

**PENGENALAN POLA TULISAN TANGAN UNTUK BALITA
MENGUNAKAN JARINGAN SYARAF TIRUAN**

TESIS

**TEKNIK ELEKTRO KONSENTRASI SISTEM KOMUNIKASI DAN
INFORMATIKA**

**Diajukan untuk memenuhi persyaratan
Memperoleh gelar Magister Teknik**



**ZUNITA WULANSARI
NIM. 126060300111031**

**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2017**

**PENGENALAN POLA TULISAN TANGAN UNTUK BALITA MENGGUNAKAN
JARINGAN SYARAF TIRUAN**

TESIS



oleh:

Nama : Zunita Wulansari
 NIM : 126060300111031
 Program Magister : Teknik Elektro
 Minat : Sistem Komunikasi dan Informatika

Naskah tesis ini telah direvisi dan disetujui oleh dosen pembimbing
 Pada tanggal _____

Ketua,

Anggota,

M. Aziz Muslim, S.T., M.T., Ph.D.
 NIP. 19741203 200012 1 001

Dr. Ing. Onny Setyawati, S.T., M.T., M.Sc.
 NIP. 19740417 200003 2 007

Mengetahui,
 Ketua Program Studi Magister Teknik Elektro

Dr. Eng. Panca Mudjirahardjo, ST., MT.
 NIP. 19700329 200012 1 001

**JURUSAN TEKNIK ELEKTRO
 FAKULTAS TEKNIK
 UNIVERSITAS BRAWIJAYA
 MALANG
 2017**



JUDUL TESIS:

**PENGENALAN POLA TULISAN TANGAN UNTUK BALITA MENGGUNAKAN
JARINGAN SYARAF TIRUAN**

Nama Mahasiswa : Zunita Wulansari
NIM : 126060300111031
Program Studi : Program Magister Teknik Elektro
Minat : Sistem Komunikasi dan Informatika

KOMISI PEMBIMBING

Ketua : M. Aziz Muslim, S.T., M.T., Ph.D.
Anggota : Dr.Ing. Onny Setyawati, S.T., M.T., M.Sc.

TIM DOSEN PENGUJI

Dosen Penguji 1 : Dr. Ir. Sholeh Hadi Pramono, M.S.
Dosen Penguji 2 : Ir. Purnomo Budi Santoso, M.Sc., Ph.D.
Tanggal Ujian : 10 November 2017
SK Penguji : 1507 Tahun 2017



PERNYATAAN ORISINALITAS PENELITIAN TESIS

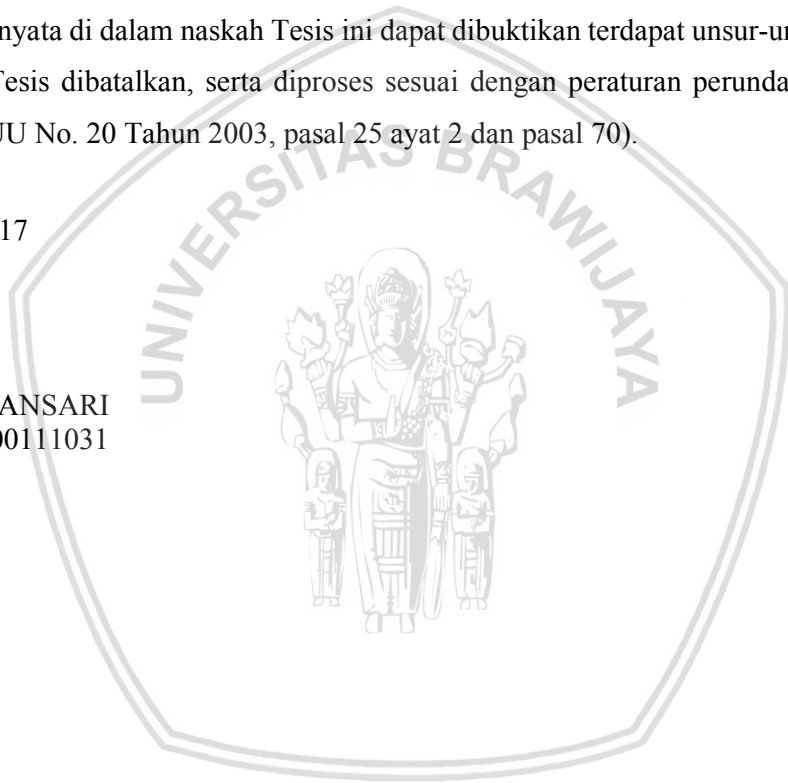
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Tesis ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Tesis ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Tesis dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, Juli 2017

Mahasiswa,

ZUNITA WULANSARI
NIM. 126060300111031





Kagem:

Garwo Kulo, M. Taofik Chulkaandi

Romo Simon

Ibunda Mardiyah

Putri Kulo, Anindya Niswara Ramadyani

Ugi Kagem:

Sederek Kulo Sedoyo

Rancang-Rancang

Sedoyo Dosen Teknik Elektro UB

RIWAYAT HIDUP

Zunita Wulansari, Blitar, 08 Juni 1986, anak dari Bapak Simon dan Ibu Mardiyah, MI Darul Huda Tanggung, SMP Negeri 1 Garum, MAN Negeri Wlingi tahun 2005. Studi S1 Teknik Elektro UIN Malang tahun 2010. Melanjutkan studi program Magister (S2) di Program Magister Teknik Elektro Jurusan Elektro Fakultas Teknik Universitas Brawijaya pada tahun 2012-2017.

Malang, Juli 2017

Penulis



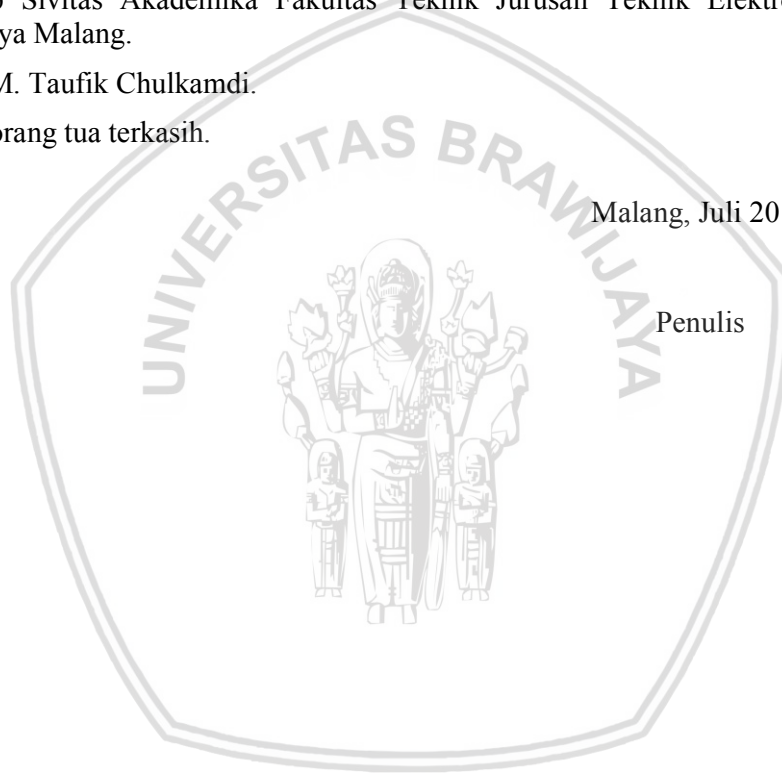
UCAPAN TERIMA KASIH

Dalam penyelesaian penelitian tesis ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Untuk itu, penulis menyampaikan ucapan terima kasih sebesar-besarnya dan setulusnya kepada:

1. M. Azis Muslim, ST., MT., Ph.D. selaku pembimbing utama yang senantiasa memberikan arahan dan garis besar di setiap bimbingan sehingga benar-benar menyalakan semangat penulis dalam penelitian tesis ini.
2. Dr.Ing. Onny Setyawati, ST., MT., M.Sc. selaku pembimbing kedua yang selalu aktif memberikan masukan-masukan teknis sehingga esensi penelitian tesis ini benar-benar muncul ke permukaan.
3. Segenap Sivitas Akademika Fakultas Teknik Jurusan Teknik Elektro Universitas Brawijaya Malang.
4. Suami M. Taufik Chulkamdi.
5. Kedua orang tua terkasih.

Malang, Juli 2017

Penulis



RINGKASAN

Zunita Wulansari, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juli 2017, *Pengenalan Pola Tulisan Tangan Untuk Balita Menggunakan Jaringan Syaraf Tiruan*, Dosen Pembimbing: M. Azis Muslim dan Onny Setyawati.

Komponen dari keterampilan berbahasa adalah keterampilan berbicara, keterampilan menyimak, keterampilan membaca dan keterampilan menulis. Anak usia 4 hingga 5 tahun lebih mudah belajar membaca dan menulis dengan menggunakan media gambar, cerita, lagu atau suara dibandingkan dengan menggunakan buku acuan yang di jual di toko-toko. Anak usia 4 hingga 5 tahun membutuhkan pendamping dalam belajar membaca dan menulis.

Kemampuan menulis permulaan merupakan bekal utama bagi anak untuk dapat memahami mata pelajaran yang diberikan di sekolah. Salah satu alternatif untuk membantu meningkatkan minat dan kemampuan anak dalam belajar membaca dan menulis adalah dengan menggunakan media pembelajaran berupa perangkat lunak. Perangkat lunak tersebut dapat memberikan pembelajaran menulis untuk anak prasekolah dengan tiga tingkat kesulitan berbeda yang bertujuan untuk mengasah kemampuan menulis anak.

Metode *backpropagation* menghasilkan prosentase tertinggi untuk mengenali tulisan, sehingga dipilih metode *backpropagation* untuk mengenali huruf dan angka dalam penelitian ini. Metode *backpropagation* difungsikan untuk mengenali pola tulisan sehingga diketahui *mean square error* dari tulisan tersebut. Selain hal tersebut metode *backpropagation* dapat melatih jaringan syaraf *multi-layer* sehingga dapat mempelajari representasi internal yang sesuai dan memungkinkan untuk mempelajari pemetaan *input* yang tidak sesuai terhadap *output*.

Proses pengenalan tulisan ini dilakukan dalam 5 tahap. Tahap pertama adalah tahap *preprocessing*. Pada tahap ini dilakukan penyederhanaan citra sehingga citra siap untuk dianalisis. Tahap kedua segmentasi citra, yaitu setiap huruf akan dikenali jika pendeteksian titik sudut menampilkan titik koordinat piksel untuk setiap sudut-sudut huruf. Tahap ketiga *normalization* hasil dari proses segmentasi yang dilakukan sebelumnya adalah posisi, lebar dan tinggi untuk masing-masing huruf. Setiap huruf hasil segmentasi akan dibaca oleh jaringan sebagai nilai input awal jaringan. Agar huruf tersebut dapat dibaca oleh jaringan maka huruf dinormalisasi menjadi citra *grayscale* 8 bit. Tahap keempat pengenalan karakter dengan *neural network backpropagation* dilakukan dengan melewati tahap pelatihan (*training*) terlebih dahulu kemudian dilanjutkan ke tahap pengenalan. Tahap kelima *Character Recognition* dilakukan ekstraksi terhadap bagian-bagian yang ada dalam pengenalan tulisan. Setelah pola tulisan dikenali karakternya, aplikasi akan mengekstraksi pengenalan pola tulisan sehingga tulisan dapat dikenali.

Prosentase keberhasilan *backpropagation* untuk mengenali tulisan huruf kapital A-Z adalah 61,4256% huruf kecil a-z adalah 83,0042% dan angka 0-9 adalah 75%. Hasil uji terapan kepada anak usia 4-5 tahun dilakukan selama 6 hari yang dilaksanakan di TK dan PAUD. Dari hasil uji terapan kepada anak usia 4-5 tahun didapatkan rata-rata 6,63 (dengan skala nilai 0 hingga 100) untuk peningkatan kemampuan siswa dari hari ke hari.

Kata kunci : tulisan tangan, pengenalan pola, balita, jaringan syaraf tiruan, *backpropagation*

SUMMARY

Zunita Wulandari, *Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, July 2017, Handwriting Recognition for Under 5 year-old Children using Artificial Neural Network, Academic Supervisor: M. Aziz Muslim and Onny Setyawati.*

The linguistic ability components consist of speaking, observing, reading, and writing. For 4 to 5 year-old children, it is easier to learn the ability of reading and writing using multimedia, e.g. pictures, stories, songs, and voices, rather than using textbooks. They need good teachers in learning process.

The early capability to write is a main requirement for children to understand subjects given in elementary school. One choice to help improve children's interest and capability in reading and writing could be by utilizing software as the learning media. Such software can then be developed to challenge children with several different levels in order to accelerate their writing ability.

The backpropagation method delivers the highest percentage in handwriting recognition among other methods, which makes it as the main option to implement in this research. This method is then used to recognize children's handwriting pattern and then to give scores to them based on the MSE (mean squared error). Furthermore, this method can train multi-layer neuron network so it learns the appropriate internal representation and make it possible to analyze mismatches in input to output mapping.

This handwriting recognition process will be done in five sequences. The first stage is preprocessing, which increases image clarity against noise and simplifies the image to make it ready to be analyzed. The second stage is image segmentation, which divide every character into segmented distinct images. The third stage is normalization of each previous segmentation result, and automatically defines position, width, and height of each character. Every segmented character image is read by the network as initial input value. The fourth stage is recognition, which is done by training process in backpropagation neural network previously, and this stage recognizes by comparing its network value with the current one, resulting the squared error. The final stage is the extraction process, which produce the calculation of MSE of entire handwriting sequence. Soon after the handwriting pattern recognized, the software then displays the scores of every child.

Keywords : *handwriting, pattern recognition, children under five, artificial neural network, backpropagation*

KATA PENGANTAR

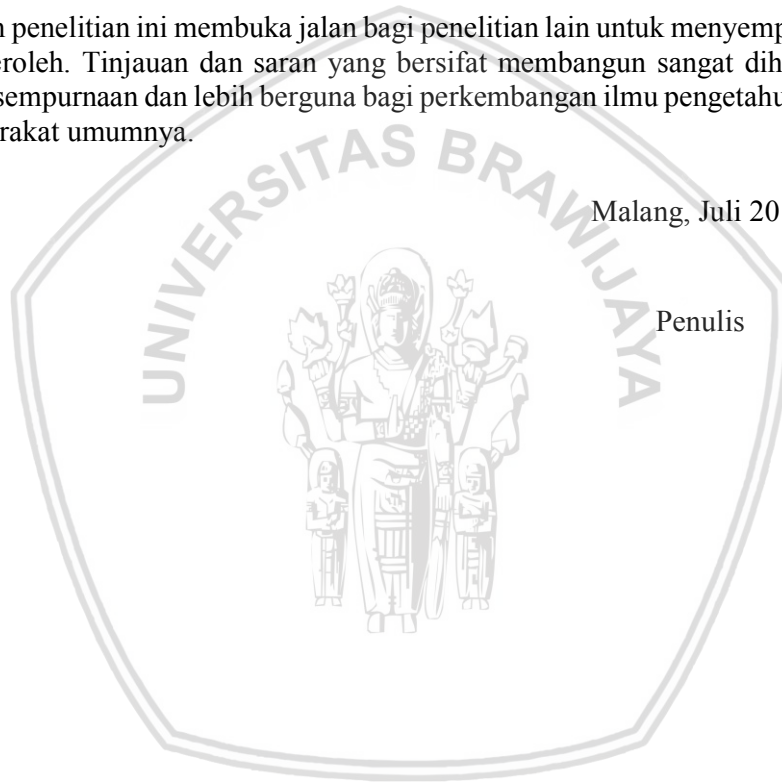
Kemampuan menulis permulaan merupakan bekal utama bagi anak untuk dapat memahami mata pelajaran yang diberikan di sekolah. Salah satu alternatif untuk membantu meningkatkan minat dan kemampuan anak dalam belajar membaca dan menulis adalah dengan menggunakan media pembelajaran berupa perangkat lunak. Penelitian ini mencoba untuk membuat sebuah pendekatan pengenalan tulisan tangan untuk balita dengan jaringan syaraf tiruan menggunakan metode backpropagation, yang dikemas dalam paket perangkat lunak yang dapat diakses oleh guru sekolah TK dan PAUD.

Penelitian ini juga diajukan sebagai tesis dalam rangka menyelesaikan studi di Program Magister Teknik Elektro di Universitas Brawijaya Malang bidang keahlian Sistem Komunikasi dan Informatika.

Diharapkan penelitian ini membuka jalan bagi penelitian lain untuk menyempurnakan hasil yang telah diperoleh. Tinjauan dan saran yang bersifat membangun sangat diharapkan demi peningkatan kesempurnaan dan lebih berguna bagi perkembangan ilmu pengetahuan khususnya dan bagi masyarakat umumnya.

Malang, Juli 2017

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
IDENTITAS PENGUJI TESIS	iii
PERNYATAAN ORISINALITAS PENELITIAN TESIS	iv
LEMBAR PERSEMBAHAN	v
RIWAYAT HIDUP	vi
UCAPAN TERIMA KASIH	vii
RINGKASAN	viii
SUMMARY	ix
KATA PENGANTAR	x
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
BAB II PENELITIAN RELEVAN DAN DASAR TEORI	5
2.1 Penelitian Terdahulu	5
2.2 Dasar Teori	8
2.2.1 Pendekatan Belajar Menulis	8
2.2.2 Dasar Teori Pengolahan Citra	8
2.2.3 Operasi Aras Pikel.....	11
2.2.4 Grayscaleing	13
2.2.5 Thresholding.....	14
2.2.6 Segmentasi	15
2.2.7 Edge Detection	15
2.2.8 Cropping.....	20
2.2.9 Normalisasi	21
2.2.10 Ekstraksi Ciri.....	22
2.2.11 Pengenalan Huruf Tulisan Tangan	23
2.2.12 Pra-Pemrosesan	24
2.2.13 Feature Extraction.....	25



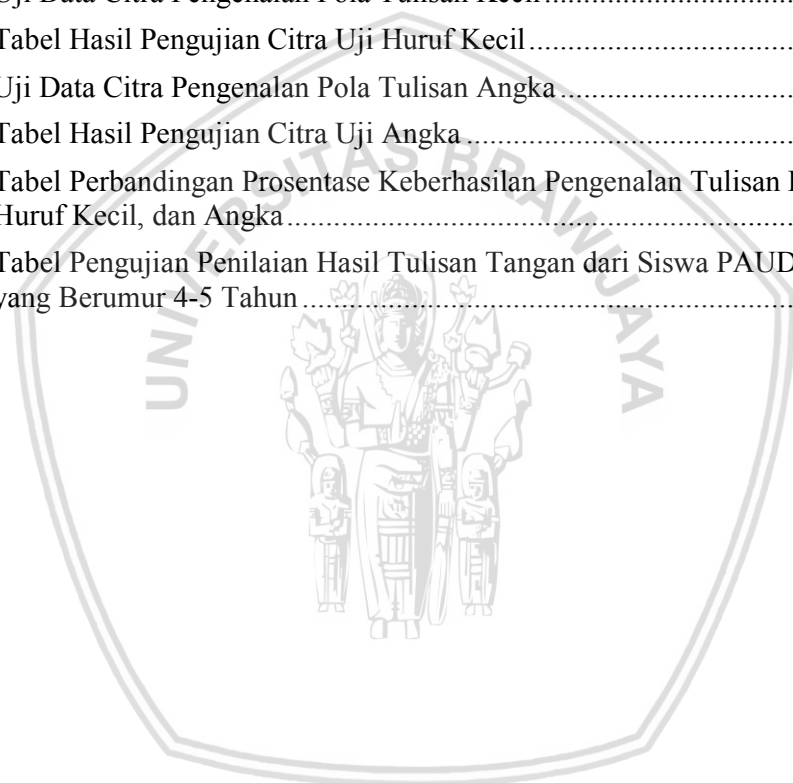
2.2.14	Optical Character Recognition (OCR).....	26
2.2.15	Artificial Neural Network	27
2.2.16	Back Propagation.....	30
2.2.17	Tahap Propagasi Balik.....	33
BAB III KERANGKA KONSEP PENELITIAN		37
3.1	Kerangka Konsep Berpikir	37
3.2	Rancangan Perangkat Lunak	38
3.3	Edge Detection dengan Operator Laplacian.....	40
3.4	Pengenalan Pola Tulisan Tangan Menggunakan Jaringan Saraf Tiruan Backpropagation	41
3.5	Pra-Pemrosesan Pengolahan Data Training dan Testing	42
3.6	Perancangan Jaringan Syaraf Tiruan.....	43
3.7	Desain Antarmuka.....	45
3.7.1	Desain Form Tampilan Menu Utama	45
3.7.2	Desain Form File	46
3.7.3	Desain Form View.....	46
3.7.4	Desain Form Proses dan Draw	46
3.8	Hipotesis	47
BAB IV METODE PENELITIAN.....		49
4.1	Langkah-Langkah Penelitian.....	49
4.2	Variabel dan Aturan Analisis.....	50
4.3	Tahap Pengumpulan dan Analisis Data.....	50
4.4	Perancangan Aplikasi Pengenalan Tulisan untuk Balita.....	51
4.5	Implementasi.....	51
4.6	Evaluasi	51
4.7	Pengambilan Kesimpulan.....	52
BAB V HASIL DAN PEMBAHASAN.....		53
5.1	Implementasi Aplikasi.....	53
5.1.1	Implementasi Proses Pembuatan dan Masukan Data	53
5.1.2	Thresholding.....	55
5.1.3	Proses Segmentasi	56
5.1.4	Cropping dan Normalisasi	58
5.1.5	Reduksi	59
5.1.6	Back Propagation.....	60
5.1.7	Training untuk Back Propagation.....	60
5.1.8	Testing untuk Back Propagation	61
5.1.9	Proses Pengenalan	65

5.1.10	Proses Penyimpanan	66
5.2	Uji Validasi Aplikasi	67
5.3	Uji Terapan Siswa Usia 4 Tahun hingga 5 Tahun	73
BAB VI KESIMPULAN DAN SARAN		75
6.1	Kesimpulan	75
6.2	Saran	76
DAFTAR PUSTAKA		



DAFTAR TABEL

No.	Nama Tabel	Halaman
Tabel 2.1	Karakter Tulisan Arab yang Dikenali.....	6
Tabel 2.2	Perbandingan Pengenalan Menggunakan Teknik Berbeda.....	6
Tabel 2.3	Perbandingan Pengenalan dalam Segala Bentuk Menggunakan Teknik Berbeda	6
Tabel 5.1	Uji Data Citra Pengenalan Pola Tulisan Huruf Kapital	67
Tabel 5.2	Tabel Hasil Pengujian Citra Uji Huruf Kapital dan Angka	68
Tabel 5.3	Uji Data Citra Pengenalan Pola Tulisan Kecil	70
Tabel 5.4	Tabel Hasil Pengujian Citra Uji Huruf Kecil	70
Tabel 5.5	Uji Data Citra Pengenalan Pola Tulisan Angka	71
Tabel 5.6	Tabel Hasil Pengujian Citra Uji Angka	72
Tabel 5.7	Tabel Perbandingan Prosentase Keberhasilan Pengenalan Tulisan Huruf Besar, Huruf Kecil, dan Angka	73
Tabel 5.8	Tabel Pengujian Penilaian Hasil Tulisan Tangan dari Siswa PAUD dan TK yang Berumur 4-5 Tahun	74



DAFTAR GAMBAR

No.	Nama Gambar	Halaman
Gambar 2.1	Ilustrasi proses pendeteksian tepian citra.....	16
Gambar 2.2	Contoh citra asal dan hasil pendeteksian tepiannya.	16
Gambar 2.3	Contoh citra asal dan hasil operasi pendeteksian tepi dengan filter $H_x, y = -11$	17
Gambar 2.4	Contoh hasil pendeteksian tepian menggunakan Metode Robert.	18
Gambar 2.5	Contoh hasil pendeteksian tepian menggunakan Metode Prewitt.....	18
Gambar 2.6	Contoh hasil pendeteksian tepian menggunakan Metode Sobel.	19
Gambar 2.7	Perbedaan pendeteksian tepian menggunakan Metode Sobel, Prewitt, Laplace, Roberts, dan Canny.....	20
Gambar 2.8	Contoh normalisasi citra dengan menggunakan pembesaran.	22
Gambar 2.9	Contoh ekstraksi ciri pada huruf “A”	23
Gambar 2.10	Klasifikasi tipe segmen.....	26
Gambar 2.11	Model dari JST.....	28
Gambar 2.12	Struktur dari multi-layer pada JST.....	31
Gambar 3.1	Kerangka konsep berpikir pada penelitian ini.....	37
Gambar 3.2	Diagram pengenalan tulisan tangan secara umum.	39
Gambar 3.3	Diagram umum pra-pemrosesan citra.....	40
Gambar 3.4	Diagram alir deteksi tepi.....	41
Gambar 3.5	Diagram alir pra-pemrosesan citra untuk training dan testing.....	42
Gambar 3.6	Diagram umum proses pelatihan.....	43
Gambar 3.7	Diagram umum proses pengenalan.....	44
Gambar 3.8	Diagram alir proses pelatihan.....	44
Gambar 3.9	Rancangan menu utama.	46
Gambar 3.10	Tampilan proses percobaan pengenalan tulisan tangan.....	47
Gambar 4.1	Diagram alir pengenalan pola tulisan tangan untuk balita.....	50
Gambar 5.1	Tampilan form utama.....	54
Gambar 5.2	Form input data referensi dengan menulis langsung.....	54
Gambar 5.3	Form input data referensi melalui font.	54
Gambar 5.4	Contoh tulisan tangan pada kanvas aplikasi.....	55
Gambar 5.5	Contoh sebuah citra grayscale yang akan dikenai operasi thresholding.....	55
Gambar 5.6	Hasil operasi thresholding citra dari Gambar 5.5.....	56
Gambar 5.7	Citra yang akan dikenai proses segmentasi.....	57
Gambar 5.8	Hasil segmentasi dari citra pada Gambar 5.7.....	57
Gambar 5.9	Hasil segmentasi huruf dari citra pada Gambar 5.8.....	58



Gambar 5.10	Hasil operasi cropping dan normalisasi huruf A dari Gambar 5.8.....	59
Gambar 5.11	Form training backpropagation.....	61
Gambar 5.12	Hasil testing backpropagation ke-1 pengenalan huruf.....	62
Gambar 5.13	Hasil testing backpropagation ke-3 pengenalan huruf.....	62
Gambar 5.14	Hasil testing backpropagation ke-1 pengenalan angka.....	63
Gambar 5.15	Hasil testing backpropagation ke-2 pengenalan angka.....	63
Gambar 5.16	Hasil testing backpropagation ke-3 pengenalan angka.....	64
Gambar 5.17	Hasil testing backpropagation ke-4 pengenalan angka.....	64
Gambar 5.18	Hasil testing backpropagation ke-5 pengenalan angka.....	65
Gambar 5.19	Hasil testing backpropagation ke-5 pengenalan angka.....	65
Gambar 5.20	Hasil recognize citra tulisan tangan.....	66
Gambar 5.21	Menu penyimpanan pada aplikasi.....	67



BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai insan yang akan meneruskan bangsa, anak-anak harus dipersiapkan mulai dari dini supaya memiliki karakter, kemampuan, dan kepedulian kepada perkembangan bangsa dan negara. (Izhar, 2013)

Pendidikan yang diajarkan untuk anak mulai anak lahir sampai dewasa yaitu ketrampilan berbahasa yang didapat dari proses hubungan yang teratur dan dalam lingkup yang sederhana. Setiap aspek keterampilan saling berhubungan dan merupakan kesatuan yang utuh. Dimulainya proses belajar saat kecil, anak belajar menyimak bahasa terlebih yang terdapat di sekelilingnya, kemudian anak belajar berbicara, selanjutnya mereka akan belajar membaca kemudian menulis. Komponen dari keterampilan berbahasa adalah keterampilan berbicara, memperhatikan, menulis, dan membaca. Dalam penelitian tersebut dibahas secara khusus aplikasi yang bisa membantu belajar menulis untuk anak umur pra-sekolah. (Raharjo & Ahyani, 2011)

Berdasarkan penelitian Limanto, anak umur 4 sampai dengan 5 tahun akan lebih tertarik belajar menulis dan membaca menggunakan multimedia, yaitu dengan gambar, cerita, lagu dan juga suara bila dibandingkan dengan menggunakan buku panduan yang di jual di toko-toko. Anak umur 4 sampai dengan 5 tahun membutuhkan pendamping dalam proses belajar menulis dan belajar membaca. Wawancara yang dilakukan kepala sekolah Taman Kanak-kanak Al-Hidayah. Hasil dari wawancara tersebut menunjukkan siswa mampu mengenali huruf, rangkaian kata dalam bentuk hafalan, namun sudah terbentuk dalam kalimat siswa menjadi kesulitan dalam menganalisisnya disinilah peranan penting keluarga untuk mendampingi anak selama dalam proses belajar menulis dan belajar membaca. (Limanto, 2008)

Kemampuan menulis merupakan bekal utama untuk anak yang akan sekolah agar dapat memahami pelajaran pada saat di bangku sekolah. Cara untuk dapat membantu meningkatkan kemampuan dan kemauan anak untuk belajar menulis dan belajar membaca yaitu dengan menggunakan media pembelajaran dalam bentuk *software* (perangkat lunak). (Hendrawan, *et. al.*, 2012)

Software (perangkat lunak) yang dirancang dapat membantu proses belajar menulis yang difungsikan untuk anak sebelum memasuki bangku sekolah dasar dengan 3 tingkat kesulitan berbeda, dimana tujuannya untuk menguji kemampuan anak dalam proses belajar menulis anak.

Lukas melakukan penelitian untuk mengenali data gambar pada numerik arab, metode yang digunakan adalah *backpropagation*. Hasil penelitian lukas tersebut memiliki prosentase keberhasilan 92%. (Lukas, 2004)

Peneliti lain menggunakan metode *backpropagation* untuk mengenali tulisan tangan adalah Verma di mana prosentase keberhasilan yang didapatkan adalah 91%. (Verma, *et. al.*, 2001)

Selanjutnya Saputro melakukan penelitian pengenalan citra karakter tulisan tangan dengan metode *backpropagation* mendapatkan prosentase keberhasilan 93,64%. (Saputro, 2012)

Kesulitan utama yang dihadapi anak umur 4 sampai dengan 5 tahun saat belajar menulis dan belajar membaca adalah kurangnya minat akibat metode pengajaran yang konvensional dan kurang mampu memotivasi minat belajar. Diharapkan media pembelajaran berupa perangkat lunak akan mampu untuk meningkatkan minat anak dan tidak membosankan.

Metode *backpropagation* menghasilkan prosentase tertinggi untuk mengenali tulisan, sehingga dipilih metode *backpropagation* untuk mengenali angka dan huruf dalam penelitian ini. Metode *backpropagation* difungsikan untuk menguji pola tulisan sehingga diketahui *mean square error* dari tulisan tersebut. Selain hal tersebut metode *backpropagation* dapat melatih jaringan syaraf *multi-layer* sehingga dapat mempelajari representasi internal yang sesuai dan memungkinkan untuk mempelajari pemetaan *input* yang tidak sesuai terhadap *output*.

Proses pengenalan pola tulisan tangan ini dilakukan dalam 5 tahap. Tahap pertama adalah tahap *preprocessing*. Tahap pertama adalah menyederhanakan citra agar *image* siap untuk dianalisis. Tahap kedua adalah segmentasi *image*, yaitu mengenali setiap huruf jika pendeteksian titik sudut menunjukkan titik koordinat *pixel* pada setiap sudut-sudut huruf. Tahap ketiga adalah *normalization*, hasil pada proses segmentasi *image* yaitu dapat diketahui letak, tinggi dan lebar pada masing-masing huruf. Masing-masing huruf hasil dari segmentasi dibaca oleh JST sebagai nilai masukan awal jaringan. Supaya huruf tersebut terbaca oleh JST maka huruf dinormalisasikan menjadi citra *grayscale* 8 bit. Tahap keempat

adalah pengenalan terhadap karakter dengan menggunakan JST, tahap keempat ini dilakukan pelatihan (*training*) kemudian dilanjutkan tahap pengenalan. Tahap kelima adalah *Character Recognition* dimana pada tahap ekstraksi terhadap bagian-bagian yang terdapat didalam pengenalan tulisan. Apabila pola tulisan sudah dikenali karakternya, maka akan mengekstraksi pengenalan pola tulisan sehingga tulisan dapat dikenali.

Metode *backpropagation* digunakan pada penelitian ini karena efisiensinya dalam melakukan peremajaan bobot tiap neuron ketika proses *training*, sehingga diharapkan proses ini terjadi dalam waktu yang lebih singkat dari pada metode yang sudah ada.

1.2 Rumusan Masalah

Rumusan masalah yang sesuai untuk pembelajaran menulis dan membaca anak usia 4 tahun sampai dengan 5 tahun adalah sebagai berikut:

1. Bagaimana membangun sistem untuk proses belajar menulis dan membaca anak usia 4 tahun sampai dengan 5 tahun.
2. Bagaimana menerapkan pengolahan citra menggunakan parameter citra biner dengan format citra 1 bit berupa tulisan tangan anak usia 4 tahun sampai dengan 5 tahun.
3. Bagaimana memfungsikan jaringan saraf tiruan menggunakan metode *backpropagation* untuk menguji akurasi data pengenalan tulisan tangan di mana pelatihan dilakukan maksimal 10.000 kali.

1.3 Tujuan Penelitian

Jika didasarkan dari latar belakang dan rumusan masalah, maka tujuan penelitian ini adalah:

1. Untuk membangun sistem untuk proses belajar anak usia 4 tahun sampai dengan 5 tahun.
2. Untuk menerapkan pengolahan citra menggunakan parameter citra biner dengan format citra 1 bit berupa tulisan tangan anak usia 4 tahun sampai dengan 5 tahun.
3. Untuk menguji akurasi data pengenalan tulisan tangan di mana pelatihan dilakukan maksimal 10.000 kali dengan menggunakan jaringan saraf tiruan menggunakan metode *backpropagation*.

1.4 Batasan Masalah

Jika didasarkan dari rumusan masalah dan tujuan dari penelitian ini, agar fokus pada pembahasan maka dijelaskan ruang lingkup untuk batasan masalah dalam penelitian sebagai berikut:

1. *File* citra yang difungsikan ialah *file* citra *bitmap* (*.bmp) *grayscale* 8 bit yang dikonversi ke citra biner 1 bit.
2. *File image* dalam bentuk tulisan tangan ditulis menggunakan bolpen yang selanjutnya dilakukan *scanning* hingga mendapatkan suatu *image* atau *image* tulisan tangan yang dicatat secara langsung di dalam program.
3. Huruf yang dikenali dalam program adalah huruf kapital dari A sampai Z dan huruf kecil dari a sampai z. Angka yang dikenali adalah angka dari 0 sampai dengan 9.
4. Jarak spasi antar huruf minimal 10 piksel, jarak spasi antar kata minimal 22 piksel.
5. *Input* data pengenalan tulisan tidak dalam bentuk tulisan sambung.

1.5 Manfaat Penelitian

Sistem yang dibangun diharapkan dapat memberi manfaat yang lebih baik untuk anak yang sedang belajar menulis dan membaca, sehingga pada saat menggunakan komputer anak tidak bermain game saja akan tetapi juga belajar. Karena pada saat ini untuk memasuki kelas 1 sekolah dasar anak-anak diwajibkan sudah dapat membaca, menulis dan berhitung. Bagi orang tua yang kurang mampu dan tidak memiliki komputer dapat menggunakan komputer yang terdapat di sekolah. Pada PAUD Iklas Nurani terdapat 2 komputer yang dapat dioperasikan sebagai perangkat pembelajaran.

BAB II

PENELITIAN RELEVAN DAN DASAR TEORI

Pada Bab ini dijelaskan tentang studi pustaka yang memuat penelitian yang relevan mengenai topik pengenalan pola tulisan tangan, bab ini juga memuat dasar teori pendukung dari tema penelitian yang dilakukan.

2.1 Penelitian Terdahulu

Evaluasi kinerja *Handwritten Tamil* (tulisan India) menggunakan metode *Octal Graph Conversion and the Metrics* menunjukkan hasil efisiensi yang baik. Efisiensi keseluruhan system tersebut memiliki prosentasi keberhasilan 82%. Hasil penelitian menunjukkan hasil yang optimal sehingga dapat digunakan untuk mengenali karakter skrip India. (Kannan, 2008)

Pada tahun 2004, Lukas melakukan penelitian untuk mengenali data gambar pada numerik arab, dengan menggunakan metode *backpropagation*. Hasil yang diperoleh dari data tersebut secara umum jaringan telah mampu mengenali masing-masing karakter dengan presentase tingkat pengenalan yang berbeda. Perbedaan tersebut tergantung dari nilai keluaran dan nilai ambang batasnya. Dari 90 data yang diuji, 83 data berhasil dikenali dengan baik, dengan nilai keluaran yang berbeda-beda, sehingga dapat diketahui keberhasilan aplikasi 92%. (Lukas, 2004)

Dunia pendidikan yang semakin hari semakin canggih membuat kita berfikir untuk mengembangkan suatu perangkat lunak yang dapat membantu kita dalam mengetahui segala hal, termasuk dalam hal mengenali tulisan. Banyak peneliti melakukan penelitian untuk mengenali tulisan tangan, misalnya menggabungkan dua metode yaitu metode *backpropagation* dan radial basis *function* untuk mengenali kata pada tulisan tangan. Hasil prosentase penggabungan dua metode untuk mengenali kata pada tulisan tangan adalah 91%. Hasil prosentase yang sangat yang sangat memuaskan. (Verma, *et. al.*, 2001)

Sebuah kerangka terpadu berdasarkan inferensi Bayesian diusulkan oleh Cheung pada tahun 1998 untuk pemodelan, pencocokan, dan mengklasifikasikan pola yang menunjukkan variasi yang besar dalam bentuk karakter tulisan. *Deformable model-based* dimasukkan sebagai komponen penting dalam kerangka Bayesian. Pengenalan karakter tulisan tangan yang digunakan untuk menyediakan kesalahan uji dan realistis untuk kerangka

deformablemodels. Untuk tulisan tangan angka dari NIST SD-1 dataset, dengan hanya menggunakan 23 prototipe, telah mencapai akurasi 94,7% pada 11.791 contoh uji. (Cheung, 1998)

Ekstraksi fitur diagonal diusulkan Pradeep untuk mengenali tulisan tangan *off-line* dari alfabet inggris. Dua pengenalan menggunakan 54 fitur dan 69 fitur yang dipilih untuk membangun sistem *Neural Network Recognition*. Hasil penelitian mengungkapkan bahwa 69 fitur memberikan akurasi pengenalan yang lebih baik dari 54 fitur untuk semua jenis ekstraksi fitur. Dari hasil tes itu diidentifikasi bahwa metode diagonal ekstraksi fitur menghasilkan akurasi tertinggi dengan pengenalan 97,8 % untuk 54 fitur dan 98,5 % untuk 69 fitur. Metode diagonal ekstraksi fitur diverifikasi menggunakan sejumlah citra uji. (Pradeep, *et. al.*, 2011)

Penelitian yang dilakukan oleh Lawgali pada tahun 2011 meneliti pengenalan karakter tulisan arab. Selain DCT (*discrete cosine transform*), teknik DWT (*discrete wavelet transform*) juga digunakan dalam penelitian tersebut. Selain itu, penelitian tersebut membandingkan metode DCT dan DWT untuk pengenalan tulisan arab. (Lawgali, *et. al.*, 2011)

Tabel 2.1
Karakter Tulisan Arab yang Dikenali

ط	ص	س	ر	د	ح	ب	ا
ي	و	ه	ن	م	ل	ف	ع

Sumber : Lawgali, *et. al.*, 2011.

Tabel 2.2
Perbandingan Pengenalan Menggunakan Teknik Berbeda

Ukuran	DCT	DWT
64×64	94,87%	59,81%
128×128	96,56%	57,00%
256×256	92,82%	39,62%

Sumber : Lawgali, *et. al.*, 2011.

Tabel 2.3
Perbandingan Pengenalan dalam Segala Bentuk Menggunakan Teknik Berbeda

Ukuran	DCT	DWT
64×64	79,87%	40,71%
128×128	78,82%	27,85%
256×256	73,82%	10,57%

Sumber : Lawgali, *et. al.*, 2011.



Pada Tabel 2.1 ditunjukkan karakteristik tulisan arab yang dikenali. Hasil prosentase keberhasilan menggunakan DCT dan DWT terdapat pada Tabel 2.2 dan Tabel 2.3.

Dalam penelitian Saputro (2006) dibangun suatu sistem yang dapat melakukan pengenalan citra karakter tulisan tangan menggunakan *JST Resilient Backpropagation* (RPROP). Data yang diambil berupa karakter numeral dari 13 orang dengan menggunakan media kertas. Dari 13 orang tersebut 10 orang menuliskan karakter sebanyak 2 kali, di mana penulisan pertama digunakan untuk data pelatihan dan penulisan kedua digunakan untuk data pengujian. Tiga orang sisanya hanya menuliskan karakter sebanyak 1 kali, karakter tulisan tangan dari 2 orang digunakan untuk menambah data pelatihan sedangkan karakter tulisan tangan dari 1 orang lainnya digunakan untuk menambah data pengujian. Karakter-karakter tulisan tangan tersebut selanjutnya diubah menjadi *image* digital dengan menggunakan *scanner* dan setiap karakter disimpan sebagai satu *file* citra, sehingga didapat 230 buah *file* citra. Dari data tersebut 120 citra digunakan untuk data pelatihan dan 110 *imagelainnya* digunakan untuk data pengujian. Sistem yang dibangun menggunakan *graphical user interface* (GUI) karena sistem mudah digunakan. Dari hasil percobaan didapatkan parameter maksimal yaitu JST dengan 80 neuron *hidden*, Faktor Naik sebesar 1,2 dan Faktor Turun sebesar 0,7. Dengan penggunaan parameter-parameter tersebut sistem dapat mengenali karakter tulisan tangan dengan tingkat generalisasi tertinggi sebesar 93,64%. (Saputro, 2006)

Selanjutnya Gaurav Jain dan Jason Ko pada tahun 2008 melakukan penelitian untuk mengenali angka pada tulisan tangan. Selain menggunakan *Principal Component Analysis* (PCA), metode *k-Nearest Neighbors Algorithm* (k-NN) juga digunakan dalam penelitian itu. Hasil prosentase keberhasilan penelitian tersebut adalah 78.4%. Hasil prosentase yang diperoleh masih lebih tinggi menggunakan metode *backpropagation* dan *radial basis function*. (Jain & Ko, 2008)

Mengenali kata dalam tulisan tangan juga dilakukan oleh Koerich. Perbedaan dari penelitian Verma adalah pada metode yang digunakan. Penelitian tersebut menggunakan *Hidden Markov Models* (HMM) dan *Segmental Neural Network* (SNN). Hasil prosentase keberhasilan penelitian tersebut adalah 71%. Hasil yang menurun dibandingkan dengan penelitian yang dilakukan Jain. (Koerich, *et. al.*, 2006)

Berdasarkan penelitian terdahulu dan literatur tersebut, maka dalam penelitian ini akan dikembangkan sistem pengenalan pola tulisan tangan yang diimplementasikan dalam media

touch screen misalnya komputer atau *laptop touch screen* untuk belajar menulis anak usia dini (4 sampai dengan 5 tahun) dengan menggunakan jaringan saraf tiruan *Backpropagation*.

2.2 Dasar Teori

Kemampuan yang terpenting untuk anak-anak adalah dapat menulis dan membaca. Karena kemampuan menulis dan membaca adalah bekal utama untuk anak pada saat mulai masuk Sekolah dasar karena untuk memahami dan mengerjakan tugas dari guru semua harus dibekali pengetahuan menulis dan membaca. (Stephens, 2004)

Kemampuan menulis dan membaca dapat mulai dan diasah sejak umur sebelum memasuki sekolah dasar. Cara belajar menulis dan membaca dengan bantuan alat tulis pensil dan penggunaan kertas tidak lagi diminati oleh anak-anak. Hal ini disebabkan karena anak-anak lebih senang jika belajar menggunakan multimedia, yaitu penggabungan tulisan, gambar, pewarnaan, musik, lagu, suara, dan mendengarkan cerita jika dibandingkan dengan cara belajar konvensional. (Izhar, 1998)

Belajar menggunakan perangkat lunak dapat memicu semangat anak untuk belajar menulis dan membaca. Ini terbukti dengan penelitian Macaruso & Adelaide yang membangun aplikasi untuk proses belajar anak untuk menulis dan membaca. (Macaruso & Adelaide, 2008)

Penggunaan *software* bantu dengan bantuan komputer akan memudahkan penerimaan dan pemahaman anak-anak, karena komputer mampu memuat materi dalam bentuk multimedia. Dalam hal ini, anak-anak akan merasa sedang melakukan permainan, walaupun dalam kenyataannya sedang melakukan aktivitas belajar.

2.2.1 Pendekatan Belajar Menulis

Permainan menulis melalui persiapan menulis dan membentuk tulisan. Tahap memulai menulis adalah tahap melakukan atau kemauan yang melatih sensor motorik anak, seperti dengan menggambar, mewarna, meronce dan masih banyak yang lainnya. Tahap yang selanjutnya adalah membentuk tulisan, yang dilakukan dengan proses membentuk tulisan yaitu mencoret, tulisan horisontal, menulis acak dan menulis nama bilangan. (Mudjito, 2007)

2.2.2 Dasar Teori Pengolahan Citra

Citra merupakan himpunan titik-titik dalam permukaan dua dimensi yang memiliki warna atau intensitas berbeda. Secara matematis, citra adalah fungsi kontinuitas (fungsi

berkesinambungan) terhadap kecerahan cahaya pada permukaan dua dimensi. Pada permukaan citra yang bersifat pasif, dengan adanya cahaya yang memberikan intensitas penyinaran dengan warna tertentu akan dipantulkan oleh permukaan tersebut sesuai dengan panjang gelombang yang dipantulkan, sementara sebagian lainnya diserap. Pada permukaan citra yang bersifat aktif, cahaya dihasilkan oleh titik-titik permukaan dipancarkan (dihamburkan) dengan warna tertentu. Hamburan cahaya dengan warna atau panjang gelombang dan intensitas tertentu tersebut dibelokkan dengan bantuan lensa dan ditangkap oleh sensor optis, seperti kamera (menggunakan sensor elektronik atau kimiawi) atau mata (menggunakan sensor biologis seperti retina) dalam bentuk rekaman citra bayangan (*image*), kemudian diinterpretasikan pada bagian pemrosesan.

Penurunan mutu pada citra yang telah ditangkap antara lain adanya *noise* (derau), kontras warna yang berlebihan, *over exposure*, kabur, dan lain-lain. Pada citra yang mengalami penurunan mutu, informasi yang ditangkap akan menjadi berkurang sehingga sulit untuk diinterpretasikan. Proses peningkatan kembali (atau perbaikan) mutu citra perlu dilakukan sehingga akan memenuhi syarat untuk dilakukan ekstraksi informasi. Proses demikian disebut dengan pengolahan citra digital (*digital image processing*).

Proses pengolahan citra merupakan sebuah usaha untuk meningkatkan mutu citra. Penggunaan perangkat komputer dengan perangkat lunak tertentu sering dilakukan dalam usaha tersebut. Perbaikan atau restorasi citra dilakukan dengan menonjolkan informasi yang akan diekstrak. (Munir, 2004:3)

Secara lebih rinci pada ranah matematis, citra digital adalah fungsi $f(x, y)$ yang mana x dan y merupakan koordinat spasial dua dimensi, dan f merupakan tingkat kecerahan atau intensitas pada koordinat (x, y) . (Gonzalez & Woods, 2002)

$$f(x, y) = i(x, y) \cdot r(x, y) \dots\dots\dots(2-1)$$

Dengan:

- f = fungsi intensitas cahaya
- i = intensitas cahaya datang
- r = derajat kemampuan pantul titik
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)



Persamaan (2-1) menunjukkan fungsi intensitas cahaya pada koordinat (x, y), yang mana untuk indeks kolom dimulai dari kiri ke kanan, sedangkan untuk indeks baris dimulai dari atas ke bawah. Secara baku, citra digital dengan ukuran N baris dan M kolom bisa dinyatakan dengan matriks seperti pada Persamaan (2-2).

$$f(x, y) = \begin{Bmatrix} f_{0,0} & f_{1,0} & \cdots & f_{M-1,0} \\ f_{0,1} & f_{1,1} & \cdots & f_{M-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{0,N-1} & f_{1,N-1} & \cdots & f_{M-1,N-1} \end{Bmatrix} \dots\dots\dots (2-2)$$

Dengan:

- f = fungsi intensitas cahaya
- N = ukuran baris
- M = ukuran kolom
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)

Pada citra digital, setiap titik disebut dengan *pixel*, yaitu singkatan dari *picture element*, yang dalam Bahasa Indonesia telah diserap dan disebut dengan piksel. Setiap piksel dalam citra digital bisa memiliki jangkauan nilai f (fungsi intensitas cahaya) yang beragam disesuaikan dengan format citra tersebut.

1. Pada citra hitam putih, atau disebut dengan *binary image*, nilai f adalah nilai bilangan bulat 0 atau 1, di mana 0 melambangkan warna hitam dan 1 melambangkan warna putih. Citra ini menggunakan 1 bit (seperdelapan byte – 2¹) guna merepresentasikan setiap pikselnya.
2. Pada citra keabuan, atau disebut dengan *grayscale image*, nilai f adalah nilai bilangan bulat mulai dari 0 hingga 255, di mana 0 melambangkan warna hitam sempurna hingga 255 yang melambangkan warna putih sempurna. Nilai di antara 0 dan 255 merupakan nilai dengan derajat keabuan tertentu, yang semakin gelap (menuju ke hitam) ketika nilai berkurang, dan semakin terang (menuju ke putih) ketika nilai bertambah. Citra ini membutuhkan 8 bit (1 byte – 2⁸) untuk representasi setiap pikselnya.



3. Pada citra warna lengkap, atau disebut dengan *true color image*, atau *RGB image*, nilai f adalah nilai bilangan bulat mulai dari 0 hingga 16.777.215. Nilai f tersebut sebenarnya terdiri dari pecahan komponen warna cahaya merah, warna cahaya hijau, dan warna cahaya biru, yang masing-masing bernilai bilangan bulat dengan jangkauan mulai dari 0 hingga 255. Cahaya merah disimbolkan sebagai R (dari kata *red*), sementara cahaya hijau disimbolkan dengan G (dari kata *green*), dan cahaya biru disimbolkan menggunakan B (dari kata *blue*). Nilai 0 dari tiap komponen merupakan nilai di mana tidak ada cahaya komponen sama sekali, hingga 255 yang merupakan nilai di mana cahaya komponen (merah, hijau, atau biru) bernilai maksimum. Setiap komponen warna disebut dengan *channel*, yang masing-masing membutuhkan 8 bit (1 byte – 2^8) ruang penyimpanan, sehingga untuk ketiga elemen membutuhkan 24 bit (3 byte – 2^{24}) untuk representasi setiap pikselnya.

Selain citra warna lengkap, masih terdapat citra warna RGB yang membutuhkan lebih sedikit ruang penyimpanan, seperti RGB555 (15 bit), *indexed image* (8 bit), tetapi sangat jarang dipakai karena kualitas yang tidak optimal dan penyimpanan yang membuang ekstra 1 bit. Ada pula citra warna lengkap yang membutuhkan 32 bit untuk representasi setiap pikselnya, yaitu RGBA (*red-green-blue-alpha*) di mana *alpha* digunakan untuk menyimpan data transparansi.

2.2.3 Operasi Aras Piksel

Operasi yang dilakukan pada setiap piksel tunggal pada citra digital disebut dengan operasi aras piksel. Pada operasi jenis ini, akses dilakukan langsung menuju piksel individual pada citra, kemudian data setiap piksel tersebut dikenai operasi tertentu (baik bersifat linier ataupun non-linier), dan hasil operasi tersebut dikembalikan lagi pada tiap piksel yang bersangkutan yang membentuk citra baru. Persamaan (2-3) menunjukkan operasi matematis pada aras piksel.

$$f'(x, y) = O_{pix}(f(x, y)) \dots \dots \dots (2-3)$$

Dengan:

- f' = fungsi intensitas cahaya hasil operasi
- f = fungsi intensitas cahaya asli
- O_{pix} = operasi aras piksel, linier maupun non-linier
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)



y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)

Pada operasi aras piksel, terdapat tiga kategori operasi, yaitu berdasarkan intensitas, berdasarkan geometri, dan berdasarkan keduanya. Operasi aras piksel yang berdasarkan intensitas adalah transformasi h pada intensitas u suatu piksel menuju nilai baru v. Persamaan (2-4) menunjukkan transformasi tersebut.

$$v = h(u) \cdot u; v \in \{0, L\} \dots \dots \dots (2-4)$$

Dengan:

- v = nilai intensitas piksel hasil transformasi
- h = operator transformasi
- u = nilai intensitas piksel awal

Pada Persamaan (2-5) ditunjukkan contoh operasi *thresholding*, yaitu bila terdapat citra keabuan yang dipetakan menjadi citra hitam putih.

$$f'(x, y) = \begin{cases} 0; & f(x, y) < T \\ 1; & f(x, y) \geq T \end{cases} \dots \dots \dots (2-5)$$

Dengan:

- f' = fungsi intensitas cahaya hasil operasi
- f = fungsi intensitas cahaya asli
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)
- T = nilai ambang

Pada contoh Persamaan (2-5) tersebut, piksel yang memiliki nilai intensitas kurang dari T dijadikan bernilai 0 (hitam), sedangkan piksel yang memiliki nilai intensitas lebih dari atau sama dengan T dijadikan bernilai 1 (putih).

Operasi aras piksel yang berdasarkan geometri merupakan operasi yang berkenaan dengan pemindahan lokasi atau koordinat piksel tersebut menuju koordinat baru. Contoh dari operasi jenis ini meliputi penggeseran (*translation*), pemutaran (*rotation*), dan pembesaran (*scaling*). Pada Persamaan (2-6) ditunjukkan contoh fungsi untuk melakukan operasi penggeseran.



$$f'(x, y) = f(x + T_x, y + T_y) \dots\dots\dots(2-6)$$

Dengan:

- f' = fungsi intensitas cahaya hasil operasi
- f = fungsi intensitas cahaya asli
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)
- T_x = nilai operasi penggeseran pada sumbu x
- T_y = nilai operasi penggeseran pada sumbu y

Sedangkan kategori ketiga adalah gabungan operasi aras piksel berdasarkan intensitas dan geometri. Kategori ini merupakan penggabungan operasi yang harus dilakukan secara berurutan antara intensitas dengan geometri, sesuai dengan kebutuhan mana dulu yang dilakukan.

2.2.4 Grayscale

Proses grayscale adalah suatu proses untuk mengubah citra warna lengkap (RGB, *true color image*) menjadi citra keabuan (*grayscale*). Operasi yang dilakukan pada proses ini adalah operasi aras piksel berdasarkan intensitas. Normalnya pada operasi ini, nilai piksel RGB diambil rata-ratanya kemudian diberikan pada nilai piksel *grayscale* yang baru. Namun karena persepsi dan kepekaan mata manusia pada masing-masing warna dasar tidak sama, maka diperlukan pembobotan pada tiap *channel* warna.

$$f_{gray}(x, y) = 0,299 \cdot f_{red}(x, y) + 0,587 \cdot f_{green}(x, y) + 0,114 \cdot f_{blue}(x, y) \dots\dots\dots(2-7)$$

Dengan:

- f_{gray} = fungsi intensitas cahaya hasil operasi
- f_{red} = fungsi intensitas cahaya merah awal
- f_{green} = fungsi intensitas cahaya hijau awal
- f_{blue} = fungsi intensitas cahaya biru awal
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)



Sesuai dengan Persamaan (2-7), kepekaan relatif mata manusia terhadap warna merah, adalah 0,299, sementara kepekaan relatif mata manusia terhadap warna hijau adalah 0,587, dan kepekaan relatif mata manusia terhadap warna biru adalah 0,114. Oleh karena itu, dengan intensitas keabuan yang sama, warna merah akan terlihat lebih “gelap” dibandingkan warna hijau, dan warna biru akan jauh lebih “gelap” dibandingkan warna merah.

2.2.5 Thresholding

Thresholding citra adalah salah satu metode untuk melakukan transformasi citra dari citra keabuan menjadi citra hitam putih. Salah satu cara paling sederhana untuk melakukan *thresholding* adalah dengan mengganti nilai piksel dengan warna hitam jika nilai asalnya kurang dari nilai ambang tertentu, selain itu nilai piksel itu diganti dengan warna putih.

Sebagaimana dijelaskan di atas, operasi ini melakukan mentransformasikan nilai intensitas keabuan setiap piksel menjadi dua, yaitu putih dan hitam. Dalam citra tujuan, setiap piksel yang ada pada citra awal dipetakan menggunakan dua kemungkinan nilai, yaitu satu atau nol, dengan fungsi seperti pada Persamaan (2-8).

$$f'(x, y) = \begin{cases} 0; & f(x, y) < T \\ 1; & f(x, y) \geq T \end{cases} \dots\dots\dots (2-9)$$

Dengan:

- f' = fungsi intensitas cahaya hasil operasi
- f = fungsi intensitas cahaya asli
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)
- T = nilai ambang

Pada Persamaan (2-8), f' merupakan citra hitam putih yang terbentuk dari citra keabuan f , di mana T menyatakan nilai batas ambang. Nilai T menentukan kualitas citra hitam putih yang akan dihasilkan melalui operasi ini.

Proses *thresholding* dibagi menjadi dua, yaitu *global thresholding* dan *locally adaptive thresholding*. Nilai ambang T diambil (ditentukan sebagai variabel masukan) menjadi nilai global untuk melakukan konversi pada *global thresholding*. Informasi yang hilang pada teknik ini relatif besar karena menggunakan nilai T tunggal. Namun untuk *locally adaptive*

thresholding akan menjadi lebih akurat karena untuk setiap blok-blok lokal kecil nilai T dibuat berbeda.

2.2.6 Segmentasi

Segmentasi adalah pembagian daerah dalam suatu citra digital yang mana setiap area terbagi menurut kemiripan atributnya. Segmentasi dilakukan guna membagi suatu citra menuju beberapa segmen yang memiliki fitur berbeda. Daerah yang terpisah biasanya merupakan objek yang berbeda. Proses segmentasi ini merupakan bagian yang sangat penting dan biasanya merupakan proses yang rumit dalam pemrosesan citra digital.

Banyak metode yang dapat digunakan untuk segmentasi citra seperti pengambungan (*thresholding*), penandaan komponen terhubung (*connected component labeling*), segmentasi berbasis *clustering*, pendeteksian tepi menggunakan filter Sobel, Prewitt, ataupun yang lainnya. Namun secara umum metode ini dibagi dalam tiga kategori, yaitu berdasar klasifikasi, berdasar tepian, dan berdasar area, yang dijelaskan sebagai berikut:

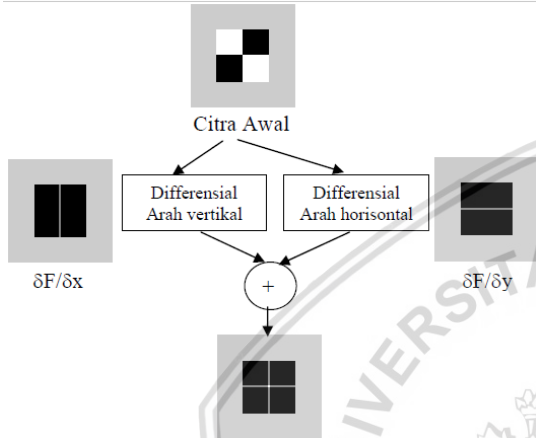
1. Segmentasi berdasarkan klasifikasi (*classification based segmentation*), yaitu segmentasi yang didasarkan pada kemiripan nilai piksel. *Thresholding* merupakan suatu teknik yang termudah dan sederhana. Histogram digunakan sebagai dasar pada *global thresholding*. Sedangkan pada *locally adaptive thresholding*, posisi citra akan menjadi dasar pembagian tersebut, yang mana dipecah-pecah menjadi beberapa kluster yang bersifat dinamis dan saling melengkapi.
2. Segmentasi berdasarkan tepian (*edge based segmentation*), yaitu proses segmentasi guna menghasilkan tepian objek sehingga dalam citra akan jelas pemisahan antar objek.
3. Segmentasi berdasarkan area (*region based segmentation*), yaitu segmentasi yang dilakukan dengan dasar himpunan piksel dengan keserupaan warna, tekstur, atau derajat keabuan mulai dari satu piksel ke piksel lain yang bertetangga. (Rudy Adipranata, 2005)

2.2.7 Edge Detection

Edge detection (pendeteksian tepian) merupakan operasi pengenalan fitur citra sehingga akan menghasilkan tepian yang menggambarkan objek berbeda dalam citra, dengan tujuan antara lain:

4. Melakukan penandaan bagian tertentu yang lebih rinci dalam citra.
5. Memperbaiki rincian citra yang tidak jelas, yang biasanya terjadi karena *noise* atau media pengambilan citra yang kurang sempurna.

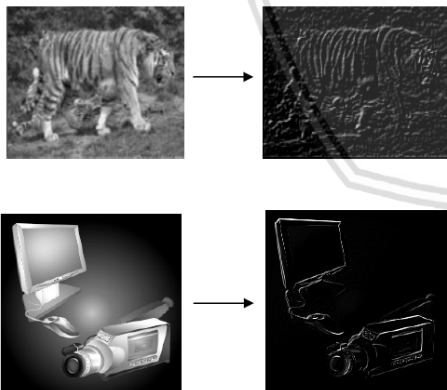
Sebuah piksel akan disebut sebagai tepi jika piksel tersebut memiliki perbedaan cukup tinggi dengan piksel di sebelahnya (piksel tetangganya). Gambar 2.1 menggambarkan cara menghasilkan tepian suatu citra.



Gambar 2.1 Ilustrasi proses pendeteksian tepian citra.

Sumber : Munir, 2004.

Gambar 2.2 merupakan dua contoh hasil proses pendeteksian tepian citra yang didasarkan pada model diferensial pada Gambar 2.1.



Gambar 2.2 Contoh citra asal dan hasil pendeteksian tepiannya.

Sumber : Munir, 2004.

Ditunjukkan pada Gambar 2.2 bahwa hasil deteksi tepi merupakan garis-garis tepian citra seperti pada proses penggambaran kartun. Tepian dari objek dalam citra adalah bagian yang mana piksel aslinya memiliki perbedaan cukup tinggi dengan piksel tetangganya.

Tepian objek dalam citra digital dapat dihasilkan menggunakan *high pass filter* (HPF), dengan ciri seperti pada Persamaan (2-9).

$$\sum_y \sum_x H(x, y) = 0 \dots\dots\dots(2-9)$$

Dengan:

H = *filter mask*

x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)

y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)

Jika diketahui sebuah filter $H(x, y) = \begin{bmatrix} -1 & 1 \end{bmatrix}$, maka Gambar 2.3 adalah contoh citra asal dan hasil operasi dari filter tersebut.



Gambar 2.3 Contoh citra asal dan hasil operasi pendeteksian tepi dengan filter $H(x, y) = \begin{bmatrix} -1 & 1 \end{bmatrix}$.

Sumber : Munir, 2004.

A. Metode Robert

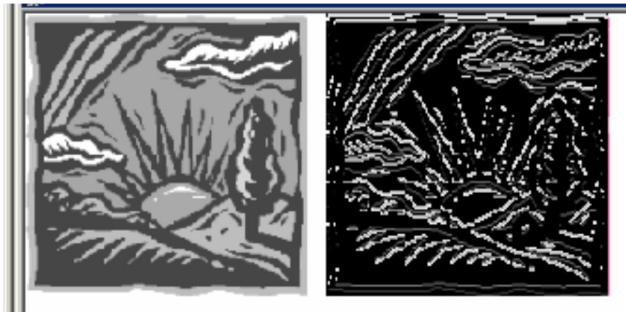
Metode Robert merupakan metode diferensial seperti yang telah ditunjukkan pada Persamaan (2-9), dengan menggunakan diferensial arah x dan arah y, yang dilengkapi proses transformasi menuju ke citra hitam putih. Teknik konversi ke citra hitam putih yang terbaik adalah membuat distribusi warna hitam dan putih menjadi merata. Teknik ini mirip dengan teknik DPCM (*Differential Pulse Code Modulation*).

Metode ini menggunakan kernel filter seperti pada Persamaan (2-10).

$$H = \begin{bmatrix} -1 & 1 \end{bmatrix}; V = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \dots\dots\dots(2-10)$$

Gambar 2.4 merupakan contoh citra yang dikenai operasi pendeteksian tepian menggunakan Metode Robert.





Gambar 2.4 Contoh hasil pendeteksian tepian menggunakan Metode Robert.
 Sumber : Munir, 2004.

B. Metode Prewitt

Metode Prewitt ini adalah pendeteksian tepian yang merupakan perluasan dari Metode Robert sebelumnya, yaitu penggunaan filter HPF yang memiliki sebuah angka nol sebagai penyangga. Teknik ini menggunakan prinsip pada fungsi Laplacian yang sering disebut fungsi pembangkitan HPF. Metode Prewitt menggunakan kernel filter seperti pada Persamaan (2-11).

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}; V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \dots\dots\dots (2-11)$$

Berikutnya pada Gambar 2.5 ditunjukkan contoh citra yang dikenai operasi pendeteksian tepian menggunakan Metode Prewitt.



Gambar 2.5 Contoh hasil pendeteksian tepian menggunakan Metode Prewitt.
 Sumber : Munir, 2004.

C. Metode Sobel

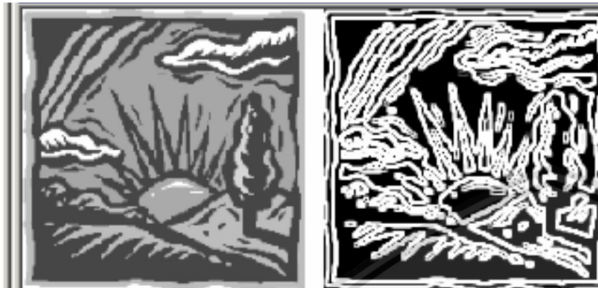
Metode Sobel adalah perluasan dari Metode Robert, hanya saja metode ini menggunakan filter HPF dengan sebuah angka nol sebagai penyangga. Fungsi Laplacian dan Gaussian merupakan prinsip yang diambil dan disebut fungsi pembangkitan HPF. Metode ini memiliki keuntungan, yaitu terdapat kemampuan menekan angka derau sebelum pendeteksian tepian dilakukan.



Metode Sobel menggunakan kernel filter seperti pada Persamaan (2-12).

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \dots\dots\dots(2-12)$$

Berikutnya pada Gambar 2.6 ditunjukkan contoh citra yang dikenai operasi pendeteksian tepian menggunakan Metode Sobel.



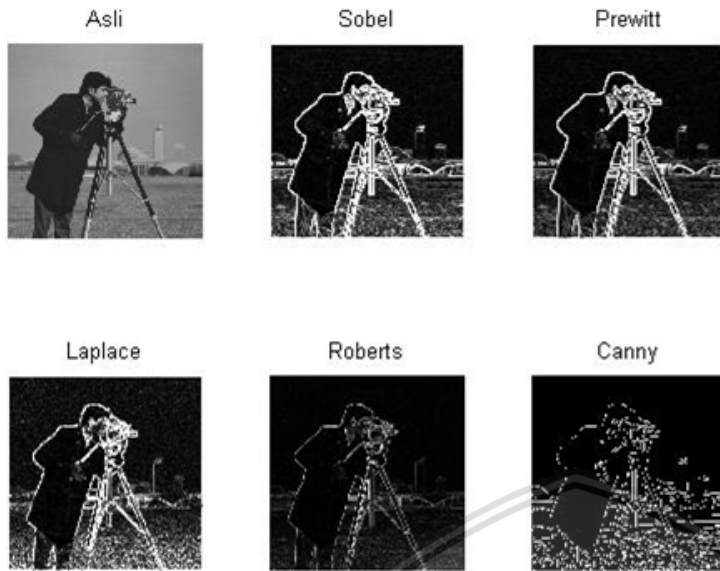
Gambar 2.6 Contoh hasil pendeteksian tepian menggunakan Metode Sobel.
 Sumber : Munir, 2004.

D. Metode Laplacian

Metode Laplacian merupakan cara yang digunakan untuk menyelesaikan persamaan diferensial. Operator ini menebalkan bagian tepi ke segala arah, atau disebut bersifat *omnidirectional*. Tetapi operator ini memiliki kelemahan, yakni peka terhadap *noise*, ketebalan yang dihasilkan ganda, dan tidak memiliki kemampuan pendeteksian arah dari tepian. (Gonzalez dan Woods, 2002)

Pada Gambar 2.7 ditunjukkan perbedaan pendeteksian tepian menggunakan metode Sobel, Prewitt, Laplace, Robert, dan Canny.





Gambar 2.7 Perbedaan pendeteksian tepian menggunakan Metode Sobel, Prewitt, Laplace, Roberts, dan Canny.

Sumber : Gonzales & Woods, 2002.

Dari hasil Gambar 2.7 tersebut, citra hasil operasi Metode Canny adalah citra yang paling bagus karena kejelasan deteksi tepinya dan *noise* yang lebih sedikit. Sedangkan citra hasil yang paling buruk adalah hasil operasi Metode Laplace karena terdapat banyak *noise*. Dalam penelitian ini Metode Laplacian merupakan teknik yang akan digunakan. Konsep dasar dari operator Laplacian ialah menggunakan *second derivative* (turunan orde kedua) untuk mengetahui pinggiran yang dihasilkan dari perubahan kontras derajat keabuan, untuk diteruskan dengan pencarian titik pelana (*zero crossing*) guna menjadi tempat garis tepian (*edge*).

2.2.8 Cropping

Untuk memperoleh citra yang diharapkan, maka dilakukan pemotongan bagian-bagian dalam citra yang disebut dengan *cropping*. *Cropping* citra berguna untuk memperbesar atau memperjelas suatu objek atau citra agar tampak lebih rinci. Dapat berguna juga untuk melihat suatu vegetasi pada citra. Ukuran dari pemotongan citra akan berubah disesuaikan dengan dimensi citra asal. Dalam melakukan *cropping* suatu citra diperlukan penentuan koordinat dari citra tersebut, dan menentukan batasan dari citra yang akan dipotong. Untuk menghitung suatu luasan objek yang akan dihitung maka ukuran *cropping* dapat dirumuskan melalui Persamaan (2-13) hingga (2-14).

$$W' = (x_r - x_l) + 1 \dots \dots \dots (2-13)$$



$$H' = (y_b - y_t) + 1 \dots\dots\dots (2-14)$$

Dengan:

- W' = lebar citra baru hasil *cropping*
- H' = tinggi citra baru hasil *cropping*
- x_l = lokasi x awal
- x_r = lokasi x akhir
- y_t = lokasi y awal
- y_b = lokasi y akhir

2.2.9 Normalisasi

Secara umum, pengorganisasian data ke dalam tabel-tabel guna memenuhi kebutuhan pengguna dalam suatu organisasi disebut proses normalisasi. Pada ranah pengolahan citra, normalisasi merupakan proses transformasi citra menuju bentuk normal yang disesuaikan dengan kebutuhan. Secara umum normalisasi bertujuan untuk:

1. Menekan redundansi data;
2. Menekan kompleksitas data; dan
3. Mempermudah proses modifikasi data.

Namun dalam pengolahan citra, normalisasi bertujuan agar hasil pengolahan citra sebelumnya dapat disesuaikan dengan kebutuhan sehingga lebih mudah untuk pemrosesan selanjutnya.

Jika suatu citra telah dinormalisasi, maka dapat dilakukan proses pembesaran (*scaling*) agar nilai ciri berada pada suatu *range* tertentu. Persamaan (2-15) hingga (2-16) menunjukkan fungsi untuk pembesaran (*scaling*) citra.

$$x' = S_x \cdot x \dots\dots\dots (2-15)$$

$$y' = S_y \cdot y \dots\dots\dots (2-16)$$

Dengan:

- x' = nilai x baru setelah pembesaran
- y' = nilai y baru setelah pembesaran
- S_x = konstanta pembesaran pada sumbu x
- S_y = konstanta pembesaran pada sumbu y



- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)

Pada Gambar 2.8 ditunjukkan normalisasi citra yang dilakukan dengan pembesaran citra.



Gambar 2.8 Contoh normalisasi citra dengan menggunakan pembesaran.

Sumber : Ahmad & Kartika, 2005.

2.2.10 Ekstraksi Ciri

Dalam pengolahan citra, ekstraksi ciri merupakan proses untuk mengambil nilai-nilai pola yang terdapat pada citra secara keseluruhan, berdasarkan beberapa klasifikasi nilai piksel terhadap tetangganya. Fungsi untuk melakukan ekstraksi ciri dapat ditunjukkan pada Persamaan (2-17).

$$f'(x, y) = \begin{cases} 1; & f(x, y) = 0 \\ -1; & f(x, y) \neq 0 \end{cases} \dots\dots\dots (2-17)$$

Dengan:

- f' = fungsi intensitas cahaya hasil ekstraksi ciri
- f = fungsi intensitas cahaya asli
- x = indeks kolom titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya kolom dikurangi satu)
- y = indeks baris titik citra (elemen bilangan bulat mulai dari nol hingga banyaknya baris dikurangi satu)

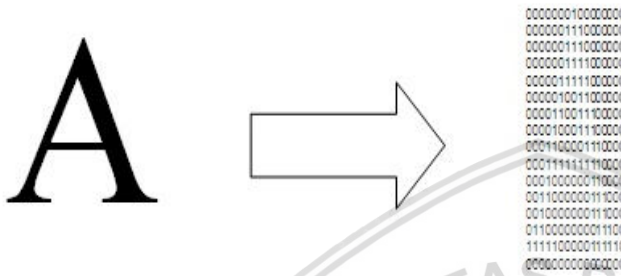
Setiap piksel hitam akan diberi kode satu dan setiap piksel putih diberi kode minus satu. Warna putih dan hitam akan dikonversi menjadi 1 dan -1 (berurutan) karena hal ini disesuaikan dengan fungsi sigmoid yang akan digunakan pada jaringan syaraf tiruan, yaitu fungsi Sigmoid Bipolar yang memiliki rentang nilai dari -1 sampai 1.

Secara matematis, setiap ekstraksi ciri merupakan ekstraksi kode dari vektor berdimensi n yang selanjutnya disebut sebagai vektor ciri. Dengan pemrosesan citra dan *analytic*



technique komponen ini akan dicari dan berfungsi guna melakukan perbandingan citra dalam satu kelompok.

Terdapat tiga klasifikasi pada ekstraksi ciri ini, yaitu: *low level feature* (menggunakan isi visual, seperti tekstur dan warna), *middle level feature* (menggunakan segmentasi sebagai dasar wilayah citra), dan *high level feature* (menggunakan dasar informasi semantik yang terdapat pada citra asal). (Maarques & Furht, 2002)



Gambar 2.9 Contoh ekstraksi ciri pada huruf “A”.

Sumber : Ahmad & Kartika, 2005.

Pada Gambar 2.9 merupakan sebuah contoh ekstraksi ciri pada huruf “A”.

2.2.11 Pengenalan Huruf Tulisan Tangan

Salah satu kemampuan yang dimiliki oleh komputer adalah kemampuan dalam menerima dan melakukan interpretasi masukan berupa tulisan tangan. Masukan ini bisa bersumber dari dokumen kertas, layar sentuh, foto, ataupun perangkat lainnya. Kemampuan ini disebut dengan pengenalan tulisan tangan.

Sebelumnya, pengenalan tulisan menggunakan tulisan pada dokumen kertas yang dipindai dan dimasukkan ke komputer. Dalam hal ini, ciri yang terkandung dalam citra hasil pindaian perlu untuk diekstrak.

Ada dua cara untuk mengenali tulisan tangan, yaitu pengenalan secara *offline* maupun *online*. Penulisan tangan *online* memproses informasi temporal atau dinamis dari penulisan goresan (*stroke*) sehingga pengenalan dilakukan secara bersamaan pada saat penulisan dilakukan. Sedangkan pengenalan secara *offline* memerlukan proses mediasi yaitu pemindaian kertas dokumen yang menjadi asal tulisan tangan tersebut. Pada proses *offline*, proses penulisan harus diselesaikan terlebih dahulu sebelum dokumen tersebut disa dipindai dan pada akhirnya dilakukan proses pengenalan. (Santosh, 2009)

Tulisan tangan terdiri atas kumpulan goresan (*stroke*) berdasarkan urutan waktu. *Stroke* merupakan sekumpulan titik koordinat berdasarkan urutan waktu yang terekam dari satu

urutan *pen down*, *pen move*, dan *pen up*. Jumlah goresan untuk setiap karakter berbeda. Dalam hal ukuran, tingkat variasi yang dimiliki tulisan tangan cukup tinggi. (Jaeger, 2003)

Proses pengenalan tulisan tangan secara umum meliputi pra-proses, *feature extraction*, dan klasifikasi segmen, algoritma pembelajaran dan pengenalan pola untuk meningkatkan hasil dari pengenalan.

2.2.12 Pra-Pemrosesan

Pra-pemrosesan adalah tahapan yang dilakukan sebelum terjadi tahapan utama terhadap citra yang akan dikenali, yaitu segmentasi citra. Pra-pemrosesan yang digunakan dalam penelitian ini meliputi penghalusan citra, penipisan citra, dan penentuan titik penting.

A. Penghalusan Citra

Penghalusan citra (*image smoothing*) merupakan langkah untuk menghilangkan derau (*noise*). Citra warna lengkap diubah terlebih dahulu menjadi citra hitam putih sebelum menghilangkan *noise* yang ada pada suatu citra. Saat melakukan proses perubahan atau kompresi, akan muncul *noise* dalam citra. Kualitas citra akan turun pada intensitas *noise* yang tinggi maupun rendah. Selain itu, *noise* akan mengakibatkan hilangnya sebagian rincian informasi citra. Beberapa teknik *filtering* bisa dilakukan untuk menghilangkan *noise*. Salah satunya adalah *median filtering* yang menggunakan informasi kecerahan piksel pada *filtering window* serta melakukan penggantian piksel yang berlokasi pada *window center* dengan nilai *median*. Pada proses penghalusan ini, piksel tetangga akan diperhitungkan nilai statistiknya. (Gonzalez & Woods, 2002)

Ide dasar untuk teknik *filtering* menganggap bahwa *noise* merupakan nilai yang muncul paling jarang dalam suatu populasi. Anggapan ini menyebabkan teknik *filtering* dalam kondisi terbaiknya adalah pada citra asli yang memiliki variasi intensitas atau warna yang cenderung homogen.

B. Penipisan Citra

Dalam pra-pemrosesan citra digital, penipisan citra digunakan untuk objek piksel yang diinginkan pada citra hitam putih di mana semua garis direduksi sehingga ketebalannya menjadi satu piksel. Untuk menghilangkan piksel terluar dapat dilakukan dengan teknik *iterative boundary erosion process* sehingga menghasilkan piksel kerangka (*skeleton*).

C. Penentuan Titik Penting

Titik penting yang dimaksud adalah piksel dalam kerangka yang mempunyai tetangga dengan warna latar depan tidak sama dengan dua, jenis dari titik penting ini adalah titik ujung dan titik percabangan. Titik ujung merupakan titik yang menghubungkan tiga segmen atau lebih.

Algoritma yang digunakan untuk mendeteksi dan mengklasifikasikan titik penting adalah dengan cara menghitung transisi dari warna latar depan ke latar belakang pada tetangga dari titik tersebut. Jika jumlah transisi lebih kecil dari dua maka titik tersebut diklasifikasikan sebagai titik cabang. (Kusumoputro & Emanuel, 2001)

Secara garis besar algoritma penentuan titik penting dilakukan dengan pencarian *depth first search* (DFS) sebagai berikut: (Pavlidis, 1992)

1. Langkah 1, yaitu menelusuri citra secara horisontal, mulai dari kiri atas hingga kanan bawah.
2. Langkah 2, yaitu melakukan pengecekan pada jumlah tetangga yang dimiliki setiap piksel p , di mana $(x_1, x_2, x_3, \dots, x_8)$ adalah nilai untuk kedelapan tetangga dari p , pengurutan dimulai dari tetangga sebelah kanan dan seterusnya mengikuti arah jarum jam.
3. Langkah 3 adalah jika p memiliki setidaknya tiga tetangga, maka kategorikan sebagai titik percabangan kemudian simpan ke dalam matriks percabangan.
4. Langkah 4, yaitu jika p hanya memiliki satu tetangga, maka kategorikan sebagai titik ujung, kemudian simpan ke dalam matriks ujung.

Citra masukan yang telah dilakukan proses penipisan citra selanjutnya akan dicari koordinat titik ujung dan titik percabangan.

2.2.13 Feature Extraction

Feature extraction adalah fase penting dalam identifikasi karena setiap huruf mempunyai keunikan tersendiri sehingga membedakan dirinya dari huruf yang lain. *Feature extraction* memiliki tujuan agar karakteristik tertentu dari citra akan muncul dan memiliki perbedaan dengan karakteristik citra lainnya, yang disebut *feature*. Karakteristik ini dapat berupa titik koordinat awal dan akhir dari suatu goresan dan fragmen garis penyusun karakter. *Feature* yang diperoleh digunakan untuk mendefinisikan kelas suatu karakter.

Feature merupakan *high-level-attribute* dari data goresan karakter. Setelah proses *feature extraction* selesai, maka didapat *feature* dari sebuah huruf. *Feature* ini mempresentasikan informasi struktural dari sebuah huruf. Untuk mendapatkan fragmen sesuai kategori, suatu pola tulisan tangan dipisahkan pada *segmentation point*-nya.

Berikut akan dijelaskan tahapan-tahapan *feature extraction* dimulai dari tahapan penelusuran jalur dalam kerangka, klasifikasi segmen gelang, klasifikasi segmen garis, klasifikasi segmen kurva, sampai pada representasi huruf berdasarkan segmen pembentuk. Tahapan-tahapan tersebut dari suatu jalur kerangka yang dibentuk oleh suatu huruf. Klasifikasi tipe segmen ditunjukkan oleh Gambar 2.10.

Klasifikasi Tipe Segmen

	<i>line</i>	<i>right</i>	<i>left</i>	<i>loop</i>
<i>horizontal</i>	—	⤴	⤵	
<i>vertical</i>		⤴	⤵	
<i>right slope</i>	↘	⤴	⤵	○
<i>left slope</i>	↙	⤴	⤵	

Gambar 2.10 Klasifikasi tipe segmen.

Sumber : Gilewski, 1997.

2.2.14 Optical Character Recognition (OCR)

Pengertian *optical character recognition (OCR)* menurut Kaur adalah teknik di mana informasi tekstual yang terkandung dalam citra diekstrak dan dikonversi ke format teks yang dapat diedit. OCR menawarkan berbagai paket perangkat lunak yang tersedia di komputer *desktop* dengan *scanner* yang memiliki satu kendala pada mobilitas utamanya.

Penelitian OCR dan pengembangan lanjutan, tuntutan pada pengenalan tulisan tangan juga meningkat karena banyak data (seperti sebagai alamat tertulis pada amplop, jumlah ditulis pada cek, nama, alamat, nomor identitas, dan nilai dolar yang ditulis pada faktur dan bentuk) yang ditulis tangan dan mereka harus dimasukkan ke dalam komputer untuk pengolahan. Tapi awal teknik OCR didasarkan terutama pada pencocokan *template*, garis sederhana dan fitur geometris, deteksi ketukan, dan ekstraksi turunannya. Teknik seperti itu tidak cukup canggih untuk praktis pengakuan data tulisan tangan pada formulir atau dokumen. Untuk mengatasi ini, komite standar di Amerika Serikat, Kanada, Jepang, dan beberapa negara di Eropa merancang beberapa model *handprint* pada 1970-an dan 1980-an untuk orang-orang untuk menulis mereka dalam kotak. Oleh karena itu, karakter yang tertulis dalam bentuk tertentu seperti tak berbeda terlalu banyak dalam gaya, dan mereka



dapat diakui lebih mudah oleh mesin-mesin OCR, terutama ketika data yang dimasukkan oleh kelompok-kelompok yang dikontrol orang, misalnya, karyawan dari perusahaan yang sama diminta untuk menulis data mereka seperti model dibela. Kadang-kadang penulis diminta untuk mengikuti petunjuk tertentu tambahan untuk meningkatkan kualitas sampel mereka, misalnya, menulis besar, menutup *loop*, menggunakan bentuk sederhana, tidak menghubungkan karakter dan sebagainya. Dengan kendala-kendala tersebut, pengakuan OCR cetakan tangan mampu berkembang selama beberapa tahun. (Kaur, *et. al.*, 2010)

2.2.15 Artificial Neural Network

Artificial neural network (ANN) atau jaringan syaraf tiruan (JST) merupakan paradigma pengolahan informasi yang terinspirasi jaringan syaraf alamiah. (Puspitaningrum, 2006)

Struktur elemen pemrosesan pada JST dibuat seperti jaringan (keterhubungan antar elemen) kemudian saling melakukan pemrosesan dengan operasi bersamaan (paralel). Hal ini yang menyumbang faktor kemiripan antara JST dengan jaringan syaraf alami. Elemen-elemen pemroses (Gambar 2.10) pada JST di antaranya:

1. Unit atau *Neuron*

Unit merupakan elemen pengolah informasi. Unit terletak pada *input*, *output* dan *hidden layer*. Informasi yang sudah diolah akan diteruskan ke unit selanjutnya melalui *link* koneksi.

2. *Input* (x_j)

Input merupakan elemen yang menerima secara langsung sinyal masukan.

3. *Output*

Output merupakan elemen yang memberikan solusi atau hasil dari data masukan.

4. Bobot (w_{ij})

Koneksi antar *neuron* memiliki nilai matematis dengan mentransfer data antar lapisan dan digunakan untuk melakukan pengaturan jaringan (proses belajar) agar JST bisa menghasilkan keluaran seperti yang dibutuhkan. (Munakata, 2008)

5. *Bias* (w_0) dan *Threshold*

Bias dan *threshold* digunakan untuk mengatur daerah nilai ambang. Nilai *bias* digunakan untuk memperbesar nilai input untuk fungsi aktivasi. (Pandjaitan, 2007)

Sedangkan nilai *threshold* digunakan untuk memperkecil nilai input fungsi aktivasi. (Munakata, 2008)

6. *Summation Function*

Summation function dipakai untuk mencari nilai rerata bobot semua elemen masukan dengan cara mencari nilai hasil kali *input* (X_j) dengan bobot (W_{ij}) dan nilai jumlahnya (S_i). (Munakata, 2008)

Untuk N elemen *input* pada *summation function* disajikan dalam Persamaan (2-18).

$$S_i = \sum_{j=1}^N W_{ij} * X_j \dots\dots\dots (2-18)$$

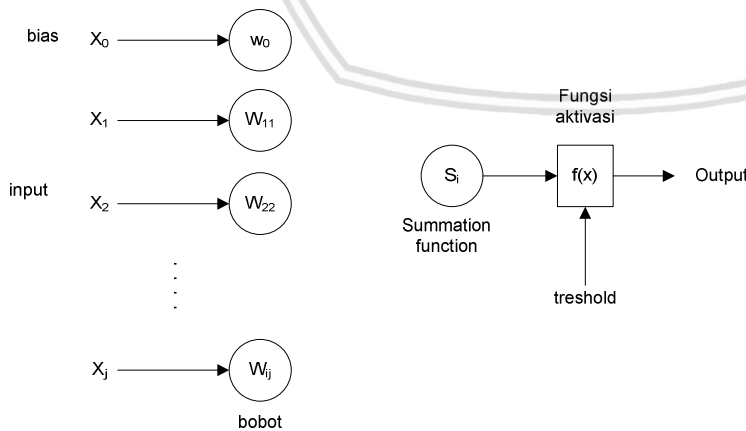
Dengan:

- S_i = nilai *summation function*
- N = jumlah bobot
- W_{ij} = bobot ke- i yang terhubung dengan *input* ke- j
- X_j = *input* ke- j

7. Fungsi Aktivasi

Fungsi yang menentukan aktif tidaknya nilai *input* berdasarkan nilai *threshold* tertentu untuk menjadi *output*. (Pandjaitan, 2007)

Nilai *output* yang dihasilkan akan dikirimkan ke unit pada lapisan selanjutnya melalui bobot koneksi.



Gambar 2.11 Model dari JST.

Sumber : Pandjaitan, 2007.



Pada Gambar 2.11 ditunjukkan model dari JST, di mana elemen pemrosesnya antara lain:

- x_j = nilai *input*,
- x_0 = *bias*,
- w_{ij} = bobot,
- S_i = *summation fuction*
- $f(x)$ = fungsi aktivasi

Cara belajar JST adalah sebagai berikut: pertama informasi dimasukkan ke dalam JST melalui unit masukan. Semua nilai bobot antar koneksi dalam JST diinisialisasi, kemudian JST dioperasikan. Semua nilai bobot ini akan dipakai untuk proses pembelajaran dan pengingatan pola informasi. Bobot akan diatur terus menerus menggunakan kriteria tertentu hingga hasil *output* yang diharapkan diperoleh. (Sutojo, *et. al.*, 2010)

Tujuan dari pelatihan JST adalah keseimbangan antara kemampuan memori dan generalisasi. Memorisasi yakni kemampuan untuk memanggil kembali sebuah pola yang telah dipelajari tanpa ada kesalahan, sedangkan generalisasi merupakan kemampuan untuk mendapatkan respon yang dapat diterima pada pola masukan yang memiliki kemiripan (tidak harus sama) sesuai pola yang telah dilatihkan sebelumnya. (Fausett, 1994)

Kualitas *training* pada JST ditentukan arsitektur jaringan. Arsitektur jaringan adalah sejumlah unit atau *neuron* dalam lapisan-lapisan (*layer*) yang saling terhubung satu sama lain antar lapisan. Arsitektur jaringan terdiri atas sejumlah unit di masing-masing *layer* (*input*, *output* dan *hidden layer*), sejumlah link koneksi yang memiliki bobot, *summation function* dan fungsi aktivasi (Sumanthi, 2010). Penentuan jumlah unit pada *input* dan *output layer* didasarkan pada permasalahan, tetapi tidak ada ketentuan untuk menentukan jumlah unit *hidden layer*. Beberapa peneliti telah mengusulkan dan menguji beberapa persamaan rumus empiris untuk menentukan jumlah unit pada *hidden layer*, yang dapat dilihat dalam Persamaan (2-19) hingga (2-25), di antaranya:

$$H = 2M + 1 \dots\dots\dots(2-19)$$

$$H = \sqrt{M * N} \dots\dots\dots(2-20)$$

$$H = \sqrt{M + N} + a \dots\dots\dots(2-21)$$

$$H = \sqrt{M(N + 2)} + 1 \dots\dots\dots(2-22)$$



$$H = 3M \dots\dots\dots (2-23)$$

$$H = 2M \dots\dots\dots (2-24)$$

$$H = \frac{2M}{3} \dots\dots\dots (2-25)$$

Dengan:

H = jumlah unit pada *hidden layer*

M = jumlah unit pada *input layer*

N = jumlah unit pada *output layer*

a = konstanta 1-10

Arsitektur yang sering digunakan dalam JST adalah *single layer* dan *multi layer*. *Single layer* terdiri atas *input layer* dan *output layer*, sedangkan *multi layer* tersusun dari *input layer*, *hidden layer* dan *output layer*. Jenis arsitektur JST dibagi ke dalam 2 kelompok yaitu *feed forwarded network* dan *recurrent atau feedback network* (Konar, 1999). *Feed forwarded network* adalah graf yang tidak mempunyai *loop*, sedangkan *recurrent atau feedback network* memiliki ciri-ciri adanya *loop* koneksi balik.

2.2.16 Back Propagation

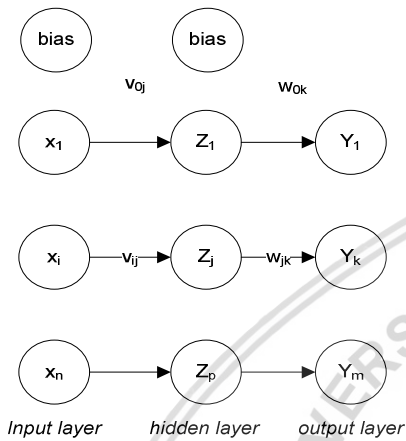
Salah satu teknik pembelajaran yang cukup bagus untuk menangani permasalahan yang lebih rumit adalah *back propagation*. *Back propagation* biasanya digunakan oleh *multi layer neural network*. *Back propagation* dapat diartikan sebagai propagasi balik yang diambil dari cara kerja *network*. (Freeman, 1992)

Ada dua tahap perhitungan yang dilakukan pada *back propagation*. Menghitung *error* antara nilai *output* dan nilai target adalah perhitungan maju. Mempropagasikan balik *error* tersebut untuk memperbaiki bobot-bobot pada semua unit disebut perhitungan mundur.

Algoritma *back propagation* dibagi menjadi 2 bagian yaitu algoritma pelatihan (*training*) dan algoritma aplikasi (*testing*). Algoritma pelatihan atau *training* terdiri atas tiga tahapan yaitu: tahapan pengumpanan maju pada pola pelatihan masukan, tahapan propagasi balik, dan tahapan mengatur bobot. Algoritma pada aplikasi hanya berupa tahap umpan maju. (Sumanthi, *et. al.*, 2010)

Dalam Gambar 2.12 *multi layer neural network* terdiri atas 3 *layer*, di antaranya *input layer*, *hidden layer* dan *output layer*. Untuk masing-masing *layer* akan terdapat sejumlah unit atau *neuron*. Sejumlah unit *input* (x_i) pada *input layer*, sejumlah unithidden (z_j) pada

hidden layer, sejumlah unit *output* pada *output layer* dan satu unit *bias* pada *hidden* dan *output layer*. Setiap unit pada layer satu dihubungkan oleh link koneksi dengan unit pada layer selanjutnya. Masing-masing link koneksi memiliki nilai bobot atau nilai bias. Nilai bobot yang menghubungkan *input layer* dengan *hidden layer* yaitu v_{ij} dan nilai bobot antara *hidden layer* dan *output layer* adalah w_{jk} . Nilai bias pada *hidden layer* adalah v_{0j} dan nilai bias untuk *output layer* adalah w_{0k} .



Gambar 2.12 Struktur dari *multi-layer* pada JST.

Sumber : Sumanthi, *et. al.*, 2010.

Unit-unit pada *multi-layer neural network* adalah:

- x_i = *input unit*,
- z_j = *hidden unit*,
- Y_k = *output unit*,
- v_{ij} = bobot antara *layer-i* *layer-j*
- w_{jk} = bobot antara *layer-j* dan *layer-k*

Algoritma *training back propaggation* adalah sebagai berikut: (Sumanthi, 2010)

1. Nilai bobot diberi nilai awal
2. Dikerjakanlah tahap-tahap sebagai berikut selama kondisi berhenti bernilai salah:

Tahap umpan maju: masing-masing unit *input* (X_i , $i = 1, 2, 3... n$) mendapatkan sinyal x_i . Unit *input* akan meneruskan sinyal masukan pada semua unit pada *hidden layer*; Setiap unit *hidden* (Z_j , $j = 1, 2, 3... p$) melakukan penjumlahan bobot sinyal *input* (z_in_j) seperti dapat dilihat dalam Persamaan (2-26).

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2-26)$$



Dengan:

- z_{in_j} = jumlah bobot pada unit *hidden*
- v_{0j} = nilai bias pada unit *hidden* ke-j
- n = jumlah unit *input*
- x_i = nilai unit *input* ke -i
- v_{ij} = nilai bobot antara unit *input* ke-i dan unit *hidden* ke-j

dan menerapkan fungsi sigmoid guna menghitung sinyal *output* pada unit *hidden* ke-j (z_j) yang dapat dilihat dalam persamaan (2-27).

$$z_j = f(z_{in_j}) \dots\dots\dots (2-27)$$

3. Sinyal *output* dikirim ke masing-masing unit *output*. Fungsi sigmoid yang digunakan adalah fungsi sigmoid biner yang memiliki interval nilai 0 sampai 1. Fungsi sigmoid biner dapat dilihat dalam persamaan (2-28).

$$f(x) = \frac{1}{1 + e^{-x}} \dots\dots\dots (2-28)$$

Maka persamaan untuk menghitung sinyal *output* pada unit *hidden* ke-j dapat dilihat dalam persamaan (2-29).

$$z_j = \frac{1}{1 + e^{-z_{in_j}}} \dots\dots\dots (2-29)$$

4. Semua unit *output* ($Y_k, k = 1, 2, 3 \dots m$) akan menghitung jumlah bobot sinyal *input* (y_{in_k}) yang dapat dilihat dalam persamaan (2-30).

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \dots\dots\dots (2-30)$$

Dengan:

- y_{in_k} = jumlah bobot pada unit *output*
- w_{0k} = nilai bias pada unit *output* ke-k
- p = jumlah unit *hidden*
- z_j = nilai unit *hidden* ke-j
- w_{jk} = nilai bobot antara unit *hidden* ke-j dan unit *output* ke-k

Kemudian fungsi aktivasi akan diterapkan untuk menghitung sinyal *output* pada unit *output* ke-k (Y_k) yang dapat dilihat dalam persamaan (2-31).

$$Y_k = \frac{1}{1 + e^{-y_{in_k}}} \dots\dots\dots (2-31)$$



2.2.17 Tahap Propagasi Balik

Dalam tahap ini, pola dari target (t_k) yang memiliki kecocokan dengan pola input pelatihan akan diterima oleh setiap unit *output* ($Y_k, k=1,2,3,..,m$), kemudian menghitung nilai *error* (δ_k) yang dapat dilihat dalam Persamaan (2-32) hingga (2-30).

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \dots\dots\dots (2-32)$$

$$f'(y_{in_k}) = \left(\frac{1}{1+e^{-y_{in_k}}}\right) * \left(1 - \left(\frac{1}{1+e^{-y_{in_k}}}\right)\right) \dots\dots\dots (2-33)$$

Dengan:

- δ_k = nilai *error* pada unit *output* ke-k
- t_k = nilai target
- y_k = nilai output
- f' = turunan fungsi aktivasi
- y_{in_k} = jumlah bobot pada unit *output*

Setelah itu akan dihitung koreksi bobot (Δw_{jk}) dan koreksi bias (Δw_{0k}) yang dapat dilihat dalam Persamaan (2-34).

$$\Delta w_{jk} = \alpha \delta_k z_j \text{ dan } \Delta w_{0k} = \alpha \delta_k \dots\dots\dots (2-34)$$

Dengan:

- Δw_{jk} = koreksi bobot dari unit *hidden* ke-j dengan unit *output* ke-k
- Δw_{0k} = koreksi bias pada unit output ke-k
- α = *learning rate*
- δ_k = nilai *error* pada unit *output* ke-k
- z_j = nilai unit *hidden* ke-j

Pada masing-masing unit *hidden* ($z_j, j = 1, 2, 3... p$), dilakukan penjumlahkan delta *input* (berdasarkan semua unit yang ada pada *layer* berikutnya) yang dapat dilihat dalam Persamaan (2-35).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \dots\dots\dots (2-35)$$

Dengan:

- δ_k = nilai *error* pada unit *output* ke-k
- m = jumlah unit *output*
- w_{jk} = nilai bobot yang ada di antara unit *hidden* ke-j dan unit *output* ke-k



Perhitungan informasi *error* pada unit *hidden* ke-j (δ_j) diperoleh dengan mengalikan Persamaan (2-35) dengan turunan dari fungsi aktivasinya yang dapat dilihat dalam Persamaan (2-36) dan Persamaan (2-37).

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \dots \dots \dots (2-36)$$

$$\delta_j = \delta_{in_j} * \left(\left(\frac{1}{1+e^{-z_{in_j}}} \right) * \left(1 - \left(\frac{1}{1+e^{-z_{in_j}}} \right) \right) \right) \dots \dots \dots (2-37)$$

Dengan:

δ_j = nilai *error* pada unit *hidden* ke-j

z_{in_j} = jumlah bobot pada unit *hidden*

Setelah itu, proses menghitung koreksi bobot (Δv_{jk}) dan koreksi bias (Δv_{0j}) yang dapat dilihat dalam Persamaan (2-38).

$$\Delta v_{ij} = \alpha \delta_j x_i \text{ dan } \Delta v_{0j} = \alpha \delta_j \dots \dots \dots (2-38)$$

Dengan:

Δv_{jk} = koreksi bobot antara unit *hidden* ke-j dan unit *input* ke-i

Δv_{0j} = koreksi bias pada unit *input* ke-i

α = learning rate

δ_j = nilai *error* pada unit *hidden* ke-j

x_i = nilai unit *input* ke-i

Pada setiap unit *output* (Y_k , $k = 1, 2, 3 \dots m$) akan dilakukan pengubahan bobot dan pengubahan *bias* ($j = 1, 2, 3 \dots p$) yang dapat dilihat dalam Persamaan (2-39).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots \dots \dots (2-39)$$

Dengan:

$w_{jk}(\text{baru})$ = bobot baru antara unit *hidden* ke-j dan unit *output* ke-k

$w_{jk}(\text{lama})$ = bobot lama antara unit *hidden* ke-j dan unit *output* ke-k

Δw_{jk} = koreksi bobot antara unit *hidden* ke-j dan unit *output* ke-k

Pada setiap unit *hidden* (z_j , $j = 1, 2, 3 \dots p$) akan dikenakan pengubahan bobot dan pengubahan *bias* ($i = 0, 1, 2 \dots n$) yang dapat dilihat dalam Persamaan (2-40).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots \dots \dots (2-40)$$



Dengan:

$v_{ij}(\text{baru})$ = bobot baru antara unit *hidden* ke- j dan unit *input* ke- i

$v_{ij}(\text{lama})$ = bobot lama antara unit *hidden* ke- j dan unit *input* ke- i

Δv_{ij} = koreksi bobot antara unit *hidden* ke- j dan unit *input* ke- i



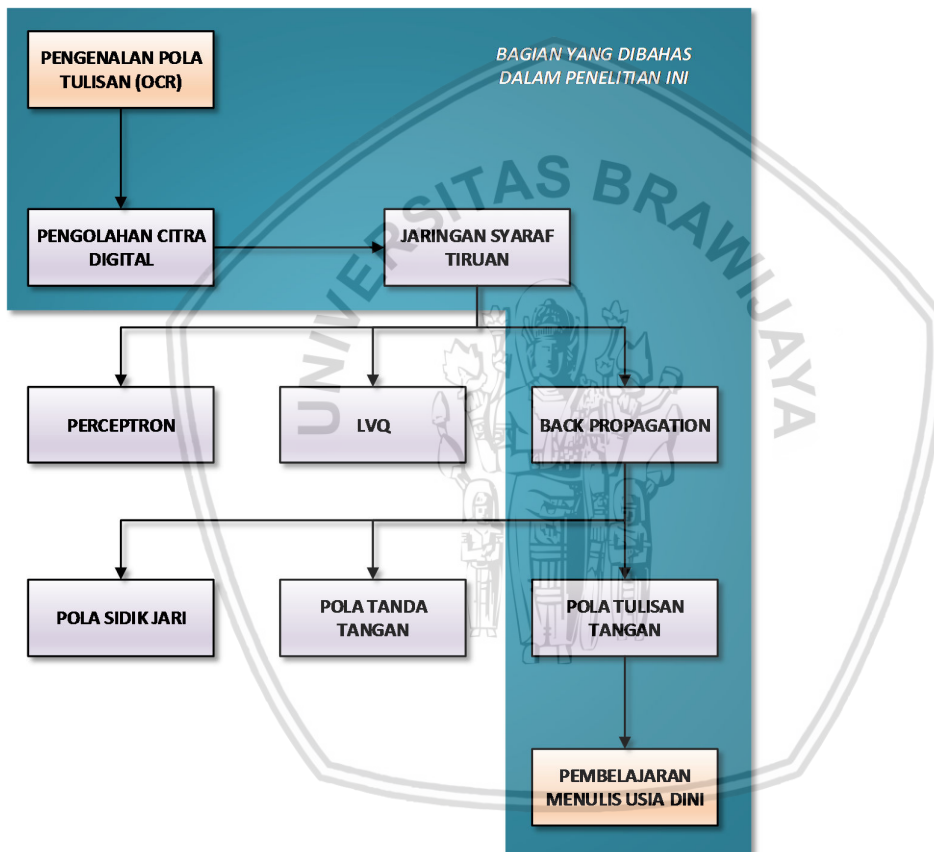


BAB III

KERANGKA KONSEP PENELITIAN

3.1 Kerangka Konsep Berpikir

Pada Gambar 3.1 ditunjukkan tentang kerangka konsep penelitian yang berisi struktur yang dibahas dalam penelitian ini. Bagian yang bersisir merah adalah yang dibahas dalam penelitian ini.



Gambar 3.1 Kerangka konsep berpikir pada penelitian ini.

Berdasarkan kerangka konsep penelitian pada Gambar 3.1 dijelaskan algoritma yang dipakai untuk menyelesaikan permasalahan pengenalan pola tulisan menggunakan jaringan saraf tiruan dengan metode *back propagation*. Sesuai dengan diagram pada Gambar 3.1 di atas bisa dijelaskan bahwa pengenalan pola tulisan pertama diproses menggunakan pengolahan citra digital yaitu untuk mengedit citra masukan, kemudian citra diubah menjadi bentuk *text* dengan menggunakan OCR. Yang kemudian dilakukan pelatihan untuk

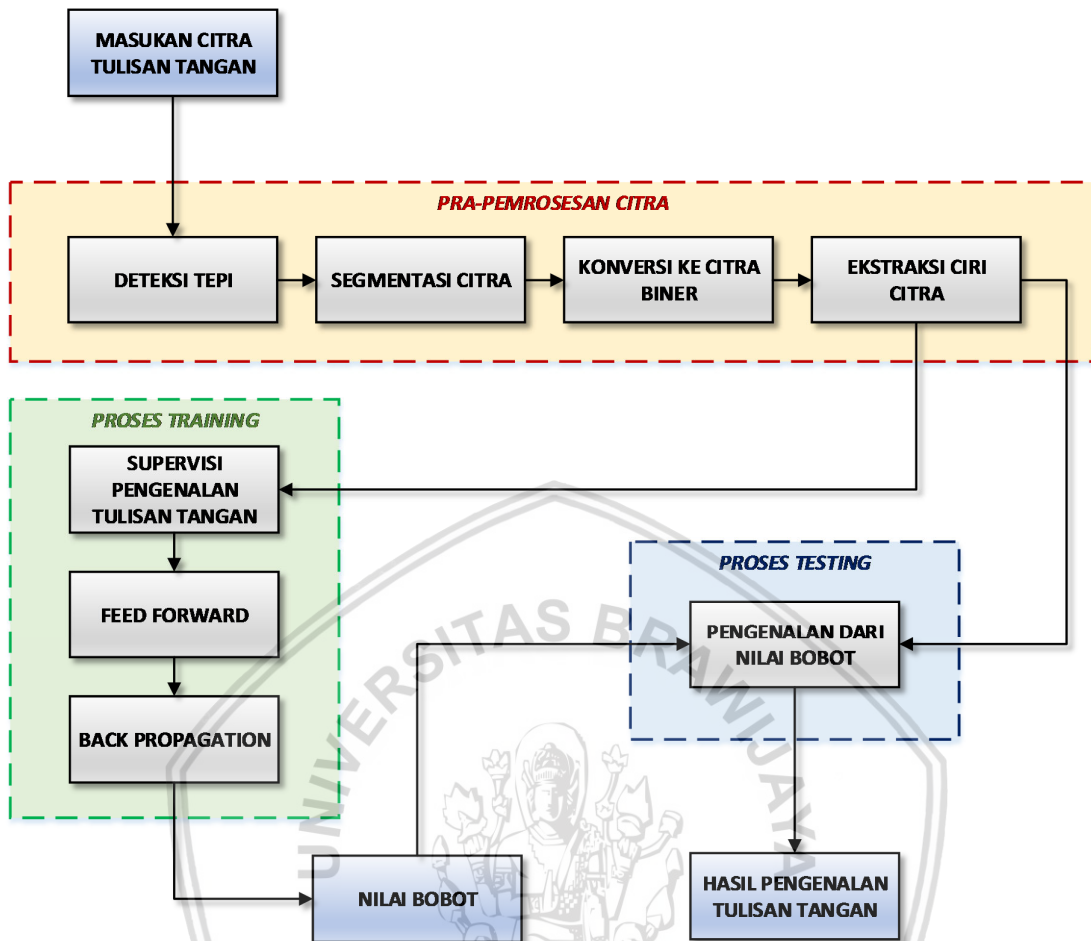
menyempurkan pengenalan huruf, digunakan jaringan saraf tiruan. Jaringan Saraf Tiruan (JST) terdiri atas *perceptron*, *hebb rule*, *delta rule*, *back propagation*, *heteroassociative memory*, *bidirectional associative memory*, dan *learning vector quantization*.

Dari banyaknya metode pada jaringan saraf tiruan yang dipilih untuk mengenali tulisan tangan pada aplikasi yang akan dibangun adalah metode *backpropagation*. Karena metode *Backpropagation* telah teruji keberhasilannya apabila perangkat lunak dijalankan dalam personal komputer. Sehingga untuk mengembangkan perangkat lunak yang sudah ada peneliti mencoba mengembangkan pengenalan pola tulisan tangan yang difungsikan untuk proses belajar menulis dan membaca anak umur 4 tahun hingga 5 tahun.

Pada usia 4 tahun hingga 5 tahun, anak diberikan pengajaran menulis kemudian sistem mengenali tulisan tersebut benar atau salah penulisannya. Misalnya anak menulis huruf “p” di mana huruf “p” memiliki perut di atas, akan tetapi anak menuliskan huruf “p” tersebut meletakkan perutnya di bawah. Maka sistem akan mengenali huruf tersebut sebagai huruf “b”. Kemudian sistem menginstruksikan bahwa penulisan huruf “p” memiliki perut di atas tidak di bawah.

3.2 Rancangan Perangkat Lunak

Perancangan pengenalan pola tulisan secara keseluruhan terdapat dalam Gambar 3.2 di mana tahap awal pengenalan tulisan adalah *pre-processing*. Pada tahap *pre-processing* akan dilakukan penyederhanaan citra agar siap untuk dianalisis. Proses *pre-processing* terbagi ke dalam 4 proses, yaitu *grayscale*, *median*, *histogram equalization* dan *thresholding*. Tahap selanjutnya segmentasi citra, yang merupakan tahapan proses segmentasi bertujuan untuk menghilangkan bagian yang tidak merupakan bagian dari tulisan. Untuk mendapatkan koordinat piksel proses ini menggunakan operasi aras titik dengan mencari titik empat penjuru sudut yang terdapat dalam tulisan. Untuk memulai proses segmentasi pada citra, maka setiap karakter yang akan dideteksi harus menunjukkan titik lokasi piksel pada setiap sudut pada huruf.

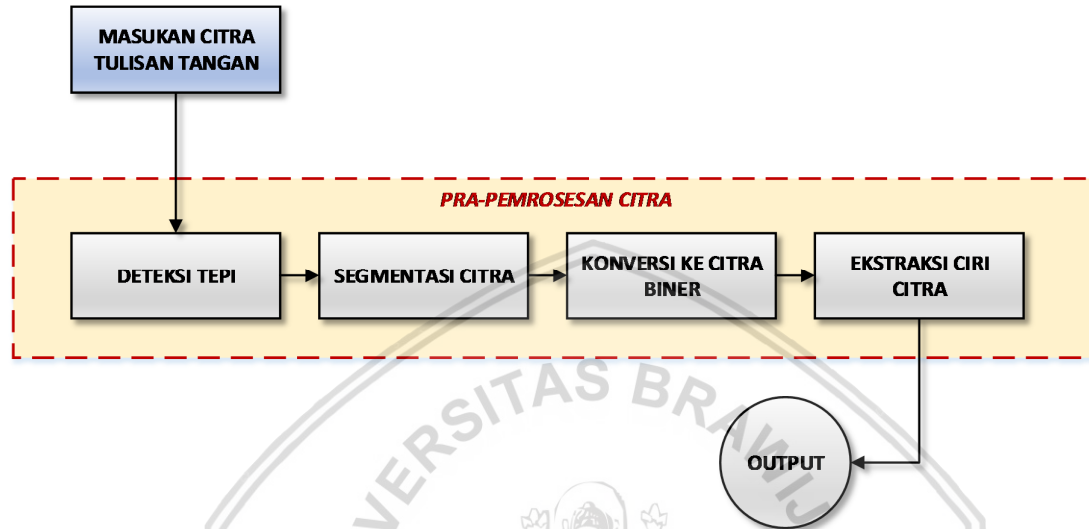


Gambar 3.2 Diagram pengenalan tulisan tangan secara umum.

Tahap selanjutnya *normalization* hasil pada proses segmentasi yang dilakukan sebelumnya ialah posisi, lebar dan tinggi pada setiap huruf. Masing-masing huruf hasil segmentasi akan dibaca oleh JST sebagai nilai *input* awal jaringan. Supaya huruf tersebut terbaca oleh jaringan maka huruf dinormalisasikan menjadi *image greyscale* (8 bit). Selanjutnya pengenalan karakter dengan menggunakan *neural network backpropagation* dilakukan untuk melalui tahap pelatihan (*training*) kemudian dilanjutkan pada tahap pengenalan. Sebelum masuk ke dalam proses JST, *input* karakter masukan harus dalam bentuk citra hitam putih. Citra hitam putih karakter tersebut sudah melalui proses normalisasi ukuran agar masukkan ke dalam jaringannya sama. Tahap pelatihan dilakukan yang pertama sampai mencapai target *error* yang diinginkan. Selanjutnya, tahap pengenalan dapat dilakukan. Pada proses *Character Recognition* dilakukan pengekstraksian terhadap bagian-bagian yang terdapat pada pengenalan tulisan. Sesudah pola tulisan dikenali karakternya, aplikasi akan mengekstraksi pengenalan pola tulisan sehingga tulisan dapat dikenali.

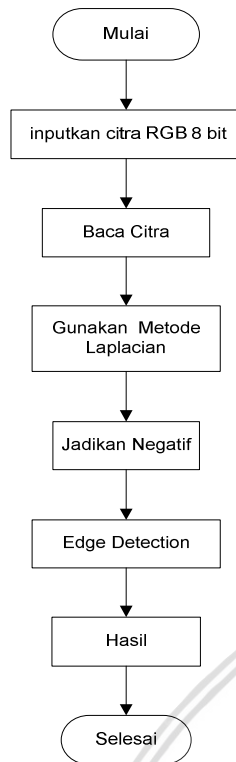
3.3 Edge Detection dengan Operator Laplacian

Edge detection termasuk dalam sebuah bagian proses pra-pemrosesan citra. Pada Gambar 3.2 yang merupakan proses perancangan keseluruhan secara umum, Gambar 3.3 menunjukkan diagram pra-pemrosesan secara umum.



Gambar 3.3 Diagram umum pra-pemrosesan citra.

Turunan kedua akan digunakan guna melakukan pendeteksian tepian yang dihasilkan oleh selisih intensitas keabuan pada citra. Selanjutnya pencarian titik pelana (*zero crossing*) akan dilakukan untuk menentukan lokasi *edge* (garis tepian). *Edge detection* (pendeteksian tepian) akan menggunakan operator *laplacian* yang merupakan salah satu pengembangan dari teknik *edge detection* yang telah dibahas, dengan ciri khas matriks dengan ukuran 3×3 , dan *flowchart* dari diagram tersebut terdapat dalam Gambar 3.4.



Gambar 3.4 Diagram alir deteksi tepi.

3.4 Pengenalan Pola Tulisan Tangan Menggunakan Jaringan Saraf Tiruan Backpropagation

Input image yang digunakan ialah citra dengan resolusi 512×256 piksel dan memiliki 256 derajat keabuan pada baris dan kolom yang menandakan sebuah titik untuk citra juga merupakan kesesuaian dari nilai elemen matriks untuk mengidentifikasi derajat keabuan dari titik tersebut. Karena komputasi matriks dengan dimensi besar membutuhkan waktu pemrosesan yang cukup lama maka diperlukan reduksi dimensi matrik untuk mempercepat proses. Proses untuk melakukan identifikasi tulisan tangan ini didasarkan pada pengenalan pola dengan menggunakan pendekatan JST. Metode JST yang digunakan dalam penelitian ini ialah *back propagation*.

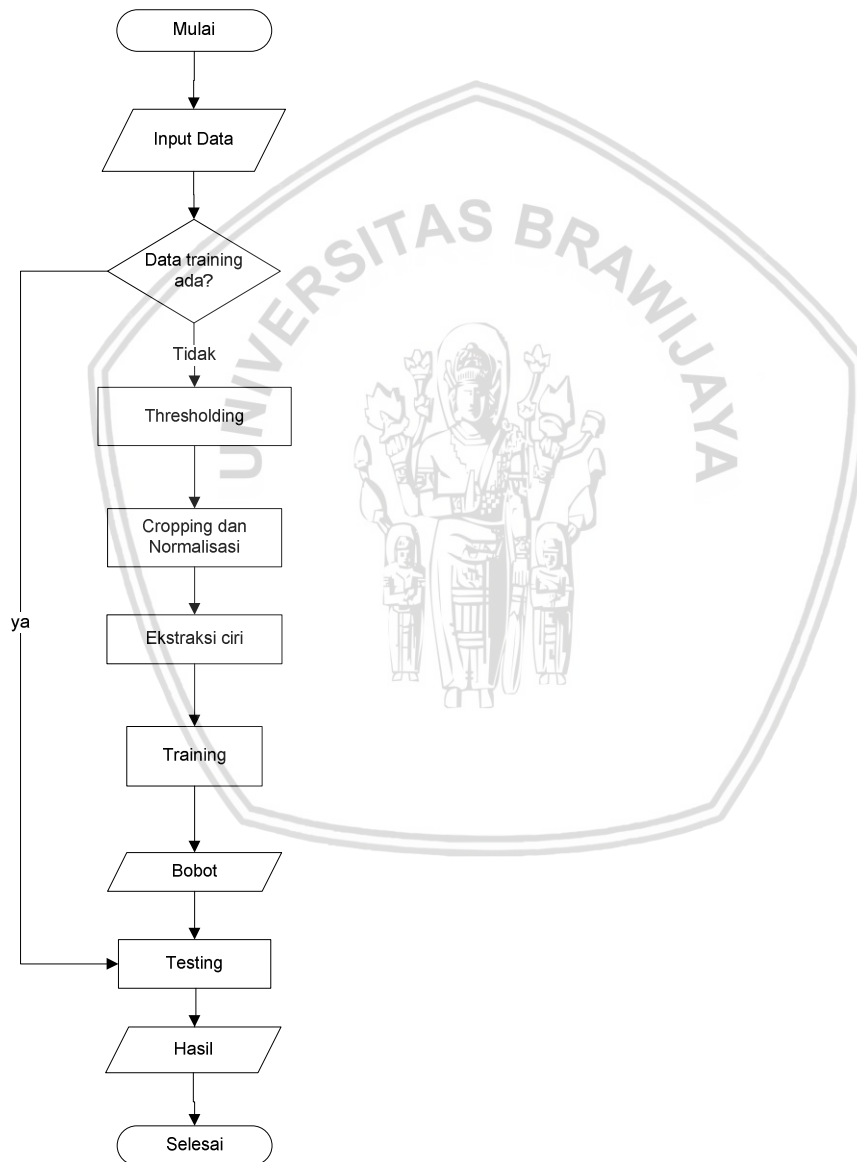
Proses identifikasi tulisan tangan untuk balita ini dilakukan dengan cara perbandingan citra masukan dengan citra yang telah dilatihkan pada sistem. Tahap proses pengenalan tulisan tangan ini adalah sebagai berikut:

1. Input berupa citra digital.
2. Proses pengolahan *image* meliputi *Edge detection* menggunakan operator *Laplacian*

3. Proses pembuatan data masukan dengan menggunakan jaringan syaraf tiruan.
4. Proses *training* (pelatihan) dan *testing* (pengujian).

3.5 Pra-Pemrosesan Pengolahan Data Training dan Testing

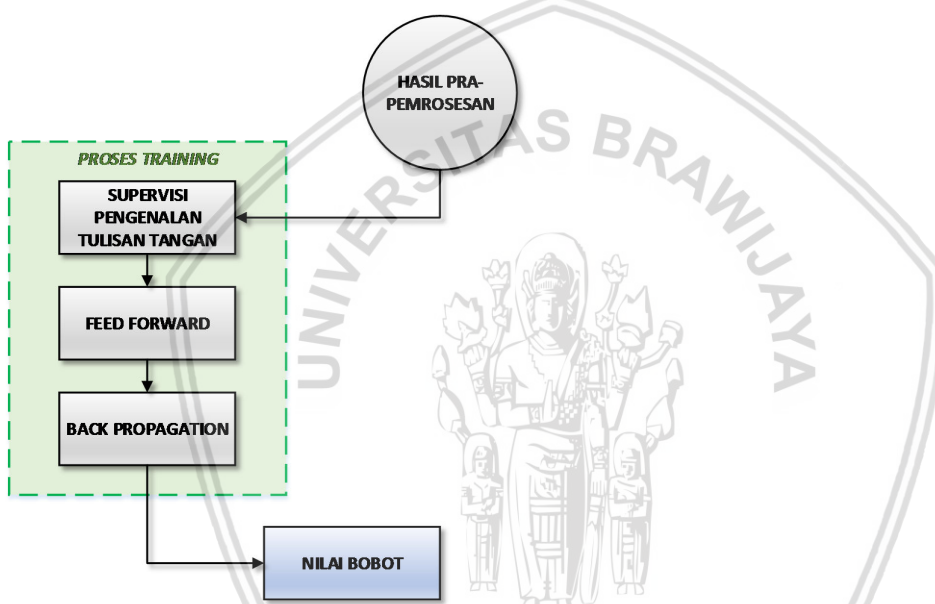
Secara umum, diagram untuk melakukan pra-pemrosesan citra telah ditunjukkan pada Gambar 3.2. Sedangkan secara rinci, langkah-langkah proses dari data *training* dan *testing* pengenalan pola tulisandapat dilihat pada *flowchart* yang terdapat pada Gambar 3.5.



Gambar 3.5 Diagram alir pra-pemrosesan citra untuk *training* dan *testing*.

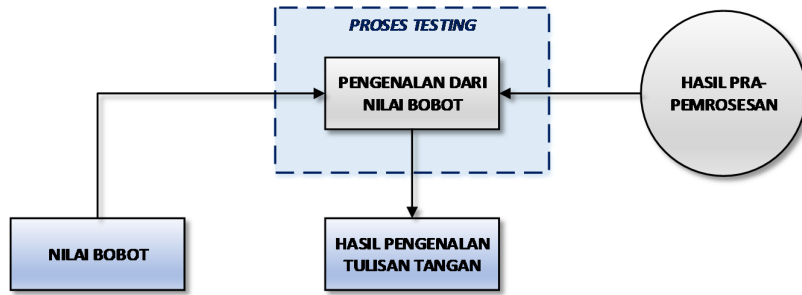
3.6 Perancangan Jaringan Syaraf Tiruan

Secara umum, proses pelatihan memiliki diagram seperti pada Gambar 3.6. Proses pelatihan akan dilakukan dengan awalan proses inialisasi bobot yang kemudian dilanjutkan dengan tahap *feed forward*, yaitu menjumlahkan sinyal-sinyal masukan bobot awal pada *hidden layer* dan *output layer*. Semua sinyal yang telah diterima akan diaktivasi dengan menggunakan fungsi sigmoid biner. Tahap berikutnya adalah tahap *back propagation* yang melakukan penghitungan *error*. Nilai *error* didapatkan dari selisih antara nilai target yang sudah ditentukan terhadap nilai keluaran dari *output layer*. Nilai *error* tersebut difungsikan untuk memberikan koreksi pada bobot yang ada di unit *output* dan *hidden*.



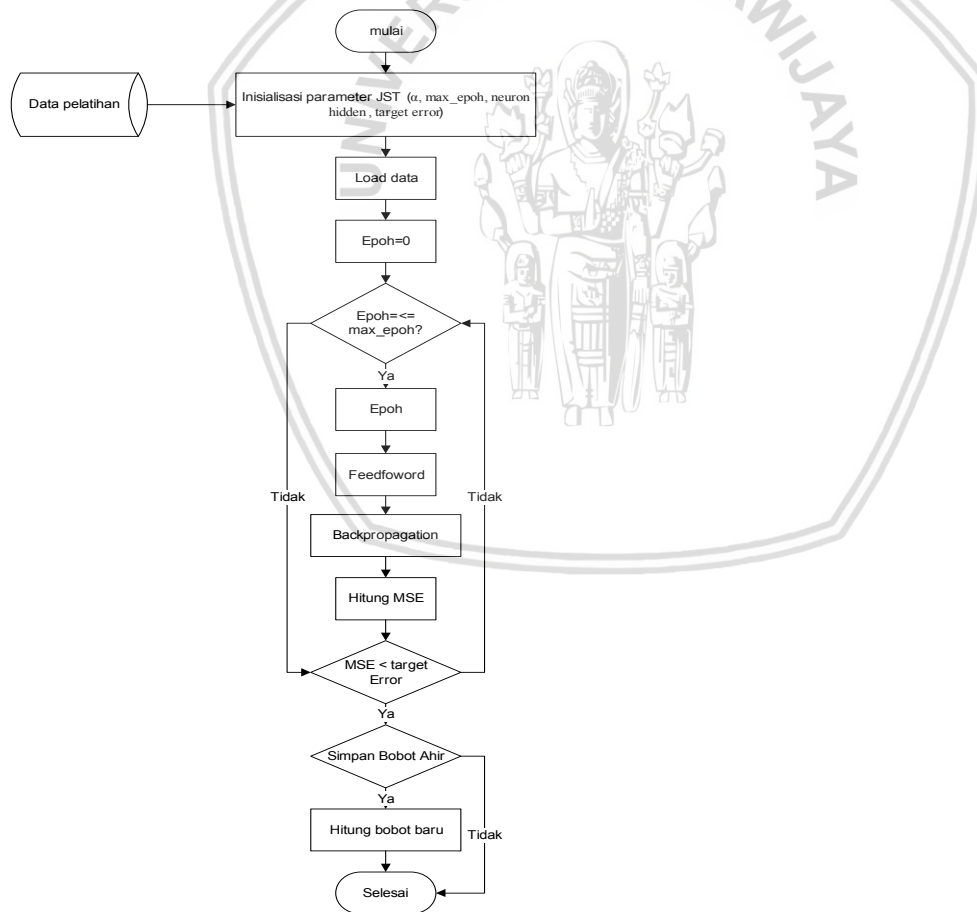
Gambar 3.6 Diagram umum proses pelatihan.

Gambar 3.7 menunjukkan diagram umum proses pengenalan. Dalam hal ini, koreksi nilai bias akan dilakukan pada semua unit keluaran dan unit tersembunyi. Setiap unit keluaran dan unit tersembunyi akan secara otomatis melakukan perbaikan bobot dan bias yang dimiliki. Langkah yang terdapat di atas dilakukan berulang-ulang sampai mendapatkan target *error*.



Gambar 3.7 Diagram umum proses pengenalan.

Mean Square Error (MSE) pada proses pelatihan didapat dari penjumlahan kuadrat *error* pada setiap *training* selama sekali *epoch* yang dibagi dengan banyaknya data *training*. Semakin tinggi iterasi *epoch* akan cenderung mendapatkan nilai *error* yang semakin kecil. Akan tetapi *training* yang terlalu banyak dipengaruhi oleh berbagai faktor, yaitu penentuan *learning rate*, banyaknya *hidden layer* beserta arsitektur jaringannya.



Gambar 3.8 Diagram alir proses pelatihan.

Input dari proses ini adalah kode nilai intensitas dari setiap *pixel* hasil pengolahan proses ekstraksi ciri sebelumnya. Nilai intensitas setiap *pixel* ini akan digunakan untuk menghitung layer unit dengan menggunakan fungsi aktivasi untuk mencari perubahan bobot sampai diperoleh hasil dengan *error rate* yang ditentukan. Nilai learning rate yang akan digunakan sebagai acuan adalah 0,001, *Error Rate* = 0,01, *maximum epoch* = 100, bobot awal = -0,5 hingga 0,5, *input pattern* = matriks 30×30, jumlah *node* pada *input neuron* = 600, jumlah *node output neuron* = 600 serta *hidden layer* = 55.

Dari matriks 30×30 yang dibentuk, terdapat total 900 piksel. Karena 900 piksel terlalu banyak sebagai total jumlah input neuron yang mengakibatkan jumlah koneksi melampaui batas memory komputer, maka dari 900 piksel ini dilakukan reduksi sehingga jumlah node yang dipakai sebagai *input neuron* menjadi 600. Proses reduksi tersebut adalah dengan menggabung 2 piksel vertikal menjadi 1 piksel dengan mengganti nilai 1 piksel menjadi nilai nol jika kedua piksel tersebut berbeda nilai. Penggabungan nilai 2 piksel menjadi 1 tersebut dilakukan setiap 6 piksel vertikal pertama dan dilompati sebanyak 3 piksel vertikal, sehingga untuk setiap 9 piksel akan terbentuk 3 piksel hasil gabungan 6 piksel dan 3 piksel asli, berjumlah total 6 piksel.

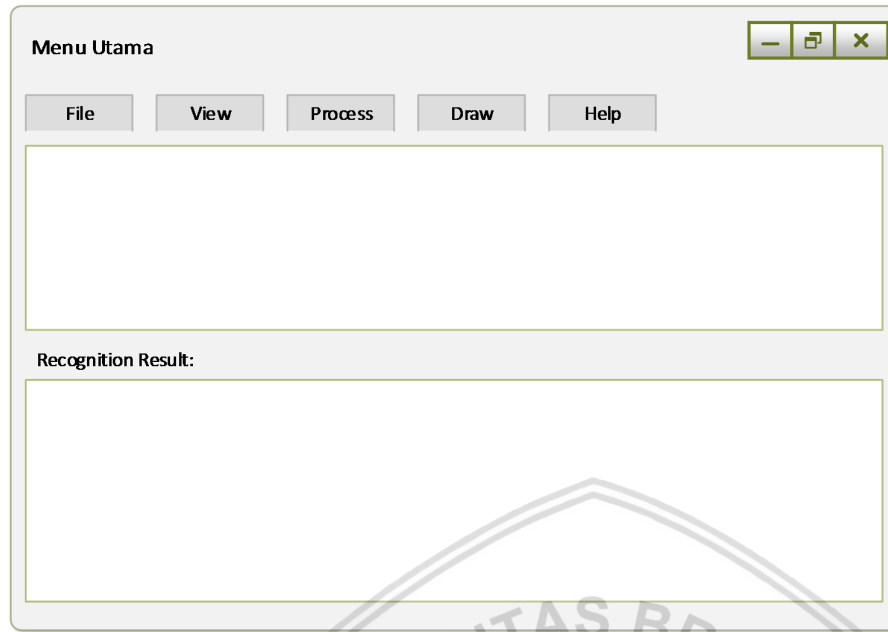
3.7 Desain Antarmuka

Aplikasi ini dirancang menggunakan desain *user interface form* yang tersusun atas empat form. Empat form itu adalah form utama, form proses awal, form *save*, dan form *training*.

3.7.1 Desain Form Tampilan Menu Utama

Gambar 3.9 berfungsi sebagai layar utama, pada halaman tersebut terdapat 5 menu, yaitu:

1. *File* terdiri atas *New Knowledge*, *Open Knowledge*, *Save Knowledge*, *Save Knowledge As*, *New Picture*, *Open Picture*, *Save Picture*, *Save Picture As*, *Save Result*, *Save Result As*, *Exit*.
2. *View* terdiri atas *Toolbar*, *Status Bar*, *Picture Title* dan *Result Title*
3. *Process* terdiri dari *Training*, *Recognition* dan *Option*
4. *Draw* terdiri dari *Pencil*, *Eraser* dan *Clear*



Gambar 3.9 Rancangan menu utama.

3.7.2 Desain Form File

Fungsi dari menu-menu form *file* adalah membuat pembelajaran baru, membuka pembelajaran, menyimpan pembelajaran, menyimpan kembali pembelajaran, membuat citra baru, membuka citra baru, menyimpan citra baru, menyimpan kembali citra baru, menyimpan hasil, menyimpan kembali hasil, dan yang terakhir adalah keluar dari aplikasi.

3.7.3 Desain Form View

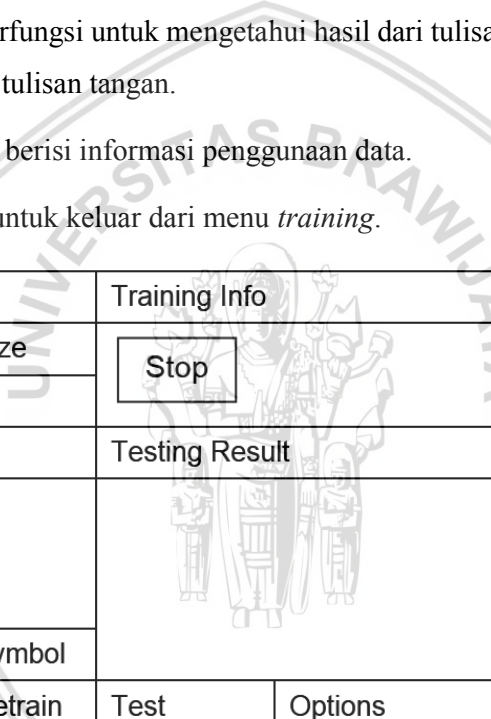
Pada form *View* terdapat fungsi-fungsi yang hanya digunakan sebagai penunjuk keterangan.

3.7.4 Desain Form Proses dan Draw

Fungsi dari masing-masing tombol, sesuai pada Gambar 3.10 adalah:

1. *Label input Pattren* terdiri atas 4 pilihan. Pilihan pertama *Fonts* memiliki fungsi untuk memilih jenis-jenis tulisan yang akan di tampilkan pada tampilan *handwriting*. Pilihan kedua *Chars* berfungsi untuk memilih karakter, angka dan huruf yang akan dituliskan pada pada tampilan *handwriting*. Pilihan ketiga *Size* berfungsi untuk mengatur besar dan kecil tulisan yang terdapat pada tampilan *handwriting*. Pilihan keempat tombol *handwriting*, pada Edit 1 berfungsi untuk menulis tulisan yang akan dicocokkan yang hasilnya dapat dilihat pada Edit 2.

2. *Label Training Info* berisi untuk memberikan informasi percobaan dari pelatihan pengenalan tulisan tangan. Kemudian tombol *Stop* berfungsi untuk menghentikan percobaan.
3. *Label Testing Result* berisi informasi keberhasilan dari tulisan tangan yang telah dituliskan.
4. *Tombol Add* berfungsi untuk mengenali tulisan tangan.
5. Tombol *Train* berfungsi untuk melakukan pencocokan tulisan tangan.
6. Tombol *Retrain* berfungsi untuk mengulang kembali pencocokan tulisan tangan.
7. Tombol *Test* berfungsi untuk mengetahui hasil dari tulisan tangan setelah dilakukan uji pencocokan tulisan tangan.
8. Tombol *Option* berisi informasi penggunaan data.
9. Tombol *Close* untuk keluar dari menu *training*.

Input Pattern			Training Info		
Fonts	Chars	Size	<div style="border: 1px solid black; padding: 5px; display: inline-block;">Stop</div>		
Handwriting			Testing Result		
Edit 1		Edit 2			
Pencil	Eraser	Symbol			
Add	Train	Retrain			

Gambar 3.10 Tampilan proses percobaan pengenalan tulisan tangan.

3.8 Hipotesis

Metode *JST back propagation* yang diimplementasikan pada perangkat lunak komputer diharapkan mampu melakukan pengenalan tulisan tangan sebagai media pembelajaran membaca dan menulis bagi anak usia 4 tahun sampai dengan 5 tahun.



BAB IV

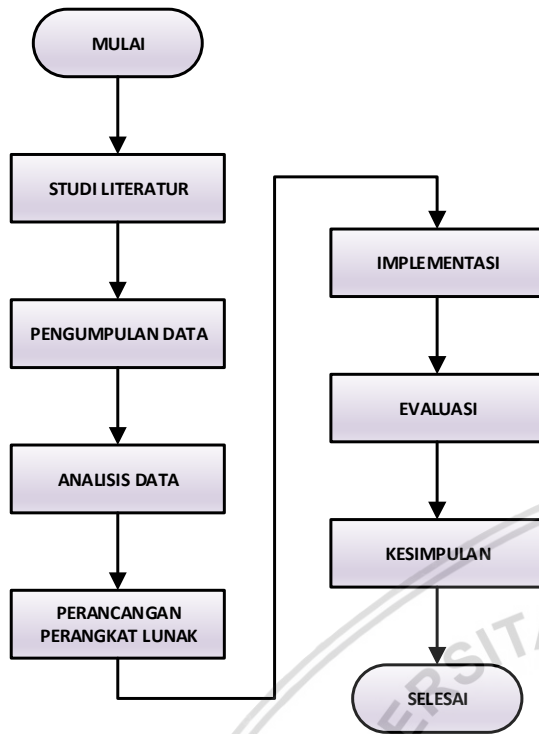
METODE PENELITIAN

Untuk penyelesaian masalah dalam perumusan permasalahan dan untuk merealisasikan tujuan penelitian ini yang terdapat di dalam Bab 1, yaitu pendahuluan, maka digunakan metode penelitian untuk penyelesaiannya. Dalam bab ini akan di bahas langkah-langkah, variabel dan aturan analisis pengenalan tulisan tangan untuk balita (anak usia 4 sampai dengan 5 tahun).

4.1 Langkah-Langkah Penelitian

Penelitian ini secara keseluruhan memiliki langkah-langkah seperti terdapat pada Gambar 4.1. Pada awalnya, dilakukan studi literatur, yang kemudian diikuti dengan pengumpulan data. Proses pengumpulan data dilakukan pada PAUD Iklas Nurani Tanggung Bence, RT 01, RW 01, Garum, Blitar. Setelah itu dilakukan analisis pada data yang telah diambil.

Langkah berikutnya adalah melakukan perancangan perangkat lunak, yang meliputi perancangan sistem, database, algoritma, dan antarmuka. Setelah itu proses berikutnya adalah melakukan implementasi. Langkah berikutnya adalah melakukan pengujian pada perangkat lunak dan siswa pada beberapa PAUD yang ada di Kabupaten Blitar. Setelah itu terdapat tahap evaluasi dan tahap pengambilan kesimpulan.



Gambar 4.1 Diagram alir pengenalan pola tulisan tangan untuk balita.

4.2 Variabel dan Aturan Analisis

Adapun variabel penelitian yang difungsikan untuk penelitian ini ialah sebagai berikut:

1. Nilai *learning rate* (α).
2. Nilai *error rate*, sebagai nilai *error* terbesar yang dapat diterima sehingga nantinya dapat menghasilkan *output* yang optimal (E).
3. Jumlah maksimum iterasi atau *epoch* sampai perulangan berhenti.
4. Bobot awal yang digunakan (v_{ij} dan w_{jk}).
5. Jumlah *neuron* pada *hidden layer* (j).

4.3 Tahap Pengumpulan dan Analisis Data

Data yang digunakan adalah data citra digital yang berupa tulisan tangan anak PAUD usia 4 sampai dengan 5 tahun. Tempat penelitian di PAUD Iklas Nurani Tanggung Bence RT : 01 / RW : 01 Garum Blitar. Siswa PAUD Iklas Nurani 12 anak usia 4 sampai dengan 5 tahun. Pengumpulan citra dilakukan dengan memindai langsung berbagai macam tulisan tangan (tidak berupa tulisan sambung) atau data yang berupa tulisan tangan pada komputer yang ditulis langsung secara realtime di dalam program.

Setelah data-data terkumpul, tahap berikutnya adalah melakukan pengecekan terhadap citra tersebut. Apakah layak untuk dijadikan citra uji maupun citra latih. Format citra yang digunakan ialah format citra dengan tipe file *bitmap* (*.bmp) atau citra yang ditulis langsung (*real time*) pada aplikasi yang telah dibuat.

4.4 Perancangan Aplikasi Pengenalan Tulisan untuk Balita

Perancangan aplikasi pengenalan tulisan untuk balita dilakukan dengan menggunakan bantuan perangkat lunak Delphi 7 dan Microsoft Visio untuk membuat diagram alir. Di dalam perangkat lunak tersebut sudah mendukung kemampuan untuk melakukan penggambaran diagram *flowchart*, diagram antarmuka (*forms*), dan diagram database, meskipun tidak terdapat kemampuan untuk melakukan pemeriksaan validasi diagram tersebut. Validasi diagram dilakukan secara visual dan manual. Pada Sub Bab 3.2 telah dijelaskan rancangan perangkat lunak yang dimaksudkan.

4.5 Implementasi

Pada tahapan ini akan dilakukan penerapan pada hasil rancangan sistem menuju proses pembuatan aplikasi. Beberapa komponen pendukung yang akan digunakan untuk tahap implementasi ini adalah menggunakan Bahasa Pemrograman untuk aplikasi konversi citra tulisan tangan ke teks digital dikembangkan menggunakan bahasa pemrograman Delphi 7.

Proses yang dilakukan pada tahap implementasi ini adalah :

1. Mengimplementasikan rancangan ke dalam *code* program dengan menggunakan bahasa pemrograman Delphi 7.
2. Mengimplementasikan perancangan antar muka.

4.6 Evaluasi

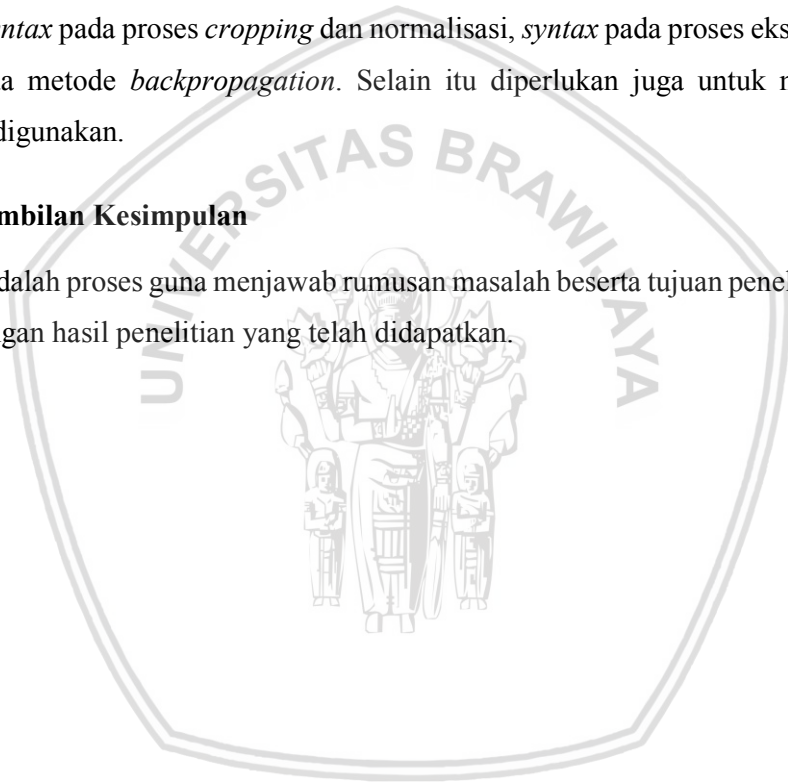
Pada tahap evaluasi dilakukan pengujian terhadap aplikasi pengenalan tulisan untuk balita menggunakan metode *whitebox* dan metode *blackbox*. *Whitebox testing* ialah pengujian yang berdasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol pada desain program secara prosedur guna membagi pengujian didalam beberapa kasus *testing*. Sedangkan *Blackbox Testing* ialah pengujian aspek fundamental aplikasi pengenalan tulisan untuk balita tanpa memperhatikan struktur logika yang terdapat dalam aplikasi pengenalan tulisan untuk balita.

Untuk perangkat lunak ini pengujian *Blackbox* dilakukan dengan memberi masukan ke dalam perangkat lunak kemudian melihat apakah keluaran yang dihasilkan sesuai dengan yang diharapkan. Keluaran yang diharapkan dari perangkat lunak ini adalah berupa teks digitaldi mana terdiri dari huruf A besar sampai z kecil dan beberapa tanda baca.

Pengujian *Whitebox* testing untuk perangkat lunak ini adalah dengan melihat atau menguji presentase kebenaran dari keluaran yang dihasilkan. Apabila presentase kebenaran bernilai sangat kecil misalnya di bawah 10%, maka perlu diadakan kajian ulang baik terhadap *syntax* program yang memuat metode-metode yang digunakan di dalam program ini. Adapun *syntax* program untuk metode yang perlu diuji yaitu: *syntax* pada proses *thresholding*, *syntax* pada proses *cropping* dan normalisasi, *syntax* pada proses ekstraksi ciri, dan *syntax* pada metode *backpropagation*. Selain itu diperlukan juga untuk memeriksa *database* yang digunakan.

4.7 Pengambilan Kesimpulan

Tahap ini adalah proses guna menjawab rumusan masalah beserta tujuan penelitian yang disesuaikan dengan hasil penelitian yang telah didapatkan.



BAB V

HASIL DAN PEMBAHASAN

Uraian hasil penelitian dan pembahasan akan diuraikan pada bab ini. Meliputi penggunaan dari aplikasi yang sudah dibuat. Uraian aplikasi yang telah di buat meliputi tampilan aplikasi, fungsi kontrol pada aplikasi, proses penyusunan perangkat lunak aplikasi (*coding*), pengujian (*testing*) dan cara penggunaanya serta akan dijelaskan tentang penggunaan setiap sistem menu tampilan dan fungsi aplikasi

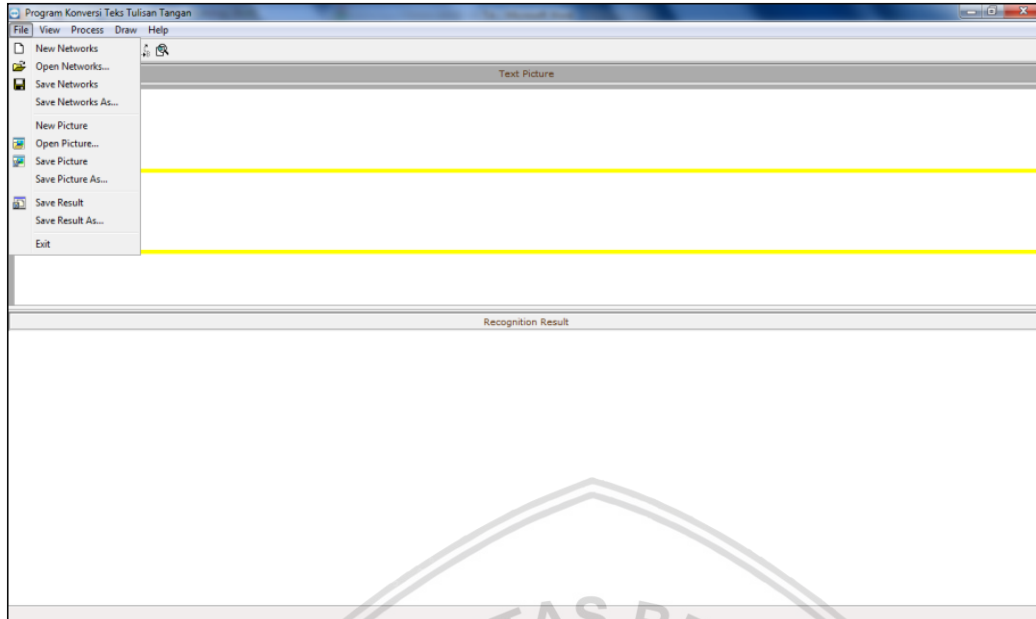
5.1 Implementasi Aplikasi

Aplikasi Konversi Tulisan Tangan ke Teks Digital ini dibuat menggunakan bahasa pemrograman Delphi 10. Selain dilakukan implementasi logika aplikasi, diperlukan juga implementasi tatap muka pengguna sehingga aplikasi lebih mudah dipahami oleh pengguna nantinya. Berikut ini adalah beberapa tampilan program, prosedur dan fungsi program yang digunakan di dalam aplikasi ini.

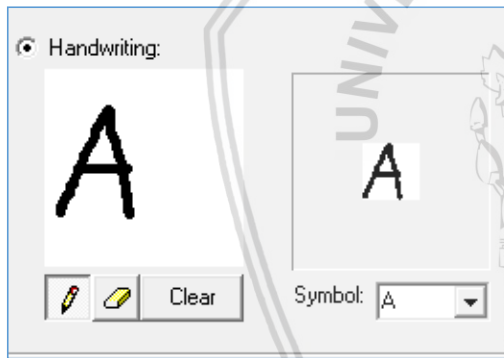
5.1.1 Implementasi Proses Pembuatan dan Masukan Data

Pada proses ini dilakukan pembuatan data yang digunakan sebagai masukan untuk program. Di mana teknik dalam proses pengambilan datanya melalui 2 cara yaitu melalui pengambilan langsung *file* di dalam komputer, maupun dengan cara pembuatan langsung data di dalam aplikasi. Untuk pembuatan data di dalam aplikasi juga dapat dilakukan dengan 2 cara yaitu : pengguna dapat langsung membuat menggunakan mouse secara *real time*, maupun pengguna dapat memasukan data berdasarkan *font* yang telah di sediakan. Untuk tampilan pengguna dapat dilihat pada Gambar 5.1.

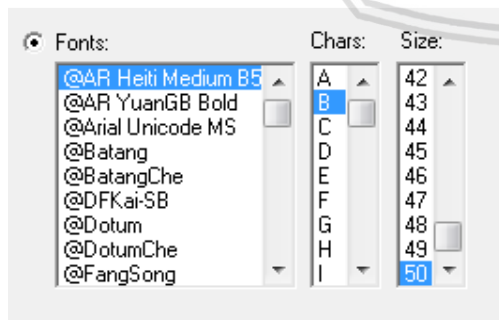
Pertama-tama pengguna dapat memilih sendiri cara citra uji dimasukan, dapat melalui input langsung dari *file* citra tulisan tangan di komputer (dengan format *.bmp) ataupun pengguna dapat menuliskan langsung tulisan tangan yang ingin diuji dengan menulis pada kanvas yang telah disediakan di dalam aplikasi (*real time*). Setelah itu sebelum kepada proses pengenalan atau konversi, pengguna terlebih dahulu harus memasukan data citra referensi. Di mana data citra referensi yang dimasukan dapat melalui *font* yang telah disediakan di dalam aplikasi ataupun dengan cara menuliskan langsung karakter yang diinginkan untuk digunakan sebagai citra referensi dapat dilihat pada Gambar 5.2 dan Gambar 5.3.



Gambar 5.1 Tampilan form utama.



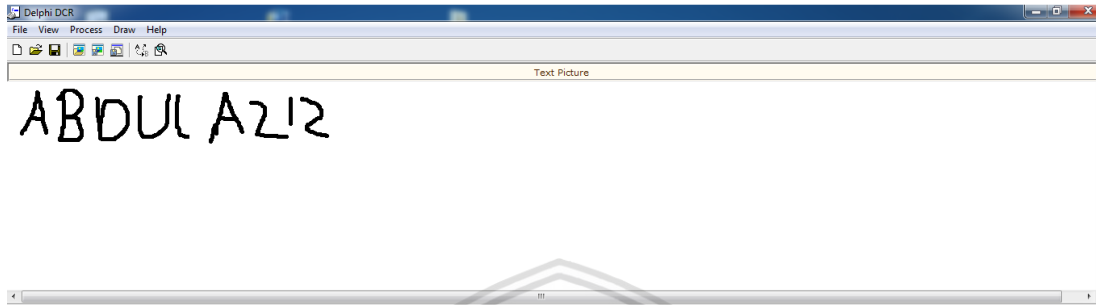
Gambar 5.2 Form input data referensi dengan menulis langsung.



Gambar 5.3 Form input data referensi melalui font.

Gambar 5.4 adalah tampilan form untuk memasukan *file* citra eksternal. Format citra yang dapat diterima oleh sistem adalah format citra bitmap(*.bmp). Diharapkan dengan

menggunakan format citra bitmap kualitas citra yang dihasilkan hampir menyerupai aslinya. *Input* citra ini bisa disisipkan di awal ataupun setelah melakukan data *training* dan *testing*. Namun sebaiknya disisipkan di awal agar kita mengetahui kira-kira karakter apa saja yang akan digunakan sehingga mempercepat proses *training* dan *testing*nya.



Gambar 5.4 Contoh tulisan tangan pada kanvas aplikasi.

Setelah data dimasukkan, terlebih dahulu data masukan akan di-*preprocessing*. Di mana tahapan-tahapan dalam *preprocessing* adalah sebagai berikut.

5.1.2 Thresholding

Proses *thresholding* memiliki tujuan untuk mencari nilai biner pada bit setiap *pixel* yang pada akhirnya difungsikan untuk fungsi selanjutnya pada proses segmentasi citra. Proses *thresholding* juga diterapkan proses *grayscale* jika citra yang digunakan adalah citra berwarna. Pada dasarnya, proses *thresholding* ini adalah sebuah langkah yang menggolongkan nilai *grayscale* 8 bit menuju nilai hitam-putih 1 bit.



Gambar 5.5 Contoh sebuah citra *grayscale* yang akan dikenai operasi *thresholding*.

Sebuah nilai yang disebut nilai ambang (*threshold*) ditentukan untuk menjadi batas antara putih dan hitam. Cara paling sederhana adalah dengan menggunakan nilai 50% sebagai nilai ambang batas, yang dalam hal ini merupakan nilai 127 (setengah dari 255). Gambar 5.5 menunjukkan citra *grayscale* yang menjadi contoh sumber.



Gambar 5.6 Hasil operasi *thresholding* citra dari Gambar 5.5.

Menggunakan nilai ambang 50%, maka operasi *thresholding* pada Gambar 5.5 akan menghasilkan citra seperti pada Gambar 5.6. Proses *thresholding* ini hanya dijalankan pada proses konversi teks, di mana persamaan yang digunakan adalah Persamaan (2-7).

5.1.3 Proses Segmentasi

Proses Segmentasi tujuannya adalah untuk menghilangkan area-area yang bukan termasuk dalam sebuah tulisan. Untuk mendapatkan setiap koordinat pixel proses ini menggunakan operasi aras titik dengan mencari titik sudut bawah, atas, kanan dan kiri dari tulisan. Dalam penerapan proses segmentasi *image*, setiap huruf akan dikenali jika perlakuan deteksi titik sudut menunjukkan titik koordinat *pixel* pada setiap sudut huruf. Ada 3 langkah dalam proses segmentasi untuk menentukan *image* setiap huruf.

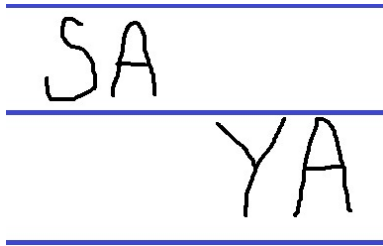
Langkah pertama adalah deteksi segmentasi baris untuk mendeteksi pada titik-titik koordinat *pixel* pada sudut bawah hingga sudut atas dalam pemisahan tulisan dengan latar belakangnya. *Input* untuk proses ini adalah matriks dari proses sebelumnya, seperti pada Persamaan (5-1).

$$f_b(i,j) = \begin{cases} 1 & f(i,j) < T \\ 0 & f(i,j) \geq T \end{cases} \dots\dots\dots (5-1)$$

Adapun langkah dalam proses segmentasi baris ini adalah sebagai berikut :

1. Inisialisasi $j=0$, jumlah baris = 0, kemudian carilah titik atas dari citra uji yang diinisialkan dengan YT.
2. Jika $f_b(i,j) = 1$ maka $YT = jT$.
3. Jika $YT = jT$ maka jumlah baris = jumlah baris + 1 dan cari titik bawah citra yang dilambangkan dengan YB.
4. Jika $YB = jB$ maka tinggi baris = $YB - YT$ dan ulangi langkah 2-3.
5. Selama jumlah baris ≥ 1 , ulangi langkah 2-4.





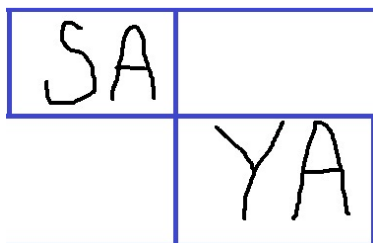
Gambar 5.7 Citra yang akan dikenai proses segmentasi.

Langkah kedua adalah pendeteksian segmentasi setiap kata untuk mendeteksi titik-titik koordinat *pixel* dari sudut kanan hingga sudut kiri dengan nilai parameter yang digunakan adalah lebar spasi yang memisahkan setiap kata. Karena setiap spasi antara suatu kata memiliki ukuran yang lebih lebar dibanding spasi antar huruf di dalam kata. Untuk masukannya diambil dari proses segmentasi baris yaitu seperti pada Persamaan (5-2).

$$f_b(i, \text{tinggi baris}) = \begin{cases} 1 & f(i, j) < T \\ 0 & f(i, j) \geq T \end{cases} \dots\dots\dots (5-2)$$

Adapun langkah prosesnya adalah sebagai berikut :

1. Inisialisasi $i = 0$, jumlah kata = 0, tinggi baris = j , kemudian cari titik kiri yang dilambangkan dengan XL .
2. Jika $fb(i, j) = 1$ maka $XL = iL$.
3. Jika $XL = iL$ maka carilah titik kanannya yang dilambangkan dengan XR .
4. Jika $XR = iR$ maka lebar kata = $XR - XL$ dan ulangi langkah 2-4.
5. Jika $[XR_{(\text{jumlahkata}+1)} - XL_{(\text{jumlahkata})}] \geq \text{spasi}$ maka $\text{jumlahkata} = \text{jumlahkata} + 1$ dan ulangi langkah 2-4.
6. Selama $\text{jumlahkata} \geq 1$, ulangi langkah 2 sampai 5.



Gambar 5.8 Hasil segmentasi dari citra pada Gambar 5.7.

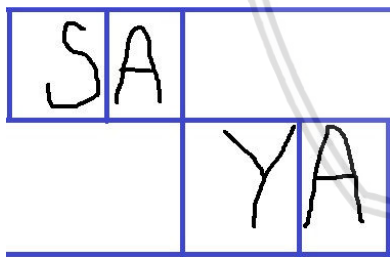


Pendeteksian segmentasi huruf dilakukan untuk mendeteksi titik-titik koordinat sudut kanan, kiri, atas dan bawah pada kata-kata yang telah terpisah, kemudian setiap huruf pada kata akan dipisahkan. Input untuk segmentasi huruf ini adalah seperti pada Persamaan (5-3).

$$f_b(\text{lebar kata}, \text{tinggi baris}) = \begin{cases} 1 & f(i, j) < T \\ 0 & f(i, j) \geq T \end{cases} \dots\dots\dots (5-3)$$

Berikut ini adalah langkah-langkahnya :

1. Inisialisasi IDkata = 0, jumlahhuruf = 0, lebarkata = i, tinggibar = j, kemudian cari titik kiri (XL), cari titik atas (YT).
2. Jika jumlahkata ≥ 1 maka IDkata = IDkata + 1.
3. Jika fB (i,j) = 1 maka XL = iL dan YT = jT dan cari titik kanan (XR) dan cari titik bawah (YB).
4. Jika XR = iR dan YB = jB maka jumlahhuruf = jumlahhuruf + 1 dan lebarhuruf = XR – XL dan tinggihuruf = YB – YT
5. Jika jumlahhuruf ≥ 1 maka ulangi langkah 3-4.
6. Selama IDkata ≥ 1, ulangi langkah 3 – 5.
7. Setelah semua titik koordinat piksel setiap huruf telah diketahui, proses dilanjutkan ke proses cropping dan normalisasi.



Gambar 5.9 Hasil segmentasi huruf dari citra pada Gambar 5.8.

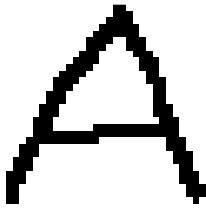
Hasil dari segmentasi ini selanjutnya akan diproses pada proses *cropping* sehingga didapatkan citra untuk masing-masing karakter.

5.1.4 Cropping dan Normalisasi

Pada tahap ini dilakukan proses pemotongan citra dan penskalaan citra ke ukuran normal (normalisasi). Proses *cropping* ini bertujuan untuk memotong citra hasil dari proses segmentasi sebelumnya. Dari proses ini akan didapatkan citra dari setiap karakter yang



diisikan. Kemudian citra hasil *cropping* ini akan dinormalisasi untuk mengembalikannya ke ukuran normal dalam program ini ukuran normal citra yang digunakan adalah 30×30 piksel. Gambar 5.9 menunjukkan hasil *cropping* dan normalisasi huruf A terdapat dalam Gambar 5.8.



Gambar 5.10 Hasil operasi *cropping* dan normalisasi huruf A dari Gambar 5.8.

Sehingga penskalaan dalam proses normalisasi ini menggunakan Persamaan (2-12) dan Persamaan (2-13) yaitu:

1. Untuk penskalaan lebar digunakan $x' = S_x \cdot x$
2. Untuk penskalaan panjang atau tinggi citra digunakan $y' = S_y \cdot y$

Pada aplikasi ini melalui form *cropping* dan normalisasi dapat dilihat 3 informasi citra yaitu: citra dalam keadaan normal sebelum proses *cropping*, citra setelah di-*cropping* dan citra hasil normalisasi.

5.1.5 Reduksi

Proses reduksi adalah proses akhir dalam tahap *preprocessing* sebelum citra dapat dijadikan masukan untuk proses pengenalan menggunakan *backpropagation*. Proses ini bertujuan untuk menurunkan jumlah total piksel yang akan dipakai sebagai data input pada input neuron dari 900 menjadi 600 piksel. Pada proses ini, awalnya terbentuk sebuah array atau larik dua dimensi dengan ukuran 30×30 , yang mana setiap elemennya terdiri dari bilangan boolean yang hanya memiliki kemungkinan nilai 0 ataupun 1.

Kemudian dari piksel vertikal yang ada, maka dilakukan penggabungan sebagian sehingga jumlah total piksel berkurang. Proses ini adalah dengan menggabung setiap 2 piksel vertikal menjadi 1 piksel untuk 6 piksel vertikal pertama yang menghasilkan 3 piksel vertikal, dan selanjutnya untuk 3 piksel sisanya dilompati untuk dibiarkan seperti aslinya. Total dari setiap 9 piksel vertikal akan tereduksi menjadi 6 piksel vertikal. Proses penggabungan 2 piksel menjadi 1 piksel akan melibatkan operasi seperti aturan berikut:

1. Bila kedua piksel sama-sama bernilai 0, maka piksel gabungan akan bernilai 0.

2. Bila kedua piksel sama-sama bernilai 1, maka piksel gabungan akan bernilai 1.
3. Bila kedua piksel berbeda nilai, maka piksel gabungan akan diberi nilai random (acak) antara 0 dan 1.

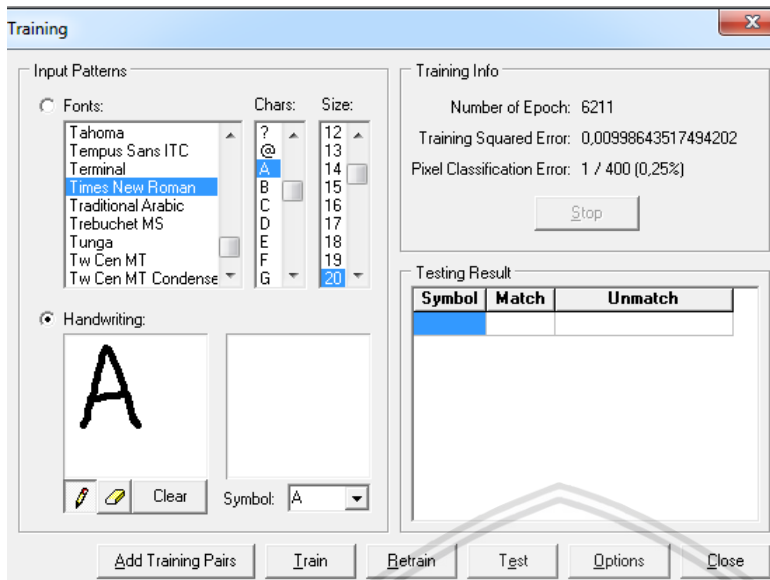
Hasil dari proses reduksi ini akan menghasilkan matriks 20×30 yang masing-masing elemennya bernilai biner dan akan dipakai sebagai masukan pada *input neuron*, dengan jumlah total elemennya adalah sebanyak 600.

5.1.6 Back Propagation

Algoritma *back propagation* di sini digunakan untuk proses pengenalan karakter, di mana pada *back propagation* terdapat 2 tahap yaitu: tahap pertama adalah proses training, kemudian tahap kedua adalah proses testing. Pada proses training ditujukan untuk memperoleh nilai bias dan bobot yang selanjutnya akan digunakan untuk proses *testing*. Proses *testing* bertujuan untuk menguji apakah dari bias dan bobot yang dihasilkan oleh proses *training* sebelumnya benar atau mendekati kebenaran hasil dari aplikasi. Apabila bobot dan bias yang dihasilkan dari proses *training* sebelumnya dirasa belum cukup maka dilakukan proses *training* ulang (*retrain*) untuk mendapatkan bobot dan bias baru yang lebih sesuai.

5.1.7 Training untuk Back Propagation

Pada proses pelatihan *back propagation* tujuannya untuk mencari bobot pada setiap pasang data karakter tulisan. Sehingga nantinya diharapkan sistem mampu mengenali data citra uji yang diberikan. Diharapkan pada proses *training* ini citra yang diinputkan sebanyak-banyaknya sehingga proses pengenalan karakter di dalam aplikasi lebih baik. Semakin banyak citra yang dimasukkan dalam proses pembelajaran (*training*), semakin banyak pula system memiliki data referensi. Terlebih lagi data karakter tulisan yang dimiliki untuk masing-masing karakter mempunyai jenis baik itu *font* maupun ukuran yang berbeda-beda. Dengan demikian proses pengenalan karakter nantinya akan menjadi lebih baik. Hasil *training back propagation* ditunjukkan pada Gambar 5.10.

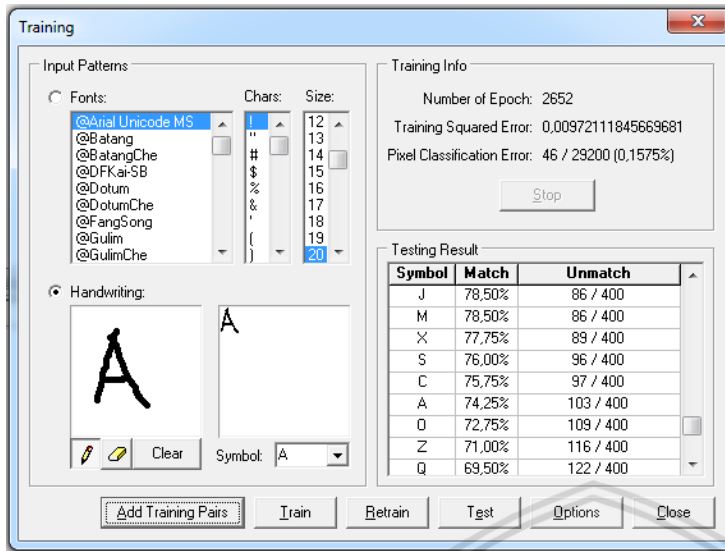


Gambar 5.11 Form training backpropagation.

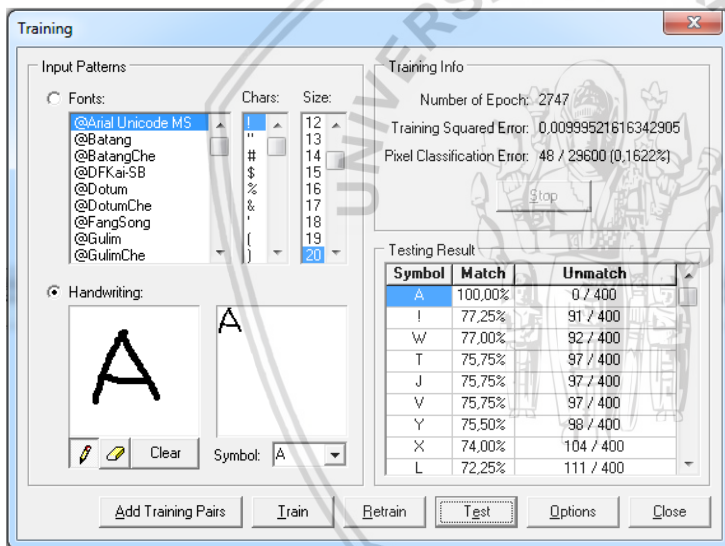
Proses *training* ini akan menghasilkan bobot dan bias untuk proses *testing* selanjutnya. Apabila hasil dari bobot dan bias yang dihasilkan oleh proses *training* pertama dirasa kurang baik, maka pengguna dapat melakukan proses *training* ulang dengan menekan tombol ‘retrain’ sampai didapatkan bobot dan bias yang sesuai. Selanjutnya proses *training* untuk semua huruf dan angka telah selesai proses selanjutnya adalah menguji aplikasi, sudah sesuai dengan yang diujikan atau belum.

5.1.8 Testing untuk Back Propagation

Proses *testing* dilakukan setelah proses *training* berakhir, di mana proses *testing* ini akan menggunakan bias dan bobot yang dihasilkan dari proses *training* sebelumnya. Apabila hasil dari proses *testing* ini dirasa kurang baik maka pengguna dapat melakukan *training* ulang untuk mendapatkan bobot dan bias yang baru. Hasil *training* yang berulang-ulang dapat memberikan nilai yang lebih bagus dibandingkan dengan *training* yang dilakukan satu kali. Hasil *testing backpropagation* pada Gambar 5.11 dilakukan 1 kali mendapatkan hasil kemiripan dengan huruf aslinya 74,25%. Pada Gambar 5.12 dilakukan *training* 3 kali mendapatkan hasil kemiripan dengan huruf aslinya 100%.

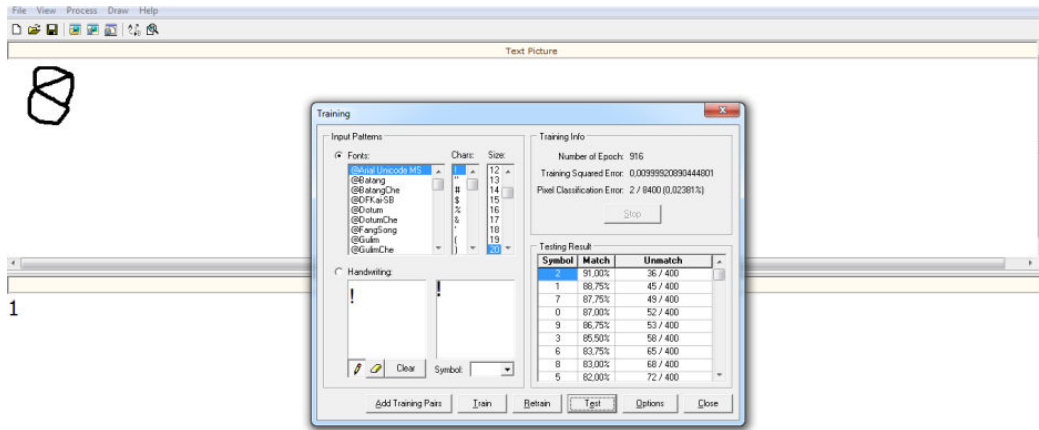


Gambar 5.12 Hasil testing backpropagation ke-1 pengenalan huruf.



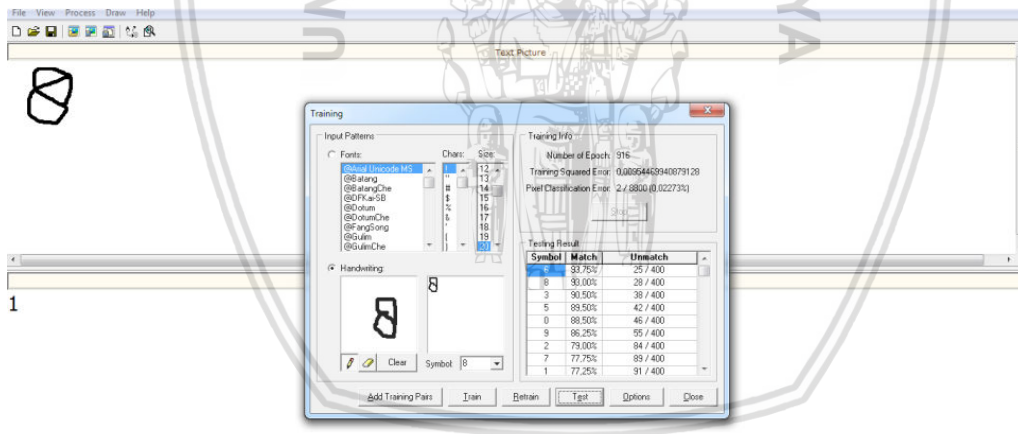
Gambar 5.13 Hasil testing backpropagation ke-3 pengenalan huruf.

Pengenalan angka yang sulit terletak pada pengenalan angka 8, pada Gambar 5.13 ditunjukkan hasil *training* dari uji pertama hasil kemiripan tulisan adalah 83,00% sehingga angka delapan yang dikenali angka 1.



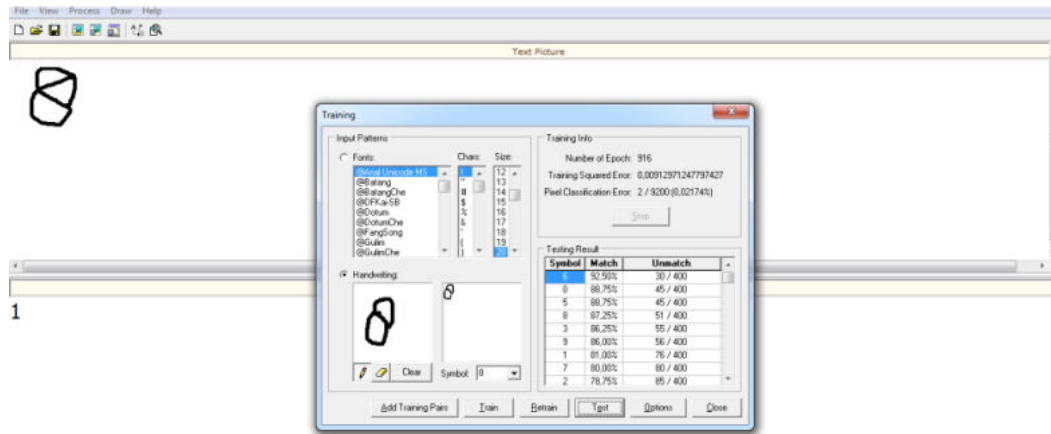
Gambar 5.14 Hasil testing backpropagation ke-1 pengenalan angka.

Gambar 5.14 menunjukkan hasil *training* ke-2, kemiripan tulisan memiliki prosentase 93,00%. kemiripan tulisan pada prosentase 93,00% belum bisa mengenali masih dikenali angka 1.



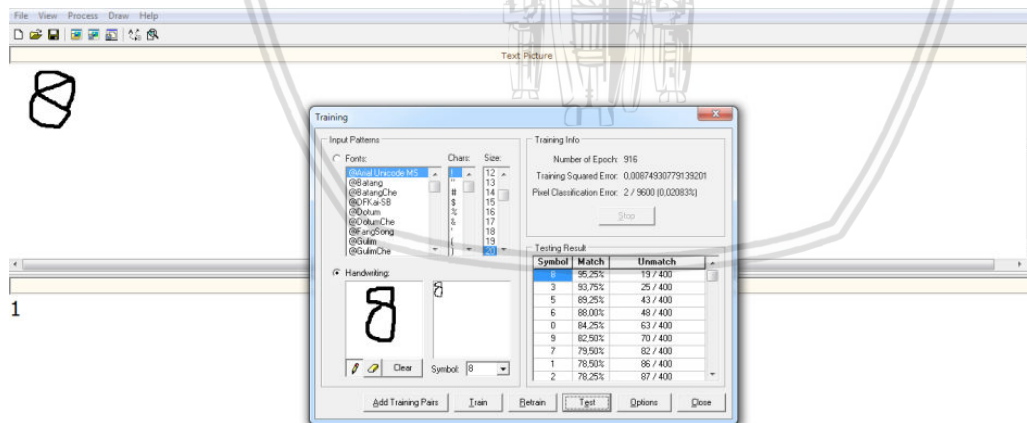
Gambar 5.15 Hasil testing backpropagation ke-2 pengenalan angka.

Gambar 5.15 menunjukkan hasil uji yang ke-3, pada pengenalan yang ke-3 ini mengalami peneurunan prosentase kemiripan tulisan yaitu menjadi 92,50% hal ini dikarenakan dilakukan percobaan perbedaan arah penulisan pada angka 8 sehingga masih dikenali angka 1.

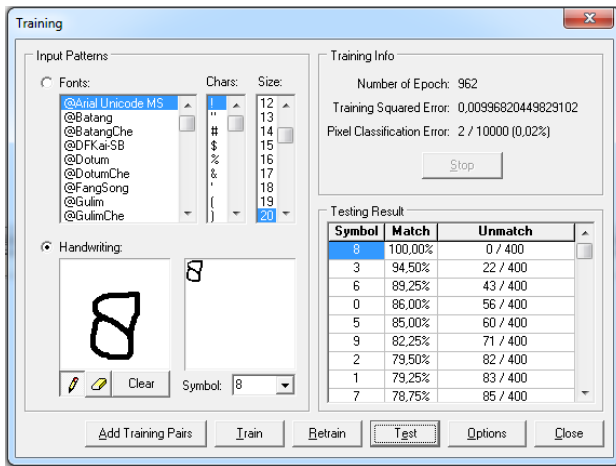


Gambar 5.16 Hasil testing *backpropagation* ke-3 pengenalan angka.

Gambar 5.16 adalah hasil uji pengenalan tulisan angka 8 yang ke-4, hasil kemiripan tulisan tangan mengalami peningkatan prosentase yaitu 95,25%. Pada prosentase ini angka delapan masih dikenali angka 1. Berarti di sini perlu dilakukan uji ulang supaya mendapatkan prosentase keberhasilan 100%, sehingga dapat dikenali menjadi angka delapan. Perubahan tersebut dapat dilihat pada Gambar 5.17 di mana prosentase kemiripan angka 8 prosentasenya mencapai 100% yang hasilnya ditunjukkan pada Gambar 5.18.



Gambar 5.17 Hasil testing *backpropagation* ke-4 pengenalan angka.



Gambar 5.18 Hasil testing backpropagation ke-5 pengenalan angka.



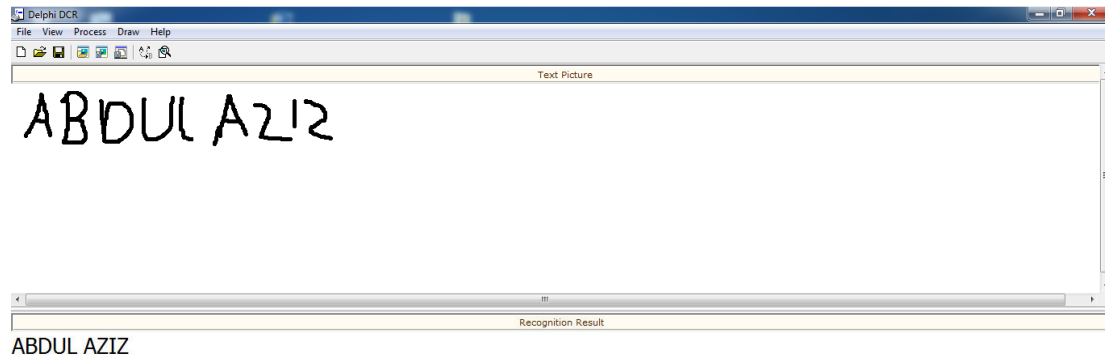
Gambar 5.19 Hasil testing backpropagation ke-5 pengenalan angka.

Percobaan angka 8 yang dilakukan sebanyak 5 kali dengan tulisan yang sama yang dikenali angka 1 karena penulisan awal dari angka 8 mirip dengan angka 1. Yaitu terdapat garis miring pada lingkaran atas angka 8, sehingga dikenali angka 1.

5.1.9 Proses Pengenalan

Proses pengenalan karakter dapat dilakukan setelah *system* mempunyai data karakter tulisan yang telah diuji melalui proses *training* dan *testing* sebelumnya. Apabila *system* tidak mempunyai data karakter tulisan maka *system* tidak dapat mengenali citra tulisan tangan yang dimasukan. Proses pengenalan karakter ini sangat bergantung terhadap citra tulisan yang dimasukan, semakin banyak kemiripan antara citra tulisan dan citra uji maka semakin

baik pula *system* dalam mengenali pola masing-masing huruf. Hasil *recognize* citra tulisan tangan terdapat pada Gambar 5.7.

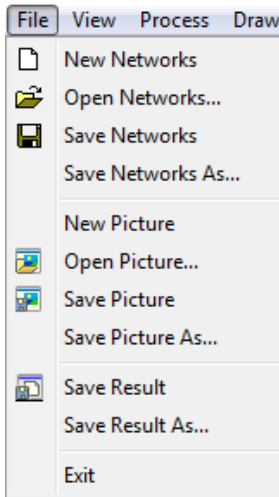


Gambar 5.20 Hasil *recognize* citra tulisan tangan.

5.1.10 Proses Penyimpanan

Perlu diketahui bahwa program ini tidak memiliki *database*, sehingga setiap proses dalam memasukan data tulisan sistem tidak secara otomatis menyimpan data tulisan tersebut. Agar data tulisan yang telah dimasukan tidak hilang pengguna perlu menyimpan data tersebut secara manual melalui menu \rightarrow *save network* atau *save network as*. Sehingga untuk dikemudian hari dalam proses pengenalan karakter selanjutnya, tidak diperlukan memasukan data tulisan secara ulang. Dengan catatan karakter citra data referensi sesuai dengan data citra uji. Misalnya data citra uji yang dimasukan mempunyai karakter huruf 'A' maka data citra tulisan yang dimiliki *system* harus mempunyai huruf 'A'. Jika tidak, maka proses pengenalan karakter akan menjadi tidak baik.

Selain penyimpanan untuk data tulisan, diperlukan juga penyimpanan untuk hasil dari pengenalan karakter yang dilakukan. Sehingga pengguna tidak perlu untuk copy – paste hasil pengenalan lagi. Untuk penyimpanan hasil, pengguna cukup menekan menu \rightarrow *save result* atau *save result as*. Menu penyimpanan terdapat pada Gambar 5.20.



Gambar 5.21 Menu penyimpanan pada aplikasi.

5.2 Uji Validasi Aplikasi

Percobaan untuk mendapatkan data hasil uji dilakukan 3 kali, dengan uji coba aplikasi ini dilakukan dengan memasukan 12 data uji. Di mana setiap 1 siswa menulis 3 kali dan data citra uji dengan penulisan secara *real time* di dalam aplikasi. Pada pengujian pertama hasil prosentasinya tidak bagus disebabkan pada penulisan pertama masih belum mengenal aplikasi yang dibuat. Pada pengujian kedua ada peningkatan hasil karena sudah ada proses belajar anak pada pengujian pertama. Hasil pengujian ketiga akan ditunjukkan pada tabel 5.1 sampai dengan 5.6. Data pengenalan tulisannya digunakan sebanyak 660 data referensi yang terdiri dari 10 jenis huruf untuk masing-masing karakter dari “ A - Z”, “a - z” dan “0 - 9”. Uji data citra pengenalan pola tulisan huruf kapital terdapat pada Tabel 5.1.

Tabel 5.1
Uji Data Citra Pengenalan Pola Tulisan Huruf Kapital

No	Karakter	Jenis	Tampilan
1	ABDUL AZIZ	Real time	ABDUL AZIZ
2	ZUDA BACHTIAR	Real time	ZUDA BACHTIAR
3	AIDA SAFITRI	Real time	AIDA SAFITRI
4	CINTA MUTIARA VALENT	Real time	CINTA MUTIARA VALENT
5	CHIARA FARDAH	Real time	CHIARA FARDAH
6	CITRA HARTOMO	Real time	CITRA HARTOMO
7	SARA HAYA GHOZALA	Real time	SARA HAYA GHOZALA
8	LINTANG FAIZA	Real time	LINTANG FAIZA
9	GILANG BAGASKARA	Real time	GILANG BAGASKARA

No	Karakter	Jenis	Tampilan
10	SURYA SAPUTRA	Real time	SURYA SAPUTRA
11	ALVIANA MUAWANAH	Real time	ALVIANA MUAWANAH
12	ABI BEKTI SYAPUTRA	Real time	ABI BEKTI SYAPUTA

Pada Tabel 5.1 dijelaskan *input* berupa nama siswa PAUD Iklas Nurani yang diambil secara *real time*. Dalam aplikasi ini isi tabel tampilan adalah hasil tulisan anak yang ditulis dalam kanvas aplikasi. Pada baris 12 dengan nama Abi Bektı Saputra anak menulis terdapat satu huruf yang belum ditulis sehingga menjadi *ABI BEKTI SYAPUTA*.

Setelah menyiapkan citra referensi dan citra uji, maka proses selanjutnya adalah dilakukan pengujian, yaitu dengan cara memasukan citra uji ke dalam aplikasi. Dan hasil yang didapatkan dari aplikasi berdasarkan citra uji yang dimasukan terdapat pada Tabel 5.2.

Tabel 5.2

Tabel Hasil Pengujian Citra Uji Huruf Kapital dan Angka

No	Input	Hasil	Presentase Pengenalan
1	ABDUL AZIZ	ABDUL AZIZ	$\frac{9}{9} \times 100\% = 100$
2	ZUDA BACHTIAR	YU CA DYCYTI YI	$\frac{5}{12} \times 100\% = 41,67$
3	CIARA FARDAH	AICA CWFITKI	$\frac{7}{11} \times 100\% = 63,64$
4	CINTA MUTIARA VALENT	CIYTA MQTIARA VALEHT	$\frac{15}{18} \times 100\% = 83,33$
5	CIARA FARDAH	CIAFA FAROA H	$\frac{9}{11} \times 100\% = 81,81$
6	CITRA HARTOMO	RITRP PARTOMC	$\frac{8}{12} \times 100\% = 66,67$
7	SARA HAYY GHUZALA	DAZJ HAYW CJRZYLA	$\frac{7}{15} \times 100\% = 46,67$
8	LINTANG KAIZA	LIRTAYC KAIZA	$\frac{8}{12} \times 100\% = 66,67$
9	GILANG BAGASKARA	YILAWC FAPASWAYZ	$\frac{7}{15} \times 100\% = 46,67$
10	SURYA SAPUTRA	SUTY A SAPUFXP	$\frac{8}{12} \times 100\% = 66,67$
11	ALVIANA MUAWANAH	TLVIAWA KUAWAIAH	$\frac{11}{15} \times 100\% = 73,33$
12	ABI BEKTI SYAPUTA	FBI DFK TI YYAPUTA	$\frac{11}{15} \times 100\% = 73,33$

No	Input	Hasil	Presentase Pengenalan
	Rata - rata		61,4275%

Hasil prosentase keberhasilan penulisan pada setiap anak berbeda. Prosentase keberhasilan tertinggi adalah 100% dengan nama Abdul Aziz dan prosentase keberhasilan terendah adalah 41,67%. Selanjutnya dilakukan perhitungan prosentase rata-rata keberhasilan pengenalan tulisan dari 12 siswa. Persentase pengenalan didapatkan dari Persamaan (5-1).

$$P = \frac{b}{s} \times 100\% \dots\dots\dots (5-1)$$

Dengan:

- P = Persentase pengenalan
- b = Jumlah karakter benar
- s = Jumlah keseluruhan karakter

Sehingga rata-rata pengenalan program dari 12 data uji yang diambil adalah seperti pada Persamaan (5-2) hingga (5-4).

$$P_3 = \frac{\sum_{n=1}^{12} \frac{b}{s} \times 100\%}{n} \dots\dots\dots (5-2)$$

$$P_3 = \frac{737,13\%}{12} \dots\dots\dots (5-3)$$

$$P_3 = 61,4275\% \dots\dots\dots (5-4)$$

Dari rata-rata akurasi program maka dapat disimpulkan bahwa program baik dalam proses pengenalan karakter karena memiliki prosentase keberhasilan lebih dari 50%. Hanya saja karakter-karakter yang mempunyai kemiripan bentuk sedikit susah untuk diprediksi. Apalagi dengan jenis tulisan tangan yang bentuk dan jumlahnya sangat beraneka ragam. Hal ini dapat menjadi kendala dalam pengenalan karakter, maka dari itu perlu juga diberikan batasan yang jelas dan spesifik untuk masing-masing karakter. Misalnya saja karakter “0”, “O”, dan “o” memiliki bentuk dan susunan matriks yang hampir sama, di sinilah diperlukan batasan tersebut. Misalnya, untuk karakter angka ”0” tulisan tangan harus dibuat lebih memanjang, untuk karakter huruf “ O” tulisan tangan dibuat lebih melebar.

Selanjutnya dilakukan uji data citra pengenalan pola tulisan huruf kecil terdapat pada Tabel 5.3.



Tabel 5.3
Uji Data Citra Pengenalan Pola Tulisan Kecil

No	Karakter	Jenis	TAMPILAN
1	abdul aziz	Real time	abdul aziz
2	zuda bachtiar	Real time	zuda bachtiar
3	aida safitri	Real time	aida safitri
4	cinta mutiara valent	Real time	cinta mutiara valent
5	chiara fardah	Real time	chiara fardah
6	citra hartomo	Real time	citra hartomo
7	sara haya ghozala	Real time	sara haya ghozala
8	lintang faiza	Real time	lintang faiza
9	gilang bagaskara	Real time	gilang bagaskara
10	surya saputra	Real time	surya saputra
11	alviana muawanah	Real time	alviana muawanah
12	abi bekti syaputra	Real time	abi bekti syaputra

Pada Tabel 5.3 dijelaskan *input* berupa nama siswa PAUD Iklas Nurani yang diambil secara *real time*. Dalam aplikasi ini isi tabel tampilan adalah hasil tulisan anak yang ditulis dalam kanvas aplikasi berbentuk huruf kecil. Padapenulisan huruf kecil ini tidak ditemukan kekurangan huruf dalam peulisan nama seperti pada Tabel 5.1 baris 12.

Setelah menyiapkan citra referensi dan citra uji, maka proses selanjutnya adalah dilakukan pengujian, yaitu dengan cara memasukan citra uji ke dalam aplikasi. Dan hasil yang didapatkan dari aplikasi berdasarkan citra uji yang dimasukan terdapat pada Tabel 5.4.

Tabel 5.4
Tabel Hasil Pengujian Citra Uji Huruf Kecil

No	Input	Hasil	Presentase Pengenalan
1	abdul aziz	audul azlz	$\frac{8}{9} \times 100\% = 88,89$
2	zuda bachtiar	zuda tacttrar	$\frac{9}{12} \times 100\% = 75$
3	aida safitri	rida safitri	$\frac{10}{11} \times 100\% = 90,91$

No	Input	Hasil	Presentase Pengenalan
4	Cinta Muklara Valente	cinta mutiurc valutt	$\frac{14}{18} \times 100\% = 77,78$
5	Ciarafardah	ciarafardrh	$\frac{10}{11} \times 100\% = 90,91$
6	Citra Hartomo	citra hartcmc	$\frac{10}{12} \times 100\% = 83,33$
7	Sara Haya Ghozala	sarg rayg ghozgrs	$\frac{9}{15} \times 100\% = 60$
8	Lintang Faiza	lintang faiza	$\frac{12}{12} \times 100\% = 100$
9	Gilang Bagaskara	gilang kagaskbra	$\frac{13}{15} \times 100\% = 86,67$
10	Surya Saputra	surya saputra	$\frac{12}{12} \times 100\% = 100$
11	Alfiana Muawah	alfiafa nlawaf	$\frac{9}{13} \times 100\% = 69,23$
12	Abi Bekti Saputra	aki tekti sapbtre	$\frac{11}{15} \times 100\% = 73,33$
Rata-rata			83,0042%

Tabel 5.4 menunjukkan hasil prosentase keberhasilan tertinggi mencapai 100% dengan nama Lintang Faiza dan Surya Saputra dan prosentase terendah adalah 60% dengan nama Sara Haya Ghozala.

Selanjutnya dihitung rata-rata pengenalan dari 12 data uji yang diambil. Hasil dari rata-rata akurasi pengenalan tulisan huruf kecil cukup signifikan karena dari 61,4275% menjadi 83,0042%. Maka dapat disimpulkan bahwa program lebih baik dalam proses pengenalan karakter tulisan huruf kecil. Permasalahan sama yaitu kemiripan tulisan menyebabkan tulisan tidak dikenali sebagaimana mestinya. Dalam hal ini perlu adanya spesifikasi tulisan yang lebih jelas dan spesifik dalam proses pelatihan.

Pada Tabel 5.5 dijelaskan uji hasil tulisan tangan dalam bentuk angka dalam kanvas aplikasi. Di mana satu siswa menulis angka 0-9 sebanyak 3 kali percobaan. Pada Tabel 5.5 berikut ini adalah hasil percobaan ketiga. Terdapat nama siswa, karakter tulisan yang ditulis dan pada kolom tampilan adalah hasil dari percobaan penulisan dalam kanvas aplikasi.

Tabel 5.5
Uji Data Citra Pengenalan Pola Tulisan Angka

No	Nama	Karakter	Jenis	Tampilan
1	Abdul Aziz	0123456789	Real time	0123456789
2	Zuda Bachtiar	0123456789	Real time	0123456789

No	Nama	Karakter	Jenis	Tampilan
3	Aida Safitri	0123456789	Real time	0123456789
4	Cinta Mutiara Valent	0123456789	Real time	0123456789
5	Chiara Fardah	0123456789	Real time	0123456789
6	Citra Hartomo	0123456789	Real time	0123456789
7	Sara Haya Ghozala	0123456789	Real time	0123456789
8	Lintang faiza	0123456789	Real time	0123456789
9	Gilang Bagaskara	0123456789	Real time	0123456789
10	Surya Saputra	0123456789	Real time	0123456789
11	Alviana Muawanah	0123456789	Real time	0123456789
12	Abi Bakti Syaputra	0123456789	Real time	0123456789
Rata - rata				

Tabel 5.6 adalah hasil uji aplikasi penulisan angka pada kanvas aplikasi dimana terdapat nama berisi nama siswa yang menulis, input berisi hasil tulisan anak dalam kanvas aplikasi, hasil berisi hasil pengenalan tulisan yang dideteksi oleh aplikasi dan yang terakhir prosentase keberhasilan adalah hasil uji aplikasi dalam bentuk persen.

Tabel 5.6
Tabel Hasil Pengujian Citra Uji Angka

No	Nama	Input	Hasil	Prosentase Keberhasilan
1	Abdul Aziz	0123456789	0 1 234 0 6 729	80%
2	Zuda Bachtiar	0123456789	0123056783	80%
3	Aida Safitri	0123456789	0123976729	50%
4	Cinta Mutiara Valent	0123456789	0123756713	70%
5	Chiara Fardah	0123456789	0123456747	80%
6	Citra Hartomo	0123456789	0123926729	60%
7	Sara Haya Ghozala	0123456789	0123436789	90%
8	Lintang faiza	0123456789	01 23456739	90%
9	Gilang Bagaskara	0123456789	0123456709	90%
10	Surya Saputra	0123456789	012345 6769	90%
11	Alviana Muawanah	0123456789	0123926729	70%
12	Abi Bakti Syaputra	0123456789	0123416777	70%

Rata-rata prosentase keberhasilan tulisan angka adalah 75%, prosentase keberhasilan tertinggi 90% dengan nama Sara Haya Ghozala, Lintang faiza, Gilang Bagaskara dan Surya Saputra. Hasil Prosentase terkecil 60% dengan nama Citra Hartomo.

Perbandingan prosentase keberhasilan pada huruf besar A hingga Z, huruf kecil a hingga z, dan angka 0 hingga 9 jauh berbeda hal tersebut dikarenakan proses penulisan huruf kecil



bersambung dan huruf besar tidak bersambung terputus-putus pada setiap penarikan garis vertikal, horisontal, lingkaran dan setengah lingkaran. Hasil perbandingan prosentase keberhasilan penulisan tangan anak usia 4-5 tahun ditunjukkan pada Tabel 5.7.

Tabel 5.7

Tabel Perbandingan Prosentase Keberhasilan Pengenalan Tulisan Huruf Besar, Huruf Kecil, dan Angka

No	Huruf	Prosentase Keberhasilan
1	A-Z	61,4275%
2	a-z	83,0042%
3	0-9	75%

Kesalahan penulisan rata-rata karena anak kurang memahami cara menulis yang baik, tepat dan benar. Cara menulis yang baik, tepat dan benar itu dimulai dari mana anak-anak kurang memahami hal tersebut. Jadi anak-anak hanya asal mencoret dalam kanvas pemrograman tanpa memperhatikan diawali dengan garis vertikal, horisontal, lingkaran dan setengah lingkaran. Sehingga didapat hasil dengan prosentase kebenaran 25% untuk percobaan pertama dan untuk percobaan kedua sudah dapat pengarahan dari admin untuk penulisan yang baik dan benar dan ada peningkatan prosentase keberhasilan menjadi 35%.

Faktor lain yang mempengaruhi keberhasilan anak dalam proses belajar menulis dan membaca adalah latar belakang orang tua. Bagi anak yang orang tuanya menjadi pendidik lebih cepat proses belajar pengenalan terhadap tulisan sedangkan anak yang latar belakang orang tuanya wiraswasta lebih lambat. Faktor selanjutnya adalah umur, bagi anak yang memiliki umur lebih tua juga lebih cepat proses pengenalan terhadap pengenalan angka dibandingkan dengan anak yang lebih muda. Hal ini disebabkan anak dengan usia lebih tua sudah belajar terlebih dahulu dibandingkan dengan yang lebih muda.

5.3 Uji Terapan Siswa Usia 4 Tahun hingga 5 Tahun

Setelah melakukan uji aplikasi, maka dengan aplikasi tersebut dibuat pengujian penilaian hasil tulisan tangan dari siswa PAUD dan TK yang berumur 4-5 tahun. Pengujian tersebut dilakukan setiap hari guna mendapatkan data kemajuan kemampuan penulisan tangan siswa, dengan materi penilaian diklasifikasikan sebagai berikut:

1. Menulis huruf besar dari A hingga Z.
2. Menulis huruf kecil dari a hingga z.
3. Menulis angka dari 0 hingga 9.

4. Menulis nama sendiri menggunakan huruf besar.

Rentang waktu pengujian diambil dari tanggal 11 Juli 2017 hingga 18 Juli 2017 setiap hari aktif sekolah, sehingga total pengujian dilakukan sebanyak 6 kali untuk setiap materi yang sama. Pengujian yang dilakukan pada beberapa tempat ditunjukkan pada Tabel 5.8.

Tabel 5.8

Tabel Pengujian Penilaian Hasil Tulisan Tangan dari Siswa PAUD dan TK yang Berumur 4-5 Tahun

No	Nama Sekolah	Alamat Sekolah	Jumlah Siswa	Rata-rata Peningkatan	Keterangan
1	PAUD Al Hikmah	Bence	38	8,4 %	PAUD +
2	PAUD Al Hidayah 1	Bence	35	8,2 %	Umum
3	PAUD Al Hidayah 2	Bence	32	5,8	Umum
4	TK Al Hikmah	Bence	35	7,9 %	TK +
5	TK Al Hidayah 1	Bence	32	6,3	Umum
6	TK Al Hidayah 2	Bence	35	6,3	Umum
7	TK Al Hidayah	Tawangsari	35	7,9	Umum
8	TK Al Hidayah	Slorok	35	9,1	TK +
9	TK Al Hidayah	Pucungsari	34	7,5	Umum
10	TK Al Hidayah	Plosorejo	35	5,8	Umum
11	TK Aisyiyah Bustanul Athfal	Slorok	35	4,2	Umum
12	TK Tunas Harapan	Garum	32	7,2	Umum
13	TK Al Hidayah	Ngebrak	27	6,1	Umum
14	TK Al Hidayah	Kebonsari	30	5,8	Umum
15	TK Al Amien	Garum	30	6,5	Umum

Dari hasil pengujian pada Tabel 5.8, terdapat data rata-rata peningkatan yang berupa kemajuan setiap siswa dari masing-masing pengujian yang dihitung berdasarkan kemajuan nilai dari hari sebelumnya menuju ke hari berikutnya. Dari semua pengujian di beberapa TK dan PAUD tersebut didapatkan rata-rata 6,63 nilai (dengan skala nilai 0 hingga 100) untuk peningkatan kemampuan siswa dari hari ke hari.

BAB VI

KESIMPULAN DAN SARAN

Berdasarkan perancangan algoritma yang sudah diimplementasikan maka kesimpulan dan saran yang diperoleh dari pembuatan aplikasi media belajar anak dengan menggunakan metode *back propagation* adalah sebagai berikut.

6.1 Kesimpulan

Berdasarkan dari hasil pengujian aplikasi menggunakan metode *back propagation* dan pengujian terhadap siswa TK dan PAUD yang berlokasi di Blitar dapat ditarik kesimpulan sebagai berikut:

1. Sistem untuk proses belajar menulis dan membaca anak usia 4 tahun sampai dengan 5 tahun telah berhasil dibangun dengan menggunakan Borland Delphi 10, berdasarkan kaidah-kaidah yang ada pada metode pengenalan citra JST *back propagation*. Untuk pengujian pada kemampuan siswa telah menunjukkan peningkatan yang didapatkan sebagai rata-rata kemajuan siswa di beberapa TK dan PAUD dari hari ke hari pengujian. Hasil uji aplikasi menunjukkan proses belajar menulis dalam kurun waktu 6 hari (12 Juni 2017 – 17 Juni 2017) anak dapat menulis meski masih perlu pendampingan untuk menulis yang baik dan benar.
2. Pengolahan citra menggunakan parameter citra biner dengan format citra 1 bit berupa tulisan tangan anak usia 4 tahun sampai dengan 5 tahun telah berhasil dilakukan. Meskipun demikian, belum semua huruf dan angka dapat dikenali dengan sempurna, contohnya karena faktor kemiripan seperti huruf “O”, “o” dan “0”. Hal ini membutuhkan spesifikasi pengenalan yang berulang-ulang supaya mendapatkan hasil yang lebih baik. Hasil prosentase keberhasilan pengenalan tulisan huruf kapital 61,4256%. Pengenalan angka prosentase keberhasilannya 75,00% dan pengenalan huruf kecil 83,0042%. Melihat hasil prosentase keberhasilan dari pengenalan tulisan memiliki hasil yang lebih baik pada tulisan huruf kecil hal ini dikarenakan tulisan huruf kecil penulisannya bersambung tidak terputus-putus seperti pada huruf kapital.

3. Dengan memfungsikan jaringan saraf tiruan menggunakan metode *back propagation* untuk menguji akurasi data pengenalan tulisan tangan telah berhasil dilakukan, dengan banyaknya pelatihan yang dilakukan maksimal 10.000 kali.

6.2 Saran

Dalam pengembangan JST menggunakan metode *backpropagation* aplikasi pengenalan pola tulisan tangan masih banyak yang harus dikembangkan untuk penelitian lebih lanjut sehingga mendapatkan hasil yang lebih baik lagi yaitu dengan menambahkan metode lain dalam proses pengenalan tulisan pada proses deteksi tepi. Misalnya penggabungan metode Laplacian dengan metode Canny. Kemudian pada proses pengujian dengan jaringan saraf tiruan dilakukan pula penggabungan metode *back propagation* dengan metode *radial basic function*. Sehingga nantinya penulisan dapat dikenali meski dilakukan dengan cara penulisan yang salah.



DAFTAR PUSTAKA

- Ahmad, B.; F. Kartika. 2005. Teknik Pengolahan Citra menggunakan Delphi. Yogyakarta: Ardi Publishing
- Arifianto, T. 2011. Membuat Interface Aplikasi Android Lebih keren dengan LWUIT. Andi. Yogyakarta.
- Barve, S. 2012. Optical Character Recognition Using Artificial Neural Network. IJATER (2) 2 : 139-142.
- Cheriet, M at al. 2007. Character Recognition Systems A Guide for Students and Practioners. Wiley products. United States of America.
- Cheung, K.; D. Yeung and Chin, R. 1998. A Bayesian Framework for Deformable Pattern Recognition With Application to Handwritten Character Recognition. IEEE. (20) 12 : 1382-1388.
- Harjono. 2013. Pengubahan Tulisan Tangan Kebentuk Text Ketikan Menggunakan Metode Bacpropagation. Politeknosains (12) 2 : 28-37.
- Hermawan, D. S. 2012. Aplikasi Belajar Menulis Untuk Anak Prasekolah Berbasis Android. JSIKA (1) 1 : 1-15.
- Hull, J J.; C Allan and Tin-Kam Ho. 1990. Multiple Algorithms for Handwritten Character Recognition. Buffalo. Bell Hall. 226 : 117-129.
- Izhar, S. 1998. Unit Pendidikan Pra Sekolah. (Online) (http://members.tripod.com/ip_ipda/unit_pra.htm) diakses tanggal 8 Agustus 2013.
- Kannan, R.J.; R. Prabhakar. 2008. An Improved Handwritten Tamil Character Recognition System using Octal Graph. Computer Science (4) 7 : 509-516. ISSN 1549-3636
- Kaur, S.; Kaura H K and O. Mritunjay. 2010. Development of Optical Character Recognition Software Package for Mobile Phones. IJCCT (1) 2 : 3-8.
- Limanto, S. 2008. Peningkatan Minat Dan Kemampuan Anak Usia Prasekolah Untuk Belajar Membaca Dan Menulis Permulaan Menggunakan Komputer Aided Learning. Genetika (9) 01 : 113-120.
- Lukas, S.; Meiliyana dan Sepreni. 2004. Pengenalan Tulisan Tangan Pada Karakter Numerik Arab Menggunakan Jaringan Saraf Tiruan (Artificial Neural Network) Model Backpropagation. Ilmiah Ilmu Komputer (2) 2 : 131-140.
- Mudjito. 2007. Pedoman Pembelajaran Persiapan Membaca Dan Menulis Melalui Permainan Ditaman Kanan-Kanan. Departemen Pendidikan Nasional. Jakarta.
- Muis, S. 2006. Teknik Jaringan Saraf Tiruan. Graha Ilmu. Yogyakarta.
- Munir, R. 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika.
- Prasojo. A. 2011. Pengenalan Karakter Alfabet Menggunakan Jaringan Saraf Tiruan. Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro.
- Plamondon, R.; R. Srihari. 2000. On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey. IEEE (22) 01 : 63-84.

- Pradeep, J.; E. Srinivasan and Himavathi, S. 2011. Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network. IJCSIT (3) 1 : 27-38.
- Puspasari, B D. 2013. Aplikasi Pengenalan Huruf Alfabet Bagi Anak Usia Dini Menggunakan Metode Pengolahan Citra Berbasis Data Suara. ELTEK (11) 01 : 158-170.
- Raharjo, T.; Latifah, N.A. 2010. Identifikasi Kesulitan Belajar Pada Anak Pendidikan Usia Dini. ISSN : 1979-6889.
- Rianto, S at al. 2005. Step By Step Pengenalan Citra Digital. Andi Offset. Yogyakarta.
- R.C. Gonzalez, R.E. Woods. 1992. Digital Image Processing. USA: Addison-Wesley Publishing Company.
- Rokhani, E. 2012. Pengenalan pola tulisan tangan menggunakan jaringan saraf tiruan dengan algoritma backpropagation. Fakultas Matematika dan Pengetahuan Alam. UNY. Yogyakarta.
- Safaat, N. H. 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung.
- Saludin, M. 2006. Teknik Jaringan Saraf Tiruan. Graha Ilmu. Yogyakarta.
- Samsuryadi. 2009. Pengidentifikasian Pembuat Tulisan Tangan Dengan Pengenalan Pola Biometrik. Generic (4) 2 : 31-33.
- Saputro, D. W. 2006. Pengenalan Karakter Tulisan Tangan Dengan Menggunakan Jaringan Syaraf Tiruan Propagasi Balik Resilient. Fakultas Matematika Dan Ilmu Pengetahuan Alam Institut Pertanian. Bogor. 1-19.
- Setiawan, K. 2003. Paradigma System Cerdas (Jaringan Saraf Tiruan, Logical Fazi dan Algoritma Genetik). Bayu Media Publisng. Malang.
- Singh, S. 2013, Optical Character Recognition Techniques : A Survey. Emerging Trends in Computing and Information Sciences (4) 6 : 15-23.
- Stephens, K. 2004. Creative Ways to Lead Kids to Reading and Writing. Parenting Exchange. Diambil dari (Online) (<http://www.oh-pin.org/articles/pex-06-creative-ways-to-lead-kid.pdf>) diakses tanggal 8 Agustus 2013.
- Sutanto,A D. 2013. Perancangan aplikasi edukasi “smart brain kids” berbasis Android sebagai media pembelajaran untuk anak usia dini, STMIK Amikom Yogyakarta.
- Sutojo, T.; E. Mulyanto dan Suhartono. 2011. Kecerdasan Buatan. 1. Andi offset. Semarang.
- Stephanus, H. 2011. Mudah Membuat Aplikasi Android. Andi. Yogyakarta.
- Usman, A. 2005. Pengolahan citra digital dan teknik pemrograman. Graha Ilmu. Yogyakarta <http://www.eclipse.org> (5 Desember 2013).
- <http://dwiriyantikasyabaniyah.blogspot.com/p/definisi.html> (3 November 2013).
- <http://belajarpsikologi.com/metode-pengumpulan-data/Haryanto> (25 February 2013).
- <http://blog.ub.ac.id/fakhry/> (29 September 2013).