

**PENGENALAN ANGKA JAWA MENGGUNAKAN EKSTRAKSI CIRI
*CENTROID FEATURE DAN MULTILAYER PERCEPTRON (MLP)***

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer

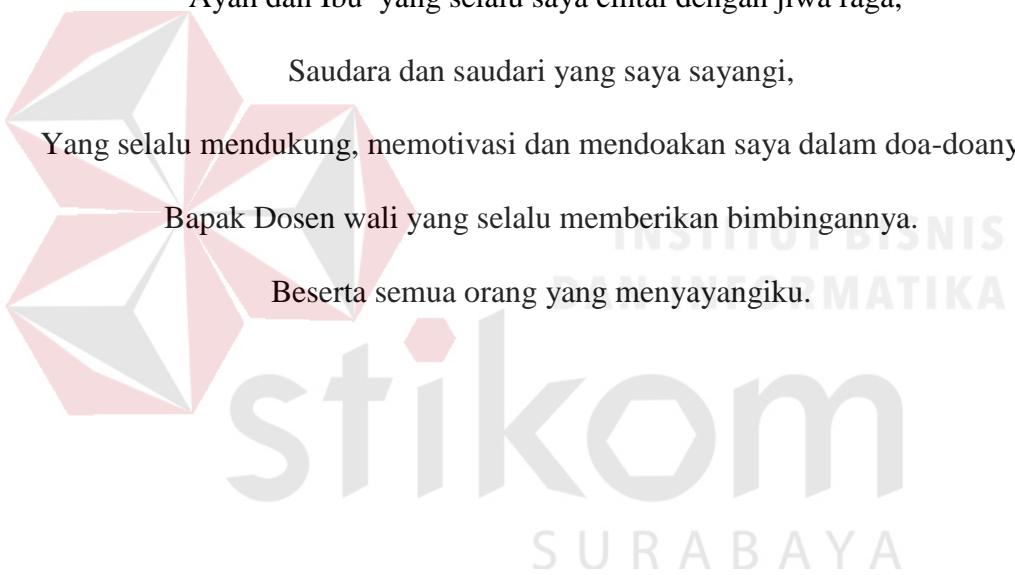


INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2015

Bismillahirrahmanirrahim, dengan mengucap basmalah dan syukur
Alhamdulillah, dengan bangga saya ingin mempersembahkan skripsi ini untuk:

Ayah dan Ibu yang selalu saya cintai dengan jiwa raga,
Saudara dan saudari yang saya sayangi,
Yang selalu mendukung, memotivasi dan mendoakan saya dalam doa-doanya.
Bapak Dosen wali yang selalu memberikan bimbingannya.
Beserta semua orang yang menyayangiku.



Tugas Akhir

**PENGENALAN ANGKA JAWA MENGGUNAKAN EKSTRAKSI CIRI
*CENTROID FEATURE DAN MULTILAYER PERCEPTRONE (MLP)***

dipersiapkan dan disusun oleh

Akhmad Yani Bakhtiyar

NIM : 09.41020.0085

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji
pada : 30 Maret 2015



Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Dr. Jusak
Dekan Fakultas Teknologi Dan Informatika

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

PERNYATAAN

Dengan ini menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam tugas akhir ini adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka saya. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya tugas akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.



ABSTRAK

Indonesia memiliki tulisan Aksara Jawa yang merupakan tulisan unik atau khas sejak zaman peradaban kerajaan – kerajaan kuno yang berasal dari pulau Jawa. Aksara Jawa tersebut merupakan salah satu budaya dari peradaban Indonesia yang diwariskan oleh nenek moyang sebagai jati diri sebuah bangsa. Akan tetapi pada saat ini banyak generasi penerus bangsa Indonesia khususnya yang berasal dari Jawa tidak dapat membaca tulisan Aksara Jawa.

Telah banyak aplikasi yang telah dibuat untuk mengenali sebuah gambar atau tulisan tangan ke dalam bentuk teks sehingga memudahkan manusia untuk mengenali suatu gambar. Diantaranya adalah menggunakan metode *Centroid Feature* dan *Multilayer Perceptron (MLP)*. Metode *Centroid Feature* adalah metode yang digunakan untuk mengekstraksi pola titik berat gambar dari suatu citra. Sedangkan *MLP* adalah model dari kecerdasan buatan atau *artificial intelligent* yang digunakan untuk mengenali sesuatu dengan prinsip yang mirip seperti cara manusia berpikir.

Dengan menggunakan ekstraksi *centroid feature* yang kemudian diintegrasikan terhadap *MLP* dalam mengenali sebuah tulisan Angka Jawa yang merupakan sebagai bagian dari Aksara Jawa dihasilkan sebuah pengenalan terhadap sampel Angka Jawa yang pernah dilatihkan terhadap *MLP* sebesar 100% sedangkan untuk pengenalan terhadap sampel Angka Jawa yang belum pernah dilatihkan menunjukkan tingkat keberhasilan sebesar 81%.

Keyword : Aksara Jawa, Angka Jawa, *Centroid Feature*, *Multilayer Perceptron*.

KATA PENGANTAR

Pertama - tama penulis panjatkan puji syukur kehadirat Allah SWT, karena berkat limpahan Rahmat dan Karunia-Nya penulis dapat menyelesaikan Tugas Akhir yang merupakan persyaratan dalam menyelesaikan Program Studi Strata Satu Jurusan Sistem Komputer di Institut Bisnis dan Informatika Stikom Surabaya Surabaya. Tugas Akhir ini berjudul “Pengenalan Angka Jawa Menggunakan Ekstraksi Ciri *Centroid Feature* dan *Multilayer Perceptron (MLP)*“.

Pada kesempatan ini, penulis juga mengucapkan terima kasih kepada :

1. Bpk. Prof. Dr. Budi Jatmiko, M.Pd selaku ketua STIKOM Surabaya yang peduli terhadap program jurusan Sistem Komputer.
2. Bpk. Harianto, S.Kom., M.Eng. selaku Dosen Pembimbing I, dan Bpk. Madha Christian Wibowo, S.Kom. selaku Dosen Pembimbing II, yang telah membimbing penulis dalam menyelesaikan Tugas Akhir ini.
3. Ibu Yosefine Triwidayastuti, M.T. selaku Dosen Penguji I, dan Bpk. Heri Pratikno, M.T. selaku Dosen Penguji II, yang telah memberikan masukan dalam penyusunan buku Tugas Akhir.
4. Segenap Dosen Pengajar program studi S-1 Sistem Komputer.
5. Sahabat – sahabatku yang telah senantiasa saling menopang penulis dalam keadaan suka maupun duka.
6. Teman - teman penulis di Jurusan Sistem Komputer yang telah membantu dalam perkuliahan selama di STIKOM Surabaya.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki lagi. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat

membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepannya. Penulis juga memohon maaf yang sebesar-besarnya jika terdapat kata-kata yang salah serta menyinggung perasaan pembaca. Akhir kata penulis ucapan banyak-banyak terima kasih yang sebesar-besarnya kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, 30 Maret 2015



DAFTAR ISI

ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Kontribusi.....	4
1.6 Sistematika Laporan	5
BAB II LANDASAN TEORI	6
2.1 Aksara Jawa	6
2.2 <i>Optical Character Recognition (OCR)</i>	7
2.3 Ekstraksi Fitur	8
2.4 <i>Region Of Interest (ROI)</i>	8
2.5 <i>Centroid Feature</i>	8
2.6 <i>Handwritten ‘Bangla’ Alphabet Recognition Using An MLP Based Classifier</i>	10
2.7 <i>An MLP Based Approach For Recognition of Handwritten ‘Bangla’</i>	12
2.8 <i>Handwritten Arabic Numeral Recognition Using Multi Layer Perceptrone ..</i>	12

2.9 Pengenalan Pola Tulisan Tangan Jawa ‘‘Ha Na Ca Ra Ka’’ Menggunakan <i>Multi Layer Perceptron</i>	13
2.10 <i>Multi Layer Perceptron</i>	14
BAB III METODE PENELITIAN.....	16
3.1 Rancangan Penelitian	16
3.2 Alat dan Bahan Penelitian.....	17
3.2.1 Alat Penelitian.....	17
3.2.2 Bahan Penelitian	17
3.3 Tahapan Penelitian	18
3.3.1 Perancangan Perangkat Lunak	20
3.4 Teknik Pengumpulan & Analisis Data.....	33
3.4.1 Pengunmpulan Data	33
3.4.2 Pelatihan MLP.....	33
3.4.3 Analisis Data	34
BAB IV HASIL DAN PENGUJIAN	36
4.1 Pengujian Perangkat Lunak.....	36
4.1.1 Pengujian <i>Picturebox</i> Sebagai Media Menggambar	36
4.1.2 Pengujian <i>Picturebox</i> Sebagai Media Pemuat <i>File Gambar</i>	37
4.1.3 Pengujian <i>Region Of Interest</i> (ROI).....	39
4.1.4 Pengujian <i>Resize</i>	41
4.1.5 Pengujian Aplikasi <i>Centroid Feature</i>	44
4.1.6 Pengujian Aplikasi Pelatihan Menggunakan MLP	49
4.1.7 Pengujian Pengenalan Angka Jawa.....	51
4.1.8 Pengujian Metode <i>Centroid Feature Extraction</i>	53

4.2 Pengujian Pengenalan Sampel <i>Training</i>	57
4.3 Pengujian Pengenalan Sampel <i>Testing</i>	59
BAB V PENUTUP.....	62
5.1 Kesimpulan.....	62
5.2 Saran	63
DAFTAR PUSTAKA	64
Lampiran1. Keseluruhan <i>Source Code</i>	67
BIODATA PENULIS	98



DAFTAR GAMBAR

Gambar 2.1. Aksara Jawa <i>Nglegena</i>	8
Gambar 2.2. Angka Jawa	8
Gambar 2.3. Tanda “ <i>Pada Pangkat</i> ” untuk Mengapit Penulisan Angka Jawa	9
Gambar 2.4. (a-b) ilustrasi untuk <i>Centroid Feature</i>	11
Gambar 2.5. Huruf Alfabetik Bangla (a) Vokal dan (b) Konsonan	13
Gambar 2.6. Bentuk Tulisan Angka ‘Bangla’	14
Gambar 2.7. Bentuk Tulisan Angka Arab.....	15
Gambar 2.8. <i>Flowchart</i> Pembuatan Miniatur	17
Gambar 3.1. Blok Diagram Pengenalan Pola menggunakan <i>Centroid Feature</i> dan MLP	19
Gambar 3.2. Diagram Alir Tugas Akhir	21
Gambar 3.3. Kanvas	24
Gambar 3.4. Tampilan Kanvas dan Properti <i>PictureBox</i>	25
Gambar 3.5. Pengaturan Kanvas (PictureBox)	25
Gambar 3.6. Citra yang dimuat atau digambar di dalam canvas	27
Gambar 3.7. Mencari ukuran gambar yang akan di- <i>cropping</i>	28
Gambar 3.8. Citra setelah melalui proses ROI.....	28
Gambar 3.9. Citra.....	29
Gambar 3.10. <i>Flowchart</i> Preprocessing (ROI dan <i>Resize</i>)	30
Gambar 3.11. Pembagian citra menjadi 8 oktan	31
Gambar 3.12. Contoh mencari titik berat dari kotak kuning	32
Gambar 3.13. Koordinat titik tengah pada masing – masing oktan.....	32

Gambar 3.14. Nilai titik tengah yang telah didapatkan pada setiap oktan.....	33
Gambar 3.15. <i>Flowchart</i> ekstraksi fitur <i>Centroid</i>	34
Gambar 3.16. <i>Output</i> Dari <i>Centroid Feature</i> Menjadi <i>Inputan MLP</i>	35
Gambar 3.17. <i>Flowchart</i> <i>MLP</i>	35
Gambar 4.1. Hasil gambar yang dibentuk.....	39
Gambar 4.2. <i>Load</i> gambar ukuran 256 x 256 piksel.....	40
Gambar 4.3. <i>Load</i> gambar ukuran 64 x 64 piksel.....	41
Gambar 4.4. Hasil proses ROI	42
Gambar 4.5. Hasil penyimpanan sampel gambar.....	43
Gambar 4.6. <i>Rezize</i> pada gambar yang berukuran lebih kecil dari 64 x 64 piksel.....	44
Gambar 4.7. <i>Rezize</i> pada gambar yang berukuran lebih besar dari 64 x 64 piksel.....	45
Gambar 4.8. Perbandingan <i>rezize</i> antara hasil sampel <i>rezize</i> dengan <i>software Pixilion</i>	45
Gambar 4.9. <i>Centroid</i> yang dihasilkan	47
Gambar 4.10. Penyimpanan gambar sampel <i>Centroid</i>	48
Gambar 4.11. Penyimpanan data hasil ekstraksi <i>Centroid</i>	49
Gambar4.12.Penyimpanan data hasil ekstraksi <i>Centroid</i> yang telah dinormalisasi.....	50
Gambar 4.13. Proses <i>training</i> pada <i>MLP</i>	52
Gambar 4.14. Hasil penyimpanan data <i>training</i>	53
Gambar 4.15. Hasil pengenalan angka jawa.....	54
Gambar 4.16. hasil pemetaan <i>pixel</i> yang dihasilkan oleh <i>Matlab 7.0</i>	57



DAFTAR TABEL

Tabel 2.1. Performa Keberhasilan MLP dengan Variasi Jumlah Neuron.....	13
Tabel 4.1. Perbandingan Fitur <i>Centroid</i> dengan penghitungan manual (<i>MS EXCEL</i>).....	57
Tabel 4.2. Presentase Keberhasilan Pengenalan Pada Sampel Pelatihan.....	60
Tabel 4.3. Presentase Keberhasilan Pengenalan Pada Sampel <i>Test</i>	61
Tabel 4.3. Analisa Kesalahan Identifikasi Pada Tiap Karakter	62



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Aksara jawa merupakan aset penting dari bagian sejarah bangsa Indonesia. Walaupun tulisan ini berasal dari pulau jawa namun pengaruhnya pada zaman dahulu berkembang sejalan dengan kekuasaan dari pemerintahan dari jawa yang menyebar ke wilayah nusantara. Pentingnya pelestarian tulisan aksara ini merupakan tanggung jawab suatu bangsa terhadap generasi penerus selanjutnya. Pada negara - negara lain misalnya cina dan jepang, selain digunakan sebagai tulisan sehari – hari, mereka juga mempertahankan tulisan asal mereka sendiri ditengah – tengah era globalisasi untuk diaplikasikan pada kehidupan sehari – hari, baik pada perangkat digital contohnya *handphone* dan *computer*. Dengan mengaplikasikan kepada perangkat digital tersebut mereka secara tidak langsung memudahkan serta memberikan kenyamanan generasi bangsanya untuk mengenali tulisan yang merupakan simbol dari peradaban dan kebudayaan

Terdapat beragam jenis aplikasi atau perangkat lunak yang telah dikembangkan atau improvisasi dalam mengenali penulisan tangan. Untuk melakukan pengenalan tulisan dari tangan tersebut dapat dilakukan dengan cara melakukan identifikasi suatu bentuk pola dari gambar kemudian mengubahnya ke dalam bentuk teks. Terdapat banyak metode yang dapat dilakukan. Diantaranya yaitu metode *Centroid Feature* yang digunakan untuk mengekstraksi suatu pola titik berat dari gambar dan selain itu terdapat pula metode dari sebuah model

artificial intelligent atau dapat disebut dengan kecerdasan buatan yang salah satunya merupakan metode *Multi Layer Perceptron* atau MLP.

Basu,dkk pada 2005 melakukan penelitian tentang pengenalan pola pada huruf Bangla *Handwritten ‘Bangla’ Alphabet Recognition Using An MLP Based Classifier*. Dalam penelitian tersebut masukkan dari MLP adalah keluaran dari proses ekstraksi fitur dari metode *shadow feature*, *Centroid Feature* dan *longrest-run*.

Tahun 2013 Wibowo dan Wirakusuma melakukan penelitian dalam pengenalan pola pada karakter Aksara Jawa “Ha Na Ca Ra Ka” hanya menggunakan MLP. Pada hasil penelitian tersebut beliau memberikan saran berupa “Akan lebih baik bila ada informasi yang ditambahkan pada pola *input*”.

Berdasarkan penelitian-penelitian yang sudah ada, dalam Tugas Akhir ini dilakukan proses pengambilan ciri (*Centroid Feature*) sebelum dikomparasikan dengan MLP seperti yang disarankan oleh Wibowo dan Wirakusuma. *Centroid Feature* merupakan salah satu metode yang digunakan dalam mengenali suatu pola gambar dengan mencari titik berat dari gambar itu sendiri. Obyek yang dikenali adalah citra Angka Jawa, sedangkan pola yang dikenali adalah 10 jenis Angka Jawa yang merupakan satu set dari tulisan Angka Jawa. Data yang digunakan adalah 100 set tulisan tangan Angka Jawa untuk pelatihan, dan 50 set tulisan tangan Angka Jawa yang berbeda untuk pengujian.

Data tersebut diambil melalui serangkaian proses ROI (*Region of Interest*) dan ekstraksi ciri *Centroid*. Kemudian data dari hasil tersebut dimasukkan ke dalam proses pelatihan dengan menggunakan MLP (*MultiLayer*

Perceptron) untuk memperoleh bobot yang optimal. Sedangkan hasil pengenalan pola akan diuji menggunakan sampel pelatihan dan sampel uji.

Hasil pengujian pada masing – masing sampel akan dikomparasikan untuk melihat signifikansi penggunaan ekstraksi ciri pada MLP untuk pengenalan pola Angka Jawa.

1.2 Perumusan Masalah

Adapun permasalahan yang akan dihadapi dalam pengerjaan tugas akhir ini diantaranya adalah :

1. Bagaimana cara mengekstraksi fitur angka jawa menggunakan ekstraksi *Centroid Feature*.
2. Bagaimana metode ekstraksi *Centroid Feature* dapat mengenali tulisan Angka Jawa dengan menggunakan MLP.
3. Bagaimana hasil dari pengenalan angka jawa yang telah diekstraksi *Centroid Feature* dengan menggunakan MLP pada sampel *training* terhadap sampel *testing*.

1.3 Batasan Masalah

Adapun batasan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut :

1. Objek yang dideteksi adalah pola tulisan tangan Angka Jawa tanpa *noise*.
2. Pengerjaan tugas akhir hanya pada bagian ekstraksi ciri *Centroid Feature* saja.
3. Media interaktif untuk pengumpulan data adalah kanvas virtual atau tablet.

1.4 Tujuan

Tujuan dari pembuatan aplikasi yang akan dilakukan pada Tugas Akhir ini adalah sebagai berikut:

1. Mengoptimalkan pengenalan karakter dengan menggunakan *Centroid Feature Extraction*.
2. Mengintegrasikan *Centroid Feature* dengan MLP.
3. Menguji hasil pengenalan pola MLP yang menggunakan ekstraksi *Centroid Feature*.

1.5 Kontribusi

Pada 2013 Wibowo dan Wirakusuma meneliti tentang pengenalan pola tulisan tangan untuk Aksara Jawa *nglegena* dengan menggunakan metode *multi layer perceptron*. Dalam penelitian tersebut, data yang digunakan pada jaringan MLP adalah data citra mentah.

Penelitian ini berfokus pada Angka Jawa dan pengolahan ekstraksi ciri *Centroid Feature* untuk kemudian digunakan sebagai data masukan pada jaringan MLP. Pengujian hasil ekstraksi ciri *Centroid Feature* ini dilakukan dengan pelatihan data ekstraksi secara signifikan terlebih dahulu terhadap MLP. Dari penelitian ini diharapkan tingkat akurasi pengenalan pola tulisan Angka Jawa lebih meningkat serta penelitian tentang OCR (*Optical Character Recognition*) untuk Aksara Jawa dapat lebih berkembang. Penelitian ini juga diharapkan menjadi dasar untuk melakukan pengenalan Aksara Jawa secara menyeluruuh hingga Aksara Jawa kuno.

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penulisan laporan tugas akhir, kontribusi dan sistematika penulisan tugas akhir.

BAB II : LANDASAN TEORI

Bab ini membahas tentang berbagai teori yang mendukung tugas akhir ini. Hal tersebut meliputi *Centroid Feature*, *optical character recognition*, ekstraksi fitur, angka jawa, *Multilayer Perceptron (MLP)*, selain itu dijabarkan pula referensi-referensi terkait penelitian yang pernah dilakukan sebelumnya.

BAB III : METODE PENELITIAN

Bab ini berisi penjelasan tentang tahap-tahap yang dikerjakan dalam penyelesaian TA dalam pembuatan aplikasi *Centroid Feature extraction* sebagai ekstraksi ciri dan penggabungan data dengan MLP

BAB IV : PENGUJIAN DAN ANALISIS SISTEM

Bab ini berisi tentang pengujian ekstraksi ciri dan pengenalan pola tulisan angka jawa. Pengujian yang dilakukan meliputi ROI(*Region of Interest*), *resize*, *Centroid Feature Extraction*, pengenalan pola pada MLP dengan *Centroid Feature Extraction*.

BAB V : PENUTUP

Bab ini berisi tentang kesimpulan penelitian serta saran untuk pengembangan penelitian berikutnya.

BAB II

LANDASAN TEORI

2.1 Aksara Jawa

Carakan (Abjad atau Aksara Jawa) adalah huruf yang digunakan dalam ejaan bahasa Jawa yang pada dasarnya terdiri atas dua puluh aksara pokok (*nglegena*), yang ditunjukkan pada gambar 2.1, yang bersifat silabik (kesukukataan). (Darusuprapta, dkk., 2002). Bentuk kontemporer Aksara Jawa terbentuk sejak masa Kerajaan Mataram pada abad ke-17.

Selain dua puluh aksara pokok tersebut, Aksara Jawa juga memiliki kelompok aksara kapital (*murda*), vokal (*swara*), rekaan (*rekan*), pengubah bunyi (*sandangan*), penanda gugus konsonan, penutup konsonan(*pasangan*), *pangkon*, tanda baca, dan angka.

ଠା	ଠା	ଠା	ଠା	ଠା
ha	na	ca	ra	ka
ଠା	ଠା	ଠା	ଠା	ଠା
da	ta	sa	wa	la
ଠା	ଠା	ଠା	ଠା	ଠା
pa	dha	ja	ya	nya
ଠା	ଠା	ଠା	ଠା	ଠା
ma	ga	ba	tha	nga

Gambar 2.1 Aksara Jawa *Nglegena*

ଠା	ଠା	ଠା	୩	ଠା	ଠା	ଠା	ଠା	ଠା	୦
1	2	3	4	5	6	7	8	9	0

Gambar 2.2 Angka Jawa



Gambar 2.3 Tanda “*Pada Pangkat*” untuk Mengapit Penulisan Angka Jawa

Beberapa dari Angka Jawa, seperti pada Gambar 2.2, dalam Aksara Jawa memiliki bentuk yang mirip dengan kelompok Aksara Jawa lainnya. Beberapa di antaranya misalkan angka “1” dalam Aksara Jawa memiliki bentuk yang sama dengan huruf “Ga” pada kelompok Aksara Jawa *Nglegena*, kemudian angka “9” sama dengan huruf “Ya” dalam kelompok Aksara Jawa *Nglegena*. Oleh karena itu, penulisan Angka Jawa, menurut pedoman yang ditulis oleh Darusuprasta dkk., harus diapit oleh tanda baca yang memiliki istilah “*pada pangkat*” seperti pada Gambar 2.3.

2.2 *Optical Character Recognition (OCR)*

Optical Character Recognition (OCR) adalah proses pengenalan karakter dari *file* citra dan juga pengenalan alfanumerik yang dicetak atau penulisan tangan dari karakter, teks, angka, huruf dan simbol agar bisa dibaca oleh komputer (Mukherjee, 2010). OCR merupakan pengenalan pola dan pengolahan karakter tulisan tangan yang didasari keinginan manusia untuk meningkatkan komunikasi dengan komputer. Banyak penelitian yang telah dilakukan untuk mengembangkan sistem OCR dalam berbagai skrip. OCR meliputi *scanning* gambar kemudian diterjemahkan menjadi teks, dari dokumen kertas menjadi gambar, kemudian gambar diterjemahkan ke dalam kode karakter yang pada komputer seperti membaca kode ASCII. Pengenalan kode pos, pengolahan bentuk, verifikasi tanda tangan adalah bentuk-bentuk penerapan OCR (Mukherjee, 2010).

2.3 Ekstraksi Fitur

Ekstraksi fitur dapat didefinisikan sebagai ekstraksi yang mengutamakan informasi dari ciri yang mewakili sebuah baris data, yang mana data diminimalisasi ke dalam pola yang berbeda-beda (Trier, 1996). Untuk tujuan inilah, kumpulan dari ciri (fitur) yang diekstrak pada tiap-tiap kelas akan membantu untuk membedakan pola yang diekstrak dari pola lain.

2.4 Region Of Interest (ROI)

Region Of Interest (ROI) merupakan salah satu fitur yang tersedia dalam JPEG2000. ROI memungkinkan dilakukannya pengkodean secara berbeda pada area tertentu dari citra digital, sehingga mempunyai kualitas yang lebih baik dari area sekitarnya. Fitur ini menjadi sangat penting, bila terdapat bagian tertentu dari citra digital yang dirasakan lebih penting dari bagian yang lainnya.

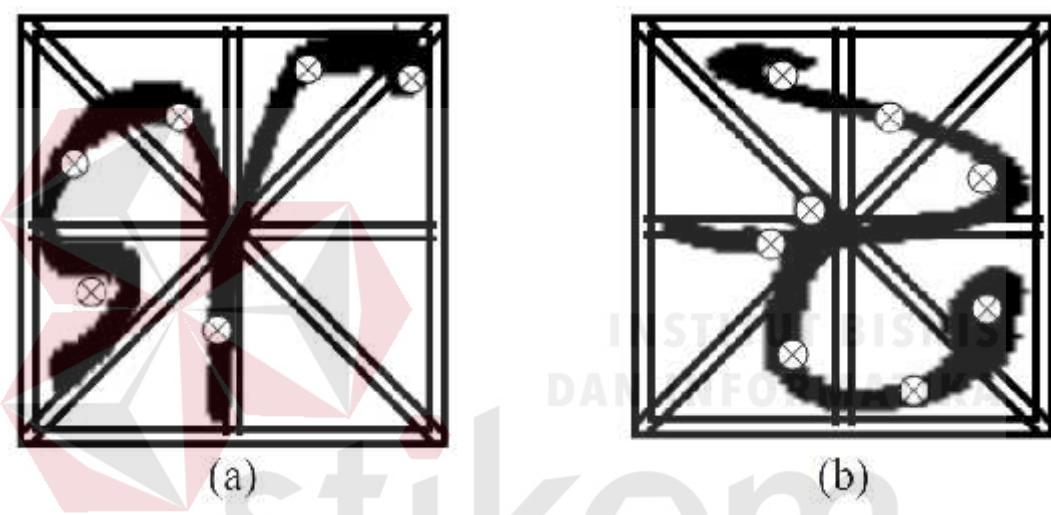
Sebuah piksel memiliki intensitas yang menentukan warna dari piksel itu sendiri, kumpulan dari piksel – piksel yang memiliki intensitas tertinggi dari keseluruhan citra merupakan bentuk gambar yang mendominasi sebuah citra. Perbandingan dari kumpulan piksel yang memiliki intensitas tinggi dengan yang berintensitas rendahlah yang akan menentukan letak dari suatu gambar (Nurtanio dkk, 2013).

2.5 Centroid Feature

Centroid adalah metode ekstraksi fitur yang digunakan untuk mengenali suatu pola tulisan, angka, atau simbol pada citra dengan cara mencari titik berat dari sebuah citra (Das dkk, 2006). Piksel – piksel dari citra yang digunakan dalam metode ini dikonversi terlebih dahulu menjadi *grayscale* (hitam atau putih). Baru

kemudian dilakukan proses pencarian titik berat dari sebuah citra (ekstraksi) yang dilakukan dengan cara *scanning* pada seluruh area citra, *Scanning* dilakukan untuk memperoleh nilai dari tiap-tiap pixel citra yang bernilai 1 (hitam).

Konsep dasar *Centroid Extraction* adalah pembagian sebuah citra menjadi 8 area atau oktan dengan mencari titik berat pada tiap - tiap oktan tersebut. Gambar 2.4(a-b) menunjukkan perbandingan dari titik berat yang terbentuk di masing – masing oktannya.



Gambar 2.4(a-b) ilustrasi untuk *Centroid Feature*.

Titik berat hasil dari ekstraksi *Centroid* pada masing – masing oktan memiliki 2 buah kordinat yaitu terhadap sumbu x dan sumbu y, sehingga dari sebuah citra didapatkan 16 fitur yang terbentuk(Nibaran das dkk, 2006). Hasil dari tiap – tiap titik berat tersebut dinormalisasi dengan cara dibagi dengan besaran jumlah pixel dari citra tersebut.

2.6 Handwritten ‘Bangla’ Alphabet Recognition using an MLP Based Classifier

Penelitian serupa pernah dilakukan oleh Basu pada tahun 2005. Dalam penelitian tersebut, metode yang digunakan adalah *multi layer perceptron*. MLP digunakan untuk melakukan pengenalan terhadap huruf-huruf alfabetik Bangla. Huruf alfabetik Bangla terdiri dari 50 jenis huruf yang terdiri dari 11 huruf vokal dan 39 huruf konsonan seperti pada Gambar 2.5.

Di dalam penelitian yang dilakukan oleh Basu, masing - masing sampel dari citra huruf Bangla dibagi menjadi 64 x 64 piksel yang kemudian dilanjutkan dengan melakukan ekstraksi fitur sebanyak 76 fitur yang terdiri dari 24 *shadow of features*, 16 *Centroid of Features*, dan 36 *longest-run of features*. Sampel huruf yang digunakan dalam penelitian ini adalah 10.000 sampel huruf yang didapatkan dari tulisan tangan 200 orang dengan jenis kelamin dan umur yang bervariasi. Dari 10.000 sampel tersebut, 8.000 sampel digunakan sebagai sampel pelatihan dan 2.000 sampel digunakan sebagai sampel uji (Basu, dkk., 2005).

ଅ	ଆ	ରୁ ରୁ
ତୁ	କୁ	ଶୁ ଅ
ରୁ	ବୁ	ଦୁ

(a)

କ୍ରମ	ଶ୍ଵର	ଶ୍ଵର
ଟ୍ରେନିଂ	ଶ୍ଵର	ଶ୍ଵର
ପ୍ରସ୍ତର	ଶ୍ଵର	ଶ୍ଵର
ଶ୍ଵର	ଶ୍ଵର	ଶ୍ଵର

(b)

Gambar 2.5 Huruf Alfabetik Bangla (a) Vokal dan (b) Konsonan

Tabel 2.1. Performa Keberhasilan MLP dengan Variasi Jumlah Neuron

Jumlah Neuron	35	40	45	50	55	60	65	70	75
Pembelajaran (%)	80,35	80,93	82,8	84,7	85,7	86,46	87,36	86,69	86,65
Uji (%)	70,35	70,7	71,6	73,15	74,2	75,05	73,65	74,7	72

Proses *training* dilakukan dengan menggunakan sebuah *hidden layer* atau *layer* tersembunyi dan dengan beberapa konfigurasi jumlah neuron yang berbeda-beda. Masing-masing proses pelatihan dilakukan sampai dengan 2.000 iterasi, laju pembelajaran sebesar 0,8, dan nilai momentum sebesar 0,7.

Hasil percobaan dari penelitian ini ditunjukkan pada Tabel 2.1, di mana hasil yang paling baik didapatkan dari jaringan dengan jumlah neuron 60. Pada percobaan tersebut, persentase keberhasilan dari sampel pembelajaran adalah 86,46% sedangkan untuk sampel uji adalah 75,05%.

2.7 An MLP Based Approach for Recognition of Handwritten ‘Bangla’ Numerals

Seperti halnya pada Sub-Bab 2.6, pada penelitian ini peneliti yang sama dan dengan metode yang sama membuat pengenalan pola tulisan tangan untuk angka ‘Bangla’. Angka Bangla terdiri dari 10 jenis angka yang merepresentasikan 0 – 9 seperti pada Gambar 2.6.

০	১	২	৩	৪	৮	৫	৬	৭	৮	৯
0	1	2	3	4	5	6	7	8	9	

Gambar 2.6 Bentuk Tulisan Angka ‘Bangla’

Pada penelitian kali ini digunakan data sampel tulisan tangan untuk angka-angka ‘Bangla’ dari 600 orang, sehingga didapatkan 6.000 sampel pola. 4.000 sampel digunakan sebagai sampel pembelajaran dan 2.000 sampel digunakan sebagai sampel uji.

Hasil dari penelitian ini menunjukkan bahwa tingkat keberhasilan paling tinggi didapatkan dari percobaan dengan jumlah neuron 65, yaitu 96,67% (Basu, dkk., 2005).

2.8 Handwritten Arabic Numeral Recognition using a Multi layer Perceptron

Di tahun 2006, oleh peneliti yang berbeda ada penelitian dengan tema yang sama, yaitu tentang *optical character recognition* (OCR). Sama halnya seperti dua penelitian sebelumnya pada Sub Bab 2.6 dan 2.7, pada penelitian ini

metode yang digunakan juga masih sama. Hanya saja perbedaannya adalah jenis karakter yang dikenali adalah angka Arab.

Angka Arab terdiri dari 10 jenis karakter yang merepresentasikan 0 – 9 seperti pada Gambar 2.7. Dalam penelitian ini sebanyak 300 set tulisan angka Arab dari orang-orang yang berbeda digunakan sebagai sampel. Yang digunakan sebagai sampel uji sebanyak 2.000 sampel, sedangkan sisanya digunakan sebagai sampel uji.

Hasil paling baik didapatkan pada jaringan dengan jumlah neuron pada *layer* tersembunyi sebanyak 54 neuron. Persentase keberhasilan dari bentuk jaringan tersebut mencapai 95% (Das, dkk., 2006).

٠	١	٢	٣	٤	٥	٦	٧	٨	٩
---	---	---	---	---	---	---	---	---	---

Gambar 2.7 Bentuk Tulisan Angka Arab

2.9 Pengenalan Pola Tulisan Tangan Aksara Jawa “Ha Na Ca Ra Ka”

Menggunakan Multi Layer Perceptron

Pada 2013, Wibowo dan Wirakusuma telah melakukan kajian terhadap topik ini. Namun tulisan tersebut menggunakan 5 huruf awal, yaitu “Ha”, “Na”, “Ca”, “Ra”, dan “Ka”. Jumlah sampel yang digunakan untuk pembelajaran adalah 15 set sampel dan pengujian dilakukan terhadap 5 set sampel.

Data yang digunakan untuk dimasukkan dalam jaringan MLP adalah data mentah dari citra yang dikecilkan saja. Dalam penelitian ini tidak digunakan proses ekstraksi ciri.

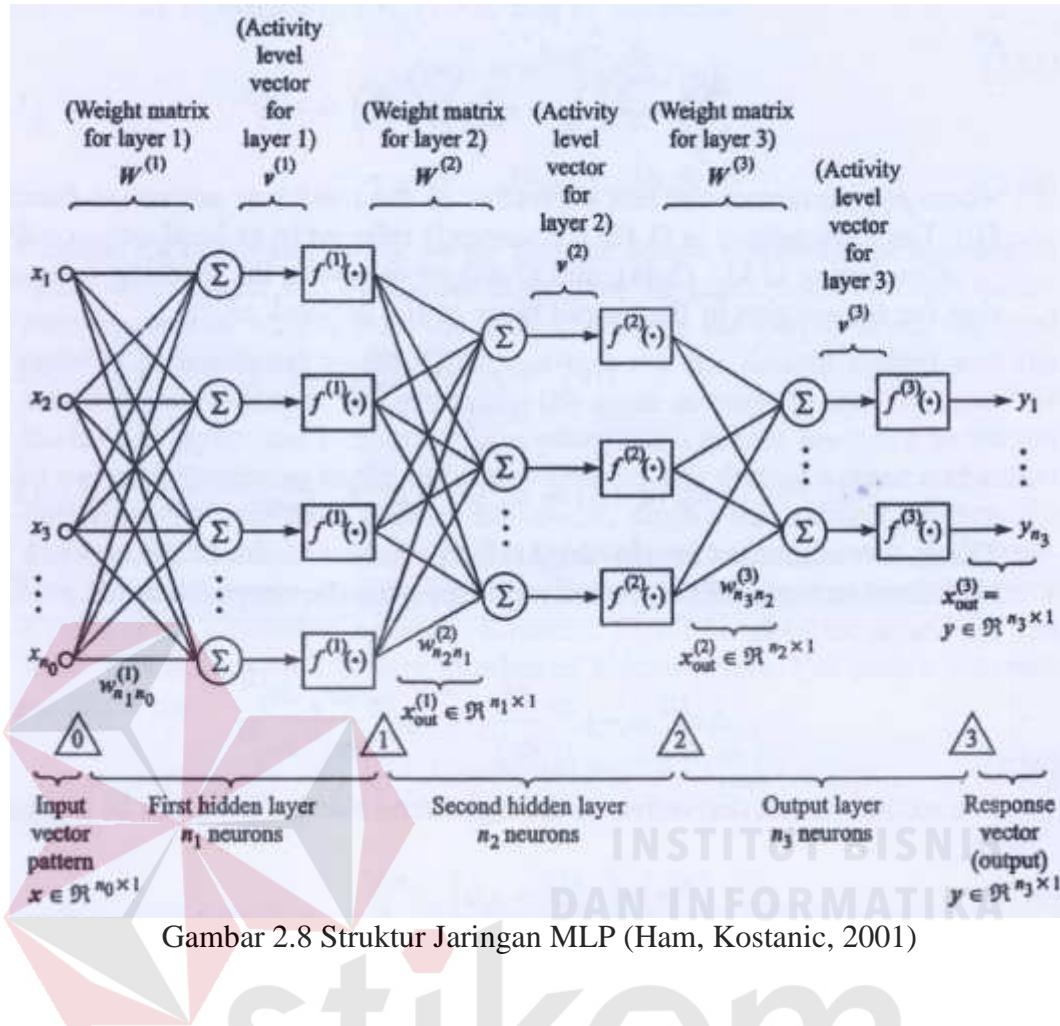
Hasil dari penelitian tersebut menunjukkan proses pembelajaran yang konvergen dengan nilai SSE (Sum Square Error) sebesar 0.00096118032275095 setelah iterasi ke-738.522. Konfigurasi yang digunakan pada penelitian tersebut adalah:

- Jumlah *hidden layer* : 2
- Jumlah neuron input : 300
- Jumlah neuron layer 1 : 80
- Jumlah neuron layer 2 : 80
- Jumlah neuron output : 5
- Laju pembelajaran (μ) : 0.1
- Error minimum : $10E^{-4}$
- Iterasi maksimum : $10E^{+5}$

Pengujian yang dilakukan mendapatkan hasil bahwa seluruh sampel pembelajaran yang dicoba untuk dikenali ulang berhasil dikenali dengan benar. Sedangkan untuk sampel uji, 14 huruf dari 25 huruf uji dapat dikenali dengan benar (Wibowo dan Wirakusuma, 2013).

2.10 Multi Layer Perceptron

Multi Layer Perceptron (MLP) adalah perkembangan dari algoritma *perceptron* yang memiliki *layer* lebih dari sekedar *input layer* dan *output layer*. Pada MLP, terdapat *layer - layer* tambahan yang disebut dengan *hidden layer*. Dengan adanya *hidden layer* ini MLP dimungkinkan untuk dapat mengenali pola yang kompleks yang tidak dapat dikenali oleh struktur jaringan *perceptron* konvensional. Struktur jaringan MLP dapat dilihat pada gambar 2.8.



Gambar 2.8 Struktur Jaringan MLP (Ham, Kostanic, 2001)

Neuron - neuron pada *hidden layer* memiliki proses perhitungan yang sama dengan neuron - neuron pada *output layer*. Hanya saja nilai hasil perhitungan dari setiap neuron pada *hidden layer* ini nantinya akan digunakan sebagai masukan untuk *layer - layer* berikutnya. Proses perhitungan dari masing - masing neuron pada tiap *layer* melibatkan luaran dari neuron - neuron di *layer* sebelumnya dan bobot dari koneksi antara neuron tersebut dengan neuron - neuron di *layer* sebelumnya. Di mana w adalah bobot antara luaran x dari neuron di *layer* sebelumnya dengan neuron yang akan dihitung luarannya. Namun nilai x tersebut bukanlah luaran akhir dari neuron. Untuk itu nilai x tersebut harus ditambah dengan *bias* atau dikurangi *threshold* untuk menjadi v sebelum dimasukkan ke

dalam fungsi aktivasi. Jika fungsi aktifasi yang dipilih menggunakan fungsi *sigmoid unipolar* maka secara umum y didapat dengan persamaan :

Keterangan :

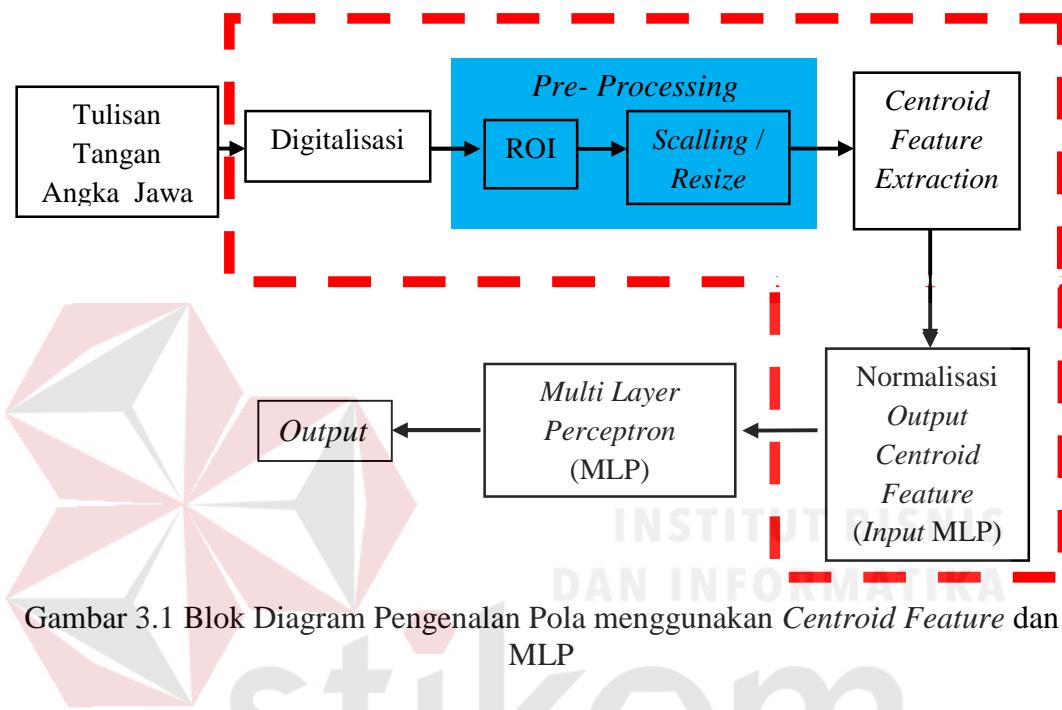
- $Y_q = \text{output}$
 - $f_{bs} = \text{fungsi sigmoid unipolar}$
 - $vq = \text{sigma neuron output ke}$
 - $e = \text{eksponen}$
 - $\beta = \text{konstanta 1}$

Luaran y dari neuron di *output layer* adalah luaran akhir dari MLP. Proses yang terjadi mulai dari *input layer* hingga *output layer* ini disebut dengan *feed-forward*. Nilai luaran yang dihasilkan akan sangat bergantung pada konfigurasi bobot yang terhubung antar neuron. Untuk mendapatkan nilai bobot yang sesuai maka diperlukan proses belajar.

BAB III

METODE PENELITIAN

3.1 Rancangan Penelitian



Gambar 3.1 adalah blok diagram sistem keseluruhan perangkat lunak yang dibuat, sedangkan pada blok di dalam garis putus-putus merupakan fokus utama dari pengerjaan Tugas Akhir ini. Citra yang akan dijadikan sampel diperoleh melalui media *input* interaktif yang memungkinkan untuk diberi tulisan tangan (Picturebox pada VB6 dan Tablet). Data citra itu sendiri memiliki resolusi yang bervariasi.

Tahap awal adalah dengan melakukan Proses Digitalisasi citra yang digambar ke dalam bentuk biner, selanjutnya adalah *preprocessing*, kemudian melakukan ekstraksi *Centroid Feature* sekaligus normalisasi nilai fitur dan

terakhir adalah pengujian melalui MLP. Ekstraksi ciri dilakukan dengan mencari titik tengah atau bobot tengah suatu gambar di dalam sebuah citra. Namun sebelum dilakukan proses ekstraksi, citra terlebih dahulu melalui tahap *preprocessing* yang mana didalamnya terdapat proses dari ROI (*region of interest*) pada citra dengan sisi bingkai yang sama (persegi) dan proses *resize* dengan ukuran 64 piksel x 64 piksel.

3.2 Alat dan Bahan Penelitian

Alat dan bahan yang akan digunakan dalam penelitian ini hanya terdiri dari beberapa alat pendukung untuk pembuatan program. Alat dan bahan yang akan digunakan sebagai berikut.

3.2.1 Alat Penelitian

Alat yang digunakan untuk menyelesaikan penelitian ini adalah *Personal Computer (PC) / Laptop*. Alat tersebut bertujuan untuk membuat aplikasi dengan menggunakan program *visual basic 6.0*.

3.2.2 Bahan Penelitian

Bahan yang akan diteliti oleh penulis dalam penelitian kali ini meliputi beberapa aspek sebagai berikut.

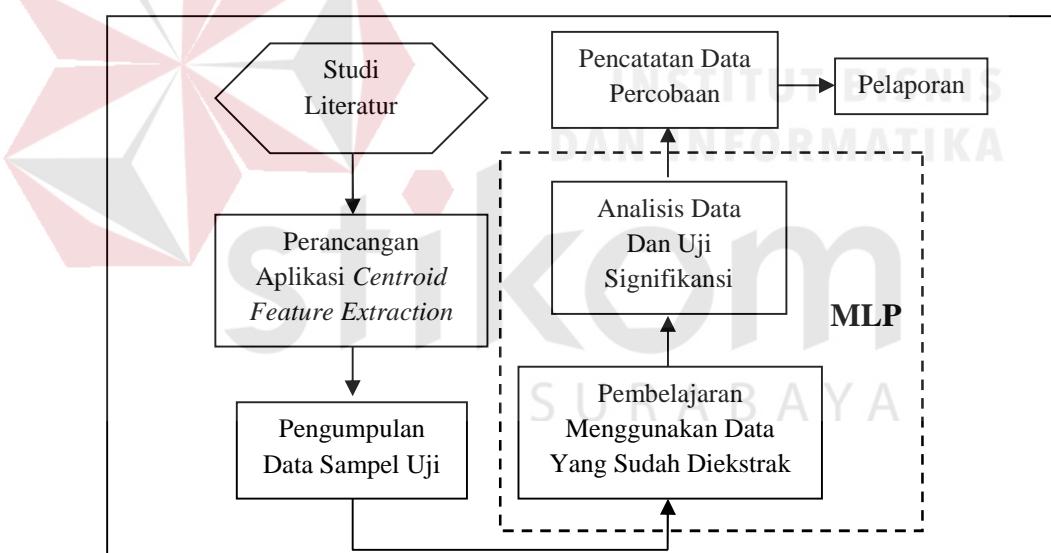
1. Pengambilan *Centroid Feature* menggunakan *software visual basic 6.0*.
2. Data sampel sebanyak 150 *set* citra atau 1500 citra (1 *set* = 10 citra) yang diperoleh dari 150 orang dengan masing – masing orang menuliskan 1 *set* yang terdiri dari angka 0 hingga 9. Data tersebut nantinya akan

digunakan sebanyak 100 *set* untuk keperluan training pada aplikasi dan 50 *set* untuk keperluan *testing*.

3. Microsoft excel 2007 yang digunakan untuk penyimpanan nilai hasil ekstraksi maupun hasil pengujian serta analisa keberhasilan dalam mengenali pola.
4. Matlab 7.0 yang digunakan untuk memetakan citra secara manual demi keperluan saat pengujian secara manual.

3.3 Tahapan Penelitian

Langkah – Langkah yang digunakan dalam menyelesaikan tugas akhir ini adalah seperti pada Gambar 3.2:



Gambar 3.2 Diagram Alir Tugas Akhir

- a. Mengumpulkan literatur yang berhubungan dengan tugas akhir yang akan dikerjakan baik dari buku, jurnal, maupun dari internet. Serta

memahami dengan rinci pengenalan pola tulisan menggunakan menggunakan ekstraksi *Centroid Feature*.

- b. Melakukan perancangan perangkat lunak yang akan dibuat. Perancangan perangkat lunak dibagi menjadi beberapa tahap, yaitu :
 1. Merancang media untuk memasukkan pola huruf angka jawa untuk keperluan pengujian sampel dan merancang ruang penyimpanan citra yang didapatkan tersebut.
 2. Merancang proses ROI (*Region Of Interest*) untuk mengambil bagian gambar yang diinginkan pada citra dengan meniadakan bagian gambar yang tidak diperlukan.
 3. Merancang proses *Resize* citra sebesar dengan ukuran sebesar 64 piksel x 64 piksel.
 4. Merancang proses ekstraksi *Centroid Feature* dan Merancang ruang penyimpanan pola huruf yang sudah dimasukkan agar tidak perlu membuat pola baru untuk memudahkan saat pengujian.
 5. Merancang sistem proses evaluasi dan pencatatan.
- c. Mengambil sampel data pembelajaran pola huruf pada media yang dibuat pada perangkat lunak. Masing – masing pola huruf yang diambil harus bervariasi. Ada 10 pola huruf yang harus dikenali dan masing – masing pola memiliki 100 variasi sampel data pembelajaran dan 50 variasi sampel data untuk evaluasi.

- d. Menjalankan proses pembelajaran sesuai dengan data pembelajaran yang diberikan point c, kemudian diekstrak menggunakan ekstraksi *Centroid Feature*.
- e. Proses pembelajaran menggunakan MLP yang sebelumnya citra telah diekstrak fitur - fiturnya menggunakan ekstraksi *Centroid Feature*.
- f. Pencatatan data dan menjalankan sistem keseluruhan yang meliputi parameter - parameter yang sesuai dengan perumusan masalah yang dikemukakan.
- g. Melakukan analisa, penarikan kesimpulan dan pengujian signifikansi hasil yang didapat dari sampel yang diolah terlebih dahulu dengan mengekstrak cirinya dibandingkan dengan pembelajaran menggunakan data mentah.
- h. Melaporkan dan mempublikasikan tugas akhir dalam bentuk jurnal.

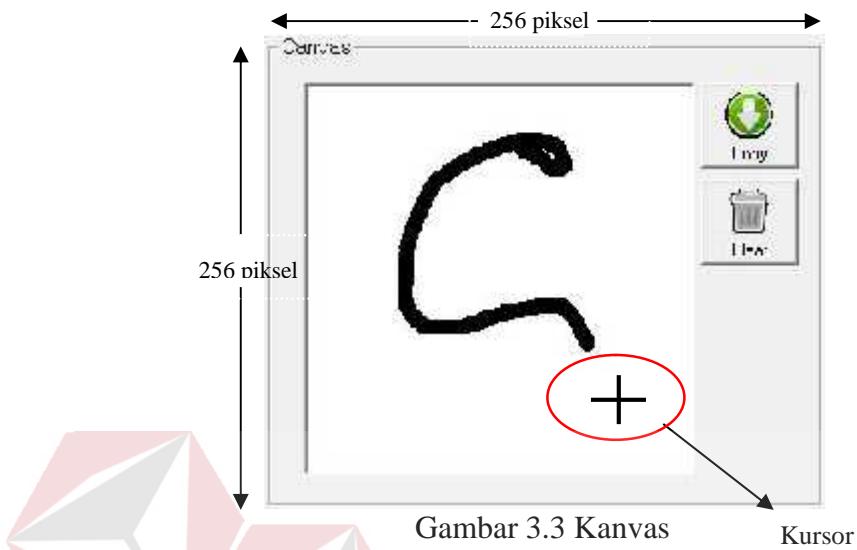
3.3.1 Perancangan Perangkat Lunak

Melakukan perancangan perangkat lunak yang akan dibuat. Perancangan perangkat lunak dibagi menjadi beberapa tahap, yaitu :

1. Desain Kanvas

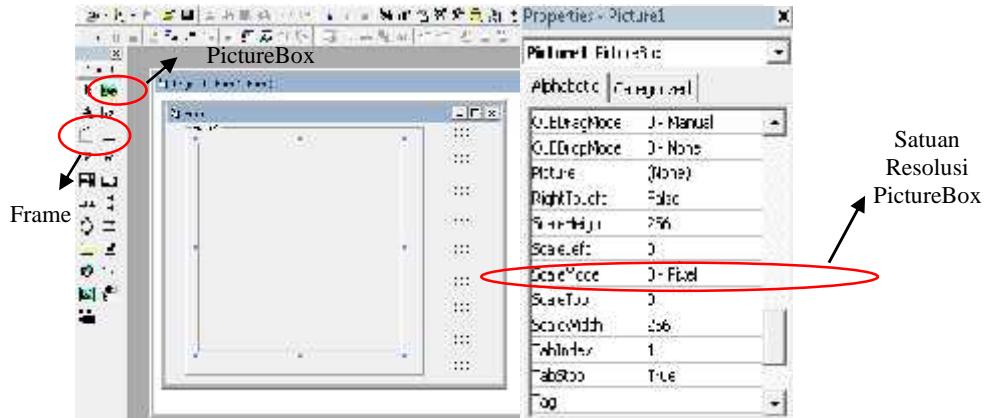
Pengambilan pola penulisan angka jawa dari *participant* dilakukan pada media interaktif berupa kanvas yang digunakan untuk menangkap pola tulisan tangan dari *participant*. Kanvas tersebut akan dibuat dengan *software visual basic* menggunakan *tool picturebox*. Resolusi yang dimiliki kanvas sebesar 256 piksel x 256 piksel.

Dalam penggunaan kanvas tersebut *Participant* cukup menggerakkan (*Drag*) kursor sesuai pola angka jawa yang akan digambar pada kanvas menggunakan *Mouse/Mouse Pen* seperti pada Gambar 3.3 berikut.



Pola penulisan angka jawa dari *participant* disimpan dengan format *bitmap*. Langkah-langkah pembuatan kanvas adalah sebagai berikut:

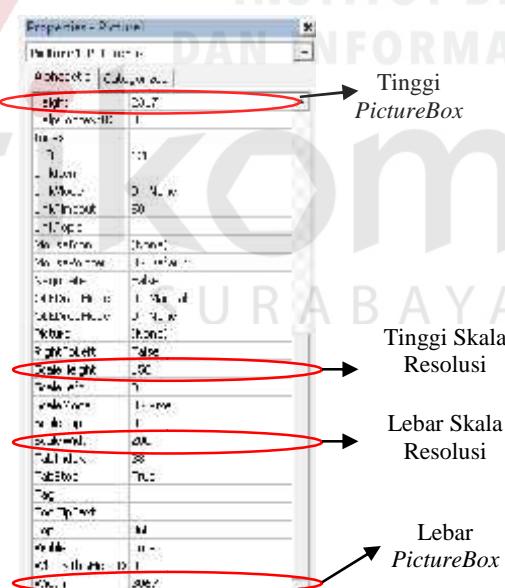
1. Arahkan kursor pada *toolbox*, kemudian pilih *frame*
2. Kemudian klik dan *drag* pada *form project*.
3. Kemudian arahkan kursor pada *toolbox*, kemudian pilih *picturebox*.
4. Klik dan *drag* *picturebox* kedalam area *frame*.
5. kemudian set satuan resolusi *picturebox* dalam piksel dengan mengatur *ScaleMode* pada *picturebox* dengan *pixel* seperti Gambar 3.4.



Gambar 3.4 Tampilan Kanvas dan Properti *PictureBox*

6. Kemudian atur lebar dan tinggi dari *picturebox* dengan mengatur nilai *Width* sebesar 3900 dan nilai *Height* sebesar 3900. Hal ini ditujukan agar memperoleh skala resolusi dari *picturebox* sebesar 256 x 256. seperti Gambar 3.5 berikut:

Gambar 3.5 berikut:



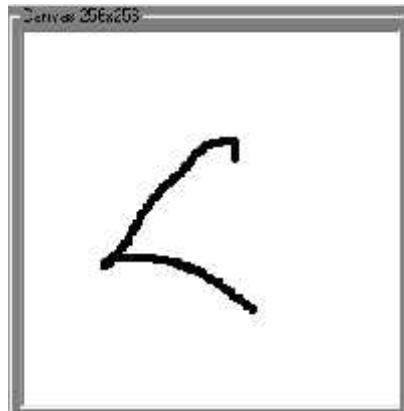
Gambar 3.5 Pengaturan Kanvas (PictureBox)

2. Digitalisasi

Digitalisisasi adalah proses pembacaan warna piksel untuk kemudian dibedakan menjadi 2 nilai. Di dalam sebuah piksel terkandung sebuah informasi yang menentukan warna dari piksel itu sendiri. Informasi tersebut berupa nilai dengan range 0 (hitam) hingga 255 (putih). Dalam hal ini data yang diperlukan untuk mengindikasikan adanya suatu goresan atau gambar pada piksel adalah data yang memiliki nilai mendekati 0. Sehingga diperlukan upaya untuk membuat *tresshold* dengan membuat sebuah kondisi bahwa nilai yang kurang dari 128 adalah data yang disinyalir sebagai warna hitam atau goresan.

3. Pembuatan ROI (*Region Of Interest*) dan *Scalling / Resize*

Proses ROI berfungsi untuk mengambil bagian (piksel) yang terdapat gambar dan mengabaikan bagian (piksel) dari kanvas yang dirasa tidak perlu untuk diolah datanya. Sebelum citra masuk ke dalam proses ekstraksi *Centroid Feature*, citra tersebut terlebih dahulu diproses melalui penentuan ROI. Proses ini dimaksudkan apabila *user* menulis angka jawa lebih kecil atau tidak tepat ukurannya sesuai dengan ukuran citra yang disediakan (bingkai), maka akan dilakukan *cropping* terhadap citra agar lebih fokus pada citra yang akan di ekstrak.



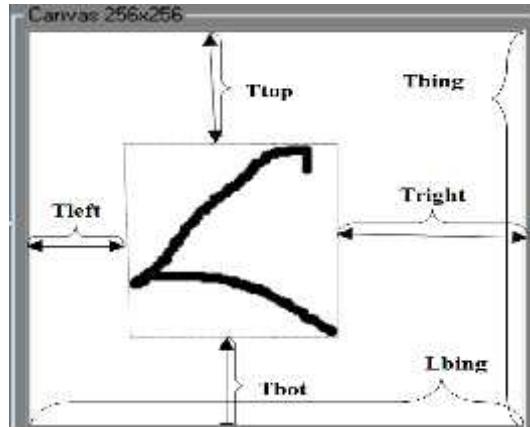
Gambar 3.6 Citra yang dimuat atau digambar di dalam canvas

Proses *cropping* ini dilakukan dengan melakukan scanning terlebih dahulu pada citra yang telah dimuat atau digambar di dalam canvas untuk mengetahui batas tepi suatu gambar atau goresan terhadap bingkai (Gambar 3.6). Dengan demikian tinggi gambar dapat diketahui dengan menggunakan rumus 3.1 dan 3.2.

Keterangan :

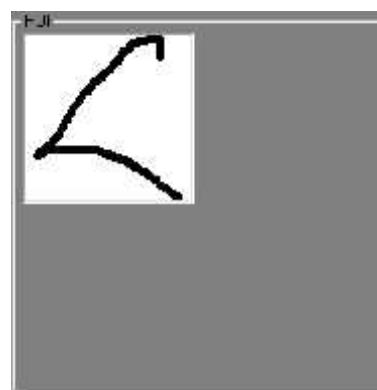
- Tbing = tinggi bingkai (256 piksel)
 - Lbing = lebar bingkai (256 piksel)
 - Ttop = jarak antara gambar paling atas dengan bingkai atas
 - Tbot = jarak antara gambar paling bawah dengan bingkai bawah
 - Lleft = jarak antara gambar paling kiri dengan bingkai kiri
 - Lright = jarak antara gambar paling kanan dengan bingkai kanan

Gambaran dari rumus 3.1 dan 3.2 dapat dilihat seperti yang ditunjukkan pada gambar 3.7.



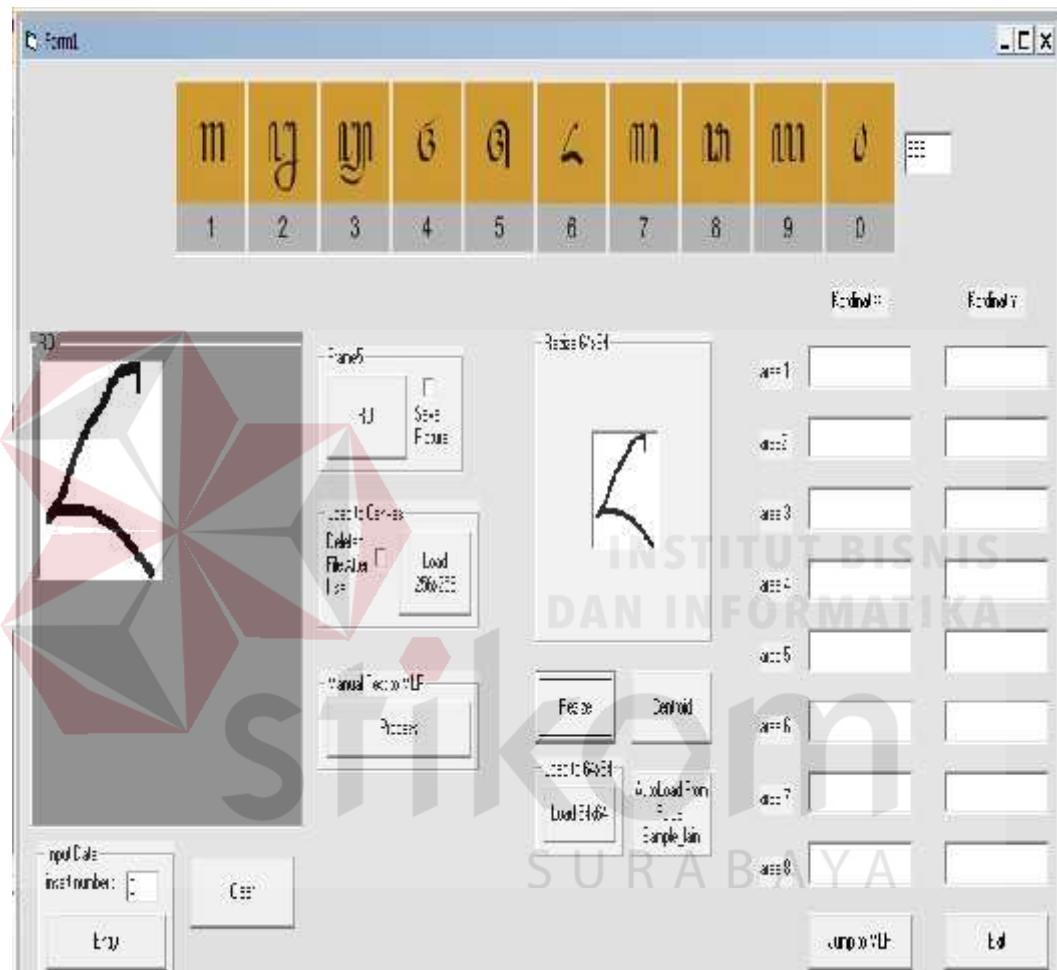
Gambar 3.7 Mencari ukuran gambar yang akan di-cropping

Kemudian dilakukan perbandingan antara tinggi gambar dengan lebar gambar untuk menentukan panjang sisi dari sebuah bingkai yang akan digunakan sebagai wadah baru bagi citra yang telah melalui proses *cropping*. Sehingga bingkai yang terbentuk tersebut memiliki panjang sisi yang sama. Selain itu ditentukan pula penempatan gambar tepat di tengah – tengah bingkai dengan cara mencari panjang tepi gambar yang kosong terlebih dahulu, yakni selisih antara tinggi gambar dengan lebar gambar. Jumlah dari panjang kedua tepi gambar yang kosong tersebut dibagi 2 sehingga hasil yang didapatkan digunakan sebagai posisi awal dalam mencetak gambar. Gambar 3.8 adalah hasil proses ROI.



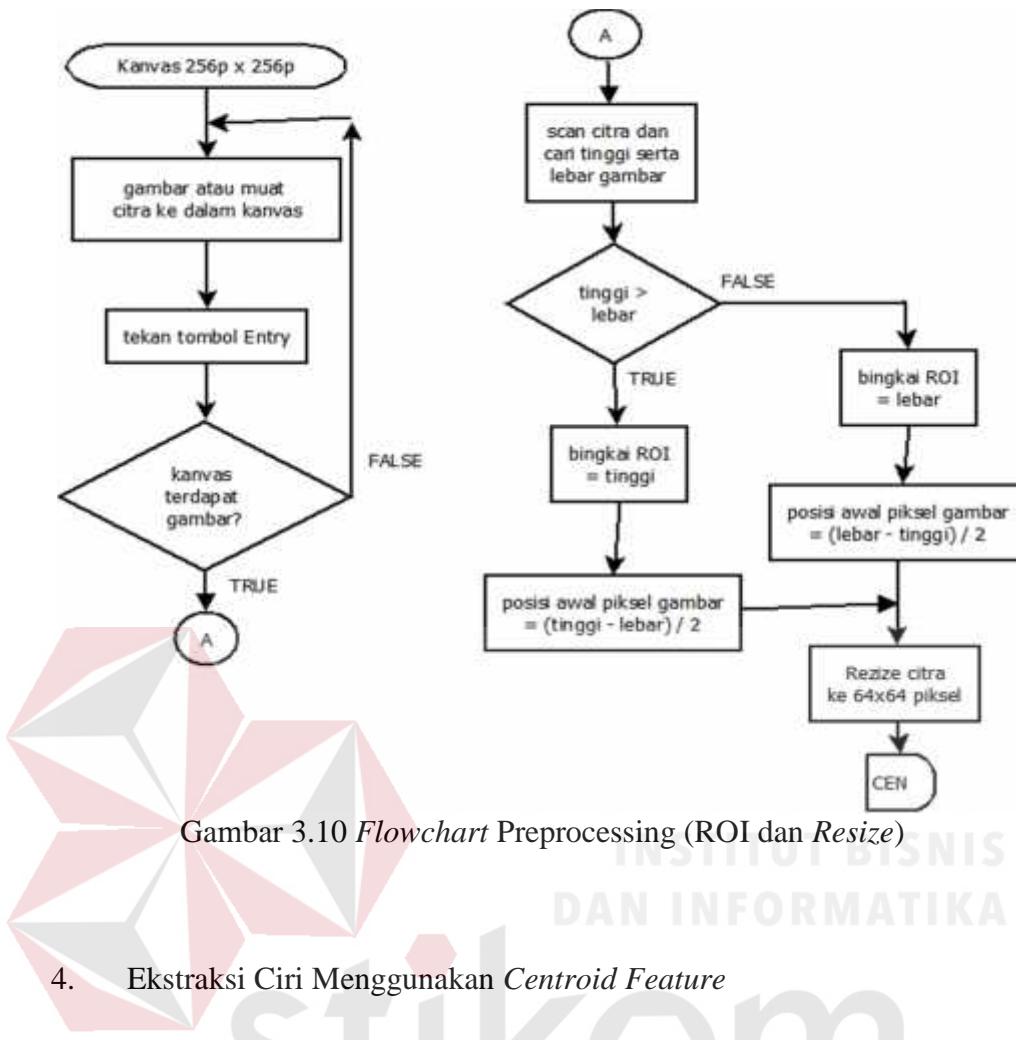
Gambar 3.8 Citra setelah melalui proses ROI

Setelah itu dilakukan proses *resize* pada hasil ROI tersebut dengan ukuran sebesar 64 piksel x 64 piksel. Pola gambar dari citra yang dihasilkan pada tahap ini tidak mengalami perubahan dengan pola gambar aslinya, hal itu dikarenakan citra hanya mengalami perubahan rasio saja (Gambar 3.9).



Gambar 3.9 Citra

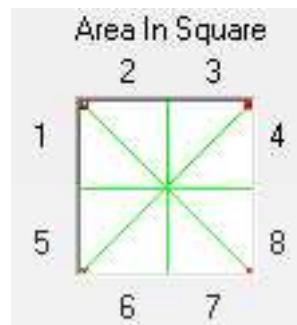
Langkah-langkah untuk melakukan ROI dalam program digambarkan secara umum dalam flowchart pada gambar 3.10.



4. Ekstraksi Ciri Menggunakan *Centroid Feature*

Ekstraksi *Centroid Feature* adalah pengenalan pola dari suatu citra dengan cara mencari posisi titik tengah atau titik berat suatu goresan atau gambar dari citra. Seluruh piksel dari citra tersebut di-scanning untuk memperoleh informasi tentang keberadaan lokasi piksel – piksel yang bernilai kurang dari 128 (warna hitam). Seperti pada penjelasan sebelumnya piksel yang bernilai kurang dari 128 adalah titik hitam (piksel) yang merupakan bagian dari gambar pada sebuah citra. Koordinat dari piksel inilah yang nantinya akan digunakan. Pada saat proses *scanning* berlangsung, secara bersamaan dilakukan pula penyimpanan koordinat piksel tersebut dan dihitung jumlah piksel yang disimpan tadi sesuai

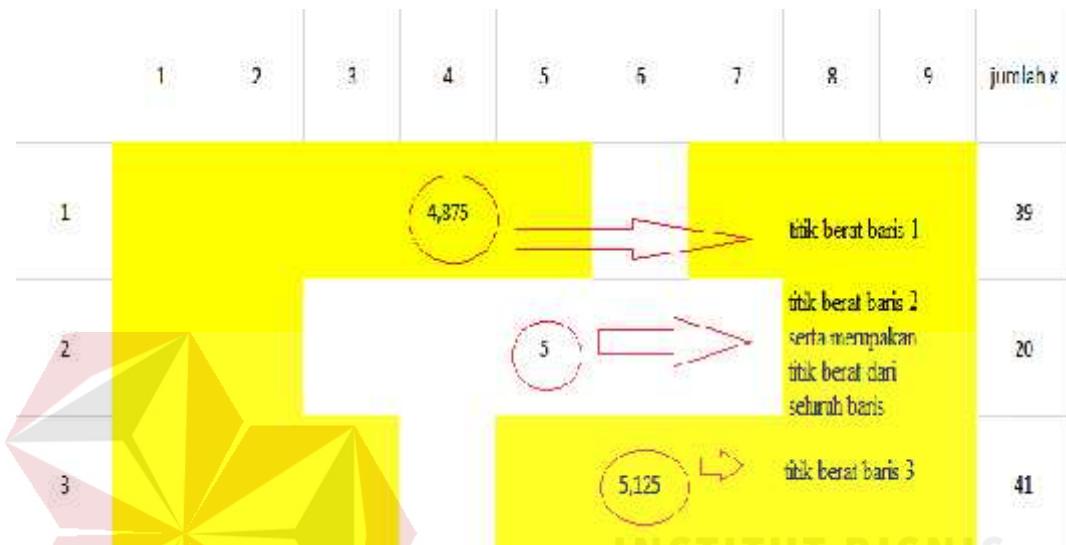
dengan masing – masing sumbunya (x dan y). Penyimpanan tersebut dikelompokkan ke dalam oktannya masing – masing dengan cara menyimpannya kedalam masing – masing indeks yang mewakili tiap – tiap oktan. Untuk menentukan keberadaan piksel terhadap letak oktan atau area adalah dengan mengacu terhadap pembatas yang telah dibuat seperti yang terlihat pada gambar 3.11 berikut ini.



Gambar 3.11 Pembagian citra menjadi 8 oktan

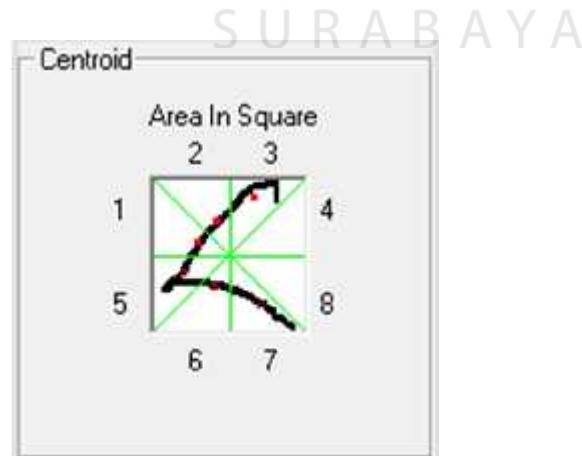
Dari gambar 3.11 tersebut garis vertikal adalah pembatas area yang terletak pada tengah – tengah sumbu x, sedangkan garis horizontal adalah pembatas area yang terletak pada tengah – tengah sumbu y. Khusus untuk garis diagonal diperlukan sebuah variabel bantu. Untuk membentuk diagonal yang menyerong dari kanan atas ke kiri bawah adalah dengan membuat titik – titik piksel yang mana titik piksel tersebut diperoleh dari pertemuan antara nilai sumbu y terhadap nilai sumbu x (nilai sumbu x diperoleh dari 64 dikurangi nilai sumbu y itu sendiri). Sedangkan untuk diagonal yang menyerong dari kiri atas ke kanan bawah merupakan titik – titik piksel yang mana di dalam masing – masing piksel tersebut memiliki koordinat antara nilai sumbu x dan nilai sumbu y yang sama (pertemuan antara sumbu x dengan sumbu y yang bernilai sama).

Selanjutnya proses pencarian titik tengah dilakukan dengan cara menjumlahkan seluruh koordinat yang telah tersimpan tadi dan dibagi dengan jumlah pikselnya. Sehingga didapatkan koordinat titik tengah (sumbu x dan y) di setiap oktan seperti pada contoh yang ditunjukkan oleh gambar 3.12.

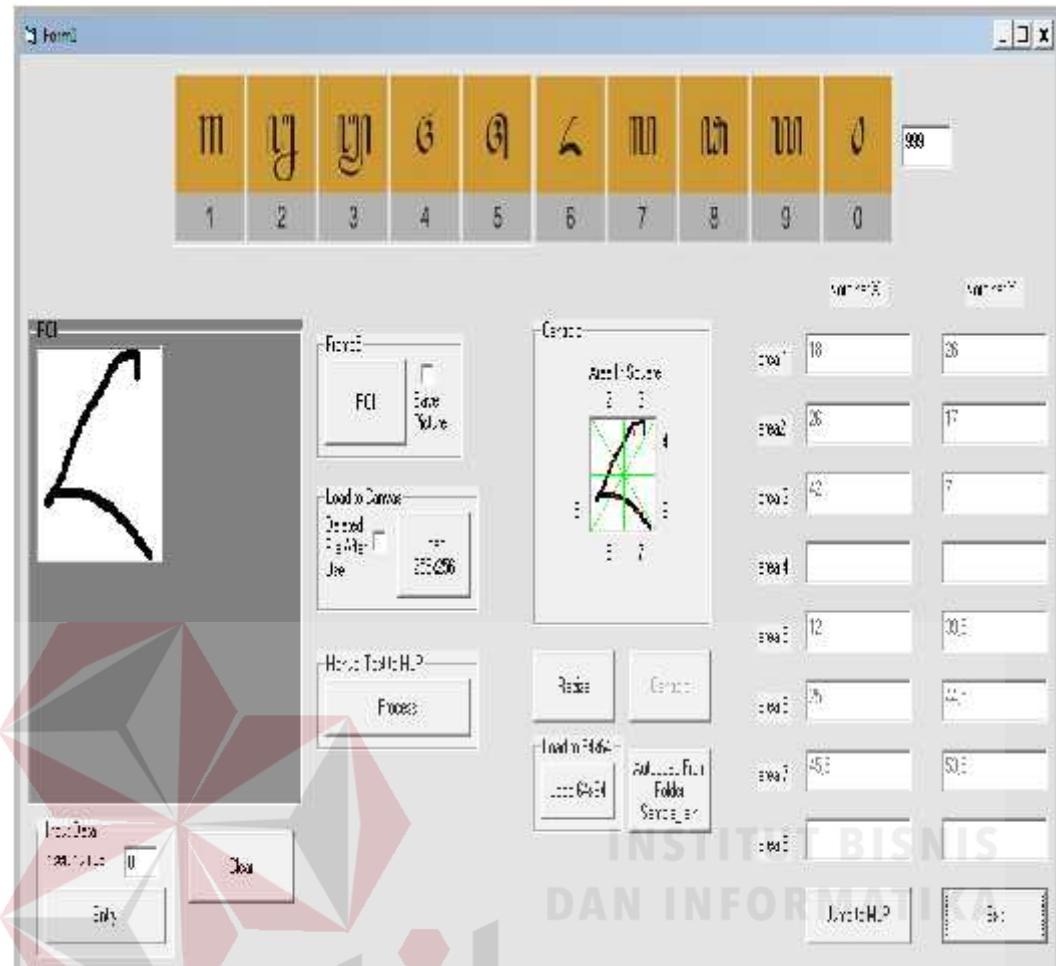


Gambar 3.12 Contoh mencari titik berat dari kotak kuning

Kotak merah pada gambar 3.13 Menunjukkan koordinat hasil titik tengah yang telah didapatkan. Sedangkan nilai titik tengah pada masing – masing oktan dilihat pada gambar 3.14.

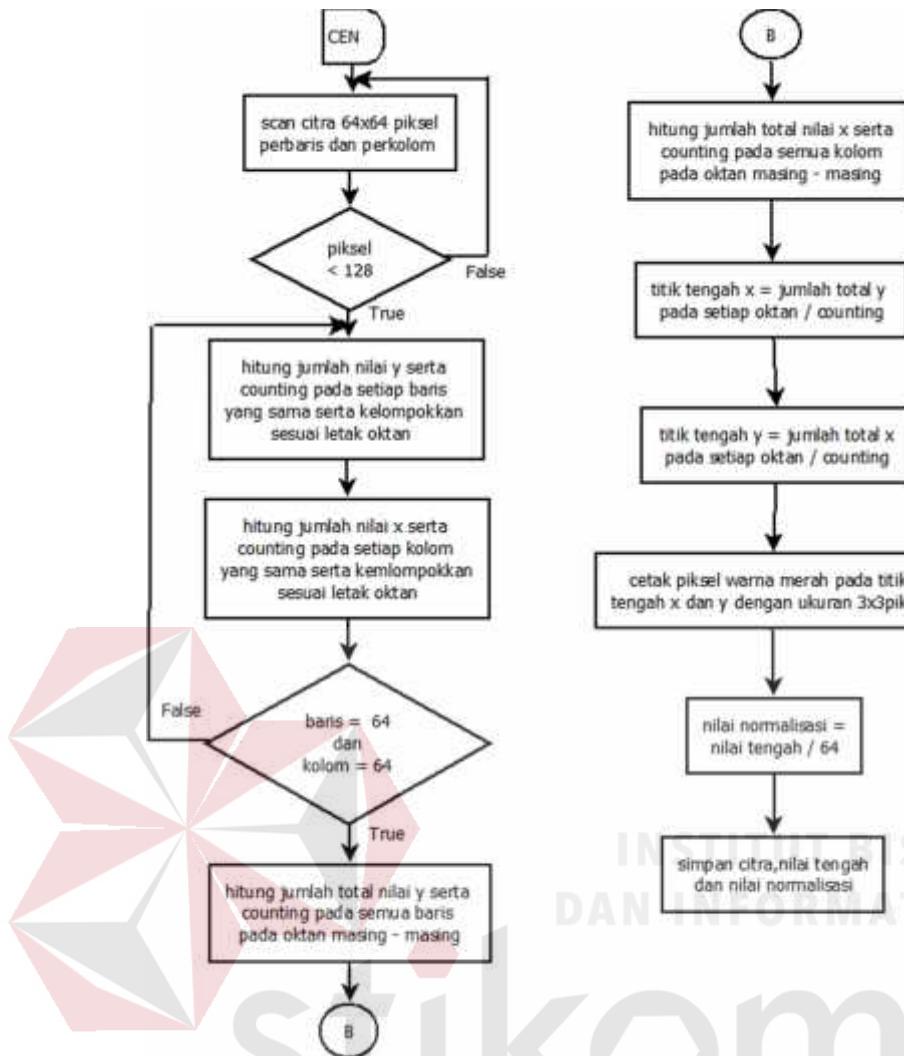


Gambar 3.13 Koordinat titik tengah pada masing – masing oktan



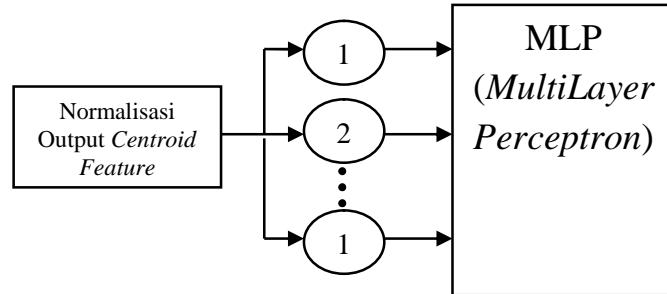
Gambar 3.14 Nilai titik tengah yang telah didapatkan pada setiap oktan

Berikut adalah *flowchart* dari ekstraksi *Centroid Feature* yang ditunjukkan gambar 3.15.

Gambar 3.15 Flowchart ekstraksi fitur *Centroid*

5. Input Multi Layer Perceptron (MLP)

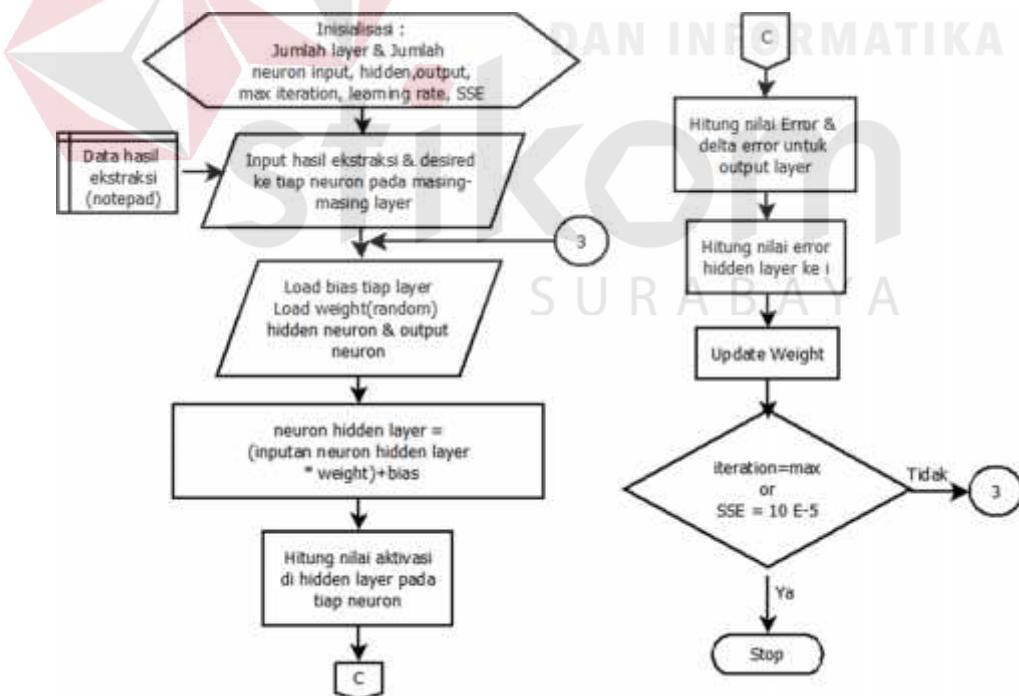
Keluaran dari ekstraksi *Centroid Feature* adalah koordinat titik tengah dari masing – masing oktan yang telah dinormalisasikan. Nilai fitur titik tengah dari tiap oktan ini dinormalisasi dengan cara dibagi dengan jumlah piksel pada masing – masing sumbu.



Gambar 3.16 *Output Dari Centroid Feature Menjadi Inputan MLP*

Jumlah nilai yang telah dinormalisasi berjumlah 16 buah begitu juga dengan *neuron input* dari MLP seperti pada gambar 3.16. Nilai normalisasi tersebut dimasukkan ke dalam MLP untuk dilakukan pembelajaran.

Langkah meng-*inputkan* nilai output ekstraksi ke dalam MLP serta pengolahan data ekstraksi diperlihatkan dalam *flowchart* Gambar 3.17 berikut ini.



Gambar 3.17 *Flowchart MLP*

3.4 Teknik Pengumpulan & Analisis Data

3.4.1 Pengumpulan Data

Pengumpulan data primer dilakukan dengan mengumpulkan data sampel untuk 1 *set* (10 pola angka) yang diperoleh dari hasil tulisan tangan oleh orang yang berbeda. Data tersebut diperoleh dengan 2 cara yaitu diperoleh dengan cara menggambar langsung pada canvas dengan mouse atau dengan cara memuat gambar yang sebelumnya dihasilkan oleh tablet ke dalam canvas. Dikarenakan data yang dibutuhkan sebanyak 100*set* data sampel pembelajaran dan 50*set* data sampel uji untuk masing - masing angka, diperlukan 150 orang dengan masing - masing memberikan 1 *set* sampel tulisan tangannya dalam Angka Jawa.

Pengumpulan data sekunder dilakukan dengan melakukan percobaan di laboratorium setelah perangkat lunak sudah selesai dibuat dan sudah diperoleh bobot yang optimal seperti yang dijelaskan pada sub bab 3.3. Sistem MLP dengan bobot optimal tersebut akan diuji menggunakan pola Angka, baik dari sampel yang digunakan untuk pembelajaran, pengujian maupun citra yang baru digambarkan. Hasil pengujian antara sampel pelatihan, pengujian, dan data baru akan saling dibandingkan.

3.4.2 Pelatihan MLP

Aplikasi ini akan menghasilkan 2 buah file log untuk bobot yang dihasilkan. Bobot hasil akhir pembelajaran yang tercapai apabila syarat error minimum atau iterasi maksimum terpenuhi, akan disimpan ke dalam file *weight.dat*. Selain itu, terdapat file bernama *weight2.dat* untuk menyimpan informasi bobot setiap 100 iterasi. Hal ini diperlukan sebagai tindakan preventif

bila terjadi hal-hal yang dapat menghentikan aplikasi sebelum mencapai akhir pembelajaran, yang mana disebabkan karena komputer *hang* atau komputer mengalami putus aliran listrik.

Proses pembelajaran dilakukan dengan konfigurasi sebagai berikut:

- Jumlah hidden layer : 3
- Jumlah neuron input : 16
- Jumlah neuron layer 1 : 47
- Jumlah neuron layer 2 : 47
- Jumlah neuron layer 3 : 47
- Jumlah neuron output : 10
- Laju pembelajaran (μ) : 0.1
- Error minimum : $10E-4$
- Iterasi maksimum : $10E+5$

Proses pembelajaran dilakukan sebanyak 1000001 iterasi dengan menggunakan 100 set sampel input dengan nilai error akhir sebesar 2.00036739554103. Ini berarti bahwa proses pembelajaran selesai hingga target error minimum terhadap data sampel terpenuhi. Bobot yang dihasilkan akan disimpan pada file *weight.dat*.

3.4.3 Analysis Data

Data yang sudah terkumpul baik citra maupun hasil normalisasi ekstraksi Centroid(*notepad*) akan dilatihkan terhadap MLP. Dan dilakukan pengujian kembali pada data sampel yang diperoleh untuk menghitung nilai *error* pada perangkat lunak setelah pembelajaran.

Setelah menganalisa data sampel untuk pelatihan berjumlah 100 *set* tulisan angka jawa, proses selanjutnya adalah analisa data sampel pengujian. Sampel pengujian adalah data yang belum pernah dilatihkan pada MLP sebelumnya. Data tersebut diperoleh dari *participant* sebanyak 50 *set* tulisan angka jawa yang telah disimpan hasil normalisasi ekstraksinya. Kemudian dilanjutkan dengan pengujian secara langsung pada canvas dengan citra tulisan angka jawa yang baru dicetak saat itu juga. Hasil yang didapatkan pada saat menguji setiap sampel adalah pengenalan benar atau pengenalan salah.



BAB IV

PENGUJIAN SISTEM

4.1 Pengujian Perangkat Lunak

Dalam mengetahui simulasi ini dapat berjalan atau tidak maka diperlukan tahap pengujian. Alat dan bahan yang digunakan sebelum pengujian:

1. *Personal Computer (PC) / Laptop.*
2. *Software Visual Basic 6.0.*
3. *Microsoft Excel 2007.*
4. *MATLAB 7.0*
5. *Software Pixilion*

Berikut merupakan hasil pengujian dari tiap modul yang telah dibuat:

4.1.1 Pengujian Picturebox Sebagai Media Menggambar

Pengujian ini dilakukan dengan menekan klik kiri dan menahannya pada *mouse* di dalam *picturebox* lalu menggeserkannya untuk membuat goresan yang diinginkan.

A. Tujuan

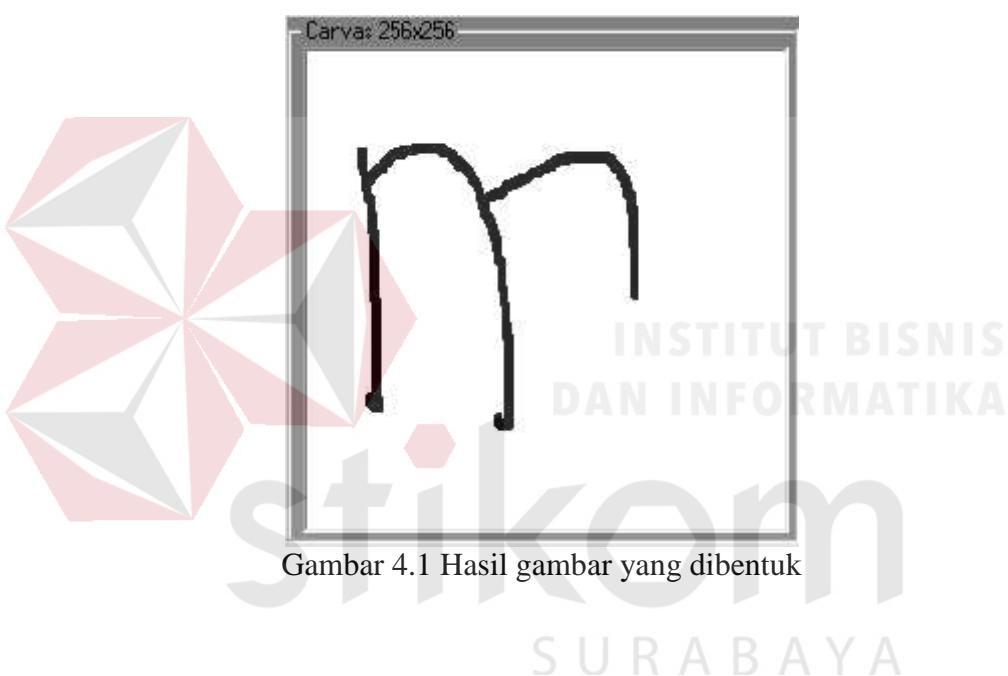
Pengujian ini bertujuan untuk mengetahui apakah gambar dapat dibentuk di dalam *picturebox*. *Picturebox* ini berfungsi sebagai media untuk membuat sebuah gambar berupa angka yang nantinya dapat diekstrak fitur titik tengahnya. Ekstraksi fitur titik tengah tersebut digunakan baik sebagai keperluan sampel pelatihan maupun pengujian pola gambar.

B. Prosedur pengujian

1. Menjalankan aplikasi

2. Mengarahkan kursor ke area picturebox
 3. Klik kiri pada *mouse* lalu tahan untuk mengaktifkan mode menggambar
 4. Dan lepas klik kiri pada *mouse* untuk mematikan mode menggambar
 5. Geser kursor untuk membuat goresan
- C. Hasil Pengujian

Hasil gambar yang telah dibentuk di dalam *picturebox* dapat dilihat pada gambar 4.1.



D. Pembahasan

Hasil pengujian menunjukkan bahwa *picturebox* dapat digambar menggunakan *pointer mouse*

4.1.2 Pengujian Picturebox Sebagai Media Pemuat File Gambar

Pengujian ini dilakukan dengan mengambil *file* gambar dengan kualitas *bmp* ke dalam *picturebox*.

A. Tujuan

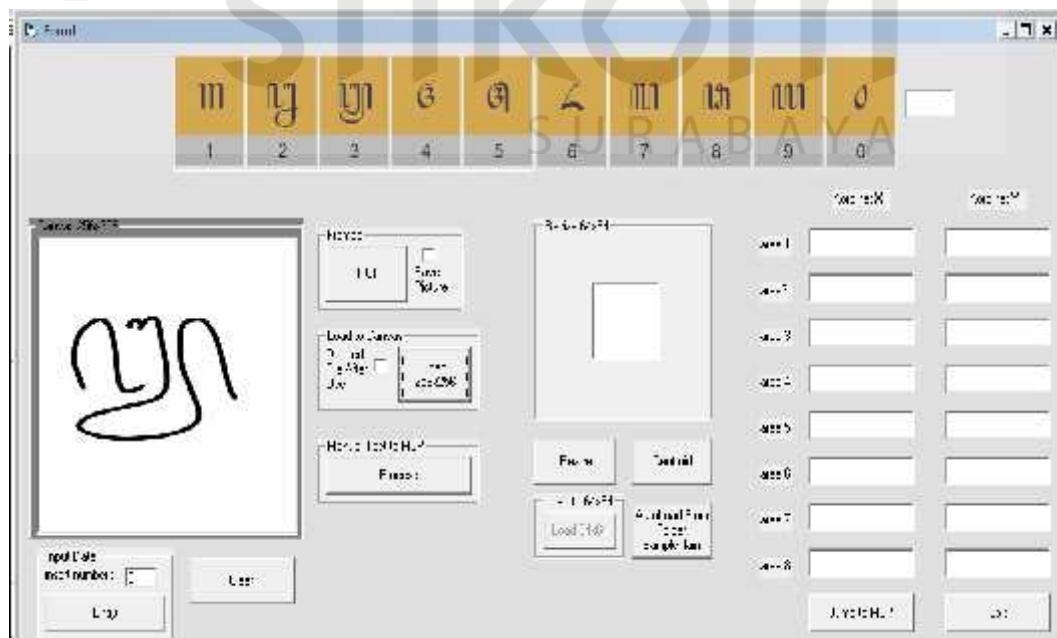
Pengujian ini bertujuan untuk mengetahui apakah gambar berkualitas bmp dengan ukuran 64 x 64 piksel dan 256 x 256 piksel dapat disimpan ke dalam *picturebox*.

B. Prosedur pengujian

1. Menjalankan aplikasi
2. Tekan tombol “Load 64x64” untuk mengambil *file* gambar berukuran 64 x 64 piksel atau tekan tombol “Load 256x256” untuk mengambil *file* gambar berukuran 256 x 256 piksel.
3. Khusus untuk “Load 256x256” terdapat opsi untuk menghapus *file* yang dipilih pada saat penyimpanan *file* dilakukan (tombol “Entry”).

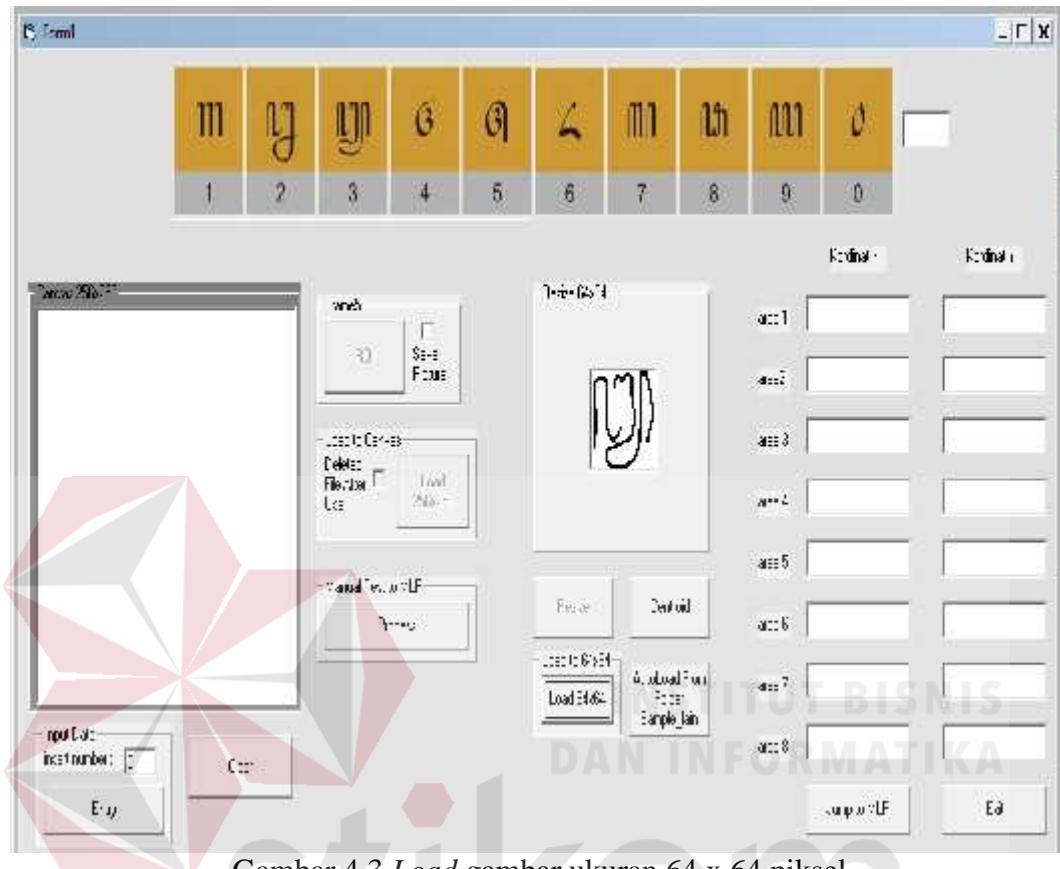
C. Hasil Pengujian

Hasil dari gambar bmp berukuran 256 x 256 piksel yang berhasil dimuat ulang ke dalam *picturebox* “Canvas 256x256” ditunjukkan pada gambar 4.2.



Gambar 4.2 *Load* gambar ukuran 256 x 256 piksel

Sedangkan hasil dari gambar bmp berukuran 64 x 64 piksel yang berhasil dimuat ulang ke dalam *picturebox* “Rezize 64x64” ditunjukkan pada gambar 4.3.



Gambar 4.3 Load gambar ukuran 64 x 64 piksel

D. Pembahasan

Hasil pengujian menunjukkan bahwa *picturebox* dapat di-load dengan gambar. Gambar yang dimuat berupa gambar yang sesuai dengan ukuran piksel pada masing – masing *picturebox*.

4.1.3 Pengujian Region of Interest(ROI)

Pengujian ini dilakukan dengan menampilkan gambar hasil *cropping* pada *picturebox*.

A. Tujuan

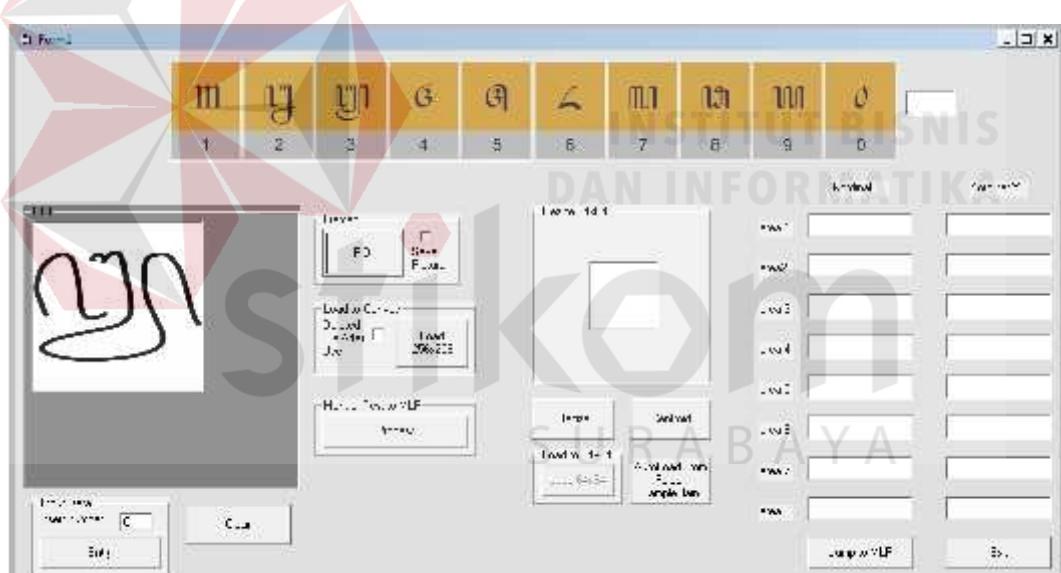
Pengujian ROI dilakukan untuk memastikan bahwa citra berhasil di-*cropping* serta penempatan gambar tepat berada di tengah citra.

B. Prosedur Pengujian

1. Pastikan canvas 256x256 telah tercetak sebuah gambar.
2. Tekan tombol “ROI” di sebelah kanan Canvas.
3. Centang *checkbox* “Save Picture” untuk menyimpan gambar aslinya.

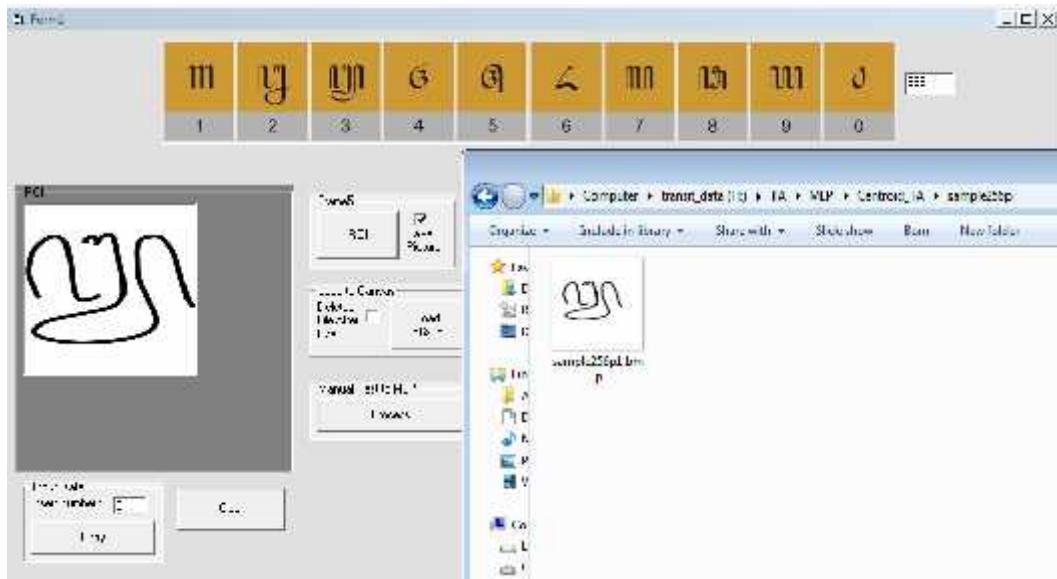
C. Hasil pengujian

Hasil gambar yang telah diproses melalui ROI dapat dilihat pada gambar 4.4.



Gambar 4.4 Hasil proses ROI

Hasil penyimpanan gambar asli yang belum mengalami proses ROI ditunjukkan pada gambar 4.5.



Gambar 4.5 Hasil penyimpanan sampel gambar

D. Pembahasan

Hasil pengujian menunjukkan bahwa proses ROI berhasil dilakukan. Hal itu ditunjukkan dengan posisi terpanjang gambar yaitu pada ukuran lebar, posisi tepi gambar berdempetan dengan bingkai citra yang berukuran persegi. Selain itu posisi gambar juga tepat berada pada tengah – tengah gambar.

4.1.4 Pengujian Rezize

Pengujian ini dilakukan dengan menampilkan gambar hasil penyesuaian ukuran ke dalam *picturebox* dengan skala 64 x 64 piksel.

A. Tujuan

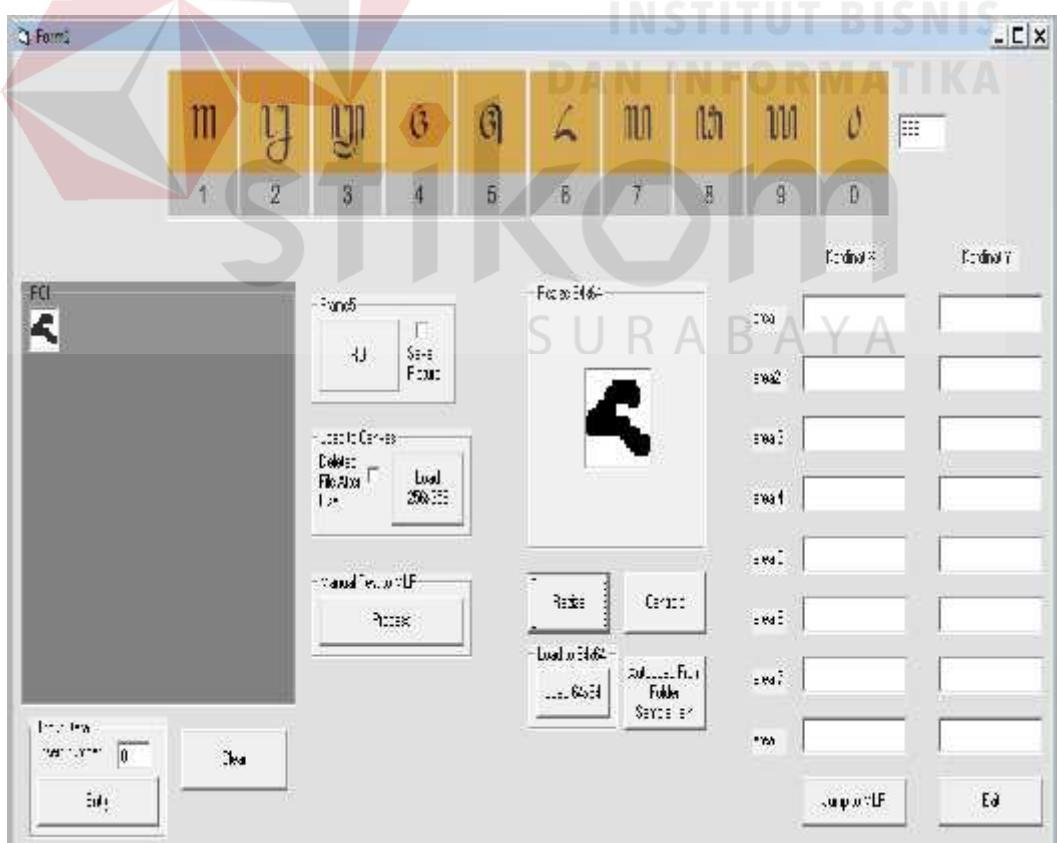
Pengujian *rezize* dilakukan untuk mengetahui apakah ukuran piksel gambar dapat dibesarkan maupun dikecilkan menjadi 64 x 64 piksel.

B. Prosedur Pengujian

1. Pastikan gambar di dalam *picturebox* “Canvas 256x256” telah selesai diproses melalui ROI.
2. Tekan tombol “Rezize” di bawah *picturebox* “Rezize 64x64”.
3. Jalankan *software* Pixilion.exe.
4. Open direktori untuk load gambar dari pixilion yang akan diuji.
5. Kemudian pada pixilion klik *Effect* untuk mengatur *height* dan *width* citra untuk dikonversi menjadi 64 x 64 piksel. Lalu klik *Convert*.

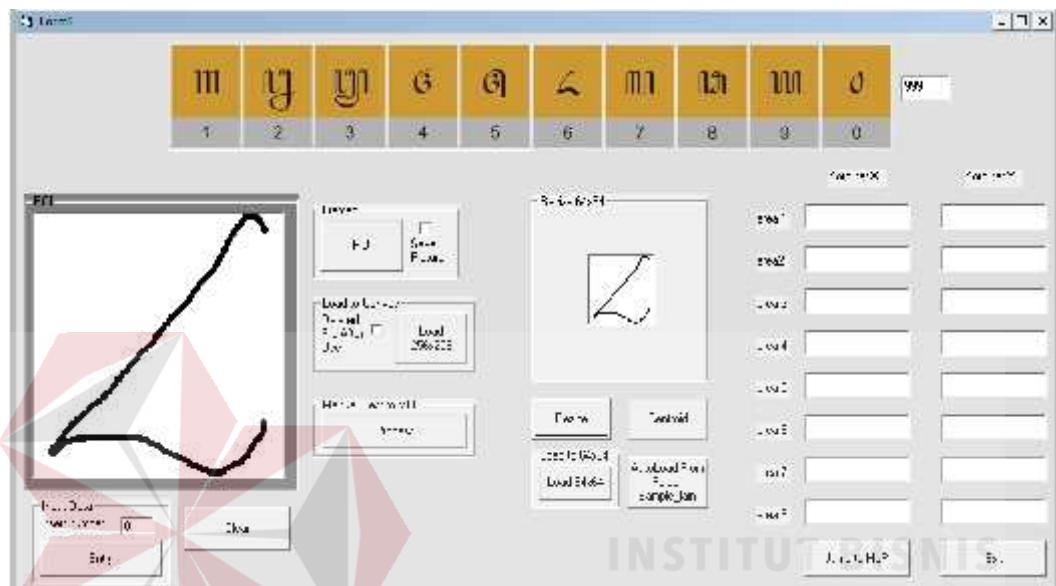
C. Hasil pengujian

Hasil dari percobaan pertama untuk gambar yang berukuran kurang dari 64 piksel x 64 piksel dapat dilihat pada gambar 4.6.

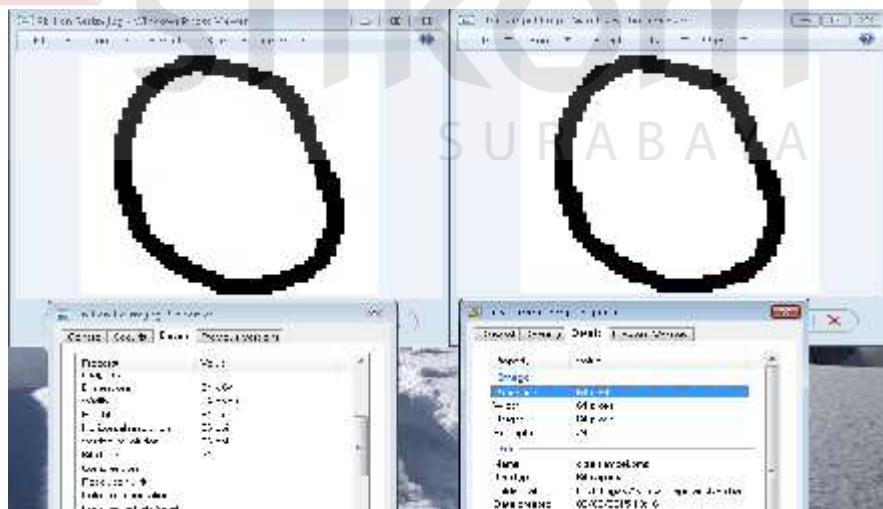


Gambar 4.6 *Rezize* pada gambar yang berukuran lebih kecil dari 64 x 64 piksel

Hasil dari percobaan kedua untuk gambar yang berukuran lebih dari 64 x 64 piksel dapat dilihat pada gambar 4.7, sedangkan untuk mengetahui perbandingan yang dihasilkan oleh proses *rezize* dengan hasil yang diperoleh oleh *software pixilion* ditunjukkan oleh gambar 4.8.



Gambar 4.7 *Rezize* pada gambar yang berukuran lebih besar dari 64 x 64 piksel



Gambar 4.8 Perbandingan *rezize* antara hasil sampel *rezize* dengan *software Pixilion*

D. Pembahasan

Hasil pada pengujian menunjukkan bahwa proses *rezize* berhasil dengan baik. Baik gambar yang berukuran lebih kecil maupun lebih besar dari 64 x 64 piksel, berhasil dirubah menjadi ukuran 64 x 64 piksel. Pembuktian tersebut dibandingkan dengan *rezize* yang dilakukan melalui aplikasi *pixilion* untuk merubah ukuran gambar menjadi 64 x 64 piksel.

4.1.5 Pengujian Aplikasi *Centroid Feature*

Pengujian ini dilakukan dengan melihat posisi dari titik yang terbentuk pada masing – masing oktan.

A. Tujuan

Pengujian metode *Centroid Feature* dilakukan untuk mengetahui apakah hasil ekstraksi yang dilakukan oleh aplikasi telah sesuai, yaitu posisi titik tengah berada tepat pada bagian tengah gambar.

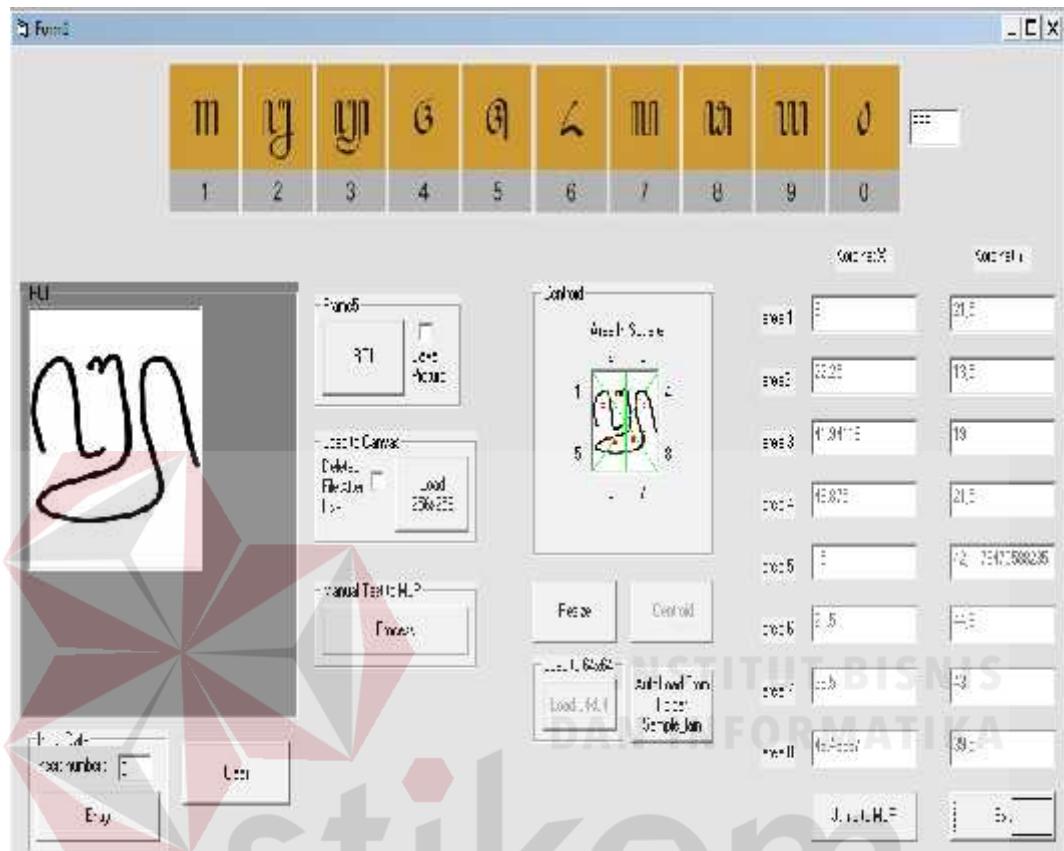
B. Prosedur Pengujian

1. Pastikan gambar telah melewati Proses ROI dan Rezize.
2. Setelah melewati semua prosedur pada point 1, tekan tombol “Centroid”.
3. Tekan tombol Entry untuk menyimpan data berupa gambar 64 x 64 piksel, nilai hasil ekstraksi beserta nilai yang telah dinormalisasikan.

C. Hasil Pengujian

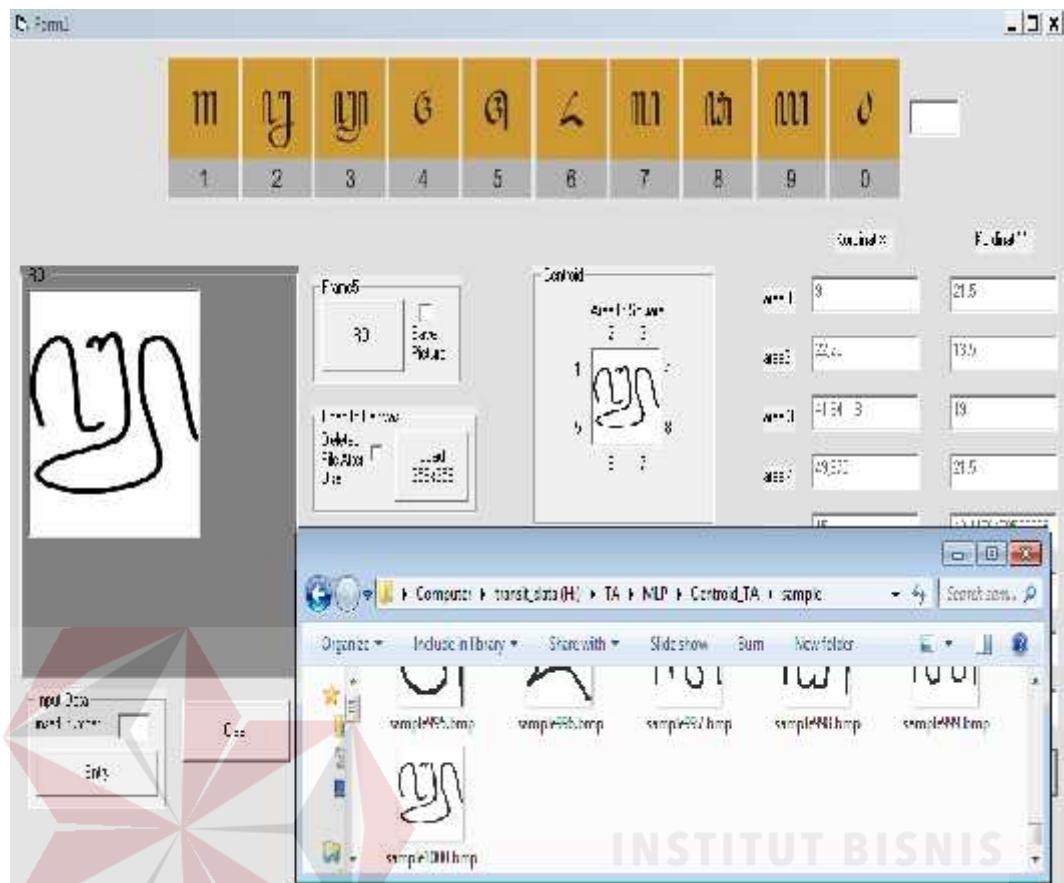
Pada hasil pengujian ini terdapat garis hijau yang merupakan pembagi dari masing – masing area citra. Dari masing – masing citra tersebut terdapat titik

– titik gambar berwarna merah seluas 3×3 piksel, titik tersebut yang merupakan letak titik tengah pada tiap area. Hasil tersebut dapat dilihat pada gambar 4.9.



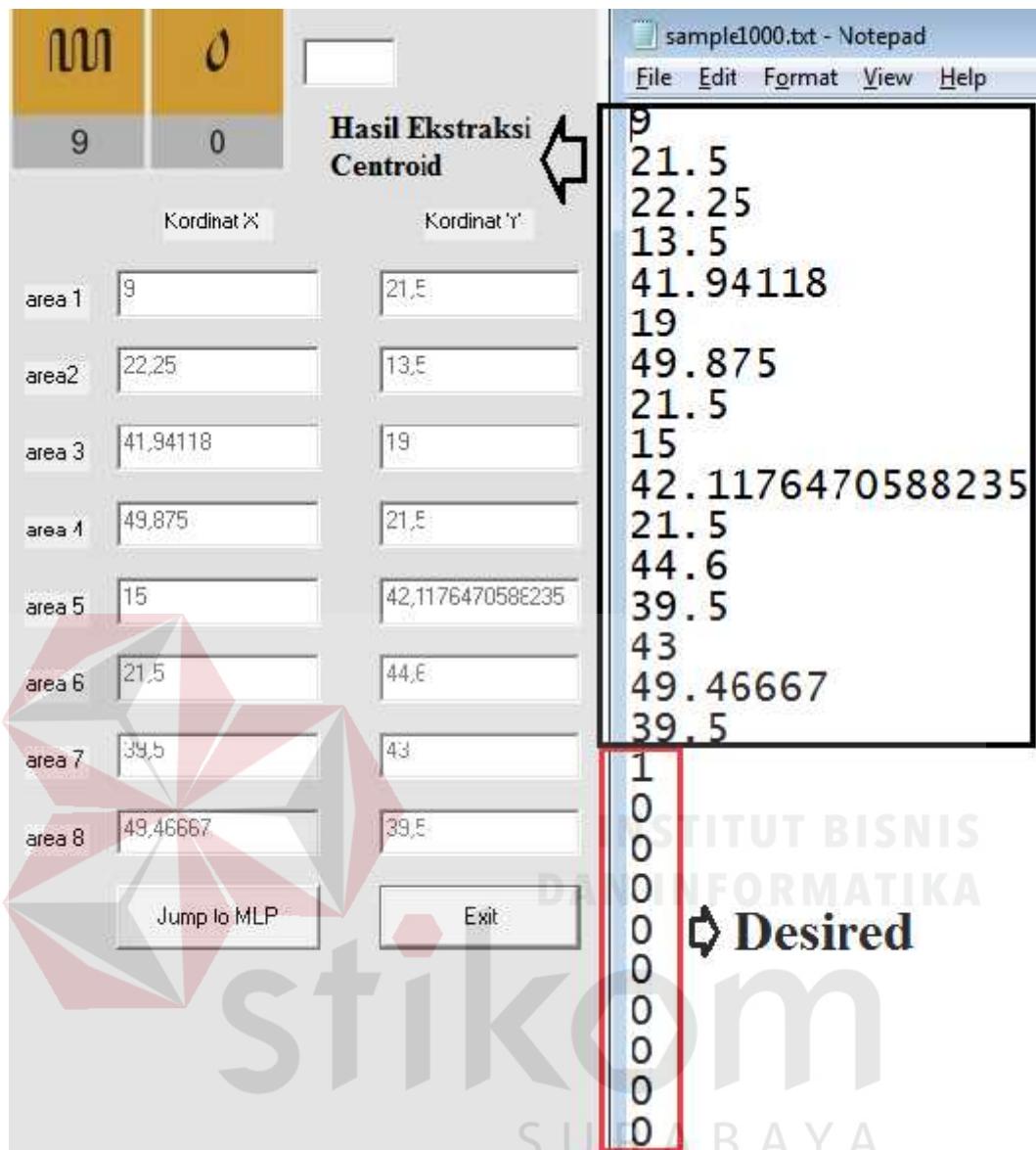
Gambar 4.9 *Centroid* yang dihasilkan

Penyimpanan gambar yang merupakan sampel dari ekstraksi disimpan dalam format .bmp dengan ukuran 64×64 piksel. Hasil penyimpanan yang dimaksudkan dapat dilihat pada gambar 4.10.



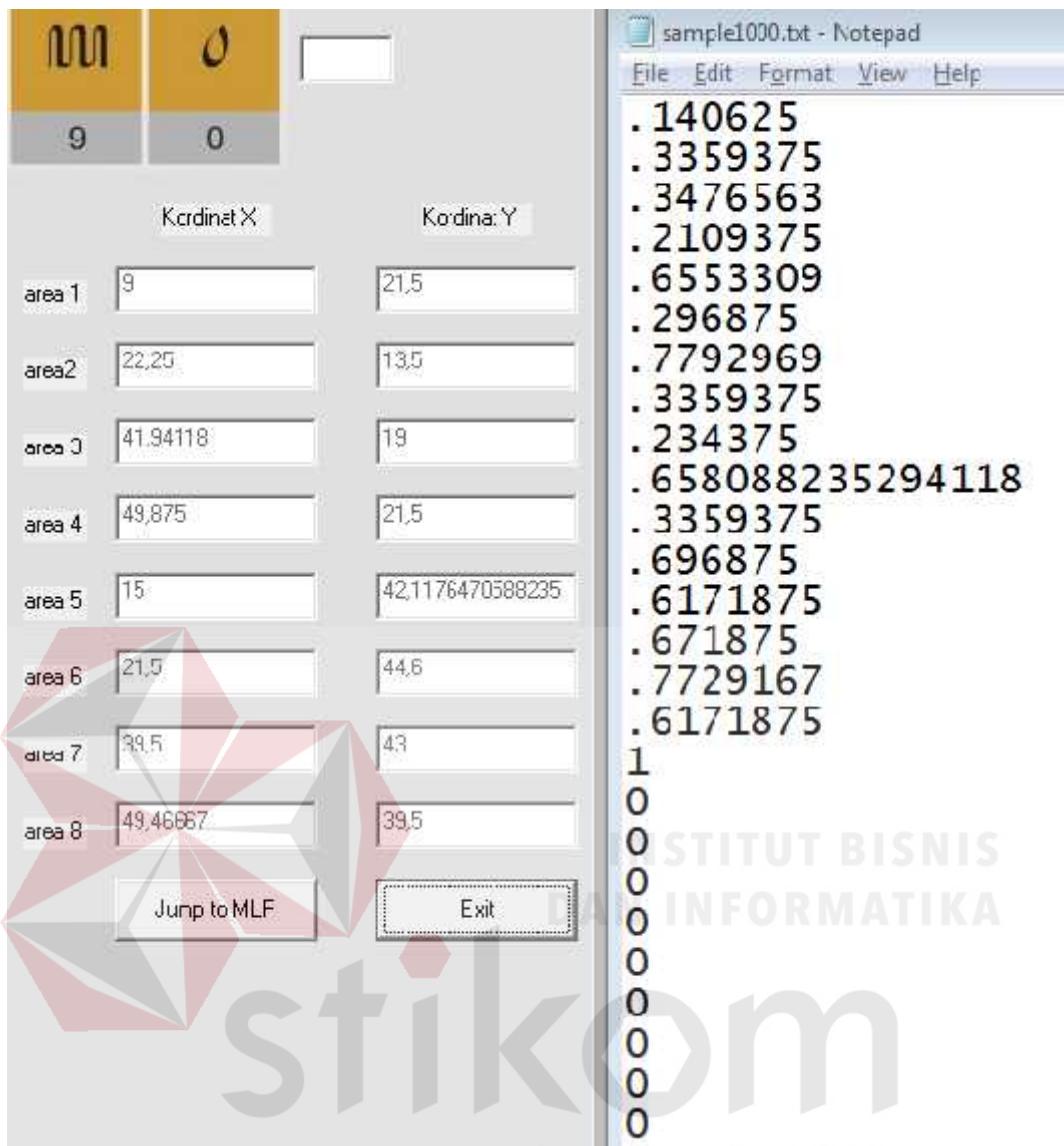
Gambar 4.10 Penyimpanan gambar sampel Centroid

Penyimpanan nilai hasil ekstraksi disimpan dalam bentuk format .dat. Nilai ini merupakan titik tengah pada masing – masing area citra. Hasil penyimpanan yang dimaksudkan dapat dilihat pada gambar 4.11.



Gambar 4.11 Penyimpanan data hasil ekstraksi *Centroid*

Penyimpanan nilai hasil ekstraksi yang telah dinormalisasikan adalah hasil dari pembagian nilai titik tengah dengan panjang dari citra (64). Hasil penyimpanan tersebut dapat dilihat pada gambar 4.12.



Gambar 4.12 Penyimpanan data hasil ekstraksi *Centroid* yang telah dinormalisasi

D. Pembahasan

Hasil pengujian menunjukkan bahwa ekstraksi *Centroid* berhasil dilakukan. Seperti yang ditunjukkan pada gambar 4.9, posisi kotak merah yang mewakili titik tengah dari gambar berada tepat di tengah – tengah gambar. Masing – masing titik tengah dari setiap oktan disimpan kedalam sebuah notepad sehingga memudahkan proses *training* nantinya.

4.1.6 Pengujian Aplikasi Pelatihan Menggunakan MLP

Pengujian ini dilakukan dengan melihat parameter kesalahan (*Sum Square Error*) dari pengenalan, yaitu selisih antara nilai masukan dengan nilai keluaran (angka yang dikehendaki) pada proses MLP. SSE tersebut berasal dari kesalahan – kesalahan yang terjadi pada setiap pengenalan angka pada masing – masing sampel yang kemudian diakumulasikan.

A. Tujuan

Pengujian pelatihan ini dilakukan untuk mengetahui apakah proses pelatihan menggunakan MLP ini dapat berjalan dan menghasilkan tingkat pengenalan dengan akurasi yang tinggi.

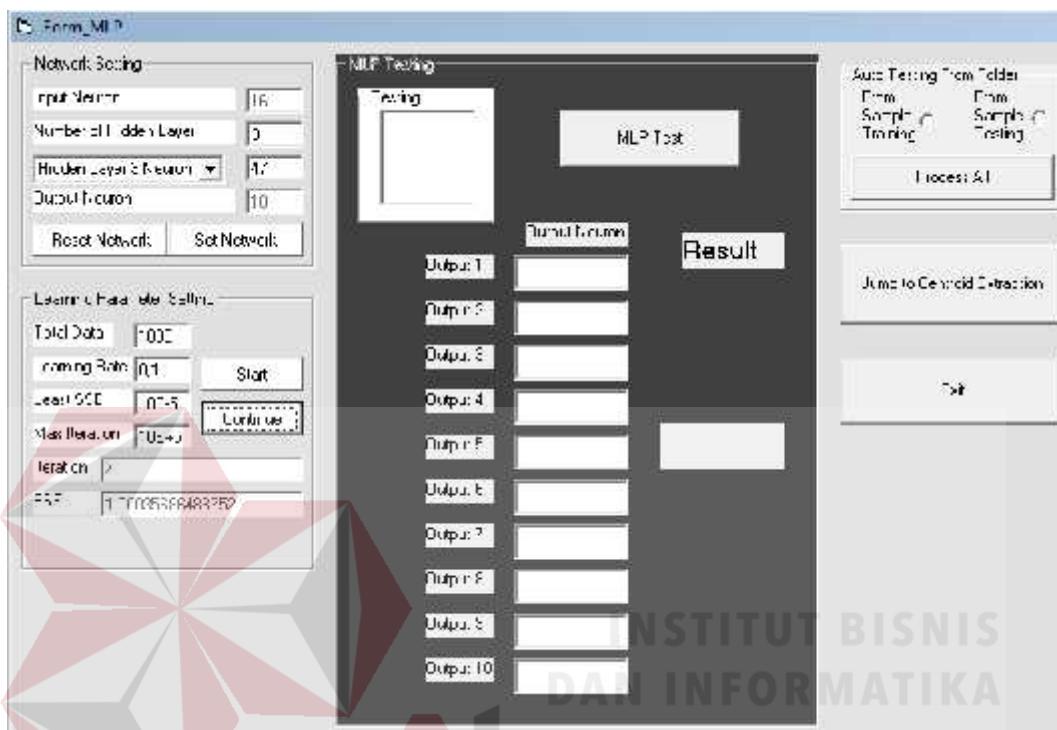
B. Prosedur Pengujian

1. Menjalankan aplikasi.
2. Tekan tombol “Jump to MPL” untuk menampilkan aplikasi MLP
3. Atur “Number Hidden Layer” pada frame “Network setting” untuk menentukan jumlah *hidden layer* yang akan digunakan.
4. Atur “Hidden Layer Neuron” pada frame yang sama untuk menentukan masing – masing jumlah *neuron* pada *layer* tertentu.
5. Tekan tombol “Start” pada frame “Layer Parameter Setting”.
6. Amati kotak “Iteration” untuk informasi perulangan proses dan kotak “SSE” untuk melihat tingkat *eror* yang dicapai. .

C. Hasil Pengujian

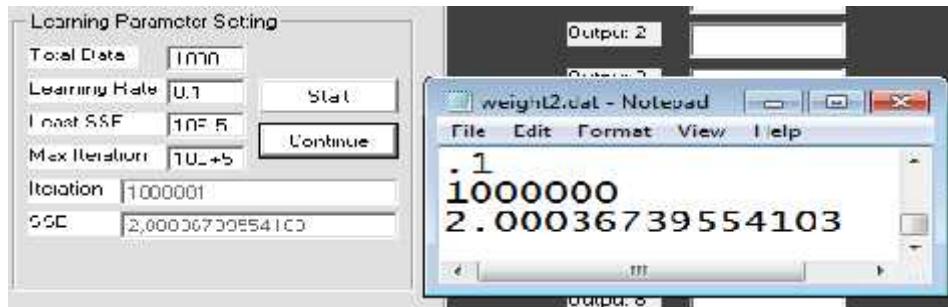
Hasil yang diharapkan adalah pencapaian SSE kurang dari 1×10^{-5} (*Convergent*). Akan tetapi pada percobaan ini penulis hanya melakukan pelatihan

dengan nilai SSE mencapai 1.20217253857658E-03 dengan *iteration* sebesar 194700. Hasil yang ditunjukkan oleh aplikasi saat percobaan pelatihan dapat dilihat pada gambar 4.13.



Gambar 4.13 Proses *training* pada MLP

Saat pelatihan mencapai *iteration* ke 100 maka data sementara hasil pelatihan tersebut akan disimpan kedalam *file* dengan format .dat. Hal ini berfungsi sebagai *backup* data yang nantinya pelatihan dapat dilanjutkan kembali tanpa mengulang proses pelatihan dari awal apabila proses pelatihan berhenti secara tiba – tiba atau dihentikan sementara. Data yang disimpan berupa informasi tentang jumlah *layer* dan jumlah *neuron*, nilai bobot yang diperoleh, nilai bias, perata pelatihan, *iteration* dan SSE. Hasil dari data yang diperoleh dapat dilihat pada gambar 4.14.



Gambar 4.14 Hasil penyimpanan data *training*

D. Pembahasan

Dari hasil pelatihan aplikasi tersebut telah terlihat bahwa aplikasi telah berhasil menjalankan proses *training*. Jumlah data yang dilatihkan atau *training* adalah 1000 sampel hasil ekstraksi dari sampel *training* yang berupa file *notepad*. Didalam masing – masing sampel tersebut telah terkandung informasi tentang ekstraksi titik tengah beserta *desired*-nya. Pada proses *training* ini dilakukan pelatihan selama iterasi mencapai 1000001 kali dengan hasil SSE (*Sum Square Error*) mencapai 2.00036739554103.

4.1.7 Pengujian Pengenalan Angka Jawa

Merupakan tujuan dari tugas akhir ini yang mana dilakukan dengan melihat hasil tebakan aplikasi akan masukan dari gambar yang diujikan. Aplikasi akan menunjukkan sebuah gambar angka yang disinyalir cocok dengan pola yang diidentifikasi pada gambar uji.

A. Tujuan

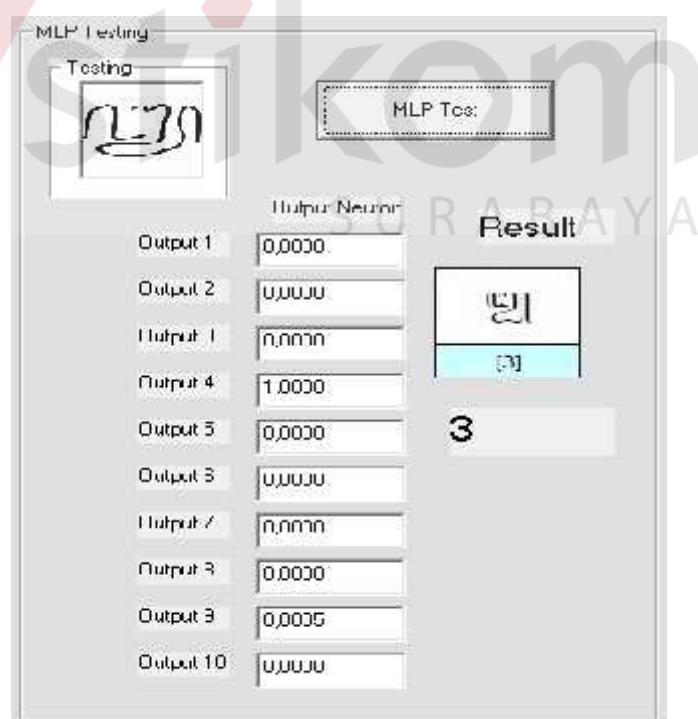
Pengujian pengenalan angka jawa dilakukan untuk mengetahui keberhasilan aplikasi dalam mengenali pola – pola angka yang diujikan.

B. Prosedur Pengujian

1. Menjalankan aplikasi
2. Cetak gambar pada *picturebox* “Canvas 256x256” atau muat gambar baik ke dalam *picturebox* “Canvas 256x256” ataupun ke dalam *picturebox* “Rezize 64x64”.
3. Tekan tombol “Process”
4. Setelah masuk ke dalam aplikasi MLP dan gambar berhasil di cetak ke dalam *picturebox* “Testing” maka tekan tombol “MLP Test”.

C. Hasil Pengujian

Hasil dari pengujian ini adalah informasi tentang hasil pengenalan aplikasi terhadap sampel gambar angka yang diujikan, yaitu berupa hasil keluaran *neuron output* serta tampilan angka yang ditebak. Hasil tersebut dapat dilihat pada gambar 4.15.



Gambar 4.15 Hasil pengenalan angka jawa

D. Pembahasan

Dari proses pengenalan uji coba tersebut telah diketahui hasil pengenalan dari aplikasi adalah dengan menampakkan gambar angka pada gambar *Result* serta berupa angka yang ditunjukkan pada baris di bawah gambar tersebut. Pada kotak – kotak di sebelah kiri gambar *result* tersebut merupakan hasil masing – masing *neuron output* yang telah melalui proses perhitungan *input neuron* dari data sampel dengan bobot.

4.1.8 Pengujian Metode *Centroid Feature Extraction*

Pengujian dilakukan dengan memetakan gambar berupa potongan – potongan *cell* yang disusun sesuai dengan letak posisi piksel pada gambar. Kemudian setiap *cell* tersebut dikelompokkan dan dijumlahkan nilainya menurut perbaris dan perkolom. Dari sitolah akan diketahui nilai titik tengah dengan membagi hasilnya dengan jumlah data yang didapat.

A. Tujuan

Pengujian metode *Centroid Feature* dilakukan untuk mengetahui apakah hasil ekstraksi yang dilakukan sebelumnya oleh aplikasi *Centroid Feature* (point 4.1.5.) telah sesuai menurut teori.

B. Prosedur Pengujian

1. *Run software Matlab 7.0 (user/manual).*
2. Kemudian buat *project* baru dan tuliskan *syntax* sebagai berikut:

```
Clc //untuk membersihkan layar
I=imread(("file destination")); //membuka file
```

```
level = graythresh(I); //mengkonversi warna piksel menjadi hitam  
atau putih(255 atau 0)  
BW = im2bw(I,level); //mengkonversi piksel menjadi biner(0 atau  
1)  
imshow(BW); //menampilkan gambar
```

3. Syntax tersebut berfungsi untuk mengubah citra menjadi bit biner 0 dan 1 yang akan di tampilkan pada *array editor*, Hasil dari *Run Program* ada pada array editor lihat lampiran.
 4. Buka *file excel* “uji_centroid.xlsx
 5. Pindahkan Data citra pada *array editor* ke dalam kotak *cell* yang telah disediakan pada *file Microsoft Excel*.
 6. Arahkan alamat *cell* pada kolom CX dan CY menurut data yang tampil sesuai pada kolom oktan di atasnya.
 7. Jalankan aplikasi.
- C. Hasil pengujian

Pengujian dilakukan dengan membandingkan membandingkan hasil dari aplikasi dengan data yang yang dipetakan dari *matlab* 7.0 ke dalam *microsoft excel*. Pemetaan yang telah dilakukan oleh *software Matlab* 7.0 dapat dilihat pada gambar 4.16.



Gambar 4.16 hasil pemetaan *pixel* yang dihasilkan oleh *Matlab 7.0*

Sedangkan perbandingan antara hasil ekstraksi *Centroid* pada aplikasi dengan hasil perhitungan manual dari *ms excel* ditunjukkan pada tabel 4.1.

Tabel 4.1. Perbandingan Fitur *Centroid* dengan penghitungan manual (*MS EXCEL*)

sampel	Oktan	Aplikasi		Excel		Eror	
		X	Y	X	Y	X	Y
1	1	14,233	24,591	14,233	24,591	0	0
	2	24,3	12,1	24,3	12,1	0	0
	3	45,673	3,16	45,673	3,16	0	0

	4	54,66	20	54,66	20	0	0
	5	8,375	38,667	8,375	38,667	0	0
	6	14,923	60,479	14,923	60,479	0	0
	7	41,188	55,676	41,188	55,676	0	0
	8	51,529	40,889	51,529	40,889	0	0
2	1	20,864	26,3	20,864	26,3	0	0
	2	21,57	5,1731	21,57	5,1731	0	0
	3	40,074	9,4167	40,074	9,4167	0	0
	4	52,479	21,964	52,479	21,964	0	0
	5	19,6	39,25	19,6	39,25	0	0
	6	18,563	52,1	18,563	52,1	0	0
	7	43,174	53	43,174	53	0	0
	8	51,642	39,813	51,642	39,813	0	0
3	1	10,617	21,898	10,617	21,898	0	0
	2	20,92	6,4167	20,92	6,4167	0	0
	3	43,818	3,4167	43,818	3,4167	0	0
	4	50,196	28,167	50,196	28,167	0	0
	5	13,567	37,714	13,567	37,714	0	0
	6	26,4	47,084	26,4	47,084	0	0
	7	42,482	56,174	42,482	56,174	0	0
	8	48,325	36,5	48,325	36,5	0	0
4	1	7,5753	19,234	7,5753	19,234	0	0
	2	18,15	5,3636	18,15	5,3636	0	0
	3	43,11	4,9762	43,11	4,9762	0	0
	4	53,779	19,545	53,779	19,545	0	0
	5	7,8895	38,75	7,8895	38,75	0	0
	6	21,861	52,072	21,861	52,072	0	0
	7	40,375	52,212	40,375	52,212	0	0
	8	50,367	38,8	50,367	38,8	0	0
5	1	12,537	23,5	12,537	23,5	0	0
	2	25,639	14,482	25,639	14,482	0	0
	3	38,786	9,4167	38,786	9,4167	0	0
	4	43,05	25,75	43,05	25,75	0	0
	5	8,6136	45,25	8,6136	45,25	0	0
	6	17,769	58,75	17,769	58,75	0	0
	7	42,429	57,455	42,429	57,455	0	0
	8	51,574	38,257	51,574	38,257	0	0
6	1	12,312	24,646	12,312	24,646	0	0
	2	24,648	11,473	24,648	11,473	0	0
	3	37,001	10,396	37,001	10,396	0	0
	4	49,357	24,821	49,357	24,821	0	0

	5	8,6229	39,694	8,6229	39,694	0	0
	6	22,455	51,55	22,455	51,55	0	0
	7	44,022	49,633	44,022	49,633	0	0
	8	52,031	44,05	52,031	44,05	0	0
7	1	24,143	29,222	24,143	29,222	0	0
	2	28,773	22,786	28,773	22,786	0	0
	3	38,676	8,9545	38,676	8,9545	0	0
	4	0	0	0	0	0	0
	5	10,853	43,832	10,853	43,832	0	0
	6	18,8	46,452	18,8	46,452	0	0
	7	46,306	51,31	46,306	51,31	0	0
	8	58,438	56,091	58,438	56,091	0	0
8	1	8,8973	17,371	8,8973	17,371	0	0
	2	28,632	17,308	28,632	17,308	0	0
	3	32,222	27,75	32,222	27,75	0	0
	4	53,157	20,859	53,157	20,859	0	0
	5	8,5717	39	8,5717	39	0	0
	6	28,1	47,833	28,1	47,833	0	0
	7	34,016	45,067	34,016	45,067	0	0
	8	55,98	37,882	55,98	37,882	0	0
9	1	8,9063	20,1	8,9063	20,1	0	0
	2	18,538	11,375	18,538	11,375	0	0
	3	37,219	17,387	37,219	17,387	0	0
	4	48,1	25,98	48,1	25,98	0	0
	5	6,3303	38,111	6,3303	38,111	0	0
	6	21,589	51,119	21,589	51,119	0	0
	7	38,458	48,182	38,458	48,182	0	0
	8	56,435	35,056	56,435	35,056	0	0

4.2 Pengujian Pengenalan Sampel Training

A. Tujuan

Pengujian data sampel *training* adalah pengujian yang dilakukan untuk mengetahui apakah sampel *training* sebanyak 1000 sampel yang telah dilatihkan kepada perangkat lunak dapat dikenali.

B. Prosedur Pengujian

1. Run program Centroid Feature.exe (*user/manual*).
2. Kemudian klik tombol “Jump to MLP”.
3. Pilih “From Sample Training” untuk memilih data yang berasal dari sampel yang digunakan untuk training.
4. Tekan tombol “Process All”.

C. Hasil Pengujian

Tabel 4.2 Presentase Keberhasilan Pengenalan Pada Sampel Pelatihan

Angka	Jumlah Data (buah)	Data Yang Berhasil Dikenali (buah)	Kesalahan Pengenalan (buah)	Presentase Keberhasilan (%)
0	100	100	0	100,00 %
1	100	100	0	100,00 %
2	100	100	0	100,00 %
3	100	100	0	100,00 %
4	100	100	0	100,00 %
5	100	100	0	100,00 %
6	100	100	0	100,00 %
7	100	100	0	100,00 %
8	100	100	0	100,00 %
9	100	100	0	100,00 %
Jumlah Presentase Total				100,00%

D. Pembahasan

Untuk mengetahui presentase keberhasilan dari pengujian, pada Microsoft excel nilai target dan hasil pengenalan dibandingkan jika hasil pengenalan sama dengan target maka dianggap hasil pengenalan berhasil. Dari hasil pengujian ini jumlah data pada masing-masing sampel angka (0 – 9) sebanyak 100 *set* data, Presentase keberhasilan pengenalan pola terhadap sampel *training* sebesar 100%.

4.3 Pengujian Pengenalan Sampel *Testing*

A. Tujuan

Pengujian pengenalan sampel *testing* dilakukan untuk mengetahui apakah sampel yang didapatkan dari *participant* sebanyak 50 *set* dapat dikenali perangkat. Data tersebut yang belum pernah dilatihkan terhadap perangkat lunak.

B. Prosedur Pengujian

1. *RUN* program Centroid Feature.exe (*user/manual*).
2. Kemudian klik tombol “Jump to MLP”.
3. Pilih “From Sample Testing” untuk memilih data yang berasal dari sampel yang digunakan untuk training.
4. Tekan tombol “Process All”.

C. Hasil Pengujian

Tabel 4.4 Presentase Keberhasilan Pengenalan Pada Sampel *Test*

Angka	Jumlah Data (buah)	Data Yang Berhasil Dikenali (buah)	Data Yang Tidak Berhasil Dikenali (buah)	Presentase Keberhasilan (%)
0	50	45	5	90,00 %
1	50	41	9	82,00 %
2	50	43	7	86,00 %
3	50	43	7	86,00 %
4	50	40	10	80,00 %
5	50	35	15	70,00 %
6	50	45	5	90,00 %
7	50	35	15	70,00 %
8	50	44	6	88,00 %
9	50	33	17	66,00 %
Total	500	404	96	80,80%

D. Pembahasan

Untuk mengetahui presentase keberhasilan dari pengujian, pada Microsoft excel nilai target dengan hasil pengenalan dibandingkan, apabila hasil pengenalan sama dengan target maka dianggap hasil pengenalan berhasil. Dari hasil pengujian ini jumlah data pada masing-masing sampel angka (0 – 9) sebanyak 50 set data, dan berhasil dikenali 404 atau sebesar 80.8% saja. Terjadi kesalahan identifikasi pada pengenalan pola angka dari yang terbesar adalah “9” , “5”, “7”, “4” dan “1”, ,”2”, “3”, “8” , ”0”dan “6”. Pada Tabel 4.5 adalah analisa kesalahan identifikasi kesalahan pengenalan pola dan berapa kali dikenali sebagai karakter angka yang lain.

Tabel 4.5 Analisa Kesalahan Identifikasi Pada Tiap Karakter

Angka	Dikenali Sebagai Angka										Presentase Error
	0	1	2	3	4	5	6	7	8	9	
0	0	0	0	2	2	0	0	0	1	0	10%
1	2	0	0	2	0	1	0	2	2	0	18%
2	2	0	0	0	0	0	0	3	2	0	14%
3	2	1	2	0	0	0	0	2	0	0	14%
4	7	0	1	1	0	0	0	0	0	1	20%
5	0	0	5	1	7	0	0	1	0	1	30%
6	2	0	0	0	2	0	0	1	0	0	10%
7	2	1	0	1	1	1	0	0	8	1	30%
8	2	0	0	0	2	0	0	2	0	0	12%
9	5	1	0	2	0	0	0	8	1	0	34%

Kesalahan pengenalan pada angka “9” adalah yang paling banyak dengan presentase error sebesar 34%, dan angka “7” dan “4” yang masing - masing sebesar 30% dari 50 sampel data. Selain itu kesalahan yang sering terjadi

adalah identifikasi terhadap angka “9” yang dikenali sebagai angka “7”. Hal itu ditunjukkan dengan kesalahan yang terjadi sebanyak 8 kali. Kemungkinan yang terjadi pada kesalahan tersebut dikarenakan pola titik tengah dari citra tersebut mirip atau memang pola dari citra pada sampel yang digunakan tidak sesuai dengan pola Angka jawa yang semestinya.

Hasil analisa dari pengujian yang dilakukan terhadap 500 sampel angka jawa dapat dilihat pada gambar 4.17.

Jumlah Sample	500	Myu	0,1	
Jumlah Gagal	96	Iteration	1000001	
Keberhasilan%	81	SSE	2,00036739554103E+14	
Sampel	Total	Failed	Success	Success%
Angka 0	50	5	45	90
Angka 1	50	9	41	82
Angka 2	50	7	43	86
Angka 3	50	7	43	86
Angka 4	50	10	40	80
Angka 5	50	15	35	70
Angka 6	50	5	45	90
Angka 7	50	15	35	70
Angka 8	50	6	44	88
Angka 9	50	17	33	66
Total	500	96	404	80,8

Gambar 4.17 Analisa dari hasil pengujian

BAB V

PENUTUP

Berdasarkan hasil pengujian pada perangkat lunak yang dipergunakan dalam Pengambilan Fitur Menggunakan ekstraksi *Centroid Feature*, maka dapat diambil beberapa kesimpulan dan saran sebagai berikut.

5.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil penelitian yang telah dilakukan adalah sebagai berikut :

1. Citra Angka Jawa telah berhasil diekstraksi titik tengahnya sesuai dengan metode *Centroid Feature*.
2. Dari hasil analisa pada sampel uji tingkat prosentase keberhasilan metode *Fentroid Feature* dalam pengenalan tulisan Angka Jawa sebesar 80,8%.
3. Hasil analisa pada sampel *training* menunjukkan tingkat keberhasilan sempurna atau 100% sedangkan untuk tingkat keberhasilan pada sampel *testing* menurun sebesar 19,2%.

Selain itu kesimpulan lain yang diperoleh setelah melakukan analisa hasil dari sistem yang telah dibuat, yaitu terjadi kesalahan yang dikarenakan adanya kemiripan bentuk (pola titik tengah) pada beberapa angka Jawa. Adapun kesalahan terbesar pertama yang terjadi terletak antara angka 9 “” yang memiliki kemiripan dengan angka 7 ”” (angka Jawa). Berikutnya karena sampel yang telah dikumpulkan dari *participant* tidak memenuhi pola penulisan angka Jawa yang seharusnya.

5.2 Saran

Saran yang dapat diberikan untuk mengembangkan sistem ini sebagai berikut :

1. Sebaiknya ditambahkan dengan ekstraksi fitur yang lain agar mendapatkan hasil pengenalan karakter yang lebih optimal.
2. Perangkat lunak dapat dikembangkan untuk mengenali per-kata bahkan sebuah kalimat dari kombinasi tulisan Angka Jawa maupun tulisan Aksara Jawa yang lainnya untuk membantu peneliti tulisan jawa maupun budayawan.



DAFTAR PUSTAKA

- Basu, S., dkk., 2005, Handwritten ‘Bangla’ Alphabet Recognition using an MLP Based Classifier, Proceeding of the 2nd National Conference on Computer Processing of Bangla, hal. 285 – 291. Dhaka.
- Basu, S., dkk., 2005, An MLP Based Approach for Recognition of Handwritten ‘Bangla’ Numerals, Proceeding 2nd Indian International Conference on Artificial Intelligence, hal. 407 – 417. Pune.
- Chaudhuri, B.B. dan Bhattacharya, U., 2000, Efficient Training and Improved Performance of Multilayer Perceptron in Pattern Classification, *Neurocomputing*, vol. 34, hal. 11-27.
- Darusuprapta, dkk., 2002, *Pedoman Penulisan Aksara Jawa*, Yayasan Pustaka Nusatama, Yogyakarta.
- Das, N., dkk., 2006, Handwritten Arabic Numeral Recognition using a *Multi layerPerceptron*, Proceeding National Conference on Recent Trends in Information Systems, hal. 200 – 203.
- Das, N., dkk., 2010, Handwritten Bangla Basic and Compound Character Recognition Using MLP and SVM, Journal of Computing, Vol. 2, No. 2, hal. 109 – 115.
- Fausett, L. 2006. *Fundamentals of Neural Networks*. Prentice-Hall, New York.

Fiset, R., dkk., 1998, Map-Image Matching Using a Multi-Layerperceptron: The Case of the Road Network, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 53, hal. 76-84.

Ham, F.M. dan Kostanic, I., 2001, *Principles of Neurocomputing for Science & Engineering*. McGraw-Hill, New York.

Hasibuan, F.M., 2011, *Desain dan Implementasi Sistem Penerjemah Aksara Jawa Ke Huruf Latin Berbasis Pengolahan Citra Digital dan Jaringan Syaraf Tiruan Self-Organizing Map (SOM)*, Tugas Akhir Mahasiswa Institut Teknologi Telkom.

Kiong, L.V., 2006, *Visual Basic 6 Made Easy*, ISBN: 141962895X, 2006

Mukherjee, S., 2010, Recognition of Handwritten Bengali Character Based On Character Features, (M.Tech IT Thesis), Jadavpur University, Kolkata, India.

McCulloch, W.S. dan Pitts, W., 1943, A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, vol 5: hal. 115-133.

Nurtanio, Inggrid, Astuti, E. R., Purnama, I. K. E., Hariadi, M., Purnomo, M. H.,
Classifying Cyst and Tumor Lesion Using Support Vector Machine Based on Dental Panoramic Images Texture Features, IAENG International Journal of Computer Science, vol. 40, no. 1, Feb. 2013.

Rosenblatt, F., 1958, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, vol. 5: hal 368-408.

Rumelhart, D.E. dkk., 1986, Learning Representations by Back-Propagating Errors. *Nature*, vol. 323: hal. 533-536.

Wibowo, M.C. dan Wirakusuma, S., 2013, *Pengenalan Pola Tulisan Tangan Aksara Jawa "Ha Na Ca Ra Ka" Menggunakan Multi Layer Perceptron*, Prosiding Seminar Nasional Teknologi Informasi (SNASTI) 2013, Oktober 2013, Surabaya, hal. ICCS-27 – ICCS-32

Zhang, Z. dkk., 1998, Comparison Between Geometry-Based and Gabor-Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron.

Automatic Face and Gesture Recognition. Third IEEE International Conference. Naara

Trier O.D., Anil K.J., dan Torfinn,T. ,” Feature Extraction methods for Character Recognition –A survey “, *Pattern Recognition*, Vol29, No.4, pp-641-662, 1996

