]> + ReadLookUp(filename : String) : void + SaveDataTraining(DataBinari : StringBuffer[], filename : String) : void

Gambar 4. 12 Kelas FileOperation

Tabel 4. 7 Penjelasan fungsi method – method yang terdapat pada kelas FileOperation

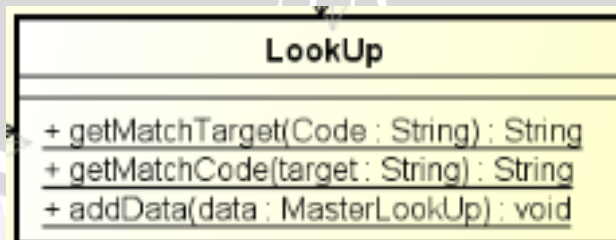
No	Nama Method	Keterangan
1	private BufferedReader BaseReadOperation(String filename)	Method ini digunakan untuk melakukan proses dasar dalam pembacaan file. Method ini juga membutuhkan masukan berupa String alamat dimana file yang akan dibaca.
2	private BufferedWriter BaseWriteOperation(String filename)	Method ini digunakan untuk melakukan proses dasar dalam penulisan pada file. Method ini juga membutuhkan masukan berupa String alamat dimana file yang akan ditulis.
3	public static void ReadDataTraining()	Method ini digunakan untuk melakukan proses pembacaan data training dari file.
4	public static void SaveWeight()	Method ini digunakan untuk melakukan

		proses penyimpanan data bobot hasil pelatihan pada file.
5	public static void ReadWeight()	Method ini digunakan untuk melakukan proses pembacaan data bobot, yang nantinya digunakan dalam pengujian (pengenalan tulisan tangan)
6	public HashMap<String, File[]> getListFiles()	Method ini digunakan untuk melakukan proses pembacaan direktori dimana file gambar data pengujian berada.
7	public void ReadLookUp()	Method ini digunakan untuk melakukan proses pembacaan data pemetaan dari kode data target ke karakter aslinya.
8	public void SaveDataTraining()	Method ini digunakan untuk melakukan proses penyimpanan data training yang di-generate oleh aplikasi ke dalam suatu file.

3. Kelas LookUp

Kelas LookUp adalah kelas yang digunakan untuk melakukan proses-proses pemetaan dari kode yang dihasilkan ke karakter yang sebenarnya, atau proses sebaliknya yaitu pemetaan dari karakter yang sebenarnya ke kode yang merepresentasikannya.

Adapun method-method yang terdapat pada kelas LookUp adalah getMatchTarget dan getMatchCode. getMatchTarget digunakan untuk melakukan proses pemetaan dari kode ke karakter aslinya, sedangkan getMatchCode digunakan untuk melakukan proses pemetaan dari karakter aslinya ke kode yang merepresentasikannya.



Gambar 4. 13 Kelas LookUp

```

1 public static String getMatchTarget(String Code) {
2     String Result = "_";
3     for (int i = 0; i < DataLookup.size(); i++) {
  
```

```

4         if(DataLookup.get(i).getCode().equals(Code)){
5             Result = DataLookup.get(i).getTarget();
6             break;
7         }
8     }
9
10    return Result;
11 }

```

Sourcecode 4. 4 memetakan dari kode ke karakter aslinya

```

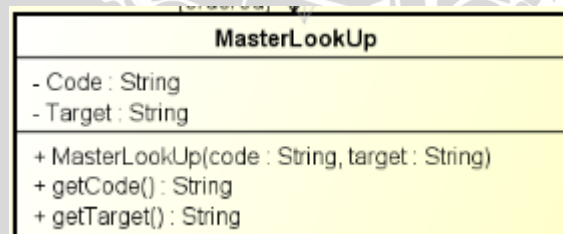
1 public static String getMatchCode(String target){
2     String Result = "unknown";
3     for (int I = 0; I < DataLookup.size(); i++) {
4         if(DataLookup.get(i).getTarget().equals(target)){
5             Result = DataLookup.get(i).getCode();
6             break;
7         }
8     }
9
10    return Result;
11 }

```

Sourcecode 4. 5 memetakan karakter asli ke kode

4. Kelas MasterLookup

Kelas MasterLookup adalah kelas yang digunakan untuk menyimpan data kode dan karakter aslinya, sehingga kelas ini dapat digunakan dalam proses pemetaan yang dilakukan pada kelas Lookup.



Gambar 4. 14 Kelas MasterLookup

```

1 public class MasterLookup {
2     private String Code;
3     private String Target;
4
5     public MasterLookup(String code, String target){
6         this.Code = code;
7         this.Target = target;
8     }
9
10    public String getCode() {
11        return Code;
12    }
13
14    public String getTarget() {
15        return Target;
16    }
17 }

```



```

16     }
17 }

```

Sourcecode 4. 6 kelas MasterLookUp

5. Kelas Network

Kelas Network adalah kelas yang digunakan untuk melakukan proses-proses yang ada di dalam neural network optical backpropagation, seperti proses pelatihan dan proses pengujian. Kelas Network ini implements terhadap kelas Runnable, yang mana kelas Runnable ini adalah kelas yang telah disediakan Java yang memungkinkan kelas tersebut dapat dijalankan secara multi thread. Adapun method-method yang terdapat pada kelas Network dapat dilihat pada gambar 4.15, sedangkan fungsi-fungsinya dapat dilihat pada tabel 4.8.

```

Network
~ status_proses : int = 0

+ Network(input_unit : int, hidden_unit : int, output_unit : int, max_epoch : int, data_training : int, learning_rate : double, minimum_error : double)
+ Network(Input : ArrayList<String[]>, Target : ArrayList<String[]>)
+ Network()
+ StopProses() : void
+ StartTraining() : void
+ SaveWeightTraining(filename : String) : void
- InitiateWeight() : void
+ Feedforward(datake : int) : void
+ Feedforward(DataUji : int[]) : double[]
- OptBackpropagasi(datake : int) : void
- PureBackpropagasi(datake : int) : void
- UpdateWeight(datake : int) : void
+ Recognition(DataUji : int[]) : double[]
+ run() : void

```

Gambar 4. 15 Kelas Network

Tabel 4. 8 Method-method yang terdapat pada kelas Network

No	Nama Method	Keterangan
1	public Network()	Method ini merupakan constructor dari kelas Network
2	private void InitiateWeight()	Method ini digunakan untuk melakukan proses inisialisasi bobot awal dengan menggunakan algoritma <i>Nguyen-Widrow</i> .
3	public void <i>Feedforward</i> ()	Method ini digunakan untuk melakukan proses pada tahapan <i>feedforward</i> .
4	private void OptBackpropagasi()	Method ini digunakan untuk melakukan proses pada tahapan propagasi balik.
5	private void UpdateWeight()	Method ini digunakan untuk melakukan proses pembaharuan bobot .
6	public void StartTraining()	Method ini digunakan untuk melakukan proses training, yang mana method ini

	menggabungkan fungsi dari method-method yang lain, seperti InitiateWeight, Feedforward, OptBackpropa-gasi, dan UpdateWeight.
--	--

```

1 public void StartTraining() {
2     InitiateWeight();
3     double[] DataMSE = new
4     double[DataNeural.DataTrainingInput.length];
5     int BarValue = 0;
6     for (int epc = 0; epc < DataNeural.MAX_EPOCH; epc++) {
7         for (int i = 0; i <
8         DataNeural.DataTrainingInput.length; i++) {
9             Feedforward(i);
10
11             DataMSE[i] =
12             Tools.MSE(DataNeural.DataTrainingTarget, DataNeural.ActivasiOutput,
13             i);
14             if(DataNeural.NET_MODE == 0){
15                 OptBackpropagasi(i);
16             }else if(DataNeural.NET_MODE == 1){
17                 PureBackpropagasi(i);
18             }
19
20             UpdateWeight(i);
21         }
22
23         int countError = 0;
24         int SizeInput = DataNeural.DataTrainingInput.length;
25         double SumError = 0.0;
26         for (int i = 0; i < SizeInput; i++) {
27             if(DataMSE[i] < DataNeural.MINIMUM_ERROR){
28                 countError++;
29             }
30             SumError += DataMSE[i];
31         }
32
33         if(countError == SizeInput){
34             break;
35         }
36     }
37 }

```

Sourcecode 4. 7 fungsi untuk pelatihan

6. Kelas Recognition

Kelas Recognition adalah kelas yang digunakan untuk melakukan proses pengujian (pengenalan tulisan tangan) setelah dilakukan pelatihan pada jaringan. Kelas Recognition ini implements terhadap kelas Runnable seperti pada kelas Network, sehingga proses dapat dilakukan secara multi thread.

Recognition
- DATA_UJI : int[] - Result : String - Index : int
+ Recognition(dataUji : int[]) + setIndex(index : int) : void + Proses() : void + run() : void

Gambar 4. 16 Kelas Recognition

7. Kelas Tools

Kelas Tools adalah kelas yang digunakan untuk melakukan proses-proses perhitungan dari sebuah formula matematika. Adapun perhitungan-perhitungan yang ditangani adalah perhitungan sigmoid, turunan sigmoid, dan mean square error (MSE).

Tools
+ Sigmoid(val : double) : double + SigmoidDerivative(val : double) : double + MSE(actual_output : int[], output : double[], datake : int) : double

Gambar 4. 17 Kelas Tools

```

1 public static double Sigmoid(double val){
2     return (1.0 / (1.0 + Math.exp(-val)));
3 }

```

Sourcecode 4. 8 fungsi Sigmoid

```

1 public static double SigmoidDerivative(double val){
2     return (val * (1.0 - val));
3 }

```

Sourcecode 4. 9 fungsi turunan sigmoid

```

1 public static double MSE(int [][] actual_output, double []
2 output, int datake){
3     double hasil = 0.0;
4     for(int i=0;i<output.length;i++){
5         hasil += Math.pow((actual_output[datake][i]-
6 output[i]), 2);
7     }
8     return 0.5*hasil;
9 }

```

Sourcecode 4. 10 fungsi Mean Square Error (MSE)

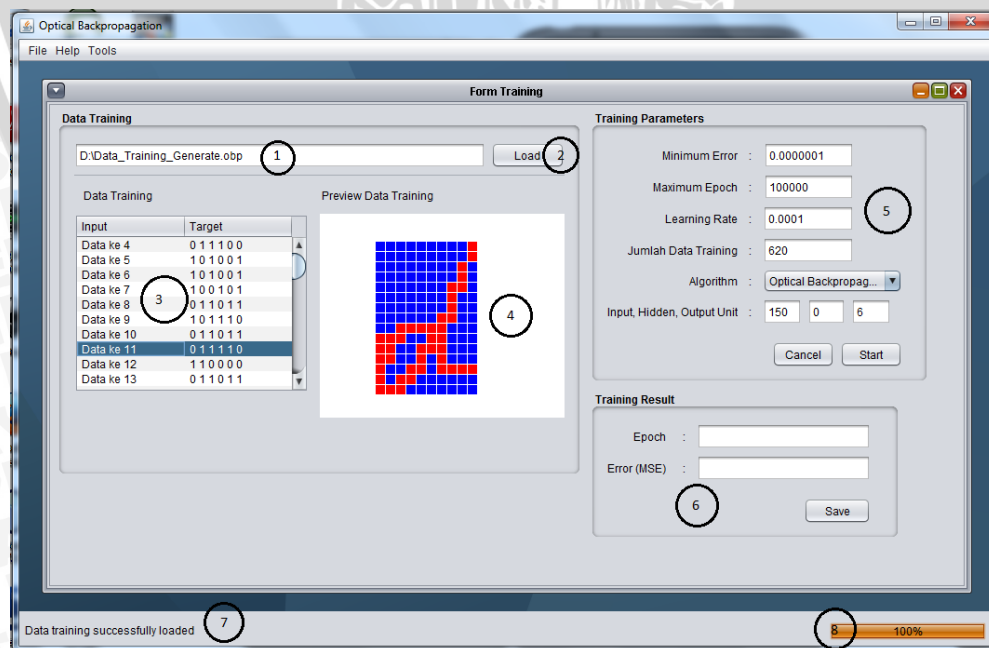
4.2.2. Implementasi Antarmuka

Implementasi antarmuka pada aplikasi pengenalan tulisan tangan, terdapat dua form utama, yaitu form yang digunakan untuk pelatihan (Form Training) dan form yang digunakan untuk pengenalan (Form Recognition).

1. Form Training

Form Training adalah form yang digunakan untuk menangani proses-proses yang berhubungan dengan pelatihan jaringan. Adapun proses-proses yang terdapat pada form training adalah proses memasukkan data latih (ditunjukkan nomor 1 dan 2 pada gambar 4.18), melihat representasi data latih dalam bentuk gambar (ditunjukkan nomor 3 dan 4 pada gambar 4. 18), setting parameter-parameter untuk proses pelatihan jaringan (ditunjukkan nomor 5 pada gambar 4.18), dan proses untuk menyimpan bobot hasil pelatihan (ditunjukkan nomor 6 pada gambar 4.18).

Pada form tersebut juga terdapat status proses yang sedang berjalan, yaitu berupa text dan progress bar (ditunjukkan nomor 7 dan 8 pada gambar 4.18). Status tersebut berfungsi untuk menginformasikan kepada user apakah proses yang dilakukan sudah selesai atau masih berjalan.

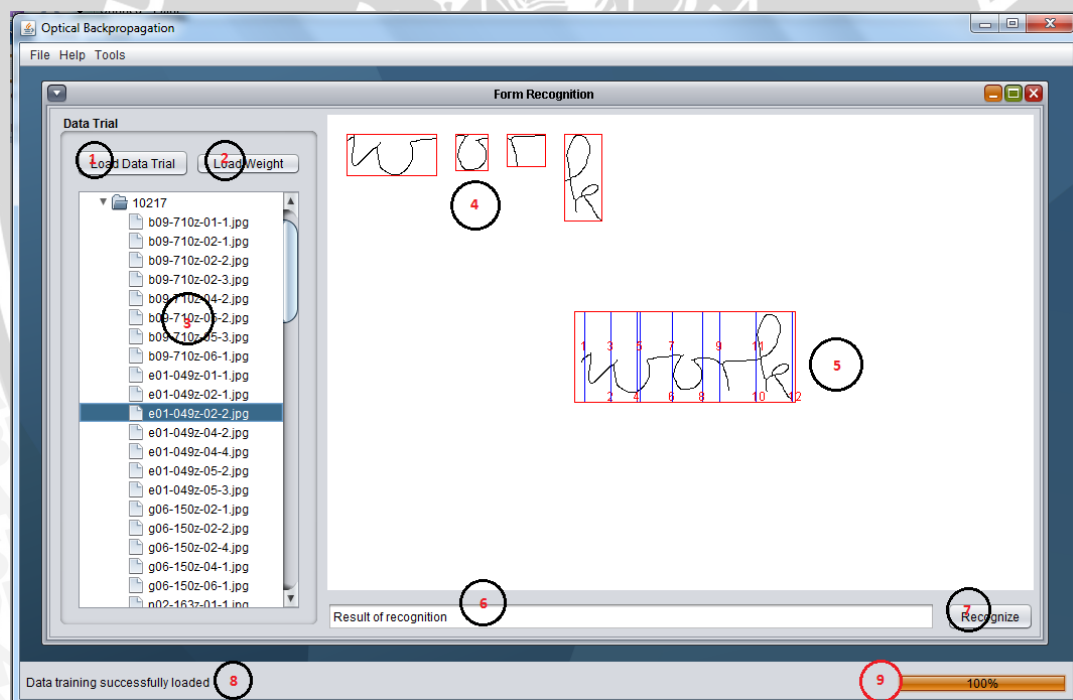


Gambar 4. 18 Form Training

2. Form Recognition

Form Recognition adalah form yang digunakan untuk menangani proses-proses yang berhubungan dengan pengenalan tulisan tangan (pengujian jaringan). Adapun proses-proses yang terdapat pada form recognition adalah proses memasukkan data uji (ditunjukkan nomor 1 pada gambar 4.19), memasukkan bobot hasil pelatihan (ditunjukkan nomor 2 pada gambar 4.19), menampilkan data uji (ditunjukkan nomor 3 dan 5 pada gambar 4.19), menampilkan hasil segmentasi data uji (ditunjukkan nomor 4 pada gambar 4.19), menampilkan hasil pengenalan data uji (ditunjukkan nomor 6 dan 7 pada gambar 4.19).

Seperti pada form training, pada form recognition juga terdapat status proses yang sedang berjalan, baik berupa text maupun berupa progress bar (ditunjukkan nomor 8 dan 9 pada gambar 4.19).



Gambar 4. 19 Form Recognition

4.3. Implementasi Uji Coba

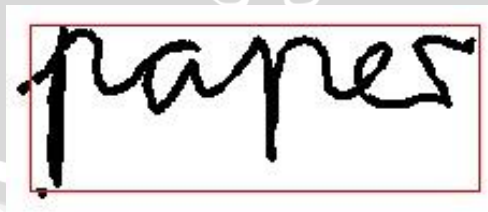
Pada subbab ini akan dilakukan pembahasan mengenai hasil pengujian yang telah dilakukan pada sistem dan hasil analisis terhadap hasil pengujian tersebut.

4.3.1. Hasil Uji Coba Pelatihan

Pengujian dilakukan terhadap data latih dengan variasi jumlah hidden unit pada hidden layer dan variasi besar learning rate. Adapun variasi jumlah hidden unit yang diujikan adalah 30, 50, 70, 90, dan 100 hidden unit. Sedangkan untuk variasi besar learning rate adalah 0.1, 0.3, 0.5, 0.7, dan 0.9. Adapun jumlah input unit yaitu 150, jumlah output unit adalah 6, dan dengan maksimum epoch adalah 10000.

Data yang digunakan dalam proses pelatihan merupakan data yang diperoleh dari IAM Handwriting Database (<http://www.iam.unibe.ch/fki/databases/iam-handwriting-database>) yang sudah dilakukan beberapa *pre-processing*. Adapun *pre-processing* yang dilakukan pada data tersebut adalah pemotongan per-kata dari kalimat yang ada di gambar. Kemudian dari hasil pemotongan tersebut, dimasukkan ke dalam sistem untuk dilakukan *binarization*, *thinning* dan pemotongan per-huruf. Dari hasil proses tersebut setiap piksel dari setiap huruf akan direpresentasikan dalam bentuk biner (1 untuk foreground dan 0 untuk background).

Pada *pre-processing binarization*, telah diujikan beberapa nilai *threshold*, yaitu #222222, #2e2d2d, #777777, #0f0f0f, #dcdede, dan diambil kesimpulan bahwa *threshold* yang bisa digunakan yang sesuai dengan data set yang digunakan pada penelitian ini yaitu dalam rentang #222222 dan #777777, jika diberi nilai warna yang lebih gelap (#0f0f0f), maka akan terdapat garis-garis yang terpisah, begitupun juga jika diberi nilai warna yang lebih terang (#dcdede), maka akan terdapat garis-garis yang terlalu tebal.



Gambar 4. 20 Threshold #222222



Gambar 4. 21 Threshold #777777



Gambar 4. 22 Threshold #0f0f0f

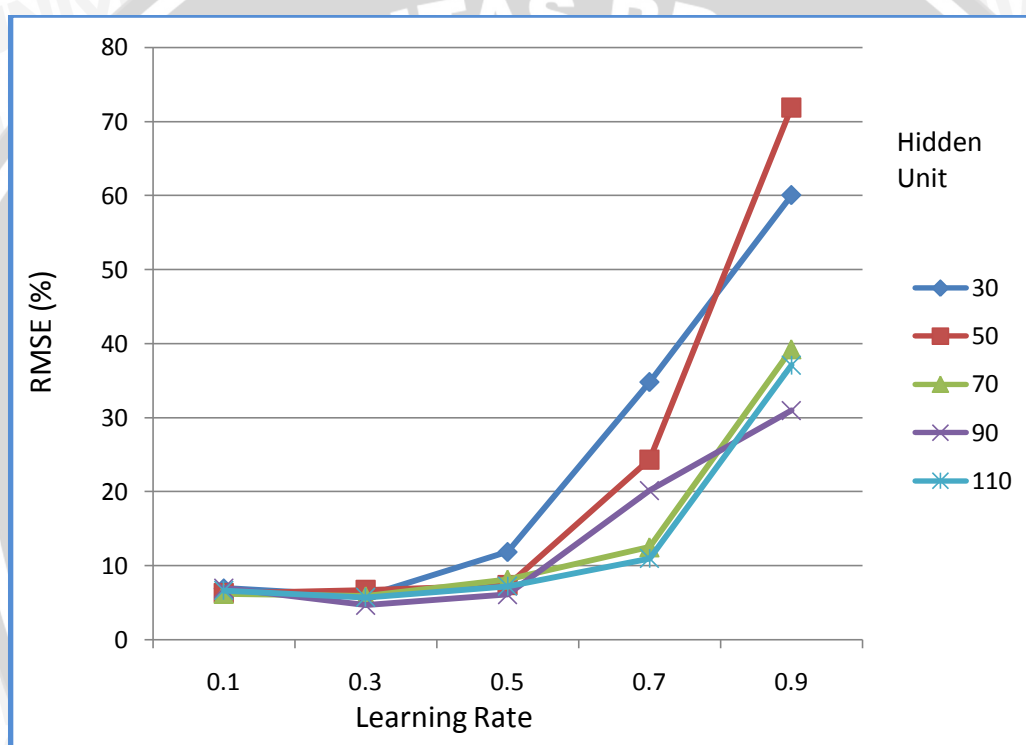


Gambar 4. 23 Threshold #dedcdc

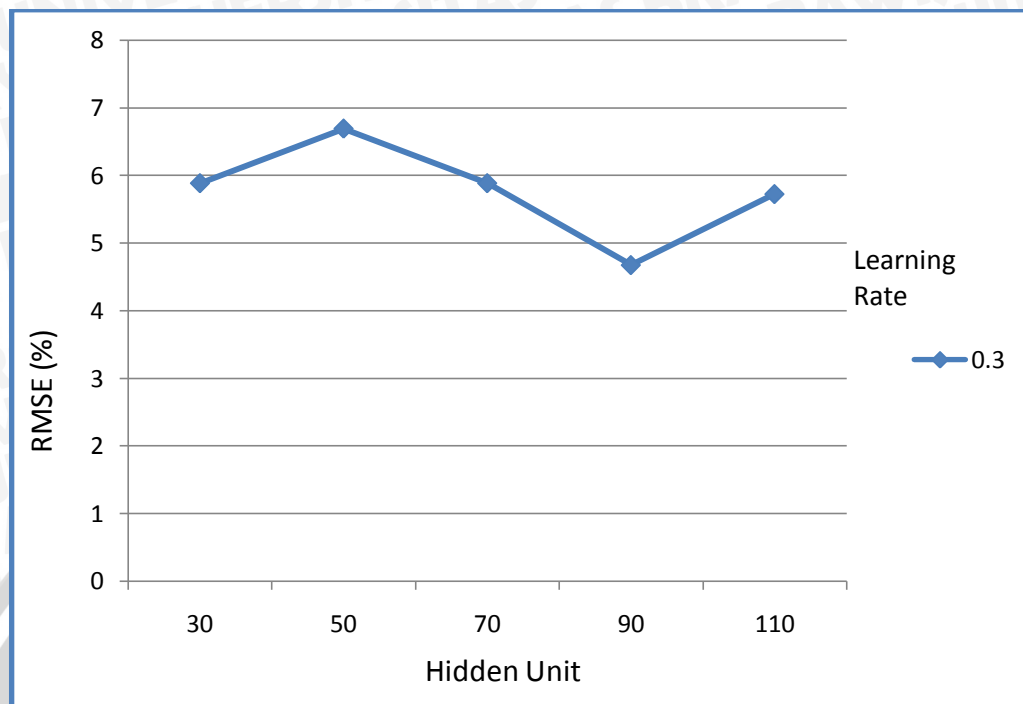
Nilai error (rata-rata MSE) pada training Optical Backpropagation Neural Network (OBPNN) ditunjukkan pada tabel 4.9. Dari tabel tersebut dapat dilihat bahwa untuk jumlah hidden unit yang sama, dengan semakin bertambahnya nilai learning rate akan menghasilkan error yang semakin besar, begitupun juga sebaliknya, semakin kecil learning rate semakin kecil error yang dihasilkan. Grafik pengaruh learning rate terhadap error yang dihasilkan dapat dilihat pada gambar 4.24. Sedangkan untuk nilai learning rate yang sama, perubahan jumlah hidden unit menghasilkan error yang tidak terpaut jauh satu dengan yang lain. Grafik pengaruh jumlah hidden unit terhadap error yang dihasilkan dapat dilihat pada gambar 4.25.

Tabel 4. 9 nilai error (RMSE) untuk setiap perubahan hidden unit dan learning rate pada pelatihan.

LEARNING RATE	JUMLAH HIDDEN UNIT				
	30	50	70	90	110
0.1	6.94	6.29	6.21	7.02	6.69
0.3	5.89	6.69	5.89	4.68	5.73
0.5	11.85	7.34	8.15	6.13	7.26
0.7	34.81	24.35	12.50	20.16	10.97
0.9	60.06	71.92	39.26	30.96	37.09



Gambar 4. 24 Grafik pengaruh learning rate terhadap error (RMSE) yang dihasilkan

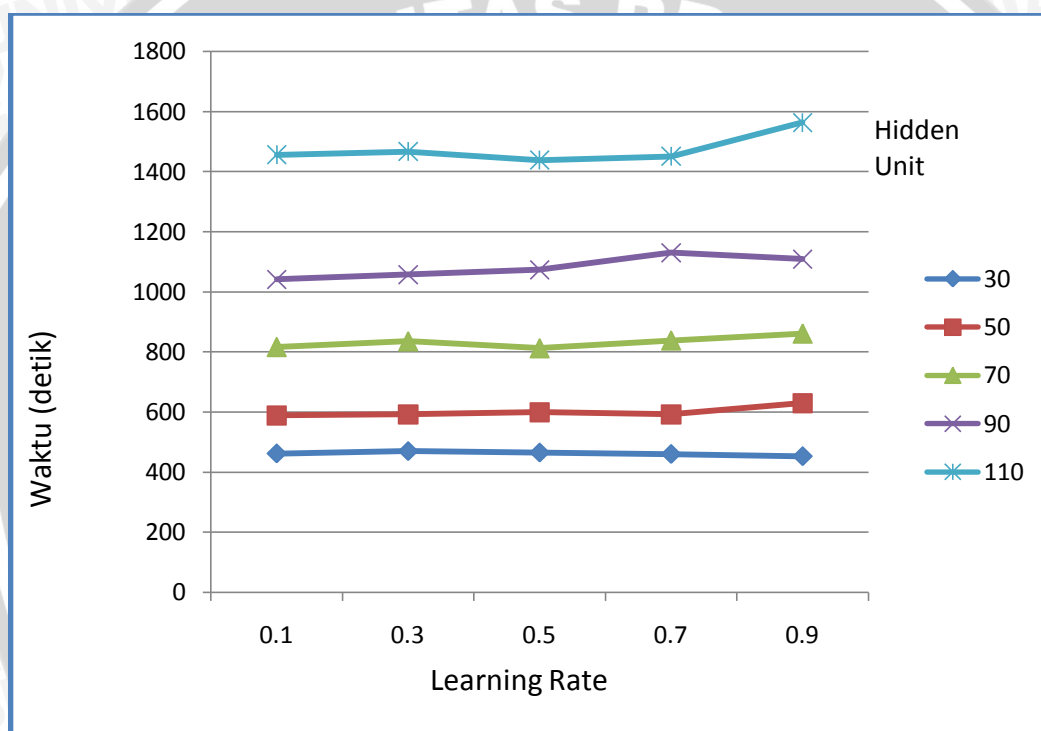


Gambar 4. 25 Grafik pengaruh hidden unit terhadap perubahan error (RMSE)

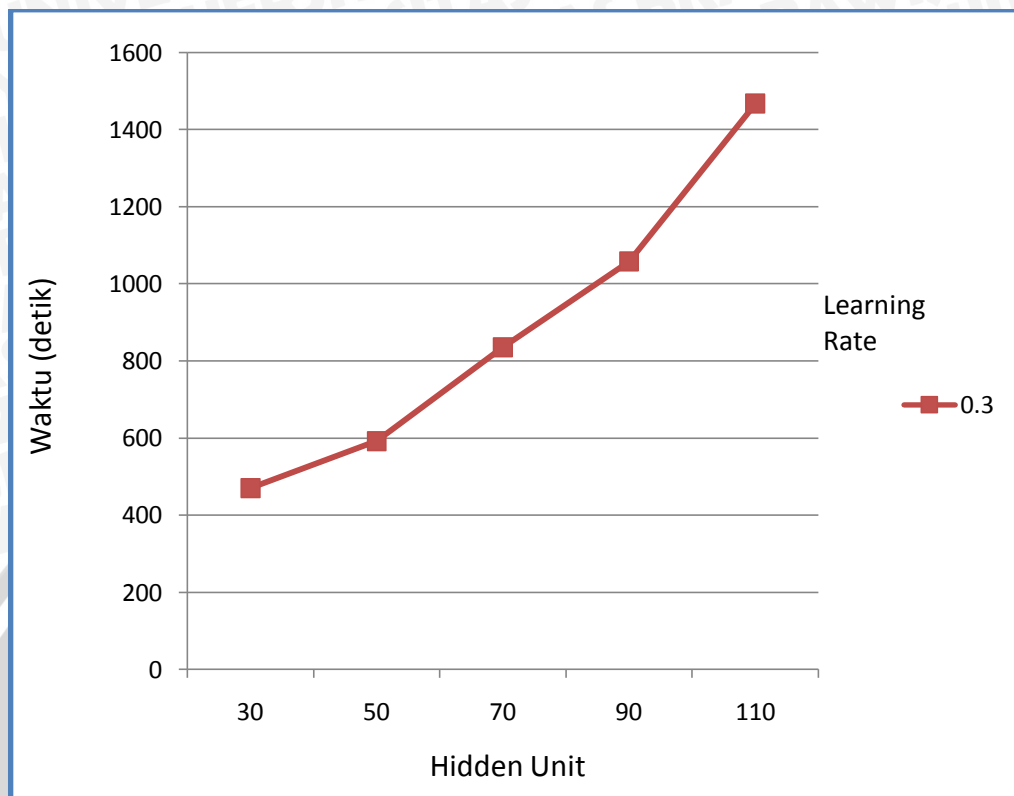
Lama waktu yang dibutuhkan pada training Optical Backpropagation Neural Network (OBPNN) ditunjukkan pada tabel 4.10 Dari tabel tersebut dapat dilihat bahwa untuk nilai learning rate yang sama, dengan semakin bertambahnya jumlah hidden unit akan membutuhkan waktu yang lebih banyak, begitupun juga sebaliknya, semakin berkurang jumlah hidden unit, maka waktu yang dibutuhkan juga semakin sedikit. Grafik pengaruh jumlah hidden unit terhadap waktu yang diperlukan untuk pelatihan dapat dilihat pada gambar 4.27 Sedangkan pada jumlah hidden unit yang sama, perubahan learning rate menghasilkan jumlah waktu tidak terpaut jauh satu dengan yang lainnya. Grafik pengaruh nilai learning rate terhadap waktu yang diperlukan untuk pelatihan dapat dilihat pada gambar 4.26.

Tabel 4. 10 waktu yang dibutuhkan untuk pelatihan untuk setiap hidden unit dan learning rate.

LEARNING RATE	JUMLAH HIDDEN UNIT				
	30	50	70	90	110
0.1	462.47	589.06	817.02	1041.91	1456.44
0.3	470.31	591.96	835.76	1058.10	1467.43
0.5	465.54	599.11	813.03	1073.79	1438.76
0.7	460.58	591.92	838.61	1130.41	1451.66
0.9	453.10	629.91	861.26	1109.65	1564.19



Gambar 4. 26 Grafik pengaruh learning rate terhadap waktu pelatihan



Gambar 4. 27 Grafik pengaruh hidden unit terhadap waktu yang dibutuhkan untuk pelatihan.

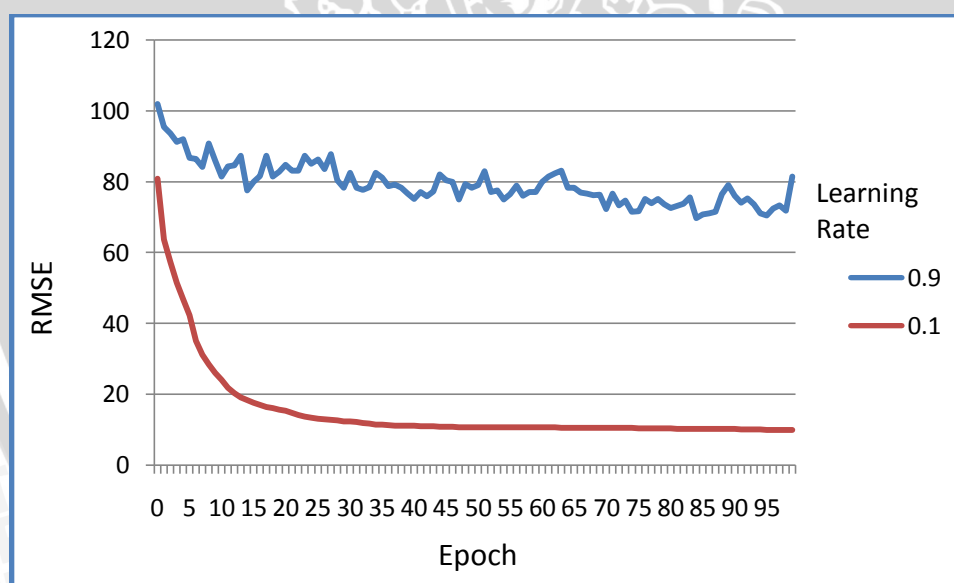
4.3.2. Hasil Uji Coba Pengenalan

Pelatihan yang telah dilakukan akan menghasilkan suatu bobot yang digunakan pada tahapan pengenalan. Proses pelatihan pada algoritma Optical Backpropagation Neural Network (OBPNN) melibatkan tiga tahapan proses yaitu *feedforward*, backpropagation, dan pengubahan bobot. Sedangkan pada proses pengenalan hanya melibatkan satu tahapan proses yaitu *feedforward*, dengan menggunakan bobot yang diperoleh dari hasil pelatihan. Data set yang digunakan sebagai pengujian adalah sejumlah 239 gambar tulisan tangan (setiap gambar berisi satu kata). Adapun data hasil pengenalan terhadap sejumlah data set tersebut dapat dilihat pada lampiran 2 dan data hasil segmentasi dapat dilihat pada lampiran 1, yang mana rata-rata hasil segmentasi karakter sebesar 61.52% dan rata-rata hasil pengenalan sebesar 52.50%.

4.3.3. Analisa Hasil Pelatihan

1. Analisa Pengaruh Learning Rate Terhadap Training Error

Learning rate merupakan suatu konstanta yang digunakan pada training optical backpropagation neural network (OBPNN), yang berpengaruh pada perhitungan perubahan bobot. Semakin besar nilai learning rate yang digunakan, maka semakin besar pula perubahan bobot disetiap siklusnya (epoch). Perubahan bobot ini bisa bernilai positif maupun negatif, sehingga nilai bobot bisa naik maupun turun. Backpropagation neural network pada dasarnya menerapkan perubahan yang kecil pada bobot untuk setiap siklusnya. Jika perubahan yang dibuat pada bobot terlalu besar, maka akan terjadi nilai error yang naik turun, sehingga juga sulit untuk mencapai konvergen. Perbandingan grafik perubahan error antara pelatihan menggunakan learning rate 0.9 dengan pelatihan menggunakan learning rate 0.1 dapat dilihat pada gambar 4.28.



Gambar 4. 28 Grafik perubahan error (RMSE)

Pada grafik dapat kita lihat, untuk learning rate 0.9 membentuk grafik yang naik turun. Dengan nilai learning rate 0.9, menyebabkan perubahan yang terlalu besar pada nilai bobot (Δw), dan dikarenakan nilai perubahan bobot (Δw) yang dapat bernilai positif maupun negatif sehingga menyebabkan grafik perubahan error (RMSE) membentuk pola naik turun. Sedangkan pada learning

rate 0.1, pada grafik terlihat semakin bertambahnya epoch, error yang dihasilkan berkurang secara signifikan (mendekati konvergen), tanpa adanya pola naik turun. Dengan penggunaan learning rate 0.1, tidak terjadi perubahan bobot yang drastis seperti yang terjadi pada learning rate 0.9.

Seperti yang terlihat pada gambar 4.24, bahwa semakin kecil nilai learning rate yang digunakan maka error yang dihasilkan akan semakin kecil pula, akan tetapi juga dibutuhkan lebih banyak epoch.

2. Analisa Pengaruh Learning Rate Terhadap Kecepatan Training

Learning rate secara tidak langsung mempengaruhi kecepatan training. Disebutkan secara tidak langsung dikarenakan learning rate lebih erat kaitannya dengan laju perubahan bobot disetiap siklusnya (epoch). Menentukan learning rate yang sesuai merupakan hal yang penting dalam optical backpropagation neural network (OBPP). Karena learning rate tersebut sangat berpengaruh dalam error maupun kecepatan training yang dihasilkan. Learning rate yang terlalu besar akan menyebabkan perubahan bobot yang terlalu besar pula, sehingga error akan naik turun seperti pada gambar 4.28. Untuk itu diperlukan waktu yang lebih lama untuk mencapai konvergen, atau bahkan mungkin tidak bisa mencapai konvergen (berulang pada error yang sama disetiap siklusnya). Akan tetapi jika learning rate terlalu kecil, maka perubahan bobot juga akan terlalu kecil, sehingga akan memperlambat perubahan error untuk mencapai konvergen, sehingga juga membutuhkan waktu yang lebih lama untuk mencapai konvergen.

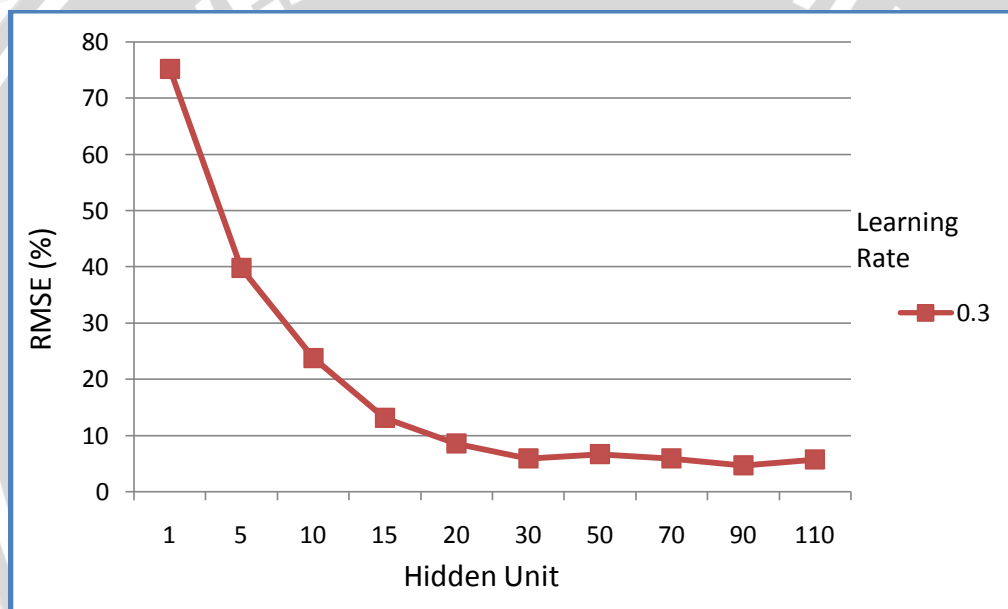
Dapat dilihat pada gambar 4.26, bahwa untuk setiap learning rate, waktu yang diperlukan tidak terpaut jauh nilainya. Ini dikarenakan pelatihan tidak mencapai batas minimum error sehingga epoch yang ditempuh untuk setiap learning rate adalah sama, yaitu maksimum epoch.

3. Analisa Pengaruh Jumlah Hidden Unit Terhadap Training Error

Hidden unit merupakan neuron penghubung antara input unit dengan output unit. Penentuan berapa banyak hidden unit yang harus digunakan sama pentingnya dengan penentuan besar learning rate. Jika jumlah hidden unit yang digunakan terlalu kecil maka akan terjadi underfitting, yaitu arsitektur jaringan yang terlalu

sederhana, sehingga masih banyak informasi dari data set yang tidak terproses oleh jaringan. Oleh karena itu hidden unit yang terlalu sedikit akan menghasilkan error yang tinggi pula, seperti terlihat pada gambar 4.29

Jika jumlah hidden unit yang digunakan terlalu banyak, maka akan terjadi overfitting, yaitu ketika kapasitas pengolahan proses (neuron pada jaringan) terlalu banyak dari informasi yang diolah, maka akan ada sejumlah hidden unit yang tidak ikut terlatih untuk mengenali pola dataset. Oleh Karena itu hidden unit yang tidak ikut terlatih akan menjadi noise, yang juga akan menyebabkan error yang dihasilkan juga lebih tinggi.



Gambar 4. 29 Grafik pengaruh hidden unit terhadap training error (hidden unit mulai dari 1 - 110)

4. Analisa Pengaruh Jumlah Hidden Unit Terhadap Kecepatan Training

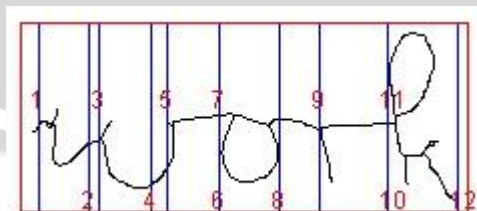
Seperti yang terlihat pada gambar 4.27, bahwa semakin banyak hidden unit yang digunakan maka semakin banyak pula waktu yang dibutuhkan untuk pelatihan. Hidden unit merupakan neuron penghubung antara input unit dan output unit. Setiap penghubung mempunyai nilai bobot masing-masing. Oleh karena itu, semakin banyak hidden unit yang digunakan, maka bobot yang harus

diproses juga semakin banyak, yang ini juga menyebabkan meningkatnya waktu untuk proses pelatihan tersebut.

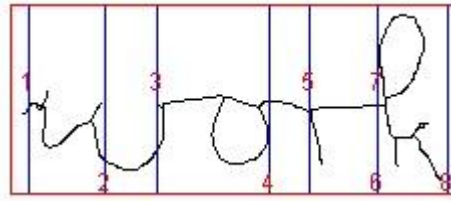
4.3.4. Analisa Hasil Pengenalan

Pelatihan yang telah dilakukan akan menghasilkan suatu bobot yang digunakan pada tahapan pengenalan. Proses pelatihan pada algoritma Optical Backpropagation Neural Network (OBPNN) melibatkan tiga tahapan proses yaitu *feedforward*, backpropagation, dan pengubahan bobot. Sedangkan pada proses pengenalan hanya melibatkan satu tahapan proses yaitu *feedforward*, dengan menggunakan bobot yang diperoleh dari hasil pelatihan. Dalam pengenalan tulisan tangan bersambung, terdapat dua hal penting yaitu segmentasi per karakter dan pengenalan karakter itu sendiri.

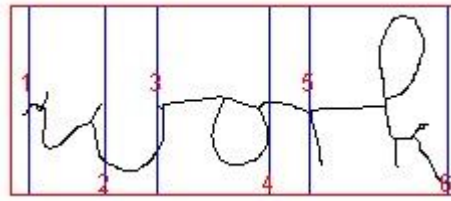
Salah satu parameter yang digunakan untuk proses penentuan segmentasi adalah estimasi lebar huruf. Ada beberapa nilai yang telah diujikan untuk menentukan estimasi lebar huruf yang sesuai untuk data set yang digunakan, diantaranya 63%, 73%, 83%, dan 93% dari tinggi huruf. Didapatkan kesimpulan bahwa estimasi lebar huruf sebesar 83% dari tinggi huruf adalah yang paling sesuai untuk bisa diterapkan pada penelitian ini. Ketika menggunakan 63% dari tinggi huruf, maka akan terdapat huruf-huruf yang tersegmen lebih dari satu segmen, dikarenakan estimasi lebar huruf yang terlalu kecil. Begitupun juga ketika menggunakan 93% dari tinggi huruf, maka akan terdapat dua huruf yang tersegmen menjadi satu segmen, yang dikarenakan estimasi lebar huruf yang terlalu besar.



Gambar 4. 30 Lebar 63% dari tinggi huruf



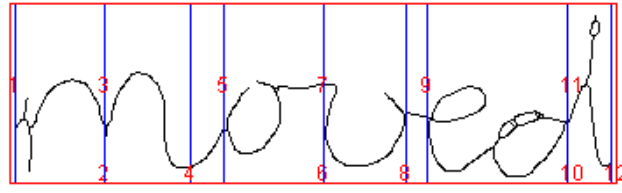
Gambar 4. 31 Lebar 83% dari tinggi huruf



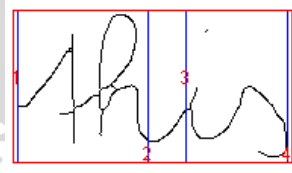
Gambar 4. 32 Lebar 93% dari tinggi huruf

Pada proses pengenalan, terdapat batas error yang digunakan *optical backpropagation* untuk menentukan apakah segmen tersebut termasuk karakter atau bukan. Ketika error yang dihasilkan kurang dari batas error maka bisa dianggap sebagai karakter, dan begitu sebaliknya. Adapun beberapa nilai yang telah diujikan untuk memperoleh batas error yang sesuai yaitu 0.4, 0.3, 0.2, 0.1, dan 0.01. Didapatkan kesimpulan bahwa batas error 0.01 adalah yang paling sesuai untuk bisa diterapkan pada penelitian ini. Jika menggunakan batas error 0.4, toleransi error yang digunakan masih terlalu tinggi, sehingga terdapat segmen-segmen yang seharusnya bukan karakter, akan tetapi tetap dikenali sebagai karakter.

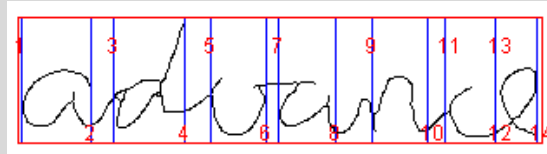
Segmentasi karakter memegang peranan penting dalam pengenalan tulisan tangan bersambung, segmentasi yang benar akan memudahkan dalam proses pengenalan. Begitu juga sebaliknya, segmentasi yang salah maka akan mempersulit dalam proses pengenalan. Adapun beberapa contoh hasil segmentasi yang benar dan salah dapat dilihat pada gambar 4.33, 4.34, dan 4.35. Terdapat satu karakter yang memiliki banyak titik segmen, dan juga terdapat beberapa karakter yang berada dalam satu segmen..



Gambar 4. 33 Karakter “m” yang memiliki lebih dari satu titik segmen



Gambar 4. 34 Karakter “t” dan “h” yang berada dalam satu segmen



Gambar 4. 35 Hasil segmentasi yang benar

Segmentasi yang dilakukan berdasarkan nilai histogram vertical density dan estimasi lebar huruf yang telah ditentukan diawal, yang kemudian dievaluasi dengan neural network. Dikarenakan estimasi lebar huruf sudah ditentukan diawal, terdapat huruf-huruf yang sulit untuk disegmentasi yang memang lebar hurufnya lebih kecil dari rata-rata lebar huruf yang lainnya, seperti huruf “i” dan “l”.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah :

1. Dari hasil percobaan, diperoleh kombinasi *learning rate* dan jumlah *hidden unit* yang terbaik untuk implementasi Optical Backpropagation Neural Network pada kasus pengenalan tulisan tangan huruf bersambung yaitu 0.3 dan 90. Kombinasi *learning rate* dan jumlah *hidden unit* tersebut menghasilkan *error training* sebesar 4.68%.
2. Semakin kecil *learning rate* yang digunakan maka *error* yang dihasilkan dari pelatihan Optical Backpropagation Neural Network pada kasus pengenalan tulisan tangan huruf bersambung juga akan semakin kecil, akan tetapi juga akan membutuhkan siklus (epoch) yang lebih banyak. Sedangkan untuk *hidden unit* dalam rentang nilai 30 – 110 tidak begitu berpengaruh terhadap *error* yang dihasilkan, akan tetapi jika digunakan nilai yang terlalu kecil maka akan terjadi *underfitting*, yang juga menyebabkan *error* yang dihasilkan semakin besar.
3. Semakin besar nilai *hidden unit* yang digunakan, maka waktu yang dibutuhkan untuk pelatihan juga akan semakin lama. Sedangkan untuk *learning rate* tidak begitu berpengaruh terhadap waktu yang dibutuhkan untuk pelatihan.
4. Pengenalan tulisan tangan huruf bersambung sangat bergantung dengan hasil segmentasi huruf yang dihasilkan, jika segmentasi yang dihasilkan tidak baik, maka akan mempersulit dalam proses pengenalannya. Dari percobaan yang dilakukan pada sejumlah data set, dihasilkan rata-rata hasil segmentasi sebesar 61.52% dan rata-rata hasil pengenalan sebesar 52.50%.

5.2. Saran

Beberapa saran untuk pengembangan lebih lanjut yang dapat diberikan oleh penulis adalah:

1. Dilakukan segmentasi dengan tidak hanya berdasarkan *histogram density* dan lebar huruf, tetapi bisa dilengkapi dengan fitur-fitur lain yang bisa diekstrak, seperti *Contour Extraction* dan *Hole Location*.
2. Dilakukan perbandingan dengan algoritma Backpropagation.



DAFTAR PUSTAKA

- [FAU-93] Fausett, L. (1993). *Addamentals of Neural Networks Architectures, Algorithms, and Applications*. Florida: Addison-Wesley Publishing Company, inc.
- [GAT-06] Gatos, B., Pratikakis, I., & Perantonis, S. J. (2006). Hybrd Off-Line Cursive Handwriting Word Recognition. *IEEE* , 1-4.
- [KRI-04] Kristanto, A. (2004). *Jaringan Syaraf Tiruan: Teori dan Aplikasi*. Yogyakarta: Andi.
- [KUR-11] Kurniawan, F., Shafry, M., Daman, D., Rehman, A., Mohamad, D., & Shamsudin, S. M. (2011). Region-Based Touched Character Segmentation in Handwritten Words. *ICIC International* , 1-14.
- [KUS-03] Kusumadewi, S. (2003). *Artificial Intelligence (Teknik dan Aplikasi)*. Yogyakarta: Graha Ilmu.
- [LEE-09] Lee, H., & Brijesh, V. (2009). Binary Segmentation with Neural Validation for Cursive Handwriting Recognition. *IEEE* , 1730-1735.
- [MUN-04] Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika Bandung.
- [SAL-05] Salameh, W. A., & Otair, M. A. (2005). Online Handwritten Character Recognition Using an Optical Backpropagation Neural Network. *Issues in Informing Science and Information Technology* , 787-795.
- [SIA-05] Siang, J. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.

[ZUR-09] Zurnawita, & Suar, Z. (2009). Algoritma Image Thinning. Elektron , 29-37.



LAMPIRAN

Lampiran 1 Data Hasil Segmentasi

Folder	File	Text Asli	Segmentasi Benar	Persentase (%)
10018	d06-025-01-1.jpg	Today	2	40.00
	d06-025-01-2-dumm.jpg	wonderful	9	44.44
	d06-025-01-2.jpg	wonderful	4	44.44
	d06-025-02-2.jpg	dawn	3	75.00
	d06-025-02-3.jpg	was	2	66.67
	d06-025-02-4.jpg	given	2	40.00
	d06-025-03-1.jpg	thought	2	28.57
	d06-025-04-1.jpg	plan	4	100.00
	d06-025-04-2.jpg	hid	3	100.00
	d06-025-04-3.jpg	secret	3	50.00
	d06-025-05-3.jpg	hand	2	50.00
	d06-025-05-4.jpg	hand	2	50.00
	d06-025-06-1.jpg	big	1	33.33
	d06-025-06-2.jpg	Because	1	14.29
	d06-025-06-3.jpg	of	2	100.00
	d06-025-07-1.jpg	plan	4	100.00
	e02-074-00-1.jpg	But	1	33.33
	e02-074-00-2.jpg	why	1	33.33
	e02-074-00-3.jpg	wood work	0	0.00
	e02-074-00-4.jpg	particular	3	30.00
	e02-074-00-5.jpg	Well	0	0.00
	e02-074-01-1.jpg	are	2	66.67
	e02-074-01-2.jpg	many	0	0.00
	e02-074-01-4.jpg	and	2	66.67
	e02-074-01-5.jpg	wood	2	50.00
	e02-074-02-1.jpg	common	2	33.33
	e02-074-02-2.jpg	making	3	50.00
	e02-074-02-3.jpg	supply	1	16.67
	e04-047-01-3.jpg	many	1	25.00
	e04-047-02-1.jpg	country	3	42.86
	e04-047-02-2.jpg	and	2	66.67
	e04-047-02-3.jpg	particularl y	6	50.00
	e04-047-03-1.jpg	Canada	2	33.33

e04-047-03-2.jpg	use	3	100.00
e04-047-03-3.jpg	commercial	4	40.00
e04-047-05-1.jpg	Over	2	50.00
e04-047-05-3.jpg	about	4	80.00
e04-047-06-1.jpg	last	0	0.00
e04-047-06-2.jpg	one	3	100.00
e04-047-06-3.jpg	made	3	75.00
e04-047-06-4.jpg	here	4	100.00
e04-047-07-1.jpg	Prices	3	50.00
e04-047-07-2.jpg	depend	2	33.33
e04-047-07-3.jpg	of	2	100.00
e04-047-08-1.jpg	on	2	100.00
e04-047-08-2.jpg	local	4	80.00
e04-047-08-3.jpg	availability	2	16.67
e04-047-09-1.jpg	and	1	33.33
e04-047-09-2.jpg	are	3	100.00
e04-047-09-3.jpg	many	2	50.00
e04-047-09-5.jpg	ways	3	75.00
e04-047-10-1.jpg	can	3	100.00
e04-047-10-2.jpg	be	2	100.00
e04-047-10-3.jpg	saved	0	0.00
e07-076-00-1.jpg	The	1	33.33
e07-076-00-2.jpg	main	1	25.00
e07-076-00-3.jpg	day	0	0.00
e07-076-01-1.jpg	even	4	100.00
e07-076-01-2.jpg	reasonably	3	30.00
e07-076-02-1.jpg	annual	1	16.67
e07-076-02-2.jpg	around	3	50.00
e07-076-03-1.jpg	per	3	100.00
e07-076-03-2.jpg	cont	0	0.00
e07-076-03-3.jpg	and	3	100.00
e07-076-03-4.jpg	where	1	20.00
e07-076-03-5.jpg	handling	6	75.00
e07-076-03-6.jpg	and	3	100.00
e07-076-04-1.jpg	good	2	50.00
e07-076-04-2.jpg	are	1	33.33
f04-053-00-2.jpg	which	3	60.00
f04-053-00-3.jpg	had	0	0.00
f04-053-00-4.jpg	dragged	3	42.86

f04-053-00-5.jpg	with	0	0.00
f04-053-01-2.jpg	had	2	66.67
f04-053-01-3.jpg	to	2	100.00
f04-053-01-4.jpg	be	2	100.00
f04-053-02-3.jpg	person	0	0.00
f04-053-03-1.jpg	unknown	4	57.14
f04-053-03-3.jpg	due	1	33.33
f04-053-03-4.jpg	of	2	100.00
f04-053-03-5.jpg	Wedgwood	0	0.00
f04-053-04-1.jpg	had	3	100.00
f04-053-04-3.jpg	The	0	0.00
f04-053-04-4.jpg	of	2	100.00
f04-053-042.jpg	been	3	75.00
g04-081-00-2.jpg	was	2	66.67
g04-081-00-3.jpg	leave	3	60.00
g04-081-00-4.jpg	and	1	33.33
g04-081-02-1.jpg	Purely	0	0.00
g04-081-02-2.jpg	she	3	100.00
g04-081-02-3.jpg	come	0	0.00
g04-081-02-4.jpg	and	3	100.00
g04-081-02-5.jpg	me	0	0.00
g04-081-02-6.jpg	was	3	100.00
g04-081-03-1.jpg	real	2	50.00
g04-081-03-4.jpg	yeoman	3	50.00
g04-081-03-6.jpg	and	1	33.33
g04-081-03-8.jpg	all	0	0.00
g04-081-032.jpg	things	6	100.00
g04-081-05-1.jpg	who	0	0.00
g04-081-05-3.jpg	how	1	33.33
g04-081-05-5.jpg	look	2	50.00
g04-081-05-7.jpg	who	1	33.33
g04-081-06-1.jpg	who	1	33.33
g04-081-06-5.jpg	of	0	0.00
g04-081-06-6.jpg	house	2	40.00
g04-081-06-7.jpg	when	0	0.00
h02-049-01-2.jpg	treasure	1	12.50
h02-049-02-2.jpg	empowered	0	0.00
h02-049-03-1.jpg	tribunal	3	37.50
h02-049-03-2.jpg	draw	2	50.00

	h02-049-05-1.jpg	dangerous	7	77.78
	h02-049-06-1.jpg	such	1	25.00
	h02-049-06-2.jpg	doctor	5	83.33
	h02-049-08-3.jpg	offence	5	71.43
	h02-049-10-1.jpg	for	1	33.33
	h02-049-10-2.jpg	dangerous	7	77.78
	h07-025-00-1.jpg	and	3	100.00
	h07-025-00-2.jpg	Peterlee	2	25.00
	h07-025-00-3.jpg	The	1	33.33
	h07-025-00-4.jpg	corby	1	20.00
	h07-025-01-1.jpg	need	3	75.00
	h07-025-01-3.jpg	prime	1	20.00
	h07-025-01-5.jpg	has	3	100.00
	h07-025-03-2.jpg	are	3	100.00
	Rata-rata			50.49
10217	b09-710z-01-1.jpg	has	3	100.00
	b09-710z-02-1.jpg	book	4	100.00
	b09-710z-02-2.jpg	at	2	100.00
	b09-710z-02-3.jpg	advance	7	100.00
	b09-710z-04-2.jpg	paper	5	100.00
	b09-710z-05-2.jpg	murray	3	50.00
	b09-710z-05-3.jpg	yours	5	100.00
	b09-710z-06-1.jpg	Threipland	4	40.00
	e01-049z-01-1.jpg	Obtainable	1	10.00
	e01-049z-02-1.jpg	your	4	100.00
	e01-049z-02-2.jpg	work	4	100.00
	e01-049z-04-2.jpg	worker	1	16.67
	e01-049z-04-4.jpg	number	1	16.67
	e01-049z-05-2.jpg	of	2	100.00
	e01-049z-05-3.jpg	upon	4	100.00
	g06-150z-02-1.jpg	moved	1	20.00
	g06-150z-02-2.jpg	his	1	33.33
	g06-150z-02-4.jpg	and	2	66.67
	g06-150z-04-1.jpg	there	5	100.00
	g06-150z-06-1.jpg	place	2	40.00
	n02-163z-01-1.jpg	would	2	40.00
	n02-163z-01-2.jpg	have	3	75.00
	n02-163z-01-3.jpg	cared	5	100.00
	n02-163z-02-1.jpg	her	3	100.00
	n02-163z-03-1.jpg	The	3	100.00

	n02-163z-03-2.jpg	hurt	4	100.00
	n02-163z-04-1.jpg	her	2	66.67
	n02-163z-04-3.jpg	of	2	100.00
	n02-163z-05-1.jpg	had	2	66.67
	n02-163z-05-2.jpg	apparent	4	50.00
	n02-163z-07-1.jpg	come	2	50.00
	p04-468z-02-1.jpg	you	0	0.00
	p04-468z-02-2.jpg	have	4	100.00
	p04-468z-02-3.jpg	me	1	50.00
	p04-468z-02-4.jpg	before	3	50.00
	p04-468z-03-2.jpg	and	3	100.00
	p04-468z-03-3.jpg	when	2	50.00
	p04-468z-03-4.jpg	my	2	100.00
	p04-468z-03-5.jpg	body	2	50.00
	p04-468z-04-5.jpg	traced	1	16.67
	p04-468z-05-1.jpg	back	4	100.00
	p04-468z-05-2.jpg	your	4	100.00
	p04-468z-05-3.jpg	gun	3	100.00
	r09-638z-01-1.jpg	middle	4	66.67
	r09-638z-01-2.jpg	age	1	33.33
	r09-638z-01-3.jpg	has	3	100.00
	r09-638z-01-4.jpg	no	2	100.00
	r09-638z-02-1.jpg	nobody	4	66.67
	r09-638z-02-2.jpg	not	3	100.00
	r09-638z-02-3.jpg	like	2	50.00
	r09-638z-03-1.jpg	dear	2	50.00
	r09-638z-03-2.jpg	this	0	0.00
	r09-638z-03-3.jpg	Once	4	100.00
	r09-638z-03-4.jpg	having	6	100.00
	r09-638z-04-1.jpg	his	3	100.00
	r09-638z-06-2.jpg	makes	0	0.00
	Rata-rata			70.09
10218	b10-739z-03-1.jpg	Bunbury	4	57.14
	b10-739z-03-2.jpg	with	2	50.00
	b10-739z-06-1.jpg	achieve	5	71.43
	e02-077z-01-1.jpg	more	2	50.00
	e02-077z-02-1.jpg	however	7	100.00
	e02-077z-02-2.jpg	worked	2	33.33
	e02-077z-03-2.jpg	and	3	100.00
	e02-077z-04-2.jpg	operation	4	44.44

e02-077z-05-2.jpg	acurate	3	42.86
e02-077z-05-3.jpg	but	2	66.67
e02-077z-06-1.jpg	workmans hip	6	54.55
e02-077z-06-2.jpg	perhaps	1	14.29
e02-077z-07-1.jpg	awkward	4	57.14
e02-077z-07-3.jpg	that	4	100.00
e02-077z-07-5.jpg	plane	1	20.00
e02-077z-08-1.jpg	can only	3	37.50
g06-172z-03-1.jpg	he had been	7	63.64
g06-172z-04-1.jpg	had	3	100.00
g06-172z-05-1.jpg	had	3	100.00
g06-172z-07-1.jpg	today	5	100.00
g06-172z-08-3.jpg	been	1	25.00
k03-154z-01-2.jpg	heard	5	100.00
k03-154z-02-1.jpg	behind	0	0.00
k03-154z-03-1.jpg	my	0	0.00
k03-154z-04-1.jpg	moment	3	50.00
l05-577z-01-1.jpg	and	3	100.00
l05-577z-01-2.jpg	the	3	100.00
l05-577z-01-3.jpg	good	4	100.00
l05-577z-01-4.jpg	her	3	100.00
l05-577z-02-1.jpg	everything	2	20.00
l05-577z-02-2.jpg	one	3	100.00
l05-577z-03-1.jpg	can	0	0.00
l05-577z-03-2.jpg	keep	2	50.00
l05-577z-05-1.jpg	you	2	66.67
l05-577z-05-2.jpg	forgot	2	33.33
n03-188z-01-1.jpg	got	3	100.00
n03-188z-02-3.jpg	even	2	50.00
n03-188z-04-1.jpg	aint	4	100.00
n03-188z-04-2.jpg	you	1	33.33
n03-188z-04-3.jpg	by	2	100.00
p04-492z-01-1.jpg	next	2	50.00
p04-492z-01-2.jpg	out	2	66.67
p04-492z-02-2.jpg	knew	2	50.00
p04-492z-02-3.jpg	he	2	100.00
p04-492z-03-1.jpg	her	2	66.67
p04-492z-03-2.jpg	darling	2	28.57
r09-664z-01-1.jpg	But	3	100.00

r09-664z-01-2.jpg	now	1	33.33
r09-664z-01-3.jpg	grateful	3	37.50
r09-664z-01-4.jpg	for	1	33.33
r09-664z-02-1.jpg	none	4	100.00
r09-664z-03-1.jpg	dummy	1	20.00
r09-664z-03-2.jpg	run	3	100.00
r09-664z-03-3.jpg	more	4	100.00
r09-664z-04-1.jpg	when	4	100.00
r09-664z-05-1.jpg	man	1	33.33
r09-664z-05-4.jpg	One	3	100.00
Untitled.jpg	saya	4	100.00
Rata-rata			63.98
Rata-rata Seluruhnya			61.52

Lampiran 2 Data Hasil Pengenalan

Fold er	File	Text Asli	Text yang Dikenali	Pengenalan Benar	Persentase (%)
100 18	d06-025-01-1.jpg	Today	hay	2	40.00
	d06-025-01-2-dumm.jpg	wonderfu l	onde	3	33.33
	d06-025-01-2.jpg	wonderfu l	_ond__e	3	33.33
	d06-025-02-2.jpg	dawn	dauun	3	75.00
	d06-025-02-3.jpg	was	wa_	2	66.67
	d06-025-02-4.jpg	given	ge_	1	20.00
	d06-025-03-1.jpg	thought	_a__f	0	0.00
	d06-025-04-1.jpg	plan	nlan	3	75.00
	d06-025-04-2.jpg	hid	hmd	2	66.67
	d06-025-04-3.jpg	secret	secd	3	50.00
	d06-025-05-3.jpg	hand	eeend	2	50.00
	d06-025-05-4.jpg	hand	_wvd	1	25.00
	d06-025-06-1.jpg	big	_g	1	33.33

d06-025-06-2.jpg	Because	ha__	1	14.29
d06-025-06-3.jpg	of	of	2	100.00
d06-025-07-1.jpg	plan	plan	4	100.00
e02-074-00-1.jpg	But	et	1	33.33
e02-074-00-2.jpg	why	_y	1	33.33
e02-074-00-3.jpg	wood work	_e__	0	0.00
e02-074-00-4.jpg	particular	arne_r	3	30.00
e02-074-00-5.jpg	Well	__	0	0.00
e02-074-01-1.jpg	are	a_e	2	66.67
e02-074-01-2.jpg	many	n_y	0	0.00
e02-074-01-4.jpg	and	_nd	2	66.67
e02-074-01-5.jpg	wood	__od	2	50.00
e02-074-02-1.jpg	common	co__y	2	33.33
e02-074-02-2.jpg	making	_am_g	2	33.33
e02-074-02-3.jpg	supply	_y__y	1	16.67
e04-047-01-3.jpg	many	_yy	1	25.00
e04-047-02-1.jpg	country	sou_y	2	28.57
e04-047-02-2.jpg	and	a_d	2	66.67
e04-047-02-3.jpg	particular ly	artr__a_y	5	41.67
e04-047-03-1.jpg	Canada	aate_	1	16.67
e04-047-03-2.jpg	use	use	3	100.00
e04-047-03-3.jpg	commercial	son__en_m	2	20.00
e04-047-05-1.jpg	Over	ve	2	50.00
e04-047-05-3.jpg	about	abo__t	4	80.00
e04-047-06-1.jpg	last	__	0	0.00
e04-047-06-2.jpg	one	one	3	100.00
e04-047-06-3.jpg	made	r__ade	3	75.00
e04-047-06-4.jpg	here	here	4	100.00
e04-047-07-1.jpg	Prices	nnee_	1	16.67
e04-047-07-2.jpg	depend	d_d	1	16.67
e04-047-07-3.jpg	of	of	2	100.00
e04-047-08-1.jpg	on	on	2	100.00
e04-047-08-2.jpg	local	eoca_	3	60.00
e04-047-08-3.jpg	availability	aaab_e_	2	16.67
e04-047-09-1.jpg	and	ad	1	33.33

e04-047-09-2.jpg	are	are	3	100.00
e04-047-09-3.jpg	many	o_ny	2	50.00
e04-047-09-5.jpg	ways	way_	3	75.00
e04-047-10-1.jpg	can	ean	2	66.67
e04-047-10-2.jpg	be	be	2	100.00
e04-047-10-3.jpg	saved	_eo_	0	0.00
e07-076-00-1.jpg	The	_e	1	33.33
e07-076-00-2.jpg	main	namn	0	0.00
e07-076-00-3.jpg	day	_	0	0.00
e07-076-01-1.jpg	even	even	4	100.00
e07-076-01-2.jpg	reasonabl y	_aoo_d_	2	20.00
e07-076-02-1.jpg	annual	a_d	1	16.67
e07-076-02-2.jpg	around	a_and	3	50.00
e07-076-03-1.jpg	per	ner	2	66.67
e07-076-03-2.jpg	cont	d	0	0.00
e07-076-03-3.jpg	and	and	3	100.00
e07-076-03-4.jpg	where	rh_	1	20.00
e07-076-03-5.jpg	handling	handely	6	75.00
e07-076-03-6.jpg	and	ane	2	66.67
e07-076-04-1.jpg	good	go_l	2	50.00
e07-076-04-2.jpg	are	an	1	33.33
f04-053-00-2.jpg	which	wh_h	3	60.00
f04-053-00-3.jpg	had	_e	0	0.00
f04-053-00-4.jpg	dragged	d_gg_	3	42.86
f04-053-00-5.jpg	with	_	0	0.00
f04-053-01-2.jpg	had	anae	0	0.00
f04-053-01-3.jpg	to	to	2	100.00
f04-053-01-4.jpg	be	be	2	100.00
f04-053-02-3.jpg	person	_y_	0	0.00
f04-053-03-1.jpg	unknown	ushwn	3	42.86
f04-053-03-3.jpg	due	dy	1	33.33
f04-053-03-4.jpg	of	of	2	100.00
f04-053-03-5.jpg	Wedgwo od	a__e	0	0.00
f04-053-04-1.jpg	had	had	3	100.00
f04-053-04-3.jpg	The	_	0	0.00
f04-053-04-4.jpg	of	of	2	100.00
f04-053-042.jpg	been	bee_	3	75.00
g04-081-00- 2.jpg	was	b_os	1	33.33

g04-081-00-3.jpg	leave	_ave	3	60.00
g04-081-00-4.jpg	and	a__	1	33.33
g04-081-02-1.jpg	Purely	f_n_	0	0.00
g04-081-02-2.jpg	she	she	3	100.00
g04-081-02-3.jpg	come	a	0	0.00
g04-081-02-4.jpg	and	and	3	100.00
g04-081-02-5.jpg	me	n_	0	0.00
g04-081-02-6.jpg	was	wos	2	66.67
g04-081-03-1.jpg	real	_ae	1	25.00
g04-081-03-4.jpg	yeoman	yeo_a	3	50.00
g04-081-03-6.jpg	and	a_el	1	33.33
g04-081-03-8.jpg	all	__	0	0.00
g04-081-032.jpg	things	tneasun	5	83.33
g04-081-05-1.jpg	who	__y	0	0.00
g04-081-05-3.jpg	how	do	1	33.33
g04-081-05-5.jpg	look	lo_	1	25.00
g04-081-05-7.jpg	who	_o_	1	33.33
g04-081-06-1.jpg	who	w_	1	33.33
g04-081-06-5.jpg	of	v	0	0.00
g04-081-06-6.jpg	house	ow_	1	20.00
g04-081-06-7.jpg	when	a	0	0.00
h02-049-01-2.jpg	treasure	ns	1	12.50
h02-049-02-2.jpg	empower ed	__n_e	0	0.00
h02-049-03-	tribunal	_hnal	3	37.50

	1.jpg				
	h02-049-03-2.jpg	draw	daw	2	50.00
	h02-049-05-1.jpg	dangerous	dani_ous	6	66.67
	h02-049-06-1.jpg	such	ue	1	25.00
	h02-049-06-2.jpg	doctor	octor	5	83.33
	h02-049-08-3.jpg	offence	o_ence	5	71.43
	h02-049-10-1.jpg	for	_r	1	33.33
	h02-049-10-2.jpg	dangerous	dang_our	6	66.67
	h07-025-00-1.jpg	and	and	3	100.00
	h07-025-00-2.jpg	Peterlee	_gree	2	25.00
	h07-025-00-3.jpg	The	_e	1	33.33
	h07-025-00-4.jpg	corby	ar__	1	20.00
	h07-025-01-1.jpg	need	_eed	3	75.00
	h07-025-01-3.jpg	prime	r_n_	1	20.00
	h07-025-01-5.jpg	has	has	3	100.00
	h07-025-03-2.jpg	are	are	3	100.00
	Rata-rata				44.96
102 17	b09-710z-01-1.jpg	has	has	3	100.00
	b09-710z-02-1.jpg	book	eoom	2	50.00
	b09-710z-02-2.jpg	at	at	2	100.00
	b09-710z-02-3.jpg	advance	advanee	6	85.71
	b09-710z-04-2.jpg	paper	nar__	1	20.00
	b09-710z-05-2.jpg	murray	iyay	2	33.33
	b09-710z-05-3.jpg	yours	you_e	3	60.00

b09-710z-06-1.jpg	Threipla nd	fh__end	3	30.00
e01-049z-01-1.jpg	Obtainab le	O_d_	1	10.00
e01-049z-02-1.jpg	your	you_	3	75.00
e01-049z-02-2.jpg	work	wori	3	75.00
e01-049z-04-2.jpg	worker	_n__r	1	16.67
e01-049z-04-4.jpg	number	_yar	1	16.67
e01-049z-05-2.jpg	of	oh	1	50.00
e01-049z-05-3.jpg	upon	unon	3	75.00
g06-150z-02-1.jpg	moved	nywy_	0	0.00
g06-150z-02-2.jpg	his	h_	1	33.33
g06-150z-02-4.jpg	and	ano_	2	66.67
g06-150z-04-1.jpg	there	ufe_e	2	40.00
g06-150z-06-1.jpg	place	ueee	1	20.00
n02-163z-01-1.jpg	would	_imd	1	20.00
n02-163z-01-2.jpg	have	_ave	3	75.00
n02-163z-01-3.jpg	cared	cared	5	100.00
n02-163z-02-1.jpg	her	her	3	100.00
n02-163z-03-1.jpg	The	bhe	2	66.67
n02-163z-03-2.jpg	hurt	hu_ru	3	75.00
n02-163z-04-1.jpg	her	he	2	66.67
n02-163z-04-3.jpg	of	oh	1	50.00
n02-163z-05-1.jpg	had	h_d	2	66.67
n02-163z-05-2.jpg	apparent	annad	2	25.00

n02-163z-07-1.jpg	come	con_	2	50.00
p04-468z-02-1.jpg	you	—	0	0.00
p04-468z-02-2.jpg	have	have	4	100.00
p04-468z-02-3.jpg	me	r_e	1	50.00
p04-468z-02-4.jpg	before	be_eufe	3	50.00
p04-468z-03-2.jpg	and	and	3	100.00
p04-468z-03-3.jpg	when	__en	2	50.00
p04-468z-03-4.jpg	my	iy	1	50.00
p04-468z-03-5.jpg	body	__dy	2	50.00
p04-468z-04-5.jpg	traced	tyn	1	16.67
p04-468z-05-1.jpg	back	dabg	1	25.00
p04-468z-05-2.jpg	your	you_r	4	100.00
p04-468z-05-3.jpg	gun	gun	3	100.00
r09-638z-01-1.jpg	middle	n_ddle	4	66.67
r09-638z-01-2.jpg	age	aa_	1	33.33
r09-638z-01-3.jpg	has	has	3	100.00
r09-638z-01-4.jpg	no	no	2	100.00
r09-638z-02-1.jpg	nobody	noedy	3	50.00
r09-638z-02-2.jpg	not	nop	2	66.67
r09-638z-02-3.jpg	like	_ke	2	50.00
r09-638z-03-1.jpg	dear	_ar	2	50.00
r09-638z-03-2.jpg	this	_e	0	0.00
r09-638z-03-3.jpg	Once	Onne	3	75.00

	r09-638z-03-4.jpg	having	hav_ng	5	83.33
	r09-638z-04-1.jpg	his	hls	2	66.67
	r09-638z-06-2.jpg	makes	_a	0	0.00
	Rata-rata				55.10
102 18	b10-739z-03-1.jpg	Bunbury	aueny	2	28.57
	b10-739z-03-2.jpg	with	w_h	2	50.00
	b10-739z-06-1.jpg	achieve	aeheve	4	57.14
	e02-077z-01-1.jpg	more	yre	2	50.00
	e02-077z-02-1.jpg	however	hogeve	6	85.71
	e02-077z-02-2.jpg	worked	aded	2	33.33
	e02-077z-03-2.jpg	and	and	3	100.00
	e02-077z-04-2.jpg	operation	ope__n	4	44.44
	e02-077z-05-2.jpg	acurate	aauae	3	42.86
	e02-077z-05-3.jpg	but	eu_	1	33.33
	e02-077z-06-1.jpg	workman ship	wonam__h_	4	36.36
	e02-077z-06-2.jpg	perhaps	a__p_	1	14.29
	e02-077z-07-1.jpg	awkward	aw_n_d	3	42.86
	e02-077z-07-3.jpg	that	that	4	100.00
	e02-077z-07-5.jpg	plane	pi	1	20.00
	e02-077z-08-1.jpg	can only	c__on__	3	37.50
	g06-172z-03-1.jpg	he had been	he had ban	7	63.64
	g06-172z-04-1.jpg	had	had	3	100.00
	g06-172z-05-1.jpg	had	had	3	100.00
	g06-172z-07-	today	today	5	100.00

1.jpg				
g06-172z-08-3.jpg	been	by_	1	25.00
k03-154z-01-2.jpg	heard	heard	5	100.00
k03-154z-02-1.jpg	behind	—	0	0.00
k03-154z-03-1.jpg	my	y	0	0.00
k03-154z-04-1.jpg	moment	i_o__nt	3	50.00
l05-577z-01-1.jpg	and	and	3	100.00
l05-577z-01-2.jpg	the	the	3	100.00
l05-577z-01-3.jpg	good	good	4	100.00
l05-577z-01-4.jpg	her	her	3	100.00
l05-577z-02-1.jpg	everythin g	__yh_y	1	10.00
l05-577z-02-2.jpg	one	one	3	100.00
l05-577z-03-1.jpg	can	a__	0	0.00
l05-577z-03-2.jpg	keep	g_n	0	0.00
l05-577z-05-1.jpg	you	yo_	2	66.67
l05-577z-05-2.jpg	forgot	__op	1	16.67
n03-188z-01-1.jpg	got	got	3	100.00
n03-188z-02-3.jpg	even	ev__	2	50.00
n03-188z-04-1.jpg	aint	aent	3	75.00
n03-188z-04-2.jpg	you	y__	1	33.33
n03-188z-04-3.jpg	by	by	2	100.00
p04-492z-01-1.jpg	next	_yvt	1	25.00
p04-492z-01-2.jpg	out	ou_	2	66.67
p04-492z-02-	knew	kn__	2	50.00

2.jpg				
p04-492z-02-3.jpg	he	he	2	100.00
p04-492z-03-1.jpg	her	he_	2	66.67
p04-492z-03-2.jpg	darling	da_	2	28.57
r09-664z-01-1.jpg	But	aut	2	66.67
r09-664z-01-2.jpg	now	myw	1	33.33
r09-664z-01-3.jpg	grateful	_atad	2	25.00
r09-664z-01-4.jpg	for	_r	1	33.33
r09-664z-02-1.jpg	none	none	4	100.00
r09-664z-03-1.jpg	dummy	_n_y	1	20.00
r09-664z-03-2.jpg	run	ru_n	3	100.00
r09-664z-03-3.jpg	more	iore	3	75.00
r09-664z-04-1.jpg	when	yhen	3	75.00
r09-664z-05-1.jpg	man	i_r	0	0.00
r09-664z-05-4.jpg	One	One	3	100.00
Untitled.jpg	saya	saya	4	100.00
Rata-rata				57.45
Rata-rata Seluruhnya				52.50

Lampiran 3 Data Kode Karakter

KARAKTER	KODE	KARAKTER	KODE
A	000001	a	011011
B	000010	b	011100
C	000011	c	011101
D	000100	d	011110
E	000101	e	011111
F	000110	f	100000
G	000111	g	100001
H	001000	h	100010
I	001001	i	100011
J	001010	j	100100
K	001011	k	100101
L	001100	l	100110
M	001101	m	100111
N	001110	n	101000
O	001111	o	101001
P	010000	p	101010
Q	010001	q	101011
R	010010	r	101100
S	010011	s	101101
T	010100	t	101110
U	010101	u	101111
V	010110	v	110000
W	010111	w	110001
X	011000	x	110010
Y	011001	y	110011
Z	011010	z	110100