

**[4] PENGENALAN TULISAN TANGAN HURUF HANGUL
DENGAN MENGGUNAKAN JARINGAN SARAF TIRUAN PROPAGASI
BALIK**

SKRIPSI

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Program Studi Teknik Informatika



Oleh :
Purbarini Sulysthian
135314125

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2018**

**PENGENALAN TULISAN TANGAN HURUF HANGUL
DENGAN MENGGUNAKAN JARINGAN SARAF TIRUAN PROPAGASI
BALIK**

SKRIPSI

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Program Studi Teknik Informatika



Oleh :
Purbarini Sulysthian
135314125

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2018**

**RECOGNITION HAND WRITTEN HANGUL CHARACTERS USING
ARTIFICIAL NEURAL NETWORK BACK PROPAGATION**

THESIS

Presented as Partial Fulfillment of the Requirements
to Obtain Sarjana Komputer Degree
in Informatic Engineering Department



Created By :
Purbarini Sulysthian
135314125

INFORMATICS ENGINEERING STUDY PROGRAM

DEPARTMENT OF INFORMATICS ENGINEERING

FACULTY OF SCIENCE AND TECHNOLOGY

SANATA DHARMA UNIVERSITY

YOGYAKARTA

2018

HALAMAN PERSETUJUAN

SKRIPSI

PENGENALAN TULISAN TANGAN HURUF HANGUL
DENGAN MENGGUNAKAN JARINGAN SARAF TIRUAN PROPAGASI
BALIK



Dosen Pembimbing

Dr. Anastasia Rita Widiarti

Tanggal 15 Desember 2017

HALAMAN PENGESAHAN

SKRIPSI

Pengenalan Tulisan Tangan Huruf Hangul Dengan
Menggunakan Jaringan Saraf Tiruan Propagasi Balik

Dipersiapkan dan ditulis oleh:

PURBARINI SULYSTHIAN

NIM : 135314125

Telah dipertahankan di depan Panitia Penguji

Pada tanggal 17 Januari 2018

Dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Nama Lengkap

Tanda Tangan

Ketua Alb. Agung Hadhiatma, M.T.

Sekretaris Eko Hari Parmadi, M.Kom

Anggota I Dr Anastasia Rita Widiarti




Yogyakarta, 26 Januari 2018

Fakultas Sains dan Teknologi

Universitas Sanata Dharma

Dekan




Sudi Mungkasi, S.Si., M.Math.Sc., Ph.D.

MOTTO

“Pertolonganku ialah dari Tuhan, yang menjadikan langit dan bumi”

Mazmur 121:2

행복하자 그리고 아프지 말고

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa di dalam skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah

Yogyakarta, 26 Januari 2018

Penulis



Purbarini Sulysthian

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS**

Yang bertanda tangan di bawah ini, saya mahasiswa Universitas Sanata Dharma:

Nama : Purbarini Sulysthian

NIM : 135314125

Demi pengembangan ilmu pengetahuan, saya memberikan kepada Perpustakaan Universitas Sanata Dharma karya ilmiah yang berjudul:

**Pengenalan Tulisan Tangan Huruf Hangul Dengan
Menggunakan Jaringan Saraf Tiruan Propagasi Balik**

Beserta Perangkat yang diperlukan (bila ada). Dengan demikian saya memberikan kepada perpustakaan Sanata Dharma hak untuk menyimpan, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data, mendistribusikan secara terbatas dan mempublikasikan di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya maupun memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini saya buat dengan sebenarnya.

Yogyakarta, 26 Januari 2018

Yang menyatakan



Purbarini Sulysthian

ABSTRAK

Perkembangan teknologi semakin memudahkan manusia dalam mengakses dan menerima informasi dari berbagai hal. Berbagai hal dapat diakses dengan mudah, murah, dimana saja, dan kapan saja. Salah satu yang kita dapat akses adalah kebudayaan dari berbagai macam negara, contohnya adalah Korea Selatan. Negara dengan budaya K-Pop nya yang saat ini sedang hangat dibicarakan oleh anak muda Indonesia. Berbicara tentang budaya K-Pop dan kebudayaan Korea maka berbicara juga tentang Bahasa Korea. Bahasa Korea adalah bahasa yang menggunakan aksara dalam penulisannya, aksara ini disebut dengan Hangul.

Dengan banyaknya masyarakat yang belajar bahasa Korea dan banyaknya turis Indonesia yang datang ke Korea, maka dibutuhkan suatu teknologi yang dapat memberikan informasi kepada penggunanya tentang bahasa Korea. Dalam penelitian ini akan membuat sistem dengan menggunakan salah satu penerapan dari Jaringan Saraf Tiruan. Sistem ini dibangun dengan Jaringan Saraf Tiruan Propagasi Balik sebagai pemodelan system dan Intensity of Character dan Mark Direction sebagai metode ekstraksi cirinya. Sistem ini nantinya diharapkan mampu mengenali pola tulisan huruf Hangul secara otomatis dan memberikan informasi terkait dengan pola yang dikenalnya.

Percobaan proses pengenalan dengan menggunakan Jaringan Saraf Tiruan Propagasi Balik dengan melakukan optimalisasi arsitektur jaringan dan menghasilkan akurasi 80,4762%. Percobaan ini menggunakan 2 lapisan tersembunyi dengan jumlah neuron pada lapisan tersembunyi 1 sebanyak 100 dan pada lapisan tersembunyi 2 sebanyak 95 neuron.

Kata Kunci: Pengenalan Pola, Hangul, *Intensity of Character*, *Mark Direction*, Jaringan Saraf Tiruan Propagasi Balik

ABSTRACT

The development of technology makes it easier for human to access and receive information from various things. Various things can be accessed easily, cheaply, anywhere, and anytime. One that we can access is the culture of various countries, for example is South Korea. The country with its K-Pop culture is currently being warmly discussed by young Indonesians. Speaking of K-Pop culture and Korean culture then speak also about Korean Language. Korean is a language that uses script in writing, this script is called Hangul.

With so many people learning Korean and the number of Indonesian tourists coming to Korea, it needs a technology that can give users information about Korean language. In this research will make the system by using one application of Neural Network. This system is built with Artificial Back Propagation Neural Networks as system modeling and Intensity of Character and Mark Direction as a method of character extraction. This system is expected to be able to recognize Hangul pattern writing automatically and provide information related to familiar pattern.

Attempt the introduction process by using Artificial Propagation Neural Network by optimizing the network architecture and yield 80,4762% accuracy. This experiment uses 2 hidden layers with the number of neurons in the hidden layer 1 of 100 and in the hidden layer 2 of 95 neurons.

Keywords: *Pattern Introduction, Hangul, Intensity of Character, Mark Direction, Artificial Neural Network Back Propagation*

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Tuhan Yesus Kristus oleh karena berkat dan kasih-Nya sehingga tugas akhir saya yang berjudul “PENGENALAN TULISAN TANGAN HURUF HANGUL DENGAN MENGGUNAKAN JARINGAN SARAF TIRUAN PROPAGASI BALIK” dapat diselesaikan dengan baik. Tugas akhir ini merupakan salah satu persyaratan wajib ditempuh pada Program Studi Teknik Informatika Universitas Sanata Dharma untuk memperoleh gelar Sarjana Komputer. Selama persiapan proposal dan penyusunan tugas akhir ini saya mendapat banyak dukungan dan bantuan dari berbagai pihak, sehingga saya menyampaikan terimakasih kepada:

1. Tuhan Yesus Kristus yang membuat segala sesuatu indah pada waktu-Nya dan bukan indah pada waktu manusia. Pribadi yang menyertai saya dari awal penulisan hingga dapat menyelesaikan tugas akhir ini, saya tahu benar jika tanpa pernyertaan dan hikmat-Nya saya tidak bisa sampai di titik ini.
2. Ibu Dr. Anastasia Rita, selaku dosen pembimbing skripsi yang telah memberikan banyak arahan, kekuatan, masukan, motivasi, dan kesempatan selama menyelesaikan skripsi.
3. Kepala Program Studi dan Mahasiswa Semester 5 jurusan Bahasa Korea Universitas Gadjah Mada, yang telah bersedia memberikan ijin dan data tulisan tangan tulisan tangan huruf Hangul.
4. Kedua orang tua saya tercinta Bapak Eddy dan Ibu Ning, yang telah banyak memberikan dukungan moral dan materi, yang selalu mempercayai apapun yang saya lakukan hingga penulisan tugas ini berakhir, dan yang selalu sabar dalam menunggu saya menyelesaikan tugas akhir ini.
5. Adik-adikku Sari, Chandra, dan Berlian yang selalu memotivasi saya untuk cepat menyelesaikan tugas akhir ini dan yang selalu bersama menjalani proses hidup dalam keluarga kita.
6. Keluarga rohaniku tercinta di Youth Impact, komsel We Are One dan komsel Caritas yang selalu memberikan motivasi saya bahwa saat berjalan

dengan Tuhan, semuanya pasti mungkin dan berhasil. *So proud to be youth impact!*

7. Sahabat-sahabatku Valencia Maria, Angela Mediatrice, Windia Sahara, Kasih Handoyo, Irenius Kristanto, Satrio Bagus, Yohanes Enggal yang selama ini telah menjadi keluarga saya selama kuliah. Terimakasih untuk semua kenangan dan perjuangan kita selama perkuliahan ini.
8. Teman-teman seperjuangan di Teknik Informatika 2013 Sanata Dharma dan TI C, terimakasih untuk semua cerita selama perkuliahan ini.
9. Teman-teman satu bimbingan skripsi (Fanny, Ratri, Rusdi, Kasih, Jonathan, Tanto, Vrengky, Filemon, Bowi) yang sudah berjuang bersama dalam menyelesaikan tugas akhir ini.
10. Semua pihak yang tidak dapat disebutkan satu persatu dalam membantu penulis dalam pengerjaan tugas akhir ini.

Penulis menyadari masih banyak kekurangan dari tugas akhir ini sehingga penulis mengharapkan kritik dan saran yang bersifat membangun untuk menyempurnakannya. Semoga tulisan tugas akhir ini dapat bermanfaat bagi siapapun yang membaca dan saya khususkan bagi penulis.

Yogyakarta, 26 Januari 2018

Penulis



Purbarini Sulysthian

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN JUDUL (INGGRIS).....	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
MOTTO	v
PERNYATAAN KEASLIAN KARYA	vi
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK	viii
ABSTRACT	ix
KATA PENGANTAR.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
BAB I.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	4
1.6 Metode Penelitian.....	4
1.7. Sistematika Penulisan Tugas Akhir.....	5
BAB II	7
2.1 Pengenalan Pola	7
2.2 Huruf Hangul.....	7
2.3 Pemrosesan Citra Digital.....	10
2.4 Ekstraksi Fitur (Feature Extraction)	10
2.4.1 Intensity of Character.....	10
2.4.2 Mark Direction.....	10

2.4.3 Zonning.....	11
2.5 Jaringan Saraf Tiruan	13
2.6 Jaringan Saraf Tiruan Propagasi Balik (<i>Backpropagation</i>).....	13
2.6.1 Pengertian Jaringan Saraf Tiruan Propagasi Balik	13
2.6.2 Algoritma Jaringan Saraf Tiruan Propagasi Balik.....	14
2.6.3 Simbol-simbol yang digunakan dalam Jaringan Saraf Tiruan Propagasi Balik.....	15
2.6.4 Fungsi Aktivasi	16
2.7 <i>Mean Square Error</i>	20
BAB III.....	21
3.1 Bahan.....	21
3.2 Peralatan Penelitian	21
3.2.1 Hardware.....	21
3.2.2 Software	21
3.3 Tahap-tahap penelitian	22
3.3.1 Pengumpulan data dan kebutuhan	22
3.3.2 Pengolahan Data	23
3.3.4 Analisis data.....	28
3.4 Gambaran Umum Sistem	28
3.5 Perancangan Program.....	30
3.6 Perancangan Antarmuka Sistem.....	38
3.7 Skenario Pengujian.....	40
BAB IV	42
4.1 Contoh Hasil Antarmuka.....	42
4.2 Desain Jaringan	45
4.3 Hasil Uji	46
4.4 Testing Data Tunggal	66
4.5 Analisis Data	67
BAB V.....	70
5.1 Kesimpulan.....	70
5.2 Saran	70

DAFTAR PUSTAKA	72
LAMPIRAN.....	74

DAFTAR GAMBAR

Gambar 2. 1 Huruf Hangul	9
Gambar 2. 2 Struktur Mark Direction (Surinta, 2010).....	11
Gambar 2. 3 Pembagian Zona dan Perhitungan Jarak ZCZ.....	11
Gambar 2. 4 Jaringan Saraf Tiruan Backpropagation dengan 1 Lapisan Tersembunyi (Kristanto, 2004)	14
Gambar 2. 5 Binary sigmoid dengan jangkauan 0 dan 1	17
Gambar 2. 6 bipolar sigmoid dengan jangkauan -1 dan 1.....	17
Gambar 3. 1 Form Pengenalan Tulisan Tangan Huruf Hangul	22
Gambar 3. 2 Contoh Form yang Telah Diisi.....	23
Gambar 3. 3 Contoh Gambar yang Sudah di-Crop	23
Gambar 3. 4 Contoh Gambar Citra yang sudah diubah ke Citra Hitam Putih	24
Gambar 3. 5 Conth Citra yang sudah di Resize	24
Gambar 3. 6 Contoh Citra yang sudah ditipiskan	25
Gambar 3. 7 Contoh Gambar yang sudah dibagi Menjadi 4 Segmen	25
Gambar 3. 8 Contoh Citra yang sudah diekstrak Cirinya	26
Gambar 3. 9 Contoh Program Jaringan Propagasi Balik	27
Gambar 3. 10 Data Flow Diagram level 0	29
Gambar 3. 11 Data Flow Diagram level 1	29
Gambar 3. 12 Data Flow Diagram level 2	30
Gambar 3. 13 Contoh Tampilan Utama Alat Uji	38
Gambar 3. 14 Contoh Tampilan Panel Pelatihan Data	39
Gambar 3. 15 Contoh Tampilan Panel Uji Data Tunggal.....	39
Gambar 4. 1 Antarmuka Menu Utama Alat Uji.....	42
Gambar 4. 2 Antarmuka Panel Pelatihan Data.....	43
Gambar 4. 3 Contoh Hasil Pelatihan Data	43
Gambar 4. 4 Contoh Men-load Data	44
Gambar 4. 5 Antarmuka Panel Uji Data Tunggal	44
Gambar 4. 6 Contoh Hasil Pengujian Data Tunggal.....	45
Gambar 4. 7 Desain Jaringan	45
Gambar 4. 8 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 1	47
Gambar 4. 9 Hasil Perobaan 1	47
Gambar 4. 10 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 2	48
Gambar 4. 11 Hasil Percobaan 2.....	49
Gambar 4. 12 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 3	50
Gambar 4. 13 Hasil Percobaan 3.....	50
Gambar 4. 14 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 4	51
Gambar 4. 15 Hasil Percobaan 4.....	52
Gambar 4. 16 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 5	53

Gambar 4. 17 Hasil Percobaan 5.....	53
Gambar 4. 18 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 6	54
Gambar 4. 19 Hasil Percobaan 6.....	55
Gambar 4. 20 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 7	56
Gambar 4. 21 Hasil Percobaan 7.....	56
Gambar 4. 22 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 8	57
Gambar 4. 23 Hasil dari Percobaan 8.....	58
Gambar 4. 24 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 9	59
Gambar 4. 25 Hasil dari Percobaan 9.....	59
Gambar 4. 26 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 10 ..	60
Gambar 4. 27 Hasil dari Percobaan 10.....	61
Gambar 4. 28 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 11 ..	62
Gambar 4. 29 Hasil Percobaan 11	62
Gambar 4. 30 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 12 ..	63
Gambar 4. 31 Hasil dari Percobaan 12.....	64
Gambar 4. 32 Arsitektur Jaringan yang Menghasilkan Akurasi Optimal.....	68

DAFTAR TABEL

Tabel 3. 1 Tabel Contoh Hasil Ekstrasi Ciri	26
Tabel 4. 1 Tabel Pengujian Metode Training.....	65
Tabel 4. 2 Tabel Hasil Testing Data Tunggal	67

BAB I

PENDAHULUAN

1.1 Latar Belakang

Majunya dunia teknologi semakin memudahkan manusia untuk mengakses dan menerima informasi dari berbagai hal. Mendekatkan yang jauh adalah salah satu dampak nyata dari perkembangan dunia teknologi. Berbagai hal dapat kita akses dengan mudah, murah, dimana saja dan kapan saja. Salah satunya adalah kebudayaan, kita dapat mengetahui berbagai macam kebudayaan dari berbagai negara hanya dengan mengakses melalui jaringan internet saja. Berbicara tentang kebudayaan di seluruh dunia, tidak lengkap jika kita tidak berbicara tentang Korea Selatan, negara dengan berbagai kebudayaan, adat istiadat dan budaya K-Popnya. Budaya dari negara ini tidak henti-hentinya dibahas oleh anak muda Indonesia yang sedang terserang virus *Hallyu wave* (gelombang Korea). Berbicara mengenai budaya K-Pop dan kebudayaan Korea Selatan, maka berbicara juga tentang bahasa Korea. Bahasa Korea berbeda dengan bahasa lain seperti Indonesia, Inggris, Jerman, dan negara-negara lainnya yang menggunakan alfabet ABCD untuk menulis. Bahasa Korea adalah bahasa yang menggunakan aksara dalam penulisannya sama seperti Mandarin dan Jepang. Huruf ini digunakan dalam kehidupan sehari-hari, dan mudah kita temukan dimana saja, seperti lagu-lagu Korea, kebudayaan Korea, drama Korea dan lainnya. Dan hal inilah yang membuat banyak anak muda Indonesia beramai-ramai untuk belajar aksara ini agar dapat mengerti dan dapat memahami kebudayaan Korea. Hal ini diperkuat dengan adanya data dari website *kto.visitkorea.or.kr* yang menunjukkan jumlah wisatawan Indonesia yang berkunjung ke negara Ginseng ini mencapai 25,139 orang per Agustus 2016. Bahkan Indonesia menempati urutan ketujuh wisatawan terbanyak dari wisatawan seluruh dunia yang pernah berkunjung di negara Korea Selatan. Dengan adanya fenomena ini, maka dibutuhkan suatu alat yang dapat

membantu wisatawan dan penggemar budaya Korea dalam memahami bahasa Korea, khususnya aksara Hangul.

Penelitian tentang pengenalan aksara sudah pernah dan banyak dilakukan oleh beberapa orang, mulai dari aksara Jepang, Hangul, Thailand, aksara Jawa, dll. Pada penelitian pengenalan aksara Hangul sebelumnya, yang dilakukan oleh Yoon, Wook dan Joon (1999) hasil keakuratan yang dihasilkan cukup tinggi, yaitu mencapai 96,5% pada 17500 data test dan 98,9% pada 19361 data training. Penelitian ini menggunakan metode Jaringan saraf tiruan dan Fuzzy Function, Jaringan Saraf Tiruan digunakan untuk mengklasifikan aksara dan Fuzzy Function digunakan untuk menyesuaikan sistem dengan kebiasaan individu menulis aksara. Pada penelitian ini error disebabkan oleh tulisan yang tidak normal, hubungan posisi tulisan yang ambigu, karakter yang tidak dikenal dan penghubung yang terlalu banyak. Peneliti menggunakan citra dari pena elektronik yang digambarkan pada sebuah tablet dan pada proses preprocessing peneliti menghilangkan noise dan hook pada aksara. Penelitian tentang pengenalan aksara Hangul lainnya dengan menggunakan metode Jaringan Saraf Tiruan Propagasi Balik yang dilakukan oleh Fathia (tanpa tahun), juga menghasilkan tingkat keakuratan yang tinggi, yaitu mencapai 99.7217% dan tingkat error sebesar 0.2783%. Penelitian lainnya yang menggunakan Jaringan Saraf Tiruan Propagasi Balik adalah penelitian yang dilakukan oleh Nugroho (2016), penelitian ini menggunakan Jaringan Saraf Tiruan Propagasi Balik untuk mengenali pola huruf Jepang Hiragana. Dan akurasi tertinggi yang didapat adalah 86,63% dengan menggunakan 2 lapisan tersembunyi.

Jaringan Saraf Tiruan adalah Jaringan Syaraf Tiruan (*Artificial Neural Network* (ANN)), atau juga disebut *Simulated Neural Network* (SNN), atau umumnya hanya disebut *Neural Network*(NN)), adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. Jaringan syaraf tiruan merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam

mengorganisasikan sel-sel penyusunnya yang disebut neuron, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi. Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan terdapat hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima, melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut.

Sedangkan jaringan saraf propagasi balik merupakan jaringan saraf yang menggunakan konsep jaringan berlapis jamak. Lapisan pertama adalah lapisan masukan (input) dan yang terakhir adalah lapisan keluaran (output). Lapisan diantara lapisan masukan dan lapisan keluaran disebut dengan lapisan tersembunyi (hidden).

Dengan adanya berbagai penelitian tentang pengenalan aksara dan penjelasan diatas, metode Jaringan Saraf Tiruan Propagasi Balik relevan dengan topik yang saya ambil. Karena dengan metode ini dapat mengenali aksara dengan baik dan hasil keakuratannya pun tinggi.

1.2 Rumusan Masalah

Berapa tingkat keakurasian yang dihasilkan dengan menggunakan metode Jaringan Saraf Tiruan Propagasi Balik ?

1.3 Tujuan Penelitian

Mengetahui performa alat uji dalam menterjemahkan aksara Hangul ke dalam huruf Latin berdasarkan tingkat keakurasian yang dihasilkan.

1.4 Batasan Masalah

1. Citra aksara yang digunakan dalam posisi normal.
2. Citra aksara yang digunakan adalah citra tulisan tangan aksara Hangul.
3. Aksara Hangul yang akan diuji adalah 14 huruf konsonan dasar.
4. Citra yang dipakai adalah citra yang dengan menggunakan pena.
5. Format citra yang akan digunakan adalah dalam bentuk jpg.

6. Citra digital diambil dengan cara di scan dengan melalui scanner.

1.5 Manfaat Penelitian

1. Membantu pengguna dalam mempelajari bahasa korea, khususnya dalam mempelajari huruf Hangul.
2. Memudahkan pengguna untuk membaca huruf Hangul dengan menterjemahkannya ke huruf latin.

1.6 Metode Penelitian

Metode penelitian yang digunakan adalah sebagai berikut:

1. Studi Pustaka

Tahap Studi adalah dasar dari penelitian ini, karena dengan adanya tahapan studi ini mahasiswa dibimbing dan dibekali dengan berbagai ilmu pengetahuan. Sehingga dalam proses studi ini mahasiswa diharapkan menyerap ilmu pengetahuan yang telah diberikan sebanyak-banyaknya dan kemudian mengimplementasikannya ke dalam tugas akhir yang akan diambil nantinya.

2. Pengumpulan Data

Pengumpulan data dalam penelitian ini diambil melalui form, form ini nantinya akan di scan untuk mendapatkan data digital. Citra yang digunakan adalah citra yang berbentuk tulisan tangan huruf Hangul. Kemudian citra yang sudah didapatkan dan diubah formatnya ke dalam bentuk jpg.

3. Pembuatan Alat Uji

Menyusun algoritma dari metode ekstraksi ciri dan metode Jaringan Saraf Tiruan Propagasi Balik untuk merancang alat uji kemudian mengimplementasikan algoritma tersebut ke dalam bahasa pemrograman Matlab 2014.

4. Analisa Pembuatan Alat Uji

Analisa pembuatan alat uji ini dilakukan untuk menguji tingkat keakurasian alat uji yang telah dibuat. Pengujian dilakukan dengan menggunakan bahasa pemograman Matlab 2014.

5. Pembuatan Laporan

Pada tahap ini peneliti menulis semua kegiatan, algoritma, metode dan langkah-langkah pembuatan alat uji dll selama dalam penelitian. Pembuatan laporan ini bertujuan agar semua kegiatan dalam penelitian dapat didokumentasikan dengan baik dan rapi.

1.7. Sistematika Penulisan Tugas Akhir

BAB I Pendahuluan

Pada bab ini membahas latar belakang pemilihan judul tugas akhir, perumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian dan sistematika penulisan proposal pada tugas akhir ini.

BAB II Landasan Teori

Bab ini berisi teori tentang huruf Korea atau Hangul, Pengenalan Pola, Ekstrasi Fitur *Zoning* , Ekstrasi ciri *Mark of Direction*, Ekstrasi ciri *Intensity of Character* dan juga teori tentang Jaringan saraf Tiruan Propagasi Balik.

BAB III Metode Penelitian

Bab ini akan membahas bagaimana citra gambar huruf Korea didapatkan atau dengan cara apa citra didapatkan, metode yang digunakan untuk mengolah citra yang sudah siap diuji, langkah-langkah pembuatan alat uji, cara mengolah data, cara pengujian dengan alat uji dan gambaran rancangan alat yang akan dibuat.

BAB IV Hasil Uji dan Analisis

Bab ini akan berisi hasil dari pengujian alat uji yang telah dibuat dan analisis dari seluruh pengujian.

BAB V Penutup

Bab ini berisi penutup yang didalamnya memuat kesimpulan dari hasil alat uji yang telah dibuat, kesimpulan dari seluruh pengujian yang telah dilakukan dan saran untuk peneliti yang akan melanjutkan penelitian ini.

BAB II

LANDASAN TEORI

2.1 Pengenalan Pola

Pengenalan pola adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif ciri atau sifat utama dari suatu obyek. Sedangkan pola adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Pengenalan pola dapat dilakukan sebagai tindakan untuk mengolah data mentah dan membuat suatu aksi berdasar dari kategori dari pola data tersebut.

Pada dasarnya pengenalan pola terdiri dari 3 langkah utama yaitu pemrosesan awal, ekstraksi fitur dan klasifikasi. Pemrosesan awal merupakan langkah untuk memfokuskan obyek data yang akan dikenali dengan obyek lain yang tidak digunakan. Dalam hal ini pemrosesan awal yang dilakukan terhadap obyek adalah dengan mengubah citra digital menjadi citra biner. Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih (Sinaga, 2013).

2.2 Huruf Hangul

Hangul adalah sistem alfabet bahasa Korea. Hangul dibuat pada jaman Chosun tahun 1443 oleh raja Sejong dan beberapa ilmuwan. Sebelum menggunakan Hangul, masyarakat Korea menggunakan huruf karakter Cina dalam kehidupan sehari-hari. Namun karena huruf karakter Cina sangat banyak jumlahnya, rumit, serta susah dipelajari, seringkali masyarakat Korea kesulitan dalam menggunakannya. Raja Sejong yang mengasihani rakyatnya membuat sistem alfabet Korea yang sesuai untuk melambangkan bunyi bahasa Korea serta mudah untuk dipelajari. Sistem alfabet ini disebut dengan “Hangul”. Saat Hangul diciptakan, Hangul disebut dengan

‘훈민정음(Hunminjeongeum)’ yang berarti “Bunyi Tepat untuk Mengajari Rakyat”.

Hangul terbagi sesuai dengan bunyinya, yakni vokal dan konsonan. Saat pertama kali dibuat Hangul terdiri atas 17 buah konsonan dan 11 buah vokal. Namun, pada perkembangannya, Hangul modern terdiri dari 40 buah alfabet, yakni 19 buah konsonan dan 21 vokal. Alfabet konsonan dibentuk berdasarkan teori bentuk dan jumlah coretan. Maksudnya, diantara konsonan terdapat konsonan dasar (ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ) yang merujuk pada bentuk titik artikulasi, serta 12 konsonan lainnya yang dibentuk dengan menambahkan coretan pada konsonan dasar berdasarkan keras tidaknya bunyi konsonan tersebut. Huruf vokal dibentuk dengan teori “천지인의 삼재(tiga struktur)”. Berdasarkan pada ㅇ, —, |, sedangkan vokal-vokal lain dibentuk dengan menggabungkan ketiga struktur diatas. Bentuk alfabet vokal ㅇ merujuk pada bentuk langit yang melingkar, vokal — merujuk pada bentuk tanah yang datar, dan vokal | merujuk pada bentuk orang yang sedang berdiri (Hwa, dkk, 2008).

Vokal Dasar

ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ

Vokal Perluasan

ㅐ ㅒ ㅖ ㅘ ㅙ ㅚ ㅜㅣ ㅝ ㅞ ㅟ ㅠ

Konsonan Dasar

ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅈ ㅇ ㅊ ㅋ ㅌ ㅍ ㅎ

Konsonan Ganda

ㄲ ㄴㄴ ㅃ ㅆ ㅈㅈ

Gambar 2. 1 Huruf Hangul

Huruf vokal pada huruf Hangul terdiri dari 21 huruf vokal, 10 buah diantaranya adalah vokal dasar dan 11 buah adalah vokal perluasan bentukan dari bentuk vokal dasar. Vokal ditulis dengan urutan dari atas kebawah, kiri kekanan. Huruf vokal dasar dari huruf Hangul adalah ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ sedangkan huruf vokal perluasan adalah ㅐ, ㅒ, ㅖ, ㅘ, ㅙ, ㅚ, ㅜㅣ, ㅝ, ㅞ, ㅟ, ㅠ.

Huruf konsonan pada huruf Hangul terdiri dari 19 buah konsonan, 14 diantaranya adalah konsonan dasar dan 5 lainnya adalah huruf konsonan ganda. Konsonan harus bertemu dengan vokal untuk membentuk suku kata agar dapat dilafalkan, konsonan tidak dapat dibunyikan bila berdiri sendiri. Konsonan dapat dilafalkan dengan satu atau lebih cara sesuai dengan vokal yang mengikutinya dan posisinya dalam suku kata. Huruf konsonan dasar dari huruf Hangul adalah ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅈ, ㅇ, ㅊ, ㅋ, ㅌ, ㅍ, ㅎ, sedangkan konsonan ganda dari huruf Hangul adalah ㄲ, ㄴㄴ, ㅃ, ㅆ, ㅈㅈ (Hwa, 2008).

2.3 Pemrosesan Citra Digital

Sebuah citra dapat didefinisikan sebagai fungsi dua dimensi, $f(x, y)$, x dan y merupakan koordinat spatial dan amplitudo dari f dengan pasangan koordinat (x, y) disebut intensitas citra atau gray level pada titik tersebut. Ketika x , y dan nilai amplitudo dari f adalah terbatas (finite), bernilai diskrit maka suatu citra dapat disebut sebagai citra digital (digital image). Citra digital ini terbentuk dari angka yang terbatas dan dapat diukur (finite) dari setiap elemennya yang memiliki lokasi dan nilai tertentu. Elemen-elemen tersebut sering kita sebut dengan picture elements, image elements, pels, dan pixels. Pemrosesan yang terkait dengan citra digital ini dapat didefinisikan sebagai suatu proses yang memiliki input dan output berupa image(citra) dan sebagai tambahan meliputi juga proses yang mengekstraksi atribut dari citra hingga sampai pada pengenalan masing-masing objek. (Gonzales dan Woods, 2008).

2.4 Ekstraksi Fitur (Feature Extraction)

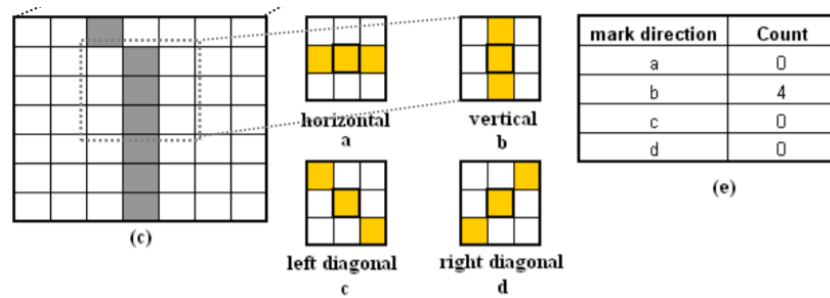
Ekstraksi fitur dilakukan untuk menyederhanakan citra dengan melakukan pengukuran fitur tertentu sehingga kita dapat mendapat atau melihat informasi dari citra tersebut (Sinaga, 2013). Pada hal ini, ekstraksi fitur yang saya gunakan adalah *Intensity Of Character* dan *Mark Direction*.

2.4.1 Intensity of Character

Intesitas karakter adalah ekstraksi ciri dengan cara menghitung jumlah piksel yang berwarna hitam.

2.4.2 Mark Direction

Tanda arah adalah ekstraksi ciri dengan cara menghitung banyaknya piksel yang memiliki tetangga yang berarah horizontal, vertical, diagonal ke kanan, dandiagonal ke kiri.



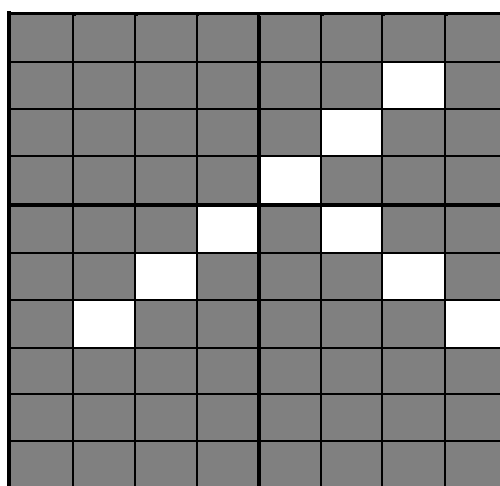
Gambar 2. 2 Struktur Mark Direction (Surinta, 2010)

2.4.3 Zonning

Zonning merupakan salah satu metode ekstrasi ciri pada citra karakter. Secara umum, metode *zoning*, akan membagi citra menjadi beberapa zona dengan ukuran yang sama. Kemudian, setiap zona akan diambil cirinya.

Metode zoning memiliki 3 variasi algoritma, yaitu *Image Centroid and Zone* (ICZ), *Zone Centroid and Zone* (ZCZ), dan gabungan ICZ dan ZCZ. Metode ekstrasi ciri *zoning* yang digunakan pada penelitian adalah ZCZ.

Sebagai ilustrasi, berikut akan diberikan contoh ekstrasi ciri menggunakan metode ekstrasi ciri *Zone Centroid and Zone*. Sebagai contoh digunakan sebuah citra berukuran 8x8, seperti berikut :



Gambar 2. 3 Pembagian Zona dan Perhitungan Jarak ZCZ

Dengan menggunakan algoritma *Zone Centroid and Zone*, pertama citra akan dibagi menjadi n zona yang sama, misalkan dibagi menjadi 4 zona, dengan masing-masing zona berukuran 4×4 pixel. Kemudian, menghitung *zone centroid* dari masing-masing zona dengan menggunakan persamaan :

$$x_c = \frac{\sum_{i=1}^x f(x,y)x_i}{\sum f(x,y)} \quad \text{dan} \quad y_c = \frac{\sum_{i=1}^y f(x,y)y_i}{\sum f(x,y)}$$

Dengan $f(x, y)$ adalah nilai pixel dari citra pada posisi tertentu. Dari Hasil Perhitungan, didapatkan hasil berikut :

Zona 1 (atas kiri) : $x_c=0, y_c=0$

Zona 2 (atas kanan) : $x_c=2, y_c=3$

Zona 3 (bawah kiri) : $x_c=3, y_c=2$

Zona 4 (bawah kanan): $x_c=3, y_c=2$

Gambar 2.3 merupakan representasi zona menggunakan ZCZ yang dibagi menjadi 3 zona sama besar pada citra aksara S. Perhitungan fitur pada zona 2 adalah sebagai berikut:

$$d_1 = \sqrt{(3-2)^2 + (2-3)^2} = 1,41$$

$$d_2 = \sqrt{(2-2)^2 + (3-3)^2} = 0$$

$$d_3 = \sqrt{(1-2)^2 + (4-3)^2} = 1,41$$

$$fitur_2 = \frac{1,41+0+1,41}{3} = 0,94$$

Berdasarkan hasil perhitungan diatas maka ciri atau fitur dari zona ke 1 adalah 0 dan nilai fitur dari zona ke 2 adalah 0,94. Cara yang sama juga berlaku untuk mencari nilai ciri dari zona ke 3 dan zona ke 4. Nilai fitur dari zona ke 3 adalah 0,94 dan nilai fitur dari zona ke 4 adalah 0,94. Setelah didapatkan masing-masing nilai fitur dari setiap zona, lalu fitur tersebut digabungkan menjadi 4 fitur yang merepresentasikan citra tersebut sehingga dapat ditulis $[0 \ 0,94 \ 0,94 \ 0,94]$;

2.5 Jaringan Saraf Tiruan

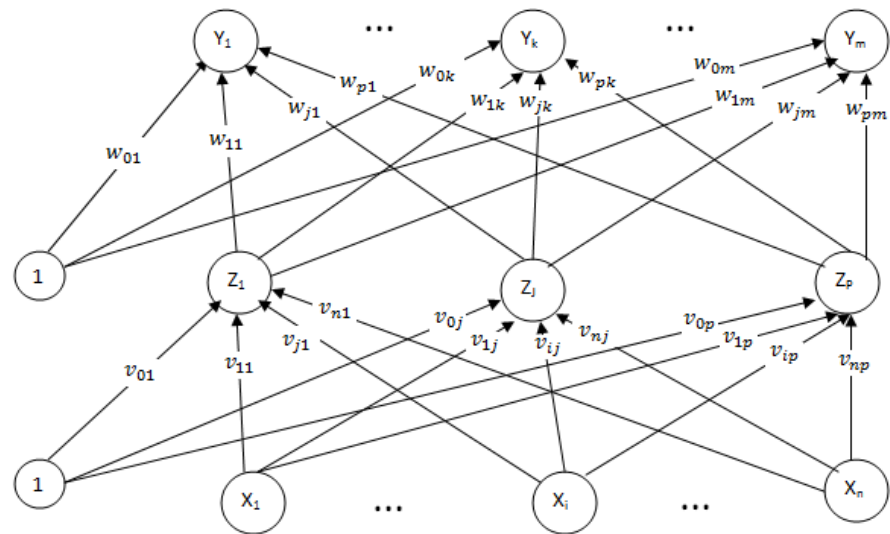
Jaringan saraf tiruan (artificial neural networks) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologi dalam otak. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi nonlinear, klasifikasi data,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

Model saraf ditunjukan dengan kemampuannya dalam emulasi, 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 (Andri, 2004).

2.6 Jaringan Saraf Tiruan Propagasi Balik (*Backpropagation*)

2.6.1 Pengertian Jaringan Saraf Tiruan Propagasi Balik

Pelatihan sebuah jaringan yang menggunakan *backpropagation* terdiri dari 3 langkah, yaitu: pelatihan pola input secara *feedforward*, perhitungan dan *backpropagation* dari kumpulam kesalahan dan penyesuaian bobot. Sesudah pelatihan, aplikasi dalam jaringan dari fase *feedforward*. Bahkan jika pelatihan menjadi lambat, sebuah jaringan yang dilatih dapat menghasilkan outputnya sendiri secara cepat. Banyak variasi dari *backpropagation* yang dapat dibangun untuk meningkatkan kecepatan proses pelatihan (Kristanto, 2004).



Gambar 2. 4 Jaringan Saraf Tiruan Backpropagation dengan 1 Lapisan Tersembunyi (Kristanto, 2004)

Jaringan saraf banyak lapisan dengan satu lapisan dari unit yang tersembunyi dapat dilihat pada gambar diatas. Unit output (unit Y) dan unit yang tersembunyi juga mempunyai bias. Bias pada unit Y_k disimbolkan dengan w_{0k} dan bias pada unit tersembunyi Z_j disimbolkan dengan v_{0j} . Bias tersebut selalu bernilai 1. Dari gambar tersebut, dapat dilihat hanya aliran informasi langsung untuk fase forward pada operasi yang ditunjukkan (Kristanto, 2004).

2.6.2 Algoritma Jaringan Saraf Tiruan Propagasi Balik

Selama kondisi feedforward, masing-masing unit input input (X_i) menerima sebuah sinyal input dan mengirimkan sinyal tersebut ke masing-masing unit tersembunyi Z_1, \dots, Z_p . Masing-masing unit tersembunyi kemudian menghitung aktivasi dan mengirimkan sinyalnya (Z_j) ke masing-masing unit output. Masing-masing unit output (Y_k) kemudian menghitung aktivasinya (y_k) untuk membentuk respon dalam jaringan yang diberi pola input.

Selama pelatihan, masing-masing unit output membandingkan aktivasi (y_k) dengan nilai target (t_k) kemudian menentukan kumpulan kesalahan untuk pola yang ada pada unit. Berdasarkan kesalahan tersebut, nilai faktor δ_k ($k=1, \dots, m$) dihitung kemudian δ_k digunakan untuk mendistribusikan kesalahan pada unit output (Y_k) kembali ke semua unit pada lapisan sebelumnya. (unit tersembunyi dihubungkan dengan (Y_k)). Kondisi semacam ini nanti akan digunakan untuk memperbaiki bobot diantara output dan lapisan tersembunyi. Untuk hal yang sama, nilai faktor δ_j ($j=1, \dots, p$) dihitung untuk masing-masing unit yang tersembunyi (Z_j). Hal ini tidak diperlukan untuk menyebarkan kesalahan ke lapisan input, tetapi δ_j digunakan untuk memperbaiki bobot diantara lapisan yang tersembunyi dan dan lapisan input.

Sesudah δ ditentukan, bobot untuk semua lapisan disesuaikan secara simultan. Penyesuaian bobot w_{jk} (dari unit tersembunyi Z_j ke unit output Y_k) didasarkan pada faktor δ_k dan aktivasi Z_j pada unit tersembunyi Z_j . Penyesuaian bobot v_{jk} (dari unit input X_i ke unit tersembunyi Z_j) didasarkan pada faktor δ_j dan aktivasi x_i pada unit input (Kristanto, 2004).

2.6.3 Simbol-simbol yang digunakan dalam Jaringan Saraf Tiruan

Propagasi Balik

Simbol-simbol yang digunakan pada algoritma pelatihan untuk jaringan backpropagation adalah sebagai berikut :

x	input vektor pelatihan $x=(x_1, \dots, x_i, \dots, x_n)$
t	output vektor target $t=(t_1, \dots, t_k, \dots, t_m)$
δ_k	informasi tentang kesalahan pada unit Y_k yang disebarkan kembali ke unit tersembunyi
δ_j	informasi tentang kesalahan lapisan output ke unit tersembunyi Z_j
α	Learning rate
X_i	Unit input 1

Untuk sebuah unit input, sinyal input dan sinyal output adalah sama yaitu x .

v_{oj} Bias pada unit tersembunyi j

Z_j unit tersembunyi j Input jaringan ke Z_j disimbolkan dengan

$$z_in_j = v_{oj} + \sum_i x_i v_{ij}$$

Sinyal output(aktivasi) pada Z_j disimbolkan dengan z_j

$$Z_j = f(z_in_j)$$

v_{ok} Bias pada unit tersembunyi k

Y_k unit tersembunyi l Input jaringan ke Y_k disimbolkan dengan

$$y_in_k = w_{ok} + \sum_i z_i w_{ik}$$

Sinyal output(aktivasi) pada Y_k disimbolkan dengan y_k

$$y_k = f(y_in_k)$$

2.6.4 Fungsi Aktivasi

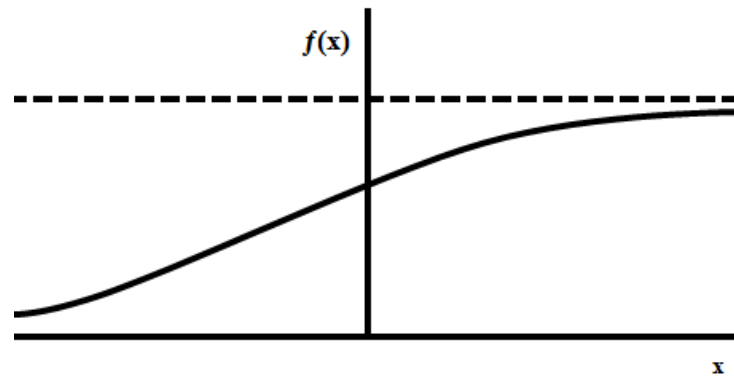
Fungsi aktivasi untuk sebuah jaringan backpropagation seharusnya mempunyai beberapa karakter yang penting, berlanjut dan berbeda. Salah satu fungsi aktivasi yang biasa digunakan pada jaringan ini adalah fungsi binary sigmoid dimana mempunyai jangkauan 0 dan 1 dan didefinisikan dengan rumus :

$$f_1(x) = \frac{1}{1 + e^{(-x)}}$$

Dengan

$$f_1(x) = f_1(x) [1 - f_1(x)]$$

Fungsi tersebut dijelaskan pada gambar berikut ini :



Gambar 2. 5 Binary sigmoid dengan jangkauan 0 dan 1

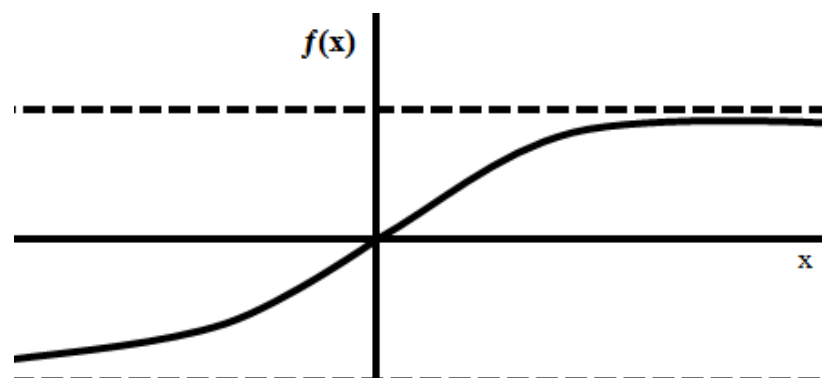
Untuk fungsi aktivasi lain adalah bipolar sigmoid yang mempunyai jangkauan $(-1, 1)$ didefinisikan dengan rumus :

$$f_2(x) = \frac{2}{1 + e^{x}} - 1$$

Dengan

$$f_2(x) = 1/2 [1 + f_1(x)] [1 - f_1(x)]$$

Fungsi bipolar sigmoid dapat dilihat pada gambar berikut :



Gambar 2. 6 bipolar sigmoid dengan jangkauan -1 dan 1

Algoritma Pelatihan

Algoritma dari jaringan backpropagation adalah sebagai berikut :

Langkah 1 inisialisasikan bobot(set ke nilai kecil secara acak)

Langkah 2 selama kondisi berhenti bernilai salah, kerjakan:

a. Untuk masing-masing pasangan pelatihan, lakukan :
Feedforward

1. masing-masing unit input (X_i , $i=1, \dots, n$) menerima sinyal input x_i dan menyebarkan sinyal ini ke semua unit lapisan atas (unit tersembunyi)
2. masing-masing unit tersembunyi (Z_j , $j=1, \dots, p$) menjumlahkan bobot sinyal input

$$z_in_j = v_{oj} + \sum_i x_i v_{ij}$$

Dan mengaplikasikan fungsi aplikasi untuk menghitung sinyal output

$$Z_j = f(z_in_j)$$

Dan mengirimkan sinyal tersebut ke semua unit pada lapisan atas (unit output)

3. Masing-masing unit output (Y_k , $k=1, \dots, m$) menjumlahkan bobot sinyal input

$$y_in_k = w_{ok} + \sum_i z_i w_{ik}$$

Dan mengaplikasikan fungsi aplikasi untuk menghitung sinyal output

$$y_k = f(y_in_k)$$

b. Untuk masing-masing pasangan pelatihan, lakukan :
Backpropagation

1. Masing-masing unit output (Y_k , $k=1, \dots, m$) menerima sebuah pola target yang bersesuaian dengan pola input pelatihan, menghitung informasi kesalahan,

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

kemudian menghitung koreksi bobot (digunakan untuk memperbaiki w_{jk})

$$\Delta w_{jk} = \alpha \delta_k z_j$$

Dan akhirnya menghitung koreksi bias (digunakan untuk memperbaiki w_{0k})

$$\Delta w_{0k} = \alpha \delta_k$$

Setelah itu mengirimkan δ_k ke unit dalam lapisan yang paling atas

2. Masing-masing unit yang tersembunyi (Z_j , $j=1, \dots, p$) menjumlahkan input delta (dari unit lapisan atas)

$$\delta_{in_i} = \sum_{k=1}^m \delta_k w_{ik}$$

Kalikan nilai ini dengan fungsi aktivasi untuk menghitung informasi kesalahan

$$\delta_i = \delta_{in_i} f' (z_{in_i})$$

Kemudian hitunglah koreksi bobot (digunakan untuk memperbaiki v_{ij})

$$\Delta v_{ij} = \alpha \delta_i x_i$$

Setelah itu hitung koreksi bias (digunakan untuk memperbaiki v_{0j})

$$\Delta v_{0i} = \alpha \delta_i$$

- c. Perbaiki bobot bias

Masing-masing unit output output (Y_k , $k=1, \dots, m$) memperbaiki bobot dan bias ($j=0, \dots, p$)

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

Masing-masing unit tersembunyi (Z_j , $j = 1, \dots, p$)
memperbaiki bobot dan bias ($i=0, \dots, n$)

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

d. Tes kondisi berhenti

2.7 Mean Square Error

Mean Square Error (MSE) dapat dihitung dengan menjumlahkan semua kesalahan peramalan pada setiap periode kemudian dikuadratkan dan membaginya dengan jumlah periode peramalan. Secara matematis, MSE dirumuskan sebagai berikut (Nasution dan Prasetyawan, 2008):

$$MSE = \frac{\sum_{i=1}^n (X_t - F_t)^2}{n}$$

Dimana :

X_t = Nilai riil pada periode ke t

F_t = Nilai peramalan pada periode ke t

n = Jumlah periode peramalan

BAB III

METODE PENELITIAN DAN PERANCANGAN

PROGRAM

3.1 Bahan

Data yang digunakan dalam tugas akhir ini adalah citra tulisan tangan huruf Hangul. Dari 40 Huruf Hangul, citra tulisan yang dipakai adalah 14 konsonan dasar huruf Hangul, yaitu ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅊ, ㅋ, ㅌ, ㅍ, ㅎ.

3.2 Peralatan Penelitian

3.2.1 Hardware

Hardware atau perangkat keras yang digunakan dalam penelitian ini adalah :

1. *Processor* : Intel(R) Core(TM) i5-2450M CPU
@2.50GHz 2.50GHz
2. *RAM* : 4.00 GB
3. *Hard Drive* : 500 GB
4. *Graphic Interface* : NVIDIA GEFORCE 525M
5. *Scanner* : Canon Lide 120

3.2.2 Software

1. Microsoft Windows 10
2. Aplikasi Paint
3. Matlab R2017a

Sistem operasi Microsoft Windows 10 digunakan agar Matlab R2017a dan Paint dapat berjalan

3.3 Tahap-tahap penelitian

3.3.1 Pengumpulan data dan kebutuhan

- a. Membuat form untuk responden menulis data tulisan tangan huruf Hangul

ㄱ		
ㄴ		
ㄷ		

Gambar 3. 1 Form Pengenalan Tulisan Tangan Huruf Hangul

Data 14 konsonan tulisan tangan huruf Hangul didapatkan dari responden yang telah mengisi form dan responden menulis huruf sebanyak dua kali.

- b. Pengumpulan data

Data didapatkan melalui form yang di bagikan kepada mahasiswa semester 5 jurusan Bahasa Korea, Universitas Gadjah Mada. Pada form yang dibagikan terdapat 2 kolom kosong disamping setiap 14 konsonan huruf Hangul. Mahasiswa diberi kesempatan untuk menulis tulisan tangan huruf Hangul sebanyak 2 x 14 data. Sehingga total yang diperoleh ada 25 x 2 x 14 data.

Form Tugas Akhir
"Pengenalan Tulisan Tangan Huruf Hangul Dengan Menggunakan Metode Jaringan Saraf Tiruan Propagasi Balik"

Huruf	Lembar Kerja Untuk Menulis Huruf Hangul	
	Lembar Kerja 1	Lembar Kerja 2
ㄱ	ㄱ	ㄱ
ㄴ	ㄴ	ㄴ
ㄷ	ㄷ	ㄷ

Gambar 3. 2 Contoh Form yang Telah Diisi

3.3.2 Pengolahan Data

Setelah mendapat data tulisan tangan Huruf Hangul, tahap selanjutnya adalah mengolah data, adapun tahap-tahap mengolah data antara lain :

a. *Men-scan* data

Setiap form yang didapatkan akan di scan melalui scanner, tujuan dari proses ini adalah mendapatkan data digital dari data tulisan tangan yang sebelumnya didapat pada form kuisioner. Setelah mendapat data tulisan tangan secara digital.

b. *Meng-crop* data digital

Kegiatan *meng-crop* data dilakukan dengan cara memotong form yang sudah di-*scan* untuk diambil data perhurufnya.



Gambar 3. 3 Contoh Gambar yang Sudah di-Crop

c. Pemrosesan Citra

Kegiatan pemrosesan citra dilakukan dengan tahap-tahap sebagai berikut :

1) Mengubah Citra Warna

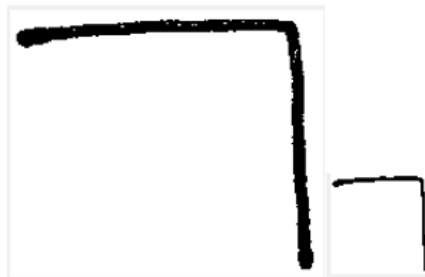
Kegiatan ini bertujuan untuk mengubah citra yang tadinya berwarna menjadi citra keabuan, setelah mendapat citra keabuan langkah selanjutnya adalah mengubah citra keabuan menjadi citra hitam putih.



Gambar 3. 4 Contoh Gambar Citra yang sudah diubah ke Citra Hitam Putih

2) *Resizing* Citra

Kegiatan ini dilakukan untuk mengubah ukuran dari citra, dari berbagai citra yang tidak sama ukurannya kemudian disamakan ukurannya menjadi 64 x 64 piksel. Hal ini dimaksudkan agar data yang diolah tidak terlalu besar dan dapat dibagi menjadi 4 segmen.



Gambar 3. 5 Conth Citra yang sudah di Resize

3) *Thinning* Citra

Kegiatan ini bertujuan untuk mengubah ketebalan citra menjadi 1 piksel



Gambar 3. 6 Contoh Citra yang sudah ditipiskan

4) Membagi Citra

Kegiatan ini bertujuan untuk membagi citra menjadi beberapa segmen yang lebih kecil. Tujuan dari kegiatan ini adalah lebih mempermudah dalam pencirian citra. Pada penelitian ini citra akan dibagi menjadi 4 segmen.

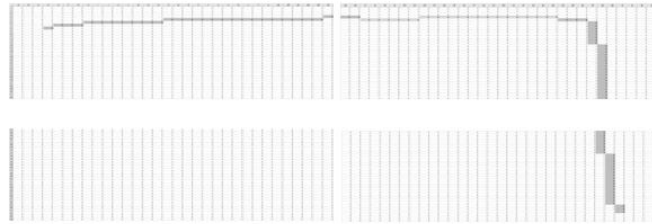


Gambar 3. 7 Contoh Gambar yang sudah dibagi Menjadi 4 Segmen

5) Mengambil Ciri Citra

Pada penelitian ini pengambila ciri citra menggunakan ekstrasi ciri *Intensity of Character* dan *Mark Direction*. *Intensity of Character* digunakan untuk mendapatkan jumlah piksel hitam dalam sebuah citra. Sedangkan *Mark Direction* digunakan untuk menghitung banyaknya piksel yang memiliki tetangga yang berarah horizontal, vertikal,

diagonal ke kanan, dandiagonal ke kiri. Pada penelitian ini, *Intensity of Character* dan *Mark Direction* digunakan pada setiap segmen dari citra yang telah dibagi menjadi 4 segmen.



Gambar 3. 8 Contoh Citra yang sudah diekstrak Cirinya

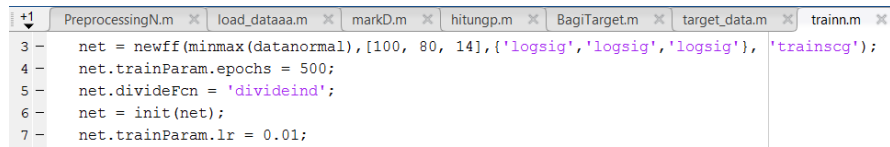
Tabel 3. 1 Tabel Contoh Hasil Ekstrasi Ciri

Segmen ke	IoC	Mark Direction	Mark Direction	Mark Direction	Mark Direction
		Horizontal	Vertikal	Diagonal 1	Diagonal 2
1	929	29	3	0	6
2	895	26	54	10	2
3	1024	0	0	0	0
4	951	3	54	0	0

- d. Mempersiapkan dan membuat alat uji algoritma *Backpropagation*

Langkah pertama dalam kegiatan ini adalah mengambil data ciri menggunakan script `load_dataaaa.m` dan menormalisasi data tersebut. Kemudian membuat target data menggunakan script `target_data.m` dan dimasukkan ke variabel `label`. Kemudian mengubah vector target menjadi sebuah indeks dengan menggunakan script `Bagi_Target.m` dan dimasukkan ke variabel `l1` sampai `l14`.

e. Pelatihan Alat



```

1 - % PreprocessingN.m
2 - % load_dataaaa.m
3 - % markD.m
4 - % hitungp.m
5 - % BagiTarget.m
6 - % target_data.m
7 - % trainn.m

3 - net = newff(minmax(dataanormal),[100, 80, 14],{'logsig','logsig','logsig'}, 'trainscg');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;

```

Gambar 3. 9 Contoh Program Jaringan Propagasi Balik

Pada penelitian ini ada 43 data untuk setiap hurufnya dan ada 14 huruf dan total semua huruf ada 602 huruf. Untuk pelatihan alat digunakan data sebanyak 2/3 bagian untuk setiap hurufnya. Jadi total data yang digunakan dalam pelatihan alat 28 data huruf. Dikarenakan ada 602 huruf, maka total data dari semua huruf yang digunakan untuk data pelatihan sebanyak 392 huruf. Fungsi aktivasi yang digunakan dalam penelitian ini adalah logsig. Jumlah neuron yang dipakai ada beberapa, mulai dari 50 unit hingga 100 unit. Target yang telah disiapkan sebanyak 14 target karena terdapat 14 huruf Hangul yang berupa matriks.

Percobaan pelatihan alat dalam penelitian ini dilakukan sebanyak 12 kali. Pada setiap percobaan telah dilakukan perubahan jumlah layer, jumlah neuron, atau fungsi aktivasi yang digunakan pada setiap layer untuk melihat peningkatan akurasi.

f. Pengujian Alat

Pengujian alat dilakukan dengan menggunakan 6 kondisi ciri yang berbeda. Setiap percobaan dengan kondisi ciri yang berbeda, jumlah unit pada setiap hidden layer, atau jumlah iterasi yang dipakai telah dirubah dan terlihat ada peningkatan dan penurunan akurasi.

3.3.4 Analisis data

Pengujian dilakukan untuk menguji apakah pengenalan pola yang dilakukan sudah sesuai atau belum. Pengujian dalam penelitian ini adalah dengan menghitung keakurasian sistem yaitu dengan menghitung hasil jumlah data yang sesuai/benar dibagi dengan jumlah seluruh data. Pengujian dilakukan dengan rumus :

$$\text{Akurasi (\%)} = \frac{\text{Jumlah data benar}}{\text{Jumlah semua data}} \times 100\%$$

3.4 Gambaran Umum Sistem

Rancangan sistem Pengenalan Tulisan Tangan Huruf Hangul dimulai dari data input yang berupa data ciri dari huruf Hangul yang akan dipakai. Untuk pengolahan di dalam hidden layer dipakai fungsi aktivasi logsig atau tansig . output dari sistem ini ada 14 karena targetnya berupa matriks 14 x1 yang akan dicocokkan pada target masing-masing huruf. Rancangan ini sudah memiliki 3 komponen yaitu *input*, *hidden layer*, dan *output*.

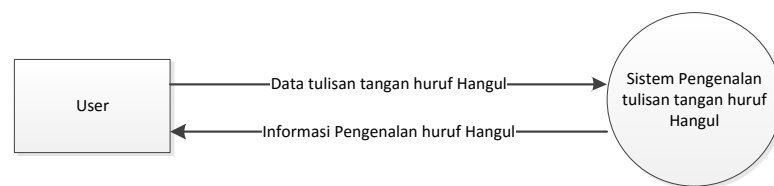
Sistem ini akan digunakan untuk mengenali 14 konsonan huruf Hangul dengan menggunakan jaringan saraf tiruan propagasi balik.

Citra tulisan tangan huruf Hangul akan disimpan dengan format .jpg. kemudian akan dilakukan *preprocessing* Citra yang sudah dipotong sesuai dengan ukurannya kemudian akan diubah menjadi citra keabuaan dan citra hitam putih. Setelah menjadi citra biner maka langkah selanjutnya adalah mengubah ukuran citra menjadi 64 x 64 piksel. Kemudian citra dengan ukuran 64 x 64 ini akan ditipiskan untuk mendapatkan ketebalan citra 1 piksel.

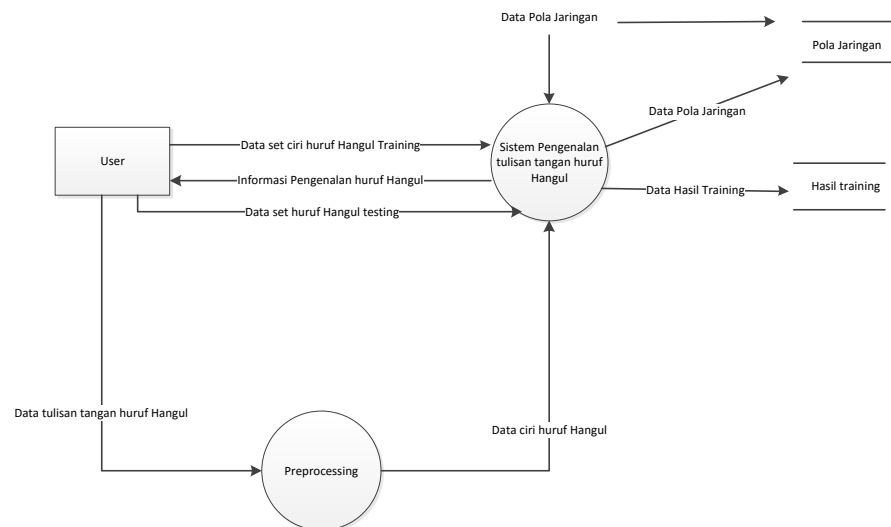
Citra yang sudah ditipiskan kemudian akan dibagi menjadi 4 segmen. Dan dari setiap segmen akan diambil cirinya menggunakan ekstrasi ciri *Intensity of Character* dan *Mark Direction*. Ciri dari setiap

segmen yang didapatkan akan dimasukkan kedalam sistem jaringan saraf tiruan. Citra tersebut akan dikenali jika memiliki kesamaan ciri dengan ciri dari data pelatihan yang dilatikan sebelumnya pada mesin *backpropagation*.

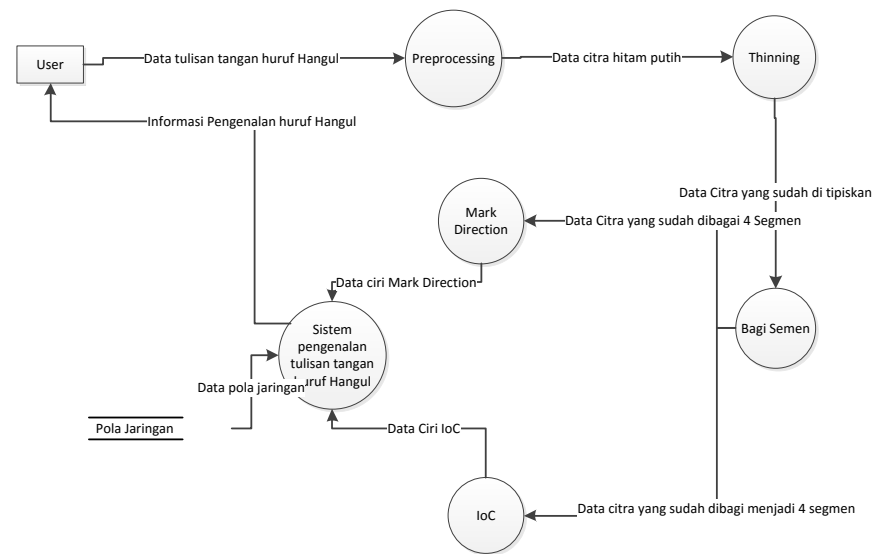
Penjelasan alur data dan kerja sistem yang telah dipaparkan diatas akan dijelaskan pada Data Flow Diagram berikut:



Gambar 3. 10 Data Flow Diagram level 0



Gambar 3. 11 Data Flow Diagram level 1



Gambar 3. 12 Data Flow Diagram level 2

3.5 Perancangan Program

Pada penelitian ini telah dibuat beberapa script yang digunakan untuk megolah citra, membagi citra menjadi 4 segemen, mengambil ciri *Intensity of Character*, mengambil ciri *Mark Direction*, mengambil data ciri, membuat targaet, membagi target, pola jaringan, dan *testing* dan akurasi. Adapun algoritma dari setiap script yang telah dibuat sebagai berikut:

1. Algoritma Preprocessing

- a. Mulai
- b. Membaca file gambar citra huruf Hangul dengan menggunakan `imread`
- c. Mengubah citra berwarna menjadi citra Grayscale dengan menggunakan fungsi `rgb2gray`
- d. Mengubah citra keabuan menjadi citra hitam putih dengan fungsi `im2bw`
- e. Mengubah ukuran citra menjadi 64x64 piksel dengan menggunakan fungsi `imresize`
- f. Menipiskan citra dengan menggunakan `Rosenfeld`

- g. Membagi huruf menjadi 4 segmen dengan ukuran 32x32 piksel
- h. Mengekstrak ciri dengan Intensity of Character pada setiap segmen untuk menghitung jumlah piksel hitam pada setiap segmen
- i. Memasukkan hasil ciri dari Intensity of Character kedalam array jh
- j. Mengekstrak ciri dengan Mark Direction pada setiap segmen untuk menghitung banyaknya piksel yang memiliki tetangga yang berarah horizontal, vertikal, diagonal ke kanan, dandiagonal ke kiri
- k. Memasukkan hasil ciri dari Mark Direction untuk arah horizontal kedalam array horz
- l. Memasukkan hasil ciri dari Mark Direction untuk arah vertikal kedalam array vert
- m. Memasukkan hasil ciri dari Mark Direction untuk arah diagonal 1 kedalam array dig1
- n. Memasukkan hasil ciri dari Mark Direction untuk arah diagonal 2 kedalam array dig2\Membuat kombinasi ciri 1 dengan memasukkan variabel jh ke array ciri1
- o. Membuat kombinasi ciri 2 dengan memasukkan variabel dig1 ke array ciri2
- p. Membuat kombinasi ciri 3 dengan memasukkan variabel jh dan dig1 ke array ciri3
- q. Membuat kombinasi ciri 4 dengan memasukkan variabel jh dan dig2 ke array ciri4
- r. Membuat kombinasi ciri 5 dengan memasukkan variabel dig1 dan dig2 ke array ciri5
- s. Membuat kombinasi ciri 6 dengan memasukkan variabel jh, dig1, dig2, horz, dan vert ke array ciri
- t. Selesai

2. Algoritma BagiZona

- a. Mulai
- b. Membuat array baris & kolom yang merupakan ukuran baru dari citra
- c. Membagi citra ke segmen pertama dari baris 1 sampai 32 dan kolom 1 sampai 32 lalu disimpan ke z1
- d. Membagi citra ke segmen kedua dari baris 1 sampai 32 dan kolom 33 sampai 64 lalu disimpan ke z2
- e. Membagi citra ke segmen ketiga dari baris 33 sampai 64 dan kolom 1 sampai 32 lalu disimpan ke z3
- f. Membagi citra ke segmen keempat dari baris 33 sampai 64 dan kolom 33 sampai 64 lalu disimpan ke z4
- g. Menyediakan keluaran berupa citra baru z1, z2, z3, dan z4
- h. Selesai

3. Algoritma hitungp

- a. Mulai
- b. Mendapatkan jumlah baris dari matriks citra
- c. Mendapatkan jumlah kolom dari matriks citra
- d. Membuat jh dan menginisialisasi jh dengan 0
- e. Untuk i=1 sampai kolom lakukan langkah h, jika sudah memenuhi sampai kolom lakukan langkah h
- f. Untuk j=1 sampai baris lakukan langkah g, jika sudah memenuhi sampai baris lakukan langkah d
- g. Jika new(i,j) sama dengan 1 tambahkan nilai jh dengan 1
- h. Menyediakan keluaran nilai jh
- i. Selesai

4. Algoritma markD

- a. Mulai
- b. Mendapatkan jumlah baris dan kolom dari matriks citra
- c. Membuat vert dan menginisialisasi vert dengan 0
- d. Membuat horz dan menginisialisasi horz dengan 0
- e. Membuat dig1 dan menginisialisasi dig1 dengan 0
- f. Membuat dig2 dan menginisialisasi dig2 dengan 0
- g. Untuk $i=1$ sampai kolom lakukan langkah h, jika sudah mmenuhi lakukan sampai kolom lakukan langkah selesai
- h. Jika $\text{new}(i,j)$ sama dengan 0 lakukan langkah i,k,m,o
- i. Jika $\text{new}(i,j) + \text{new}(i-1,j) + \text{new}(i+1,j)$ sama dengan 0 lakukan langkah j
- j. Jika $\text{new}(i-1,j-1) + \text{new}(i-1,j+1) + \text{new}(i,j-1) + \text{new}(i,j+1) + \text{new}(i+1, j) + \text{new}(i+1, j+1)$ sama dengan 6, maka tamabahkan $\text{vert} = \text{vert}+1$
- k. Jika $\text{new}(i,j) + \text{new}(i, j-1) + \text{new}(i,j+1)$ sama dengan 0 lakukan langkah l
- l. Jika $\text{new}(i-1,j-1) + \text{new}(i-1,j) + \text{new}(i-1,j+1) + \text{new}(i+1,j-1) + \text{new}(i+1, j) + \text{new}(i+1, j+1)$ sama dengan 6, maka tamabahkan $\text{horz} = \text{horz}+1$
- m. Jika $\text{new}(i,j) + \text{new}(i-1, j-1) + \text{new}(i+1,j+1)$ sama dengan 0 lakukan langkah m
- n. Jika $\text{new}(i-1,j) + \text{new}(i-1,j+1) + \text{new}(i,j-1) + \text{new}(i, j+1) + \text{new}(i+1, j-1) + \text{new}(i+1,j)$ sama dengan 6, maka tamabahkan $\text{dig1} = \text{dig1}+1$
- o. Jika $\text{new}(i-1,j-1) + \text{new}(i-1, j+1) + \text{new}(i+1,j-1)$ sama dengan 0 lakukan langkah p
- p. Jika $\text{new}(i-1,j-1) + \text{new}(i-1,j) + \text{new}(i,j-1) + \text{new}(i, j+1) + \text{new}(i+1, j) + \text{new}(i+1,j+1)$ sama dengan 6, maka tamabahkan $\text{dig2} = \text{dig2}+1$
- q. Selesai

5. Algoritma load_dataaa

- a. Mulai
- b. Membaca data ciri yang sudah siap dengan fungsi load
- c. Menetapkan kolom mana saja yang menjadi ciri
- d. Mernormalisasi data ciri dengan fungsi mapminmax
- e. Selesai

6. Algoritma target_data

- a. Mulai
- b. Membuat target 11 sampai dengan 114 dengan matriks 1x14 dan setiap matriksnya diulangi sebanyak 43 data sesuai dengan jumlah data perhurufnya dengan fungsi repmat dan kemudian ditranspose menjadi 14x1, jadi setiap 1 mempunyai matriks berukuran 14x43
- c. Memasukkan matriks 11 sampai 114 ke matriks label
- d. Selesai

7. Algoritma BagiTarget

- a. Mulai
- b. Mengambil banyak baris dan kolom pada target O (baris) dan N (kolom)
- c. Membuat variabel labelData dengan membaca variabel target dari script target data yang telah dikonversi menjadi indeks dengan menggunakan fungsi vec2ind
- d. Menginisialisasi id1 sampai id14 dengan nilai baru yaitu nilai indeks. Id1 =1, id2 =2, id3 =3, id4 =4, id5 =5, id6 =6, id7 =7, id8 =8, id9 =9, id10 =10, id11 =11, id12 =12, id13 =13, id14 =14. Hal tersebut didapatkan dengan menjalankan `id1=find(labelData==1)`
- e. Menginisialisasi variabel ind0 = 1:N

f. Selesai

8. Algoritma trainn

- a. Mulai
- b. Insialisasi variabel M dengan nilai 15 (banyaknya data testing)
- c. Menginisialisasi control random number dengan nilai 0
- d. Membuat struktur jaringan yang akan digunakan untuk perhitungan
- e. Menginisialisasi banyak iterasi yang akan dilakukan saat jaringan dilakukan dijalankan
- f. Menginisialisasi variabel net dengan jaringan yang telah dibuat tadi
- g. Membagi data testing dan data training
- h. Membuat array trnInd dengan indeks dari kolom terakhir tstInd+1 sampai N/14
- i. Membuat array tstInd dengan indeks dari 1 sampai M
- j. Menginisialisasi trainInd pada net dengan trnInd
- k. Menginisialisasi testInd pada net dengan tstInd
- l. Menjalankan jaringan JST dengan masukkan data ciri yang telah dinormalisasi dan targetnya
- m. Selesai

9. Algoritma testing_ akurasi

- a. Mulai
- b. Menjalankan jaringan untuk menguji data testing dengan memasukkan datanormal dari semua baris dan kolom tr.testInd dan dimasukkan ke dalam variabel ytst
- c. Mengambil semua label dari semua baris dan kolom tr.testInd dan dimasukkan ke dalam variabel ttst
- d. Membuat confusion matriks dengan variabel ytst dan ttst

- e. Menghitung jumlah data diagonal pada confusion matriks untuk mengetahui berapa banyak data yang benar
- f. Mengitung jumlah banyak data pada confusion matriks
- g. Menghitung akurasi dengan rumus $\text{akurasi} = (\text{benar} / \text{jumlah total data}) * 100$
- h. Selesai

10. Algoritma test_data_tunggal

- a. Mulai
- b. Melakukan pengambilan keputusan jika
- c. Jika file_data = 's1.data' maka ciri = ciri1
- d. Jika file_data = 's2.data' maka ciri = ciri2
- e. Jika file_data = 's3.data' maka ciri = ciri3
- f. Jika file_data = 's4.data' maka ciri = ciri4
- g. Jika file_data = 's5.data' maka ciri = ciri5
- h. Jika file_data = 's6.data' maka ciri = ciri6
- i. Mengisi variabel read1 dengan nilai matriks dari variabel ciri
- j. Menormalisasi variabel read1 kemudian di transpose, lalu hasilnya disimpan di variabel datanorm
- k. Menjalankan jaringan dengan masukan datanorm lalu hasilnya disimpan di ytst1
- l. Mengubah ytst1 dari vector menjadi indeks lalu hasilnya disimpan di variabel hasil
- m. Melakukan pengambilan keputusan jika
 - 1) Jika hasil = 1 maka hasil_huruf = G
 - 2) Jika hasil = 2 maka hasil_huruf = N
 - 3) Jika hasil = 3 maka hasil_huruf = D
 - 4) Jika hasil = 4 maka hasil_huruf = R
 - 5) Jika hasil = 5 maka hasil_huruf = M
 - 6) Jika hasil = 6 maka hasil_huruf = B
 - 7) Jika hasil = 7 maka hasil_huruf = S

- 8) Jika hasil = 8 maka hasil_huruf = NG
- 9) Jika hasil = 9 maka hasil_huruf = J
- 10) Jika hasil = 10 maka hasil_huruf = CH
- 11) Jika hasil = 11 maka hasil_huruf = K
- 12) Jika hasil = 12 maka hasil_huruf = T
- 13) Jika hasil = 13 maka hasil_huruf = P
- 14) Jika hasil = 14 maka hasil_huruf = H
- n. Selesai

11. Algoritma tampilan menu utama

- a. Menampilkan logo Universitas Sanata Dharma pada axes5
- b. Menampilkan judul dari penelitian pada GUI bagian atas tengah
- c. Menampilkan nama peneliti pada dibawah judul penelitian
- d. Menampilkan GUI yang dibagi menjadi 2 panel yaitu Pelatihan Data dan Uji Data Tunggal
- e. Selesai

12. Algoritma Panel Pelatihan Data

- a. Memanggil script load data pada tombol Load Data Ciri
- b. Menampilkan data ciri yang sudah di load pada table
- c. Pada panel Training Data
- d. Memasukkan jumlah hidden layer 1 pada field Hidden Layer 1
- e. Memasukkan jumlah hidden layer 2 pada field Hidden Layer 2
- f. Melakukan pelatihan data dengan menjalankan script trainn pada tombol Lakukan Pelatihan
- g. Menampilkan hasil dari script testing_akurasi pada field Akurasi%
- h. Selesai

13. Algoritma Panel Uji Data Tunggal

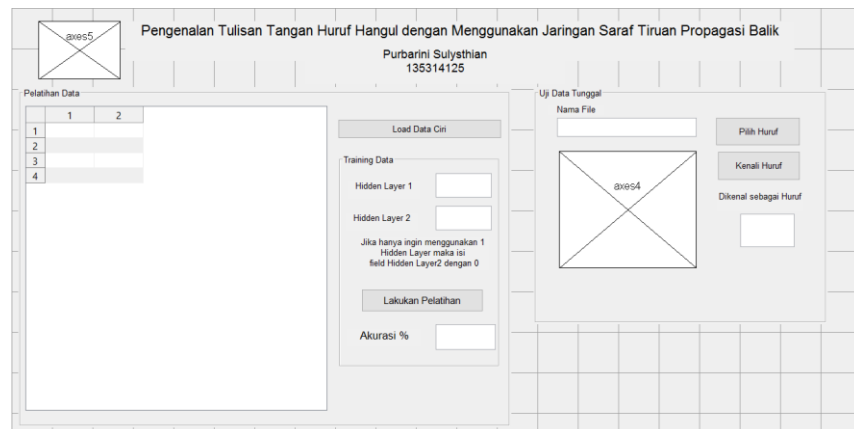
- a. Membaca data huruf yang akan di uji pada tombol pilih Huruf

- b. Menampilkan lokasi data huruf yang akan diuji pada field Nama Field
- c. Menampilkan gambar yang akan di uji pada axes4
- d. Memanggil script test_data_tunggal pada tombol Kenali Huruf
- e. Menampilkan hasil pengenalan tulisan tangan huruf Hangul pada field Dikenal sebagai Huruf
- f. Selesai

3.6 Perancangan Antarmuka Sistem

Pada penelitian ini dibuat juga *Graphic User Interface* yang dapat memudahkan pengguna lebih mudah melihat jalannya program dari alat uji. Rancangan dari GUI dapat dilihat pada gambar dibawah

1. Tampilan Utama



Gambar 3. 13 Contoh Tampilan Utama Alat Uji

2. Panel Pelatihan Data

Pelatihan Data

	1	2
1		
2		
3		
4		

Load Data Ciri

Training Data

Hidden Layer 1

Hidden Layer 2

Jika hanya ingin menggunakan 1 Hidden Layer maka isi field Hidden Layer2 dengan 0

Lakukan Pelatihan

Akurasi %

Gambar 3. 14 Contoh Tampilan Panel Pelatihan Data

3. Panel Uji Data Tunggal

Uji Data Tunggal

Nama File

Pilih Huruf

Kenali Huruf

Dikenal sebagai Huruf

axes4

Gambar 3. 15 Contoh Tampilan Panel Uji Data Tunggal

3.7 Skenario Pengujian

Langkah pertama dari pengujian ini adalah membagi data training dan testing. Jumlah total data yang dipakai adalah 602 huruf dimana masing-masing data huruf ada 43 huruf. Pembagian data training adalah $\frac{2}{3}$ dari seluruh data yang ada, sehingga data yang dipakai untuk pelatihan ada 28 data per huruf. Dan jumlah data yang dipakai untuk testing adalah $\frac{1}{3}$ dari seluruh data yang ada, sehingga data yang dipakai untuk testing ada 15 data.

Langkah selanjutnya adalah mengekstrak ciri dari masing-masing huruf yang ada. Jika semua data sudah diekstrak cirinya kemudian disimpan dalam bentuk .data. Dalam mengekstrak ciri data huruf, dilakukan 6 kombinasi dari kedua metode.

Setelah mendapat ciri dari huruf, kemudian membagi target dan mengubahnya menjadi vector target untuk digunakan dalam pelatihan jaringan. Pada pelatihan jaringan, telah disiapkan struktur jaringan menggunakan hidden layer, fungsi aktivasi dan target. Dalam penelitian ini menggunakan 90 hidden layer, fungsi aktivasi logsig, dan targetnya adalah berupa matriks berukuran 14×1 .

Setelah proses training selesai maka selanjutnya adalah testing dan akurasi. Proses ini bertujuan untuk mengetahui bagaimana performa dari alat uji yang sudah dibuat. Kemudian akan dilihat apakah akurasi dari pengenalan data testing tersebut sudah memenuhi $\geq 70\%$ atau belum. Apabila hasil akurasi yang dihasilkan belum mencapai 70% maka akan dilakukan perubahan. Perubahan yang dapat dilakukan antara lain, mengubah jumlah neuron pada *hidden layer*, mengubah fungsi aktivasi pada lapisan tersembunyi atau lapisan keluaran, mengubah fungsi pelatihan yang dipakai dalam struktur jaringan, mengubah jumlah lapisan tersembunyi dan mengubah banyaknya iterasi yang dilakukan ketika menjalankan *backpropagation*. (Nugroho, 2016)

Langkah selanjutnya adalah pengenalan data huruf dengan mencocokkan ciri dari data baru dengan data ciri yang sudah ada.

BAB IV

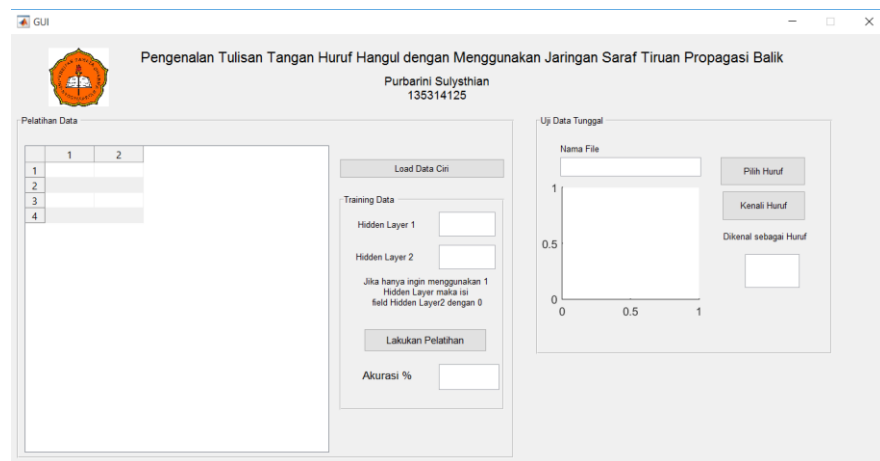
HASIL UJI DAN ANALISIS

4.1 Contoh Hasil Antarmuka

Desain antarmuka yang sudah dibuat dapat berjalan dengan baik, berikut adalah hasil dari running alat uji.

1. Menu Utama

Gambar dibawah ini menunjukkan tampilan utama dari desain antarmuka yang telah dibuat. Pada menu utama ini terdapat panel Pelatihan Data dan panel Uji Data Tunggal



Gambar 4. 1 Antarmuka Menu Utama Alat Uji

2. Panel Pelatihan Data

Gambar dibawah menunjukkan tampilan dari panel Pelatihan Data. Panel ini berfungsi untuk me-load data ciri yang sudah ada dan melakukan pelatihan data.

Pelatihan Data

	1	2	3	4	5	
1	910	839	1024	892	2	^
2	924	833	1024	933	2	
3	955	915	1024	961	0	
4	960	927	1024	979	1	
5	963	904	1024	936	1	
6	952	904	1024	964	0	
7	952	896	1024	966	0	
8	962	951	1024	994	2	
9	957	929	1024	980	0	
10	974	915	1024	969	0	
11	895	845	1024	976	4	
12	989	949	1024	998	0	
13	975	956	1024	998	0	
14	899	846	1024	924	3	
15	943	861	1024	929	0	
16	930	852	1024	925	4	
17	929	895	1024	951	0	
18	936	883	1024	942	3	v

Load Data Ciri

Training Data

Hidden Layer 1

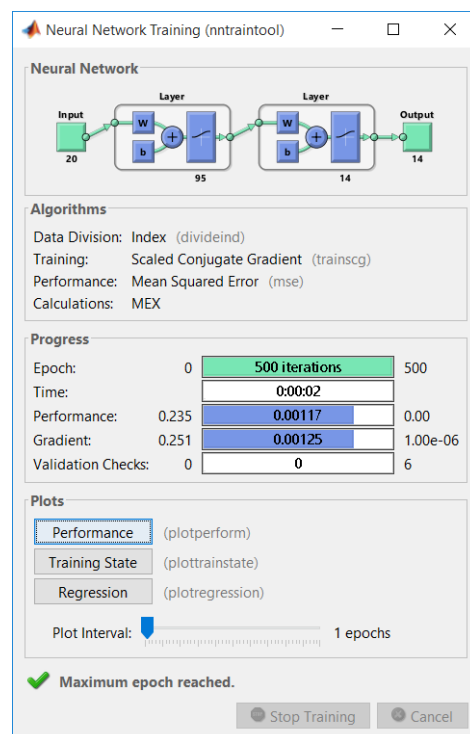
Hidden Layer 2

Jika hanya ingin menggunakan 1 Hidden Layer maka isi field Hidden Layer2 dengan 0

Lakukan Pelatihan

Akurasi %

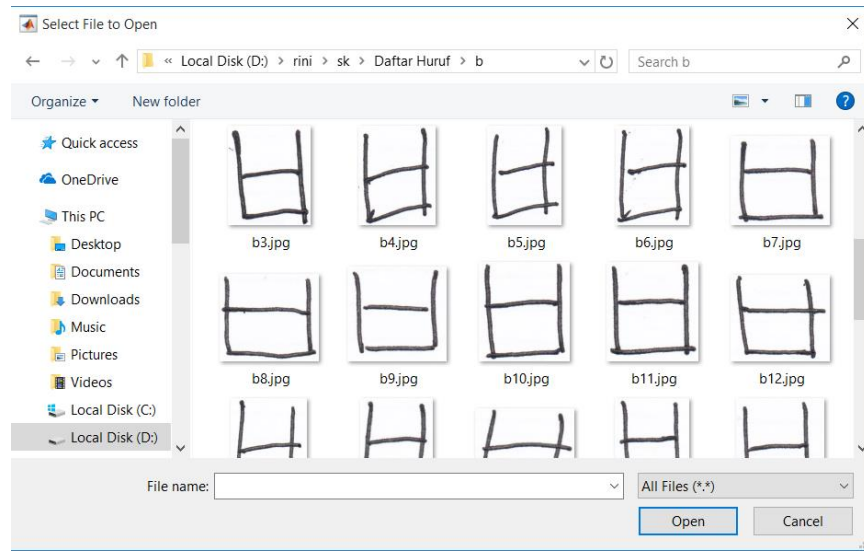
Gambar 4. 2 Antarmuka Panel Pelatihan Data



Gambar 4. 3 Contoh Hasil Pelatihan Data

3. Panel Uji Data Tunggal

Gambar dibawah menunjukkan panel Uji Data Tunggal. Panel ini berfungsi untuk melakukan pengujian data tunggal, proses kerja dari panel ini pertama pilih data yang akan diuji, kemudian tekan tombol Kenal Huruf.



Gambar 4. 4 Contoh Men-load Data



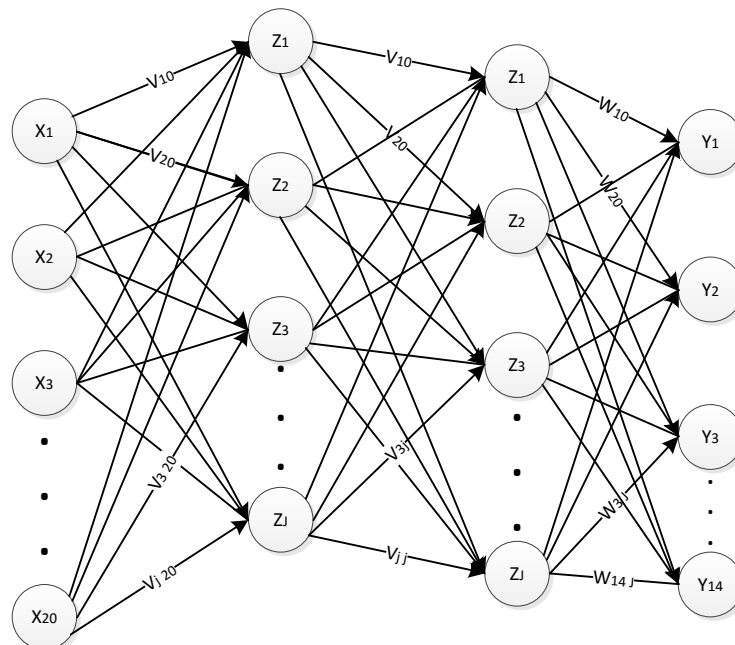
Gambar 4. 5 Antarmuka Panel Uji Data Tunggal



Gambar 4. 6 Contoh Hasil Pengujian Data Tunggal

4.2 Desain Jaringan

Pada penelitian ini penguji mendesain jaringan yang terdiri dari layer masukan, layer tersembunyi, dan layer keluaran. Desain jaringan tersebut adalah sebagai berikut:



Gambar 4. 7 Desain Jaringan

4.3 Hasil Uji

Pada penelitian ini, peneliti melakukan dua belas percobaan dengan jumlah hidden layer dan fungsi aktivasi yang berbeda. Untuk setiap percobaan kombinasi ciri yang dipakai ada 6 kombinasi, yaitu:

1. Kombinasi ciri 1(ciri dari *Intensity of Character*)
2. Kombinasi ciri 2 (ciri dari *Mark Direction* diagonal 1)
3. Kombinasi ciri 1 dan ciri 2
4. Kombinasi ciri 1 dan ciri 3 (ciri dari *Mark Direction* diagonal 2)
5. Kombinasi ciri 2 dan ciri 3
6. Kombinasi ciri 1, ciri 2, ciri 3, ciri 4 (ciri dari *Mark Direction* horisontal), dan ciri 5 (ciri dari *Mark Direction*)

Adapun hasil percobaan yang telah dilakukan dapat dilihat sebagai berikut:

1. Percobaan 1

Pada percobaan 1, peneliti menggunakan metode ekstraksi ciri Zonning ZCZ. Percobaan ini menggunakan 1 *hidden layer* dengan jumlah neuron pada *hidden layer* sebanyak 95, fungsi aktivasi pada hidden layer memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah *Epoch* yang digunakan adalah 500. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *tansig*. Berikut adalah hasil dari percobaan 1

a. Program jaringan Backpropagation

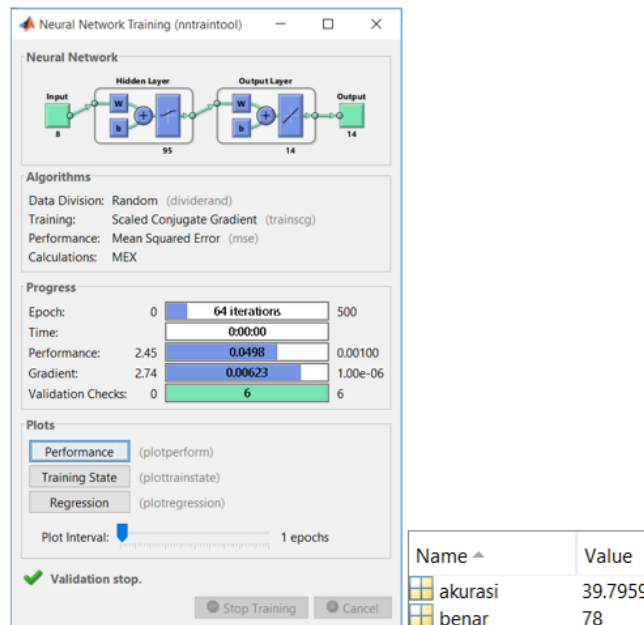
```

coba4.m
55 - pTrain = [isiData(:,id1(1:28)) isiData(:,id2(1:28))...
56         isiData(:,id3(1:28)) isiData(:,id4(1:28)) isiData(:,id5(1:28))...
57         isiData(:,id6(1:28)) isiData(:,id7(1:28)) isiData(:,id8(1:28))...
58         isiData(:,id9(1:28)) isiData(:,id10(1:28)) isiData(:,id11(1:28))...
59         isiData(:,id12(1:28)) isiData(:,id13(1:28)) isiData(:,id14(1:28))];
60
61 - lbTrain = [label(:,id1(1:28)) label(:,id2(1:28))...
62            label(:,id3(1:28)) label(:,id4(1:28)) label(:,id5(1:28))...
63            label(:,id6(1:28)) label(:,id7(1:28)) label(:,id8(1:28))...
64            label(:,id9(1:28)) label(:,id10(1:28)) label(:,id11(1:28))...
65            label(:,id12(1:28)) label(:,id13(1:28)) label(:,id14(1:28))];
66
67 - pTest = [isiData(:,id1(29:end)) isiData(:,id2(29:end))...
68           isiData(:,id3(29:end)) isiData(:,id4(29:end)) isiData(:,id5(29:end))...
69           isiData(:,id6(29:end)) isiData(:,id7(29:end)) isiData(:,id8(29:end))...
70           isiData(:,id9(29:end)) isiData(:,id10(29:end)) isiData(:,id11(29:end))...
71           isiData(:,id12(29:end)) isiData(:,id13(29:end)) isiData(:,id14(29:end))];
72
73 - lbTest = [label(:,id1(29:end)) label(:,id2(29:end))...
74            label(:,id3(29:end)) label(:,id4(29:end)) label(:,id5(29:end))...
75            label(:,id6(29:end)) label(:,id7(29:end)) label(:,id8(29:end))...
76            label(:,id9(29:end)) label(:,id10(29:end)) label(:,id11(29:end))...
77            label(:,id12(29:end)) label(:,id13(29:end)) label(:,id14(29:end))];
78
79 - net = newff(pTrain, lbTrain, 95, {'logsig'}, 'trainscg');

```

Gambar 4. 8 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 1

b. Output



Gambar 4. 9 Hasil Perobaan 1

Dapat dilihat dari gambar diatas bahwa dari 588 data huruf, data benar yang dapat terbaca ada 78 data dengan akurasi 39,7959%. Waktu dari pelatihan tersebut 00:00, MSE sebesar 0.00100 dan gradient sebesar 1.00E-06.

2. Percobaan 2

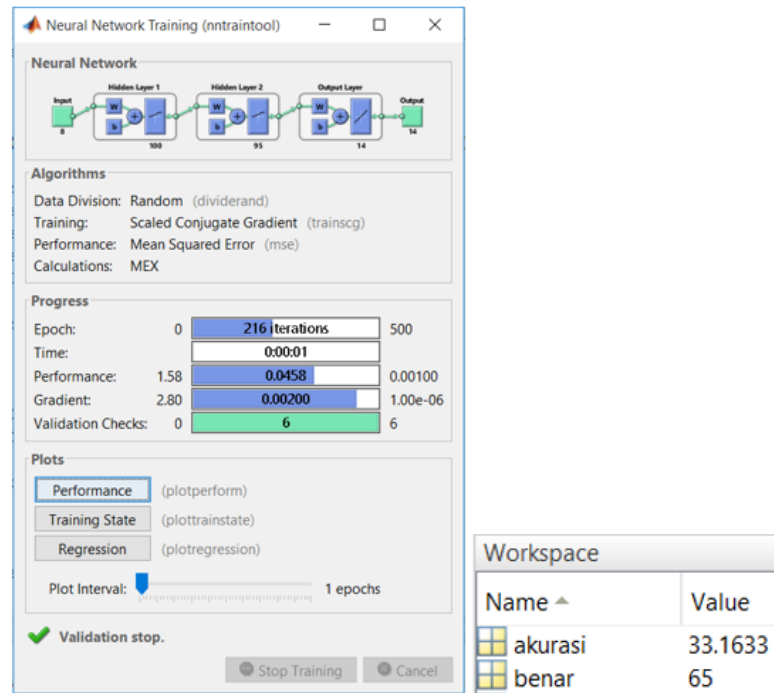
Pada percobaan 2, peneliti menggunakan metode ekstraksi ciri Zonning ZCZ. Percobaan ini menggunakan 2 *hidden layer* dengan jumlah neuron pada *hidden layer* 1 sebanyak 100 dan *hidden layer* 2 sebanyak 95, fungsi aktivasi pada hidden layer memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *tansig*. Berikut adalah hasil dari percobaan 1

a. Program jaringan Backpropagation

```
coba4.m
55 - pTrain = [isiData(:,id1(1:28)) isiData(:,id2(1:28))...
56         isiData(:,id3(1:28)) isiData(:,id4(1:28)) isiData(:,id5(1:28))...
57         isiData(:,id6(1:28)) isiData(:,id7(1:28)) isiData(:,id8(1:28))...
58         isiData(:,id9(1:28)) isiData(:,id10(1:28)) isiData(:,id11(1:28))...
59         isiData(:,id12(1:28)) isiData(:,id13(1:28)) isiData(:,id14(1:28))];
60
61 - lbTrain = [label(:,id1(1:28)) label(:,id2(1:28))...
62         label(:,id3(1:28)) label(:,id4(1:28)) label(:,id5(1:28))...
63         label(:,id6(1:28)) label(:,id7(1:28)) label(:,id8(1:28))...
64         label(:,id9(1:28)) label(:,id10(1:28)) label(:,id11(1:28))...
65         label(:,id12(1:28)) label(:,id13(1:28)) label(:,id14(1:28))];
66
67 - pTest = [isiData(:,id1(29:end)) isiData(:,id2(29:end))...
68         isiData(:,id3(29:end)) isiData(:,id4(29:end)) isiData(:,id5(29:end))...
69         isiData(:,id6(29:end)) isiData(:,id7(29:end)) isiData(:,id8(29:end))...
70         isiData(:,id9(29:end)) isiData(:,id10(29:end)) isiData(:,id11(29:end))...
71         isiData(:,id12(29:end)) isiData(:,id13(29:end)) isiData(:,id14(29:end))];
72
73 - lbTest = [label(:,id1(29:end)) label(:,id2(29:end))...
74         label(:,id3(29:end)) label(:,id4(29:end)) label(:,id5(29:end))...
75         label(:,id6(29:end)) label(:,id7(29:end)) label(:,id8(29:end))...
76         label(:,id9(29:end)) label(:,id10(29:end)) label(:,id11(29:end))...
77         label(:,id12(29:end)) label(:,id13(29:end)) label(:,id14(29:end))];
78
79 - net = newff(pTrain, lbTrain, [100, 95], {'logsig', 'logsig'}, 'trainscg');
```

Gambar 4. 10 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 2

b. Output



Gambar 4. 11 Hasil Percobaan 2

Dapat dilihat dari gambar diatas bahwa dari 588 data huruf, data benar yang dapat terbaca ada 65 data dengan akurasi 33,1633%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.00100 dan gradient sebesar 1.00E-6.

3. Percobaan 3

Pada percobaan 3, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 1 hidden layer dengan jumlah neuron pada hidden layer sebanyak 90, fungsi aktivasi pada hidden layer memakai *tansig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *tansig*. Berikut adalah hasil dari percobaan 3

a. Program jaringan Backpropagation

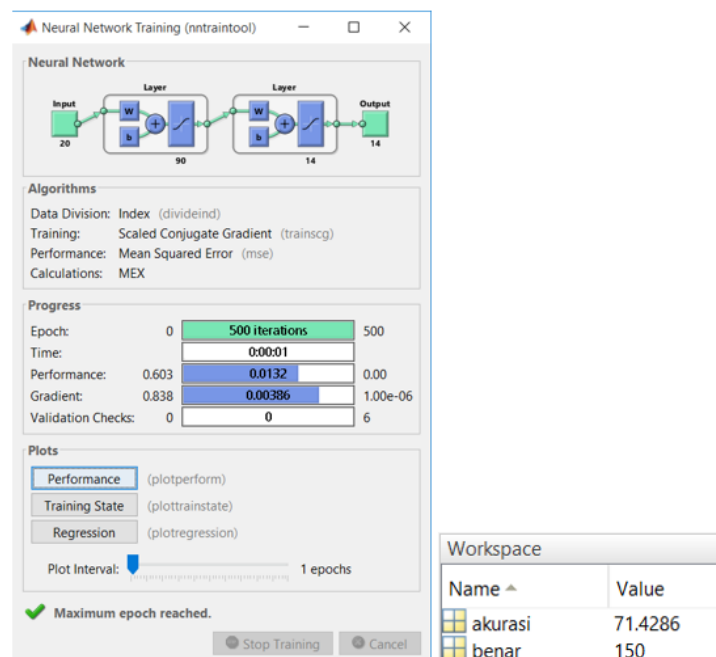
```

trainJT.m
1  M = 15;
2  rng(0)
3  net = newff(minmax(datanormal), [90, 14], {'tansig', 'tansig'}, 'trainscg');
4  net.trainParam.epochs = 500;
5  net.divideFcn = 'divideind';
6  net = init(net);
7  net.trainParam.lr = 0.01;
8
9  tstind = 1:M;
10 trnind = tstind(end)+1:N/14;
11
12 trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13          ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14          ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15          ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16          ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19          ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20          ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21          ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22          ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 net.divideParam.trainInd = trnInd;
25 net.divideParam.testInd = tstInd;
26
27 [net, tr] = train(net, datanormal, label);

```

Gambar 4. 12 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 3

b. Output



Gambar 4. 13 Hasil Percobaan 3

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 150 data dengan akurasi 71,4286%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.603 dan gradient sebesar 1.00E-06.

4. Percobaan 4

Pada percobaan 4, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 1 hidden layer dengan jumlah neuron pada hidden layer sebanyak 94, fungsi aktivasi pada hidden layer memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainrp*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 4

a. Program jaringan Backpropagation

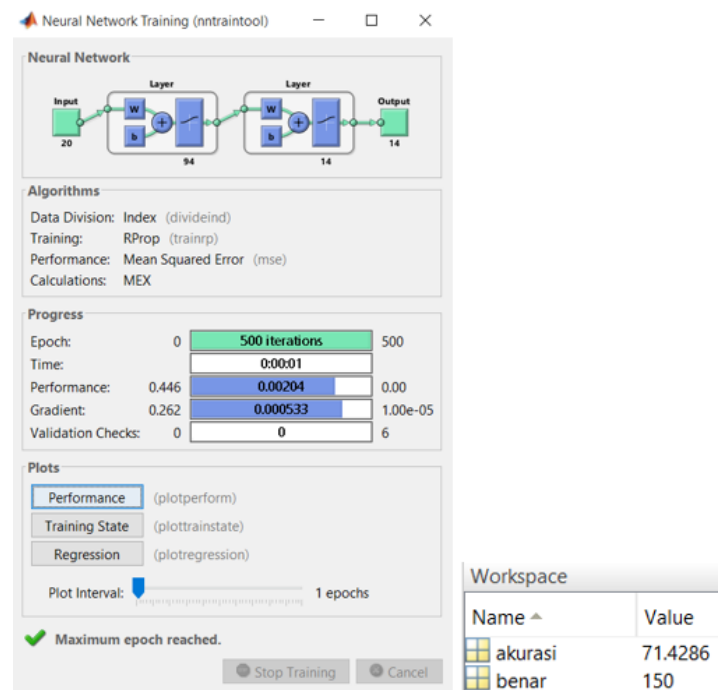
```

trainJT.m  x +
1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[94, 14],{'logsig','logsig'}, 'trainrp');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstind = 1:M;
10 - trnind = tstind(end)+1:N/14;
11
12 - trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13             ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14             ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15             ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16             ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 - tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19             ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20             ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21             ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22             ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 - net.divideParam.trainInd = trnInd;
25 - net.divideParam.testInd = tstInd;
26
27 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 14 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 4

b. Output



Gambar 4. 15 Hasil Percobaan 4

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 150 data dengan akurasi 71,4286%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.446 dan gradient sebesar 1.00E-05.

5. Percobaan 5

Pada percobaan 5, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 1 hidden layer dengan jumlah neuron pada hidden layer sebanyak 95, fungsi aktivasi pada hidden layer memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 5

a. Program jaringan Backpropagation

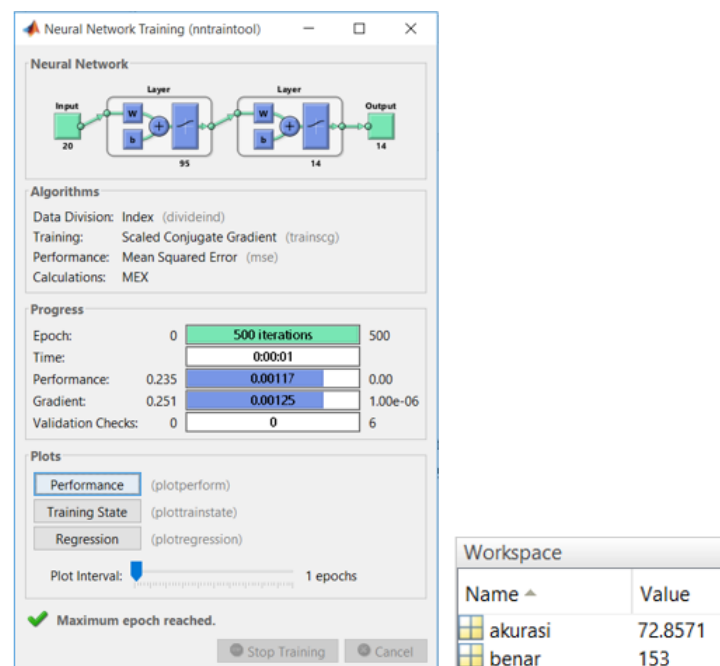
```

trainJST.m
1  M = 15;
2  rng(0)
3  net = newff(minmax(datanormal), [95, 14], {'logsig', 'logsig'}, 'trainscg');
4  net.trainParam.epochs = 500;
5  net.divideFcn = 'divideind';
6  net = init(net);
7  net.trainParam.lr = 0.01;
8
9  tstind = 1:M;
10 trnind = tstind(end)+1:N/14;
11
12 trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13          ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14          ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15          ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16          ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19          ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20          ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21          ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22          ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 net.divideParam.trainInd = trnInd;
25 net.divideParam.testInd = tstInd;
26
27 [net, tr] = train(net, datanormal, label);

```

Gambar 4. 16 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 5

b. Output



Gambar 4. 17 Hasil Percobaan 5

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 153 data dengan akurasi 72,8571%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.235 dan gradient sebesar 1.00E-06.

6. Percobaan 6

Pada percobaan 6, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 1 hidden layer dengan jumlah neuron pada hidden layer sebanyak 95, fungsi aktivasi pada hidden layer memakai *tansig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *tansig*. Berikut adalah hasil dari percobaan 6

a. Program jaringan Backpropagation

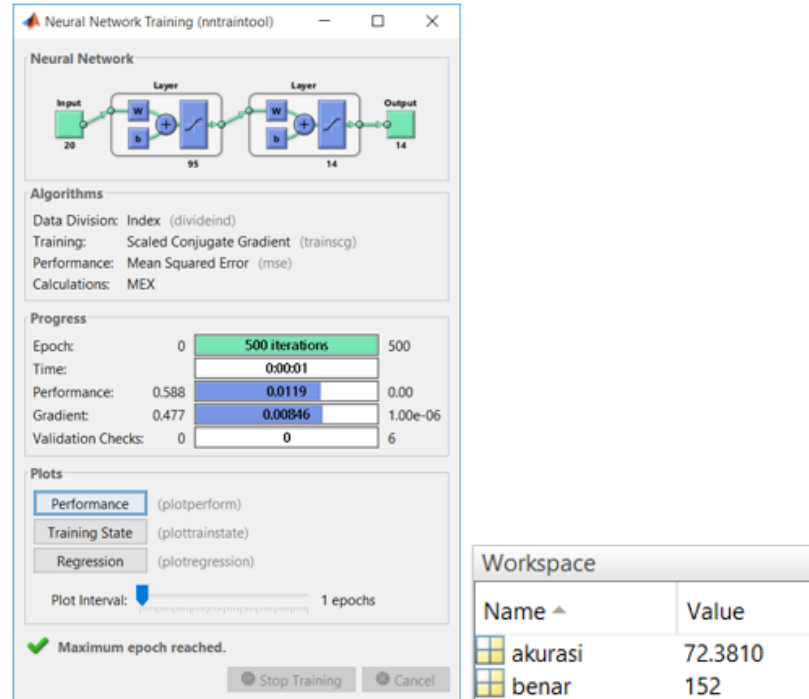
```

trainJT.m
1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[95, 14],{'tansig','tansig'}, 'trainscg');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstind = 1:M;
10 - trnind = tstind(end)+1:N/14;
11
12 - trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13             ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14             ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15             ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16             ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 - tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19             ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20             ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21             ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22             ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 - net.divideParam.trainInd = trnInd;
25 - net.divideParam.testInd = tstInd;
26
27 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 18 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 6

b. Output



Gambar 4. 19 Hasil Percobaan 6

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 152 data dengan akurasi 72,3810%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.588 dan gradient sebesar 1.00E-06.

7. Percobaan 7

Pada percobaan 7, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 1 hidden layer dengan jumlah neuron pada hidden layer sebanyak 90, fungsi aktivasi pada hidden layer memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainrp*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 7

a. Program jaringan Backpropagation

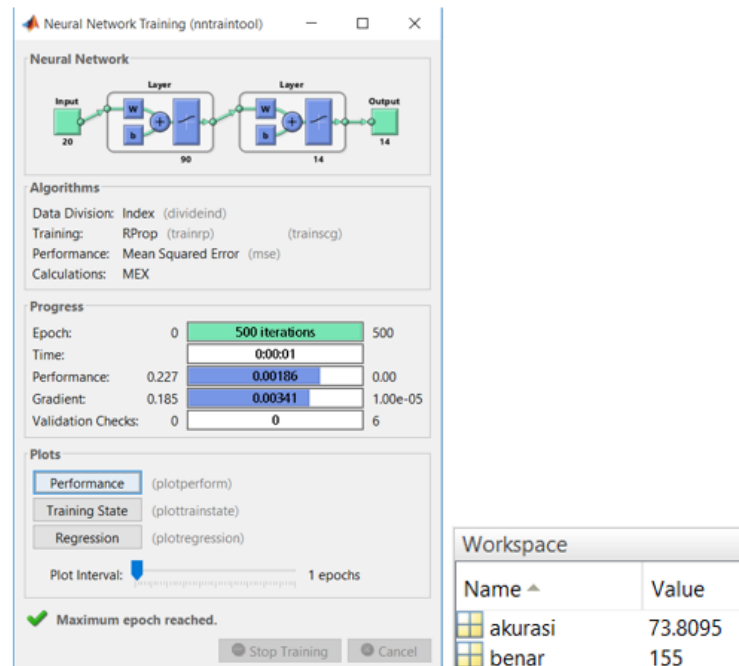
```

trainJT.m
1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[90, 14],{'logsig','logsig'}, 'trainrp');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstind = 1:M;
10 - trnind = tstind(end)+1:N/14;
11
12 - trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13            ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14            ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15            ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16            ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 - tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19            ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20            ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21            ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22            ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 - net.divideParam.trainInd = trnInd;
25 - net.divideParam.testInd = tstInd;
26
27 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 20 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 7

b. Output



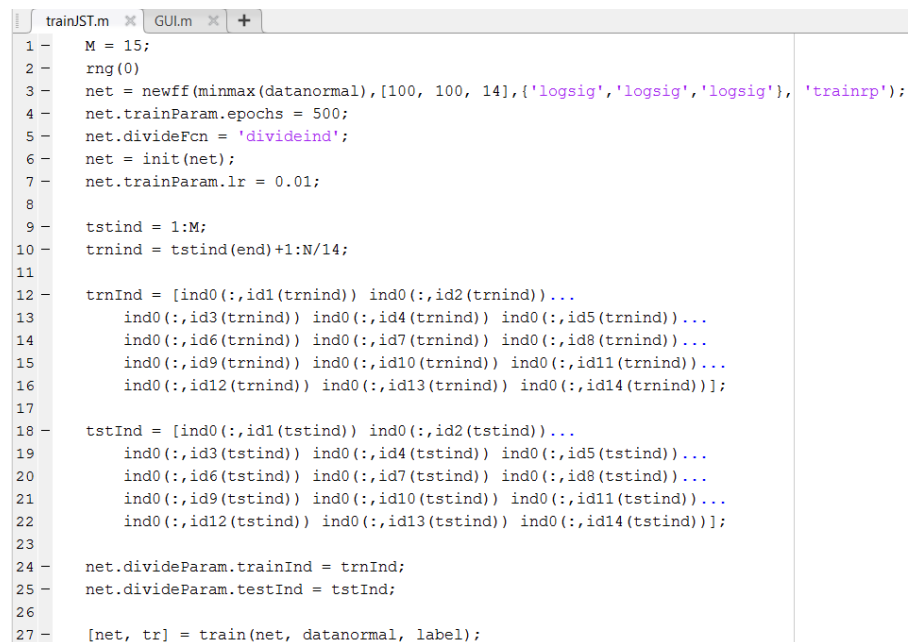
Gambar 4. 21 Hasil Percobaan 7

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 155 data dengan akurasi 73,8095%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.277 dan gradient sebesar 1.00E-05.

8. Percobaan 8

Pada percobaan 8, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 2 hidden layer dengan jumlah neuron pada hidden layer 1 sebanyak 100, hidden layer 2 sebanyak 100, fungsi aktivasi pada hidden layer 1 memakai *logsig*, fungsi aktivasi pada hidden layer 2 memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainrp*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 8

a. Program jaringan Backpropagation



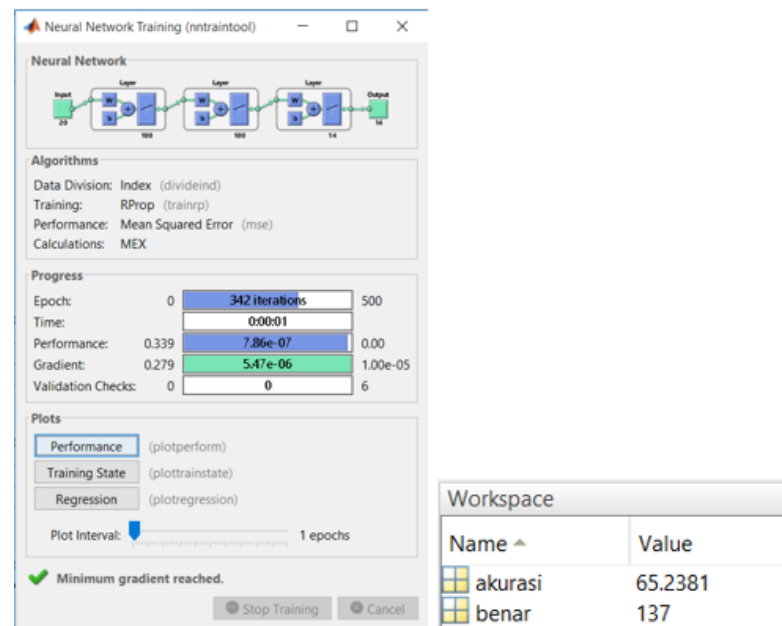
```

1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[100, 100, 14],{'logsig','logsig','logsig'}, 'trainrp');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstind = 1:M;
10 - trnind = tstind(end)+1:N/14;
11
12 - trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind)) ...
13           ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind)) ...
14           ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind)) ...
15           ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind)) ...
16           ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 - tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind)) ...
19           ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind)) ...
20           ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind)) ...
21           ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind)) ...
22           ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 - net.divideParam.trainInd = trnInd;
25 - net.divideParam.testInd = tstInd;
26
27 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 22 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 8

b. Output



Gambar 4. 23 Hasil dari Percobaan 8

Dapat dilihat dari gambar diatas bahwa dari 620 data huruf, data benar yang dapat terbaca ada 137 data dengan akurasi 65,2381%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.339 dan gradient sebesar 1.00E-05.

9. Percobaan 9

Pada percobaan 9, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 2 hidden layer dengan jumlah neuron pada hidden layer 1 sebanyak 95, hidden layer 2 sebanyak 100, fungsi aktivasi pada hidden layer 1 memakai *logsig*, fungsi aktivasi pada hidden layer 2 memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainrp*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 9

a. Program jaringan Backpropagation

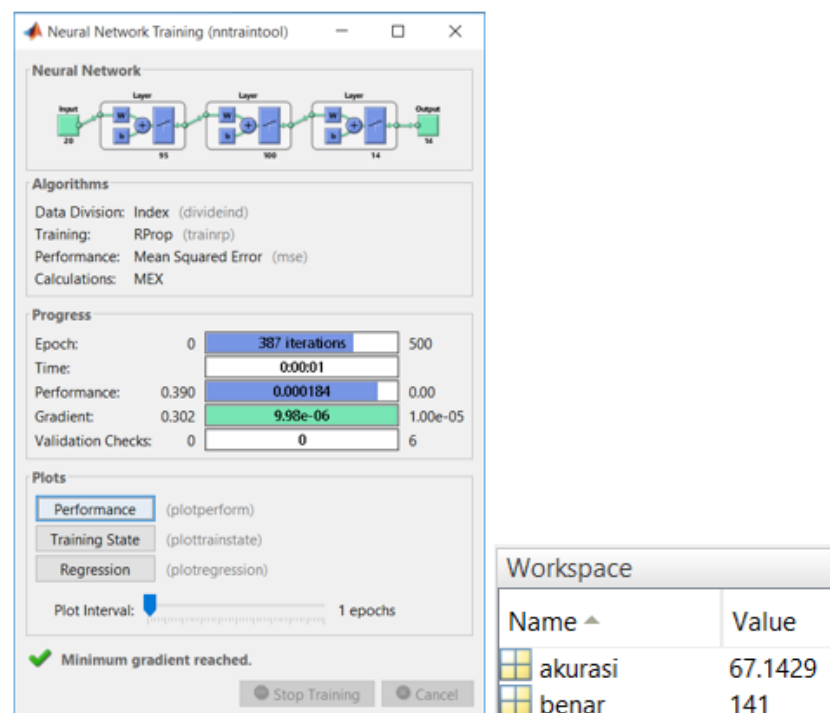
```

trainST.m  GUI.m  +
1  M = 15;
2  rng(0)
3  net = newff(minmax(datanormal),[95, 100, 14],{'logsig','logsig','logsig'}, 'trainrp');
4  net.trainParam.epochs = 500;
5  net.divideFcn = 'divideind';
6  net = init(net);
7  net.trainParam.lr = 0.01;
8
9  tstind = 1:M;
10 trnind = tstind(end)+1:N/14;
11
12 trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13          ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14          ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15          ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16          ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19          ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20          ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21          ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22          ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 net.divideParam.trainInd = trnInd;
25 net.divideParam.testInd = tstInd;
26
27 [net, tr] = train(net, datanormal, label);

```

Gambar 4. 24 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 9

b. Output



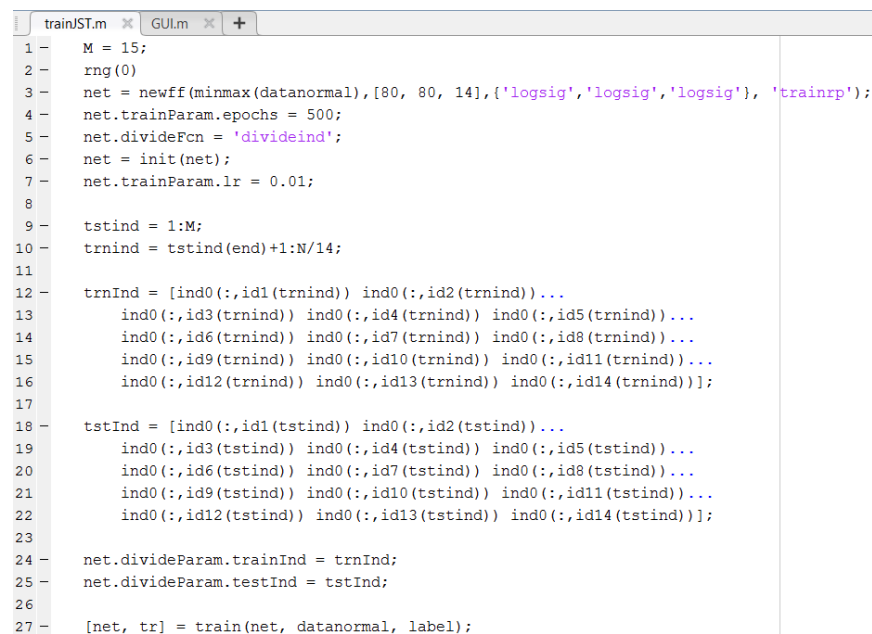
Gambar 4. 25 Hasil dari Percobaan 9

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 141 data dengan akurasi 67,1429%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.390 dan gradient sebesar 1.00E-05.

10. Percobaan 10

Pada percobaan 10, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 2 hidden layer dengan jumlah neuron pada hidden layer 1 sebanyak 80, hidden layer 2 sebanyak 80, fungsi aktivasi pada hidden layer 1 memakai *logsig*, fungsi aktivasi pada hidden layer 2 memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainrp*, dan jumlah *Epoch* sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 10

a. Program jaringan Backpropagation



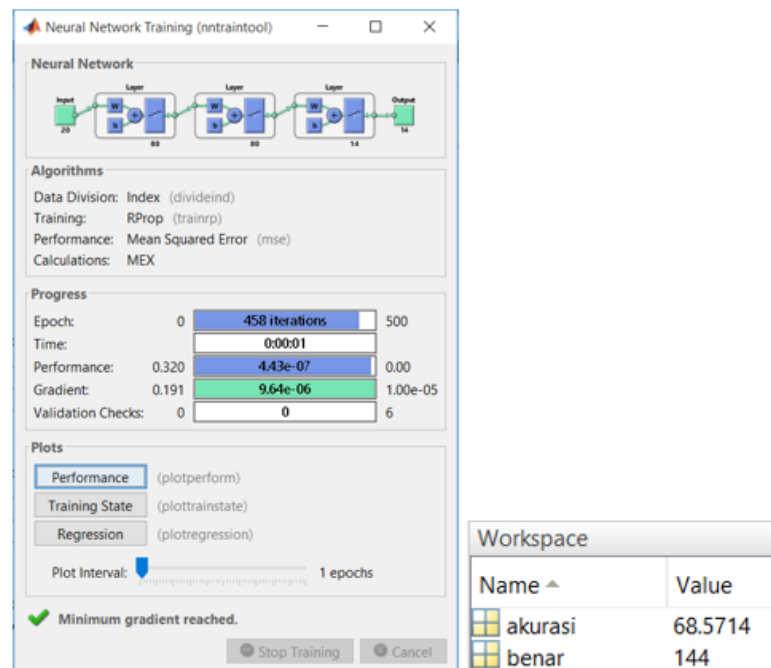
```

1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[80, 80, 14],{'logsig','logsig','logsig'}, 'trainrp');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstind = 1:M;
10 - trnind = tstind(end)+1:N/14;
11
12 - trnInd = [ind0(:,id1(trnind)) ind0(:,id2(trnind))...
13           ind0(:,id3(trnind)) ind0(:,id4(trnind)) ind0(:,id5(trnind))...
14           ind0(:,id6(trnind)) ind0(:,id7(trnind)) ind0(:,id8(trnind))...
15           ind0(:,id9(trnind)) ind0(:,id10(trnind)) ind0(:,id11(trnind))...
16           ind0(:,id12(trnind)) ind0(:,id13(trnind)) ind0(:,id14(trnind))];
17
18 - tstInd = [ind0(:,id1(tstind)) ind0(:,id2(tstind))...
19           ind0(:,id3(tstind)) ind0(:,id4(tstind)) ind0(:,id5(tstind))...
20           ind0(:,id6(tstind)) ind0(:,id7(tstind)) ind0(:,id8(tstind))...
21           ind0(:,id9(tstind)) ind0(:,id10(tstind)) ind0(:,id11(tstind))...
22           ind0(:,id12(tstind)) ind0(:,id13(tstind)) ind0(:,id14(tstind))];
23
24 - net.divideParam.trainInd = trnInd;
25 - net.divideParam.testInd = tstInd;
26
27 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 26 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 10

b. Output



Gambar 4. 27 Hasil dari Percobaan 10

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 144 data dengan akurasi 68,5741%. Waktu dari pelatihan tersebut 00:01, MSE sebesar 0.320 dan gradient sebesar 1.00E-05.

11. Percobaan 11

Pada percobaan 11, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 2 hidden layer dengan jumlah neuron pada hidden layer 1 sebanyak 100, hidden layer 2 sebanyak 100, fungsi aktivasi pada hidden layer 1 memakai *logsig*, fungsi aktivasi pada hidden layer 2 memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah iterasi sebanyak 500 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 11

a. Program jaringan Backpropagation

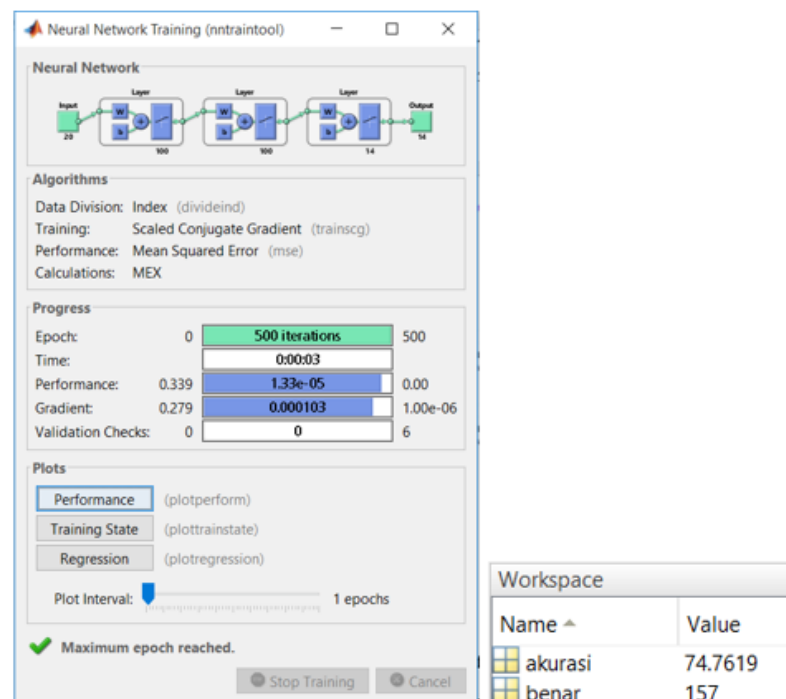
```

trainSTM.m  x  GULm  x  +
1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[100, 100, 14],{'logsig','logsig','logsig'}, 'trainscg');
4 - net.trainParam.epochs = 500;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstInd = 1:M;
10 - trnInd = tstInd(end)+1:N/14;
11
12 - trnInd = [ind0(:,id1(trnInd)) ind0(:,id2(trnInd))...
13            ind0(:,id3(trnInd)) ind0(:,id4(trnInd)) ind0(:,id5(trnInd))...
14            ind0(:,id6(trnInd)) ind0(:,id7(trnInd)) ind0(:,id8(trnInd))...
15            ind0(:,id9(trnInd)) ind0(:,id10(trnInd)) ind0(:,id11(trnInd))...
16            ind0(:,id12(trnInd)) ind0(:,id13(trnInd)) ind0(:,id14(trnInd))];
17
18 - tstInd = [ind0(:,id1(tstInd)) ind0(:,id2(tstInd))...
19            ind0(:,id3(tstInd)) ind0(:,id4(tstInd)) ind0(:,id5(tstInd))...
20            ind0(:,id6(tstInd)) ind0(:,id7(tstInd)) ind0(:,id8(tstInd))...
21            ind0(:,id9(tstInd)) ind0(:,id10(tstInd)) ind0(:,id11(tstInd))...
22            ind0(:,id12(tstInd)) ind0(:,id13(tstInd)) ind0(:,id14(tstInd))];
23
24 - net.divideParam.trainInd = trnInd;
25 - net.divideParam.testInd = tstInd;
26
27 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 28 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 11

b. Output



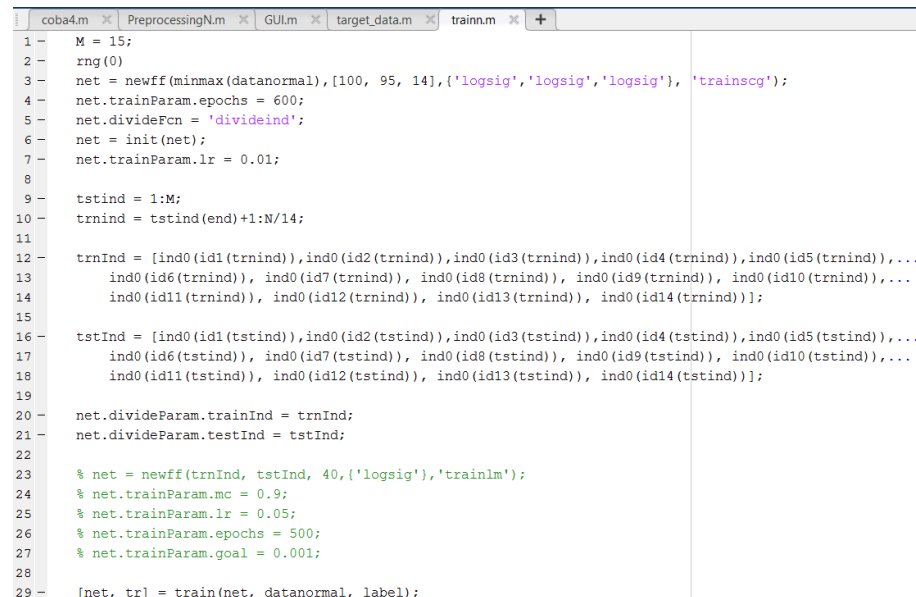
Gambar 4. 29 Hasil Percobaan 11

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 157 data dengan akurasi 74,7619%. Waktu dari pelatihan tersebut 00:03, MSE sebesar 0.339 dan gradient sebesar 1.00E-06.

12. Percobaan 12

Pada percobaan 12, peneliti menggunakan metode ekstraksi ciri *Intensity of Character* dan *Mark Direction*. Percobaan ini menggunakan 2 hidden layer dengan jumlah neuron pada hidden layer 1 sebanyak 100, hidden layer 2 sebanyak 85, fungsi aktivasi pada hidden layer 1 memakai *logsig*, fungsi aktivasi pada hidden layer 2 memakai *logsig*, fungsi pelatihan jaringan menggunakan *trainscg*, dan jumlah iterasi sebanyak 600 kali. Pada lapisan keluaran terdapat 14 unit neuron dengan fungsi aktivasi *logsig*. Berikut adalah hasil dari percobaan 12

a. Program jaringan Backpropagation



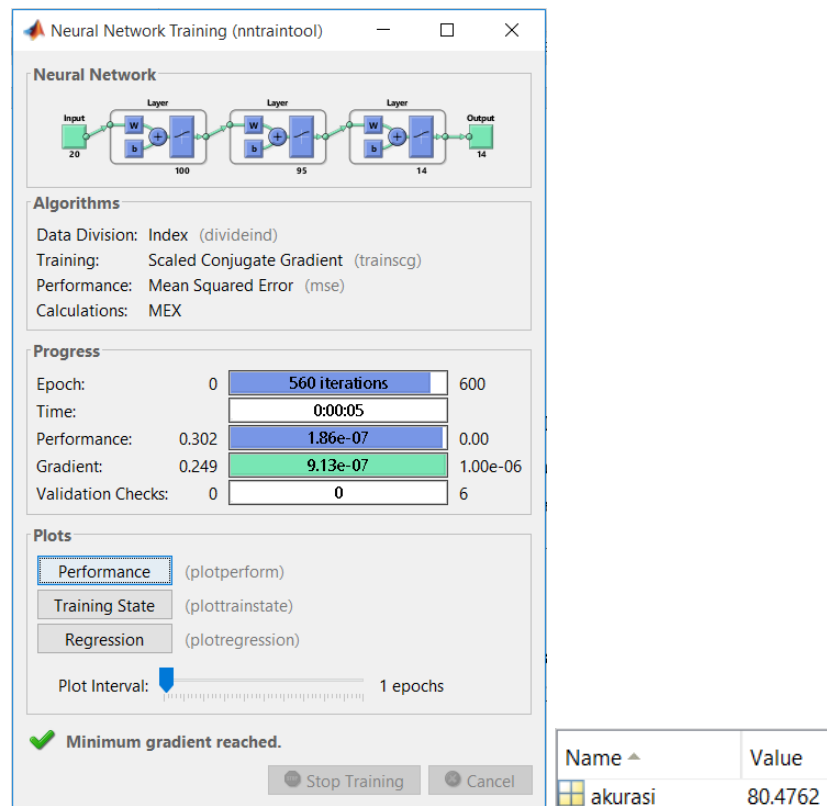
```

1 - M = 15;
2 - rng(0)
3 - net = newff(minmax(datanormal),[100, 95, 14],{'logsig','logsig','logsig'}, 'trainscg');
4 - net.trainParam.epochs = 600;
5 - net.divideFcn = 'divideind';
6 - net = init(net);
7 - net.trainParam.lr = 0.01;
8
9 - tstind = 1:M;
10 - trnind = tstind(end)+1:N/14;
11
12 - trnInd = [ind0(id1(trnind)), ind0(id2(trnind)), ind0(id3(trnind)), ind0(id4(trnind)), ind0(id5(trnind)), ...
13 -         ind0(id6(trnind)), ind0(id7(trnind)), ind0(id8(trnind)), ind0(id9(trnind)), ind0(id10(trnind)), ...
14 -         ind0(id11(trnind)), ind0(id12(trnind)), ind0(id13(trnind)), ind0(id14(trnind))];
15
16 - tstInd = [ind0(id1(tstind)), ind0(id2(tstind)), ind0(id3(tstind)), ind0(id4(tstind)), ind0(id5(tstind)), ...
17 -         ind0(id6(tstind)), ind0(id7(tstind)), ind0(id8(tstind)), ind0(id9(tstind)), ind0(id10(tstind)), ...
18 -         ind0(id11(tstind)), ind0(id12(tstind)), ind0(id13(tstind)), ind0(id14(tstind))];
19
20 - net.divideParam.trainInd = trnInd;
21 - net.divideParam.testInd = tstInd;
22
23 - % net = newff(trnInd, tstInd, 40,{'logsig'},'trainlm');
24 - % net.trainParam.mc = 0.9;
25 - % net.trainParam.lr = 0.05;
26 - % net.trainParam.epochs = 500;
27 - % net.trainParam.goal = 0.001;
28
29 - [net, tr] = train(net, datanormal, label);

```

Gambar 4. 30 Program Pelatihan Jaringan Propagasi Balik pada Percobaan 12

b. Output



Gambar 4. 31 Hasil dari Percobaan 12

Dapat dilihat dari gambar diatas bahwa dari 602 data huruf, data benar yang dapat terbaca ada 168 data dengan akurasi 80,4762%. Waktu dari pelatihan tersebut 00:03, MSE sebesar 0.302 dan gradient sebesar 1.00E-06.

Dari 12 percobaan yang dilakukan neuron yang menghasilkan hasil maksimal pada hidden layer 1 adalah 95 dan pada hidden layer 2 adalah 100 dan 95, maka peneliti akan menguji metode training yang dipakai. Berikut adalah hasil dari performa metode training yang dipakai:

Tabel 4. 1 Tabel Pengujian Metode Training

Ekstrasi Ciri	Jumlah hidden Layer	Neuron	Pelatihan 1	Pelatihan 2	Pelatihan 3	Metode Training	Epoch	Waktu Running	Akurasi
IoC & Mark Direction	1	95	logsig			trainscg	600	00.00.02	74,28%
IoC & Mark Direction	1	95	logsig			trainscg	500	00.00.01	72,85%
IoC & Mark Direction	1	95	logsig			trainrp	600	00.00.01	70,95%
IoC & Mark Direction	1	95	logsig			traingdx	600	00.00.01	61,90%
IoC & Mark Direction	1	95	logsig			traingd	600	00.00.01	7,61%
IoC & Mark Direction	1	95	logsig			trainlm	600	00.40.39	77,14%
IoC & Mark Direction	2	100&95	logsig	logsig	logsig	trainscg	600	00.00.03	80,47%
IoC & Mark Direction	2	100&95	logsig	logsig	logsig	trainrp	600	00.00.01	67,14%
IoC & Mark Direction	2	100&95	logsig	logsig	logsig	traingdx	600	00.00.02	71,42%
IoC & Mark Direction	2	100&95	logsig	logsig	logsig	traingd	600	00.00.02	7,14%

IoC & Mark Direction	2	100&95	logsig	logsig	logsig	trainlm	600	02.14.36	79,0476%
----------------------	---	--------	--------	--------	--------	---------	-----	----------	----------

Hasil optimal dari pengujian terhadap metode training yang dipakai menunjukkan bahwa metode yang paling cepat dan menghasilkan akurasi baik dalam melakukan pelatihan adalah metode training *trainscg*, *trainrp*, dan *trainlm* pada percobaan yang menggunakan 1 *hidden layer* karena menunjukkan hasil akurasi diatas 70%. Lalu pada percobaan yang menggunakan 2 *hidden layer* metode training yang optimal adalah *trainscg*, *traingdx*, dan *trainlm* dengan hasil akurasi diatas 70% persen. Metode training *trainlm* memang mengasilkan akurasi yang besar, tetapi dalam pengujiannya membutuhkan waktu yang lama.

4.4 Testing Data Tunggal

Pada penelitian ini dilakukan juga testing data tunggal dengan menggunakan 5 data pada setiap hurufnya, sehingga ada 70 data huruf yang dipakai untuk testing. Karena hasil akurasi terbaik didapatkan pada percobaan 12, maka pada pelatihan data tunggal ini akan menggunakan pola jaringan dan kombinasi ciri yang sama seperti pada percobaan 12. Berikut adalah tabel hasil testing data tunggal:

Tabel 4. 2 Tabel Hasil Testing Data Tunggal

No	Input Huruf	Output Huruf					Dikenali	Tidak kenali	Akurasi
		1	2	3	4	5			
1	G	G	G	G	G	G	5	0	100
2	N	N	N	N	N	N	5	0	100
3	D	D	N	G	N	G	1	4	20
4	R	R	R	G	G	R	3	2	60
5	M	M	M	M	B	G	3	2	60
6	B	G	B	B	M	G	2	3	40
7	S	S	S	S	S	S	5	0	100
8	NG	NG	NG	B	NG	NG	4	1	80
9	J	J	S	S	S	CH	1	4	20
10	CH	CH	J	J	CH	CH	3	2	60
11	K	K	K	K	J	M	3	2	60
12	T	T	D	T	D	D	2	3	40
13	P	H	R	D	P	S	1	4	20
14	H	H	H	H	H	H	5	0	100
Jumlah							43	27	
Rata-rata Akurasi									61,429

Hasil percobaan pengujian data tunggal data menunjukkan bahwa hasil yang didapat sangat beragam, total data yang dapat dikenali dengan baik 43 huruf sedangkan yang tidak dapat dikenali ada 24 huruf, dan rata-rata akurasi yang didapat sebesar 61,42857%. Hasil huruf Hangul yang dapat dikenali dengan baik adalah huruf G, N, S, H. Hal ini disebabkan data yang ditulis kurang baik dan kurang jelas, sehingga saat diubah menjadi citra hitam putih bagian tidak jelasnya tidak bisa dikenali.

4.5 Analisis Data

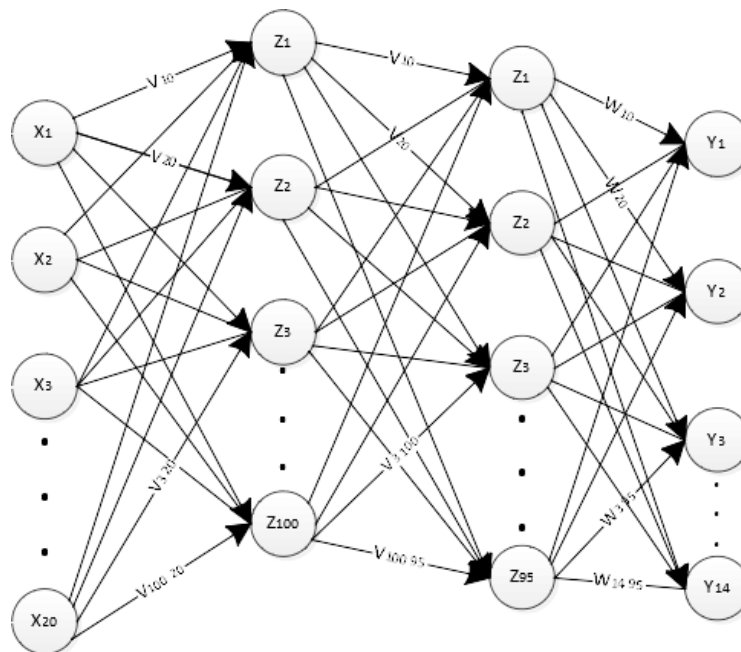
Setelah melakukan 12 kali percobaan dengan mengubah metode ekstraksi ciri, banyaknya jumlah *hidden* layer, jumlah neuron pada *hidden* layer, fungsi aktivasi pada *hidden* layer dan fungsi aktivasi pada jaringan terlihat adanya perubahan pada akurasi alat uji.

Hasil terbaik dapat dilihat pada percobaan 12, dengan banyak data benar/data yang dapat dibaca ada 168 data huruf dengan akurasi mencapai 80, 4762%. Dapat dilihat juga data huruf paling sedikit yang dapat dibaca ada pada percobaan 2, dimana akurasi yang didapat adalah 33,1633%. Dari

hasil tersebut dapat dikatakan bahwa alat uji yang telah dibuat dapat mengenali tulisan tangan huruf Hangul.

Percobaan mengubah metode ekstraksi ciri, banyaknya jumlah *hidden* layer, jumlah neuron pada *hidden* layer, fungsi aktivasi pada *hidden* layer dan fungsi aktivasi pada jaringan ternyata juga berpengaruh dalam meningkatkan dan mengurangi tingkat keakurasian dari alat uji.

Selain itu juga telah dilakukan pengujian data tunggal untuk melihat hasil dari pelatihan jaringan. Pengujian data tunggal ini menggunakan 70 data huruf. Dari 70 data, data yang dikenali dengan baik ada 43 huruf dengan rata-rata akurasi yang dihasilkan mencapai 61,42857%. Akurasi yang rendah disebabkan data yang digunakan kurang jelas dan kurang baik dalam penulisannya.



Gambar 4. 32 Arsitektur Jaringan yang Menghasilkan Akurasi Optimal

Dari hasil yang diperoleh pada penelitian ini tentang pengenalan tulisan tangan huruf Hangul dengan menggunakan jaringan saraf tiruan dengan menggunakan jaringan saraf tiruan propagasi balik diketahui bahwa:

1. Pada penelitian ini terdapat 168 tulisan tangan huruf Hangul yang dapat terbaca, maka dapat dikatakan bahwa algoritma *backpropagation* dapat mengenali tulisan tangan huruf Hangul.
2. Prosentasi akurasi dari beberapa percobaan dan fungsi yang telah dilakukan pada penelitian ini diperoleh akurasi terbaik sebesar 80,4762%.

Maka penelitian ini dapat dikatakan berhasil karena sudah memenuhi tujuan penelitian dengan akurasi sudah melebihi 70% prosentase keberhasilan.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan disimpulkan jika alat uji yang dibuat dengan menggunakan algoritma propagasi balik dapat digunakan untuk mengenali tulisan tangan huruf Hangul. Hal ini dapat dilihat dengan adanya beberapa huruf yang terbaca oleh alat uji yang menggunakan algoritma propagasi balik. Tingkat akurasi tertinggi didapat pada percobaan 12 dengan akurasi sebesar 80,4762% menggunakan kombinasi ciri 6 yang merupakan matriks kombinasi dari ciri *Intensity of Character*, *Mark Direction* horz, *Mark Direction* vert, *Mark Direction* dig1, *Mark Direction* dan dig2. Pola jaringan yang digunakan adalah jaringan dengan 2 hidden layer dan 1 lapisan keluaran, dengan jumlah unit neuron pada hidden layer 1 dan hidden layer 2 100, 95, dan pada lapisan keluaran jumlah unit neuronnya ada 14. Fungsi aktivasi yang digunakan pada hidden layer 1 dan hidden layer 2 adalah *logsig*, *logsig*, pada lapisan keluaran *logsig*, dan pada pelatihan jaringan menggunakan fungsi aktivasi *trainscg*. Epoch yang digunakan adalah 750 kali dengan waktu perhitungan 00:03, dan MSE sebesar 1.00E-06.

5.2 Saran

Dari hasil penelitian pengenalan tulisan tangan huruf Hangul ini, ada beberapa saran yang untuk memperbaiki kekurangan pada penelitian tersebut, antara lain:

1. Mencari data tulisan tangan huruf Hangul yang lebih banyak.
2. Tentukan Batasan-batasan dalam responden menulis tulisan tangan, jangan campurkan antara tulisan tangan huruf Hangul normal dan tulisan tangan huruf Hangul yang digayakan.
3. Mencari dan menggunakan metode ekstraksi ciri yang berbeda.

4. Mengubah susunan jaringan dengan menambah jumlah hidden layer, jumlah neuron pada hidden layer, fungsi aktivasi pada hidden layer, dan fungsi aktivasi pada jaringan.

DAFTAR PUSTAKA

- Keechul and Hang Joon Kim, 1999, "*On-line recognition of cursive Korean characters using graph representation*". Taegu, South Korea
- Handoyo, Erico Darmawan, Lydia Wiguna Susanto, 2011. *Penerapan Jaringan Syaraf Tiruan metode Propagasi Balik Dalam Pengenalan Tulisan Tangan Huruf Jepang Jenis Hiragana dan Katakana*. Universitas Kristen Manarata. Manarata University Pres : Bandung. ISSN 0216-4280
- Nugroho, Nicolaus Euclides Wahyu, 2016. "*Pengenalan Pola huruf Jepang Hiragana Menggunakan Algoritma Backpropagation*". Universitas Sanata Dharma, Yogyakarta.
- Handoyo, Kasih, 2017. "*Transliterasi Nama Jalan Beraksara Jawa*". Universitas Sanata Dharma, Yogyakarta.
- Gonzales, Rafael C. & Woods, Richard E., 2008. *Digital Image Processing*(3rd ed.). United States of America: Pearson Prentice Hall.
- Kristanto, Andri, 2004. "*Jaringan Syaraf Tiruan, Konsep Dasar, Algoritma dan Aplikasi*". Gava Media. Yogyakarta.
- Surinta, Olarik, 2010. "*Overview of Handwritten Thai Character Recognition*". Diakses dari <http://www.ai.rug.nl/~molarik/APSMeeting/09-07-2010%20Overview%20of%20Handwritten%20Thai%20Character%20Recognition.pdf> pada tanggal 29 september 2017.
- Fathia, Siska. 2013. "*Penerapan Jaringan Saraf Tiruan Dalam Pengenalan Tulisan Tangan Huruf Korea (Hangul) Menggunakan Metode Propagasi Balik*". Universitas Dian Nuswantoro, Semarang.
- Nasution, A. H., dan Prasetyawan, Y. 2008. *Perencanaan & Pengendalian Produksi. Edisi Pertama*. Graha Ilmu, Yogyakarta.
- Korea, Monthly Statistics. <http://kto.visitkorea.or.kr>. Diakses pada tanggal 11 Oktober 2016
- Hwa, A K, Cho H Y, Rura Ni, Agung Suray, Florian Hutagalung. 2008. "*Ayo Belajar Bahasa Korea dengan Bahasa Korea Terpadu Untuk Orang*

- Indonesia*". Korea Language Department, Korea Foundation. Secho-gu, Seoul.
- Kristanto, Andri, 2004. "*Jaringan Syaraf Tiruan, Konsep Dasar, Algoritma dan Aplikasi*". Gava Media. Yogyakarta.
- Sibagariang, S, 2016. "*Klasifikasi Citra Mammogram dengan Metode Ekstraksi Ciri Zoning Menggunakan Ssvm*". Universitas Sumatera Utara, Medan. Diakses dari <http://repository.usu.ac.id/xmlui/handle/123456789/57829> pada tanggal 5 Desember 2016.
- Sinaga, D Y, 2013. "*Pengenalan Pola Plat Nomor Kendaraan Bermotor Dengan Menggunakan PCA dan Metode Propagasi Balik*". Universitas Sanata Dharma, Yogyakarta.

LAMPIRAN

a. Form Pengenalan Tulisan Tangan Huruf Hangul

Form Tugas Akhir

“Pengenalan Tulisan Tangan Huruf Hangul Dengan Menggunakan Metode Jaringan Saraf Tiruan Propagasi Balik”

Huruf	Lembar Kerja Untuk Menulis Huruf Hangul	
	Lembar Kerja 1	Lembar Kerja 2
ㄱ		
ㄴ		
ㄷ		
ㄹ		
ㅁ		

日		
入		
〇		
丌		
𠂇		
𠂇		
𠂇		

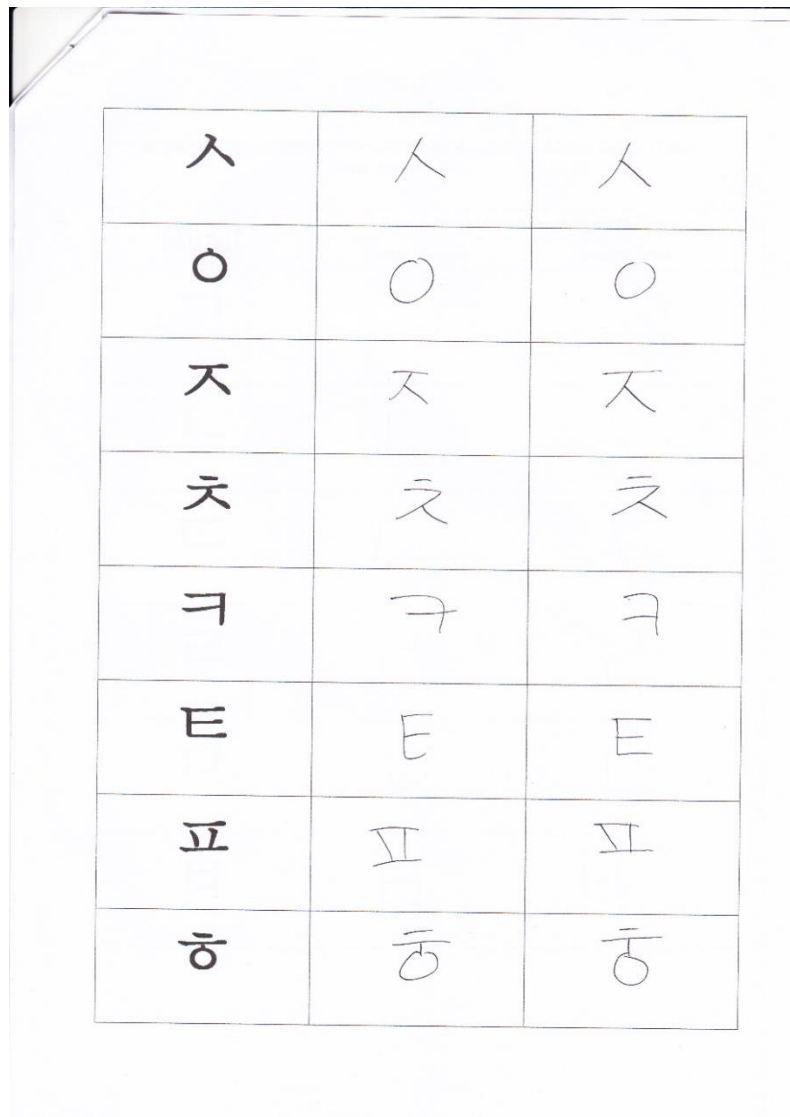
표		
능		

b. Contoh Data yang sudah di Digitalisasi

Form Tugas Akhir

“Pengenalan Tulisan Tangan Huruf Hangul Dengan Menggunakan Metode Jaringan Saraf Tiruan Propagasi Balik”

Huruf	Lembar Kerja Untuk Menulis Huruf Hangul	
	Lembar Kerja 1	Lembar Kerja 2
ㄱ	ㄱ	ㄱ
ㄴ	ㄴ	ㄴ
ㄷ	ㄷ	ㄷ
ㄹ	ㄹ	ㄹ
ㅁ	ㅁ	ㅁ
ㅂ	ㅂ	ㅂ



c. Contoh data yang sudah di-crop

