

**PENGENALAN CITRA AKSARA JAWA PADA PLANG
JALAN MENGGUNAKAN PEMODELAN
STATISTIK SPASIAL**



N a m a : Nurdana Ahmad Fadil
NIM : 16523172

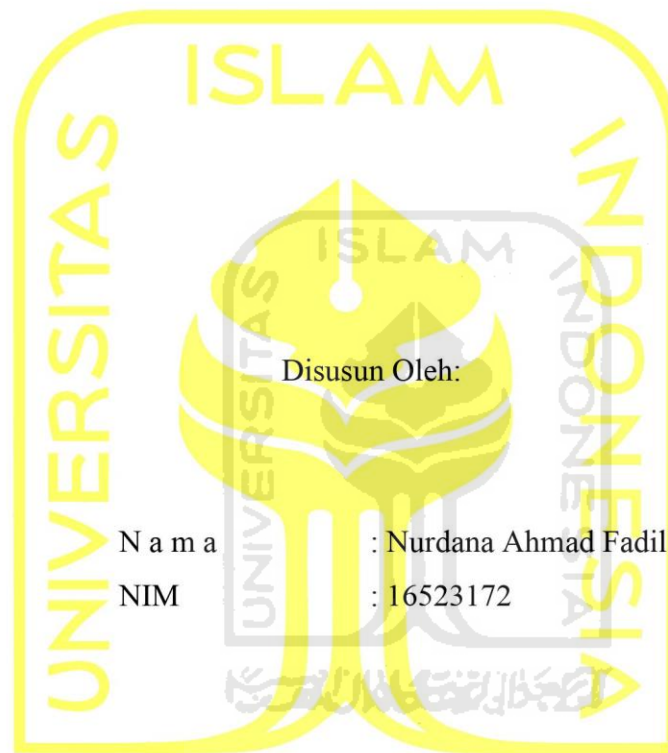
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**Pengenalan Citra Aksara Jawa Pada Plang Jalan
Menggunakan Pemodelan
Statistik Spasial**

TUGAS AKHIR



Disusun Oleh:

N a m a : Nurdana Ahmad Fadil
NIM : 16523172

الجمعة الامتية الاندونيسية
الاستد بالاندو

Yogyakarta, 3 Juli 2020

Pembimbing,

A handwritten signature in blue ink, appearing to read 'Izzati Muhimmah', is written over the name of the supervisor.

(Izzati Muhimmah, S.T., M.Sc., Ph.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**Pengenalan Citra Aksara Jawa Pada Plang
Jalan Menggunakan Pemodelan
Statistik Spasial**

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika

di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 23 Juli 2020

Tim Penguji

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Anggota 1

Septia Rani, S.T., M.Cs.

Anggota 2

Arrie Kurniawardhani, S.Si., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nurdana Ahmad Fadil
NIM : 16523172

Tugas akhir dengan judul:

Pengenalan Citra Aksara Jawa pada Plang Jalan Menggunakan Pemodelan Statistik Spasial

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 23 Juli 2020



(Nurdana Ahmad Fadil)

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin dengan mengucapkan rasa syukur kepada kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya saya akhirnya dapat menyelesaikan Tugas Akhir Program Studi Informatika Fakultas Teknologi Industri Universitas Islam Indonesia dengan baik, maka saya persembahkan maha karya ini kepada :

1. Ibu Nurul Zakiyah yang selalu saya sayangi dan cintai.
2. Ibu Nurul Zakiyah yang telah menjadi sosok Ibu dan wanita yang paling sempurna.
3. Ibu Nurul Zakiyah yang selalu memberi pelajaran dalam arti kehidupan.
4. Bapak Kardono yang telah menjadi sosok Ayah yang tak mengenal putus asa dan telah mendidik saya dengan sepenuh hati, tentu saja yang saya sayangi dan cintai.
5. Saudari Annisa syifaulhaq yang telah menjadi adik perempuan pertama saya dan proses menjadi wanita sempurna.
6. Saudari Kasifah Albaroroh yang telah menjadi adik perempuan kedua saya dan masih mencari jati diri sebagai wanita sempurna.
7. Trah Simbah Waram dan Trah Simbah Djaelani yang saya sangat kagumi kepada mereka semua.
8. Ibu Izzati Muhimmah yang telah menjadi dosen pembimbing pada tugas akhir ini dengan kesabaran dan sosok keibuan beliau tugas akhir ini bisa berakhir.
9. Keluarga besar Informatika UII sebagai wadah saya berkembang selama kurang lebih 4 Tahun ini.

HALAMAN MOTO

“Berjalan tak sesuai rencana adalah jalan yang sudah biasa,
jalan satu-satunya adalah jalani sebaik yang kau bisa”

(Gas - FSTVLST)

“Kau dan aku melintas cerita, semoga tak berubah
akan bersama selamanya, jangan berubah”

(Opus - FSTVLST)

“Coba letakkan diri dan serahkan hati ini
bahwa mereka yang tercinta pasti pergi”

(Kamis - FSTVLST)

“Idealisme adalah kemewahan terakhir yang hanya dimiliki pemuda”

(Tan Malaka)

“Menjadi terdepan tanpa matang adalah
sebuah keterpaksaan dalam mengambil keputusan”

(Hamba Allah)

“Keselarasan, keharmonisan, dan kebersamaan adalah
suatu landasan dalam berbagai hal”

(Hamba Allah)

“Imagination is more important than knowledge”

(Albert Einstein)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakaatuh

Puja dan puji syukur saya ucapkan kepada Allah SWT yang telah memberikan rahmat serta hidayahnya sehingga saya dapat menyelesaikan tugas akhir ini yang berjudul **“Pengenalan Citra Aksara Jawa pada Plang Jalan Menggunakan Pemodelan Statistik Spasial”**. Tugas akhir ini saya susun guna memenuhi syarat untuk memperoleh gelar sarjana di Program Studi Informatika Universitas Islam Indonesia dan juga untuk kedepannya agar budaya yang telah diwarisi nenek moyang akan senantiasa dilestarikan dan mempermudah setiap orang untuk membaca aksara Jawa dalam bentuk digital.

Dalam penyusunan tugas akhir ini, saya sepenuhnya menyadari bahwa tugas akhir ini tidak dapat terselesaikan tanpa adanya bimbingan, dorongan dan bantuan dari berbagai pihak yang bersifat langsung maupun tidak langsung. Dalam kesempatan ini izinkan saya mengucapkan terima kasih kepada:

1. Allah SWT, berkat limpahan rahmat, karunia dan hidayah-Nya sehingga saya mampu menyusun tugas akhir ini dengan lancar.
2. Ibu yang selalu sabar dengan kelakuan nakal anak pertamanya.
3. Ibu yang teramat sangat menjadi motivasi untuk menyelesaikan tugas akhir ini.
4. Ibu yang sangat saya cintai dan banggakan.
5. Ayah yang dengan kegigihannya telah mendidik dan membesarkan anak pertamanya dengan penuh tanggung jawab dan kedisiplinan.
6. Adik pertama yang akan melakukan studi di perguruan tinggi di tahun kelulusan saya ini.
7. Adik kedua yang teramat sangat saya sayangi yang masih menginjak pendidikan sekolah dasar.
8. Keluarga besar simbah Waram dan simbah Djaelani yang saya banggakan.
9. Ibu Izzati Muhimmah , S.T., M.Sc. sebagai dosen pembimbing tugas akhir saya yang telah sabar membantu dalam penyusunan tugas akhir ini.
10. Keluarga besar Informatika Universitas Islam Indonesia yang telah turut menyumbangkan pengalaman yang sangat berharga bagi saya.
11. Keluarga besar Himpunan Mahasiswa Teknik Informatika yang telah memberikan banyak pengalaman.
12. Keluarga besar Teater Djemuran yang sangat saya kagumi dan banggakan.
13. Keluarga besar Vendetta 523 yang semoga lebih kompak dan lebih *gayeng* lagi.

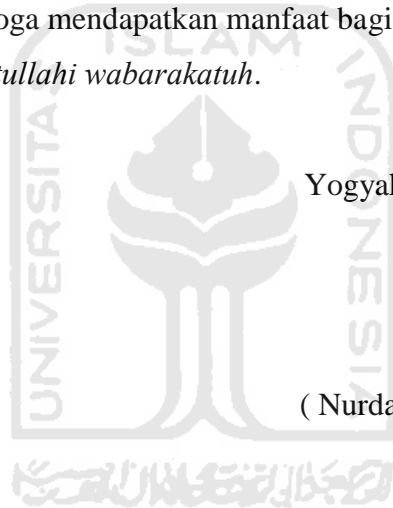
14. Keluarga besar HexaDecima yang saya sayangi dan saya banggakan karena telah menjadi keluarga di lingkungan perantauan saya.
15. Grup 'KOBE' yang penuh dengan kehidupan masa muda.
16. Grup 'U**ty Hunter' yang telah menemani saya dalam masa pandemi.
17. Grup 'Keluarga Spotify' yang telah menjadi tempat hangat untuk bercerita *ngalor ngidul*.
18. Serta teman-teman semua yang tidak bisa saya ucapkan satu-satu persatu terima kasih atas dukungannya selama ini.

Saya menyadari bahwa dalam penyusunan tugas akhir ini masih banyak kekurangan dan kesalahan yang disengaja maupun tidak disengaja maka, izinkan saya mohon maaf sebesar-besarnya atas tindakan tersebut. Semoga dalam kekurangan ataupun kesalahan bisa menjadikan inspirasi bagi generasi-generasi berikutnya, sekian kata pengantar ini dibuat. Bagi siapapun yang membaca tugas akhir ini semoga mendapatkan manfaat bagi semua pihak.

Wassalamu 'alaikum warahmatullahi wabarakatuh.

Yogyakarta, 3 Juli 2020

(Nurdana Ahmad Fadil)



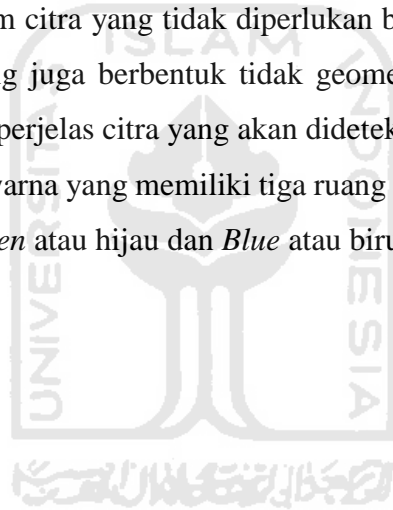
SARI

Aksara Jawa yang mengalami penurunan minat oleh sebagian anak muda menjadi fokus pada penelitian pengenalan citra aksara Jawa. Hasil penelitian terdahulu mengenai segmentasi huruf non-latin selain aksara Jawa memanglah sudah menunjukkan tingkat akurasi yang tinggi namun pada terkhusus segmentasi aksara Jawa masih ada beberapa kekurangan yaitu karakter aksara Jawa yang akan dideteksi belum berupa citra. Pada penelitian terdahulu lainnya objek aksara Jawa yang dideteksi sudah berupa citra namun hanya bisa satu jenis huruf. Penelitian ini menggunakan citra yang ditangkap oleh kamera sehingga gaya penulisan huruf dari aksara Jawa bisa berbeda-beda bukan dari input manual sistem yang menjadikan gaya penulisan aksara Jawa sama seperti yang dilakukan penelitian sebelumnya. Metode yang dilakukanpun terbilang umum digunakan seperti segmentasi, morfologi dan *thining*, namun perlu digarisbawahi di sini adalah dalam penyusunan metode tersebut ambang batas yang menjadi tolak ukur dalam metode tersebut sangat diperhatikan sedemikian rupa sehingga mendapatkan objek yang ingin dideteksi. Setelah objek telah ditemukan otomatis oleh sistem maka akan dilakukan ekstraksi ciri pada masing-masing objek dengan pemodelan Statistik Spasial yaitu menghitung jumlah piksel dengan intensitas bernilai 1 (berwarna putih) tiap karakter yang telah dipotong sembilan bagian sama rata. Setelah ekstraksi ciri didapatkan, pencocokan dengan data latih dilakukan menggunakan Jaringan Saraf Tiruan, kemudian diperoleh tingkat hasil akhir dengan rata-rata akurasi sebesar 88,7%.

Kata kunci: aksara, citra, jawa, objek, pengenalan.

GLOSARIUM

<i>Training</i>	Bagian dataset yang kita latih untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma pembelajaran mesin. Memberikan petunjuk melalui algoritma agar mesin yang kita latih bisa mencari korelasinya sendiri atau belajar pola dari data yang diberikan.
<i>Flowchart</i>	Suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail atau biasa disebut diagram alir.
<i>Grayscale</i>	Skala warna monokromatik yang terdiri dari gabungan dua warna yaitu hitam dan putih. Oleh karena itu, gambar grayscale hanya memiliki warna abu-abu.
<i>Noise</i>	Objek dalam citra yang tidak diperlukan biasanya berbentuk titik-titik kecil kadang juga berbentuk tidak geometris yang akan dihilangkan untuk memperjelas citra yang akan dideteksi.
RGB	Intensitas warna yang memiliki tiga ruang warna, diantaranya <i>Red</i> atau merah, <i>Green</i> atau hijau dan <i>Blue</i> atau biru.



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Metode Penelitian	2
1.7 Usulan Penyelesaian	3
BAB II LANDASAN TEORI	4
2.1 Aksara Jawa	4
2.1.1 Huruf	4
2.1.2 Pasangan Huruf	5
2.1.3 Aksara Murda	5
2.1.4 Aksara Swara.....	6
2.1.5 <i>Sandhangan</i>	6
2.1.6 Aksara Rekan	7
2.1.7 <i>Pratandha</i>	7
2.2 Citra.....	8
2.2.1 Citra RGB	8
2.2.2 Citra <i>Grayscale</i>	8
2.2.3 Citra Biner	9
2.3 Pengolahan Citra	10
2.4 Segmentasi	10
2.5 Morfologi	11
2.6 Thining	12
2.7 Ekstrasi Ciri.....	12
2.8 Statistik Spasial	13
2.9 Penelitian Terdahulu	13
BAB III METODOLOGI PENELITIAN	14
3.1 Pengumpulan Data	14
3.1.1 Data Uji	14
3.1.2 Data Latih	15
3.2 Perancangan Sistem	17
3.2.1 <i>Flowchart</i> Perancangan Sistem	17
3.2.2 <i>Flowchart</i> Sistem	18

3.2.3	Desain Antarmuka	23
3.3	Implementasi	23
3.4	Pengujian.....	23
3.4.1	<i>Black Box Testing</i>	24
3.4.2	Perhitungan Statistik Ketepatan	24
3.4.3	<i>Cohen's Kappa</i>	24
BAB IV	HASIL DAN PEMBAHASAN.....	25
4.1	Implementasi Sistem	25
4.1.1	Input Citra.....	25
4.1.2	Citra Skala Keabuan	26
4.1.3	Citra Skala Biner	26
4.1.4	Menghilangkan <i>Noise</i>	27
4.1.5	Thining	27
4.1.6	Pelabelan Perkarakter	28
4.1.7	Cropping Objek	28
4.1.8	Pemodelan Statistik Spasial dan Pemotongan Menjadi 9 Bagian	29
4.1.9	Penyimpanan Piksel dengan Intensitas Bernilai 1	31
4.1.10	Pembuatan Data Latih	32
4.1.11	Pembuatan Target Latih	33
4.1.12	Pembentukan Jaringan Saraf Tiruan.....	34
4.1.13	Pendeteksian Aksara Jawa.....	37
4.2	Implementasi Antarmuka	39
4.2.1	Input Citra.....	40
4.2.2	Deteksi Citra.....	41
4.2.3	Segmentasi Citra.....	42
4.2.4	Pengenalan Citra.....	44
4.2.5	Reset	45
4.3	Pengujian.....	46
4.3.1	<i>Black Box Testing</i>	46
4.3.2	Perhitungan Statistik Ketepatan	46
4.3.3	<i>Cohen's Kappa</i>	46
4.4	Kelebihan Dan Kekurangan Sistem	47
4.4.1	Kelebihan Sistem.....	47
4.4.2	Kekurangan Sistem.....	47
BAB V	KESIMPULAN DAN SARAN	47
5.1	Kesimpulan	48
5.2	Saran.....	48
	DAFTAR PUSTAKA	50
	LAMPIRAN	51

DAFTAR TABEL

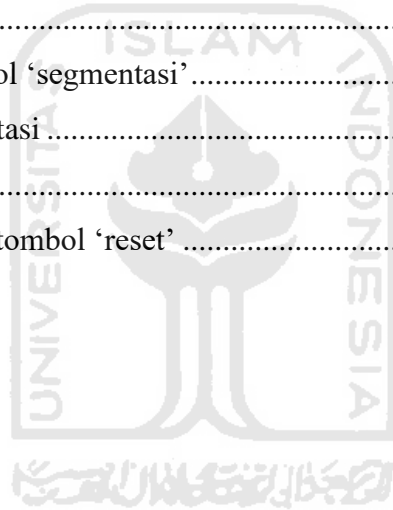
Tabel 3. 1 Total data pada tiap kelas data latih.....	16
Tabel 4. 1 Tabel penjumlahan intensitas piksel	32
Tabel 4. 2 Pembuatan data latih.....	33
Tabel 4. 3 Target.....	34



DAFTAR GAMBAR

Gambar 2. 1 Huruf aksara Jawa.....	4
Gambar 2. 2 Pasangan huruf.....	5
Gambar 2. 3 Aksara murda.....	5
Gambar 2. 4 Aksara swara.....	6
Gambar 2. 5 <i>Sandhangan</i>	6
Gambar 2. 6 Aksara rekan.....	7
Gambar 2. 7 <i>Pratandha</i>	7
Gambar 2. 8 Citra RGB.....	8
Gambar 2. 9 Citra <i>grayscale</i>	9
Gambar 2. 10 Citra biner.....	9
Gambar 2. 11 Hasil segmentasi <i>threshold</i>	11
Gambar 3. 1 <i>Flowchart</i> pengumpulan data uji.....	14
Gambar 3. 2 Contoh data uji berdasarkan pengelompokan gaya huruf.....	15
Gambar 3. 3 Contoh data latih; (a) potret. (b) Internet. (c) Menulis secara digital.....	16
Gambar 3. 4 <i>Flowchart</i> perancangan sistem.....	18
Gambar 3. 5 <i>Flowchart</i> sistem pembentukan data latih.....	19
Gambar 3. 6 <i>Flowchart</i> sistem pengujian.....	21
Gambar 3. 7 Desain antarmuka.....	23
Gambar 4. 1 Kode program input citra.....	25
Gambar 4. 2 Hasil input data citra RGB.....	25
Gambar 4. 3 Kode program skala keabuan.....	26
Gambar 4. 4 Citra skala keabuan.....	26
Gambar 4. 5 Kode program skala biner.....	26
Gambar 4. 6 Hasil citra skala biner.....	27
Gambar 4. 7 Kode program menghilangkan <i>noise</i>	27
Gambar 4. 8 <i>Noise</i> yang sudah dihilangkan pada citra.....	27
Gambar 4. 9 Kode program <i>thining</i>	28
Gambar 4. 10 Hasil citra proses <i>thining</i>	28
Gambar 4. 11 Kode program pelabelan perkarakter.....	28
Gambar 4. 12 Hasil proses pelabelan.....	28
Gambar 4. 13 Kode program <i>cropping</i>	29
Gambar 4. 14 Ilustrasi hasil <i>cropping</i>	29

Gambar 4. 15 Kode program pemotongan 9 bagian	30
Gambar 4. 16 (a) Pemotongan huruf 'Da' (b) Jumlah intensitas piksel tiap potongan	31
Gambar 4. 17 Kode program penyimpanan piksel dengan intensitas bernilai 1	32
Gambar 4. 18 <i>flowchart</i> pembuatan data latih	33
Gambar 4. 19 Hasil proses <i>training</i> jaringan saraf tiruan	35
Gambar 4. 20 Hasil performa.....	36
Gambar 4. 21 Hasil proses <i>training</i> regresi	36
Gambar 4. 22 Kode program jaringan saraf tiruan	37
Gambar 4. 23 Hasil pengenalan aksara Jawa.....	39
Gambar 4. 24.....	40
Gambar 4. 25 (a) mencari input citra (b) Menampilkan antarmuka input	41
Gambar 4. 26 Kode program tombol 'deteksi'	42
Gambar 4. 27 Hasil proses deteksi.....	42
Gambar 4. 28 Kode program tombol 'segmentasi'	44
Gambar 4. 29 Hasil proses segmentasi	44
Gambar 4. 30 Hasil proses kenali	45
Gambar 4. 31 Kode program pada tombol 'reset'	46



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang ada pada saat ini mengakibatkan semuanya akan ketergantungan dengan teknologi, tanpa dipungkiri seiring perkembangan zaman, muncul stigma “Lambat laun budaya akan ditinggalkan dan digantikan oleh teknologi”, dengan dasar tersebut, penelitian ini ingin mengubah stigma atau cara berpikir kebanyakan orang, yaitu dengan teknologi, budaya yang telah beratus-ratus tahun diwariskan dari nenek moyang juga bisa kita lestarikan. Penelitian ini akan membuktikan bahwa teknologi bukan penghalang untuk berbudaya dan diharapkan penelitian ini akan memotivasi banyak orang agar tetap bisa melestarikan budaya.

Aksara Jawa adalah salah satu aksara tradisional Nusantara yang digunakan untuk menulis bahasa Jawa. Dalam keseharian penggunaan aksara Jawa mulai digantikan dengan huruf Latin, yang mulai pertama kali dikenalkan Belanda. Seiring berjalannya waktu huruf Latin menjadi huruf nasional atau yang dikenal sebagai huruf bahasa Indonesia. Walaupun penggunaan aksara Jawa mulai menurun drastis akan tetapi di beberapa sekolah, aksara Jawa masih menjadi mata pelajaran muatan lokal yang bertujuan mengenalkan betapa pentingnya melestarikan budaya, namun seiring berkembangnya teknologi mata pelajaran muatan lokal yang mempelajari aksara Jawa mulai ditinggalkan dan dialihkan ke pelajaran yang berbau teknologi. Dalam kasus ini aksara Jawa sudah mulai hilang karena hilangnya penerus di generasi muda kita. Apabila budaya yang telah diwariskan nenek moyang kita tersebut dapat dikonversikan dalam format digital, maka akan banyak sekali manfaat yang bisa diperoleh. Selain manfaat tersebut, penelitian ini bertujuan untuk membantu orang awam dengan aksara Jawa dan menjadikannya bisa membaca dengan mudah (Widiarti, 2006). Aksara Jawa dalam penelitian ini berfokus pada plang jalan yang biasanya terdapat di sekitar jalan Yogyakarta.

Aksara Jawa sendiri memiliki huruf yang sangat unik yang masuk dalam kategori huruf non-latin, seperti halnya huruf pada bahasa Jepang, Cina, dan Thailand. Haruechaiyasak beserta temannya melakukan penelitian menggunakan teknik segmentasi pada bahasa Thailand dengan berbagai metode dengan capaian tingkat akurasi terbaik 95,79% (Haruechaiyasak, Kongyoung, & Dailey, 2008). Segmentasi pada bahasa Dzongkha dilakukan oleh Norbu beserta kawannya dengan akurasi terbaik 91,5% (Norbu, Choeje, Dendup, Hussain, & Mauz, 2010). Pada kasus aksara Jawa sendiri telah dilakukan oleh Prabantoro yang menggunakan kamus dan karakter statistik seperti jumlah karakter pada kata terpanjang, jumlah karakter pada kata terpendek, dan rata-rata panjang kata (Prabantoro, 2014). Pada penelitian Prabantoro memiliki kelemahan yaitu karakter aksara Jawa yang haruslah karakter yang diinputkan secara manual menggunakan sistem. Gaya penulisan huruf pada kasus penelitian Prabantoro ini menjadikan semua karakter aksara Jawa sama persis padahal, pada umumnya karakter aksara Jawa memiliki jenis gaya penulisan yang sangat beragam. Penelitian pengenalan aksara Jawa ini difokuskan untuk mendeteksi citra aksara Jawa, selain itu memang dimaksudkan untuk mengatasi kekurangan yang ada pada penelitian sebelumnya (Prabantoro, 2014).

Penelitian pengenalan citra aksara Jawa ini menjadikan metode-metode sederhana sebagai acuan pendeteksian awal sebuah objek aksara Jawa. Hasil akhirnya akan terlihat berapa tingkat akurasi sistem dalam mengenali objek citra aksara Jawa. Dalam proses penelitian ini selain bertujuan mencari hasil akhir juga bertujuan membantu orang awam dengan aksara Jawa dan menjadikannya bisa membaca dengan mudah.

1.2 Rumusan Masalah

Adapun perumusan masalah dari permasalahan yang sudah dipaparkan sebelumnya adalah sebagai berikut:

- a. Bagaimana membangun sistem pengenalan citra aksara Jawa menggunakan pemodelan Statistik Spasial?
- b. Bagaimana sistem yang disusun dapat mengenali tiap karakter dalam aksara Jawa?
- c. Seberapa besar tingkat akurasi sistem pengenalan citra aksara Jawa menggunakan pemodelan Statistik Spasial?

1.3 Batasan Masalah

Dalam tugas akhir ini, masalah dibatasi pada hal-hal berikut:

- a. Citra yang akan dikenali harus diambil tegak lurus dengan kamera.
- b. Lebar objek huruf pertama harus berukuran 39-41 piksel.
- c. Panjang objek huruf pertama harus berukuran 40-60 piksel.
- d. Gaya jenis huruf aksara Jawa mempunyai karakteristik khusus yang bervariasi.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk pengenalan citra aksara Jawa dan menerjemahkan ke huruf latin serta mengetahui akurasi pengenalan karakter aksara Jawa.

1.5 Manfaat Penelitian

Manfaat pada penelitian ini adalah :

- a. Memudahkan pengguna dalam menerjemahkan aksara Jawa.
- b. Bisa digunakan untuk sarana pembelajaran bagi kaum milenial.
- c. Memberikan informasi cara baca pada masing-masing karakter aksara Jawa.

1.6 Metode Penelitian

Proses yang dilakukan dalam pengembangan sistem di penelitian ini adalah sebagai berikut:

a. Pengumpulan Data

i. Observasi

Metode observasi adalah metode untuk mengumpulkan semua jenis data, baik itu data latih maupun data uji yang sesuai dengan kriteria.

ii. Studi Pustaka

Studi pustaka dilakukan dengan tujuan mengumpulkan data-data dari penelitian sebelumnya yang akan menjadi referensi untuk penelitian ini.

b. Pengembangan Sistem

i. Analisis kebutuhan sistem

Pada proses analisis kebutuhan sistem dilakukan untuk menguraikan tiap masalah yang ada dan menjabarkannya kemudian mulai menentukan langkah-langkah dalam pembuatan sistem, mulai dari proses input, proses dan output.

ii. Perancangan

Perancangan pada penelitian kali ini lebih menitik beratkan pada langkah-langkah atau tata cara pembuatan sistem yang dijelaskan dengan *flowchart*.

iii. Implementasi

Tahap implementasi menggunakan aplikasi MATLAB R16b dan beberapa fungsi bawaan dan *tools* bawaan pada aplikasi tersebut.

iv. Pengujian

Tahap pengujian dilakukan untuk menguji apakah sistem ini bekerja dengan baik atau tidak. Pada penelitian ini menggunakan tiga jenis pengujian berbeda, yaitu *Black Box Testing*, perhitungan statistik ketepatan dan *Cohen's Kappa*.

1.7 Usulan Penyelesaian

Dari permasalahan yang ada, penulis membuat langkah penyelesaian dengan cara membangun sebuah program yang dapat melakukan proses perhitungan dan deteksi citra aksara Jawa secara otomatis yang dapat menerjemahkan karakter aksara Jawa yang telah terdeteksi dengan cara membangun sebuah program yang memanfaatkan metode jaringan saraf tiruan.

BAB II LANDASAN TEORI

2.1 Aksara Jawa

Bahasa Jawa memiliki jutaan penutur, menjadikannya salah satu bahasa yang paling banyak digunakan di Indonesia dan di dunia pada umumnya (Zhangrila, 2018). Banyak aplikasi dikembangkan untuk meningkatkan kesadaran dan mendidik masyarakat dalam aksara Jawa, beberapa di antaranya menggunakan pengenalan gambar untuk menerjemahkan aksara Jawa bagi pengguna. Aksara Jawa di sini memiliki kompleksitas dalam gaya penulisan yang menjadikan basis data atau data *training* di sini lebih beraneka ragam dan sangat bervariasi. Dalam aksara Jawa terbagi ke dalam berbagai jenis yang terbagi antara huruf, pasangan huruf, aksara murda, aksara swara, *sandhangan*, aksara rekan dan *pratandha*.

2.1.1 Huruf

Dalam huruf aksara Jawa terbagi dalam 20 huruf berbeda yang menjadikannya pengucapan atau dalam bahasa Indonesia disebut suku kata. Huruf aksara Jawa bisa dilihat pada Gambar 2.1.

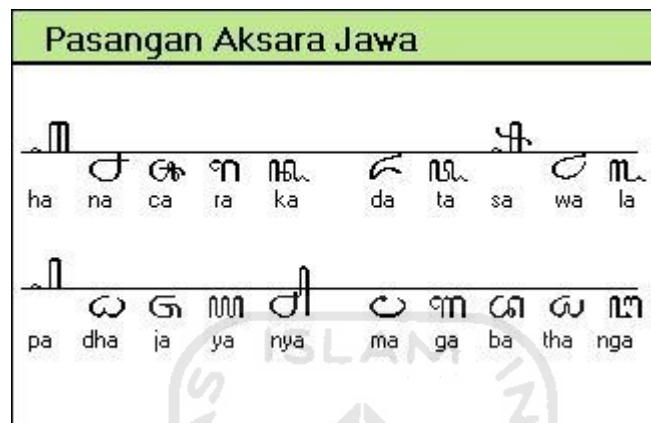


Gambar 2. 1 Huruf aksara Jawa

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.1.2 Pasangan Huruf

Pelafalan pasangan huruf aksara Jawa sama saja dengan pelafalan huruf namun huruf yang diberi pasangan akan mematikan huruf vokal. Contoh penggunaan, ketika kita ingin menuliskan kata Sandal maka huruf yang diperlukan yaitu 'sa' 'na' 'pasangan da' 'la' dan 'sandangan pangkon'. Contoh pasangan huruf aksara Jawa dapat dilihat pada Gambar 2.2.

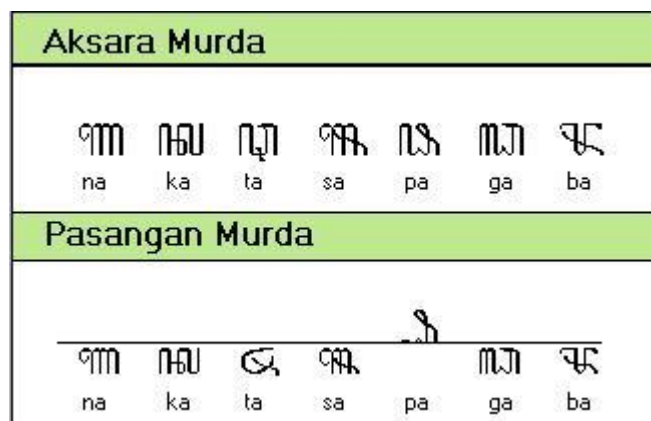


Gambar 2. 2 Pasangan huruf

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.1.3 Aksara Murda

Aksara murda digunakan untuk menuliskan awal kalimat dan kata harus menunjukkan nama gelar, kota lembaga atau nama dalam huruf besar. Contoh aksara murda dapat dilihat pada Gambar 2.3.

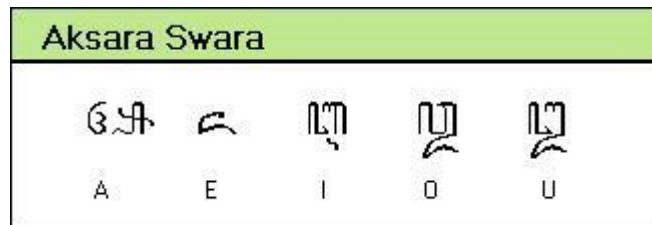


Gambar 2. 3 Aksara murda

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.1.4 Aksara Swara

Aksara swara atau dalam bahasa Indonesia disebut huruf vokal. Penggunaan aksara swara untuk nama dengan awalan vokal yang mengharuskan huruf vokal. Contoh aksara swara dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Aksara swara

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.1.5 Sandhangan

Sandhangan digunakan untuk vokal yang berada di tengah kata, dibedakan berdasarkan cara membaca. Contoh penggunaan *sandhangan* dapat terlihat pada Gambar 2.5.











Nama Sandhangan	Aksara Jawa	Keterangan
Wulu	ꦲꦸꦲ	tanda vokal i
Suku	ꦲꦸꦺ	tanda vokal u
Taling	ꦲꦸꦺꦴ	tanda vokal é
Pepet	ꦲꦸꦺꦴꦺ	tanda vokal e
Taling tarung	ꦲꦸꦺꦴꦺꦴ	tanda vokal o
Layar	ꦲꦸꦺꦴꦺꦴꦺꦴ	tanda ganti konsonan r
Wignyan	ꦲꦸꦺꦴꦺꦴꦺꦴꦺꦴ	tanda ganti konsonan h
Cecak	ꦲꦸꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴ	tanda ganti konsonan ng
Pangkon/pangku	ꦲꦸꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴ	tanda penghilang vokal
Péngkal	ꦲꦸꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴ	tanda ganti konsonan ya
Cakra	ꦲꦸꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴ	tanda ganti konsonan ra
Keret	ꦲꦸꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴꦺꦴ	tanda ganti konsonan re

Gambar 2. 5 Sandhangan

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.1.6 Aksara Rekan

Aksara rekan atau biasa disebut huruf tambahan adalah huruf yang digunakan untuk serapan bahasa asing. Contoh aksara rekan dapat dilihat pada Gambar 2.6.













Aksara Rekan				
				
kha	dza	fa/va	za	gha
Pasangan Rekan				
				
kha	dza	fa/va	za	gha

Gambar 2. 6 Aksara rekan

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.1.7 Pratandha

Penulisan dalam *pratandha* seperti tanda baca dalam bahasa Indonesia. Penggunaan *pratandha* dapat dilihat pada Gambar 2.7.

Tanda Baca	Aksara Jawa	Keterangan
Adeg-adeg		tanda awal kalimat
Pada Lungsi		tanda titik
Pada Lingsa		tanda koma
pada pangkat		penanda angka
pada guru		Awalan surat/cerita
pada pancak		Akhir surat/cerita
Pada Luhur		Awal Surat untuk derajat lebih tinggi
Pada Madya		Awal Surat untuk derajat sebaya
Pada Andhap		Awal Surat untuk derajat lebih rendah
Purwa Pada		Awalan Tembang
Madya Pada		Tengah-tengah tembang (bait)
Wasana Pada		Akhir Tembang

Gambar 2. 7 Pratandha

Sumber: <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>

2.2 Citra

Citra adalah sekumpulan informasi yang diciptakan dalam bentuk dua dimensi yang merepresentasikan suatu objek di mana isi citra merupakan susunan array dari bilangan real atau bilangan yang sangat kompleks dan diwakilkan dalam bilangan-bilangan bit terhingga. Menurut Munir, citra adalah gambar pada bidang dwimarta (dua dimensi) (Munir, 2004). Citra sendiri mempunyai dua jenis yaitu, citra kontinu dan citra diskrit. Agar sebuah citra dapat diolah dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra dan citra yang dihasilkan disebut citra digital. Secara umum, citra digital mempunyai bentuk empat persegi panjang dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (Munir, 2004). Ada tiga jenis citra yang biasanya digunakan dalam pengolahan citra yaitu citra RGB, citra *grayscale* dan citra biner.

2.2.1 Citra RGB

Citra RGB adalah citra warna yang komponen penyusunan warnanya ada tiga jenis yaitu *Red* (merah), *Green* (hijau) dan *Blue* (biru). Citra ini yang biasa kita lihat dalam citra warna digital. Contoh citra RGB bisa dilihat pada Gambar 2.8.



Gambar 2. 8 Citra RGB

Sumber: <https://putuadisusanta.wordpress.com/2015/07/17/representasi-citra-digital/>

2.2.2 Citra *Grayscale*

Citra Grayscale adalah citra yang intensitas merupakan citra yang merepresentasikan setiap piksel yang hanya satu jenis atau satu nilai warna dalam skala keabuan. Mendapatkan nilai intensitas keabuan bisa dengan cara menjumlahkan 30% dari nilai R (Red dari singkatan

RGB), 59% dari G (Green dari singkatan RGB), dan 11% dari B (Blue dari singkatan RGB). Hasil dari penjumlahan tersebut, kemudian akan direpresentasikan dalam setiap *channel Red*, *Green* dan *Blue* dengan nilai intensitas yang sama. Contoh citra *grayscale* dapat dilihat pada gambar 2.9.



Gambar 2. 9 Citra *grayscale*

Sumber: <https://putuadisusanta.wordpress.com/2015/07/17/representasi-citra-digital/>

2.2.3 Citra Biner

Citra biner adalah citra yang hanya memiliki dua nilai intensitas yaitu 1 (putih) dan 0 (hitam). Contoh citra biner dapat dilihat pada Gambar 2.10.



Gambar 2. 10 Citra biner

Sumber: <https://putuadisusanta.wordpress.com/2015/07/17/representasi-citra-digital/>

2.3 Pengolahan Citra

Pengolahan citra di sini bertujuan untuk mengambil informasi yang dirasa penting atau yang bisa mewakili dari sebuah objek yang ingin dikenali. Hasil dari pengambilan informasi ini akan sangat bermanfaat ketika nilai yang merepresentasikan objek tersebut akurat dan nantinya hasil akurasi dalam sebuah pencocokan dalam data uji pun akan semakin tinggi.

Pada pengolahan citra terdapat berbagai macam operasi pengolahan citra yang dilakukan guna menjalankan prosedur tertentu. Operasi titik adalah operasi yang dilakukan terhadap setiap piksel yang berada pada citra yang hasil keluarannya hanya ditentukan oleh nilai piksel itu sendiri. Operasi area adalah operasi yang dilakukan terhadap setiap piksel pada sebuah citra yang hasil keluarannya dipengaruhi piksel itu sendiri dan piksel lainnya dalam suatu daerah tertentu. Operasi global adalah operasi yang dilakukan terhadap setiap piksel pada setiap piksel pada sebuah citra yang keluarannya ditentukan secara keseluruhan dari piksel yang berada pada piksel tersebut.

2.4 Segmentasi

Segmentasi adalah langkah pertama untuk memproses bahasa non-latin seperti aksara Jawa. Dalam penelitian (Tanaya & Adriani, 2016) tersebut segmentasi yang dilakukan berdasarkan pendekatan huruf per huruf. Pada huruf pertama, akan dicocokkan dengan semua huruf aksara Jawa utama yang mungkin tersegmentasi menggunakan basis data yang telah dibuat. Setelah melakukan pendeteksian huruf per huruf nantinya akan membagi huruf per huruf menjadi beberapa bagian untuk dideteksi citra di setiap bagian. Dalam proses segmentasi terdapat berbagai macam metode segmentasi. Pertama *thresholding*, *Active Countour*, Segmentasi warna, deteksi tepi dan *transformasi hough*. Dalam penelitian ini hanya menggunakan metode *thresholding*.

Thresholding yang dimaksud di sini merupakan suatu nilai batasan dalam suatu metode yang mengubah citra *grayscale* ke dalam citra biner. *Thresholding* biasa juga disebut *gray level segmentation* atau pemisahan warna dalam skala keabuan. Nilai batasan dalam *thresholding* dapat kita sesuaikan keinginan kita, dalam kasus ini berarti nilai batasan dalam skala keabuan yang melebihi keinginan kita akan berubah menjadi citra hitam (bernilai 1) dan yang tidak memenuhi batasan yang kita inginkan akan menjadi citra putih (bernilai 0) atau juga bisa sebaliknya seperti yang terlihat pada Persamaan 2.1 dan hasil proses segmentasi dapat dilihat pada gambar 2.11.



Gambar 2. 11 Hasil segmentasi *threshold*

Sumber: <https://pemrogramanmatlab.com/pengolahan-citra-digital/segmentasi-citra/thresholding-menggunakan-matlab/>

$$F(x,y) = 255, \text{ jika } f(x,y) \geq T$$

$$F(x,y) = 0, \text{ jika } f(x,y) < T$$

Nilai threshold T adalah

$$\begin{aligned} T &= \frac{f_{\max} + f_{\min}}{2} \\ &= \frac{240 + 50}{2} = 145 \end{aligned} \quad (2.1)$$

2.5 Morfologi

Operasi morfologi dalam kasus ini yaitu suatu proses atau metode yang bertujuan untuk mengubah bentuk objek pada citra asli dan biasanya mengubah menjadi citra *grayscale* ataupun citra biner. Haralick mengenalkan metode dalam proses pengurangan *noise* pada citra biner agar objek dari citra biner tersebut dengan mudah memilah objek yang menjadi pusat perhatian dan menghilangkan objek tidak penting (Haralick, 1987). Pada kasus ini menggunakan metode *filling* dengan rumus.

$$A \circ B = (A \ominus B) \oplus B$$

$$\text{Menjadi} \quad A \circ B = \bigcup_{B_x \in A} B_x \quad (2.2)$$

2.6 Thining

Thining adalah sebuah metode atau algoritma yang bertujuan untuk memipihkan suatu objek atau menjadikannya *single piksel*. Proses *thining* ini pada dasarnya mencari sebuah piksel yang dapat mewakili secara garis besar di objek tersebut sehingga memunculkan informasi yang penting dan membuang informasi yang kurang penting. Arcelli dan Sanniti pernah memberikan contoh metode *fast thinning algorithm* untuk mengurangi komponen yang dirasa kurang membawa informasi dari objek tersebut (Arcelli & Di Baja, 1985). Algoritma ini kemudian mejadi populer dan kemudian banyak metode yang kita pakai sekarang adalah hasil metode yang dikembangkan dari metode tersebut.

$$X_H(p)=1$$

$$Di\ mana\ X_H(p) = \sum_{i=1}^4 b_i \quad (2.3)$$

$$b_i = \begin{cases} 1, & \text{if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0, & \text{otherwise} \end{cases}$$

x_1, x_2, \dots, x_8 adalah nilai-nilai dari sekitar p , dimulai dengan daerah sekitar timur dan diberi nomor dalam urutan berlawanan arah jarum jam.

2.7 Ekstrasi Ciri

Ekstraksi ciri adalah sebuah metode pengambilan informasi dari hasil akhir semua metode, yang nantinya akan disimpan dan akan menjadi beberapa tabel yang tiap tabel berisi informasi yang dapat mengambil informasi dari objek pada sebuah citra. Tujuan dari ekstraksi ciri sendiri adalah mendefinisikan sifat dari setiap objek yang kita tangkap dalam citra. Ekstraksi citra ini sangat beraneka ragam, berbeda objek dalam citra juga beda metode ekstraksi ciri yang kita pakai. Contoh secara umumnya adalah, jumlah lekukan pada sisi luar, jumlah tonjolan, jumlah lubang pada objek dan masih banyak lagi. Setelah ekstraksi ciri diperoleh kemudian akan dibandingkan dengan data latih yang telah diperoleh sebelumnya dicari tingkat kemiripannya dengan ambang batas. Ekstraksi ciri yang memenuhi ambang batas akan dianggap sebagai kecocokan pada data latih dan kemudian akan menjadikan objek tersebut dikenali oleh sistem dan menerjemahkannya.

2.8 Statistik Spasial

Statistik Spasial adalah segala teknik analisis untuk mengukur distribusi suatu kejadian berdasarkan keruangan (Scott & Warmerdam, 2006). Keruangan dalam maksud di sini adalah suatu variabel yang terdapat pada ruang citra seperti luas daerah, bentuk dan total piksel. Pada pendistribusian Statistik Spasial ada tiga tipe yang berbeda. Pertama *Random*, pada tipe ini kejadian pada ruang spasial yang setiap titik yang bernilai sama mungkin terjadi di setiap lokasi dan posisi titik tersebut saling tidak mempengaruhi satu sama lain. Kedua *Uniform*, tipe ini berkemungkinan setiap titik sama jauh dari semua titik yang berdekatan. Ketiga *Clustered* banyak daerah yang saling berdekatan dan beberapa daerah yang besar akan sangat sedikit apa bila ada titik-titik.

2.9 Penelitian Terdahulu

Pada penelitian terdahulu untuk pendeteksian aksara Jawa sudah ada. Sejauh ini saya menemukan dua penelitian yang mirip yaitu (Widiarti, 2006) dan (Tanaya & Adriani, 2016). Pada penelitian (Widiarti, 2006) menggunakan metode yang hampir mirip namun menggunakan data satu data uji yang merupakan satu lembar teks kuno aksara Jawa sehingga model penulisan gaya huruf yang sering kita sebut *font* relatif sama, sehingga untuk data uji baru yang nanti akan dideteksi akan sangat berbeda hasilnya. Penelitian kedua yaitu (Tanaya & Adriani, 2016) lebih menitik beratkan dalam proses pembacaan aksara Jawa yang sistem deteksi otomatisnya menggunakan metode NLP (*Natural Language Processing*) untuk bahasa non latin. Data pada penelitian tersebut merupakan karakter dari input sistem yang berarti *font* aksara Jawanya semuanya sama persis.

Pada penelitian pengenalan citra aksara Jawa pada plang jalan menggunakan pemodelan Statistik Spasial ini menjadikan metode pada penelitian (Widiarti, 2006) sebagai referensi, namun menggunakan data uji yang lebih kompleks dan data latih yang sangat beragam pastinya. Data uji yang lebih kompleks ini akan menjadikan beberapa metode harus ditambah untuk hasil akurasi yang lebih tinggi dan pada proses *training* pada penelitian ini menggunakan sistem jaringan saraf tiruan.

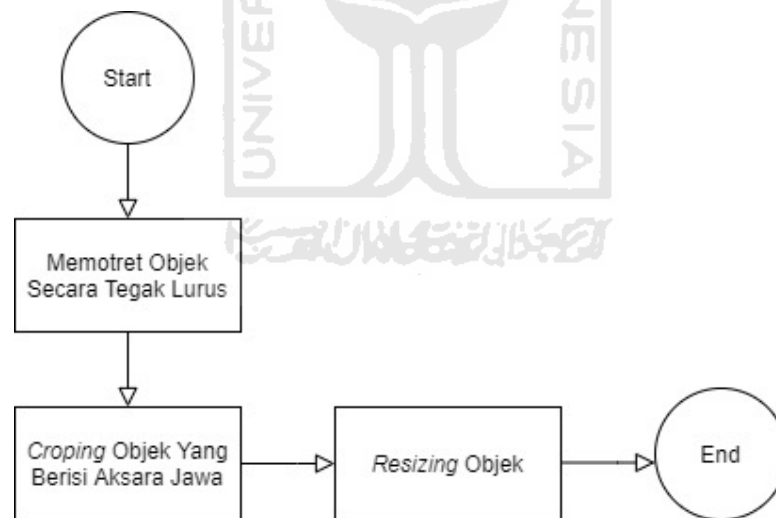
BAB III METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Pada penelitian ini pengumpulan data dilakukan dengan data primer dan data sekunder karena data yang digunakan ada yang diperoleh langsung dari sumbernya dan mengambil via internet. Data penelitian ini berbentuk objek citra atau gambar yang dibagi menjadi dua bagian.

3.1.1 Data Uji

Data uji yaitu sekumpulan data yang nantinya akan dijadikan sebagai data percobaan terhadap sistem, dari data uji ini sistem akan mendeteksi tingkat kemiripan dengan data latih dan proses pencocokan nanti akan ditemukan nilai akurasi yang merupakan hasil pendeteksian sistem terhadap data uji oleh data latih. Proses pengumpulan data uji ini dilakukan dengan cara memotret secara tegak lurus antara kamera dan objek plang jalan tersebut. Berikut merupakan alur kerja proses pengumpulan data uji yang terlihat pada Gambar 3.2.



Gambar 3. 1 *Flowchart* pengumpulan data uji

Pada data uji penelitian ini, total ada 19 yang akan dibagi menjadi 5 jenis gaya huruf yang dikelompokkan oleh penulis sendiri dengan mengamati bentuk huruf aksara Jawa. Proses pengelompokan gaya huruf pada aksara Jawa dilakukan secara manual karena pada aksara

Jawa tidak ada aturan baku penulisan gaya huruf. Contoh data uji yang telah dilakukan proses pengelompokan gaya huruf dan proses edit terlihat di Gambar 3.3.



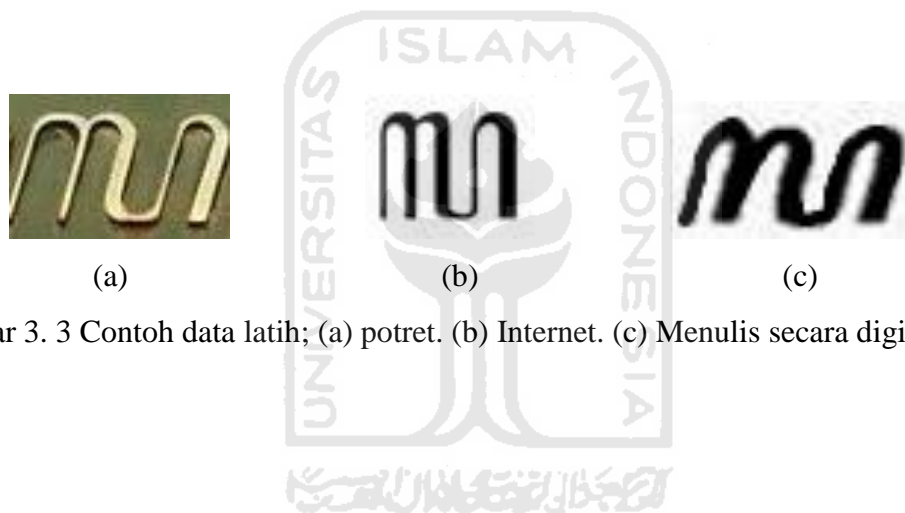
Gambar 3. 2 Contoh data uji berdasarkan pengelompokan gaya huruf

3.1.2 Data Latih

Data latih yaitu sekumpulan data yang nantinya akan diekstrasi ciri citranya dan akan disimpan sebagai acuan atau tolak ukur dari data uji. Pada data latih proses pengumpulan sangat beraneka ragam mulai dari pengunduhan via internet, menuliskan digital secara langsung, dan memotret. Data yang telah dikumpulkan juga memiliki ukuran piksel yang berbeda-beda sehingga harus melalui proses edit agar sesuai batasan masalah. Proses ini membutuhkan data yang beraneka ragam agar data yang nanti akan kita ujikan ada kemiripannya dengan data uji seperti yang ditunjukkan pada Gambar 3.2. Apabila data latih yang dikumpulkan tidak terlalu

banyak, bisa melakukan pemilahan dengan memilihnya yang mirip dengan data yang akan diujikan nantinya. Pada penelitian ini pensortiran dilakukan secara manual melihat dari gaya jenis huruf pada aksara Jawa. Data latih pada penelitian ini diperoleh sebanyak 158 karakter dan terdiri dari 50 kelas.

Pada data latih ini terdapat 158 karakter itu akan menjadi 49 kelas karakter aksara Jawa dan 1 kelas 'not found' atau total seluruh kelas berarti 50. Pada masing-masing kelas terdiri dari seluruh karakter aksara Jawa yang nantinya akan menjadi acuan kecocokan dalam proses pencocokan dengan data uji. Contoh pada kelas ini terdiri dari kelas 'ha' yang mempunyai empat data latih, kelas 'na' yang terdiri dari sebelas data latih dan begitu terus sampai ke kelas 49. Contoh data uji dapat dilihat pada Gambar 3. Tabel total data pada tiap kelas data latih dapat dilihat pada Tabel 3.1.



Gambar 3. 3 Contoh data latih; (a) potret. (b) Internet. (c) Menulis secara digital

Tabel 3. 1 Total data pada tiap kelas data latih

Kelas	Total Data	Kelas	Total Data	Kelas	Total Data	Kelas	Total Data	Kelas	Total Data
Ha	4	Pa	5	Tha Front	1	Pasangan Ta	1	Pepet	3
Na	11	Dha	2	Tha Back	1	Pasangan Ya	1	Swara A Front	1
Ca	1	Ja	5	Nga Front	4	Pasangan Pa	1	Swara A Back	1
Ra	3	Ya	3	Nga Back	3	Pangkon	12	Swara E	2

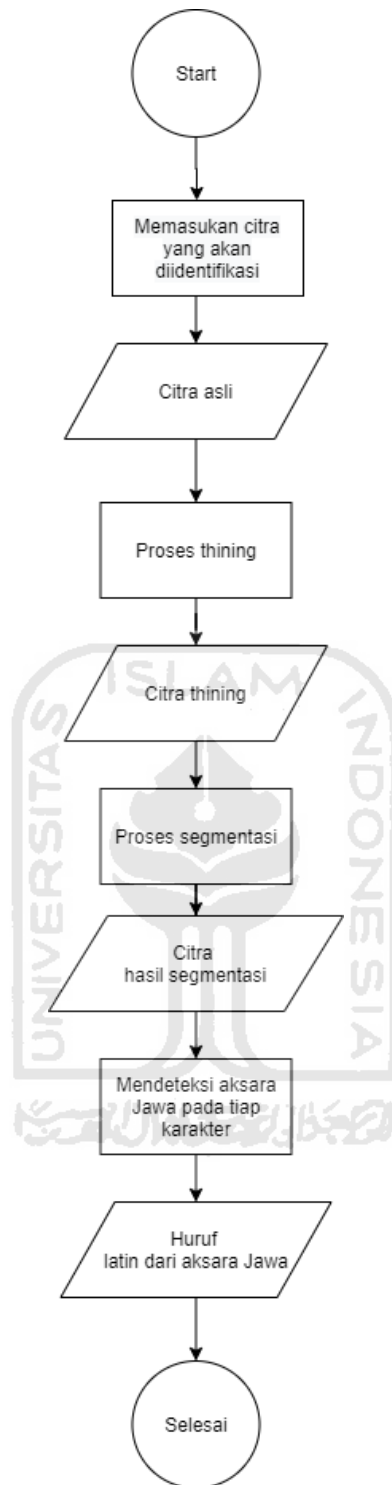
Ka	2	Nya Front	1	Pasangan La	1	Wulu	9	Wigyan	4
Da	13	Nya Back	1	Pasangan Ma	2	Titik	4	Cecak	1
Ta	4	Ma	7	Pasangan Ba	3	Taling	2	Pa Ceret	1
Sa	4	Ga	4	Pasangan Sa	1	Taling Tarung Front	1	Murda Sa	2
Wa	2	Ba Front	2	Pasangan Da	1	Taling Tarung Back	1	Murda Ga	1
La	15	Ba Back	2	Pasangan Ja	1	Layar	3		

3.2 Perancangan Sistem

Proses perancangan sistem dilakukan dengan analisis kebutuhan yang telah dilakukan. Pada proses perancangan sistem akan dibagi menjadi tiga bagian yaitu *flowchart* perancangan sistem, *flowchart* sistem dan desain antarmuka.

3.2.1 *Flowchart* Perancangan Sistem

Flowchart perancangan sistem merupakan penggambaran sederhana apa yang dilakukan oleh sistem secara keseluruhan. Bagian ini dibuat untuk mempermudah pembacaan proses-proses yang terjadi pada sistem. Contoh *flowchart* bisa terlihat pada Gambar 3.4

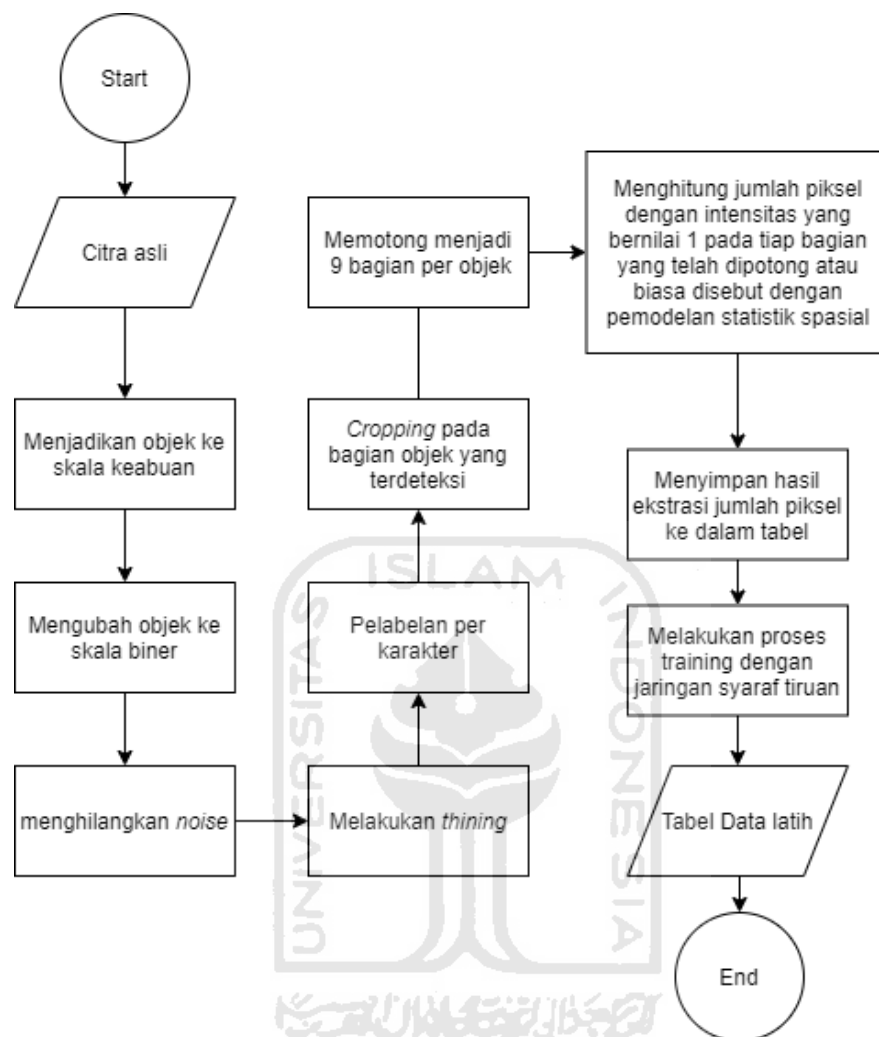


Gambar 3. 4 *Flowchart* perancangan sistem

3.2.2 Flowchart Sistem

Pada bagian ini *flowchart* akan dijelaskan lebih detail dan menerangkan apa saja proses-proses yang terjadi di dalam sistem. Untuk mempermudah penggambaran apa saja

yang terjadi di dalam sistem maka dibuat *flowchart* sistem untuk data latih dan data uji, seperti pada Gambar 3.5 dan Gambar 3.6.



Gambar 3. 5 *Flowchart* sistem pembentukan data latih

Penjelasan *flowchart* sistem pembentukan data latih

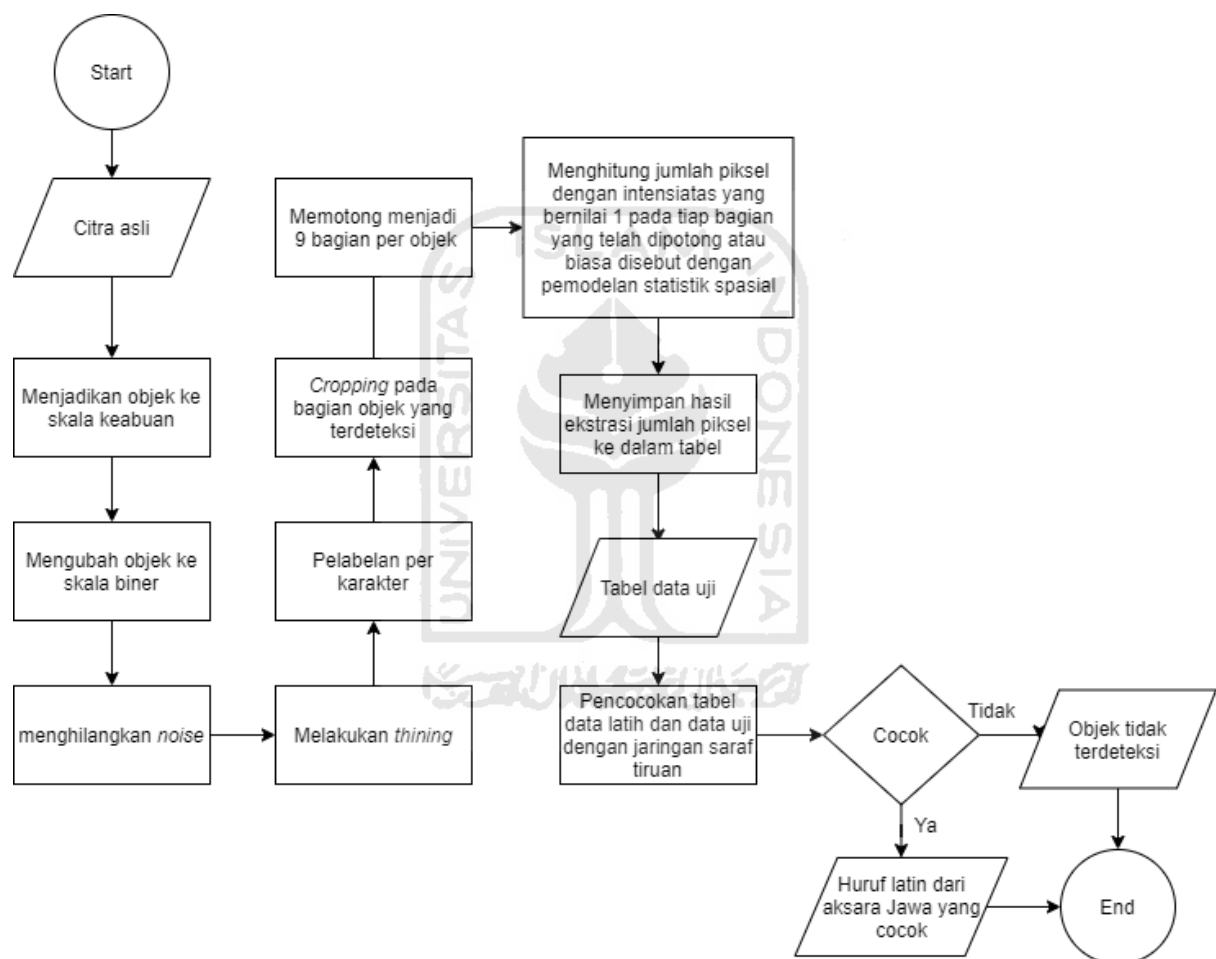
Sistem dimulai ketika menginputkan citra RGB dan kemudian akan melakukan 12 langkah yang nantinya akan menghasilkan output tabel data latih dari karakter aksara Jawa. Penjelasan mengenai 12 langkah tersebut antara lain:

- a. Memuat objek dilakukan untuk memuat data uji ke dalam sistem. Proses ini dilakukan secara manual dengan menekan tombol 'open file' pada GUI, langkah berikutnya tinggal memilih objek yang akan diuji, file yang diuji harus bertipe 'jpg' dan memiliki intensitas warna (RGB).

- b. Proses mengubah objek menjadi skala keabuan bertujuan untuk proses awal untuk mengkonversi citra digital berintensitas warna RGB ke biner. Pada proses kali ini menggunakan fungsi pada matlab yaitu 'rgb2gray'.
- c. Mengubah objek ke skala biner dilakukan untuk mendapatkan 2 intensitas warna yaitu 1 dan 0 yang nantinya piksel 1 akan kita hitung jumlahnya.
- d. Proses penghilangan *noise* bertujuan untuk mencari objek yang terdeteksi pada citra hanyalah karakter aksara Jawa yang nantinya akan kita olah, pada bagian ini menggunakan fungsi pada matlab 'bwareopen'. Proses penghilangan *noise* dilakukan dengan menggunakan ambang batas 20, yang berarti objek yang berukuran kurang dari 20 akan bernilai 0 atau hilang. Ambang batas 20 ditentukan secara manual yang berarti hasil dari percobaan dan menemukan ambang batas 20 yang sejauh ini paling akurat.
- e. Pada proses *thining* atau bisa disebut penipisan dilakukan untuk mengambil informasi dari objek seminimal mungkin namun sangat penting bagian-bagian tebal pada karakter aksara Jawa akan menjadi satu garis kecil yang akan diambil informasinya.
- f. Pelabelan per karakter bertujuan untuk menandai antar objek yang terpisah, sehingga sistem mendeteksi satu persatu karakter aksara Jawa.
- g. *Cropping* menjadikan objek yang sudah terlabeli akan dipotong-potong menjadi per karakter aksa Jawa sesuai jumlah label yang terdeteksi, kemudian menyimpannya dalam variabel bari.
- h. Pemotongan 9 bagian, setelah mendapatkan citra baru yang telah melalui proses *cropping* kemudian citra baru tersebut akan dipotong ke dalam 9 bagian dengan cara membagi 3 panjang piksel dan membagi 3 lebar piksel.
- i. Menghitung semua jumlah piksel dengan intensitas yang bernilai 1 dari 9 bagian yang telah dipotong pada tiap karakter yang biasa disebut pemodelan Statistik Spasial.
- j. Menyimpan hasil perhitungan jumlah piksel dengan intensitas yang bernilai 1 akan disimpan pada sebuah tabel baru yang nantinya akan menghasilkan jumlah kolom sesuai jumlah karakter yang terdeteksi dan menghasilkan 9 baris yang berisi nilai penjumlahan piksel dengan intensitas bernilai 1.
- k. Melakukan proses *training* dengan menggunakan jaringan saraf tiruan dari hasil ekstrasi menggunakan pemodelan Statistik Spasial dengan target latih yang telah dibuat sebelumnya. Penentuan nilai *epoch*, *goal*, *hidden layer*, dan *lr* dilakukan dengan cara melakukan percobaan manual dengan mencoba satu persatu, maka ditemukanlah *epochs* dengan nilai 1000, *goal* atau total eror dengan nilai 0,001, *hidden layer* dengan

nilai 1 yang memiliki 245 node dan lr dengan nilai 0,001 dengan penentuan nilai tersebut diharapkan akan memperoleh hasil *training* yang sempurna. Arsitektur dari jaringan saraf tiruan *feedforward* yang digunakan pada penelitian kali ini tersusun atas 1 layer input, 1 layer *hidden*, dan 1 layer output. Layer input berisi terdiri atas 9 *node*, layer *hidden* terdiri dari 245 *node* dan layer output terdiri dari 49 *node*.

1. Menampilkan tabel hasil dari proses *training* yang setiap tabelnya menunjukkan angka dari 0 sampai 1 yang menunjukkan kecocokan dengan target latih.



Gambar 3. 6 *Flowchart* sistem pengujian

Penjelasan *flowchart* sistem pengujian

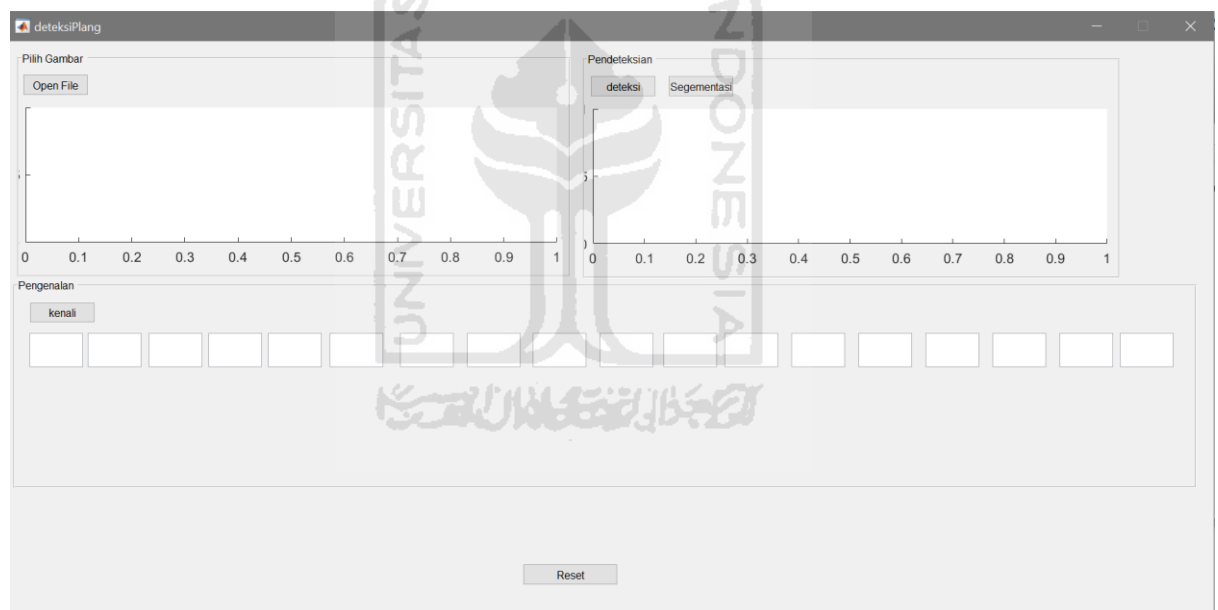
Sistem dimulai ketika menginputkan citra RGB dan kemudian akan melakukan 15 langkah yang nantinya akan menghasilkan output huruf latin dari karakter aksara Jawa atau karakter tidak terdeteksi. Penjelasan mengenai 15 proses tersebut antara lain:

- a. Memuat objek dilakukan untuk memuat data uji ke dalam sistem. Proses ini dilakukan secara manual dengan menekan tombol 'open file' pada GUI, langkah berikutnya tinggal memilih objek yang akan diuji, file yang diuji harus bertipe 'jpg' dan memiliki intensitas warna (RGB).
- b. Proses mengubah objek menjadi skala keabuan bertujuan untuk proses awal untuk mengkonversi citra digital berintensitas warna RGB ke biner. Pada proses kali ini menggunakan fungsi pada matlab yaitu 'rgb2gray'.
- c. Mengubah objek ke skala biner dilakukan untuk mendapatkan dua intensitas warna yaitu 1 dan 0 yang nantinya piksel 1 akan kita hitung jumlahnya.
- d. Proses penghilangan *noise* bertujuan untuk mencari objek yang terdeteksi pada citra hanyalah karakter aksara Jawa yang nantinya akan kita olah, pada bagian ini menggunakan fungsi pada matlab 'bwareopen'. Proses penghilangan *noise* dilakukan dengan menggunakan ambang batas 20, yang berarti objek yang berukuran kurang dari 20 akan bernilai 0 atau hilang. Ambang batas 20 ditentukan secara manual yang berarti hasil dari percobaan dan menemukan ambang batas 20 yang sejauh ini paling akurat.
- e. Pada proses *thining* atau bisa disebut penipisan dilakukan untuk mengambil informasi dari objek seminimal mungkin namun sangat penting bagian-bagian tebal pada karakter aksara Jawa akan menjadi satu garis kecil yang akan diambil informasinya.
- f. Pelabelan per karakter bertujuan untuk menandai antar objek yang terpisah, sehingga sistem mendeteksi satu persatu karakter aksara Jawa.
- g. *Cropping* menjadikan objek yang sudah terlabeli akan dipotong-potong menjadi per karakter aksa Jawa sesuai jumlah label yang terdeteksi, kemudian menyimpannya dalam variabel bari.
- h. Pemotongan sembilan bagian, setelah mendapatkan citra baru yang telah melalui proses *cropping* kemudian citra baru tersebut akan dipotong ke dalam sembilan bagian dengan cara membagi 3 panjang piksel dan membagi tiga lebar piksel.
- i. Menghitung semua jumlah piksel dengan intensitas yang bernilai 1 dari sembilan bagian yang telah dipotong pada tiap karakter yang biasa disebut pemodelan Statistik Spasial.
- j. Menyimpan hasil perhitungan jumlah piksel dengan intensitas yang bernilai 1 akan disimpan pada sebuah tabel baru yang nantinya akan menghasilkan jumlah kolom sesuai jumlah karakter yang terdeteksi dan menghasilkan sembilan baris yang berisi nilai penjumlahan piksel dengan intensitas bernilai 1.

- k. Menampilkan tabel dari data yang telah melalui ekstraksi ciri pemodelan Statistik Spasial yang berisi sembilan baris dan kolom sesuai dari karakter yang terdeteksi.
- l. Pencocokan tabel data uji dan data latih dengan jaringan saraf tiruan dengan perintah 'sim'.
- m. Jika data uji ada yang cocok dengan data latih maka sistem akan menampilkan huruf latin dari aksara Jawa dari karakter yang terbilang cocok di data latih.
- n. Jika data uji tidak ada yang cocok maka sistem akan menampilkan 'not found' .

3.2.3 Desain Antarmuka

Desain antarmuka dibuat bertujuan untuk mempermudah interaksi antara pengguna dan sistem. Tahap ini juga merupakan gambaran tentang sistem pengenalan karakter Jawa pada plang jalan dan memuat halaman proses input proses dan output. Desain antarmuka dapat dilihat pada Gambar 3.6.



Gambar 3. 7 Desain antarmuka

3.3 Implementasi

Proses implementasi dilakukan dengan aplikasi MATLAB R2016b dan menggunakan beberapa *tools* yang berada dalam aplikasi tersebut.

3.4 Pengujian

Tahap pengujian dilakukan agar mengetahui hasil dari pengenalan sistem terhadap objek yang telah terdeteksi. Pengujian dilakukan secara manual dengan menerjemahkan secara karakter per karakter. Sistem akan diuji untuk mengetahui seberapa besar keberhasilan sistem

dalam akurasi pengenalan karakter aksara Jawa. Ada tiga metode yang dilakukan untuk menguji sistem dalam penelitian ini.

3.4.1 *Black Box Testing*

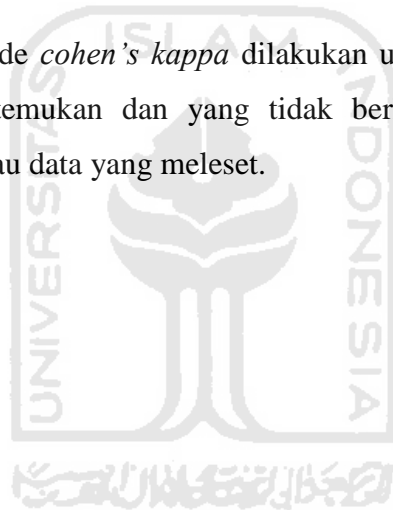
Pada langkah pengujian *black bock testing* dilakukan untuk menilai sistem dari tampilan luar mulai dari sistem pertama dijalankan menuju pencarian input data uji sampai menghasilkan nilai output untuk pengenalan citra aksara Jawa.

3.4.2 Perhitungan Statistik Ketepatan

Pengujian menggunakan perhitungan statistik dilakukan untuk menghasilkan nilai akurasi pengenalan citra aksara Jawa dari masing-masing karakter. Pengujian ini menghitung total karakter yang berhasil dideteksi dan membaginya dengan total seluruh karakter yang ada, kemudian mengalikannya dengan nilai 100%.

3.4.3 *Cohen's Kappa*

Pengujian menggunakan metode *cohen's kappa* dilakukan untuk menemukan total data dari nilai output yang berha ditemukan dan yang tidak berhasil ditemukan kemudian mengakumulasi data yang *error* atau data yang meleset.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Format Implementasi sistem menggunakan aplikasi MATLAB R2016b dengan beberapa *tools* dan fungsi bawaan yang telah tersedia pada aplikasi tersebut.

4.1.1 Input Citra

Pada proses ini bertujuan untuk si pengguna akan memasukan data uji atau objek yang akan dikenali. Citra yang akan dikenali harus memiliki intensitas warna RGB dan bertipe jpg, kemudian citra tersebut akan ditampilkan pada 'axes1'. Kode program input citra bisa dilihat pada Gambar 4.1 dan hasil dari input citra RGB dapat dilihat pada Gambar 4.2.

```
[filename,pathname] = uigetfile('*.jpg*');
if ~isequal(filename,0)
    img = imread(fullfile(pathname,filename));
    [~,~,m] = size(img);
    if m == 3
        axes(handles.axes1)
        imshow(img)
        handles.img = img;
        guidata(hObject, handles)
    else
        msgbox('Please insert RGB Image')
    end
else
end
```

Gambar 4. 1 Kode program input citra



Gambar 4. 2 Hasil input data citra RGB

4.1.2 Citra Skala Keabuan

Citra RGB yang berhasil diinputkan ke dalam sistem intensitas warnanya akan dikonversikan ke skala keabuan atau *grayscale*. Proses ini adalah langkah awal untuk mengubah citra RGB ke biner. Kode program bisa dilihat pada Gambar 4.3 dan hasil dari proses skala keabuan bisa dilihat pada gambar 4.4.

```
Img_gray = rgb2gray(img);
```

Gambar 4. 3 Kode program skala keabuan



Gambar 4. 4 Citra skala keabuan

4.1.3 Citra Skala Biner

Proses ini dilakukan ketika citra sudah berskala keabuan, setelah itu sistem akan otomatis me memilah intensitas warna dari skala keabuan menjadi angka 0 dan 1 atau biasa disebut biner. Dalam kasus ini ambang batas yang digunakan adalah ambang batas yang ditentukan sendiri oleh sistem. Kode program dapat dilihat pada Gambar 4.5 dan hasil dari proses skala biner dapat dilihat pada Gambar 4.6.

```
bw = im2bw(Img_gray,graythresh(Img_gray));
```

Gambar 4. 5 Kode program skala biner



Gambar 4. 6 Hasil citra skala biner

4.1.4 Menghilangkan *Noise*

Proses menghilangkan *noise* dilakukan secara manual dengan menghilangkan objek yang ukurannya lebih kecil dari nilai ambang batas yang kita tentukan. Pada proses ini ambang batas ditentukan secara manual yaitu 20, yang berarti objek yang ukurannya lebih kecil dari 20 sistem akan menjadikan intensitas warna menjadi 0 atau pada kasus ini hitam. Kode program bisa dilihat pada Gambar 4.7. Dapat dilihat pada Gambar 4.6 terlihat di lingkaran hijau masih terdapat *noise* dan hasil penghilangan *noise* pada Gambar 4.8 sudah tidak ada *noise* yang terdeteksi.

```
bw = bwareaopen(bw, 20);
```

Gambar 4. 7 Kode program menghilangkan *noise*Gambar 4. 8 *Noise* yang sudah dihilangkan pada citra

4.1.5 *Thining*

Proses *thining* ini dilakukan untuk menipiskan objek yang terdeteksi sehingga informasi yang kita dapatkan tergolong sedikit namun sangat penting. Bagian ini dilakukan dengan menggunakan fungsi pada matlab 'bwmorph' dengan perintah 'thicken'. Kode program bisa dilihat pada Gambar 4.9 dan hasil proses *thining* bisa dilihat pada Gambar 4.10.

```
tipis = bwmorph(bw, 'thicken', inf);
```

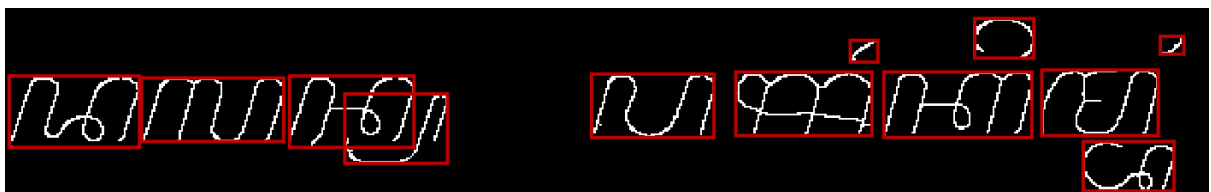
Gambar 4. 9 Kode program *thining*Gambar 4. 10 Hasil citra proses *thining*

4.1.6 Pelabelan Perkarakter

Pelabelan perkarakter bertujuan untuk memberi tanda setiap karakter pada citra yang saling terpisah. Pada kasus ini menggunakan fungsi 'bwlablel' dengan nilai ambang batas 8 kemudian menyimpannya pada dua variabel matriks yaitu 'labeled' dan 'numObjects'. Variabel 'labeled' berisi kumpulan informasi dari antar objek yang terpisah dan variabel 'numObjects' berisi jumlah karakter yang terdeteksi pada citra, kemudian variabel 'labeled' diberi informasi tambahan oleh fungsi 'regionprops' dengan properti 'BoundingBox' yang berisi letak posisi dan ukuran dalam kotak terkecil dari tiap karakter. Kode program bisa dilihat pada Gambar 4.11 dan hasil proses pelabelan perkarakter dapat dilihat pada Gambar 4.12.

```
[labeled, numObjects] = bwlablel(tipis,8);
stats = regionprops(labeled,'BoundingBox');
bbox = cat(1, stats.BoundingBox);
```

Gambar 4. 11 Kode program pelabelan perkarakter



Gambar 4. 12 Hasil proses pelabelan

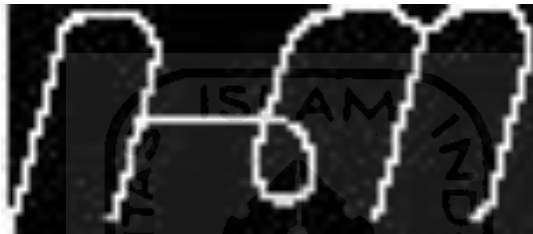
4.1.7 Cropping Objek

Cropping atau pemotongan pada objek tujuannya untuk membuat citra baru yang kemudian akan diambil informasinya dari citra baru ini. Proses ini dilakukan ketika sistem sudah mendeteksi posisi tiap karakter terpisah yang nantinya akan dipotong persis pada tepi karakter. Sebelum melakukan *cropping* terlebih dahulu harus mencari letak posisi dengan cara

membuat variabel baru ‘posi’ untuk mendapatkan posisi yang kemudian menjadi titik potong pada tiap karakter terpisah dan kode program dijalankan pada teknik *looping* sehingga proses ini akan terjadi berulang kali sampai jumlah karakter yang terdeteksi habis. Kode program bisa dilihat pada Gambar 4.11 dan ilustrasi hasil *cropping* bisa dilihat pada Gambar 4.12.

```
h = rectangle('Position', bbox(idx, : ) , 'LineWidth', 2);
posi = get(h, 'Position');
PosOb(idx, 1:4) = posi;
detec = imcrop(tipis, posi);
```

Gambar 4. 13 Kode program *cropping*



Gambar 4. 14 Ilustrasi hasil *cropping*

4.1.8 Pemodelan Statistik Spasial dan Pemotongan Menjadi 9 Bagian

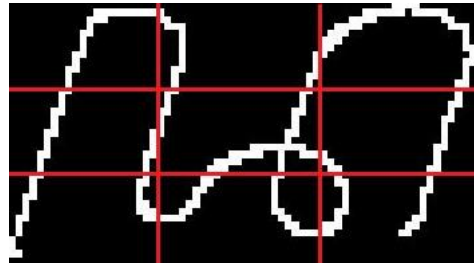
Proses implementasi pemodelan Statistik Spasial dalam penelitian ini adalah dengan mengambil setiap piksel dengan intensitas bernilai 1 (berwarna putih) pada masing-masing karakter yang telah dipotong menjadi sembilan bagian. Penggunaan pemodelan Statistik Spasial dirasa cukup ampuh karena masing-masing karakter mempunyai jumlah piksel dengan intensitas bernilai 1 yang unik dan berbeda-beda, dengan alasan ini maka ekstrasi ciri cukup hanya menggunakan pemodelan Statistik Spasial ini. Pembagian sembilan potongan ini menjadikan tingkat keunikan pada tiap karakter menjadi semakin kental dan menjadikan tiap jumlah piksel dengan intensitas bernilai 1 dalam sebuah karakter semakin berbeda-beda dan karakter yang sama akan terjadi kemiripan. Pada masa eksperimen yang menyimpulkan pemotongan sembilan bagian, sebelumnya dilakukan teknik pemotongan yang kurang dari sembilan, hasilnya saat penjumlahan intensitas piksel yang bernilai 1 akan terlalu umum dan tidak menjadikan ciri yang unik antar karakter dan begipun pada teknik pemotongan yang lebih dari 9 bagian yang menjadikan keunikan karakter terlalu unik membuat sistem sulit untuk mengenalinya.

Pemotongan antar karakter menjadi sembilan bagian bertujuan untuk menghitung jumlah piksel dengan intensitas bernilai 1 pada pembagian piksel tersebut. Pemotongan ini menggunakan teknik membagi 3 panjang dan lebar karakter sehingga sembilan bagian tersebut akan sama. Setelah pemotongan sembilan bagian, langkah berikutnya menyimpan sembilan citra baru tersebut ke dalam variabel baru. Dalam kasus ini, alasan pemotongan menjadi sembilan bagian adalah dilakukan dengan eksperimen yang telah dilakukan berulang kali sehingga memunculkan pemotongan sembilan bagian menjadi teknik yang paling efektif. Pemotongan sembilan bagian dilakukan pada pada anak subbab **4.1.8**, contoh studi kasus dalam pemodelan Statistik Spasial dilakukan pada huruf 'Da' yang dibagi menjadi sembilan bagian dan kemudian menghitung jumlah piksel dengan intensitas yang bernilai 1 tiap bagian. Proses ini dilakukan berulang kali hingga semua karakter yang terdeteksi dihitung jumlah piksel dengan intensitas bernilai 1. Kode program dapat dilihat pada Gambar dan ilustrasi pemotongan sembilan bagian dapat dilihat pada Gambar 4.13 dan ilustrasi hasil pemotongan menjadi sembilan bagian dapat dilihat pada gambar 4.14.

```
[rows, columns, numberOfColorChannels] = size(detec);
r3 = int32(rows/3);
c3 = int32(columns/3);

%Extract the 9 images.
Image1 = detec(1:r3, 1:c3);
image2 = detec(1:r3, c3+1:2*c3);
image3 = detec(1:r3, 2*c3+1:columns);
image4 = detec(r3+1:2*r3, 1:c3);
image5 = detec(r3+1:2*r3, c3+1:2*c3);
image6 = detec(r3+1:2*r3, 2*c3+1:columns);
image7 = detec(2*r3+1:end, 1:c3);
image8 = detec(2*r3+1:end, c3+1:2*c3);
image9 = detec(2*r3+1:end, 2*c3+1:columns);
```

Gambar 4. 15 kode program pemotongan 9 bagian



(a)

26	27	29
19	42	31
48	21	27

(b)

Gambar 4. 16 (a) Pemotongan huruf 'Da' (b) Jumlah intensitas piksel tiap potongan

4.1.9 Penyimpanan Piksel dengan Intensitas Bernilai 1

Penyimpanan jumlah piksel dengan intensitas bernilai 1 dilakukan perbagian karakter yang telah dibagi menjadi sembilan bagian, sehingga per karakter ada perhitungan penjumlahan piksel dengan intensitas bernilai 1 sebanyak sembilan kali dan kemudian disimpan ke dalam variabel baru kemudian akan disimpan pada sebuah tabel. Tabel tersebut berisi hasil penjumlahan piksel dengan intensitas bernilai 1 pada tahap sebelumnya. Tabel berisi sembilan baris sesuai dengan sembilan bagian penjumlahan intensitas dan kolom terdiri dari jumlah karakter yang terdeteksi. Kode program dapat dilihat pada Gambar 4.17 dan Tabel dapat dilihat pada Tabel 4.1.

```

jumlahPixel1 = sum(image1(:));
jumlahPixel2 = sum(image2(:));
jumlahPixel3 = sum(image3(:));
jumlahPixel4 = sum(image4(:));
jumlahPixel5 = sum(image5(:));
jumlahPixel6 = sum(image6(:));
jumlahPixel7 = sum(image7(:));
jumlahPixel8 = sum(image8(:));
jumlahPixel9 = sum(image9(:));

ciriUji(1,idx) = jumlahPixel1;
ciriUji(2,idx) = jumlahPixel2;
ciriUji(3,idx) = jumlahPixel3;
ciriUji(4,idx) = jumlahPixel4;
ciriUji(5,idx) = jumlahPixel5;

```

```

ciriUji(6,idx) = jumlahPixel6;
ciriUji(7,idx) = jumlahPixel7;
ciriUji(8,idx) = jumlahPixel8;
ciriUji(9,idx) = jumlahPixel9;

```

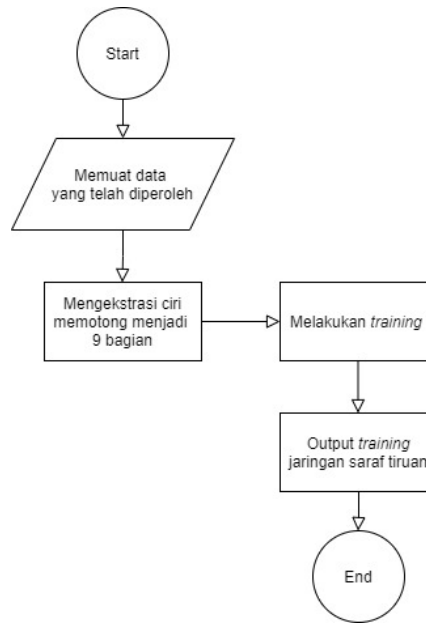
Gambar 4. 17 Kode program penyimpanan piksel dengan intensitas bernilai 1

Tabel 4. 1 Tabel penjumlahan intensitas piksel

26	32	27	35	29	52	0	36	20	30	26	0
19	42	17	15	10	36	8	24	11	32	8	0
48	38	44	36	38	44	5	52	14	37	25	3
27	25	30	30	28	44	7	34	9	19	11	0
42	17	40	18	2	38	3	36	0	19	25	2
21	29	23	32	29	14	0	27	9	28	15	5
29	27	29	33	26	11	7	26	7	23	19	4
31	26	19	22	37	14	0	22	0	29	26	7
27	29	23	6	12	37	0	25	13	15	18	0

4.1.10 Pembuatan Data Latih

Pembuatan data latih dilakukan terlebih dahulu sebelum melakukan pengenalan karena data latih di sini akan menjadi acuan bahwa sistem akan terdeteksi atau tidaknya pada bagian ini. Langkah pertama dalam pembuatan data latih sama persis dengan langkah-langkah sebelumnya namun dengan citra yang berbeda namun hampir mirip yang nantinya akan disimpan pada sebuah tabel. Tabel dalam penelitian ini terdiri dari 9 baris dan 158 kolom yang berarti data latih telah mempunyai 158 karakter yang dikenali dan menjadi sebuah titik acuan. Contoh proses pembuatan data latih dapat dilihat pada Gambar 4.18 dan hasil pembuatan data latih dapat dilihat pada Tabel 4.2.

Gambar 4. 18 *flowchart* pembuatan data latih

Tabel 4. 2 Pembuatan data latih

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
ha1	ha2	ha3	ha4	na1	na2	na3	na4	na5	na6	na7	na8	na9	na10	na11	ca	ra1	ra2	ra3	ka1	ka2	da1
44	37	36	29	59	33	34	31	29	22	23	27	34	32	21	33	12	20	19	36	36	32
25	24	25	16	69	15	14	16	11	23	22	17	14	23	21	21	21	16	10	24	24	16
57	36	38	29	60	27	27	21	24	38	30	44	28	19	39	48	18	20	17	52	40	26
32	25	26	22	43	26	28	25	21	22	23	30	28	29	21	35	0	5	14	34	29	27
17	16	13	11	56	25	27	22	11	43	37	40	26	32	43	12	11	8	0	36	30	20
29	25	25	22	46	20	21	20	11	19	21	23	20	12	24	26	11	11	13	27	24	14
33	25	25	20	50	23	12	20	23	28	26	29	26	17	26	47	0	2	12	26	18	32
20	23	24	16	25	16	20	13	32	41	31	19	19	24	36	43	10	11	1	22	24	34
23	22	23	20	37	26	21	24	21	16	14	23	17	17	24	13	9	11	11	25	17	17

4.1.11 Pembuatan Target Latih

Target latih pada penelitian menggunakan angka 0 dan 1 karena dalam proses *training* jaringan saraf tiruan pada matlab hanya bisa mendeteksi angka antara 0 sampai 1. Proses pembuatan target latih ini berdasarkan kelas yang akan kita bagi dalam data latih. Dalam penelitian ini dari 158 karakter sebagai data latih akan dibagi menjadi 46 kelas yang berbeda sehingga tabel target latih memuat 46 baris dan 158 kolom. Contoh tabel terlihat pada Tabel 4.3, dalam tampilan tabel tersebut hanya memuat dua kelas berbeda yaitu pada baris 1, kolom 1 sampai 4 kelas karakter ‘Ha’ dan baris 2, kolom 5 dan seterusnya memuat kelas karakter ‘Na’.

Tabel 4. 3 Target

target												
PLOTS		VARIABLE				VIEW						
46x158 double												
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0

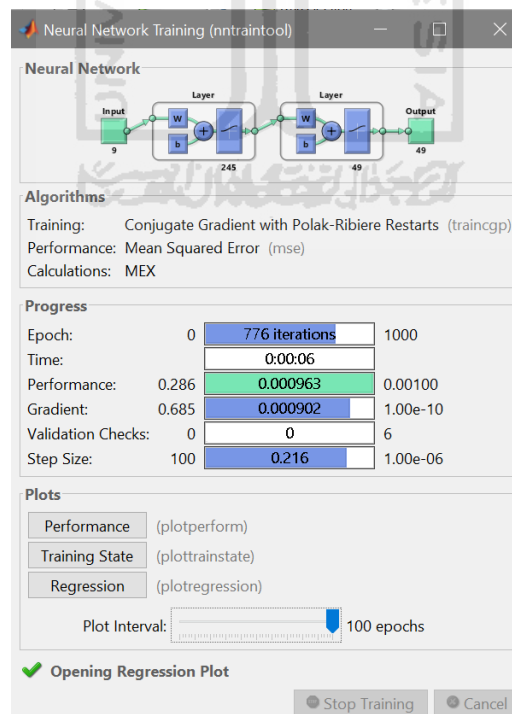
4.1.12 Pembentukan Jaringan Saraf Tiruan

Proses pembentukan jaringan saraf tiruan bertujuan untuk melakukan proses *training* pada data latih yang berpatokan pada target latih. Jaringan saraf tiruan ini bekerja dengan memperoleh fungsi aktivasi pada masing-masing input yang terhubung oleh *hidden layer* sehingga memperoleh output. Bagian pada jaringan saraf tiruan berisi input data, *synapse*, *hidden layer*, node, dan output.

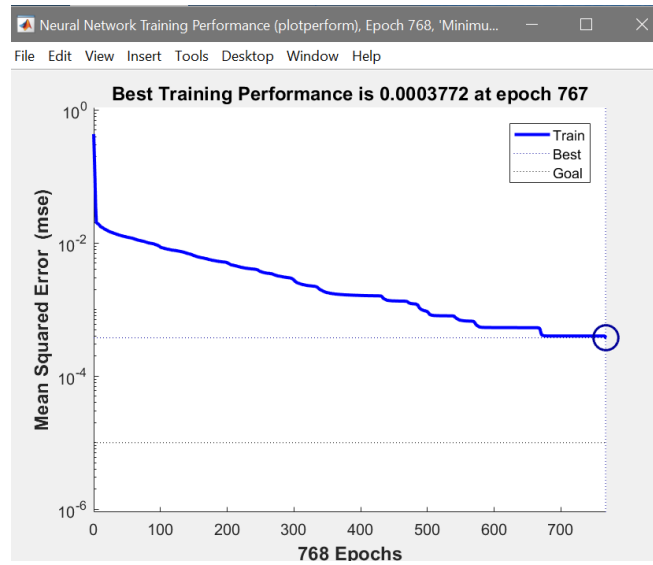
- Input data berisikan jumlah baris pada kolom tabel input yang nantinya masing-masing input akan dikalikan dengan bobot pada masing-masing garis penghubung antara input dan *hidden layer*.
- Synapse* adalah garis penghubung antara proses input data dan *hidden layer* yang masing-masing *synapse* memiliki bobot untuk dikalikan dengan data input.
- Hidden layer* atau sering disebut neurons prosesnya lebih kompleks, prosesnya menambahkan semua hasil perkalian bobot yang telah dibawa oleh *synapse* kemudian mengaktifkan fungsi aktivasi pada semua proses penambahan tersebut.
- Epoch* adalah sebuah pengaturan di dalam jaringan saraf tiruan. *Epoch* bertugas mengatur ukuran berapa kali semua vektor yang digunakan dalam satu kali untuk memperbarui bobot.

- e) Node adalah berisi pengumpulan perhitungan antara yang telah dilakukan oleh *synapse*. Biasanya ada beberapa node dalam *hidden layer*.
- f) Output pada proses ini *hidden layer* yang terakhir dari kelas atau target yang ditentukan akan menjadikan *hidden layer* menjadi proses output dan menjadikan nilai output sebagai nilai prediksi dari proses input data.

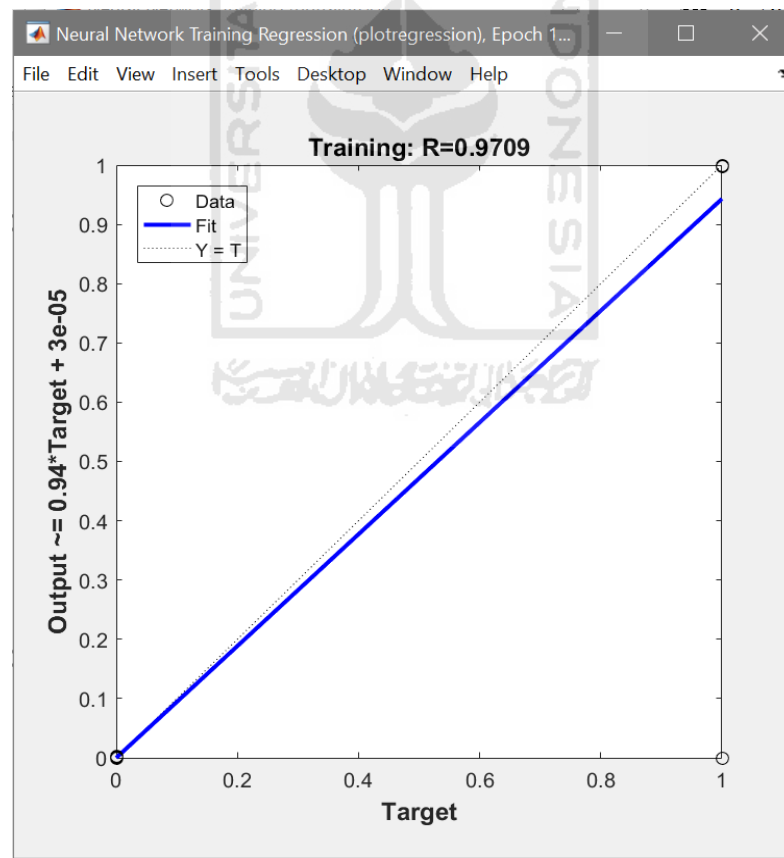
Proses jaringan saraf tiruan ini akan menghasilkan tabel data yang berisi nilai antara 0 sampai satu yang berarti nilai 0 yaitu tidak memiliki bobot dan nilai 1 yang menjadi acuan sempurna. Pada pembentukan jaringan saraf tiruan dalam penelitian ini menggunakan jaringan *feedforward* menggunakan fungsi 'newff' yang disediakan oleh *toolbox neural network* pada Matlab dan mengkombinasikannya dengan *backpropagation*. Ilustrasi hasil proses *training* menggunakan jaringan saraf tiruan bisa terlihat pada Gambar 4.17. Pada Gambar 4.19 menjelaskan proses *training* performa terbaik yang hampir mendekati *goal*. Pada Gambar 4.20 menjelaskan hasil performa dari proses *training*. Pada Gambar 4.21 menjelaskan hasil regresi antara hasil *training* dan target yang hampir mendekati sempurna. Pembuatan Kode program jaringan saraf tiruan *backpropagation* dengan 'newff' terlihat pada Gambar 4.22.



Gambar 4. 19 Hasil proses *training* jaringan saraf tiruan



Gambar 4. 20 Hasil performa

Gambar 4. 21 Hasil proses *training* regresi

```
Node_hidden_layer = 245;
```

```

output=49;

net= newff(minmax(data_latih), [node_hidden_layer, output],
{'logsig','logsig'}, 'traincgp');

net.trainParam.epochs = 1000;
net.trainParam.goal = 0.001;
net.trainParam.lr=0.001;
net=init(net);
[net,tr]=train(net,data_latih, target);
save net;
hasil = sim(net, data_latih)

```

Gambar 4. 22 Kode program jaringan saraf tiruan

4.1.13 Pendeteksian Aksara Jawa

Pendeteksian aksara dilakukan berdasarkan hasil jaringan saraf tiruan dari data latih dan target latih sebagai acuan awal. Data uji yang hampir mendekati hasil jaringan saraf tiruan akan dicetak dengan huruf latin. Pada penelitian ini ketika tabel karakter pertama dan baris pertama nilainya bernilai lebih dari 0.5 akan dicetak sebagai huruf 'Ha', ketika karakter pertama dan baris kedua bernilai lebih dari 0.5 akan dicetak sebagai huruf 'Na' dan begitu seterusnya hingga semua karakter aksara Jawa masuk dalam percabangan dan dimuat ulang oleh perulangan yang memuat semua karakter yang terdeteksi. Kode program dapat dilihat pada Gambar 4.23.

```

[p, l]= size(ciriUji);
jumlah = l;
hasil = sim(net, ciriUji)
baca = zeros(9,jumlah);
n=1;
m=1;
for (a=1:jumlah)
    if find(hasil(1,n)>0.5)
        display 'ha';
    elseif find(hasil(2,n)>0.5)
        display 'na';
    elseif find(hasil(3,n)>0.5)
        display 'ca';
    elseif find(hasil(4,n)>0.5)
        display 'ra';
    elseif find(hasil(5,n)>0.5)
        display 'ka';
    elseif find(hasil(6,n)>0.5)
        display 'da';
    elseif find(hasil(7,n)>0.5)

```

```
        display 'ta';
elseif find(hasil(8,n)>0.5)
    display 'sa';
elseif find(hasil(9,n)>0.5)
    display 'wa';
elseif find(hasil(10,n)>0.5)
    display 'la';
elseif find(hasil(11,n)>0.5)
    display 'pa';
elseif find(hasil(12,n)>0.5)
    display 'dha';
elseif find(hasil(13,n)>0.5)
    display 'ja';
elseif find(hasil(14,n)>0.5)
    display 'ya';
elseif find(hasil(15,n)>0.5)
    display 'nya_depan';
elseif find(hasil(16,n)>0.5)
    display 'nya_belakang';
elseif find(hasil(17,n)>0.5)
    display 'ma';
elseif find(hasil(18,n)>0.5)
    display 'ga';
elseif find(hasil(19,n)>0.5)
    display 'ba_depan';
elseif find(hasil(20,n)>0.5)
    display 'ba_belakang';
elseif find(hasil(21,n)>0.5)
    display 'tha_depan';
elseif find(hasil(22,n)>0.5)
    display 'tha_belakang';
elseif find(hasil(23,n)>0.5)
    display 'nga_depan';
elseif find(hasil(24,n)>0.5)
    display 'nga_belakang';
elseif find(hasil(25,n)>0.5)
    display 'pasangan_la';
elseif find(hasil(26,n)>0.5)
    display 'pasangan_ma';
elseif find(hasil(27,n)>0.5)
    display 'pasangan_ba';
elseif find(hasil(28,n)>0.5)
    display 'pasangan_sa';
elseif find(hasil(29,n)>0.5)
    display 'pasangan_da';
elseif find(hasil(30,n)>0.5)
    display 'pasangan_ja';
elseif find(hasil(31,n)>0.5)
    display 'pasangan_ta';
elseif find(hasil(32,n)>0.5)
```

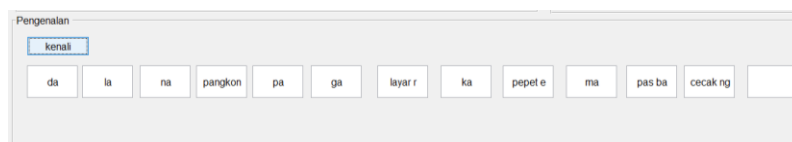
```

        display 'pasangan ya';
elseif find(hasil(33,n)>0.5)
    display 'pangkon';
elseif find(hasil(34,n)>0.5)
    display 'wulu i';
elseif find(hasil(35,n)>0.5)
    display 'titik';
elseif find(hasil(36,n)>0.5)
    display 'taling e';
elseif find(hasil(37,n)>0.5)
    display 'taling tarung depan o';
elseif find(hasil(38,n)>0.5)
    display 'taling tarung belakang o';
elseif find(hasil(39,n)>0.5)
    display 'layar r';
elseif find(hasil(40,n)>0.5)
    display 'pepet e';
elseif find(hasil(41,n)>0.5)
    display 'swara depan a';
elseif find(hasil(42,n)>0.5)
    display 'swara belakang a';
elseif find(hasil(43,n)>0.5)
    display 'swara e';
elseif find(hasil(44,n)>0.5)
    display 'wigyan h';
elseif find(hasil(45,n)>0.5)
    display 'cecak ng';
elseif find(hasil(46,n)>0.5)
    display 'ceret';
else
    display 'tidak terdeteksi';
end
n = n+1;

```

end

Gambar 4. 23 Kode program pendeteksian aksara Jawa



Gambar 4. 23 Hasil pengenalan aksara Jawa

4.2 Implementasi Antarmuka

Antar muka dibuat agar semua mempermudah interaksi antar pengguna dan sistem. Dalam penelitian ini desain antarmuka akan menampilkan dari proses awal yaitu input citra

yang akan dideteksi sampai proses output yaitu huruf latin hasil pengenalan karakter aksara Jawa. Desain antarmuka ini dibuat dengan *tool* bawaan matlab R2016b yang bernama Guide.

4.2.1 Input Citra

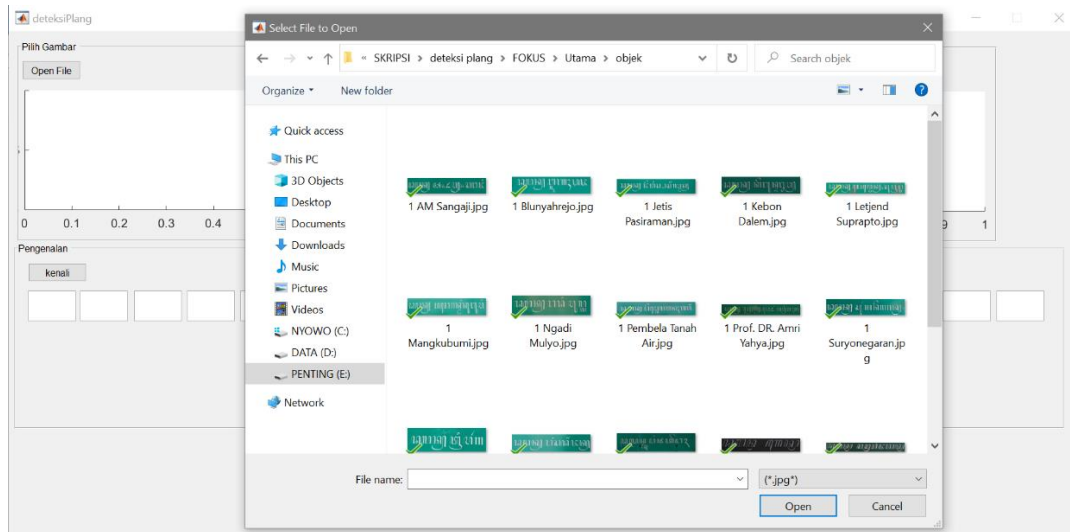
Proses ini dilakukan dengan cara mengklik tombol 'open file' kemudian akan muncul halaman yang berisi kumpulan citra uji yang akan dipilih dan akan diproses oleh sistem, kemudian sistem akan secara otomatis menampilkan citra yang dipilih. Kode program pada tombol 'open file' dapat dilihat pada Gambar 4.24 dan antarmuka sistem dilihat pada Gambar 4.25.

```
[filename,pathname] = uigetfile('*.jpg*');

if ~isequal(filename,0)

    img = imread(fullfile(pathname,filename));
    [~,~,m] = size(img);
    if m == 3
        axes(handles.axes1)
        nama = filename
        nama = sprintf(nama)
        handles.nama.String = nama;
        imshow(img)
        handles.img = img;
        guidata(hObject, handles)
    else
        msgbox('Please insert RGB Image')
    end
else
end
```

Gambar 4. 24



(a)



(b)

Gambar 4. 25 (a) mencari input citra (b) Menampilkan antarmuka input

4.2.2 Deteksi Citra

Proses deteksi citra berjalan ketika tombol 'deteksi' ditekan akan menampilkan citra yang telah diproses sebelumnya sampai proses *thining*. Kode program pada tombol 'deteksi' dapat dilihat pada Gambar 4.26 dan antarmuka hasil deteksi bisa dilihat pada Gambar 4.27.

```
img = handles.img;
Img_gray = rgb2gray(img);
bw = im2bw(Img_gray,graythresh(Img_gray));
bw = bwareaopen(bw, 20);
bw = ~bw;
```

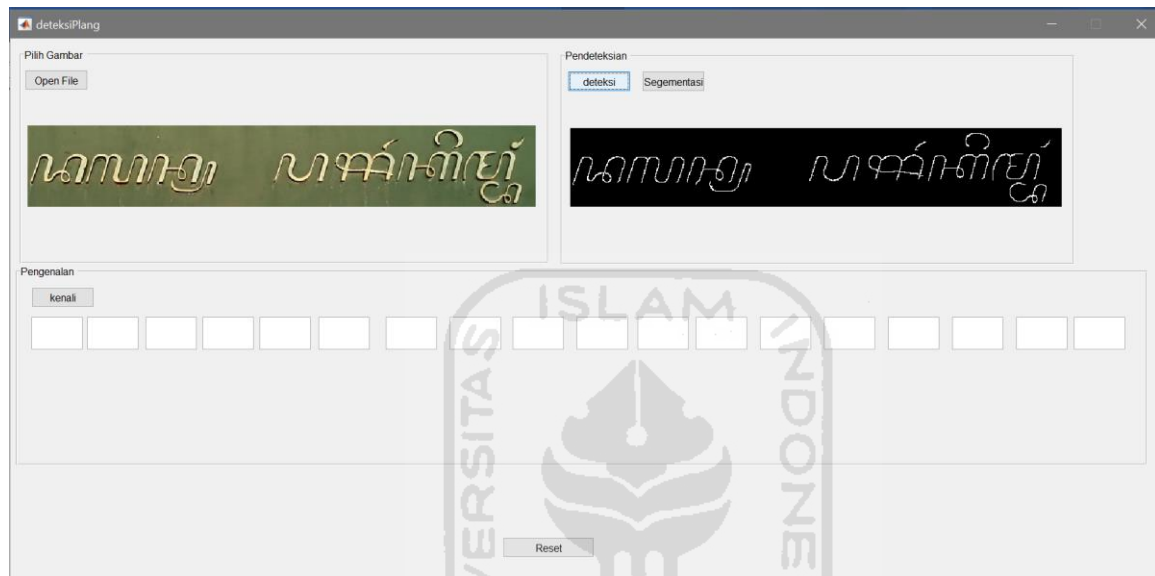


```

tipis = bwmorph(bw, 'thicken', inf);
tipis = ~tipis;
% tipis = bwareaopen(tipis, 20);
axes(handles.axes2)
imshow(tipis)
handles.tipis = tipis;
guidata(hObject, handles)

```

Gambar 4. 26 Kode program tombol ‘deteksi’



Gambar 4. 27 Hasil proses deteksi

4.2.3 Segmentasi Citra

Proses segmentasi dilakukan ketika tombol segmentasi diklik dan kemudian menampilkan citra yang telah terseleksi antar karakter. Kode program pada tombol ‘segmentasi’ bisa dilihat pada Gambar 4.28 dan contoh hasil segmentasi bisa dilihat pada Gambar 4.29.

```

tipis = handles.tipis;
[labeled, numObjects] = bwlabel(tipis,8);
stats = regionprops(labeled, 'BoundingBox');
bbox = cat(1, stats.BoundingBox);

```

```

%menampilkan citra rgb hasil segmentasi
hold on;
for idx = 1 : numObjects
    PosOb = zeros(numObjects,4);
    h = rectangle('Position',bbox(idx,:), 'LineWidth',2);
    posi = get(h, 'Position');
    PosOb(idx,1:4) = posi;
    set(h, 'EdgeColor', [.75 0 0]);
    detec = imcrop(tipis, posi);
    % Figure out where to divide it.
    [rows, columns, numberOfColorChannels] = size(detec);
    r3 = int32(rows/3);
    c3 = int32(columns/3);

    %Extract the 9 images.
    image1 = detec(1:r3, 1:c3);
    image2 = detec(1:r3, c3+1:2*c3);
    image3 = detec(1:r3, 2*c3+1:columns);
    image4 = detec(r3+1:2*r3, 1:c3);
    image5 = detec(r3+1:2*r3, c3+1:2*c3);
    image6 = detec(r3+1:2*r3, 2*c3+1:columns);
    image7 = detec(2*r3+1:end, 1:c3);
    image8 = detec(2*r3+1:end, c3+1:2*c3);
    image9 = detec(2*r3+1:end, 2*c3+1:columns);

    %hitung piksel
    jumlahPixel1 = sum(image1(:));
    jumlahPixel2 = sum(image2(:));
    jumlahPixel3 = sum(image3(:));
    jumlahPixel4 = sum(image4(:));
    jumlahPixel5 = sum(image5(:));
    jumlahPixel6 = sum(image6(:));
    jumlahPixel7 = sum(image7(:));
    jumlahPixel8 = sum(image8(:));
    jumlahPixel9 = sum(image9(:));

    ciriUji(1,idx) = jumlahPixel1;
    ciriUji(2,idx) = jumlahPixel2;
    ciriUji(3,idx) = jumlahPixel3;
    ciriUji(4,idx) = jumlahPixel4;
    ciriUji(5,idx) = jumlahPixel5;
    ciriUji(6,idx) = jumlahPixel6;
    ciriUji(7,idx) = jumlahPixel7;
    ciriUji(8,idx) = jumlahPixel8;
    ciriUji(9,idx) = jumlahPixel9;

    hold on;

```

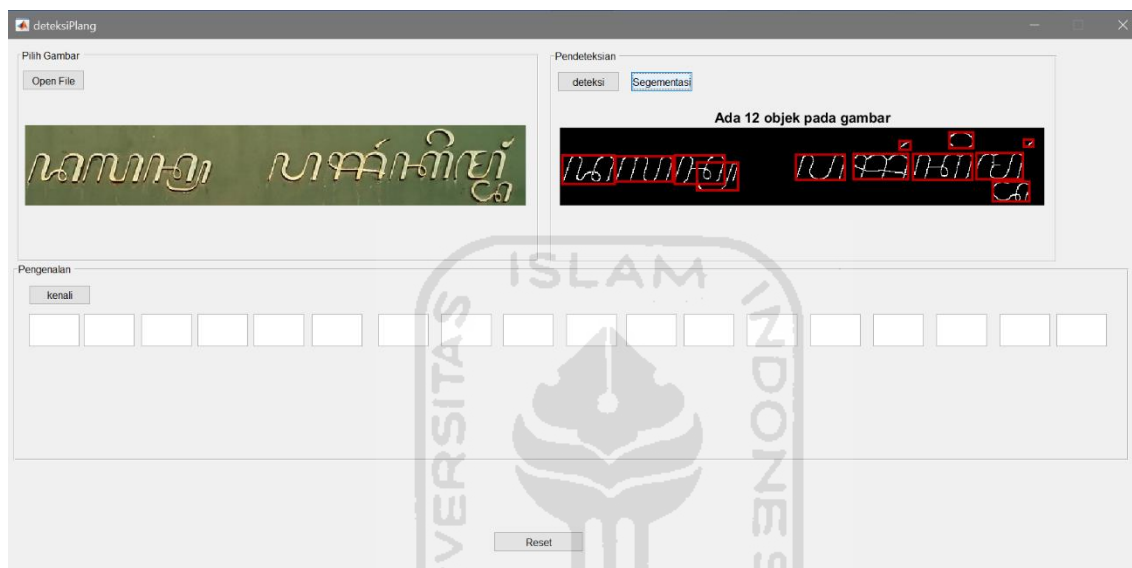
```

end

% menampilkan jumlah objek hasil segmentasi
title(['Ada ', num2str(numObjects), ...
      ' objek pada gambar']);
save ciriUji;
handles.ciriUji = ciriUji;
guidata(hObject, handles);

```

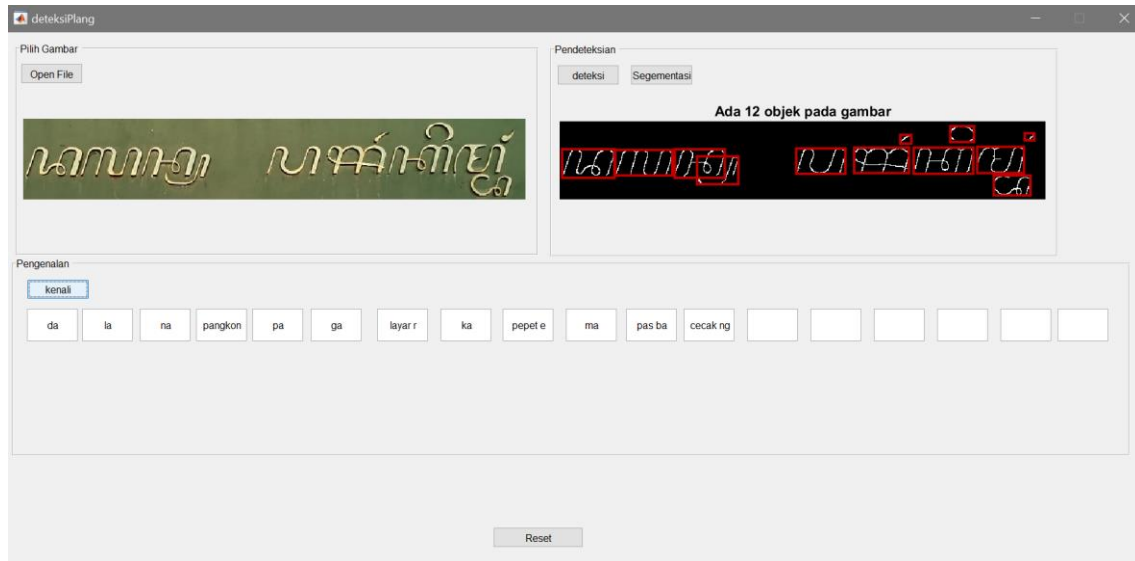
Gambar 4. 28 Kode program tombol ‘segmentasi’



Gambar 4. 29 Hasil proses segmentasi

4.2.4 Pengenalan Citra

Proses pengenalan dilakukan ketika tombol ‘kenali’ diklik dan akan memunculkan huruf latin dari karakter aksara Jawa yang telah dikenali. Pada proses kali ini kode program sama persis dengan Gambar dan contoh hasil tombol ‘kenali’ bisa dilihat pada Gambar 4.30.



Gambar 4. 30 Hasil proses kenali

4.2.5 Reset

Pada proses kali ini ketika tombol diklik akan melakukan reset pada semua nilai yang ada pada bagian antar muka dan menjadikan semuanya seperti sistem baru pertama kalo dibuka. Kode program pada tombol reset dapat dilihat pada Gambar 4.31.

```

cla(handles.axes1);
cla(handles.axes2);
set(handles.edit1, 'String', '');
set(handles.edit2, 'String', '');
set(handles.edit3, 'String', '');
set(handles.edit4, 'String', '');
set(handles.edit5, 'String', '');
set(handles.edit6, 'String', '');
set(handles.edit7, 'String', '');
set(handles.edit8, 'String', '');
set(handles.edit9, 'String', '');
set(handles.edit10, 'String', '');
set(handles.edit11, 'String', '');
set(handles.edit12, 'String', '');
set(handles.edit13, 'String', '');
set(handles.edit14, 'String', '');
set(handles.edit15, 'String', '');
set(handles.edit16, 'String', '');
set(handles.edit17, 'String', '');
set(handles.edit18, 'String', '');
set(handles.nama, 'String', '');
cla('reset');

```

Gambar 4. 31 Kode program pada tombol ‘reset’

4.3 Pengujian

Pengujian dilakukan untuk mengetahui tingkat akurasi program pada setiap data uji citra aksara Jawa pada plang jalan. Pada pengujian kali ini menggunakan metode *black box testing*, perhitungan statistik ketepatan dan *cohen's kappa*.

4.3.1 *Black Box Testing*

Metode dilakukan secara langsung oleh penulis dengan membandingkan hasil satu persatu tiap langkah secara manual pada sistem. Hasil dari pengujian *black box testing* bisa dikatakan sempurna karena semua pengujian dalam metode ini semuanya berhasil. Pengujian *Black Box Testing* secara detail bisa dilihat pada Lampiran 4.1.

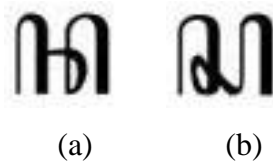
4.3.2 Perhitungan Statistik Ketepatan

Metode penghitungan statistik dilakukan dengan membandingkan hasil output sistem tiap karakter dan membandingkannya secara langsung apakah pendeteksian pada sistem berhasil atau tidak dan menghasilkan nilai akurasi. Hasil pengujian perhitungan statistik ketepatan menunjukkan 88,7%. Pengujian perhitungan statistik dapat dilihat pada Lampiran 4.2.

4.3.3 *Cohen's Kappa*

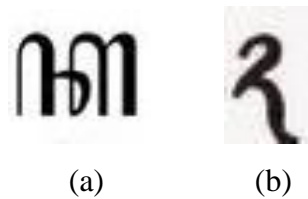
Pengujian *cohen's kappa* yang biasa menggunakan dua kriteria penguji atau lebih, di sini hanya menggunakan hasil dari sistem dan hasil yang sebenarnya. Pengujian dengan metode ini dilakukan untuk menentukan multi kategori baik data yang tepat dan tidak tepat dapat. Pengujian *cohen's kappa* menunjukkan hasil dengan nilai 0.872 yang berarti sistem yang diusulkan tergolong sangat bagus dengan keadaan sebenarnya karena indeks ketepatan bernilai antara 0.81 - 1.00. Detail dari pengujian *Cohen's Kappa* dapat dilihat pada lampiran 4.3.

Data yang tidak tepat menunjukkan nilai 0,128 yang berarti masih ada data yang meleset dan juga data yang tidak terdeteksi. Pada contoh data yang meleset ada dua karakter “Na” yang dikenali sebagai karakter “Sa” karena pada dua karakter tersebut sangatlah mirip hanya berbeda pada tanda simpul di awal dan di akhir karakter. Kemiripan karakter “Na” dan “Sa” terlihat pada Gambar 4.32.



Gambar 4. 32 Kemiripan Karakter; (a) “Na” (b) “Sa”

Data yang meleset juga terjadi pada 1 karakter “Ka” yang dikenali sebagai karakter “Wigyan”, pada data yang meleset kali ini karakter keduanya sangatlah tidak mirip namun sistem tetap mengenalinya dengan karakter yang berbeda. Kemungkinan kenapa sistem mengenalinya sama karena jumlah piksel dengan intensitas bernilai 1 pada 9 bagian ada kecocokan, karena memang pada penelitian kali ini bukan bentuk yang menjadi kecocokan melainkan jumlah piksel dengan intensitas yang bernilai 1. Contoh perbedaan karakter “Ka” dan karakter “Wigyan” dapat dilihat pada Gambar 4.33.



Gambar 4. 33 Perbedaan karakter; (a) “Ka” (b) “Wigyan”

4.4 Kelebihan Dan Kekurangan Sistem

4.4.1 Kelebihan Sistem

- Sistem dapat menampilkan hampir secara keseluruhan karakter aksara Jawa.
- Pembacaan sistem terhitung sangat cepat.
- Sistem sudah dapat mengenali karakter aksara Jawa dengan baik.

4.4.2 Kekurangan Sistem

- Pendeteksian karakter aksara Jawa yang bergabung dengan karakter lain sistem belum dapat mendeteksinya.
- Ada bagian yang tidak bisa ditampilkan oleh sistem jika karakter aksara Jawa terlalu kecil.
- Sistem belum mampu mengenali semua karakter aksara Jawa.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan semua proses penelitian, maka dapat ditarik kesimpulan sebagai berikut:

1. sistem identifikasi aksara Jawa pada plang jalan telah berhasil dibangun dengan menggunakan aplikasi Matlab R2016b. Berbagai fungsi dan tools bawaan disusun untuk mengimplementasikan metode pengambilan jumlah piksel dengan intensitas bernilai 1 pada tiap bagian pada karakter yang telah dibagi menjadi sembilan bagian atau biasa disebut pemodelan Statistik Spasial.
2. Sistem dapat mengenali tiap karakter aksara Jawa dengan cara mengekstrasi citra tiap karakter menggunakan pemodelan Statistik Spasial kemudian akan dicocokkan dengan data latih menggunakan jaringan saraf tiruan yang dibangun dengan jaringan *feedforward* menggunakan fungsi *newff* yang disediakan oleh *toolbox neural network* pada Matlab.
3. Pemodelan Statistik Spasial atau teknik keruangan guna mengekstrasi ciri dalam karakter aksara Jawa dievaluasi dengan metode perhitungan statistik menunjukkan ketepatan sebesar 88,7%. Selain itu, nilai akurasi tersebut juga dievaluasi dengan matriks *cohen's kappa* untuk mengukur ketepatan data identifikasi pada kasus multi kategori dengan nilai 0.872 yang berarti sistem yang diusulkan tergolong sangat bagus dengan keadaan sebenarnya karena indeks ketepatan bernilai antara 0.81 - 1.00.

5.2 Saran

Bagian akhir pada proses tugas akhir ini, penulis mengajukan saran agar kedepannya apabila ada peneliti lain yang ingin mengembangkan sistem ini agar lebih baik lagi. Saran yang diajukan sebagai berikut:

1. Menambahkan jumlah data latih agar akurasi dalam pengenalan lebih sempurna.
2. Menambahkan kelas dalam data latih agar karakter aksara Jawa yang bisa dikenali lebih banyak lagi.
3. Pengenalan menggunakan plang sebaiknya citra yang digunakan masih berbentuk plang jalan.
4. Mengembangkan sistem dalam platform *mobile* agar tingkat penggunaan sistem lebih mobilitas lagi dan bisa langsung terhubung pada kamera pada platform tersebut.

5. Pembacaan aksara Jawa lebih didetailkan lagi agar orang awam langsung bisa membacanya tanpa harus mengeja.
6. Menambah data uji agar sistem dalam pengenalan aksara Jawa lebih dapat dipercaya.
7. Memperhatikan *UI&UX* agar antarmuka sistem dapat dioperasikan dengan mudah oleh pengguna.

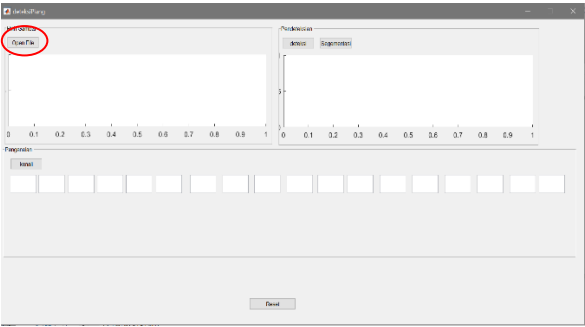
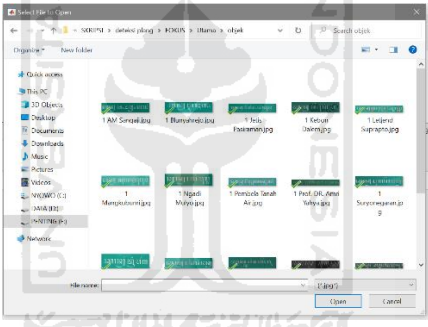
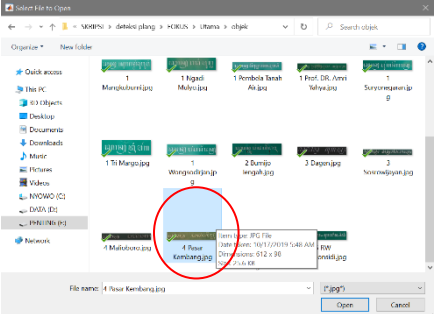
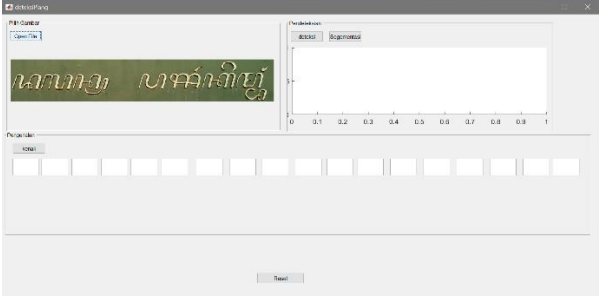


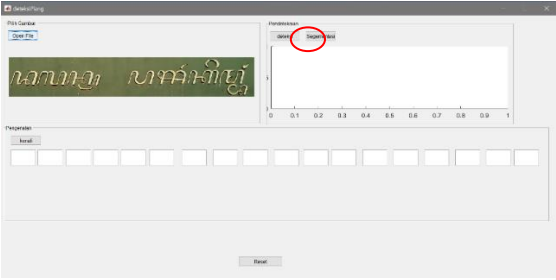
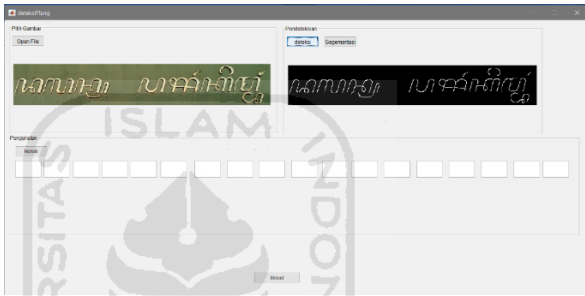
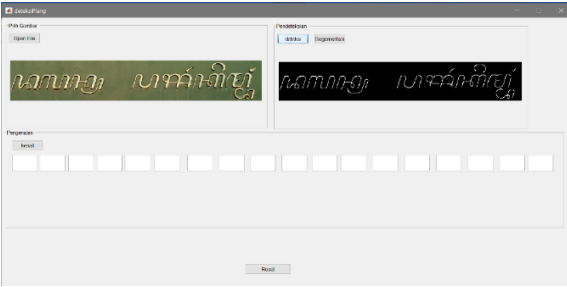
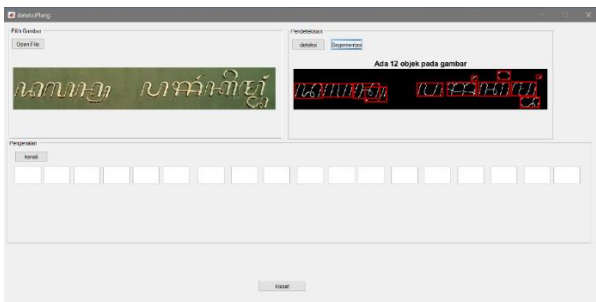
DAFTAR PUSTAKA

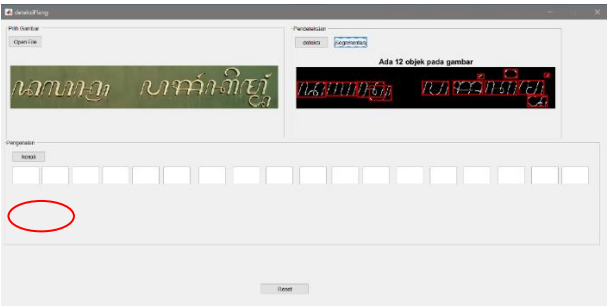
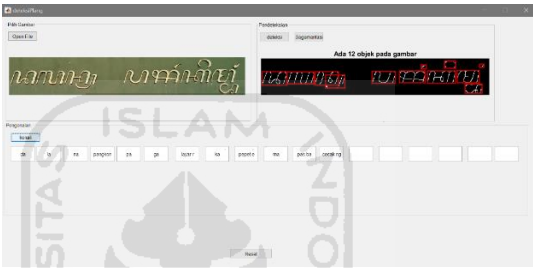
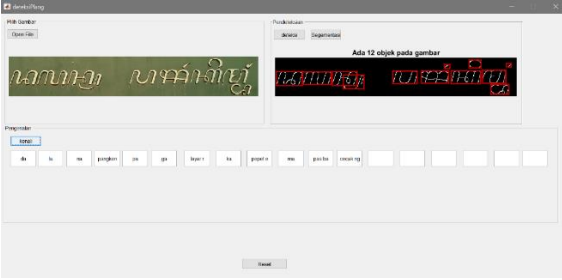
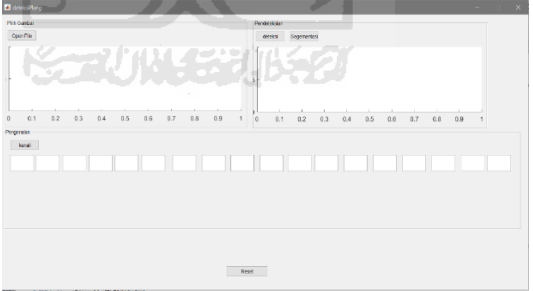
- Arcelli, C., & Di Baja, G. S. (1985). A Width-Independent Fast Thinning Algorithm. *IEEE*.
- Haralick, R. M. (1987). Digital Step Edges from Zero Crossing of Second Directional Derivatives. *Readings in Computer Vision*, 216-226.
- Haruechaiyasak, C., Kongyoung, S., & Dailey, M. (2008). A comparative study on Thai word segmentation approaches. *Computer Science and Information Management Asian Institute of Technology*.
- Hendrik, Anjomshooa, A., & Tjoa, A. M. (2014). Towards Semantic Mashup Tools For Big Data Analysis. *Proceeding of the Information & Communication Technology-EurAsia Conference 2014*, (pp. 100-145). Bali.
- Munir, R. (2004). Pengolahan citra digital dengan pendekatan algoritmik.
- Norbu, S., Choejey, P., Dendup, T., Hussain, S., & Mauz, A. (2010). Dzongkha Word Segmentation. *8th Workshop on Asian Language Resources*.
- Prabantoror, A. (2014). Segmentasi Aksara Jawa.
- Scott, L. M., & Warmerdam, N. (2006). Spatial Statistics for Public Health and Safety. *ESRI*.
- Setiawan, A. M. (2013). *Integrated Framework For Business Process Complexity Analysis*. Retrieved from ECIS 2013 Completed Research: http://aisel.aisnet.org/ecis2013_cr/49
- Tanaya, D., & Adriani, M. (2016). Dictionary-based Word Segmentation for Javanese. *5th Workshop on Spoken Language Technology for Under-resourced Languages, SLTU 2016*, 6.
- Taufiq, H. (2015). *Argumentasi dan Validitas*. Yogyakarta: Darqin.
- Wahid, F. (2014). The Antecedents And Impacts of a Green Eprocurement Infrastructure: Evidence From The Indonesian Public Sector. *International Journal of internet Protocol Technology*, 7(4), 210-218.
- Widiarti, A. R. (2006). Pengenalan citra Dokumen Sastra Jawa Konsep Dan Implementasinya. 1-2.
- Zhangrila, L. L. (2018). Accuracy Level of \$P Algorithm for Javanese Script Detection on. *3rd International Conference on Computer Science and Computational Intelligence 2018*, 9.
- Zukhri, Z. (2014). *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Andi Publisher.

LAMPIRAN

Lampiran 4.1 Pengujian Metode *Black Box Testing*.

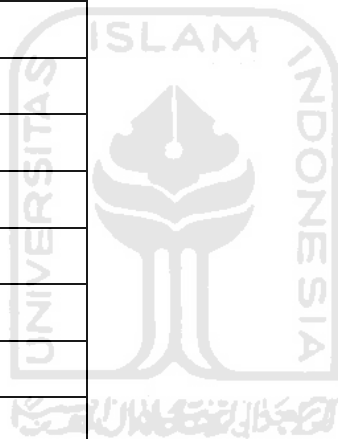
No	Skenario Pengujian	Hasil Yang Diinginkan	Kesimpulan
1	<p>Memilih data uji berupa citra bertipe jpg dengan cara klik kiri pada tombol 'open file'.</p> <p><i>Test case :</i></p> 	<p>Sistem akan muncul window yang untuk mencari file input berupa citra bertipe jpg.</p> <p>Hasil pengujian :</p> 	<p>Sistem berjalan sesuai dengan yang diinginkan.</p>
2	<p>Menampilkan citra yang berhasil dipilih ketika pengguna telah memilih citra.</p> <p><i>Test case :</i></p> 	<p>Sistem akan menampilkan citra yang telah berhasil dipilih.</p> <p>Hasil pengujian :</p> 	<p>Sistem berjalan sesuai dengan yang diinginkan.</p>

<p>3 Menampilkan pendeteksian karakter aksara Jawa dengan menggunakan metode sebelumnya sampai proses <i>thinging</i> dengan mengklik tombol deteksi.</p> <p><i>Test case :</i></p> 	<p>Sistem menampilkan citra baru yang telah melalui proses <i>thinging</i>.</p> <p>Hasil pengujian :</p> 
<p>4 Menampilkan proses segmentasi dan pelabelan pada tiap karakter aksara Jawa yang terpisah dengan menandai kotak merah tiap karakter, proses ini akan terjadi jika tombol segmentasi diklik.</p> <p><i>Test case :</i></p> 	<p>Sistem menampilkan proses segmentasi dan pelabelan pada tiap karakter</p> <p>Hasil pengujian :</p> 

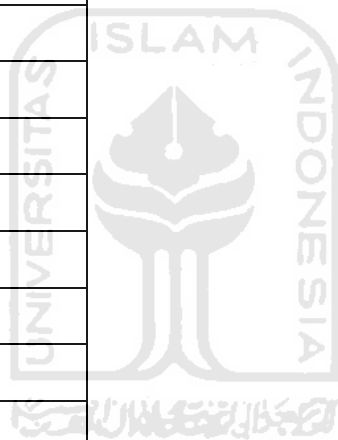
<p>5</p>	<p>Ketika tombol kenali diklik akan menampilkan proses pengenalan tiap karakter dan mengubahnya ke dalam duruf latin. <i>Test case:</i></p> 	<p>Sistem menampilkan pengenalan tiap karakter aksara Jawa dan mengubahnya ke dalam huruf latin. Hasil pengujian :</p> 	<p>Sistem berjalan sesuai dengan yang diinginkan.</p>
<p>6</p>	<p>Ketika tombol 'reset' diklik sistem akan mereset semua tampilan dan mengembalikannya seperti semula. <i>Test case :</i></p> 	<p>Sistem melakukan reset semua tampilan sehingga seperti semula Hasil pengujian :</p> 	<p>Sistem berjalan sesuai dengan yang diinginkan</p>

Lampiran 4.2 Pengujian Statistik Ketepatan

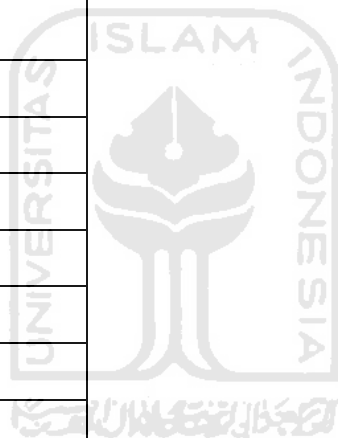
No	Citra plang jalan aksara Jawa			
	Nama file	Aksara Jawa	Tepat	Tidak Tepat
1	AM Sangaji.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Swara F a	1	0
		Swara B a	1	0
		Titik	1	0
		Titik	1	0
		Swara e	1	0
		Ma	1	0
		Pangkon	1	0
		Titik	1	0
		Titik	1	0
		Sa	1	0
		Nga F	1	0
		Nga B	1	0
		Ja	1	0
wulu	1	0		



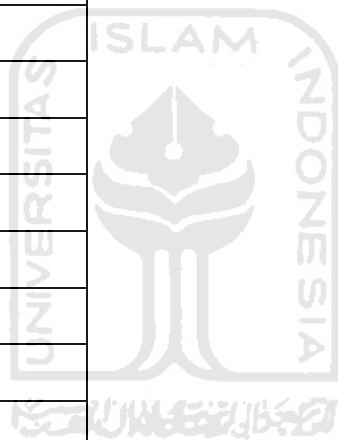
2	Blunyahrejo	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Ba F	1	0
		Ba B	1	0
		Pasangan La	1	0
		Nya F	1	0
		Nya B	1	0
		Wigyan	1	0
		Ceret	1	0
3	Bumijo Tengah.jpg	Ja	1	0
		Da	1	0
		La	1	0
		Na	1	0
		Pasangan Ba	1	0
		Ma	1	0
		Wulu	1	0
		Ja	1	0
		Ta	1	0
		Pepet	1	0
Nga F	1	0		



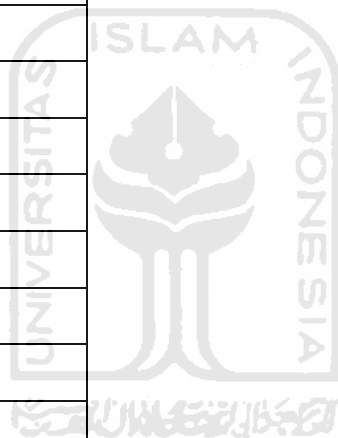
		Nga B	1	0
		Wigyan	1	0
4	Dagen.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pasangan Dha	0	1
		Taling	1	0
		Ga	1	0
		Na	1	0
		Pangkon	1	0
5	Jetis Pasiraman.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Ja	1	0
		Pepet	1	0
		Ta	1	0
		Wulu	1	0
		Sa	1	0
		Pasangan Pa	1	0
sa	1	0		



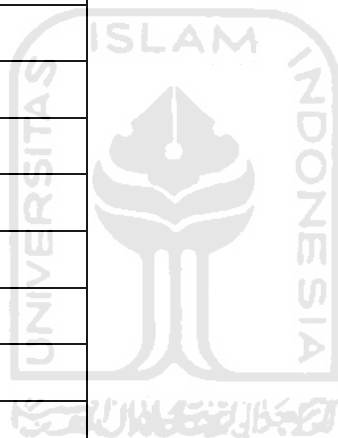
		wulu	1	0
		Layar	1	0
		Ra	1	0
		Ma	1	0
		Pas Ma	1	0
		Na	1	0
		Pangkon	1	0
6	Kebon Dalem.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Ka	1	0
		Pepet	1	0
		Ba F	0	1
		Ba B Suku	0	1
		Na	1	0
		Pas Da	1	0
		Nga F	1	0
		Nga Lelet B	0	1
		Ma	1	0
		Pangkon	1	0
7	Letjend Suprpto	Da	1	0



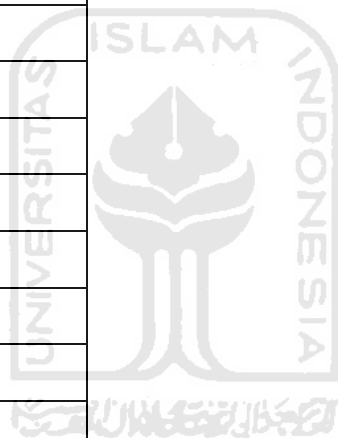
		La	1	0
		Na	1	0
		Pangkon	1	0
		Taling	1	0
		La	1	0
		Taling	1	0
		Ta	1	0
		Pas Ja	1	0
		Na	1	0
		Pangkon	1	0
		Titik	1	0
		Sa Suku	0	1
		Pa Cakra	0	1
		Pa	1	0
		Pas ta	1	0
8	Malioboro.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Ma	1	0
		La	1	0
		Wulu	1	0



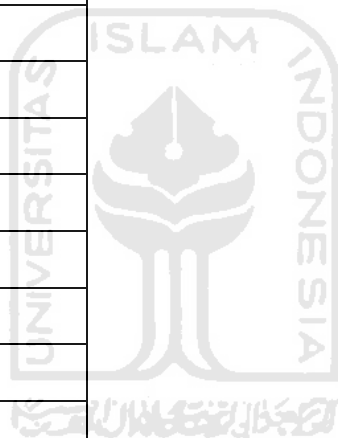
		Ya	1	0
		Ba F	0	1
		Ba b	1	0
		Ra	1	0
9	Mangkubumi.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Pa	1	0
		Taling	1	0
		Nga F	1	0
		Nga B	1	0
		Ra	1	0
		Na	1	0
		Pasa Ma	0	1
		Cecak	0	1
		Ka Suku	0	1
		Ba F	1	0
		BA B Suku	0	1
Ma	0	1		
wulu	1	0		
10	Ngadi Mulyo.jpg	Da	1	0



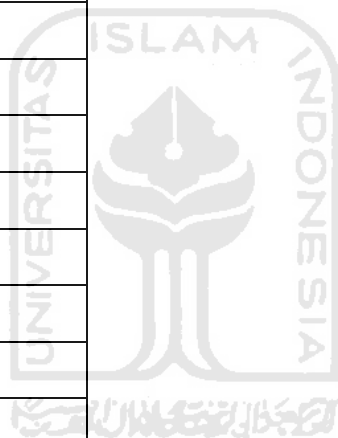
		La	1	0
		Na	1	0
		Pangkon	1	0
		Nga F	1	0
		Nga B	1	0
		Da	1	0
		Wulu	1	0
		Ma Suku	0	1
		La	1	0
		Pas Ya	1	0
11	Pasar Kembang.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Pa	1	0
		Murda Sa	1	0
		Layar	1	0
		Ka	1	0
		Pepet	1	0
		Ma	1	0
		Pas Ba	1	0
		Cecak	1	0



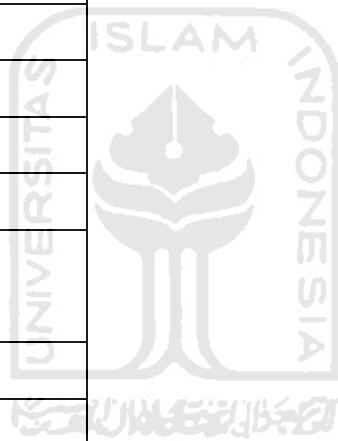
12	Pembela Tanah Air.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Pa	1	0
		Pepet	0	1
		Taling	1	0
		Ma	1	0
		Pas Ba	1	0
		La	1	0
		Ta	1	0
		Na	1	0
		Wigyan	1	0
		Ha	1	0
		Ha	1	0
Wulu	1	0		
Layar	1	0		
13	Prof. DR. Amri Yahya.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0



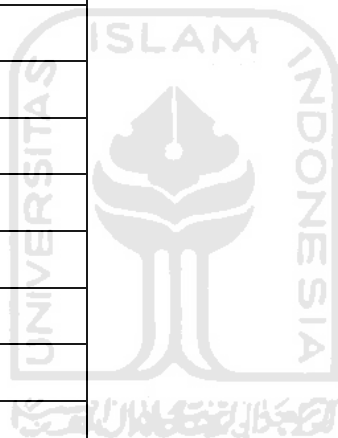
	Taling	0	1
	Tarung F		
	Pa cakra	1	0
	Taling	0	1
	Tarung F		
	Pa	1	0
	Pangkon	1	0
	F	0	1
	F	0	1
	F	0	1
	titik	1	0
	Taling	1	0
	Dha	1	0
	Swara E	1	0
	Layar	1	0
	Titik	1	0
	Ha	1	0
	Ma Cakra	0	1
	Wulu	1	0
	Ya	1	0
	Wigyan	1	0
	Ya	1	0



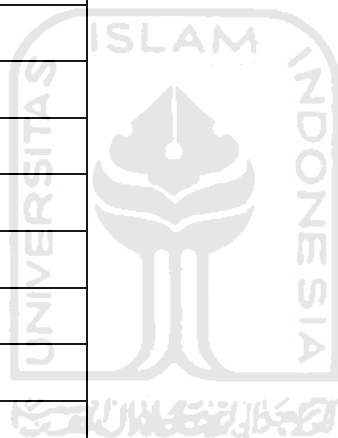
14	RW Mongonsidi.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Swara E	1	0
		Layar	1	0
		Taling	1	0
		Wa	1	0
		Titik	1	0
		Titik	1	0
		Taling Tarung F	1	0
		Ma	1	0
		Taling Tarung B	1	0
		Nga F	1	0
		Wulu	1	0
		Nga B	1	0
		Na	1	0
		Pas Sa	1	0
		Wulu	1	0
		Da	1	0



		Wulu	1	0
15	Sosrokusuman.jpg	Ga Murda	1	0
		Cecak	0	1
		Sa Murda	1	0
		Sa Cakra	0	1
		Ka Suku	0	1
		Sa Suku	0	1
		Ma	1	0
		Na	1	0
		Pangkon	1	0
16	Sosrowijayan.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Sa	1	0
		Sa cakra	0	1
		Wa	1	0
		Wulu	1	0
		Ja	1	0
		Ya	1	0
		Na	1	0
		Pangkon	1	0



17	Suryonegaran.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Sa suku	0	1
		Layar	1	0
		Ya	1	0
		Na	1	0
		Pepet	1	0
		Ga	1	0
		Ra	1	0
		Na	1	0
Pangkon	1	0		
18	Tri Margo.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Ta Cakra	0	1
		Wulu	1	0
		Ma	0	1
		Layar	1	0
		Ga	1	0



19	Wongsodirjan.jpg	Da	1	0
		La	1	0
		Na	1	0
		Pangkon	1	0
		Wa	0	1
		Cecak	0	1
		Sa	1	0
		Da	1	0
		Wulu	1	0
		Layar	1	0
		Ja	1	0
		Na	1	0
		pangkon	1	0
Total		237	30	

Keterangan:

- a) Tepat : Jumlah karakter aksara Jawa yang muncul dan telah terbukti benar.
- b) Tidak Tepat : Jumlah karakter aksara Jawa yang sistem dan nilai asli tidak sama

Hasil akurasi dapat terlihat pada persamaan 0.1.

$$= \frac{Tepat}{Total\ Data} \times 100\%$$

$$= \frac{237}{267} \times 100\% \quad (0.1)$$
$$= 88,7 \%$$



Keterangan :

- Pada tabel baris menjelaskan karakter aksara f yang berhasil dideteksi oleh sistem.
- Pada tabel kolom menjelaskan karakter aksara Jawa yang sebenarnya.

Data yang telah diolah akan dihitung nilai akurasi dan nilai gagalnya, proses pengolahan tersebut dapat dilihat pada persamaan 0.2 dan 0.3.

Pengukuran nilai koefisien kappa :

$$\begin{aligned} &= \frac{\text{Pr}(a) - \text{Pr}(e)}{1 - \text{Pr}(e)} \\ &= \frac{88,7\% - 11,3\%}{1 - 11,3\%} \\ &= \frac{77,4\%}{88,7\%} \\ &= \frac{0,774}{0,887} \\ &= 0,872 \end{aligned}$$

(0.2)



Di mana :

- $\text{Pr}(a)$: Presentase jumlah proporsi diagonal utama.
- $\text{Pr}(e)$: Presentase jumlah proporsi di luar diagonal utama.