

**PEMISAHAN TULISAN ARAB SAMBUNG MENJADI
POLA HURUF HIJAIYAH**

SKRIPSI

Oleh:

Ainatul Mardhiyah

04550069



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MALANG**

Oktober, 2008

PEMISAHAN TULISAN ARAB SAMBUNG MENJADI POLA HURUF HIJAIYAH

SKRIPSI

*Diajukan kepada Fakultas Sains dan Teknologi Universitas Islam Negeri Malang
untuk Memenuhi Salah Satu Persyaratan Guna Memperoleh Gelar
Strata Satu Sarjana Teknik Informatika (S.Kom)*

Oleh:

Ainatul Mardhiyah

04550069



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MALANG**

Oktober 2008

HALAMAN PERSETUJUAN

PEMISAHAN TULISAN ARAB SAMBUNG MENJADI POLA HURUF HIJAIYAH

Skripsi

Oleh:

Ainatul Mardhiyah
NIM. 04550069

Telah disetujui oleh:

Pembimbing I,

Pembimbing II,

Ririn Kusumawati, M. Kom
NIP. 150 368 775

Ach. Naschichudin, M.A
NIP. 150 302 531

Tanggal, 18 Oktober 2008

Mengetahui,
Ketua Jurusan Teknik Informatika

Suhartono, S.Si, M. Kom
NIP. 150 327 241

HALAMAN PENGESAHAN

PEMISAHAN TULISAN ARAB SAMBUNG MENJADI POLA HURUF HIJAIYAH

SKRIPSI

Dipersiapkan dan disusun oleh
Ainatul Mardhiyah (04550069)

Telah Dipertahankan di Depan Dewan Penguji
dan Diutamakan Diterima Sebagai Salah Satu Persyaratan dan
Untuk Memperoleh Gelar Sarjana Teknik Informatika (S.Kom)
Pada 21 Oktober 2008

Susunan Dewan Penguji	Tanda Tangan
1. Penguji Utama <u>Fatchurrochman, M. Kom</u> NIP. 150 368 774	: []
2. Ketua Penguji <u>Muhammad Faisal, M.T</u> NIP. 150 368 776	: []
3. Sekretaris Penguji <u>Ririen Kusumawati, M. Kom</u> NIP. 150 368 775	: []
4. Anggota Penguji <u>Ach. Naschichudin, M.A</u> NIP. 150 302 531	: []

Mengetahui,
Ketua Jurusan Teknik Informatika

Suhartono, S.Si, M. Kom
NIP. 150 327 241

SURAT PERNYATAAN

Dengan ini saya menyatakan, bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan pada suatu perguruan tinggi, dan sepanjang pengetahuan saya, juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 18 Oktober 2008

Penulis,

Ainatul Mardhiyah

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Karya ini saya persembahkan kepada:

Ayah M. Z. Arifin dan Ibu Dra. Nanik Maslihah yang telah mencurahkan kasih sayang, keagungan doa, motivasi, nasehat-nasehat dan segala perhatiannya. Semoga aina bisa menjadi putri bungsu yang dapat membanggakan dan berbakti kepada Ayah dan Ibu kelak. Amin...

Kakak-kakakku; Ach.Fahrurrozi, mbak sandri, dan de'rara cantik, terima kasih dah dipinjami scanner dengan cuma-cuma. Mbak Irma Fahrismaini dan mas Badrus maaf adik gak bisa bantu dengan maksimal. Serta chocoku Khalid Rahman, yang selalu membantu dan mendengarkan semua keluh kesah ai

"Tetap tersenyum walau badai menghadang, pak?!"

Mbah Hj. Fatonah tercinta yang selalu mendoakan cucunya

Semua guru-guruku dari TK hingga Perguruan Tinggi yang dengan ketulusan mendidik dan memberikan ilmunya

Teman2 Lobster Cham & Latif terima kasih atas bantuannya

L5110 dan N2415 terima kasih setiap hari bersedia mengantar penulis menuju medan perang

Emak-emak sahabatku tersayang makjendul_ajenk & makndahlul_indah..

"Di dalam hatiku mengisi kisahku denganmu sahabatku"

Seluruh sahabatku di jurusan Teknik Informatika UIN Malang angkatan 2004 [isna, dzovar, arif+dini, agung, ctur, ndahlo, ivana, afdal, andrew, trie, hatan, mujib, nia, dkk]_Tetep kompak sampai punya cucu rek?!_

Kosan Ampel 15 yu' anik, nisa, dian, bu nyai, norm, mbak siti, junet, mbak lila, bu dewi, riris, yuyun, menyun, melka, dan yu' eva

"Terima kasih cinta, luph u all"

Serta rekan-rekan dan semua pihak yang tidak dapat disebutkan satu persatu, yang telah membantu penulis dari awal kuliah hingga tersusunnya skripsi ini.

Terimakasih..

MOTTO

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَكَذَلِكَ أَنْزَلْنَاهُ قُرْءَانًا عَرَبِيًّا وَصَرَّفْنَا فِيهِ مِنَ الْوَعِيدِ لَعَلَّهُمْ
يَتَّقُونَ أَوْ يُحَدِّثُ لَهُمْ ذِكْرًا

Artinya:

*“Dan Demikianlah kami menurunkan Al Quran dalam bahasa Arab, dan kami
Telah menerangkan dengan berulang kali, di dalamnya sebahagian dari
ancaman, agar mereka bertakwa atau (agar) Al Quran itu menimbulkan
pengajaran bagi mereka”*(Q.S. Thaaha/20 ayat 3)

Sumber: *Al-Qur'an dan Terjemahnya* (Jakarta: Yayasan Penyelenggara
Penterjemah Al-Qur'an), hlm. 476.

KATA PENGANTAR

Dengan menyebut asma Allah Yang Maha Pengasih lagi Maha Penyayang. Penyusun panjatkan puji syukur ke hadirat Allah SWT, yang telah melimpahkan rahmat, taufiq dan inayah-Nya sehingga penulis mampu menyelesaikan skripsi yang berjudul *Pemisahan Tulisan Arab Sambung menjadi Pola Huruf Hijaiyah*. sebagai salah satu persyaratan guna mendapatkan gelar Strata Satu Sarjana Teknik Informatika Universitas Islam Negeri Malang, sesuai dengan waktu yang telah ditentukan.

Shalawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW yang membawa cahaya kebenaran, sehingga mengeluarkan umat manusia dari zaman kegelapan ke masa yang terang benderang dengan agama Islam.

Dalam penyusunan laporan ini tidak lepas dari bantuan berbagai pihak yang telah memberi informasi dan inspirasi, sehingga dapat menyusun dan menyelesaikan skripsi ini. Oleh karena itu penulis mengucapkan terima kasih dengan tulus teriring do'a *jazzakumullah khairan katsiran* kepada:

1. Ayah dan Ibu yang dengan ketulusan membesarkan, mendidik, merawat dan senantiasa mencurahkan segalanya baik tenaga, dukungan maupun iringan do'a yang tiada putusnya.
2. Prof. Dr. Imam Suprayogo, selaku Rektor Universitas Islam Negeri Malang.
3. Prof. Dr. Sutiman Bambang Sumitro, SU. Dsc, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Malang.
4. Suhartono, M. Kom, selaku Ketua Jurusan Teknik Informatika.
5. Ririen Kusumawati, M. Kom dan Ach. Nasichudin, M.A, selaku dosen pembimbing yang dengan kesabarannya memberikan bimbingan dan arahan serta masukan-masukan yang sangat berarti kepada penulis selama penyusunan skripsi ini.
6. Bapak-Ibu Dosen dan seluruh civitas akademik Fakultas Sains dan Teknologi yang telah memberikan ilmu dan kemudahan selama penulis berada di

Fakultas Sains dan Teknologi Universitas Islam Negeri Malang. Pak Fatchur, Pak Faisal, Pak Yaqin, Pak Zainal, Pak Amin, Pak Totok, Pak Syahid, Bu Roro, Bu Hani, Pak 9un, Mbak Indah, Pak Deni, & Pak Wawan ”*matur nuwun sanget dumateng sedaya*”

7. Kakak-kakakku yang telah banyak memberikan doa, motivasi, dan dorongan dalam penyelesaian skripsi ini serta semua saudara di rumah.
8. Teman-teman TI’04 seperjuangan, TI’05, TI’06-08, baik yang sama-sama sedang berjuang maupun yang akan berjuang dan teman-teman kosan
9. Serta seluruh pihak yang telah membantu dalam penyelesaian skripsi ini.

Penulis mengakui bahwa skripsi ini masih banyak kekurangan, kelemahan, dan masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun guna perbaikan ke depan.

Akhirnya semoga karya ini diterima di sisi Allah SWT. dan semoga mendapatkan balasan yang setimpal dari-Nya. Harapan penulis semoga karya tulis ilmiah ini dapat bermanfaat bagi penyusun khususnya, dan para pembaca pada umumnya, untuk dijadikan bahan pertimbangan dalam pengembangan pendidikan Islam ke depan dan dapat memperluas cakrawala ke-Islaman terutama untuk ilmu.

Malang, 18 Oktober 2008

Penulis,

Ainatul Mardhiyah

DAFTAR TABEL

Tabel 2.1	Format citra	11
Tabel 2.2	Daftar Huruf Arab	23
Tabel 2.3	Tabel contoh pengelompokan pola berdasarkan cirinya.....	31
Tabel 3.1	Tabel Pengenalan	64
Tabel 3.2	Tabel Pelatihan.....	64
Tabel 4.1	Lingkungan uji coba	71
Tabel 4.2	Hasil kode huruf ain yang telah di simpan dalam database	88
Tabel 4.3	Hasil kode input huruf ain posisi tulisan di depan (<i>start</i>).....	89
Tabel 4.4	Hasil uji coba ketika posisi tulisan di depan (<i>start</i>)	90
Tabel 4.5	Hasil uji coba ketika posisi tulisan di tengah (<i>middle</i>)	91
Tabel 4.6	Hasil uji coba ketika posisi tulisan di akhir (<i>end</i>).....	92



DAFTAR GAMBAR

Gambar 2.1	Diagram Pengolahan Citra	11
Gambar 2.2	Koordinat titik pojok bagian citra yang akan di crop.....	16
Gambar 2.3	Notasi piksel yang digunakan untuk memeriksa keterhubungan..	17
Gambar 2.4	Bentuk dasar huruf Arab berdasarkan kelasnya.....	22
Gambar 2.5	Contoh diagram proses metode statistik.....	36
Gambar 2.6	Arah untuk (a) 4-tetangga (b) 8-tetangga.....	37
Gambar 2.7	Contoh diagram proses <i>template matching</i>	39
Gambar 2.8	Bagian-bagian IDE Delphi	42
Gambar 2.9	Form Designer.....	43
Gambar 2.10	Object TreeView pada IDE Delphi	44
Gambar 3.1	Diagram alir sistem secara keseluruhan	51
Gambar 3.2	Diagram Perancangan Sistem.....	53
Gambar 3.3	Flowchart proses pengenalan tulisan Arab	54
Gambar 3.4	Diagram <i>preprocessing</i>	56
Gambar 3.5	Flowchart proses binerisasi	58
Gambar 3.6	Flowchart proses <i>cropping</i>	60
Gambar 3.7	Nilai Keabuan citra asal pada operasi geometri	61
Gambar 3.8	Flowchart proses pengenalan	66
Gambar 3.9	Form Utama.....	67
Gambar 3.10	Form Proses Awal	68
Gambar 3.11	Form Input Data	69
Gambar 3.12	Form Simpan	70
Gambar 4.1	Form Utama.....	73
Gambar 4.2	Form Image Process	74
Gambar 4.3	Contoh Input Image kata ' <i>aina</i> '	75
Gambar 4.4	Flowchart proses <i>thinning</i>	76
Gambar 4.5	Hasil proses <i>cropping</i>	79
Gambar 4.6	(a) Citra awal (b) hasil <i>scalling</i>	81
Gambar 4.7	Hasil proses pengkodean menggunakan integral proyeksi	83
Gambar 4.8	Form Input data	85
Gambar 4.9	Form Simpan	86
Gambar 4.10	Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf <i>Ain</i> di awal	93
Gambar 4.11	Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf <i>ain</i> di tengah.....	94
Gambar 4.12	Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf <i>ain</i> di akhir	95
Gambar 4.13	Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf <i>ya</i> di tengah	96
Gambar 4.14	Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf <i>dal</i>	97

Gambar 4.15	Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf <i>meem</i> di tengah	97
Gambar 4.16	Hasil ketika data belum ada dalam database	98
Gambar 4.17	Hasil ketika data salah memasukkan nama huruf pada proses simpan ke dalam database	99



DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERSEMBAHAN.....	vi
MOTTO	vii
KATA PENGANTAR.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR ISI.....	xiii
ABSTRAK	xvi
BAB I PENDAHULUAN.....	1
A. LATAR BELAKANG	1
B. RUMUSAN MASALAH.....	5
C. TUJUAN	5
D. BATASAN MASALAH.....	5
E. MANFAAT PENELITIAN.....	6
F. METODOLOGI PENELITIAN	7
G. SISTEMATIKA PENULISAN SKRIPSI.....	8
BAB II TINJAUAN PUSTAKA.....	10
A. TEORI DASAR PENGOLAHAN CITRA.....	10
1. Format Citra	11
a. Citra Biner	12
b. Citra Skala Keabuan.....	12
c. Citra Warna	12
d. Citra Warna Berindeks.....	13
2. Operasi Pengolahan Citra.....	13
a. Pengambilan gambar (<i>Image Tresholding</i>)	14
b. Pemisahan Karakter (<i>Segmentation</i>).....	15
c. Penipisan (<i>Thinning</i>).....	16
d. Penskalaan (<i>Scaling</i>).....	19
B. BAHASA ARAB	20
1. Pengenalan Karakter Huruf Arab.....	20
2. Keistimewaan Mempelajari Bahasa Arab Dalam Perspektif Islam ...	24

C. PENGENALAN POLA	31
1. Pengertian Pola dan Ciri.....	31
2. Tipe dari sistem pengenalan tulisan tangan	34
a. Pengenalan secara <i>offline</i>	34
b. Pengenalan secara <i>online</i>	34
3. Metode Pengenalan	35
a. <i>Statistical</i>	36
b. Model Matching.....	38
c. Model Structural dan Sintactical.....	40
D. BORLAND DELPHI.....	41
1. Mengenal IDE Delphi	41
2. Struktur File	45
3. Code Editor	46
E. FLOWCHART.....	48
BAB III DESAIN DAN PERANCANGAN	50
A. DESKRIPSI SISTEM.....	50
B. DESAIN SISTEM.....	52
1. Desain Data	53
2. Data Masukan.....	53
3. Data Selama Proses	53
C. PENGENALAN TULISAN ARAB.....	54
1. Input image.....	55
2. <i>Preprocessing</i>	56
a. Binerisasi.....	56
b. <i>Thinning</i>	58
c. <i>Segmentation</i>	59
d. <i>Scaling</i>	61
3. Pengkodean Karakter (<i>Feature extraction</i>).....	63
4. Input Data.....	63
5. Tabel.....	64
6. Pengenalan (<i>Recognize</i>)	65
D. DESAIN ANTAR MUKA.....	67
1. Desain Menu Tampilan Utama	67
2. Desain Menu Tampilan proses awal	68
3. Desain Menu Tampilan Halaman Input Data.....	69
4. Desain Menu Tampilan Halaman Simpan	70
BAB IV HASIL DAN PEMBAHASAN	71
A. LINGKUNGAN UJI COBA.....	71
B. DATA UJI COBA.....	72

C. PENJELASAN PROGRAM.....	72
1. Proses Menampilkan Halaman Utama	73
2. Proses Menampilkan Halaman Proses Awal	74
a. Input Image	75
b. Preprocessing	75
c. Pengkodean Karakter	82
d. Pengenalan (<i>Recognize</i>)	84
3. Proses Menampilkan Halaman Input data	85
4. Proses Menampilkan Halaman Simpan	86
D. PEMBAHASAN DATA HASIL UJI COBA	88
1. Hasil Uji Coba.....	88
a. Tulisan Arab berhasil dikenali	92
b. Tulisan Arab tidak berhasil dikenali	98
2. Kontribusi Program Dalam Islam	100
BAB V KESIMPULAN	101
A. KESIMPULAN.....	101
B. SARAN.....	102
DAFTAR PUSTAKA	103

ABSTRAK

Mardhiyah, Ainatul. *Pemisahan Tulisan Arab Sambung menjadi Pola huruf Hijaiyah*. Skripsi, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Malang. Ririen Kusumawati, M. Kom dan Ach. Naschichudin, M.A.

Akhir-akhir ini pengolahan citra digital di banyak negara maju menjadi bidang yang digeluti oleh banyak peneliti karena menarik untuk diterapkan pada berbagai kegiatan, baik kegiatan analisis maupun produksi. Berbagai aspek penelitian dan algoritma pengolahan citra diterapkan dalam bidang kedokteran, biologi, industri, dan bidang yang lain. Untuk keperluan analisis, pengolahan citra mampu melakukan analisis citra digital dalam menentukan pola sidik jari, pengenalan pola tulisan tangan, perhitungan koloni suatu mikroba, ataupun memperbaiki citra. Semakin berkembangnya teknologi, setiap orang semakin berpikiran serba praktis dan otomatis. Bahasa Arab merupakan bahasa yang penting bagi umat Islam, karena al-Quran sebagai pedoman hidup umat muslim ditulis menggunakan bahasa Arab. Tidak semua umat muslim mengetahui pola bentukan dari tulisan Arab sambung, maka dibutuhkanlah suatu aplikasi yang dapat memisahkan tulisan Arab sambung menjadi pola huruf pembentuknya atau huruf hijaiyah

Tujuan dari penelitian ini adalah untuk merancang dan membuat aplikasi yang dapat digunakan sebagai pemisah tulisan Arab sambung menjadi huruf dasarnya. Pembuatan aplikasi ini menggunakan pemrograman Delphi 7 dan Microsoft Access sebagai basis data. Metode pengenalan yang digunakan adalah *Template matching*, yaitu proses pengenalan karakter dengan cara membandingkan karakter yang ada pada dokumen dengan karakter pada *template*, untuk mencari karakter yang memiliki nilai kemiripan terbesar dengan *template*. Citra masukan diperoleh dari hasil *scanning* dan selanjutnya dilakukkn proses awal agar *noise* atau gangguan-gangguan lain pada citra bisa dikurangi.

Dari hasil uji coba perangkat lunak yang telah dilakukan menunjukkan bahwa perangkat lunak yang dibuat mampu memisahkan tulisan Arab sambung menjadi pola dasar pembentuknya, yaitu kembali ke huruf awal sebelum tulisan tersebut disambung. Uji coba dilakukan sebanyak 10 kali dengan tulisan yang sama, presentase tulisan Arab sambung yang dapat dipisah menjadi pola huruf pembentuknya yaitu 44% untuk huruf Arab yang ketika disambung letaknya di awal (*start*), 38% di tengah (*middle*), dan 52% di akhir (*end*). Proses pemisahan (*segmentation*) tulisan menjadi per karakter yang dilakukan pada proses awal menjadi hal yang penting untuk mencirikan masing-masing karakter dalam mengenali tulisan. Semakin berbeda cara seseorang melakukan pemisahan dari suatu tulisan Arab sambung, maka semakin banyak kode tulisan per karakter yang harus disimpan dalam database sebagai ciri pembeda.

Kata kunci: Pengenalan, Citra, Tulisan Arab, *Template Matching*

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Keberadaan manusia di muka bumi ini sebagai *khalifah fil ardh* (penguasa di bumi) mempunyai tugas untuk memelihara, melestarikan, serta membudayakan semua ciptaan Allah yang berada di bumi. Tidak dibenarkan untuk melakukan perusakan terhadap semua ciptaan Illahi. Dengan ilmu pengetahuan dan teknologi, manusia dapat melaksanakan tugas kekhalifahan tersebut. Dalam mencari ilmu, manusia harus selalu berkeyakinan bahwa Allah adalah Sang Pencipta. Sesuai dengan firman Allah SWT, dalam al-Quran surat Al-Baqarah/2 : 255 di bawah ini:

اللَّهُ لَا إِلَهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ لَا تَأْخُذُهُ سِنَّةٌ وَلَا نَوْمٌ لَهُ مَا فِي السَّمَوَاتِ وَمَا فِي الْأَرْضِ مَنْ ذَا الَّذِي يَشْفَعُ عِنْدَهُ إِلَّا بِإِذْنِهِ يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ وَلَا يُحِيطُونَ بِشَيْءٍ مِّنْ عِلْمِهِ إِلَّا بِمَا شَاءَ وَسِعَ كُرْسِيُّهُ السَّمَوَاتِ وَالْأَرْضَ وَلَا يَئُودُهُ حِفْظُهُمَا وَهُوَ الْعَلِيُّ الْعَظِيمُ

Artinya:

“Allah, tidak ada Tuhan (yang berhak disembah) melainkan dia yang hidup kekal lagi terus menerus mengurus (makhluk-Nya); tidak mengantuk dan tidak tidur. Kepunyaan-Nya apa yang di langit dan di bumi. tiada yang dapat memberi syafa’at di sisi Allah tanpa izin-Nya? Allah mengetahui apa-apa yang di hadapan Allah melainkan apa yang dikehendaki-Nya. Kursi¹ Allah meliputi langit dan

¹ Kursi dalam ayat ini oleh sebagian mufasssirin diartikan dengan ilmu Allah dan ada pula yang mengartikan dengan kekuasaan-Nya

bumi. dan Allah tidak merasa berat memelihara keduanya, dan Allah Maha Tinggi lagi Maha besar.” (QS. Al-Baqarah/2 ayat 255)

Disamping sebagai khalifah di muka bumi umat Islam juga harus berdakwah kepada umat yang lain. Di zaman teknologi yang semakin berkembang seperti sekarang ini, banyak cara yang bisa dilakukan dalam berdakwah oleh seorang muslim, baik secara sembunyi-sembunyi ataupun terang-terangan. Beberapa metode dakwah Islam telah dikembangkan di berbagai bidang, misalnya saja dalam bidang pendidikan. Universitas Islam Negeri Malang sebagai lembaga pendidikan yang bernuansa Islami telah menyediakan jurusan Teknik Informatika sebagai salah satu program studi barunya. Hal tersebut dimaksudkan agar seorang Sarjana Teknik bisa menjadi ulama yang berfikir dengan mengkorelasikan al-Quran dan ilmu pengetahuan yaitu menghubungkan ilmu agama dengan program studi tersebut. Anjuran untuk selalu berfikir di dalam segala macam kegiatan telah diterangkan dalam al-Quran surat Al-Imron/3 : 191 yang berbunyi sebagai berikut:

الَّذِينَ يَذْكُرُونَ اللَّهَ قِيَمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ وَيَتَفَكَّرُونَ فِي خَلْقِ السَّمَوَاتِ
وَالْأَرْضِ رَبَّنَا مَا خَلَقْتَ هَذَا بَطْلًا سُبْحَانَكَ فَقِنَا عَذَابَ النَّارِ ﴿١٩١﴾

Artinya

“(yaitu) orang-orang yang mengingat Allah sambil berdiri atau duduk atau dalam keadan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata): “Ya Tuhan kami, tiadalah Engkau menciptakan Ini dengan sia-sia, Maha Suci Engkau, Maka peliharalah kami dari siksa neraka.” (QS. al-Imron/3: 191)

Kemajuan ilmu pengetahuan dan teknologi ternyata telah banyak membawa perubahan bagi kehidupan manusia, baik dalam cara berpikir, sikap, gaya hidup atau tingkah laku. Perkembangan teknologi komputer menjadikan hidup lebih fleksibel, mudah menjangkau daerah-daerah yang luas, dan otomatisasi di segala bidang. Informatika sebagai salah satu bidang dari ilmu pengetahuan dan teknologi yang berkembang dengan cepat dimaksudkan agar dapat berperan membantu manusia memperoleh kehidupan yang lebih sempurna sesuai dengan syariat yang ditentukan oleh Allah dan Rasulullah SAW.

Al-Quran sebagai pedoman hidup umat Islam telah diturunkan oleh Allah dengan menggunakan bahasa Arab. Tulisan Arab cenderung disambung dan tidak semua orang bisa tepat menyambungkan huruf-huruf hijaiyah tersebut menjadi tulisan Arab yang memiliki arti. Salah pasangan ketika menyambung huruf hijaiyah menimbulkan perbedaan pada bacaan dan arti. Pemahaman dan kehati-hatian terhadap penulisan tulisan Arab terutama ayat-ayat al-Quran. Sesuai dengan firman Allah dalam al-Quran surat Az-Zukhruf/43 : 113

إِنَّا جَعَلْنَاهُ قُرْءَانًا عَرَبِيًّا لَّعَلَّكُمْ تَعْقِلُونَ

Artinya:

“Sesungguhnya kami menjadikan Al Quran dalam bahasa Arab supaya kamu memahaminya”. (Q.S Az-Zukhruf /43 ayat: 113)

Dan pada ayat yang lain juga dijelaskan bahwa al-Quran diturunkan dengan bahasa Arab bukan berarti hanya untuk orang Arab saja melainkan untuk semua umat di dunia ini yang bisa dijadikan pengajaran

وَكَذَلِكَ أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا وَصَرَّفْنَا فِيهِ مِنَ الْوَعِيدِ لَعَلَّهُمْ يَتَّقُونَ أَوْ يُحَدِّثُ لَهُمْ

ذِكْرًا

Artinya:

“Dan Demikianlah kami menurunkan Al Quran dalam bahasa Arab, dan kami Telah menerangkan dengan berulang kali, di dalamnya sebahagian dari ancaman, agar mereka bertakwa atau (agar) Al Quran itu menimbulkan pengajaran bagi mereka” (Q.S. Thaaha/20 ayat 3)

Akhir-akhir ini pengolahan citra digital di banyak negara maju menjadi bidang yang digeluti oleh banyak peneliti karena menarik untuk diterapkan pada berbagai kegiatan, baik kegiatan analisis maupun produksi. Berbagai aspek penelitian dan algoritma pengolahan citra diterapkan dalam bidang kedokteran, biologi, industri, dan bidang yang lain. Untuk keperluan analisis, pengolahan citra mampu melakukan analisis citra digital dalam menentukan pola sidik jari, pengenalan pola tulisan tangan, perhitungan koloni suatu mikroba, ataupun memperbaiki citra.

Pemisahan tulisan Arab sambung merupakan proses memotong dan mengenali suatu objek dalam citra dengan menyalin menjadi format digital, sehingga dapat mengambil keputusan yang menyamai atau hampir sama dengan kemampuan manusia, yaitu mengenali karakter masukan. Menurut pengamatan yang dilakukan oleh peneliti, masih sedikit mahasiswa Indonesia yang mengadakan riset pengenalan tulisan Arab karena mereka menganggap tulisan Arab memiliki huruf yang beraneka ragam bentuknya terutama ketika huruf tersebut disambung dengan huruf lain. Selain itu, pembelajaran bahasa Arab

merupakan hal yang penting dikarenakan beberapa keistimewaan yang dimilikinya, salah satunya adalah bahasa Arab merupakan bahasa dalam al-Quran.

Oleh karena itu, pemisahan tulisan Arab ini merupakan salah satu alternatif yang baik untuk dikembangkan karena bisa mempermudah mengenali huruf dasar pembentuk tulisan Arab yang oleh kebanyakan orang dianggap sulit, rumit, dan membingungkan.

B. RUMUSAN MASALAH

Berdasarkan latar belakang tersebut, rumusan masalah yang dapat diambil adalah bagaimana merancang dan membuat aplikasi yang dapat digunakan sebagai pemisah tulisan Arab sambung menjadi huruf dasar pembentuknya.

C. TUJUAN

Tujuan penelitian ini adalah untuk merancang dan membuat aplikasi yang dapat digunakan sebagai pemisah tulisan Arab sambung menjadi huruf dasar pembentuknya.

D. BATASAN MASALAH

Batasan masalah dalam penelitian ini adalah:

1. Input citra yang dibaca berupa tulisan huruf Arab ketikan dengan format bitmap (.bmp)
2. Pengenalan karakter hanya pada teks hasil *print*, tidak dapat mengenali karakter dengan tulisan tangan.
3. *Background* dari dokumen hasil *scan* berwarna putih

4. Aplikasi ini sebatas membaca tulisan Arab yang kemudian di tampilkan dalam tulisan Arab pisah menurut huruf dasar pembentuknya
5. Tulisan Arab tidak memakai tanda baca (kasrah(), kasrahtan(), fathah(), kasrahtan(), dhommah(), dhomahtan(), sukun(°), tanwin(), mad(˘)).
6. Tulisan Arab ditulis menggunakan font *Arabic Transparant 48*

E. MANFAAT PENELITIAN

1. Manfaat Bagi Peneliti:

Penelitian ini bermanfaat untuk mengembangkan wawasan keilmuan dan menambah pengetahuan, khususnya permasalahan pada pengolahan citra digital dalam memisahkan tulisan Arab sambung

2. Manfaat Bagi Pihak Akademis (Universitas Islam Negeri Malang):

Sebagai kontribusi positif untuk kemajuan wawasan keilmuan teknologi informasi yang diintegrasikan dengan agama serta untuk pengembangan pada masa yang akan datang

3. Manfaat Bagi Masyarakat:

Mempermudah masyarakat yang belajar atau hanya ingin mengetahui huruf dasar pembentuk tulisan Arab yang telah disambung, serta dapat dijadikan media belajar iqro secara mandiri (otodidak)

F. METODOLOGI PENELITIAN

1. Studi Literatur

Pada tahap ini dilakukan pencarian dan pemahaman literatur yang berhubungan dengan permasalahan perbaikan citra, *segmentasi*, pengkodean, pengenalan pola, dan huruf hijaiyah. Literatur yang digunakan meliputi buku referensi, buku Tugas Akhir mahasiswa jurusan teknik Informatika dan paper IEEE serta dokumentasi internet.

2. Perumusan masalah dan penyelesaiannya

Tahap ini meliputi perumusan masalah, batasan-batasan masalah dan penyelesaiannya serta penentuan parameter untuk mengukur hasilnya.

3. Perancangan dan Desain Sistem

Pada tahap ini dilakukan perancangan perangkat lunak untuk menerapkan permasalahan dan penyelesaiannya pada tahap sebelumnya.

4. Pembuatan Perangkat Lunak

Pada tahap ini dilakukan pembuatan perangkat lunak sesuai dengan perancangan perangkat lunak yang telah dilakukan.

5. Uji Coba dan Evaluasi Hasil

Tahap ini meliputi uji coba terhadap algoritma yang diterapkan pada pemisahan tulisan Arab menjadi pola huruf hijaiyah. Dalam hal ini juga dilakukan evaluasi dari setiap percobaan. Proses uji coba ini diperlukan untuk memastikan sistem yang telah dibuat sudah benar, sesuai dengan tujuan yang hendak dicapai

6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan penulisan buku Tugas Akhir yang merupakan dokumentasi dari konsep atau teori penunjang, perancangan dan desain sistem, pembuatan perangkat lunak, dokumentasi dari uji coba dan analisis, serta kesimpulan dan saran.

G. SISTEMATIKA PENULISAN SKRIPSI

Sistematika dalam penulisan skripsi ini akan dibagi menjadi beberapa bab sebagai berikut:

BAB 1 PENDAHULUAN

Berisi latar belakang, tujuan, rumusan masalah, batasan permasalahan, metodologi, manfaat, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Berisi teori dasar pendukung penelitian ini, yang mengulas secara garis besar tentang proses awal sampai dengan hasil dari pemisahan tulisan Arab menjadi pola huruf hijaiyah.

BAB 3 DESAIN DAN PERANCANGAN SISTEM

Berisi uraian tentang perancangan sistem, algoritma, dan antar muka dari pemisahan tulisan Arab menjadi pola huruf hijaiyah yang dilengkapi dengan beberapa diagram alir (*flowchart*) beserta penjelasannya.

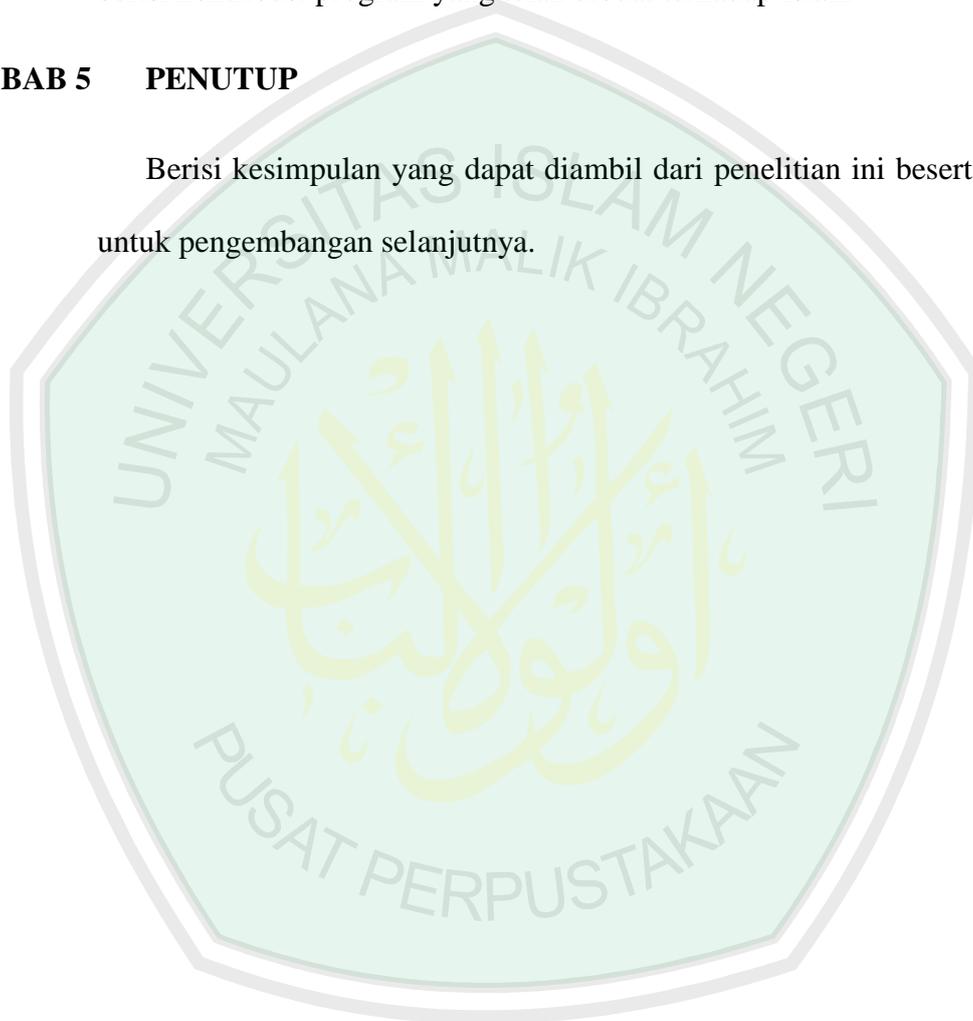
BAB 4 HASIL DAN PEMBAHASAN

Berisi hasil uji coba berdasarkan parameter-parameter yang ditetapkan,

dan kemudian dilakukan analisa terhadap hasil uji coba tersebut. Untuk mengetahui aplikasi tersebut telah dapat menyelesaikan permasalahan yang dihadapi sesuai dengan yang diharapkan. Selain itu, pada bab ini berisi kontribusi program yang telah dibuat terhadap Islam

BAB 5 PENUTUP

Berisi kesimpulan yang dapat diambil dari penelitian ini beserta saran untuk pengembangan selanjutnya.



BAB II

TINJAUAN PUSTAKA

A. TEORI DASAR PENGOLAHAN CITRA

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, *scanner*, dan sebagainya. Sehingga bayangan objek yang disebut citra tersebut terekam.

Meskipun citra kaya informasi, namun seringkali citra tersebut mengalami penurunan mutu (*degradasi*), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Sehingga citra semacam ini akan menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*).

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Operasi-operasi pengolahan citra diterapkan pada citra apabila (Munir, 2004: 3):

Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra.

Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur. Sebagian citra perlu digabung dengan bagian citra yang lain.



Gambar 2.1 Diagram Pengolahan Citra

1. Format Citra

Komputer dapat mengolah isyarat-isyarat elektronik digital yang merupakan kumpulan sinyal biner (bernilai 0 dan 1). Untuk itu, citra digital harus mempunyai format tertentu yang sesuai sehingga dapat merepresentasikan obyek pencitraan dalam bentuk kombinasi data biner. Format citra yang banyak dipakai adalah citra biner, skala keabuan, warna, dan warna berindeks (Balza, 2005: 8)

Tabel 2.1

Format citra

Skala Keabuan	Rentang Nilai Keabuan	Piksel Depth
2^1 (2 nilai)	0, 1	1 bit
2^2 (4 nilai)	0 sampai 7	2 bit
2^3 (16 nilai)	0 sampai 15	3 bit
2^8 (256 nilai)	0 sampai 255	8 bit

a. Citra Biner

Pada citra biner, setiap nilai bernilai 0 atau 1, masing-masing merepresentasikan warna tertentu. Contoh yang paling lazim adalah hitam bernilai 0 dan putih bernilai 1.

b. Citra Skala Keabuan

Disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimal dan putih sebagai warna maksimal, sehingga warna di antara keduanya adalah abu-abu.

Namun dalam prakteknya warna yang dipakai tidak terbatas pada warna abu-abu, sebagai contoh dipilih warna minimalnya adalah putih dan warna maksimalnya adalah merah, maka semakin besar nilainya semakin besar pula intensitas warna merahnya.

c. Citra Warna

Pada citra warna, setiap titik mempunyai warna spesifik yang merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru. Format citra ini sering disebut citra RGB. Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit).

Jumlah kombinasi warna yang mungkin untuk format citra ini adalah 2^{24} atau lebih dari 16 juta warna, dengan demikian bisa dianggap mencakup semua warna yang ada. Oleh karena itu, dinamakan *true color*.

d. Citra Warna Berindeks

Jumlah memori yang dibutuhkan untuk format citra warna *true color* adalah tiga kali jumlah titik yang ada dalam citra yang ditinjau. Di lain pihak, jumlah warna yang ada dalam suatu citra terkadang sangat terbatas, karena banyaknya warna dalam citra tidak mungkin melebihi banyaknya titik dalam citra itu sendiri. Dengan kasus seperti ini, disediakan format citra warna berindeks. Pada format ini, informasi setiap titik merupakan indeks dari suatu tabel yang berisi informasi warna yang tersedia, yang disebut palet warna atau *color map*.

2. Operasi Pengolahan Citra

Pengolahan citra pada dasarnya dilakukan dengan cara memodifikasi setiap titik dalam citra tersebut sesuai keperluan. Secara garis besar, modifikasi tersebut dikelompokkan menjadi:

1. Operasi titik, di mana setiap titik diolah secara tidak menempel terhadap titik-titik yang lain
2. Operasi global, di mana karakteristik global (biasanya berupa sifat statistik) dari citra digunakan untuk memodifikasi nilai setiap titik.
3. Operasi temporal/berbasis bingkai, di mana citra diolah dengan cara dikombinasikan dengan citra lain.
4. Operasi geometri, yaitu operasi pengolahan citra yang berhubungan dengan perubahan bentuk geometri citra, baik bentuk, ukuran, atau orientasinya. Beberapa contoh pada operasi geometri, di antaranya: pencerminan (*flipping*), rotasi/pemutaran (*rotating*), penskalaan (*scaling/zooming*), pemotongan (*cropping*), dan pendoyongan (*skew*)

5. Operasi banyak titik bertetangga, di mana data dari titik-titik yang bersebelahan (bertetangga) dengan titik yang ditinjau ikut berperan dalam mengubah nilai.
6. Operasi morfologi, yaitu operasi yang berdasarkan segmen atau bagian dalam citra yang menjadi perhatian.

a. Pengambangan gambar (*Image Thresholding*)

Pengambangan gambar (*Image thresholding*) digunakan untuk mengubah citra dengan format skala keabuan (*grayscale*), yang mempunyai kemungkinan nilai lebih dari 2 ke citra biner yang hanya memiliki 2 buah nilai (0 dan 1). Tujuan dari *thresholding* adalah proses untuk memisahkan *foreground* (latar depan) dengan *background* (latar belakang) dari suatu citra. Proses *thresholding* dilakukan dengan cara melihat perbedaan intensitas warna dari suatu citra. *Input* untuk proses *thresholding* ialah citra abu-abu (*grayscale image*) atau citra warna (*color image*). *Output* dari proses ini ialah *binary image*, yang mana piksel hitam mewakili *foreground* dan piksel putih mewakili *background*, atau sebaliknya. *Binary image* adalah suatu *image* yang mana pikselnya hanya memiliki dua nilai intensitas. Nilai intensitas yang sering digunakan yaitu 0 untuk piksel hitam, 1 atau 255 untuk piksel putih.

Dalam *image thresholding* ini, ditentukan nilai T setelah melihat grey level dari citra tersebut. T adalah nilai minimum di antara 2 nilai maksimal yang ada pada nilai gray level citra tersebut. Proses selanjutnya adalah mengganti setiap intensitas warna yang ada dalam citra tersebut. Jika intensitas warnanya lebih kecil atau sama dengan T maka intensitasnya diganti '0'. Tetapi jika intensitas

warnanya lebih besar atau sama dengan T maka intensitasnya diganti '1'. Secara matematis memiliki model sebagai berikut:

$$g_2(x, y) = \begin{cases} 0, & \text{jika } g_1(x, y) \leq T \\ 1, & \text{jika } g_1(x, y) \geq T \end{cases}$$

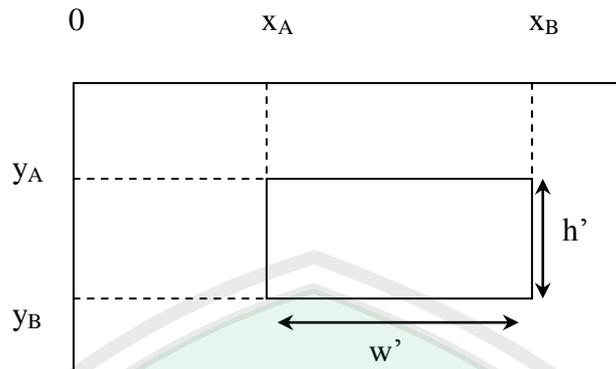
Dari proses tresholding di atas maka akan didapat hasilnya apabila jumlah warna piksel mempunyai kedalaman warna yang kurang dari nilai tresholding maka warna piksel tersebut akan menjadi 0 (hitam) dan juga sebaliknya.

b. Pemisahan Karakter (*Segmentation*)

Segmentasi merupakan proses pembagian daerah dalam suatu *image* untuk dikelompokkan ke dalam segmen-segmen tertentu. *Image* yang berisi kalimat akan dipisahkan menjadi karakter-karakter. Hasil dari proses segmentasi ini adalah karakter-karakter yang telah berdiri sendiri dan tidak menjadi bagian dari kalimat. Salah satu cara segmentasi yaitu *cropping* atau pemotongan. *Cropping* adalah memotong suatu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil. Operasi ini pada dasarnya operasi translasi, yaitu menggeser koordinat titik citra. Rumus yang digunakan untuk operasi ini adalah:

$$\begin{aligned} x' &= x - x_A \text{ untuk } x = x_A \text{ sampai } x_B \\ y' &= y - y_A \text{ untuk } y = y_A \text{ sampai } y_B \end{aligned}$$

(x_A, y_A) dan (x_B, y_B) masing-masing adalah titik pojok kiri atas dan pojok kanan bawah bagian citra yang akan *dicrop* (Gambar 2.2)



Gambar 2.2 Koordinat titik pojok bagian citra yang akan di crop

Ukuran citra menjadi:

$$w' = x_B - x_A$$

$$h' = y_B - y_A$$

dan transformasi baliknya adalah:

$$x = x' + x_A \quad \text{untuk } x' = 0 \text{ sampai } w' - 1$$

$$y = y' + y_A \quad \text{untuk } y' = 0 \text{ sampai } h' - 1$$

c. Penipisan (*Thinning*)

Penipisan (*thinning*) adalah operasi pengubahan citra menjadi vektor dengan melalui bagian-bagian yang umum. Tujuan penipisan adalah untuk merubah citra dengan suatu ketebalan tertentu menjadi citra dengan ketebalan 1 piksel. Citra yang ditipiskan adalah gambar yang hanya terdiri dari horisontal dan vertikal. Cara yang digunakan biasanya dengan pengikisan citra, baik bagian atas, bawah, kanan, ataupun kiri.

Syarat-syarat operasi *thinning* adalah sebagai berikut:

- Mempertahankan keterhubungan piksel-piksel objek
- Pola hasil penipisan harus tetap mempunyai bentuk yang menyerupai pola asalnya
- Tidak memperpendek ujung lengan dari bentuk yang ditipiskan

- Pengikisan tidak boleh dilakukan terhadap citra dengan ketebalan 1 piksel.

P_8	P_1	P_2
P_7	P_0	P_3
P_6	P_5	P_4

Gambar 2.3 Notasi piksel yang digunakan untuk memeriksa keterhubungan

Algoritma penipisan (*thinning*) yang umum adalah memeriksa piksel-piksel di dalam jendela yang berukuran 3x3 piksel dan mengelupas satu piksel pada pinggiran (batas) objek pada setiap lelaran, sampai objek berkurang menjadi garis tipis. Algoritma bekerja secara iteratif, pada setiap lelaran dilakukan pemrosesan pada jendela yang berukuran 3x3 piksel.

Algoritmanya sebagai berikut (Munir, 2004: 193):

1. Mula-mula diperiksa jumlah piksel objek yang bernilai satu, N , di dalam jendela 3x3 piksel
2. Jika N kurang atau sama dengan 2, tidak ada aksi yang dilakukan karena dalam jendela terdapat ujung lengan objek
3. Jika N lebih besar dari 7, tidak ada aksi yang dilakukan karena dapat menyebabkan pengikisan (*erosion*) objek
4. Jika N lebih besar dari 2, periksa apakah penghilangan piksel tengah menyebabkan objek tidak terhubung. Ini dilakukan dengan membentuk barisan $P_1 P_2 P_3 \dots P_8 P_1$. Jika jumlah peralihan $0 \rightarrow 1$ di dalam barisan tersebut sama dengan 1, berarti hanya terdapat satu komponen penghubung di dalam jendela 3x3. Pada kasus seperti ini, dibolehkan

menghapus piksel tengah yang bernilai 1 karena penghapusan tersebut tidak mempengaruhi keterhubungan.

Algoritma tersebut dapat ditulis secara ringkas, sebagai berikut:

Tahap I: Pengikisan pada sisi kanan, bawah, dan kiri atas

Syarat

- $2 \leq N(P_0) \leq 6$
 $N(P_0) \neq 1, N(P_0) \neq 7$
- $T(P_0) = 1$ (jumlah transisi dari 0 ke 1)
artinya ketebalan piksel harus lebih dari 1
- $P_1, P_3, P_5 = 0$ (tetangga atas, kanan, dan bawah tidak ada)
- $P_3, P_5, P_7 = 0$ (tetangga kanan, bawah, dan kiri tidak ada)

Tahap II: Pengikisan pada sisi kiri, atas, dan kanan bawah

Syarat

- $2 \leq N(P_0) \leq 6; N(P_0) \neq 1, N(P_0) \neq 7$
- $T(P_0) = 1$
- $P_1, P_3, P_7 = 0$ (tidak mempunyai tetangga atas, kanan, dan kiri)
- $P_1, P_5, P_7 = 0$ (tidak punya tetangga kiri, atas, dan bawah)

Contoh pada pengikisan huruf L di bawah ini:

Iterasi 1

(I) xxxxx	(II) xxx
xxxxx	xxxx
xxxxx	xxxx
xxxxxxxxxx	xxxxxxxxxx
xxxxxxxxxx	

Iterasi 2

(I) xxx	(II) x
xxx	xx
xxx	xx
xxxxxxxxx	xxxxxxxxx
xxxxxxxxx	xxxxxxxxx
xxxxxxxxx	

Iterasi 3

(I) x
x
x
xxxxx

d. Penskalaan (*Scaling*)

Operasi penskalaan (*scaling*) dimaksudkan untuk memperbesar (*zoom in*) dan memperkecil (*zoom out*) citra. Hal ini dapat dilakukan dengan mengatur nilai variabel, baik ke arah horizontal maupun vertikal (Balza, 2005: 183). Skala yang bernilai 1 akan memperbesar citra, sedangkan jika bernilai kurang dari 1 akan memperkecil citra. Apabila perbandingan antara lebar dan tinggi citra hendak dipertahankan, maka skala horisontal sama dengan vertikal, dapat ditulis $S_h = S_v$. Dan transformasi spasial yang dapat dipakai adalah $x' = S_h x$ dan $y' = S_v y$.

Sedangkan transformasi balik untuk persamaan diatas adalah $x = \frac{x'}{S_h}$ dan $y = \frac{y'}{S_v}$.

Dari persamaan-persamaan tersebut, untuk setiap titik pada citra hasil, dengan koordinat x' dan y' yang diketahui, dapat dicari koordinat titik asalnya yaitu x dan y . Karena adanya operasi pembagian, seringkali terjadi situasi di mana dari penerapan transformasi spasial diperoleh koordinat titik hasil (pada transformasi maju) maupun titik asal (pada transformasi balik) yang bernilai tidak bulat, padahal koordinat titik harus selalu bernilai bulat. Untuk mengatasinya, diperlukan interpolasi yang dapat digunakan dalam menentukan nilai keabuan pada koordinat titik hasil berdasarkan nilai keabuan pada koordinat titik asal. Interpolasi yang sering digunakan adalah interpolasi tetangga terdekat, bilinear, dan interpolasi dengan orde lebih tinggi.

Ukuran citra juga berubah sesuai hubungan $w' = S_h w$ dan $h' = S_v h$. Operasi pembesaran atau pengecilan dapat dilakukan dengan mengatur nilai variabel S_h dan S_v .

B. BAHASA ARAB

1. Pengenalan Karakter Huruf Arab

Bahasa Arab adalah bahasa wahyu yang mana Allah menurunkan Al-Quran dalam bahasa tersebut. Ada beberapa sebab dan hikmah mengapa Allah memilihnya, padahal bisa saja memilih Bahasa Inggris, Perancis, Yunani, karena semua bahasa di dunia ini adalah ciptaan-Nya. Menurut kajian ilmu linguistik, Bahasa Arab adalah bahasa paling indah di dunia dari segi kesusasteraan dan kehalusan metafora. Huruf-hurufnya mempunyai bunyi yang tersendiri. Jika salah dalam bunyi dan baris, makna akan berubah dan penafsiran pun akan berbeda. Di dalam al-Quran sendiri, juga dijelaskan apa saja keistimewaan bahasa Arab, antara lain ialah: 1. sejak zaman dahulu hingga sekarang bahasa Arab itu merupakan bahasa yang hidup, 2. bahasa Arab adalah bahasa yang lengkap dan luas untuk menjelaskan tentang ketuhanan dan keakhiratan. 3. bentuk-bentuk kata dalam bahasa Arab mempunyai *tasrif* (konjugasi) yang amat luas sehingga dapat mencapai 3000 bentuk perubahan, yang demikian tak terdapat dalam bahasa lain. Adapun ayat tersebut adalah sebagai berikut:

وَكَذَلِكَ أَنْزَلْنَاهُ حُكْمًا عَرَبِيًّا وَلَئِنِ اتَّبَعْتَ أَهْوَاءَهُمْ بَعْدَ مَا جَاءَكَ مِنَ الْعِلْمِ مَا

لَكَ مِنَ اللَّهِ مِنْ وَّلِيٍّ وَلَا وَاقٍ ﴿١٣﴾

Artinya:

“Dan Demikianlah, kami Telah menurunkan Al Quran itu sebagai peraturan (yang benar) dalam bahasa Arab. dan seandainya kamu mengikuti hawa nafsu mereka setelah datang pengetahuan kepadamu, Maka sekali-kali tidak ada pelindung dan pemelihara bagimu terhadap (siksa) Allah.(QS. Ar-Ra’d/13: 37)”

Selain kelebihan tersebut, bahasa Arab merupakan bahasa percakapan lebih dari 20 negara di dunia dan juga menjadi bahasa kedua di beberapa negara yang berpenduduk mayoritas beragama Islam, contohnya Iran, Malaysia, Pakistan, dan Indonesia. Tidak hanya itu saja, karakter Arab juga digunakan menulis beberapa bahasa tidak hanya oleh negara Arab saja, tetapi Urdu, Farsi, Iran, Pakistan, dan lain-lain. Al-Quran ditulis dengan menggunakan huruf Arab, yang dirangkai dari huruf hijaiyah sehingga membentuk makna tertentu. Huruf Arab berjumlah 28 buah, ditulis dari kanan ke kiri dengan bentuk dan ukuran yang berbeda. Satu huruf bisa mempunyai 4 bentuk yang berbeda ketika disambung dengan huruf/abjad Arab yang lain (Tabel 2.2). Karakteristik huruf Arab adalah sebagai berikut: (Al-Rashaideh, 2008: 13)

- 1) Huruf Arab adalah suatu bahasa jenis kursif² menulis dari kanan ke kiri, tidak ada huruf kapital, dan beberapa huruf tidak bisa dihubungkan dengan huruf yang mengikutinya.

Contohnya: **اِيَاكَ نَعْبُدُ وَ اِيَاكَ نَسْتَعِينُ**

- 2) Huruf Arab mempunyai 28 karakter dasar yang biasa disebut huruf hijaiyah. Masing-masing karakter mempunyai 2-4 format yang tergantung pada posisinya di dalam kata. Posisi tersebut antara lain: awal kata (*start*), tengah (*middle*), akhir (*end*), dan berdiri sendiri (*isolated*). Lihat tabel 2.4
- 3) Beberapa karakter huruf Arab ditempatkan di bawah garis dasar, contohnya: ل, ز, ر

² Tulisan miring; miring; posisi miring/condong

- 4) Beberapa karakter huruf Arab terdiri dari 2 bagian, contohnya لا, ظ, ط
- 5) Huruf Arab mempunyai titik di atas atau di bawah badan karakter, dan beberapa huruf mempunyai suatu *hamza* (ء) dan *madda* (~). Contohnya:

Hamza(ء) أصبرهم
Madda(~) دَابَّةُ السَّمَاءِ مَاءٌ

- 6) Huruf Arab memiliki 15 bentuk dasar yang sama, lebih dari 15 kelas untuk huruf yang berdiri sendiri, 10 kelas mempunyai dua atau lebih huruf yang berubah dengan suatu titik/lambang.

ا ب ح د ر س ص ط ع
9 8 7 6 5 4 3 2 1
ف ق م ه و ي
15 14 13 12 11 10

Gambar 2.4 Bentuk dasar huruf Arab berdasarkan kelasnya

- 7) Pada penulisan huruf Arab terdapat font dan gaya penulisan yang berbeda-beda, meskipun suatu karakter memiliki font yang sama terkadang terdapat perbedaan ukuran. Oleh karena itu, segmentasi terhadap ukuran lebar tertentu tidak bisa diaplikasikan ke pengenalan huruf Arab.

Tabel 2.2
Daftar Huruf Arab

Name	Isolated	Start	Middle	End
Alif	ا			آ
Ba	ب	بـ	ـبـ	ـبـ
Ta	ت	تـ	ـتـ	ـتـ
Tha	ث	ثـ	ـثـ	ـثـ
Jeem	ج	جـ	ـجـ	ـجـ
Hha	ح	حـ	ـحـ	ـحـ
Kha	خ	خـ	ـخـ	ـخـ
Dal	د			ذ
Thal	ذ			ذ
Ra	ر			ر
Zay	ز			ز
Seen	س	سـ	ـسـ	ـسـ
Sheen	ش	شـ	ـشـ	ـشـ
Sad	ص	صـ	ـصـ	ـصـ
Dhad	ض	ضـ	ـضـ	ـضـ
Tta	ط	طـ	ـطـ	ـطـ
Za	ظ	ظـ	ـظـ	ـظـ
Ain	ع	عـ	ـعـ	ـعـ
Ghain	غ	غـ	ـغـ	ـغـ
Fa	ف	فـ	ـفـ	ـفـ
Qaf	ق	قـ	ـقـ	ـقـ
Kaf	ك	كـ	ـكـ	ـكـ
Lam	ل	لـ	ـلـ	ـلـ
Meem	م	مـ	ـمـ	ـمـ
Noon	ن	نـ	ـنـ	ـنـ
Ha	هـ	هـ	ـهـ	ـهـ
Waow	و			و
Ya	ي	يـ	ـيـ	ـيـ

Sumber: Fahmy (2005: 3)

2. Keistimewaan Mempelajari Bahasa Arab Dalam Perspektif Islam

Beberapa karakter istimewa bahasa Arab, antara lain (Amalian: 2008):

1. Mudah

Yang dimaksud mudah disini adalah al-Quran yang berbahasa Arab itu harus mudah dihafal dan dibaca. Al-Quran yang merupakan karya agung sastra adalah bukti bahwa bahasa Arab itu mudah. Sebagaimana disebutkan dalam Q.S. Al-Qamar ayat 17, 22, 32, dan 40: "*Dan sesungguhnya telah Kami mudahkan Al-Quran itu untuk pelajaran, maka adakah orang yang mengambil pelajaran?*"

Salah satu karakteristik bahasa Arab adalah mudah untuk dihafalkan, bahkan penduduk gurun pasir yang tidak bisa baca tulis pun mampu menghafal jutaan bait syair. Dan karena mereka terbiasa menghafal apa saja di luar kepala, sampai-sampai mereka tidak terlalu butuh lagi dengan alat tulis atau dokumentasi. Kisah cerita yang tebalnya berjilid-jilid buku, bisa digubah oleh orang Arab menjadi jutaan bait puisi dalam bahasa Arab dan dihafal luar kepala dengan mudah.

2. Indah

Keindahan bahasa Arab terdiri dari tiga dimensi, antara lain dimensi ketika dibaca, didengar, dan ditulis. Yang pertama dan kedua merupakan cermin dari ilmu sastra Arab, yaitu *balaghah*. Adapun dimensi yang ketiga merupakan cerminan dari seni kaligrafi Arab.

3. Syamil

Kesempurnaan bahasa Arab bisa ditemukan dari khazanah kosa katanya

yang kaya. Untuk menjelaskan konteks yang berbeda pada pembicaraan yang sama, bahasa Arab mampu mewakili. Misalnya, perbedaan antara kata '*Rabb*' dan '*Ilah*' yang keduanya bermakna 'Tuhan'.

4. Mu'jizah

Keistimewaan bahasa Arab adalah *mu'jizah*, yang artinya menarik. Al-Quran diturunkan dengan bahasa ini, dan ia memberikan nilai lebih dari berbagai sisi.

5. Cerdas

Islam, dan bahasa Arab telah melahirkan berjuta ulama dari berbagai bangsa 'Ajam (non Arab) di berbagai disiplin ilmu.

6. Jelas

Bahasa Arab itu jelas, seperti diterangkan dalam Al-Quran, "Dan sesungguhnya Kami mengetahui bahwa mereka berkata: "Sesungguhnya Al-Quran itu diajarkan seseorang kepadanya (Muhammad)", padahal Ia (Muhammad) belajar kepadanya bahasa 'Ajam, sedang Al-Quran adalah dalam bahasa Arab yang jelas." (Q.S. An-Nahl: 103)

Bahasa Arab dipilih Allah SWT untuk menghantarkan kalam-Nya yang mulia sudah pasti dikarenakan bahasa Arab memiliki keistimewaan tersendiri. Beberapa keistimewaan tersebut, antara lain (Hamdani, 2008):

1. Bahasa Arab adalah satu-satunya bahasa yang mampu melukiskan wahyu Ilahi secara sempurna dengan sefasih-fasihnya kalam dan

seindah-indahnya susunan. Begitu indahnya bahasa Arab Al-Quran sampai ahli sastra Arab di zamannya tidak bisa mendefinisikannya (al-Muddatsir:18-25)

Kata bahasa Arab mempunyai *tashrif*³ yang amat luas. Satu kata akar bisa melahirkan 3000 kosa kata baru dan satu tema bisa diungkapkan oleh lebih dari 10 kata dan setiap kata bisa diungkapkan dalam bentuk asli atau kiasan. Sebagai contoh, bahasa Arab mempunyai lebih dari 700 kata yang berbeda untuk seekor unta dengan berbagai kondisinya atau 200 kata untuk anjing. Dengan kekayaan perbendaharaan katanya tersebut, bahasa Arab mampu menjelaskan makna isi Al-Qur'an dengan tepat. Bahasa Arab sudah memiliki istilah-istilah untuk menjelaskan proses penciptaan manusia pada (Al-Mu'minun: 12-14).

Selain itu, kata bahasa Arab bersifat konsepsional, dalam arti tidak sebatas identifikasi benda, melainkan juga bisa menggambarkan proses benda tersebut. Sebagai contoh, kata roti (*khubz*) dengan ketiga huruf pembentuknya (kha, ba, za) memotret proses pembuatannya: *bazakha* berarti memukul-memukul sesuatu, *khazaba* berarti mengubah sesuatu cepat-cepat dengan tangan, *khazaba* berarti menjadi mengembang. Jadi, *khubz* (roti) adalah sesuatu yang diolah dengan membanting dan mengubah adonan dengan tangan, kemudian adonan itu menjadi mengembang. Contoh lainnya adalah kata *baydl*

³ Perubahan

(telur) mengekspresikan sesuatu yang berwarna putih (*abyadl* atau *baydl*) yang berbentuk lonjong (*baydla*).

Kaidah-kaidah tatabahasa Arab sangat sempurna dan kuat. Kaidah-kaidah tersebut meliputi ilmu *Shorof*, *Nahwu*, *Ma'ani*, *Bayan*, *Badi'*, *'Arudh*, *Qowafi*, *Matan Lughoh*, *Qordhus Syi'ir* dan lain-lain. Kesemuanya itu mempunyai fungsionalitas tertentu dan saling melengkapi. Bersama Al-Quran, kaidah-kaidah tersebut ikut berperan menjaga orisinalitas dan kesehatan bahasa Arab

2. Bahasa Arab termasuk bahasa tertua di dunia, bahkan lebih tua dan lebih kekal dari umur sejarah manusia di bumi berdasarkan pendapat ilmuwan yang menyebutkan bahwa bahasa Arab merupakan bahasa Nabi Adam a.s. ketika di surga dan menjadi cikal bakal bahasa-bahasa di duni. Hadits Nabi SAW di awal menyebutkan bahasa Arab sebagai bahasa penduduk surga. Alih-alih punah seperti bahasa-bahasa tua dunia lainnya, bahasa Arab termasuk bahasa besar dan resmi di dunia modern saat ini.
3. Bahasa Arab adalah bahasa persatuan umat Islam. Sebagai bahasa Al-Quran, bahasa Arab tidak diturunkan untuk satu kaum saja, tapi juga untuk semua kaum di dunia ini, karena bahasa Arab merefleksikan bahasa aqidah umat Islam yang penuh persaudaraan universal. Berbicara bahasa Arab membuat citra kuat status kemusliman seseorang di kalangan komunitas yang tidak berbahasa Arab. Sebagai

orang bukan Arab, kita akan disambut layaknya seorang saudara di dalam komunitas bangsa Arab, dan dikagumi ketika kita mampu berbicara bahasa Arab klasik.

Hadits Nabi saw yang diriwayatkan oleh Al Hafidz Ibnu Asakir dengan sanad dari Malik: "Wahai sekalian manusia, sesungguhnya Rabb itu satu, bapak itu satu, dan agama itu satu. Bukanlah Arab di kalangan kamu itu sebagai bapak atau ibu. Sesungguhnya Arab itu adalah lisan (bahasa), maka barangsiapa yang berbicara dengan bahasa Arab, dia adalah orang Arab".

4. Bahasa Arab sebagai Bahasa Al-Quran.

Bahasa Al-Quran merupakan bahasa lisan yang merekam percakapan dari Allah SWT kepada Nabi Muhammad SAW. Sebagai bahasa lisan, maka tinggi rendah nada suara sangat mempengaruhi makna yang dikandungnya. Oleh karena itu, pengetahuan tatabahasa Arab akan membantu dalam memahami ayat-ayat Al-Quran yang menuntun kita dalam melagukan Al-Quran sehingga kisah heroik, misal surat Al-Kafirun, dilagukan dengan semangat berapi-api, tidak dilagukan dengan nada meratap.

Dalam QS. al-Haqqah: 42-43 diterangkan bahwa bahasa al-Qur'an adalah bahasa percakapan dari Tuhan Pencipta alam semesta kepada utusan-Nya, yang dari segi bentuk maupun kandungannya mempunyai nilai yang sangat mulia. Al-Qur'an bukan termasuk bahasa semodel bahasa para sastrawan atau penyair yang terlalu berorientasi pada

keindahan lahiriahnya saja. Bahasa Al-Qur'an juga bukan bahasa seperti bahasa para penyihir, karena bentuk bahasa model ini seringkali sulit dinalar dan tidak komunikatif. Biasanya, bahasa jenis ini memerlukan juru tafsir khusus. Kedua model bahasa tersebut hanya dimengerti oleh para tokoh dan juru tafsirnya. Pada budaya sastra waktu itu, semakin sulit dipahami oleh orang awam akan terasa semakin hebat, dan tentu saja semakin mahal harganya. Berbeda dengan Al-Quran, meski ia dinilai sebagai karya sastra paling tinggi, namun ia masih bisa dimengerti oleh semua kalangan.

Mempelajari bahasa Arab sebagai bahasa Al-Quran dan literatur Islam sangat dianjurkan bagi setiap muslim, bahkan beberapa *Atsar*⁴ menyebutkan akan kewajiban mempelajari bahasa Arab.

Banyak ayat Al-Quran yang menekankan pentingnya mempelajari bahasa Arab sebagai bahasanya, seperti firman Allah SWT berikut: "*Sesungguhnya Kami telah menurunkan Al-Qur'an yang berbahasa Arab semoga kalian berpikir*" (Q.S. Yusuf/12: 2)

Selain itu terdapat banyak ayat Al-Quran lainnya yang senada dengan ayat di atas seperti: Az-Zukhruf: 3, Az-Zumar: 28, Fushshilat: 3, 44, Ar-Ra'du: 37, Asy-Syura: 7, Thoha: 113, Al-Ahqof: 12, An-Nahl: 103, Maryam: 97 dan Asy-Syu'aro: 195.

⁴ perkataan sahabat, tabi'in dan para pengikut tabi'in

Rasulullah SAW memerintahkan kepada kita agar mempelajari bahasa Arab dan sekaligus mengajarkannya, sabda beliau adalah: "Pelajarilah kamu sekalian bahasa Arab dan ajarkanlah kepada manusia".

Bahkan, kita dituntut oleh Rasulullah SAW untuk mencintai bahasa Arab sebagaimana sabda beliau, "Cintailah bahasa Arab karena tiga hal: pertama, karena aku adalah orang Arab; kedua, karena Al-Qur'an berbahasa Arab; dan ketiga, karena bahasa penduduk surga adalah bahasa Arab." Sahabat Umar bin Al-Khatthab berkata: "Pelajarilah bahasa Arab karena sesungguhnya bahasa Arab itu merupakan suatu bagian dari agama kalian."

Beberapa ulama besar menekankan kewajiban mempelajari bahasa Arab. Al-Imam Asy-Syafi'i RA berkata, "Bahasa Arab hukumnya wajib atas setiap muslim." Al-Imam Al-Ghozaly berkata dalam kitabnya *Ihya' Ulumiddin*, "Sesungguhnya bahasa Arab dan Nahwu adalah suatu sarana untuk mengetahui Al-Qur'an dan sunnah Nabi SAW. Keduanya bukanlah termasuk dari ilmu-ilmu syar'i akan tetapi wajib hukumnya mendalami kedua ilmu tersebut karena syari'ah ini datang dengan bahasa Arab dan setiap syari'ah tidak akan tampak kecuali dengan suatu bahasa."

Asy-Syeikh Ibnu Taimiyyah menyebutkan dalam salah satu kitabnya: "Sesungguhnya bahasa Arab itu sendiri adalah sebagian dari agama Islam dan mengetahuinya adalah wajib 'ain karena merupakan sarana untuk memahami Al-Qur'an dan Al-Hadits. Tidaklah sempurna suatu kecuali dengan memahami bahasa Arab dan tidaklah sempurna suatu

kewajiban kecuai dengan suatu hal maka suatu hal tersebut adalah wajib hukumnya." Asy Syahid Hasan Al-Bana telah mewasiatkan: "Berbicaralah dengan menggunakan bahasa Arab karena hal ini merupakan bagian dari syi'ar Islam".

C. PENGENALAN POLA

1. Pengertian Pola dan Ciri

Pola adalah entitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya (*features*). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Sebagai contoh

Tabel 2.3

Tabel contoh pengelompokan pola berdasarkan cirinya

Pola	Ciri
Huruf	Tinggi, tebal, titik sudut, lengkungan garis
Suara	Amplitude, frekuensi, nada, intonasi, warna
Tanda tangan	Panjang, kerumitan, tekanan
Sidik jari	Lengkungan, jumlah garis

Ciri dari suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Khusus pada pola yang terdapat di dalam citra, ciri-ciri yang dapat diperoleh berasal dari informasi:

- a. Spasial : intensitas piksel dan histogram

- b. Tepi : arah dan kekuatan
- c. Kontur : garis, elips, dan lingkaran
- d. Wilayah/bentuk : keliling, luas, dan pusat massa
- e. Hasil transformasi Fourier: frekuensi

Pengenalan pola (*pattern recognition*) merupakan salah satu cabang ilmu komputer yang dapat diartikan sebagai pengumpulan data-data mentah untuk dapat diklasifikasikan dengan maksud dan tujuan tertentu. Pengenalan pola ini bersifat *conceptually driven processing* yang berarti bahwa proses dimulai dari pembentukan konsep pada objek yang dijumpai (informasi dari memori).

Ada banyak aplikasi yang bisa dijadikan implementasi dari pengenalan pola, di antaranya adalah pengenalan wajah pada manusia, pengenalan tulisan tangan, pengenalan penyakit berdasarkan gejala-gejala yang ditemukan pada objek. Pengenalan wajah dan tulisan tangan manusia bisa dibuat melalui pendekatan pemrosesan citra yang mana tujuan akhir dari pemrosesan citra tersebut digunakan untuk pengelompokan objek sehingga menghasilkan output yang diinginkan. Sedangkan pengenalan penyakit berdasarkan gejala-gejala yang ada pada seorang pasien bisa dilakukan dengan pendekatan analisa data, kemudian mengambil kesimpulan dari data-data yang sudah dikelompokkan. Atau dengan kata lain, pembuatan aplikasi secara otomatis bercerita dan menjelaskan secara terperinci tentang informasi yang dikandung objek. Dan bagaimana kelengkapan informasi yang dikandung dari objek tersebut, bergantung kepada kualitas dan kuantitas dari data statistik sebagaimana dalam al-Quran surat Al-Isra/17: 88 dijelaskan tentang kualitas bahasa Arab Al-Qur'an bahwasanya:

قُلْ لِّئِنْ أَجْتَمَعَتِ الْإِنْسُ وَالْجِنُّ عَلَىٰ أَنْ يَأْتُوا بِمِثْلِ هَذَا الْقُرْآنِ لَا يَأْتُونَ

بِمِثْلِهِ ۚ وَلَوْ كَانَ بَعْضُهُمْ لِبَعْضٍ ظَهِيرًا ﴿٨٨﴾

Katakanlah: "Sesungguhnya jika manusia dan jin berkumpul untuk membuat yang serupa Al Quran ini, niscaya mereka tidak akan dapat membuat yang serupa dengan Dia, sekalipun sebagian mereka menjadi pembantu bagi sebagian yang lain".(Al-Isra/17 ayat 88)

Beberapa metode yang bisa digunakan untuk pengenalan pola adalah Jaringan Syaraf Tiruan (JST), metode statistik, metode terstruktur dan lain sebagainya. Dengan JST, menganalogikan cara berfikir pada otak manusia. Jadi, informasi di proses sebagaimana otak manusia memproses informasi yang di dapat. Misalnya cara pengenalan wajah pada manusia. Sedangkan metode statistik berdasarkan hasil analisa data yang sudah terkumpul. Misalnya pengenalan dan analisa penyakit pada manusia.

Salah satu contoh pengenalan pola adalah memprediksi makanan yang sesuai untuk seseorang dilihat dari sudut pandang tempat tinggal, suhu maupun kepekaan badan orang yang bersangkutan terhadap penyakit. Pengenalan pola di atas dapat diselesaikan dengan menggunakan metode statistik. Pada tahap awal, dilakukan pengumpulan data tentang makanan serta zat yang dikandung oleh makan tersebut. Kemudian, pengumpulan data tentang zat-zat yang diperlukan tubuh untuk kondisi daerah dan suhu tertentu. Kemudian, dilakukan pemrosesan dan analisa sehingga menghasilkan informasi yang kita inginkan.

2. Tipe dari sistem pengenalan tulisan tangan

Pengenalan tulisan tangan dapat dikelompokkan menjadi beberapa kategori, yang paling utama yaitu perbedaan ciri ketika teks dikenalkan. Menurut data masukannya, pengenalan tulisan tangan dapat dilakukan secara *offline* dan *online* (Vori, 1998)

a. Pengenalan secara *offline*

Sistem pengenalan *offline*, yaitu sistem yang menerima input data dari *scanner* atau dari gambar yang menggunakan beberapa algoritma pengolahan citra sebelum langkah pengelompokan. Cara *offline* ini lebih sulit daripada *online* karena tidak punya informasi kontekstual yang berhubungan dengan gambar, seperti posisi teks, ukuran teks, urutannya, titik awal, dan titik akhir. Kelemahannya yang lain yaitu pada cara *offline* memiliki *noise* yang lebih banyak dari pada cara *online* dan juga teks tidak dapat dikenali dalam waktu yang bersamaan dengan penulisan tetapi setelah penulisan teks terselesaikan.

Metode *offline* ini tidak cocok digunakan pada komunikasi antara manusia dan mesin dikarenakan tidak memudahkan aktifitas secara *real-time*. Metode ini cocok digunakan pada konversi penulisan dokumen pada kertas menjadi bentuk dokumen elektrik yang kemudian diterjemahkan atau *postprocessing* menggunakan komputer.

b. Pengenalan secara *online*

Sistem pengenalan *online*, yaitu sistem yang menerima data secara langsung (*real time*) melalui gerakan pena dari perangkat keras (*hardware*)

komputer, misalnya *graphic tablet* dan *light pen*. Kemudian menghitung hubungan antar titik untuk mengekstraksi fitur dalam waktu itu.

Pengenalan secara *online* memiliki banyak perbedaan dengan pengenalan secara *offline* karena informasi yang dinamis pada penulisannya, kecepatan menulis, percepatan, sudut dari pena, dan tekanannya dalam tulisan.

3. Metode Pengenalan

Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Tujuan pengelompokan adalah untuk mengenali suatu objek dalam citra.

Pengenalan pola telah menjadi bagian dari kecerdasan buatan (*artificial intelligence*), karena merupakan suatu kecerdasan maka pengenalan pola memerlukan fungsi-fungsi pengambil keputusan. Dari fungsi-fungsi inilah maka muncul berbagai teknik atau metode dalam mengenali suatu pola.

Beberapa metode untuk memecahkan masalah pengenalan tulisan tangan telah disusun, tetapi sebagian besar metode yang digunakan dipinjam dari bidang pengenalan pola, pemrosesan sinyal (*signal processing*), dan analisis gambar (*image analysis*). Metode pengenalan dapat diklasifikasikan menjadi 3 kelompok utama, yaitu: *statistical*, model *matching*, model *structural* dan *sintactical* (Beigi, 1998)

Pada dasarnya metode pengenalan sangatlah mudah, setelah dilakukan proses awal, beberapa ciri kemudian diekstrak dari karakter yang tidak dikenal, kemudian diklasifikasikan ke dalam kelas-kelas yang memiliki banyak kemiripan dengan ciri-ciri tersebut. Tetapi permasalahannya ketika ditemukan gambaran dan

perbedaan ciri, pemilihan cara untuk membandingkannya, dan juga ketika menciptakan aturan-aturan dalam proses pengklasifikasian.

a. *Statistical*

Pada metode *statistical*, pengenalan karakter dilakukan dengan memilih karakter yang mungkin yaitu karakter yang memiliki kesalahan klasifikasi yang minimum. Metode ini, akan menentukan pada kelas mana pola masukan itu berada berdasarkan pengukuran ciri-ciri masukan. Contoh diagram metode *statistical* dapat dilihat di bawah ini:



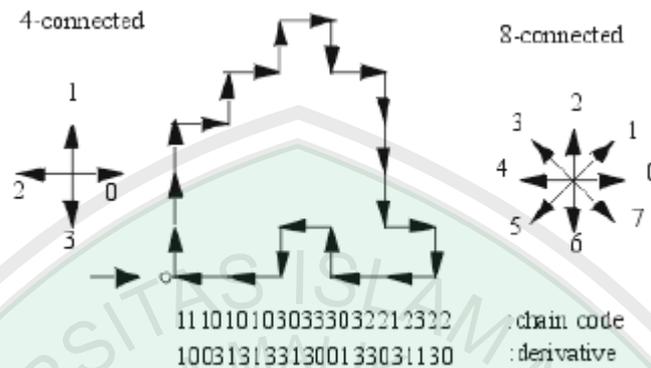
Gambar 2.5 Contoh diagram proses metode statistik

Metode ini memiliki dua bagian utama, yaitu pengambilan ciri dan penentuan kelas suatu pola. Setiap pola masukan akan diproses dengan mengambil ciri-ciri tertentu dari pola tersebut dan akan ditentukan kelasnya. Metode ini memiliki banyak kemiripan dengan *template matching*, tetapi memiliki fleksibilitas yang lebih tinggi. Contoh metode *statistical* sebagai dasar pemecahan masalah pengenalan tulisan tangan, yaitu *chain code* dan *hidden markov model*.

1) **Chain code**

Chain code is used to represent a boundary by a connected sequence of straight line segments of specified length and direction (Gonzales, 1992: 436). Kode rantai adalah notasi untuk mengkodekan senarai tepi yang membentuk batas daerah (Munir, 2004: 154). Representasi ini berdasarkan 4 atau 8 tetangga. Arah

dari masing-masing daerah dikodekan dengan menggunakan nomer seperti gambar di bawah ini.



Gambar 2.6 Arah untuk (a) 4-tetangga (b) 8-tetangga

Batas kode rantai tergantung dari titik awal. Untuk *chain code* dengan posisi awal acak maka dapat dikodekan seperti urutan yang melingkar dari nomor arah dan menjelaskan kembali titik awal, hasil urutan tersebut merupakan integer dari jarak minimum. Kita juga bisa menormalisasi perputaran dengan menggunakan selisih pertama (*first difference*) dari kode rantai sebagai kode itu sendiri. Selisih ini didapat dengan menghitung nomor arah yang terpisah antara 2 elemen yang berdekatan. Misalnya gambar dengan kode 10103322 maka kode gambar setelah di normalisasi 3133030. Jika memilih untuk memperlakukan code sama dengan urutan yang melingkar kemudian elemen I dari selisih dihitung menggunakan perpindahan awal dan akhir komponen rantai. Dari hasil kode gambar setelah di normalisasi diatas maka hasilnya menjadi 33133030. Dari kode tersebut, meskipun titik awalnya berbeda maka akan didapat kode yang sama asalkan bentuk pola dari citra tersebut sama.

Chain code (kode rantai) dapat juga berarti karakter terbagi menjadi segmen-segmen yang dikodekan. segmentasi dapat dilakukan dengan titik lokal

ekstrim (kiri, kanan, atas, bawah, intensitas maksimum dan minimum) atau jarak titik yang sama. kode dapat menampilkan langsung, panjang garis yang sama dengan arah diskret, panjang dari garis lurus dan sudut yang bersambung di antaranya, atau beberapa tipe yang berbeda dari garis, kurva, loop, dan sudut.

Klasifikasi dari kode rantai karakter dapat dilakukan dengan berbagai cara, misal *stastical methods*, *string comparison methods*, atau *dynamic programming*. Metode *statistical* untuk mengklasifikasi karakter kode rantai dijelaskan oleh Loy dan Landay. Pada metode ini, segmentasi dilakukan berdasarkan titik maksimal kurva dan karakter dianggap sebagai poligon vektor tersebut yang terdiri dari jarak antara segmen garis dari titik asal dan sudut antara segmen garis dengan sumbu y.

b. Model Matching

Metode pengenalan model *matching* berdasarkan pada anggapan bahwa karakter tulisan tangan merupakan wujud penyimpangan dari model yang ideal. Penyimpangan tersebut diterangkan pada variasi gaya penulisan, mengubah orientasi skala penulisan, dan noise dari lingkungan dan peralatan. Di dalam OCR, model *matching* yaitu dengan membandingkan 2 gambar, tetapi di dalam pengenalan secara *online* biasanya dengan pencocokan kurva.

1) Pencocokan Model (*Template Matching*)

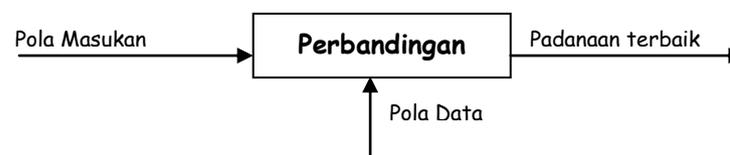
Salah satu cara pendekatan yang mudah dan terawal dalam pengenalan pola adalah pencocokan model (*template matching*). *Template matching* merupakan proses pengenalan karakter dengan cara membandingkan karakter yang ada pada dokumen dengan karakter pada *template*, untuk mencari karakter

yang memiliki nilai kemiripan terbesar dengan *template*. *Template matching* sering digunakan untuk pengenalan karakter hasil *print* dan beberapa objek yang sederhana. Setiap piksel dari karakter memiliki nilai 0 sedangkan *background* dari karakter tersebut memiliki nilai 255. Proses *template matching* dilakukan dengan menggerakkan *template* ke seluruh posisi yang mungkin pada citra.

Metode ini merupakan metode yang paling sederhana dalam pengenalan pola. Metode ini menghitung jarak dari data ke *template* dan juga untuk masing-masing *template*, dan juga membangkitkan nilai kemungkinan dari kemiripan input data dengan *template*, pola dibandingkan dengan pola yang telah ada (Vori, 1998). Pola dibandingkan begitu saja tanpa mencari ciri dari pola itu sendiri. Metode ini akan menghasilkan performansi yang memuaskan bila memenuhi hal-hal berikut:

- a) Jumlah pola data yang berbeda sedikit
- b) Variasi yang ada pada pola yang sama sedikit
- c) Antara pola yang berbeda harus mempunyai perbedaan yang besar

Diagram proses *template matching* dapat dilihat pada gambar di bawah ini



Gambar 2.7 Contoh diagram proses *template matching*

Jika *template* berdasarkan kata yang lengkap dan bukan karakter, kemudian suatu gerakan selalu menjadi sub kelompok dari kata dan mungkin

untuk melakukan pengenalan tidak dibatasi atau kursif .

Sistem pengenalan lain yang didasarkan pada *template matching* telah mengandalkan teori bahwa semua tulisan tangan terbuat dari beberapa bentuk dasar. Proses segmentasi dipotong dari penulisannya sampai pada elemen dasar dan penggolong label tersebut berdasarkan *template* yang memiliki mode sama dengan sistem *stroke matching*. Sistem tersebut dapat digunakan untuk mengarahkan kursif atau pengenalan yang tidak dipaksa dan juga bisa digunakan untuk menurunkan kerumitan pada gaya penulisan.

2) Elastic Matching

Elastic matching biasa disebut dengan *Dynamic Time Warping* (DTW), merupakan metode pencocokan non-linear yang digunakan dalam pengenalan suara. Metode ini mulai dikembangkan sekitar tahun 1970-an dan dikenalkan menjadi metode pengenalan tulisan akhir tahun 1970-an. (Beigi, 1998)

Metode *elastic matching* digunakan dalam pengenalan tulisan tangan tidak bisa merasakan penyimpangan dikarenakan perbedaan kecepatan menulis. Misalnya: variasi dari total angka dan pendistribusian titik sample dalam karakter.

c. Model Structural dan Sintactical

Metode *structural* dan *sintactical* digunakan untuk pengenalan pola yang bermacam-macam dengan dibandingkan dengan struktur pola tertinggi sehingga dapat digunakan dalam pengklasifikasian. Ada dan ketiadaan dari ciri-ciri yang pasti dalam pola tidak selalu cukup informasi untuk pengelompokannya, beberapa informasi tentang hubungan ciri-ciri juga diperlukan. Prinsip dari metode ini

adalah bahwa pola tertentu dapat diekspresikan sebagai komposisi dari pola-pola yang sederhana yang dapat diwakilkan oleh pernyataan-pernyataan seperti garis lurus dan kurva. Hasil dari metode ini dapat optimal jika struktur pola tidak kompleks dan masih dapat diekstrak ke dalam pola dasar pembentuknya.

Di dalam metode *sintactical*, ciri-ciri ditemukan pada pola dengan mempertimbangkan kesederhanaan untuk suatu bahasa formal. Bahasa tersebut terdiri dari syarat-syarat tata bahasa yang mendefinisikan kemungkinan hubungan dari ciri-ciri, atau dengan kata lain mendefinisikan struktur pola (Munir, 2004: 245). Ciri-ciri yang sangat mendasar digambarkan dengan simbol bahasa (*the terminal symbols of the language*).

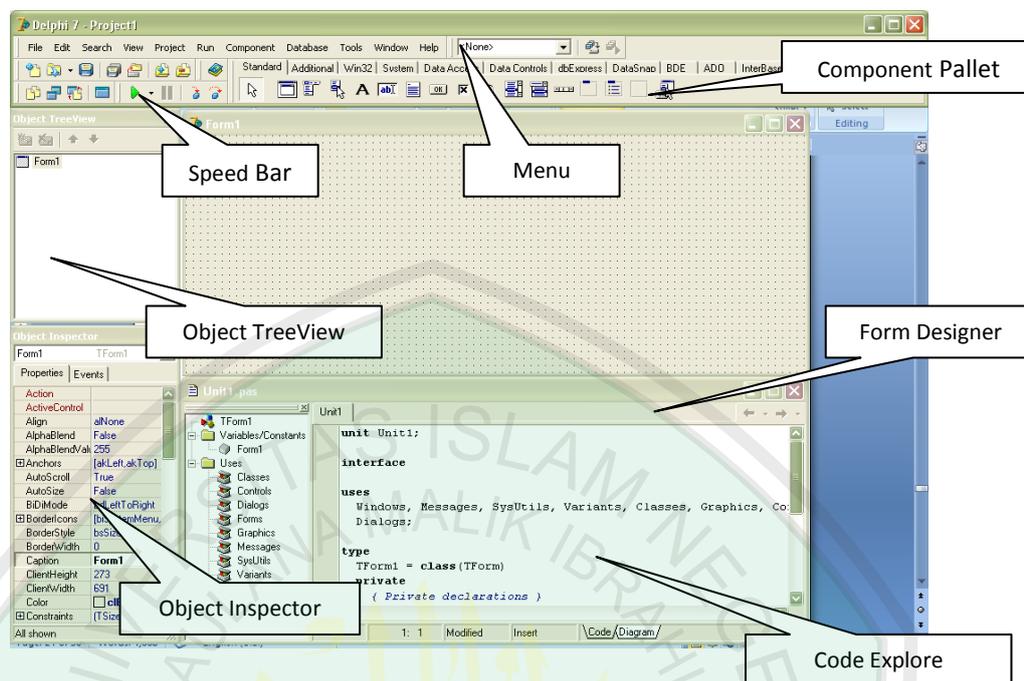
Pada metode *structural* juga berdasar pada analisis ciri dan hubungannya. Yang membedakan dari kedua metode tersebut yaitu pada *syntactical* klasifikasinya tidak menggunakan penguraian tetapi mencocokkan dan dengan berbagai macam aturan untuk mengambil suatu keputusan.

D. BORLAND DELPHI

Delphi merupakan versi visual dari Pascal. Berbagai kemudahan ditawarkan oleh Delphi, mulai dari perancangan aplikasi berbasis form, kemudahan pemberian komponen visual, manipulasi *property* dan *event* yang terintegrasi melalui *object inspector*, sampai *code insight* (Pranata, 2003: 4)

1. Mengenal IDE Delphi

Pada dasarnya IDE Delphi dibagi menjadi tujuh bagian utama, yaitu *Menu*, *Speed Bar*, *Component Palette*, *Form Designer*, *Code Explorer*, *Object Treeview*, dan *Object Inspector*.



Gambar 2.8 Bagian-bagian IDE Delphi

Menu

Menu pada Delphi memiliki kegunaan seperti menu pada aplikasi windows lainnya, semua yang ada berhubungan dengan IDE Delphi dapat dilakukan dari menu. contohnya: memanggil atau menyimpan program, menjalankan program, dan sebagainya.

Speed Bar

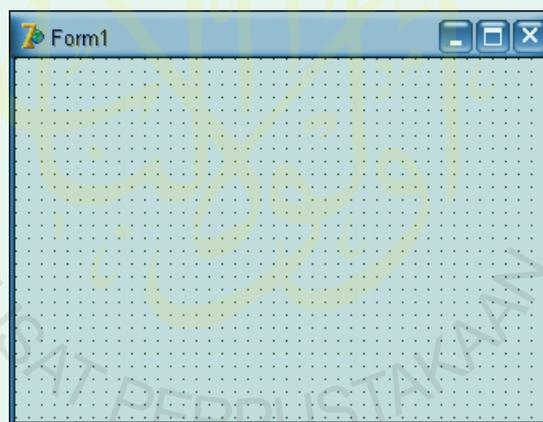
Speed Bar atau sering juga disebut toolbar berisi kumpulan tombol sebagai pengganti beberapa item menu yang sering digunakan. Dengan kata lain, setiap tombol pada speed bar menggantikan salah satu item menu. Sebagai contoh, tombol kiri atas adalah pengganti menu File | New, tombol disebelah kanannya adalah pengganti menu File | Open, dan seterusnya.

Component Palette

Component Palette berisi ikon yang melambangkan komponen-komponen pada VCL (*Visual Component Library*) atau CLX (*Component Library for Cross Platform*). VCL merupakan pustaka komponen yang dengannya dapat membangun aplikasi. Pada Component Palette terdapat beberapa tab, yaitu Standard, Additional, Data Access, dan seterusnya

Form Designer

Form Designer merupakan tempat untuk merancang jendela aplikasi. Perancangan form dilakukan dengan meletakkan komponen-komponen yang diambil dari component palette.



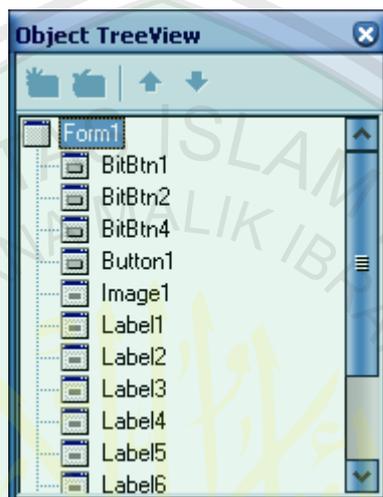
Gambar 2.9 Form Designer

Code Explorer

Code Explorer adalah tempat untuk menuliskan program. Pernyataan-pernyataan tersebut ditulis dalam bentuk bahasa object Pascal. Kita tidak perlu menuliskan semua kode sumber karena Delphi telah menuliskan semacam kerangka program.

Object TreeView

Object TreeView berisi daftar komponen yang telah diletakkan pada Form Designer. Sebagai contoh, jika meletakkan 2 komponen, TGroupBox dan TLabel, Object TreeView terlihat seperti gambar



Gambar 2.10 Object TreeView pada IDE Delphi

Object Inspector

Object inspector digunakan untuk mengubah karakteristik komponen. Pada object inspector terdapat 2 tab yaitu Properties dan Event. Tab tersebut dapat diaktifkan salah satu dengan menklik Properties atau Event.

Pada tab properties, bisa digunakan untuk mengubah property dari komponen yang telah dibuat. Sedangkan pada tab event, dapat digunakan untuk menyisipkan kode dalam menangani kejadian tertentu. Kejadian bisa dibangkitkan karena beberapa hal, seperti pengklikan mouse, penekanan tombol keyboard, penutupan jendela, dan sebagainya. Misalnya `onClick`, yang dibangkitkan bila mengklik form.

2. Struktur File

Tidak seperti Turbo Pascal, Delphi tidak hanya menyimpan file kode dengan ekstensi `.pas`, tetapi karena pada Delphi terdapat form beserta parameternya, maka ada beberapa file yang akan disimpan. Untuk memudahkan, program disebut dengan *Project*. *Project* tersebut akan berisi form, *source code* untuk form, dan *source code* untuk *project*.

Untuk form akan diberi unit, yang akan berisi kode-kode program untuk memanipulasi form tersebut, termasuk untuk *event-event* yang dimiliki oleh form tersebut.

Beberapa file yang terbentuk ketika program Delphi disimpan dalam hardisk, antara lain:

- 1) *Project file* (***.dpr**) adalah file proyek yang dibuat berisi program kecil, berisi program utama dari aplikasi yang telah dibuat untuk:
 - Mendefinisikan Unit yang ada dalam file proyek
 - Menginisialisasi data
 - Membangun form
 - Menjalankan aplikasi
- 2) *Unit file* (***.pas**) adalah unit-unit yang nantinya digunakan untuk menangani kejadian pada form, bisa terdiri satu atau banyak file. File ini berisi *source code* dari obyek-obyek Pascal maupun perintah-perintah yang ingin ditulis
- 3) *Form file* (***.dfm**) merupakan file binary yang merepresentasikan gambar dari form/tampilan yang kita buat. File ini biasanya bergabung dengan file

.pas. Untuk penambahan form baru, akan selalu dibuat file form/*dfm* dengan file *pas*.

- 4) *Resource file (*.res)* adalah file yang berisi resource, biasanya ikon, tetapi dapat juga kursor, bitmap, dll
- 5) *Option (*.dof)* dan *konfigurasi (*.cfg)* adalah file konfigurasi untuk proyek yang telah dibuat. File ini dapat diubah konfigurasi proyeknya melalui menu Project | Options

3. Code Editor

Tiap form diberikan satu file unit dan file unit itulah yang dipakai untuk menulis kode program yang berhubungan dengan form. Selain itu, setiap kali form baru di desain, maka secara otomatis pada code editor akan tampil baris-baris kode seperti di bawah ini:

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs;

type
  TForm1 = class (TForm)
  private
    {Private declarations}
  public
    {Public declarations}
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

end.

```

Baris kode tersebut disediakan oleh Delphi yang menandakan bahwa form yang akan didesain secara default merupakan *TForm1* (default object formnya adalah *Form1*), yang merupakan kelas turunan dari *TForm*. Selain itu disediakan pula bagian *private* dan *public* dari kelas *TForm1* tersebut.

Bagian *private* dan *public* inilah yang bisa dimanipulasi langsung oleh pemrogram, dengan menambahkan *fields*, *methods*, maupun *property*. Jangan mengubah deklarasi *fields* maupun *methods* yang ada di atas bagian *private*, karena bagian ini akan diubah secara otomatis oleh Delphi setiap kali pemrogram menambahkan komponen atau *event* baru.

Untuk penulisan kode program tambahan dapat dilakukan seperti halnya memanipulasi unit pada Turbo Pascal. Elemen-elemen yang dilarang diubah adalah:

1. Antara baris *TForm1 = class (TForm)* dengan baris *private*
2. Deklarasi variabel form, yaitu baris **var** dengan baris *Form1:TForm1*;

Adapun fungsi dari perintah standard pada *public* dan *private* adalah:

1. Perintah standard pada *public* digunakan untuk:
 - Mendeklarasikan field data di mana kita menginginkan metode di obyek luar unit lain dapat mengaksesnya
 - Mendeklarasikan metode di mana obyek di unit lain dapat mengaksesnya
2. Perintah standard pada *private* digunakan untuk:
 - Mendeklarasikan field data di mana hanya metode di dalam file unit aktif (*current unit file*) yang dapat mengaksesnya
 - Mendeklarasikan metode jika hanya obyek yang didefinisikan di file unit

aktif yang dapat mengaksesnya

E. FLOWCHART

Flowchart adalah suatu diagram alur yang menggambarkan logika atau urutan-instruksi program dari suatu permasalahan. Pada diagram alur, dapat dilihat secara jelas arus pengendalian algoritma, yakni bagaimana rangkaian pelaksanaan kegiatan program tersebut. Suatu diagram alur akan memberi gambaran dua dimensi berupa symbol-simbol yang masing-masing symbol tersebut telah ditetapkan lebih dahulu fungsi dan artinya.

Berikut ini seperangkat symbol diagram alur beserta fungsi dan manfaatnya yang digunakan dalam pembautan langkah algoritma pemrograman, yaitu:



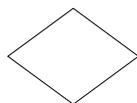
Process/Proses, satu atau beberapa himpunan penugasan yang akan dilaksanakan secara berurutan



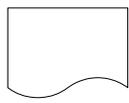
Input, data yang akan dibaca dan dimasukkan ke dalam memori computer dari suatu alat input atau data dan harus melewati memori untuk dikeluarkan dari alat-alat output



Terminal, berfungsi sebagai awal (berisi *Start*) dan juga sebagai akhir (berisi *End*) dari suatu proses alur



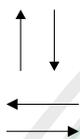
Decision, atau kotak keputusan berfungsi untuk memutuskan arah atau percabangan yang diambil sesuai dengan kondisi yang dipenuhi, yakni benar atau salah



Output/Print, berfungsi untuk mencetak (dan/atau menyimpan) hasil output/keluaran



Conector/penghubung, sebagai penghubung bila diagram alur terputus disebabkan misalnya oleh pergantian halaman (misal diagram tidak cukup dalam satu halaman)



Flowline, menunjukkan bagian arah instruksi dijalankan

Selain simbol-simbol diatas masih banyak lagi simbol-simbol lain yang kesemuanya dapat dilihat atau tergambar pada *template*, yaitu alat penggaris khusus untuk menggambarkan simbol dan kotak diagram alur

BAB III

DESAIN DAN PERANCANGAN

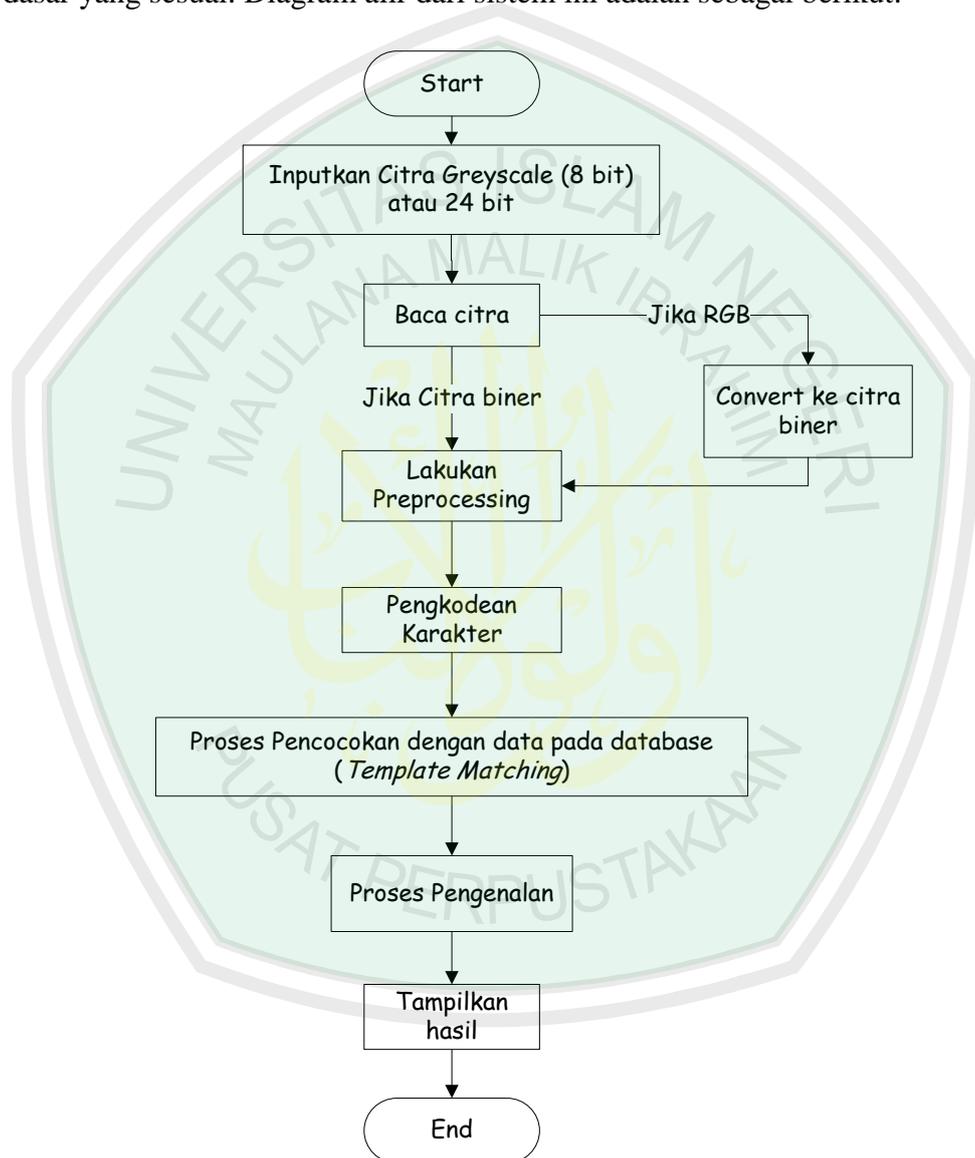
Bab ini membahas tentang desain dan perancangan metode *template matching* dalam pemisahan tulisan Arab pada citra *grayscale* ataupun *true color*. Desain dan perancangan ini meliputi deskripsi sistem, desain data, desain proses, dan desain antarmuka. Selain itu dijelaskan juga proses-proses yang digunakan dalam metode ini serta penjelasan fungsi-fungsi dan parameternya.

A. DESKRIPSI SISTEM

Subbab ini akan membahas mengenai deskripsi sistem yang dikerjakan pada skripsi. Tujuan pembuatan sistem ini adalah untuk memisahkan tulisan Arab sambung menjadi pola huruf dasarnya atau dapat dikatakan sebagai huruf hijaiyah. Pada awalnya pengguna memasukkan input data berupa citra. Citra masukan adalah citra *grayscale* 8 bit atau citra warna 24 bit. Kemudian pengguna diminta untuk melakukan *preprocessing* yaitu proses awal agar citra tersebut dapat diproses selanjutnya. Jika semua operasi pada *preprocessing* telah dikerjakan, maka sistem siap melakukan proses pengkodean yang selanjutnya akan dicocokkan dengan data dalam database.

Implementasi metode pengenalan pola huruf dasar ini terdiri dari 3 tahap utama untuk memisahkan tulisan Arab sambung yaitu proses awal (*preprocessing*), pengkodean (*feature extraction*), dan pengenalan (*recognition*). Pada tahap awal, citra masukan dilakukan proses binerisasi (hitam-putih), penipisan (*thinning*), pemisahan/pembagian menjadi perkarakter (*segmentation*),

dan penskalaan (*scalling*). Selanjutnya pada tahap pengkodean, citra akan dikodekan menurut pikselnya. Kode tersebut ditampilkan dan diproses kembali yaitu dicocokkan dengan data pada database sehingga menghasilkan huruf Arab dasar yang sesuai. Diagram alir dari sistem ini adalah sebagai berikut:



Gambar 3.1 Diagram alir sistem secara keseluruhan

Sesuai dengan gambar 3.1, jika pengguna tidak memasukkan citra *biner* (1bit) maka citra akan dirubah dulu dalam bentuk citra *biner* (*hitam-putih*).

Kemudian dilakukan *preprocessing* yang menghasilkan citra berukuran 1 piksel dan berwarna hitam putih. Hasil citra dari proses tersebut, selanjutnya dikodekan dan dicocokkan dengan data yang ada pada database. Pencocokan yang telah dilakukan telah mengenalkan huruf tersebut, sehingga menghasilkan dan menampilkan citra yang diinginkan yaitu sama seperti huruf dasar pembentuknya.

B. DESAIN SISTEM

Pada sub bab ini akan dijelaskan mengenai desain aplikasi sistem untuk implementasi metode *template matching*. Desain aplikasi ini meliputi desain data, algoritma yang digunakan dalam sistem yang digambarkan dengan diagram alir, desain proses. Desain data berisikan penjelasan data yang diperlukan untuk menerapkan beberapa metode pada pengolahan citra dan pengenalan ini. Desain data meliputi data masukan, data selama proses dan data keluaran. Desain proses antara lain menjelaskan tentang proses awal (*preprocessing*), pengkodean citra, dan pengenalannya. Pada proses awal, terdiri dari konversi ke hitam-putih, *cropping*, dan *thinning*. Pada proses pengkodean, lebar dan tinggi citra masukan dibagi menjadi 15. Setelah itu, dicek jika *piksel* yang dilewati berwarna hitam atau nilai *threshold*-nya kurang dari 180 maka *piksel* tersebut dikodekan sebagai angka 1. Sebaliknya, jika setelah dicek *piksel* yang didapat berwarna putih atau nilai *threshold*-nya lebih dari 180 maka *piksel* dikodekan sebagai angka 0. Sedangkan pada proses pengenalan langkah yang dilakukan adalah membandingkan dan memcocokkan hasil kode yang telah diproses dengan kode yang sudah ada, inilah yang disebut dengan *template matching*.

1. Desain Data

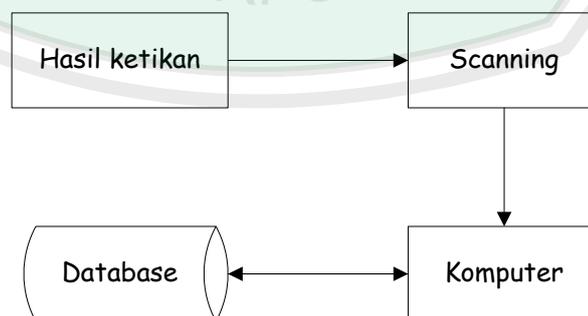
Data yang digunakan untuk implementasi perangkat lunak ini dibagi menjadi tiga bagian utama, yaitu data masukan, data yang digunakan selama proses perbaikan citra *preprocessing* dan data keluaran.

2. Data Masukan

Data masukan yang pertama dari pengguna adalah arsip citra yang dipilih oleh pengguna. Pada sistem ini citra yang dimasukkan berupa arsip citra warna 24 bit, dengan format .bmp. Data masukan kedua adalah bentuk citra yang akan dikenal, hal ini dilakukan pada operasi pemisahan karakter Arab dari kata (*segmentation*), yaitu dengan cara *men-drag* mouse dari kiri atas ke kanan bawah.

3. Data Selama Proses

Pada tahap proses *binerisasi* untuk menghilangkan atau setidaknya mengurangi noise-noise yang terdapat pada citra setelah melalui proses *scanning*. Disamping itu, proses ini juga bertujuan untuk memisahkan warna tulisan (*foreground*) dan warna belakangnya (*background*). Sehingga citra asli yang akan diolah tampak lebih jelas.

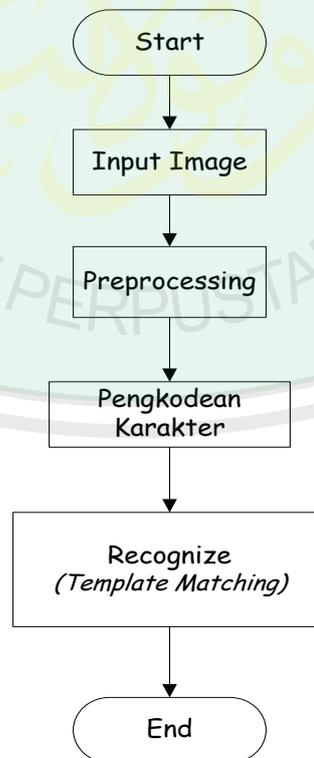


Gambar 3.2 Diagram Perancangan Sistem

Adapun uraian dari gambar diatas adalah sebagai berikut: data tulisan Arab yang akan diproses, dilakukan proses *scanning* terlebih dahulu dengan menggunakan alat yang disebut *scanner* dan dengan format bitmap (*.bmp). Setelah data tersebut diperoleh berupa data gambar dengan format bitmap, langkah selanjutnya yaitu pemrosesan pada komputer. Pada komputer data tersebut akan diproses sampai menghasilkan kode-kode yang akan disimpan dalam database. Selanjutnya kode-kode tersebut digunakan sebagai pembanding dengan data-data yang telah ada pada database.

C. PENGENALAN TULISAN ARAB

Langkah-langkah proses dari pengenalan tulisan Arab ini dapat dilihat pada gambar 3.2 di bawah ini



Gambar 3.3 Flowchart proses pengenalan tulisan Arab

Dari flowchart proses pengenalan tulisan Arab diatas dapat dilihat bahwa pada proses pengenalan tulisan Arab terdapat beberapa proses yang saling berhubungan dan ketika salah satunya tidak dijalankan maka tidak akan tercapai suatu pengenalan tulisan seperti yang diharapkan. Proses tersebut diawali dengan input citra, *preprocessing*, pengkodean, recognize, dan hasil akhir yang berupa huruf dasar Arab dari tulisan Arab sambung yang telah diinputkan. Secara lengkap urutan prosesnya sebagai berikut:

1. Input image

Input image merupakan proses yang pertama kali dilakukan untuk mendapatkan data citra yang akan diproses selanjutnya. Dalam tahap ini dilakukan suatu proses pengubahan suatu citra digital. Citra digital adalah citra yang diekspresikan oleh sekumpulan bilangan sehingga dapat diproses oleh komputer.

Dalam skripsi ini, pengenalan dilakukan secara *offline* yaitu tulisan Arab diketik menggunakan font *Arabic Transparent* ukuran 48 setelah itu di print pada selembar kertas selanjutnya di scan. Ada beberapa batasan yang diberikan yaitu:

- Tulisan Arab ditulis dengan font *Arabic Transparent* ukuran 48
- Untuk tanda baca dan angka Arab tidak diikutsertakan dalam pengenalan, kecuali letak dan jumlah titik sebagai pembeda antara huruf yang satu dengan yang lain.

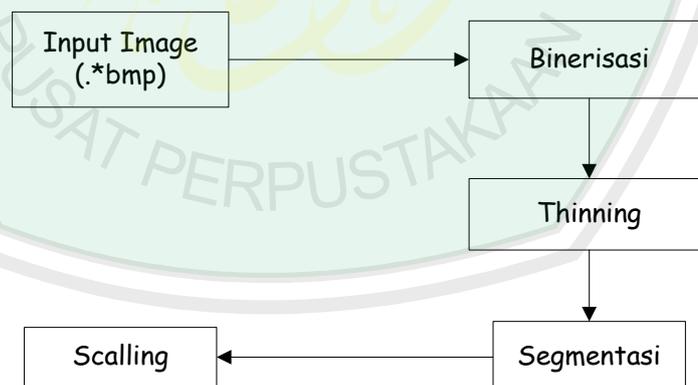
Setelah data diperoleh, data tersebut harus diubah menjadi data yang dapat diolah oleh program. Langkah selanjutnya adalah mengubah data tersebut menjadi file gambar dengan menggunakan scanner. Adapun batasan-batasan file gambar yang diperoleh dari *scanner* adalah:

Kertas yang discan harus menempel semuanya di mesin *scanner*, tidak ada kerusakan-kerusakan, seperti: tertekuk, robek, berlubang, basah, dan lain-lain. Sehingga dapat meminimalisir noda atau gangguan hasil scan tersebut

Dari *scanner* tersebut didapat suatu gambar dengan format bitmap 24 bit. Setelah data didapat, langkah selanjutnya yaitu dilakukan tahap *preprocessing*.

2. *Preprocessing*

Sebelum suatu citra mengalami pemrosesan lebih lanjut, perlu dilakukan proses awal (*preprocessing*) terlebih dahulu, yaitu pengolahan citra (*image*) dengan tujuan mendapatkan gambar dengan pola yang dapat dikodekan dengan menggunakan metode *template matching* yang menghasilkan kode tertentu menurut warna pikselnya. Sehingga dapat digunakan dalam program untuk pengenalan tulisan Arab ini. Tahapan-tahapan pada proses ini dapat dilihat pada diagram di bawah ini:



Gambar 3.4 Diagram *preprocessing*

a. Binerisasi

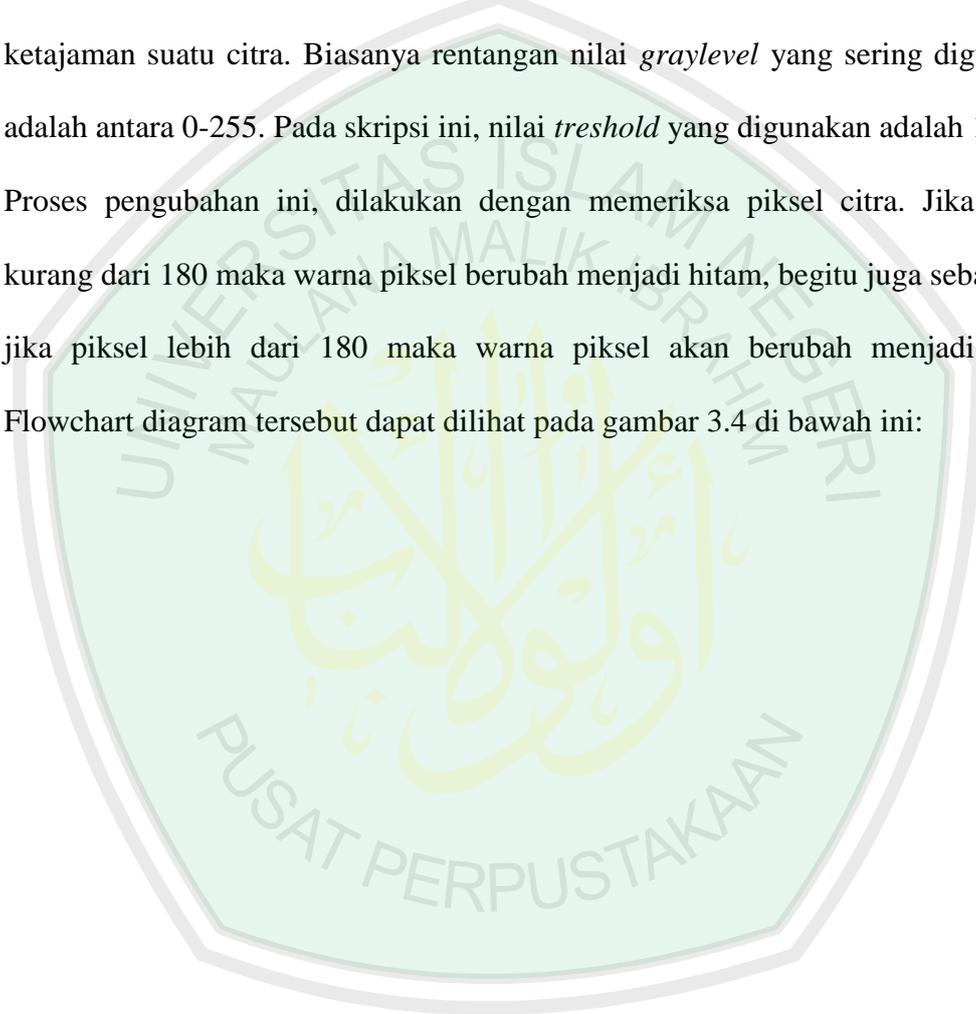
Pengkonversian citra *true color* (24 bit) menjadi citra biner (1 bit) dilakukan dengan operasi pengambangan (*thresholding*). Operasi pengambangan

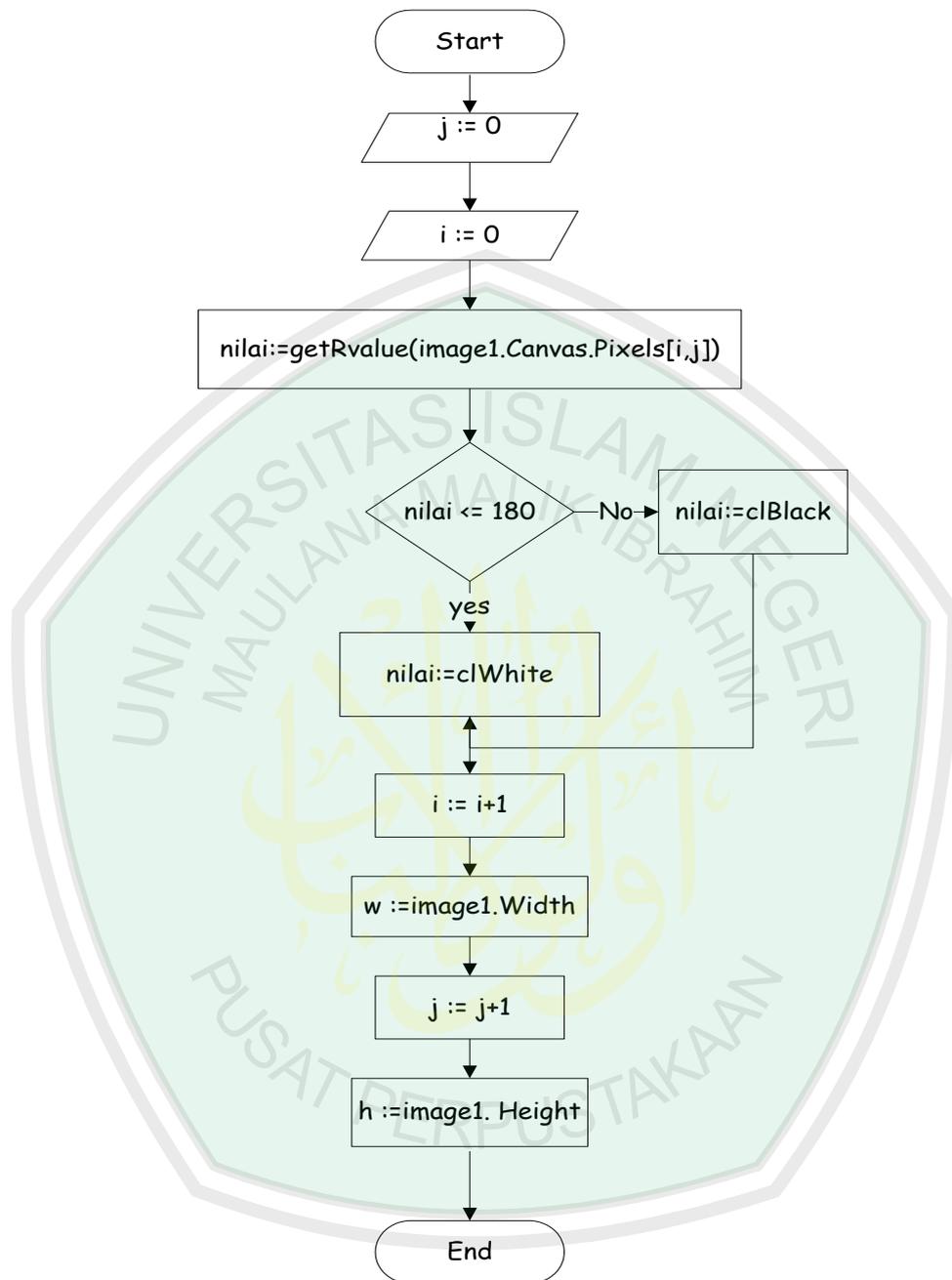
mengelompokkan nilai derajat keabuan setiap piksel ke dalam 2 kelas, hitam dan putih. Di mana hitam sebagai warna objek dan putih sebagai warna latar belakangnya.

Pemilihan nilai *threshold* yang digunakan berpengaruh terhadap ketajaman suatu citra. Biasanya rentangan nilai *graylevel* yang sering digunakan adalah antara 0-255. Pada skripsi ini, nilai *threshold* yang digunakan adalah 180.

Proses pengubahan ini, dilakukan dengan memeriksa piksel citra. Jika piksel kurang dari 180 maka warna piksel berubah menjadi hitam, begitu juga sebaliknya jika piksel lebih dari 180 maka warna piksel akan berubah menjadi putih.

Flowchart diagram tersebut dapat dilihat pada gambar 3.4 di bawah ini:





Gambar 3.5 Flowchart proses binerisasi

b. Thinning

Setelah melalui proses binerisasi, dalam hal ini membuat citra menjadi berwarna hitam putih. Maka proses selanjutnya yaitu *thinning* atau penipisan.

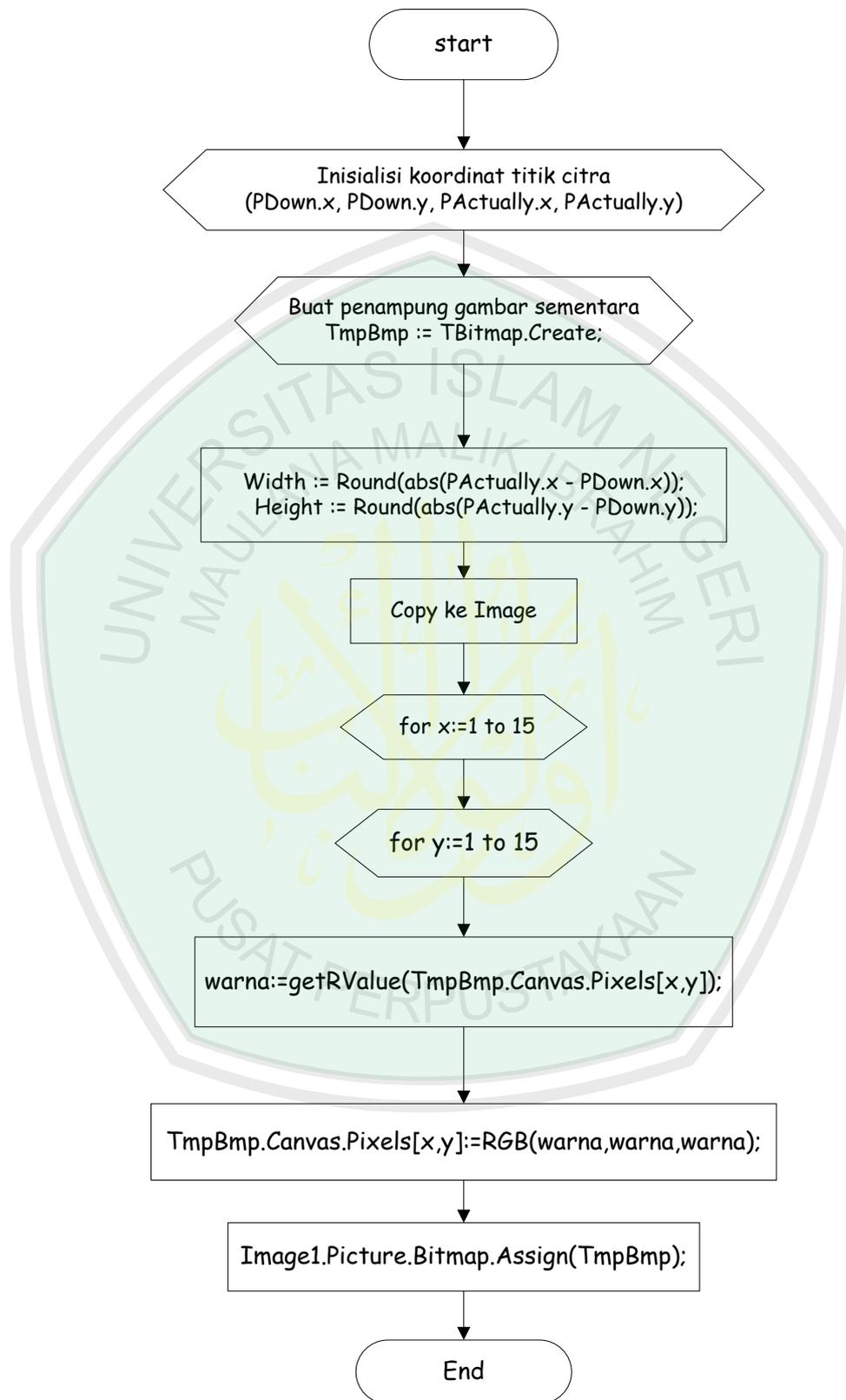
Pada citra tersebut dilakukakan *thinning* untuk mendapatkan gambar dengan ukuran 1 piksel.

Adapun manfaat setelah mendapatkan citra 1 piksel yaitu agar kode tersebut mempunyai ukuran piksel yang sama meskipun tebal dan tipisnya berbeda-beda.

c. Segmentation

Segmentasi karakter merupakan proses pembagian daerah dalam suatu *image* untuk dikelompokkan ke dalam segmen-segmen tertentu. *Image* yang berisi kalimat akan dipisahkan menjadi karakter-karakter. Hasil dari proses segmentasi ini adalah karakter-karakter yang telah berdiri sendiri dan tidak menjadi bagian dari kalimat. Salah satu cara untuk melakukan segmentasi karakter adalah dengan melakukan proses *cropping*, karena pada tulisan Arab ketika disambungkan terjadi bentuk yang berbeda dari bentuk dasarnya. Pada skripsi ini, huruf Arab yang disambung terdiri dari 4 bentuk yang berbeda, antara lain: di awal, di tengah, di akhir, dan ketika huruf tersebut berdiri sendiri. Selain alasan tersebut, pada kata ataupun kalimat Arab terdapat lekukan-lekukan yang sulit untuk dipisahkan. Jika kita salah memisahkan mungkin pengenalannya pun akan berbeda pula.

Meskipun memotongnya tidak sesuai dengan bentuk dasarnya, atau terjadi pengurangan bentuk dasarnya. Program ini akan mencocokkannya ke bentuk karakter huruf Arab yang lebih banyak kesamaannya dari data yang telah disimpan dengan kode masukan, jika kode tersebut masih belum dikenal maka bisa dimasukkan dalam database sebagai kode baru. Flowchart dari proses *cropping* tersebut adalah sebagai berikut:



Gambar 3.6 Flowchart proses *cropping*

d. *Scaling*

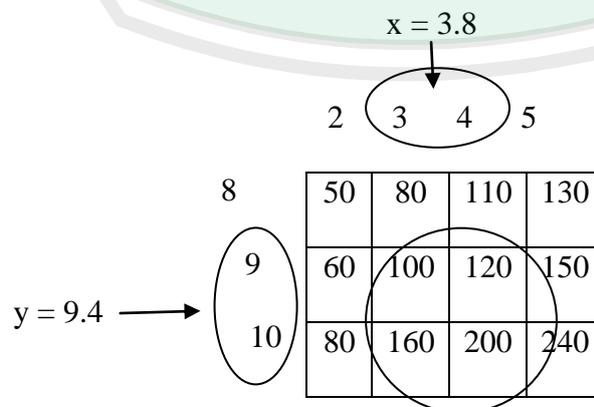
Setelah memisahkan karakter (*segmentation*) menggunakan *cropping*, maka langkah selanjutnya yaitu melakukan proses *scalling*. Adapun proses *scalling* bertujuan untuk memperbesar atau memperkecil *image* agar kode yang akan dihasilkan memiliki lebar dan tinggi yang sama.

Operasi *scalling* pada skripsi ini menggunakan interpolasi bilinear, dikarenakan citra yang dihasilkan akan terlihat lebih halus (*smooth*) dibanding dengan hasil interpolasi tetangga terdekat, terutama jika dilakukan pembesaran atau *zoom in* citra.

Untuk mencari nilai keabuan hasil menggunakan persamaan sebagai berikut:

$$K_o[x_1, y_1] = w_{xL} * w_{yT} * K_i[x_L, y_T] + (1 - w_{xL}) * w_{yT} * K_i[x_R, y_T] + w_{xL} * (1 - w_{yT}) * K_i[x_L, y_B] + (1 - w_{xL}) * (1 - w_{yT}) * K_i[x_R, y_B]$$

Karena nilai keabuan merupakan bilangan bulat, apabila penjumlahan berbobot tersebut berupa bilangan pecah, maka harus dilakukan pembulatan ke bilangan bulat terdekat. Makin dekat titik tetangga tersebut, makin besar bobotnya, dan sebaliknya makin jauh akan makin kecil bobotnya.



Gambar 3.7 Nilai Keabuan citra asal pada operasi geometri

Dari contoh di atas, bobot ke arah horisontal untuk koordinat $x = 3$ dan $x = 4$ masing-masing adalah:

$$w_x(3) = 0.2 \text{ dan } w_x(4) = 0.8$$

koordinat $x = 3.8$ lebih dekat ke $x = 4$ dibandingkan ke $x = 3$. sementara bobot ke arah vertikal untuk koordinat $y = 9$ dan $y = 10$ masing-masing adalah:

$$w_y(9) = 0.6 \text{ dan } w_y(10) = 0.4$$

jumlah bobot untuk tiap arah adalah 1. kombinasi antara bobot horisontal dan bobot vertikal memberikan bobot untuk tiap titik:

$$w(3, 9) = w_x(3) * w_y(9) = 0.2 * 0.6 = 0.12$$

$$w(3, 10) = w_x(3) * w_y(10) = 0.2 * 0.4 = 0.08$$

$$w(4, 9) = w_x(4) * w_y(9) = 0.8 * 0.6 = 0.48$$

$$w(4, 10) = w_x(4) * w_y(10) = 0.8 * 0.4 = 0.32$$

Terlihat bahwa titik yang paling dekat dengan koordinat titik asal hasil transformasi balik memiliki bobot yang paling besar (0,48), sedangkan titik terjauh memiliki bobot paling kecil (0,08). total bobot untuk keempat titik tersebut adalah 1.

Setelah semua bobot pada keempat titik bertetangga tersebut diperoleh, nilai keabuan hasil dihitung dengan rumus

$$\begin{aligned} K_o &= \sum w * Ki = w(3, 9) * Ki(3, 9) + w(3, 10) * Ki(3, 10) \\ &\quad + w(4, 9) * Ki(4, 9) + w(4, 10) * Ki(4, 10) \\ &= 0.12 * 100 + 0.08 * 160 + 0.48 * 120 + 0.32 * 200 \\ &= 146.4 \approx 146 \end{aligned}$$

3. Pengkodean Karakter (*Feature extraction*)

Feature extraction merupakan salah satu cara untuk mengenali karakter dengan melihat ciri-ciri khusus yang dimiliki oleh karakter tersebut. Oleh karena itu, setelah melakukan proses *scalling* proses selanjutnya adalah *feature extraction*. Pada proses ini, citra masukan dibagi menjadi 16 bagian yang sama tinggi dan lebarnya. Setelah itu dikodekan menurut pikselnya, jika berwarna hitam atau kurang dari nilai *treshold* maka citra tersebut dikodekan 1 sebaliknya jika berwarna putih dan lebih dari nilai *threshold* maka dikodekan 0. Tujuan dari proses ini yaitu untuk memberi kode yang berbeda pada setiap karakter sehingga karakter yang satu dengan karakter yang lain dapat dipisahkan berdasarkan kode yang dimilikinya. Disamping itu, proses pengkodean karakter ini dimaksudkan untuk mengambil ciri (*feature*) dari sebuah huruf.

Pengambilan ciri dalam skripsi ini, menggunakan teknik integral proyeksi yaitu sebuah teknik yang menjumlahkan nilai setiap kolom atau setiap baris. Integral proyeksi tersebut dapat didefinisikan dengan rumus:

$$h(i) = \sum_{j=1}^{nkolom} x(i, j)$$

$$h(j) = \sum_{i=1}^{nbaris} x(i, j)$$

4. Input Data

Input data merupakan proses awal yang dilakukan sebelum membandingkan dengan database. Data di dalamnya masih kosong sehingga perlu diadakan proses penyimpanan data awal yang menyimpan semua data karakter

masukan pada database untuk dijadikan acuan dalam proses pengenalan dengan metode *template matching*.

5. Tabel

Berikut adalah penjelasan mengenai tabel-tabel yang digunakan dalam aplikasi ini, antara lain:

1. Primary key (*) : id
 Nama tabel : tbl_pengenalan
 Fungsi : untuk menyimpan hasil proses awal yang telah dilakukan

Tabel 3.1
Tabel Pengenalan

Field	Tipe	Panjang	Keterangan
ID	Varchar	10	Id huruf
Kode_hors	Varchar	255	Hasil kode secara horisontal
Kode_vert	Varchar	255	Hasil kode secara vertikal

2. Primary key (*) : id
 Nama tabel : tbl_pelatihan
 Fungsi : untuk menyimpan hasil proses pelatihan

Tabel 3.2
Tabel Pelatihan

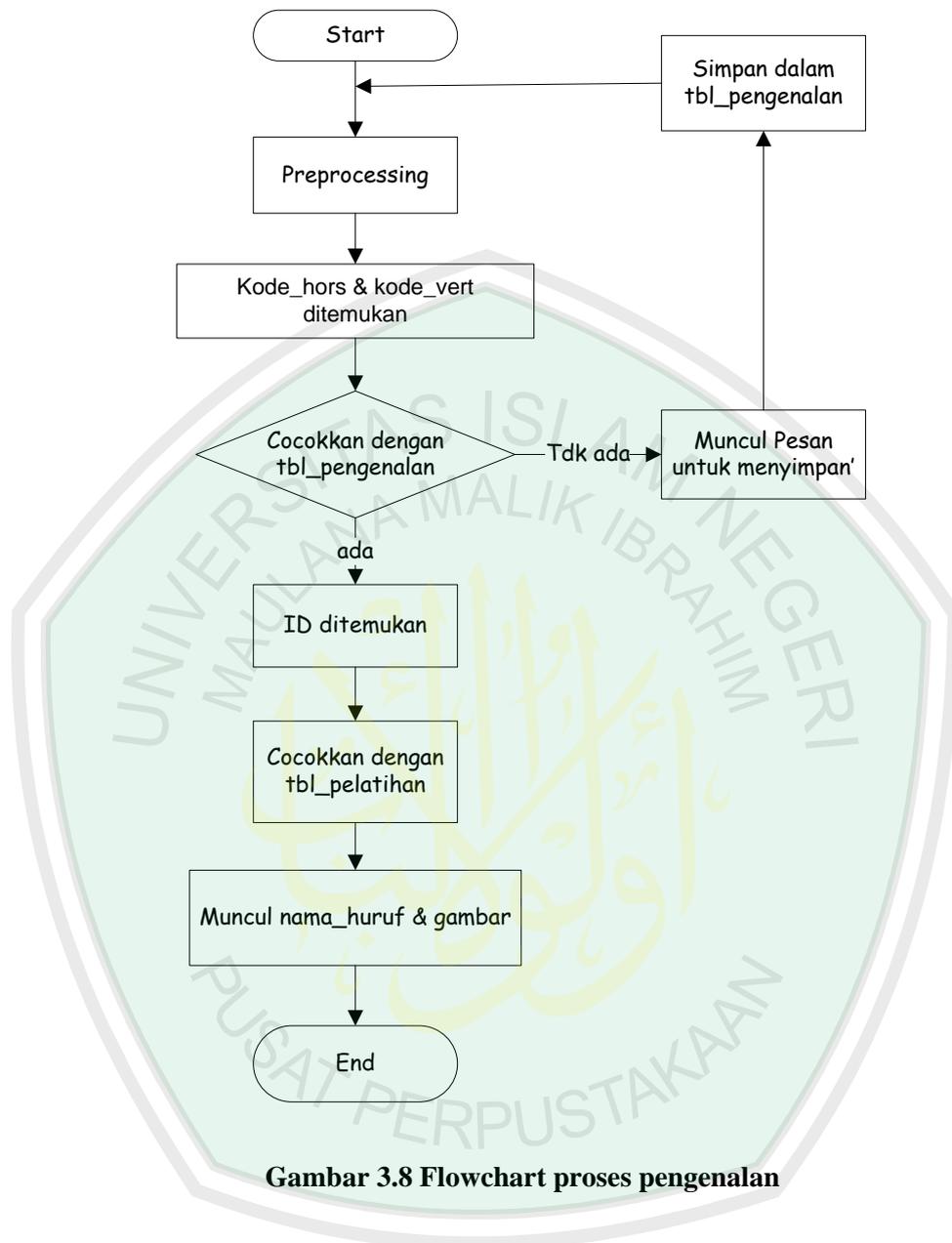
Field	Tipe	Panjang	Keterangan
ID	Varchar	10	Id huruf
Nama_huruf	Varchar	20	Nama dasar huruf
Gambar	OLE Object	-	Gambar dasar huruf

6. Pengenalan (*Recognize*)

Setelah dilakukan proses awal, maka proses selanjutnya adalah membandingkan input tulisan dengan karakter yang telah tersimpan dalam database. Setiap kode gambar yang berukuran 15x15 dibandingkan dengan data yang tersimpan dalam database dengan tujuan untuk mencari beda dari kedua gambar tersebut.

Metode *template matching* terletak pada proses pengenalan ini, dimana 2 tabel yaitu `tbl_pengenalan` dan `tbl_pelatihan` dikorelasikan. Relasi yang digunakan adalah *one to many*, yang berarti bahwa satu tabel dihubungkan dengan banyak tabel. Dalam skripsi ini, masing-masing *field* pada `tbl_pelatihan` hanya berisi 1 data tidak boleh ada yang sama mulai ID, Nama_huruf, sampai dengan gambar. Misalnya: ID 1 memiliki nama huruf alif dan gambar alif, tidak diperkenankan adanya ID 1 dengan nama huruf dan gambar yang lain selain alif. Tabel tersebut, akan dihubungkan dengan `tbl_pengenalan` yang berisi ID yang sama dengan kode yang bermacam-macam tergantung dari hasil integral proyeksi karakter huruf tersebut.

Setelah kode ditemukan, kode tersebut mencari ID yang cocok/sama dalam `tbl_pengenalan`. Jika ID tersebut ditemukan maka `tbl_pengenalan` akan mencocokkan dengan ID yang sama pada `tbl_pelatihan` dan selanjutnya mencari nama_huruf dan gambar yang sesuai dengan ID pada `tbl_pelatihan`. Adapun flowchartnya dapat dilihat seperti di bawah ini:



D. DESAIN ANTARMUKA

Aplikasi ini dibangun dengan desain user interface form yang terdiri dari 4 form, yaitu form utama, form proses awal, form simpan, dan form pelatihan.

1. Desain Menu Tampilan Utama

Pengenalan Tulisan Arab			
Home	Input Data	Preprocessing	Exit

Gambar 3.9 Form Utama

Halaman ini berfungsi sebagai layar utama, setiap user yang menjalankan aplikasi ini, harus menjalankan form ini terlebih dahulu. Pada Halaman tersebut terdapat 4 menu, yaitu:

1. *Home* : untuk kembali ke tampilan awal (form utama)
2. *Preprocessing* : untuk memproses gambar yang masih terdapat *noise* dan ukurannya tidak sesuai dengan yang ditentukan
3. *Input Data* : untuk menyimpan huruf-huruf dasar dari tulisan Arab
4. *Exit* : untuk keluar dari program

2. Desain Menu Tampilan proses awal

Desain tampilan menu proses awal adalah sebagai berikut:

The screenshot shows a window titled "Pemisahan Tulisan Arab Menjadi Pola Huruf Hijaiyah". The menu bar includes "File | Image Process | Code | Recognize | Exit". The main area contains several components: three image boxes labeled "Image", "Image1", and "Image3"; a large "StringGrid1" box; an "ImgHsl" box; and two input fields labeled "Horizontal Code" and "Vertical Code". A large watermark for "UNIVERSITAS ISLAM NEGERI MALIK IBRAHIM" is visible in the background.

Gambar 3.10 Form Proses Awal

Halaman ini berfungsi untuk melakukan proses awal pada gambar. Gambar di *load* dari menu *file* selanjutnya gambar diproses sampai diperoleh nilai kode pada *String grid*. Selanjutnya data kode di simpan dalam button *save*

Menu-menu yang terdapat dalam halaman ini antara lain:

1. *File* terdapat *open file*, *save as*, dan *exit*
2. *Image Process*, terdapat proses *thinning*, *cropping*, dan *scaling*
3. *Code*, terdapat perintah pengkodean yang selanjutnya di tampilkan pada *String grid*
4. *Recognize*, untuk mengenali gambar tulisan yang telah di proses

5. *Exit*, untuk keluar dari halaman proses awal dan kembali ke halaman muka

3. Desain Menu Tampilan Halaman Input Data

Desain tampilan menu halaman pelatihan adalah sebagai berikut:

Gambar 3.11 Form Input Data

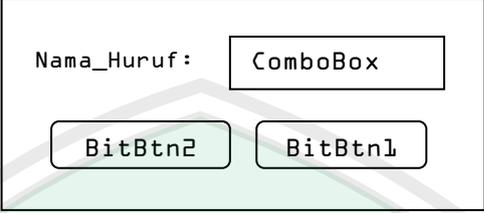
Halaman ini berfungsi untuk melakukan proses pelatihan pada gambar. Masukkan ID, nama huruf, dan gambar

Menu-menu yang terdapat dalam halaman ini antara lain:

1. *Spin Edit1*, digunakan untuk penulisan ID
2. *Edit1 dan Edit2*, digunakan untuk menyimpan nama huruf dan gambar
3. *Image2*, untuk menampilkan gambar yang akan dimasukkan dalam database
4. *Bitbtn1*, untuk tombol save dan *bitbn2* untuk kembali ke halaman utama

4. Desain Menu Tampilan Halaman Simpan

Desain tampilan menu halaman simpan adalah sebagai berikut:



The image shows a rectangular form with a white background and a black border. Inside the form, there are four elements: a label 'Nama_Huruf:' followed by a 'ComboBox' field, and two buttons labeled 'BitBtn2' and 'BitBtn1' positioned below the ComboBox.

Gambar 3.12 Form Simpan

Halaman ini berfungsi untuk melakukan proses simpan setelah ciri dari proses awal selesai dilakukan. Selanjutnya data kode di simpan dalam button *save*

Menu-menu yang terdapat dalam halaman ini antara lain:

1. *ComboBox*, digunakan untuk member nama data yang belum dikenal yang selanjutnya dilakukan proses penyimpanan
2. *Bitbtn1*, digunakan untuk menyimpan nama huruf dan gambar
3. *Bitbtn2*, digunakan untuk membatalkan proses penyimpanan

BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini dibahas mengenai hasil uji coba program yang telah dirancang dan dibuat, serta kontribusi program dalam Islam. Uji coba dilakukan untuk mengetahui apakah program dapat berjalan sebagaimana mestinya dengan lingkungan uji coba yang telah ditentukan serta dilakukan sesuai dengan skenario uji coba.

Ada beberapa hasil uji coba yang telah dilakukan terhadap data yang telah dipilih, antara lain: menguji huruf ain ketika berada di depan, tengah, dan belakang. Sebelumnya perlu diketahui lingkungan uji coba yang digunakan dalam melakukan uji coba dalam tugas akhir ini.

A. LINGKUNGAN UJI COBA

Pada subbab ini dijelaskan mengenai lingkungan uji coba yang meliputi perangkat lunak dan perangkat keras yang digunakan. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam uji coba antara lain adalah:

Tabel 4.1
Lingkungan uji coba

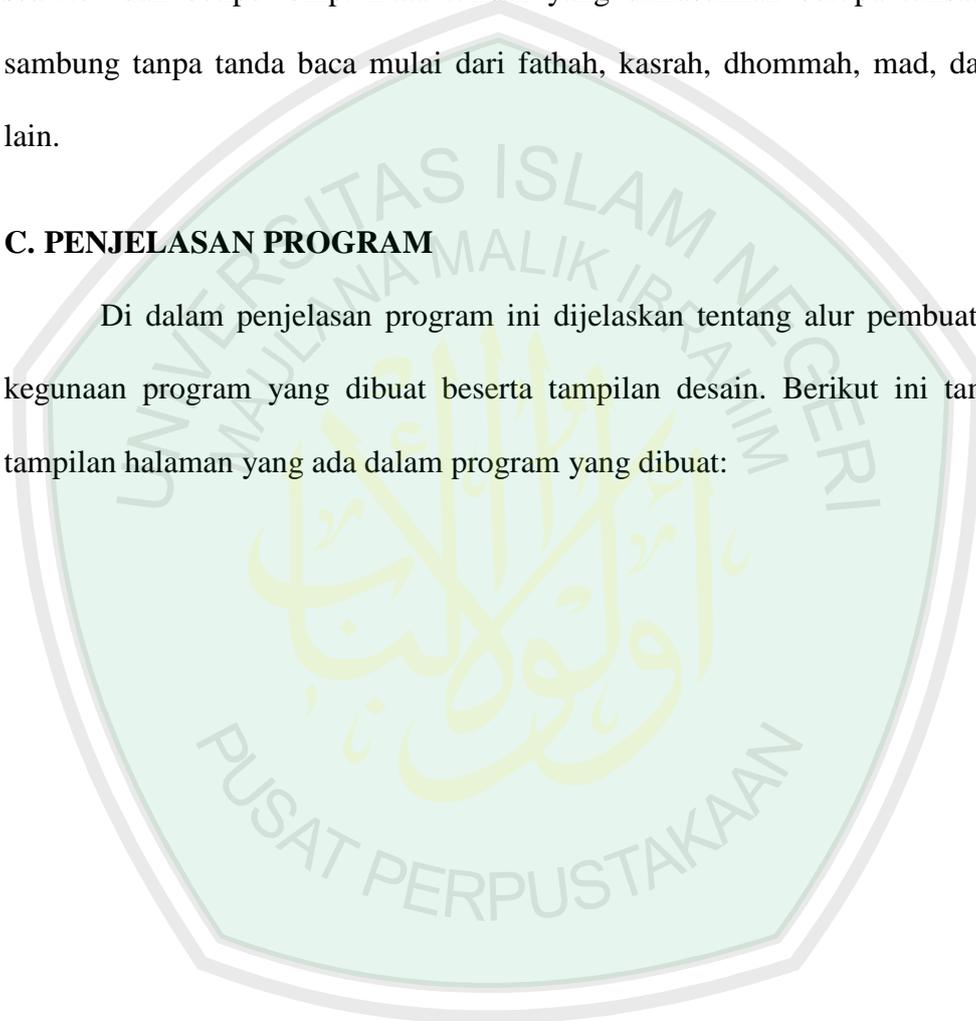
Perangkat Keras	Prosesor : Intel Pentium 4 2.66 GHz Memori : 1 GB Peranti Masukan : - Scanner HP PSC 1410 - Mouse - Keyboard
Perangkat Lunak	Sistem Operasi : Microsoft Windows XP Professional 2002 SP 2 Perangkat Pengembang : Borland Delphi 7.0

B. DATA UJI COBA

Pada uji coba yang akan dilakukan, digunakan data yang berasal dari ketikan komputer yang telah di *print* dan selanjutnya di *scan* menggunakan alat *scanner* dan betipe .bmp. Data tulisan yang dimasukkan berupa tulisan Arab sambung tanpa tanda baca mulai dari fathah, kasrah, dhommah, mad, dan lain-lain.

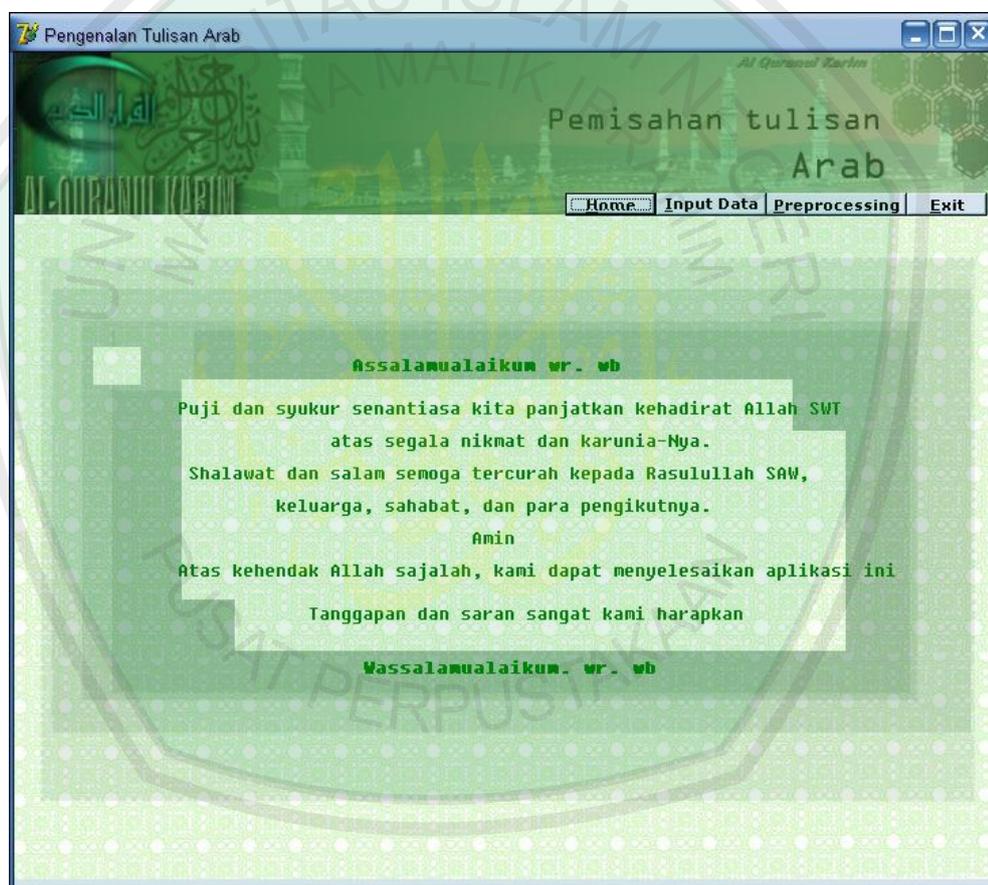
C. PENJELASAN PROGRAM

Di dalam penjelasan program ini dijelaskan tentang alur pembuatan dan kegunaan program yang dibuat beserta tampilan desain. Berikut ini tampilan-tampilan halaman yang ada dalam program yang dibuat:



1. Proses Menampilkan Halaman Utama

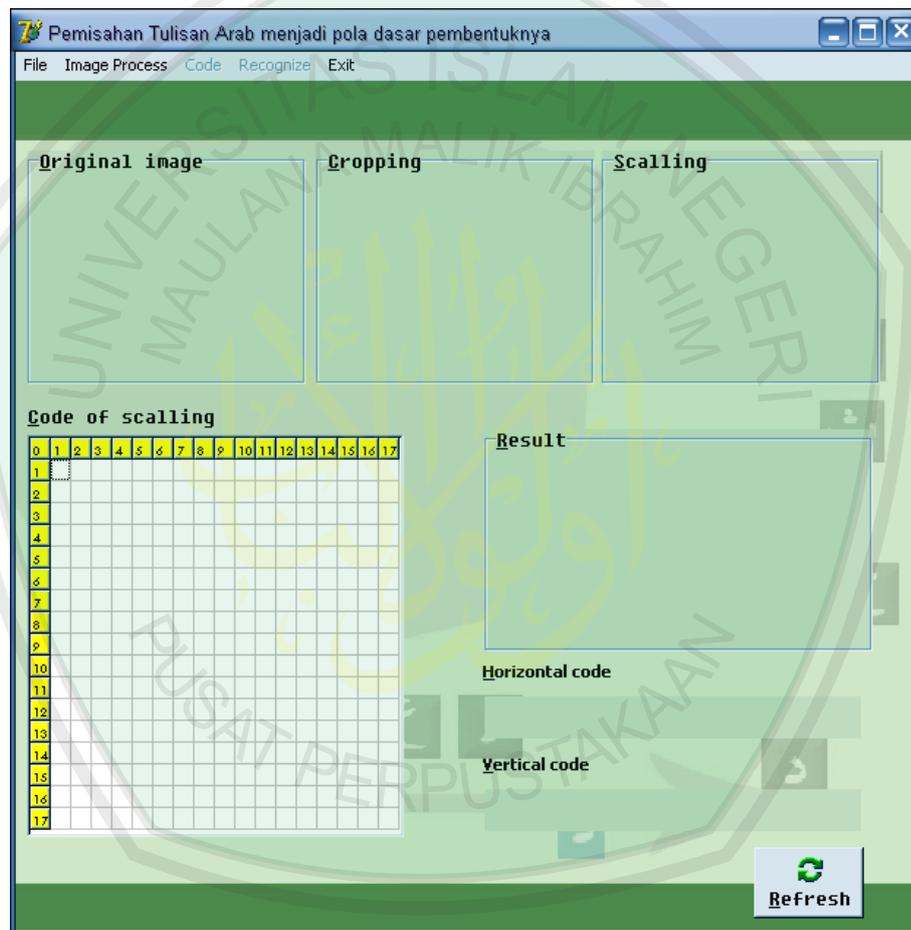
Halaman *home* merupakan halaman utama yang akan pertama kali diakses oleh pengguna. Informasi yang ditampilkan adalah pilihan menu untuk menuju proses selanjutnya. Pada halaman utama terdiri dari 4 menu yang akan ditampilkan sesuai dengan fasilitas yang disediakan perangkat lunak. Tampilan halaman utama ditunjukkan pada gambar 4.1 berikut



Gambar 4.1 Form Utama

2. Proses Menampilkan Halaman Proses Awal

Halaman pada proses awal menjelaskan mengenai proses awal yang harus dilakukan sebelum melakukan proses pemisahan tulisan Arab sambung. Adapun tampilan halamannya adalah:



Gambar 4.2 Form Image Process

Pada halaman proses awal terdapat 3 proses antara lain: *input image*, *preprocessing*, dan pengkodean. Penjelasannya sebagai berikut:

a. Input Image

Pada proses ini terjadi perubahan citra tulisan hasil *scanning* ke dalam citra digital yang dapat dibaca dan diproses selanjutnya oleh komputer. Adapun citra tulisan hasil *scanning* dapat dilihat pada gambar 4.3 dibawah ini:



Gambar 4.3 Contoh Input Image kata 'aina'

Listing program untuk proses Input Image:

```
procedure TForm2.OpenFile1Click(Sender: TObject);
begin
    image.Picture.Bitmap:=nil;
    if openPictureDialog.Execute then
image.Picture.LoadFromFile(openPictureDialog.FileName);
        image.AutoSize:=true;
        binerisasi;
end;
```

b. Preprocessing

Tahap *preprocessing* terdiri dari 4 proses, yaitu binerisasi, *thinning*, *segmentation*, dan *scalling*

1) Binerisasi

Pada tahap ini citra masukan hasil *scanning* dirubah menjadi citra biner (hitam-putih). Warna hitam sebagai warna objek dan putih sebagai warna latar belakang dari gambar. Proses binerisasi ini juga bermanfaat untuk menghilangkan sedikit *noise* yang diakibatkan pada saat proses *scanning*.

Proses ini dilakukan dengan memeriksa piksel gambar, jika piksel kurang dari nilai *threshol*d maka warna dikodekan menjadi 1 dan juga sebaliknya. Jika

warna lebih dari 150 maka dikodekan dengan 0. Cara menentukan tergantung dari masing-masing orang, tetapi biasanya nilai *threshol*d dihitung dari hasil bagi warna *true color* menjadi 2 (256 dibagi 2), semakin besar nilai *threshol*d maka warna gambar akan semakin tajam. Nilai *threshol*d yang digunakan adalah 180, dalam program bisa ditulis RGB(180, 180, 180). Program lengkap proses binerisasi adalah sebagai berikut:

```

procedure TForm2.binerisasi;
var
  i, j :integer;
begin
  for i:=0 to image.Width do
    for j:= 0 to image.Height do
      begin
        nilai:=getRvalue(image.Canvas.PikseIs[i,j]);
        if nilai<=180 then
          nilai:=clBlack
        else nilai:=clWhite;

        image.Canvas.PikseIs[i,j]:=RGB(nilai,nilai,nilai);
      end;
    end;
  end;
end;

```

2) *Thinning*

Setelah melalui proses *segmentation*, dalam hal ini memisahkan kata menjadi karakternya. Maka proses selanjutnya yaitu *thinning* atau penipisan. Pada citra tersebut dilakukakn *thinning* untuk mendapatkan gambar dengan ukuran 1 piksel. Seperti gambar di bawah ini



Gambar 4.4 Flowchart proses thinning

Listing program untuk mendapatkan citra yang mempunyai ketipisan 1 piksel adalah sebagai berikut:

```

procedure TForm2.prosesThinning(var Points : TPoints ;AWidth,
AHeight : integer);
Var
  i, j, index : integer;
  LB, LA, LN, LAN6, LAN4 : byte;
  Off : TPoint;
  Changing : boolean;
  Remove : array of boolean;
  gambar : array of array of byte;
begin
  SetLength(gambar, AWidth+3);
  for i := 0 to AWidth+2 do begin
    SetLength(gambar[i], AHeight+3);
    for j := 0 to AHeight+2 do begin
      gambar[i][j] := 0;
    end;
  end;
  for index := 0 to High(Points) do
    gambar[Points[index].X+1][Points[index].Y+1] := 1;

  SetLength(Remove, length(Points));

  Changing := True;

  while Changing do begin
    Changing := False;

    for index := 0 to High(Points) do begin

      Remove[index] := False;

      i := Points[index].X + 1;
      j := Points[index].Y + 1;

      LB := 0;
      for LN := 2 to 9 do begin
        Off := Neighbour(LN);
        Inc(LB, gambar[i+Off.X][j+Off.Y]);
      end;

      if (LB < 2) or (LB > 6) then Continue;
      LA := 0;
      for LN := 2 to 9 do begin
        Off := Neighbour(LN);
        if gambar[i+Off.X][j+Off.Y] = 0 then begin
          Off := Neighbour(LN+1);
          if gambar[i+Off.X][j+Off.Y] = 1 then Inc(LA);
        end;
        if LA > 1 then Break;
      end;

      if LA <> 1 then Continue;

      LAN6 := 0;
      for LN := 2 to 9 do begin
        Off := Neighbour(LN);

```

```

    if gambar[i+Off.X][j+Off.Y+1] = 0 then begin
        Off := Neighbour(LN+1);
        if gambar[i+Off.X][j+Off.Y+1] = 1 then Inc(LAN6);
    end;
end;

if (gambar[i][j+1] * gambar[i+1][j] * gambar[i-1][j] <> 0) and
(LAN6 = 1) then Continue;
LAN4 := 0;
for LN := 2 to 9 do begin
    Off := Neighbour(LN);
    if gambar[i+Off.X+1][j+Off.Y] = 0 then begin
        Off := Neighbour(LN+1);
        if gambar[i+Off.X+1][j+Off.Y] = 1 then Inc(LAN4);
    end;
end;

if (gambar[i][j-1] * gambar[i+1][j] * gambar[i][j+1] = 0) or
(LAN4 <> 1) then begin
    Remove[index] := True;
    Changing := True;
end;

end;

if Changing then begin
    index := 0;
    while index < length(Points) do
    begin
        if Remove[index] then
            begin

                gambar[Points[index].X+1][Points[index].Y+1] := 0;
                Points[index] := Points[High(Points)];
                SetLength(Points, High(Points));
                Remove[index] := Remove[High(Remove)];
                SetLength(Remove, High(Remove));
            end else Inc(index);
        end;
    end;
end;
end;
end;

```

3) Segmentation

Proses segmentasi ini diperlukan untuk memisahkan tulisan Arab sambung menjadi karakter. Pemisahan karakter dalam skripsi ini, menggunakan proses cropping karena tulisan Arab sambung memiliki bentuk yang berbeda-beda ketika dihubungkan dengan huruf yang lain. Proses *cropping* pada dasarnya

merupakan operasi translasi, yaitu dengan menggeser koordinat titik citra. Pemotongan dilakukan oleh user dengan menggunakan mouse, yaitu dengan menekan tombol mouse di dalam citra asal (citra yang hendak di-*crop*), masing-masing dari pojok kiri atas sampai dengan pojok kanan bawah. Citra tulisan yang sudah di segmentasi menggunakan proses cropping dapat dilihat pada gambar 4.5 seperti di bawah ini:



Gambar 4.5 Hasil proses *cropping*

Listing program untuk proses segmentasi menggunakan *cropping*

```

procedure TForm2.cropping;
var
  x, y, x1, y1, w, h, w1, h1: integer;
  PC, PH: PByteArray;
  Ki, Ri, Gi, Bi, Ko, Ro, Go, Bo : array of array of byte;
begin
  w := Image.Picture.Width;
  h := Image.Picture.Height;
  w1 := (xB-xA);
  h1 := (yB-yA);
  Image1.Picture.Bitmap.Width := w1;
  Image1.Picture.Bitmap.Height := h1;
  if (Image.Picture.Bitmap.PixelFormat=pf8bit) then
    begin
      SetLength(Ki, w, h);
      SetLength(Ko, w1, h1);
      for y := 0 to h-1 do
        begin
          PC := Image.Picture.Bitmap.ScanLine[y];
          for x := 0 to w-1 do
            Ki[x, y] := PC[x];
          end;
        for x1 := 0 to w1-1 do
          for y1 := 0 to h1-1 do
            begin
              x := x1+xA;
              y := y1+yA;
              Ko[x1, y1] := Ki[x, y];
            end;
          for y1 := 0 to h1-1 do
            begin
              PH := Image1.Picture.Bitmap.ScanLine[y1];

```

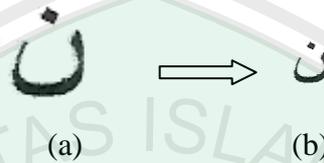
```

        for x1 := 0 to w1-1 do
            PH[x1] := Ko[x1, y1];
        end;
        Ki := nil;
        Ko := nil;
    end;
if (Image.Picture.Bitmap.PikselFormat = pf24bit)
then
begin
    SetLength(Ri, w, h);
    SetLength(Gi, w, h);
    SetLength(Bi, w, h);
    SetLength(Ro, w1, h1);
    SetLength(Go, w1, h1);
    SetLength(Bo, w1, h1);
    for y := 0 to h-1 do
        begin
            PC := Image.Picture.Bitmap.ScanLine[y];
            for x := 0 to w-1 do
                begin
                    Bi[x, y] := PC[3*x];
                    Gi[x, y] := PC[3*x+1];
                    Ri[x, y] := PC[3*x+2];
                end;
            end;
            for x1 := 0 to w1-1 do
                for y1 := 0 to h1-1 do
                    begin
                        x := x1+xA;
                        y := y1+yA;
                        Ro[x1, y1] := Ri[x, y];
                        Go[x1, y1] := Gi[x, y];
                        Bo[x1, y1] := Bi[x, y];
                    end;
                end;
            for y1 := 0 to h1-1 do
                begin
                    PH := Image1.Picture.Bitmap.ScanLine[y1];
                    for x1 := 0 to w1-1 do
                        begin
                            PH[3*x1] := Bo[x1, y1];
                            PH[3*x1+1] := Go[x1, y1];
                            PH[3*x1+2] := Ro[x1, y1];
                        end;
                    end;
                end;
            Ri := nil;
            Gi := nil;
            Bi := nil;
            Ro := nil;
            Go := nil;
            Bo := nil;
        end;
    end;
end;
end;

```

4) *Scaling*

Pada tahap ini ukuran citra diperbesar dan diperkecil sesuai dengan yang ditetapkan pada implementasi skripsi ini yaitu diperkecil menjadi ukuran 15 x 15 piksel. Citra tulisan hasil *scaling* dapat dilihat pada gambar 4.6 di bawah ini



Gambar 4.6 (a) Citra awal (b) hasil *scaling*

Listing program untuk proses *scaling* adalah sebagai berikut

```

procedure TForm2.scalling(sh1,sv1:real);
var
  x,y,x1,y1:integer;
  w,h,w1,h1:integer;
  PC,PH      :PByteArray;
  Ki,Ko      :array of array of byte;
  xAsal, yAsal, wxL, wyT, Sh, Sv :real;
begin
  sh := sh1;
  sv := sv1;
  w := image.Picture.Width;
  h := image.Picture.Height;
  w1 := Round(sh*w);
  h1 := Round(Sv*h);
  image1.Picture.Bitmap.Width :=w1;
  image1.Picture.Bitmap.Height :=h1;

  begin
    SetLength(Ki,w,h);
    SetLength(Ko,w1,h1);
    for y := 0 to h-1 do
      begin
        PC := image.Picture.Bitmap.ScanLine[y];
        for x:= 0 to w-1 do
          Ki[x,y] := PC[x];
        end;
      for x1:= 0 to w1-1 do
        for y1 := 0 to h1-1 do
          begin
            xAsal := x1/Sh;
            yAsal := y1/Sv;
            if ((Floor(xAsal)<0) or (Ceil(xAsal)>w-1) or
              (Floor(yAsal)<0) or (Ceil(yAsal)>h-1))
            then
              Ko[x1,y1] :=255
          end;
        end;
      end;
    end;
  end;

```

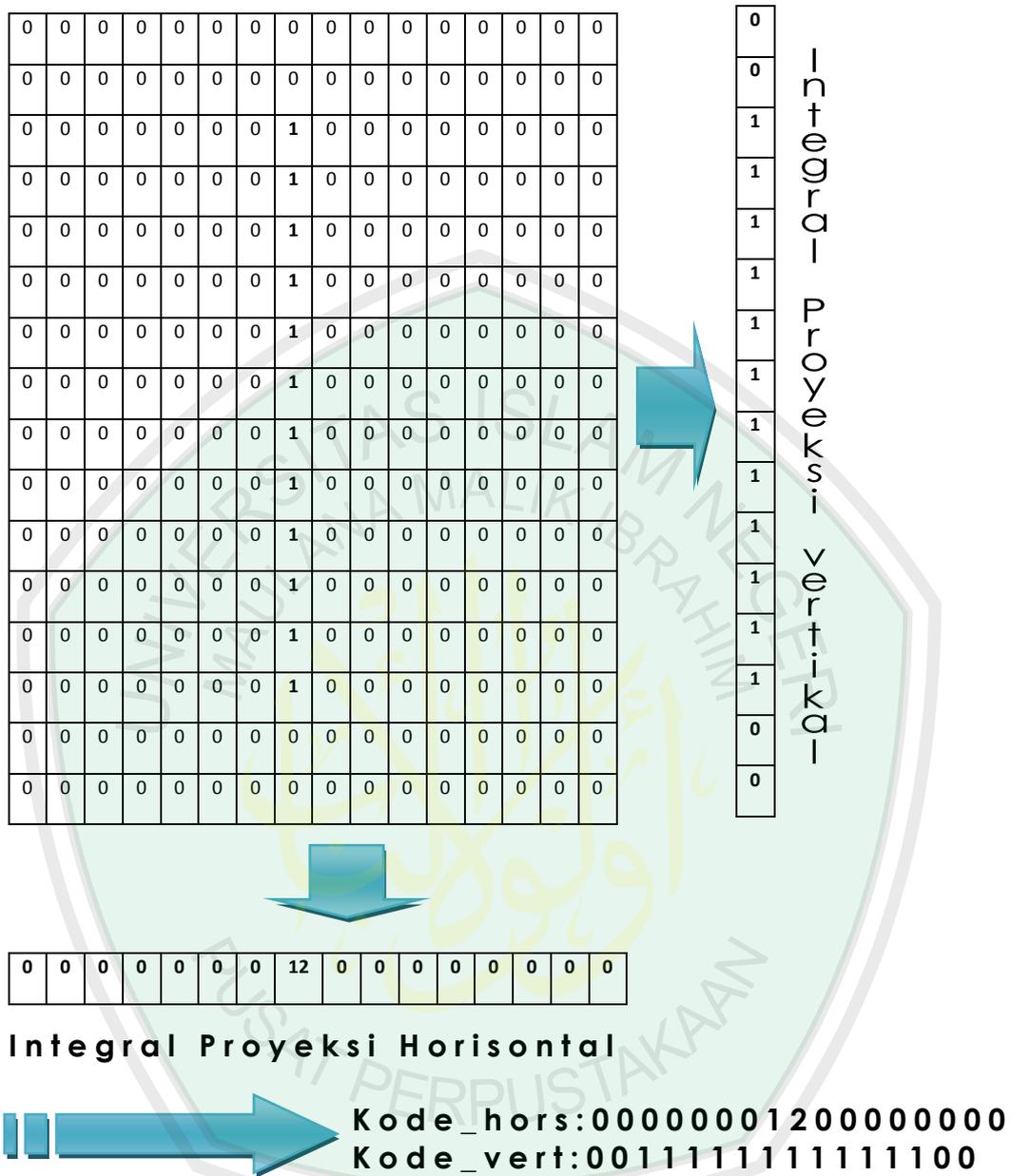
```

else
begin
  xA := Floor(xAsal);
  xB := Ceil(xAsal);
  yA := Floor(yAsal);
  yB := Ceil(yAsal);
  wxL := xB-xAsal;
  wyT := yB-yAsal;
  Ko[x1,y1]:=round(wxL*wyT*Ki[xA,yA]
  +(1-wxL)*wyT*Ki[xB,yA]
  +wxL*(1-wyT)*Ki[xA,yB]
  +(1-wxL)*(1-wyT)*Ki[xB,yB]);
end;
end;
for y1 := 0 to h1-1 do
begin
  PH := imagel.Picture.Bitmap.ScanLine[y1];
  for x1 := 0 to w1-1 do
    PH[x1] := Ko[x1,y1];
  end;
  Ki:=nil;
  Ko:=nil;
end;
imagel.Width :=imagel.Picture.Width;
imagel.Height:=imagel.Picture.Height;
end;

```

c. Pengkodean Karakter

Setelah proses awal selesai dilakukan maka proses selanjutnya yaitu pengkodean. Gambar dikodekan menjadi biner dan ditampilkan pada *stringgrid*. Gambar *downscaling* menjadi 15 x 15 piksel selanjutnya dikodekan jika berwarna hitam maka dikodekan satu dan juga sebaliknya jika berwarna putih maka dikodekan 0. Pengkodean karakter ini berdasarkan integral proyeksi seperti contoh gambar di bawah ini:



Gambar 4.7 Hasil proses pengkodean menggunakan integral proyeksi

Listing programnya sebagai berikut

```
procedure TForm2.Kode11Click(Sender: TObject);
var
  x,y:integer;
begin
  lebar:=15;
  tinggi:=15;
  for x:=0 to lebar do
    for y:=0 to tinggi do
      stringGrid1.Cells[x,y]:=intToStr(tempData[x,y]);
    end;
end;
```

d. Pengenalan (*Recognize*)

Proses terakhir yang dilakukan untuk mendapatkan sebuah huruf hijaiyah hasil dari pemisahan tulisan Arab latin apakah sudah dikenal atau belum yaitu *recognize*. Jalannya proses ini dapat dilihat pada gambar 3.7 Flowchart proses pengenalan. Listing program untuk proses ini adalah sebagai berikut:

```
procedure TForm2.btn_pengenalanClick(Sender: TObject);
var
  hs: byte;
begin
  ADOQuery1.Active := true;
  if not ADOQuery1.Locate ('Kode_hors', Edit1.Text, []) then
  begin
    MessageDlg ('kode "' + Edit1.Text + '" tidak ditemukan
    silahkan simpan', mtError, [mbYes], 0);
    Form3.Show;
    Form2.hide
  end
  else
  begin
    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('Select ID FROM tbl_Pengenalan WHERE
    Kode_hors=:kh');
    ADOQuery1.Parameters.ParamByName('kh').Value:=Edit1.Text;
    ADOQuery1.Open;
    if ADOQuery1.RecordCount>0 then
      hs := ADOQuery1.Fields[0].AsVariant;
    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('Select * from tbl_pelatihan where ID=:a');
    ADOQuery1.Parameters.ParamByName('a').Value:=hs;
    ADOQuery1.Open;
    if ADOQuery1.RecordCount>0 then begin
      MessageDlg('Data yang anda masukkan adalah : '+
```

```

ADOQuery1.Fields[1].AsString,mtInformation, [mbOK], 0);
imgHasil.Picture.Bitmap.Assign(ADOQuery1.FieldByName('gambar'));
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select*from tbl_pengenalan');
ADOQuery1.Open;
end;
end;

```

3. Proses Menampilkan Halaman Input data

Halaman input data ini merupakan proses awal yang dilakukan untuk menyimpan semua data masukan dalam database. Data di dalamnya masih kosong sehingga perlu diadakan pelatihan yang menyimpan semua data karakter masukan pada database untuk dijadikan acuan dalam proses pengenalan dengan metode *template matching*.

The screenshot shows a Windows-style application window titled "Input Data". The window has a green header bar. Inside, there are three input fields: "Kode" containing "1", "Nama Huruf" containing "alif", and "Gambar huruf" containing a file path "E:\kuLiah aiNa,S.Kom\smt SI". Below the "Gambar huruf" field is a small image of the letter 'l'. At the bottom of the window, there are two buttons: "Save" (with a green checkmark icon) and "Back To Home".

Gambar 4.8 Form Input data

Listing pogram untuk proses simpan sebagai berikut:

```

procedure TForm5.BitBtn1Click(Sender: TObject);
var
  gbr:TMemoryStream;
begin
  gbr:=TMemoryStream.Create;
  ADOQuery2.Close;
  ADOQuery2.SQL.Clear;
  ADOQuery2.SQL.Add ('select * from tbl_pelatihan where ID=:a
and Nama_huruf=:b');
  ADOQuery2.Parameters.ParamByName('a').Value:= Edit3.Text;
  ADOQuery2.Parameters.ParamByName('b').Value:=Edit1.Text;
  ADOQuery2.Open;
  if ADOQuery2.RecordCount>0 then
    ShowMessage('Maaf ID Anda Sudah Ada !!!')
  else
    begin
      ADOQuery2.Close;
      ADOQuery2.SQL.Clear;
      ADOQuery2.SQL.Add('insert into tbl_pelatihan (ID,
Nama_huruf, gambar) values (:a,:b,:c)');
      Image2.Picture.Graphic.SaveToStream(gbr);
      ADOQuery2.Parameters.ParamByName('a').Value:=Edit3.Text;
      ADOQuery2.Parameters.ParamByName('b').Value:=edit1.Text;
      ADOQuery2.Parameters.ParamByName('c').LoadFromStream(gbr,
ftBlob);
      ADOQuery2.ExecSQL;
      ADOQuery2.Close;

      ShowMessage('Simpan Ok !');
      Edit1.Text:='';
      Edit2.Text:='';
      Edit3.Text:='';
    end;
end;

```

4. Proses Menampilkan Halaman Simpan

Halaman simpan ini merupakan proses awal yang dilakukan untuk menyimpan semua data dari proses awal ke dalam database..



The image shows a dialog box with a light green background. At the top, it says "Masukkan Nama Huruf" followed by a text input field. Below the input field are two buttons: "Cancel" with a red 'X' icon and "Save" with a green checkmark icon.

Gambar 4.9 Form Simpan

Listing program untuk simpan sebagai berikut:

```

procedure TForm3.btnSaveClick(Sender: TObject);
var id : integer;
begin
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add ('select * from tbl_pengenalan where ID=:a
  and kode_hors=:kh');

  ADOQuery1.Parameters.ParamByName('kh').Value:=Form2.Edit1.Text;
  ADOQuery1.Open;
  if ADOQuery1.RecordCount>0 then
    ShowMessage('Maaf ID Sudah Ada !!!')
  else
    begin
      ADOQuery1.Close;
      ADOQuery1.SQL.Clear;
      ADOQuery1.SQL.Add ('select ID from tbl_pelatihan where
      Nama_huruf=:a');

      ADOQuery1.Parameters.ParamByName('a').Value:=ComboBox1.Text;
      ADOQuery1.Open;
      id := ADOQuery1.Fields[0].AsInteger;

      ADOQuery1.Close;
      ADOQuery1.SQL.Clear;
      ADOQuery1.SQL.Add('insert into tbl_pengenalan (ID,
      kode_hors, kode_vert) values (:a, :kh, :kv)');
      ADOQuery1.Parameters.ParamByName('a').Value:=id;
      ADOQuery1.Parameters.ParamByName('kh').Value:=Form2.edit1.
      Text;

      ADOQuery1.Parameters.ParamByName('kv').Value:=form2.Edit2.
      Text;
      ADOQuery1.ExecSQL;
      ADOQuery1.Close;

      ShowMessage('Simpan Ok !');
      Form2.Show;
      Form3.Close;
      with form2 do begin
        Image1.Picture.Bitmap:=nil;
        Image3.Picture.Bitmap:=nil;
        strBersih;
        Edit1.Text:='';
        Edit2.Text:='';
        Cropping1.Enabled:=true;
      end;

    end;
end;
end;

```

D. PEMBAHASAN DATA HASIL UJI COBA

1. Hasil Uji Coba

Setelah dilakukan uji coba, diperoleh data pembandingan antara data masukan dan data yang sudah disimpan dalam database. Perbandingan tersebut ditinjau dari sisi kode horisontal dan vertikal yang di dapat setelah *preprocessing*, hasilnya dapat dilihat pada tabel 4.2 hasil kode data input dan tabel 4.3 hasil kode yang telah disimpan dalam database.

Tabel 4.2
Hasil kode huruf ain yang telah di simpan dalam database

No.	ID	Kode_hors	Kode_vert
1.	18	143334435322112	0264112004810111
2.	18	24104343344442200	664111211151210200
3.	18	13112223233321110	3641122111159300
4.	18	4112433243421112	1783121111610302
5.	18	4112433243421112	1783121111610302
6.	18	247243223221102	004723211583001
7.	18	51194444345542220	685232222271411200
8.	18	31192233223421110	3652222222710200
9.	18	25104444344542220	584122211161311200
10.	18	31192233223421110	3652222222710200

Setelah didapatkan kode horisontal dan vertikal dari proses sebelumnya, maka pengenalan sudah bisa dilakukan. Tetapi, karena program ini membandingkan data yang ada dalam database dengan data masukan maka huruf tersebut belum bisa dikenal atau belum bisa memunculkan pola huruf hijaiyah yang sebenarnya. Kode-kode tersebut harus disimpan dalam ke dalam database terlebih dahulu agar huruf dapat dikenal. Tabel 4.2 di atas merupakan contoh database dari hasil kode huruf ain yang terletak di awal (*start*) kata/kalimat tulisan Arab

Tabel 4.3
Hasil kode input huruf ain posisi tulisan di depan (*start*)

No.	ID	Kode_hors	Kode_vert	Huruf yang dikenali (T/F)
1.	18	3994444545432100	27762022226139100	F
2.	18	11119223232422110	3552211222279300	F
3.	18	41122333232422110	46611221111610210	F
4.	18	31192233223421110	3652222222710200	T
5.	18	3594545455642220	378611211161311220	F
6.	18	31192233223421110	3652222222710200	T
7.	18	13112223233321110	3641122111159300	T
8.	18	3584344534311000	35441121111102200	F
9.	18	31192233223421110	3652222222710200	T
10.	18	24104343344442200	664111211151210200	T

Berdasarkan pada tabel 4.3 tampak bahwa dalam 10 kali uji coba terdapat 3 kali hasil kode yang sama, yaitu pada kode horisontal 31192233223421110 dan kode vertikal 3652222222710200. Hal ini menunjukkan cara proses *cropping* seseorang sama, semakin seseorang melakukan *cropping* yang sama maka semakin besar kemungkinan kode untuk dikenal. Langkah selanjutnya yaitu membandingkan data masukan tersebut dengan data yang sudah tersimpan dalam database yaitu dibandingkan dengan data pada tabel 4.2 di atas. Terdapat 5 kode yang cocok (*matching*) antara tabel 4.2 data dalam database dan 4.3 data masukan, antara lain 31192233223421110 3652222222710200, 13112223233321110 3641122111159300, dan 24104343344442200 664111211151210200. Untuk gambar hasil pengenalan dari kode-kode tersebut dapat dilihat mulai gambar 4.10.

Dari kedua contoh tabel di atas diperoleh dua kelompok hasil uji coba berdasarkan posisi huruf berada yaitu kelompok yang berhasil mengenali tulisan Arab dan kelompok yang tidak berhasil mengenali tulisan Arab tersebut. Secara

detil dapat dilihat pada tabel 4.2 hasil uji coba pemisahan tulisan Arab dari sepuluh kali percobaan untuk huruf yang sama.

Tabel 4.4
Hasil uji coba ketika posisi tulisan di depan (start)

No.	Nama Huruf	Benar	Salah	Presentase benar (%)
1.	Alif	-	-	-
2.	Ba	2	8	20
3.	Ta	4	6	40
4.	Tha	5	5	50
5.	Jeem	4	6	40
6.	Hha	4	6	40
7.	Kha	3	7	30
8.	Dal	-	-	-
9.	Thal	-	-	-
10.	Ra	-	-	-
11.	Zay	-	-	-
12.	Seen	1	9	10
13.	Sheen	3	7	30
14.	Sad	4	6	40
15.	Dhad	4	6	40
16.	Tta	5	5	50
17.	Za	5	5	50
18.	Ain	6	4	60
19.	Ghain	6	4	60
20.	Fa	2	8	20
21.	Qaf	4	6	40
22.	Kaf	5	5	50
23.	Lam	9	1	90
24.	Meem	8	2	80
25.	Noon	5	5	50
26.	Ha	2	8	20
27.	Waow	-	-	-
28.	Ya	5	5	50

Tabel 4.5
Hasil uji coba ketika posisi tulisan di tengah (*middle*)

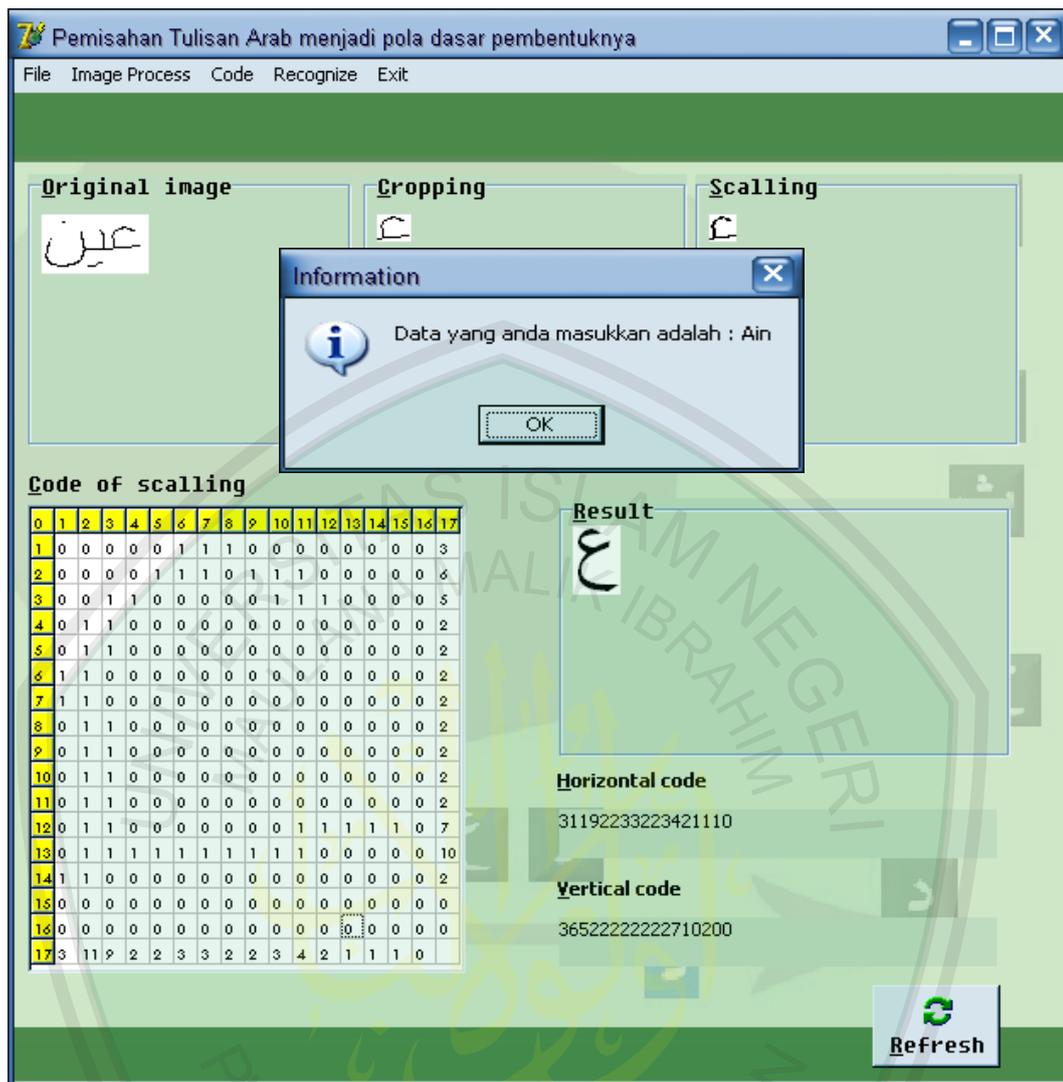
No.	Nama Huruf	Benar	Salah	Presentase benar (%)
1.	Alif	-	-	-
2.	Ba	5	5	50
3.	Ta	5	5	50
4.	Tha	5	5	50
5.	Jeem	4	6	40
6.	Hha	3	7	30
7.	Kha	1	9	10
8.	Dal	-	-	-
9.	Thal	-	-	-
10.	Ra	-	-	-
11.	Zay	-	-	-
12.	Seen	2	8	20
13.	Sheen	0	10	0
14.	Sad	4	6	40
15.	Dhad	4	6	40
16.	Tta	5	5	50
17.	Za	1	9	10
18.	Ain	3	7	30
19.	Ghain	4	6	40
20.	Fa	6	4	60
21.	Qaf	3	7	30
22.	Kaf	3	7	30
23.	Lam	5	5	50
24.	Meem	5	5	50
25.	Noon	6	4	60
26.	Ha	4	6	40
27.	Waow	-	-	-
28.	Ya	5	5	50

Tabel 4.6
Hasil uji coba ketika posisi tulisan di akhir (*end*)

No.	Nama Huruf	Benar	Salah	Presentase benar (%)
1.	Alif	5	5	50
2.	Ba	5	5	50
3.	Ta	5	5	50
4.	Tha	7	3	70
5.	Jeem	4	6	40
6.	Hha	4	6	40
7.	Kha	3	7	30
8.	Dal	6	4	60
9.	Thal	6	4	60
10.	Ra	3	7	30
11.	Zay	5	5	50
12.	Seen	4	6	40
13.	Sheen	4	6	40
14.	Sad	4	6	40
15.	Dhad	5	5	50
16.	Tta	2	8	20
17.	Za	3	7	30
18.	Ain	4	6	40
19.	Ghain	4	6	40
20.	Fa	5	5	50
21.	Qaf	5	5	50
22.	Kaf	6	4	60
23.	Lam	4	6	40
24.	Meem	7	3	70
25.	Noon	4	6	40
26.	Ha	3	7	30
27.	Waow	4	6	40
28.	Ya	5	5	50

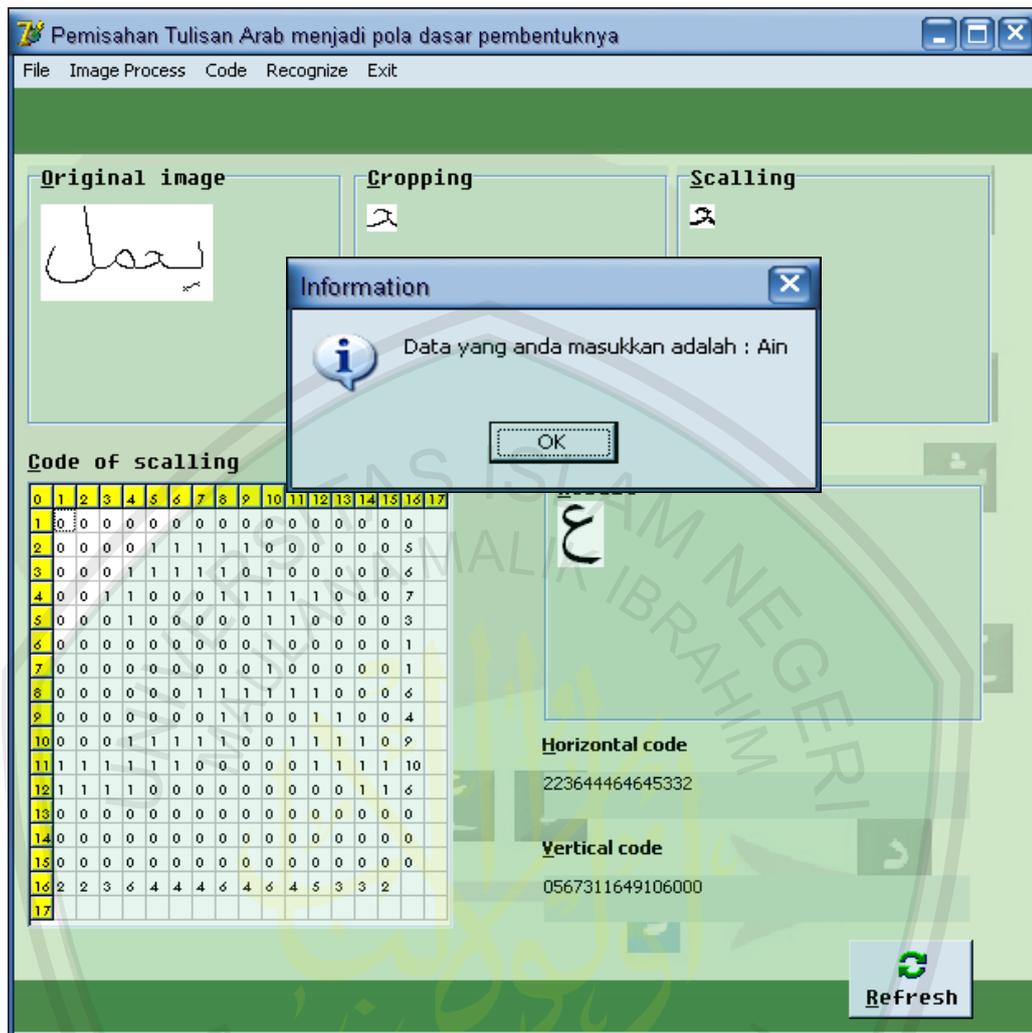
a. Tulisan Arab berhasil dikenali

Gambar 4.10 menunjukkan hasil pengenalan tulisan Arab sambung dari salah satu gambar tulisan Arab yang bisa dikenali dengan tipe gambar *.bmp



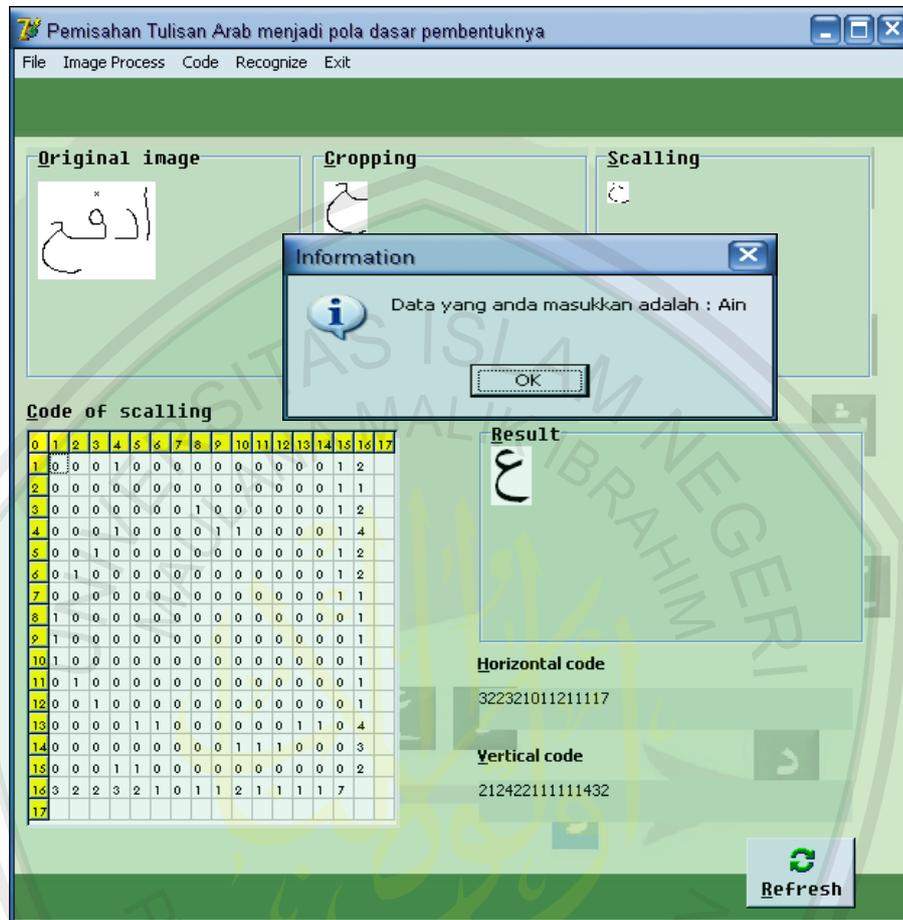
Gambar 4.10 Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf *Ain* di awal

Ketika database masih kosong, disimpan kode sebanyak 10 kali setelah itu baru dilakukan proses uji coba. Huruf *ain* di awal dapat dikenali setelah dilakukan uji coba yang ke-4. Sampai selesai percobaan ke-10 huruf tersebut dikenali sebanyak 6 kali. Jadi, perbandingan antara huruf yang dikenal dengan jumlah percobaan adalah 6:10



Gambar 4.11 Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf *ain* di tengah

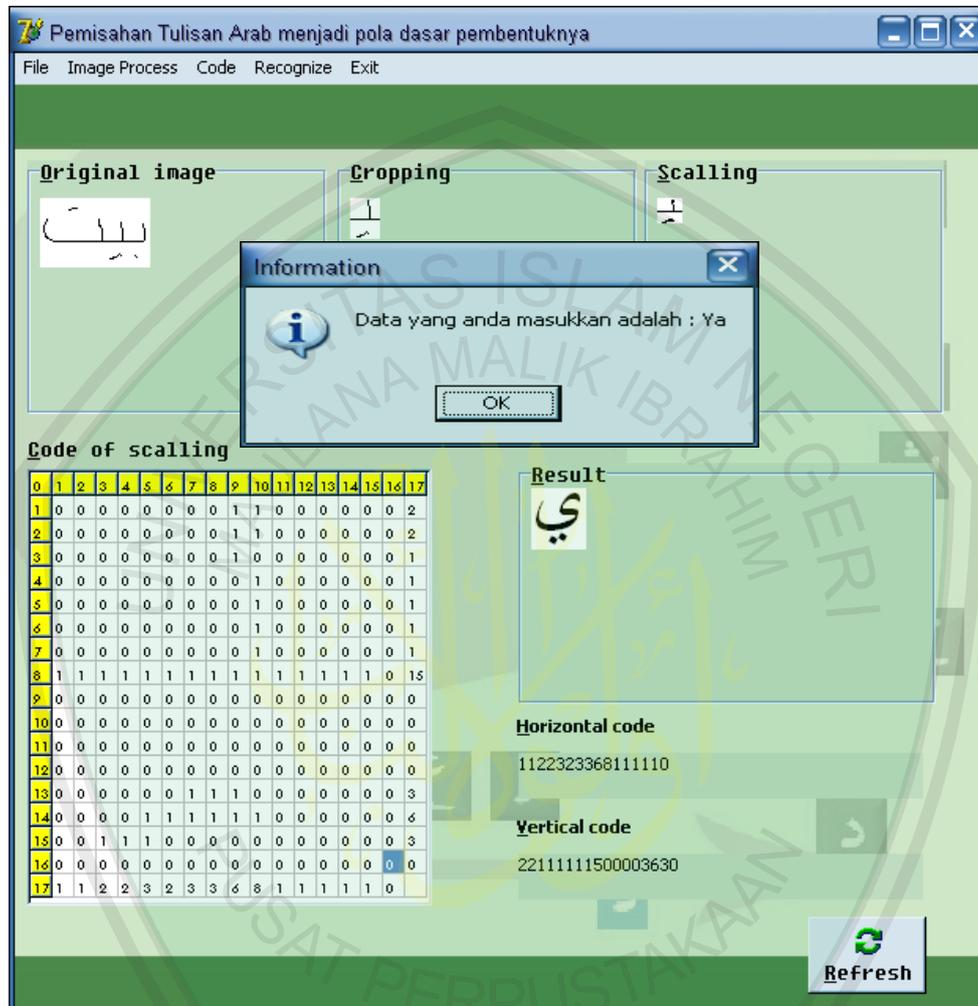
Sedangkan pada huruf *ain* ditengah, menurut data hasil uji coba terdapat benar 3 dan salah sebanyak 7 kali. Intensitas pengenalan menurun sebanding dengan huruf yang posisinya di awal.



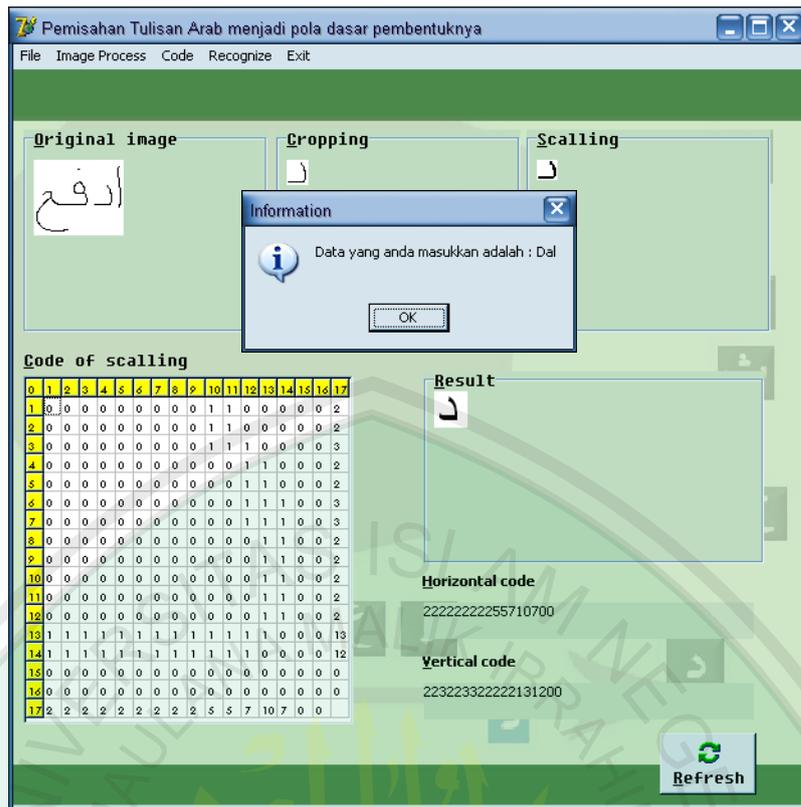
Gambar 4.12 Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf *ain* di akhir

Pada uji coba huruf *ain* yang di akhir, sebanyak 10 kali uji coba program. Setelah uji coba ke-2 program baru bisa mengenali huruf. Presentase kebenarannya hampir sama atau mendekati huruf *ain* yang posisinya di awal. Jadi, dapat dikatakan bahwa letak huruf juga mempengaruhi hasil. Disamping itu, hal yang sangat mempengaruhi tingkat keberhasilan yaitu cara *cropping* dan *mouse* yang digunakan. Setiap orang memiliki cara yang berbeda-beda dalam meng-*crop* sebuah gambar, untuk mendapatkan hasil yang maksimal diusahakan meng-*crop*

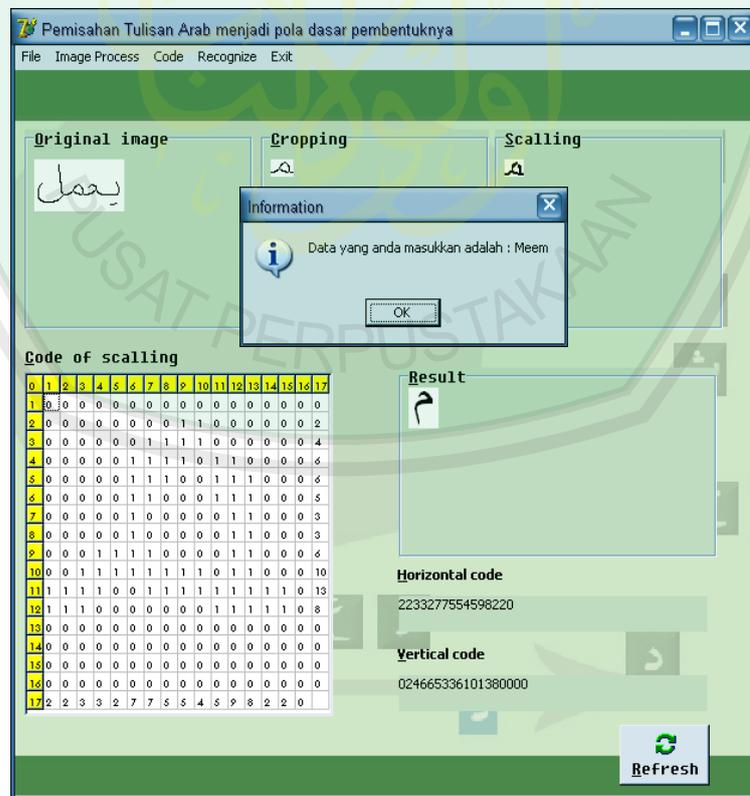
dengan hati-hati dan lebih mendekati huruf. Sedikit ruang kosong di samping kanan-kiri gambar masukan. Di bawah ini ditampilkan daftar huruf lain selain ain



Gambar 4.13 Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf *ya* di tengah



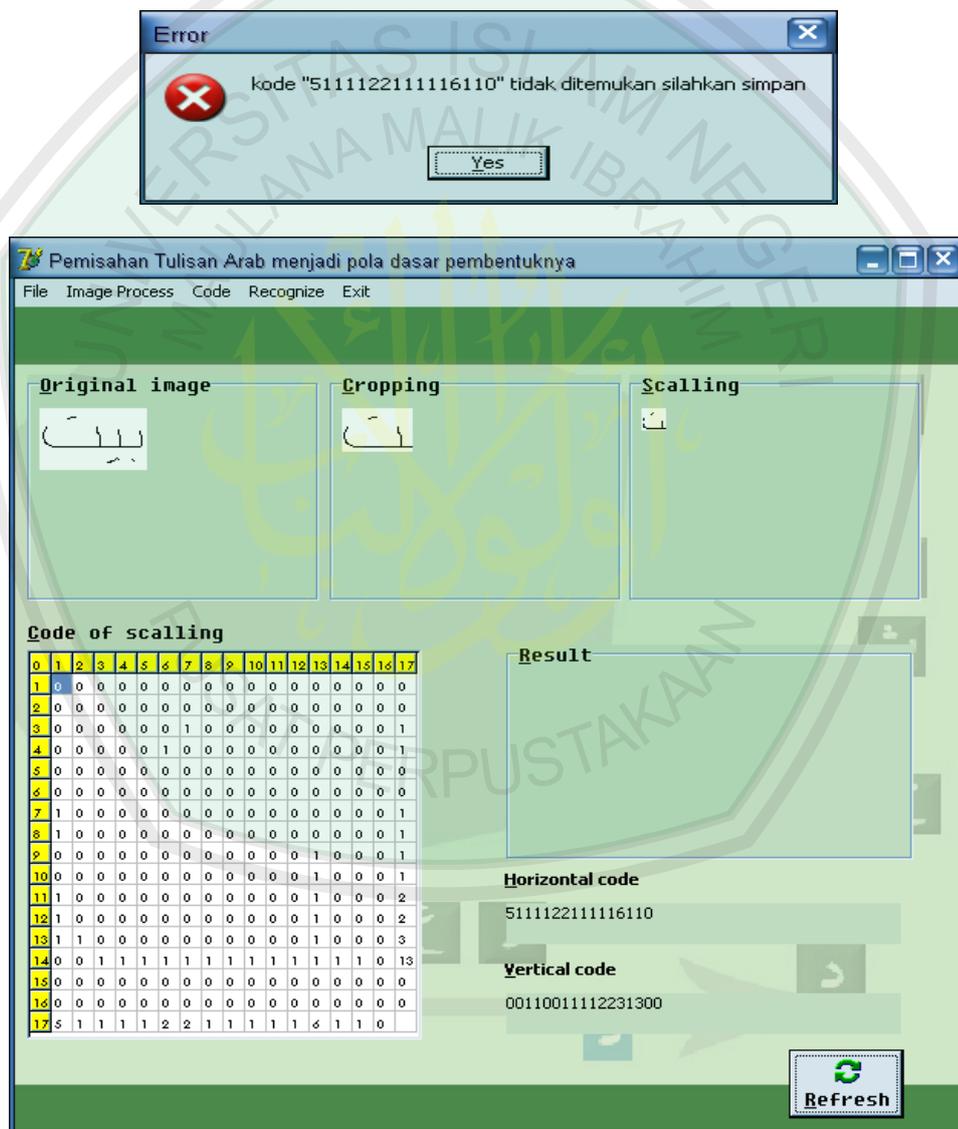
Gambar 4.14 Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf *dal*



Gambar 4.15 Hasil proses pemisahan tulisan Arab sambung yang menghasilkan pola huruf *meem* di tengah

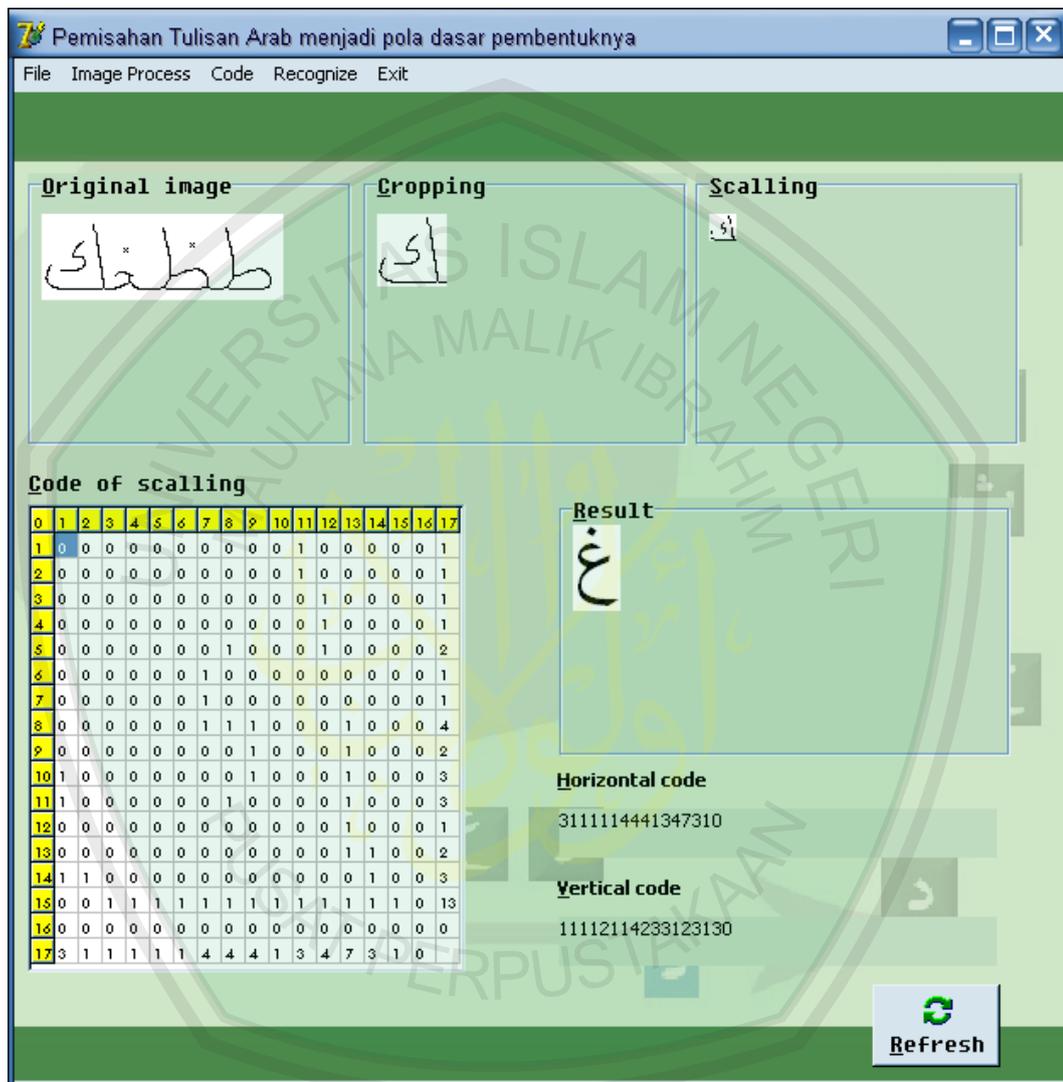
b. Tulisan Arab tidak berhasil dikenali

Jika kode tulisan Arab belum tersimpan dalam database maka ketika option *recognize* di klik akan muncul *message dialog* yang menanyakan bahwa kode belum tersimpan dan harus di simpan dulu ke dalam database. Seperti pada gambar di bawah ini:



Gambar 4.16 Hasil ketika data belum ada dalam database

Jika di klik button *yes* maka akan muncul form simpan yang selanjutnya disimpan sebagai data dalam database.



Gambar 4.17 Hasil ketika data salah memasukkan nama huruf pada proses simpan ke dalam database

2. Kontribusi Program Dalam Islam

Di zaman yang serba teknologi ini, maka al-Quran sudah sepatutnya merambah dan berkembang dengan media teknologi untuk mempermudah orang dalam memperdalam Islam, khususnya dalam mempelajari bahasa Arab bagi pemula. Sesuai dengan firman Allah yang terdapat dalam surat al-Qamar ayat 17, 22, 32, dan 40 sebagai berikut:

وَلَقَدْ يَسَّرْنَا الْقُرْآنَ لِلذِّكْرِ فَهَلْ مِنْ مُدَكِّرٍ ﴿١٧﴾

Artinya:

“Dan Sesungguhnya Telah kami mudahkan Al-Quran untuk pelajaran, Maka Adakah orang yang mengambil pelajaran?” (QS. Al-Qamar/ 54: 17)

Dalam surat tersebut dijelaskan sampai 4 kali bahwasanya Allah menurunkan al-Quran dengan bahasa Arab sehingga dapat mempermudah orang dalam mempelajarinya. Hal tersebut menunjukkan bahwa al-Quran yang berbahasa Arab itu mudah dipelajari oleh umat manusia, dalam artian mudah dihafalkan dan di baca. Seperti yang telah dijelaskan dalam keistimewaan bahasa Arab pada bab 2 di atas.

Selain itu, program yang peneliti sajikan lebih memfungsikan untuk mengenali huruf Arab yang bersambung sehingga dapat teridentifikasi satu-persatu huruf dasarnya. Setelah terketahui huruf dasar akan memudahkan orang yang belajar bahasa Arab untuk memahami akar kata dasar.

Kontribusi yang lain dalam pencitraan program ini memunculkan unsur-unsur seni dalam penggambaran huruf Arab agar lebih menarik dan indah, sebagaimana Islam sangat menghargai keindahan.

BAB V

KESIMPULAN

A. KESIMPULAN

Dari hasil uji coba yang dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

1. Uji coba dilakukan sebanyak 10 kali dengan tulisan yang sama, presentase tulisan Arab sambung yang dapat dipisah menjadi pola huruf pembentuknya yaitu 44% untuk huruf Arab yang ketika disambung letaknya di awal (*start*), 38% untuk huruf Arab yang ketika disambung letaknya di tengah (*middle*), dan 52% untuk huruf Arab yang ketika disambung letaknya di akhir (*end*).
2. Pada proses pemisahan tulisan Arab sambung ini, semakin banyak data yang disimpan semakin cepat huruf dikenal.
3. Terdapat tulisan tangan yang tidak dapat dikenali, hal ini disebabkan oleh beberapa faktor, yaitu:
 - Alat pendukung yang digunakan dalam hal ini adalah mouse pergerakannya kurang lancar saat melakukan proses segmentasi atau proses *cropping* per karakter
 - Kesalahan memasukkan nama huruf ketika simpan kode ataupun kesalahan kode pada saat input data awal sehingga antara nama huruf dan gambar tidak sesuai

B. SARAN

4. Untuk pengembangan aplikasi selanjutnya diharapkan lebih serba otomatis jadi ketika tulisan Arab hasil *scanning* masuk langsung dikenali menjadi pola huruf hijaiyah, tidak perlu dilakukan proses awal
5. Lebih mengembangkan pengenalan ke arah tulisan tangan seseorang yang berbeda-beda tingkat kemiringan dan ketebalannya
6. Untuk aplikasi selanjutnya diharapkan tulisan tangan bisa menjadi identitas seseorang, ketika orang itu menulis maka sistem langsung mengenali nama orang tersebut
7. Untuk pengembangan selanjutnya, sistem bisa mengenali tulisan Arab sambung dan bisa disimpan dalam bentuk *.txt atau *.rtf. Sehingga lebih mempermudah seseorang yang akan mengedit tulisan Arabnya setelah dilakukan proses *scanning*

DAFTAR PUSTAKA

- Abdullah Amin. 2004. *Integrasi Sains-Islam: Mempertemukan Epistemologi Islam dan Sains*. Yogyakarta: Pilar Religia
- Ahmad Balza & Firdausy Kartika. 2005. *Teknik Pengolahan Citra menggunakan Delphi*. Yogyakarta: Ardi Publishing
- Ahmad Usman. 2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta: GRAHA ILMU
- Al-Rashaideh Hasan. 2006. "Preprocessing phase for Arabic Word Handwritten Recognition". Russian Federation: Information Transmissions in Computer Networks
- Amalian. 2008. *Belajar Bahasa Arab, yuk?*. Diakses dari <http://amalian.multiply.com/journal/item/2>; Internet; diakses pada tanggal 5 Oktober 2008
- Anwar Yusuf Ali. 2006. *Islam dan Sains Modern Sentuhan Islam terhadap Berbagai Disiplin Ilmu*. Bandung: CV. PUSTAKA SETIA
- Beigi, Hamayoon SMM. 1998. *An Overview of Handwriting Recognition*. TJ Watson Research Center International Bussiness Machines
- Biadsy Fadi dkk. NN, "Online Arabic Handwriting Recognition Using Hidden Markov Models", New York: Columbia University
- Fahmy, Maged Mohamed Mahmoud dan Somaya Al Ali. 2000. *Automatic Recognize Of Handwritten Arabic Characters Using Their Geometrical Features*. Diakses dari http://www.ici.ro/ici/revista/sic2001_2/art1.htm; Internet; diakses pada tanggal 11 Maret 2008.
- Hamdani, Deni. 2008. *Memahami Tatabahasa Al-Quran*. Diakses dari <http://www.denyhamdani.de.vu>; Internet; diakses pada tanggal 5 Oktober 2008.
- Husni. 2004. *Pemrograman Database dengan Delphi*. Yogyakarta: GRAHA ILMU.
- Liana M. Loligo and Venu Govindaraju. *Off-line Arabic Handwriting Recognition : A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No.3, March 1998
- Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: INFORMATIKA

Pranata Antony. 2003. *Pemrograman Borland Delphi 6 Edisi 4*, Yogyakarta: Andi

R.C. Gonzalez, R.E. Woods. 1992. *Digital Image Processing*. USA: Addison-Wesley Publishing Company.

Vori, Vuokka. 1998. *Adaptive Online Recognition of Handwriting*

