Jurnal Teknologi Maklumat & Multimedia **3**(2006): 107-125

Penemberengan Teks Jawi Tulisan Tangan: Satu Pendekatan Gabungan

KHAIRUDDIN OMAR, RAMLAN MAHMOD, MD. NASIR SULAIMAN & ABDUL RAHMAN RAMLI

ABSTRAK

Artikel ini menjelaskan satu pendekatan gabungan untuk menyelesaikan penemberengan teks Jawi. Penemberengan adalah satu daripada beberapa fungsi utama dalam sistem Pengecaman Teks Optik Jawi atau PTOJ. Ia melibatkan proses memisahkan satu koleksi teks kepada aksara-aksara tunggal untuk dicamkan. Secara amnya teks Jawi mempunyai lima bentuk lazim, iaitu tindanan memugak, ligatur, berbaris, bersambung pada satu baris dan bersentuh antara dua aksara. Terdapat tiga pendekatan utama untuk menembereng bentuk lazim ini, iaitu Unjuran Profail Histogram (UPH), Pelabelan Komponen Terkait (PKT), dan Penentuan Titik Tembereng (PTT). UPH boleh digunakan untuk memecahkan teks Jawi kepada baris teks, kemudian perkataan. PKT boleh mengumpulkan kontur bagi komponen yang terkait, manakala PTT menekankan pencarian satu titik tembereng berpenentuan dengan mencari tembereng-tembereng simpang di antara aksara. Ketiga-tiga pendekatan ini digabungkan untuk menyelesaikan masalah penemberengan teks Jawi tulisan tangan dengan sedikit pengubahsuaian. Algoritma yang berkaitan juga dijelaskan dengan menumpukan kepada tiga bentuk lazim yang utama, iaitu tindanan memugak, ligatur dan bersambung pada satu baris. Satu uji kaji telah dijalankan dan hasilnya dibincangkan berbanding dengan pendekatan UPH.

Kata kunci: Penemberengan baris teks; penemberengan perkataan; penemberengan aksara.

ABSTRACT

This article explains a combination approach of segmenting Jawi text. Segmentation is one of several main functions in Jawi Optical Character Recognition or JOCR. It involves a process of separating a collection of text to characters for recognition. In general, the text have five basic forms which are; vertical overlap, ligature, diacritics, horizontal overlap and two connected characters. There are three main approaches to segment these forms, there

are Histogram Profile Projection (HPP), Labelled Connected Components (LCC), and Determining of Segmentation Points (DSP). HPP can be used for segmenting Jawi text to text lines, then to words. LCC can gather all contours of connected components, meanwhile DSP is stressed on determination of definitive segmentation points by searching the junction segments between characters. These three approaches are combined in order to solve the problem of segmentation for Jawi handwritten text with a little modification. The related algorithm is also described which emphasises on three main forms of Jawi characters; which are vertical overlap, ligature, and horizontal overlap. An experiment has been carried out and the results are discussed in comparison to those of HPP approach.

Keyword: Text line segmentation; word segmentation; character segmentation

PENGENALAN

Pengecaman Aksara Optik (PAO) adalah merupakan satu daripada cabang bidang pengecaman corak (Al-Badr & Mahmoud 1995). Penekanannya adalah kepada membaca aksara secara automatik. Matlamat utama PAO ialah meniru kebolehan manusia untuk membaca sepantas yang mungkin dengan mewakilkan imej-imej aksara dengan identiti-identiti simbolik.

Dalam tempoh dua dekad yang lalu perkembangan teknik pengecaman aksara Latin dan China menggunakan mesin telah mencapai tahap yang membanggakan. Bagaimanapun, teknik seumpama itu untuk aksara Arab masih di peringkat awal dan penerokaan kerumitannya masih belum dikaji dengan sepenuhnya. Perkara ini disebabkan teknik yang digunakan untuk pengecaman aksara Latin dan China tidak bersesuaian dengan aksara Arab tanpa pengubahsuaian teknik asas (Khairuddin 2000).

Perbezaan utama pengecaman aksara Arab dan Latin ialah (i) keperluan kepada satu langkah kaedah penemberengan aksara, dan (ii) keperluan kepada kaedah pengecaman yang bersesuaian (Khairuddin 2000). Perbezaan yang pertama itu memberi kesan kepada kadar pengecaman jika aksara yang ditembereng itu tidak menggambarkan aksara yang betul. Hal ini akan dibincangkan secara mendalam. Perbezaan yang kedua tidak akan dibincangkan di sini dan maklumat lanjut tentang perbezaan itu boleh dilihat dalam Khairuddin (2000). Begitu juga tentang sejarah PAO bagi teks Arab telah dibincang oleh Khairuddin dan Ramlan (1999), ciri-ciri aksara Arab oleh Altuwaijri dan Bayoumi (1995) dan Romeo-Pakker et al (1995).

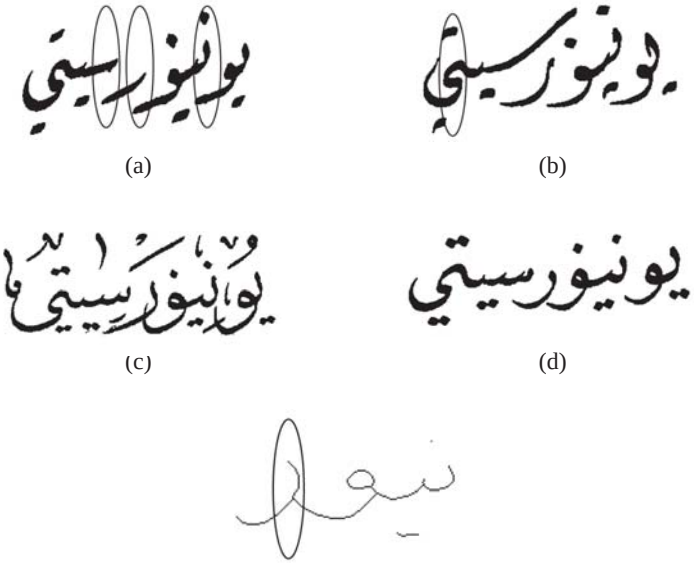
Tiga daripada sifat utama aksara Arab dan Jawi ialah aksaranya bertindan secara memugak, ligatur dan aksara berbaris sama ada di atas atau di bawah seperti yang terdapat dalam Al-Quran. Sifat-sifat ini adalah paling sukar

untuk dicam (Khairuddin 1999a, 1999b). Teks Jawi dari jenis tulisan tangan adalah paling sukar untuk ditemberengkan oleh satu sistem Pengecaman Teks Optik Arab/Jawi atau PTOAJ. Kebanyakan ralat dalam pengecaman ini berlaku ketika fasa penemberengan dan masa banyak dihabiskan di dalam fasa ini. Walaupun demikian pengecaman aksara tunggal pula didapati tidak berbeza dengan pengecaman teks Latin. Lihat Al-Badr dan Mahmoud (1995) dan Al-Badr (1992).

Penemberengan ialah satu proses memisahkan perkataan kepada aksara-aksara tunggal untuk dicamkan (Al-Badr & Mahmoud 1995). Mengecam teks tulisan tangan lebih sukar berbanding dengan teks bercetak. Ia disebabkan oleh sifat ligatur dan aksara bertindan yang lazimnya membentuk sebahagian daripada teks tulisan tangan berbanding teks bercetak. Hal ini menyebabkan penemberengan menjadi semakin sukar. Tatacara penemberengan aksara menjadi canggih apabila melibatkan keseluruhan sifat-sifat tadi (Al-Badr 1992). Rajah 1 menunjukkan lima bentuk lazim bagi sifat-sifat aksara Arab dan Jawi, iaitu tindanan memugak (a), ligatur (b), berbaris (c), bersambung pada satu baris (d), dan bersentuh di antara dua aksara, (e). Kajian menunjukkan bahawa kebanyakkan pendekatan penemberengan tidak mengambil kira keseluruhan bentuk ini (Khairuddin 2000).

Bagi bentuk 1, imej Arab/Jawi terdiri lebih daripada 1 aksara Arab/Jawi dalam satu perkataan atau subperkataan serta menggunakan ruang aksara yang lain supaya dapat memenuhi sifat-sifat aksara tersebut. Bentuk 2, imej Arab/Jawi terdiri daripada dua atau lebih aksara dalam satu perkataan atau subperkataan serta menggunakan satu ruang untuk aksara-aksara tersebut. Bentuk 3 adalah terdiri daripada baris-atas, baris-bawah dan sukun pada satu aksara dalam satu ruang. Bentuk 3 jarang digunakan dalam aksara Jawi tetapi kerap bagi aksara Arab. Bentuk 4 adalah yang paling banyak berlaku kerana apabila satu perkataan ditulis aksara-aksara Arab/Jawi akan bersambung mengikut sifat-sifat aksara jirannya. Akhir sekali ialah bentuk 5, bentuk ini berlaku apabila percantuman di antara satu aksara tunggal misalnya aksara “ ڤ ” dengan aksara “ ر ” (di dalam contoh pada bahagian yang dibulatkan dengan tanda bujur) membentuk dua aksara yang bercantum. Sedangkan sifatnya tidak boleh bercantum.

Bahagian berikut akan menjelaskan secara terperinci kaedah cadangan bagi menyelesaikan beberapa masalah di atas. Huraian dimulai dengan proses penemberengan atau pisahan teks Jawi, diikuti penjelasan tentang *titik tembereng berpotensi* atau TTB. Cerapan tentang sifat-sifat TTB bagi data uji kaji juga dinyatakan. Algoritma terperinci diberikan di bahagian berikutnya. Beberapa contoh hasil uji kaji penemberengan cadangan serta perbincangan ada dinyatakan.



(a) (b)

(c) (d)

(e)

RAJAH 1. Bentuk Lazim Aksara Arab/Jawi (a) Bentuk 1: Tindanan memugak (b) Bentuk 2: Ligatur (c) Bentuk 3: Berbaris (d) Bentuk 4: Bersambung pada satu baris (e) Bentuk 5: Sentuhan dua aksara

PENEMBERENGAN TEKS JAWI

Objektif utama adalah untuk menjelaskan proses penemberengan teks di dalam PTOAJ. Proses penemberengan atau pemisahan dibahagikan kepada tiga peringkat, iaitu peringkat pemisahan baris-baris teks, pemisahan subperkataan atau perkataan dan pemisahan aksara-aksara tunggal. Algoritma yang digunakan di dalam kajian ini adalah berasaskan UPH yang telah dilaksanakan oleh Khella (1992), PKT yang dilaksanakan Pitas (1993) dan PTT oleh Bushofa dan Spann (1997). Satu algoritma cadangan untuk memisahkan subperkataan atau perkataan teks Jawi kepada aksara-aksara tunggal dipaparkan di sini hasil daripada penggabungan tiga kaedah di atas dengan sedikit pembaharuan.

Secara amnya, tugas satu teknik penemberengan tradisional adalah menembereng perkataan-perkataan kepada aksara-aksara terpisahkan dengan memahami sifat-sifat asas titik yang menghubungkan aksara-aksara supaya boleh dipecahkan kepada aksara-aksara yang tertemberengkan. Sifat-sifat ini sangat bergantung kepada garis tapak sesuatu perkataan yang dipertimbangkan.

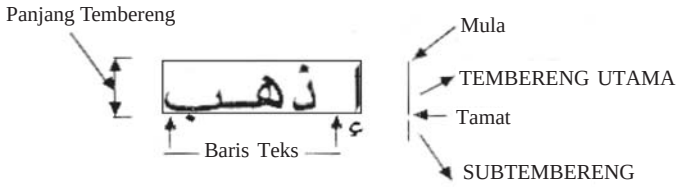
Mengenal pasti garis tapak boleh dilakukan dengan menentukan unjuran histogram secara mengufuk bagi piksel imej melalui pendekatan UPH. Kemudian, kenal pasti titik yang paling maksimum. Seterusnya perkataan boleh dipisahkan berdasarkan kepada garis tapak tadi dengan mencari satu

titik yang sangat berpotensi melalui dua puncak dalam minimum tempatan unjuran histogram supaya boleh dibawa kepada ujian seterusnya sama ada titik itu tepat atau tidak (Khella 1992).

Terdapat dua masalah pendekatan tradisional ini, iaitu:

i. Terdapat aksara yang bertindan secara memugak dan mengufuk (lihat Rajah 1).
ii. Bahagian cantuman di antara dua aksara terlalu pendek. Oleh itu terlalu sukar untuk menentukan titik tembereng. Biasanya titik tembereng tersebut diletakkan di dalam batas saiz sesuatu aksara itu sendiri berbanding dengan ruang di antara dua aksara.

Pendekatan yang dicadangkan ini cuba mengatasi masalah yang disebutkan di atas. Secara amnya, algoritma ini menembereng baris teks dengan membahagikannya kepada satu tembereng utama dan dua subtembereng. Rajah 2 menjelaskan tentang tembereng utama yang terletak pada garis tapak teks, manakala satu subtembereng itu pada bahagian bawah tembereng utama (kebiasaannya juzuk-juzuk sekunder seperti tanda-tanda baris bawah, atau huruf hamzah). Rajah ini tidak melibatkan subtembereng di bahagian atas tembereng utama.



RAJAH 2: Penemberengan baris teks menggunakan pendekatan Khella (1992)

Terdapat dua syarat yang mesti dipenuhi untuk mentakrifkan satu tembereng utama iaitu:

i. Panjang satu tembereng >= purata panjang tembereng. Purata panjang tembereng, iaitu *A*, diberikan oleh rumus (1) berikut:

$$A = \sum_{i-1}^{T} (\text{panjang tembereng}_i)/T. \quad (1)$$

dengan *T* adalah bilangan tembereng, untuk $i = 1, 2, 3, ..., T$.

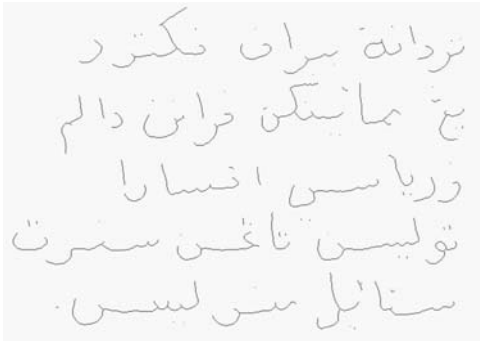
ii. Puncak tembereng ternormalkan >= 15% daripada objek piksel paling kiri bagi baris tengah tembereng secara mengufuk.

PENEMBERENGAN BARIS TEKS

Algoritma Khella (1992) untuk penemberengan baris telah dilaksanakan oleh Khairuddin (2000). Output bagi algoritma ini ialah satu fail yang mengandungi maklumat-maklumat berikut:

i. bilangan baris teks yang telah dipisahkan;
ii. Saiz bagi setiap blok baris teks; dan
iii. Blok baris-baris teks.

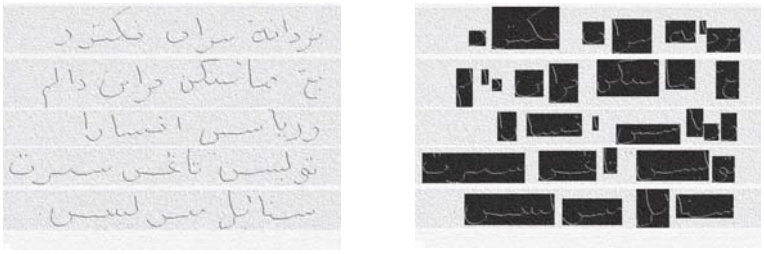
Penyimpanan maklumat-maklumat di atas adalah tambahan kepada kaedah Khella. Satu contoh aplikasi diberikan dalam Rajah 3.



RAJAH 3. Satu contoh imej teks jawi yang telah dinipiskan

PENEMBERENGAN PERKATAAN

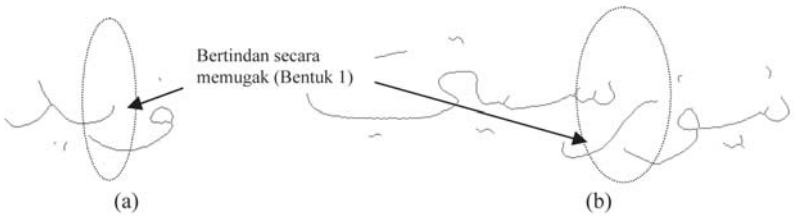
Dua algoritma telah dicadangkan untuk memisahkan subperkataan atau perkataan pada baris teks. Algoritma UPH digunakan untuk menyelesaikan teks Jawi dari bentuk 4 (iaitu subperkataan atau perkataan tidak bertindan secara memugak). Algoritma PKT digunakan untuk menyelesaikan teks dari bentuk 1. Satu contoh hasil penemberengan menggunakan satu daripada algoritma ini dalam Rajah 4 berikut:



(a) (b)

Rajah 4. Satu Contoh Imej Teks Jawi (a) Sesudah dinipiskan dan ditembereng mengikut baris (b) Sesudah ditembereng mengikut subperkataan atau perkataan

Bagaimanapun prestasi algoritma ini menjadi rapuh apabila terdapat subperkataan yang bertindan dengan subperkataan yang lain secara memugak, iaitu ia tidak dapat dipisahkan dengan betul. Rajah 5 menunjukkan dua contoh kegagalan algoritma ini.



RAJAH 5. Kegagalan Pendekatan Khella. (a) Kes Mengufuk (Bentuk1) aksara " ڤ " bertindan di bawah aksara "ڽ". (b) aksara " ڤ ", " ر " dan "ستي" bertindan di antara satu sama lain.

Untuk mengatasi masalah ini satu algoritma lain telah dicadangkan. Algoritma ini berasaskan PKT, yang bermula dengan mencari piksel hitam pertama dan memberikan nombor jujukan pertama untuk setiap blop bernilai 1 yang ditemui berjiran dengan piksel hitam yang pertama itu. Nombor jujukan ini akan dihentikan jika tidak ada lagi jiran yang bersebelahan. Piksel hitam dengan nombor jujukan ini dikenali sebagai objek pertama. PKT akan terus mencari piksel hitam yang berikut yang berhampiran dengan objek pertama ini, dan dikenali sebagai objek kedua. PKT akan terus mencari objek berikutnya sehingga tidak ada lagi objek yang ditemui. Objek-objek ini akan dikenali sebagai subperkataan di dalam teks Jawi. Secara tak langsung PKT telah membilang objek atau subperkataan yang wujud dalam imej teks Jawi. Algoritma PKT ini berasaskan jiran setempat dan melaksanakan operasi setempat secara berlelar. Algoritma ini juga telah diaplikasikan oleh Pitas (1993). Rajah 5 menunjukkan dua contoh perkataan yang apabila diaplikasikan kepada algoritma PKT ini akan menghasilkan dua objek atau subperkataan iaitu " ڤ " dan "ڽد" bagi Rajah 5(a), dan tiga objek atau subperkataan iaitu " نيو ", " ر ", dan " سيتي " bagi Rajah 5(b).

Secara ringkasnya algoritma ini menggunakan konsep 'rumput terbakar' dan dilaksanakan secara rekursi (Pitas 1993). Imej akan diimbas dalam keadaan baris demi baris sehinggalah piksel pertama dalam sempadan objek imej penduaan ditemui. Api akan di'cucuh' pada piksel ini dan akan merambat keseluruhan piksel yang bersempadan secara 8-jiran dengan piksel semasa. Operasi ini diteruskan secara rekursi sehinggalah kesemua piksel objek imej dibakar dan api kemudian 'terpadam' (Pitas 1993). Di akhir operasi ini, kesemua piksel yang dipunyai oleh objek mempunyai nilai 0 dan tidak boleh dibezakan lagi dengan latar belakang. Bilangan piksel adalah keluasan objek. Tatacara ini diteruskan sehingga kesemua objek dalam imej dilabelkan.

Tatacara ini dilaksanakan secara rekursi, oleh itu, timbunan yang besar diperlukan untuk menampung operasinya. Aplikasi algoritma ini ditunjukkan dalam Rajah 6. Tembereng dengan kontur C_i telah diperoleh dan kemudian disingkirkan daripada peta. Proses ini diteruskan sehingga tiada kontur yang tinggal.

$$C = \{C_i \mid i = 1, 2, \ldots, N\}, \quad (2)$$

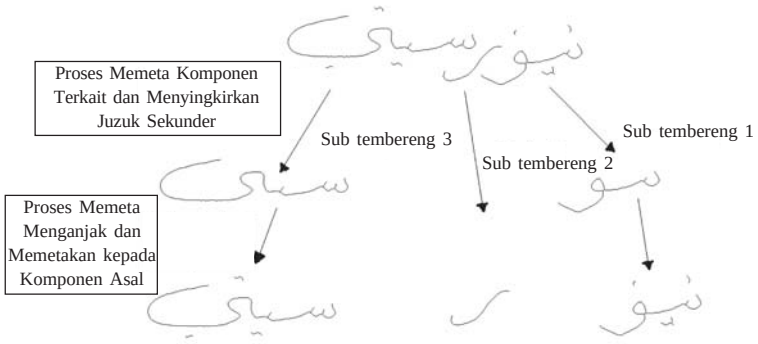
dengan N adalah jumlah kontur.

$$C_i = \{(x_{ip}, y_{ip}) \mid p = 1, 2, \ldots, L_i\}, \quad (3)$$

dengan L_i adalah panjang kontur ke i dalam sebutan bilangan piksel.

Berikut diberikan algoritma PKT:
Algoritma Penemberengan Subperkataan atau Perkataan (kaedah PKT)
Input: Imej perkataan dalam baris teks.
Output: Imej subperkataan terlabelkan atau perkataan terlabelkan (L_1, ..., L_N).
Permulaan algoritma
Langkah 1: Tandakan dengan nilai awal label = 1.
Langkah 2: Mulakan menjelajah imej atas di sebelah kanan dan semak jiran piksel menggunakan kehubungan 8-jiran. Penjelajahan diteruskan sehingga tidak ada jiran piksel hitam dikesemua 8-jirannya dan penjelajahan dihentikan. Imej input yang telah dijelajah akan digantikan dengan nilai sifar.
Langkah 3: Cari tembereng imej yang lain. Jika masih ada piksel hitam, nilai label ditokok dan pergi ke Langkah 2. Jika tiada, proses ditamatkan.
Tamat algoritma.

Rajah 6 memberikan hasil daripada penggunaan algoritma di atas. Prestasinya lebih baik berbanding dengan yang diutarakan oleh Khella (1992).



RAJAH 6. Proses pemisahan subtembereng berdasarkan algoritma PKT

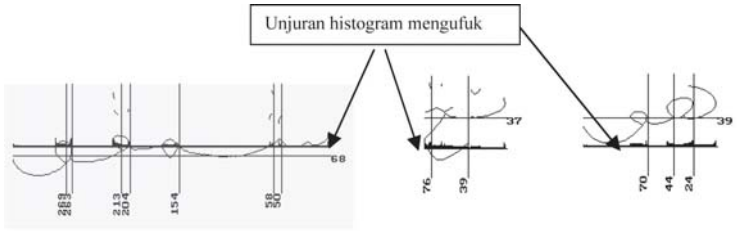
Dalam bahagian ini perbincangan khusus untuk memisahkan teks Jawi menjadi aksara yang tercerai. Dua contoh perkataan Jawi asal diberikan dalam Rajah 7.

(a) (b)

Rajah 7. Perkataan Jawi Asal. (a) Perkataan " شمڤو " yang bermaksud syampu. (b) Perkataan " هوتڠ " yang bermaksud hutang

Khella (1992) telah menggunakan kaedah penemberengan UPH yang berasaskan kepada titik minimum tempatan pada histogram pugak di antara dua puncak. Sesudah menemberengkan baris teks kepada perkataan-perkataan, masalah penemberengan telah dikurangkan dengan menemberengkan setiap perkataan atau subperkataan kepada aksara-aksara atau juzuk-juzuk aksara. Dua contoh hasil penemberengan menggunakan kaedah ini dipaparkan dalam Rajah 8.

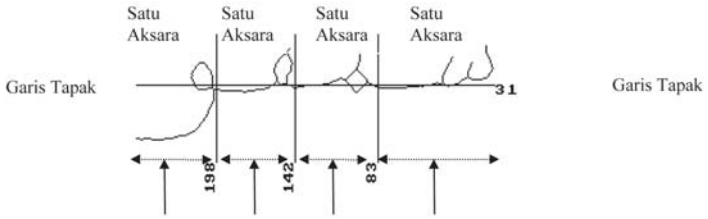
Hasil dari Rajah 8 menunjukkan dua contoh kegagalan untuk menyelesaikan masalah aksara yang bergelong. Khella juga tidak menimbangkan teks dari bentuk 2 (ligatur). Untuk mengatasi masalah ini dua algoritma terubah suai telah dicadangkan. Algoritma pertama adalah hasil dari gabungan daripada kaedah UPH memugak dan PTT. PTT akan mencari fitur-fitur global dan tempatan seperti titik silang, titik simpang, dan titik peralihan arah. Titik yang ditemui dikatakan TTB. TTB dan sifat-sifatnya akan dijelaskan di bahagian berikut. Algoritma kedua juga berasaskan kaedah yang sama tetapi mencari unjuran mengufuk (berbanding unjuran memugak untuk algoritma pertama). Kedua-dua algoritma ini juga akan dibincangkan di bahagian berikut.



(a) (b)

RAJAH 8. Kegagalan dalam pendekatan Khella. (a) aksara bergelong (aksara ڤ dan) و dan bertitik (aksara ش), (b) aksara gelong (aksara و)

Titik tembereng terbentuk pada kedudukan yang menyambungkan dua aksara dan sering berlaku pada garis tapak yang merupakan profil ufuk mengandungi bilangan piksel hitam yang maksimum (Bushofa & Spann 1997). Keadaan seumpama itu adalah sangat sesuai bagi teks Arab/Jawi yang bercetak jika dibandingkan dengan teks Arab/Jawi yang ditulis menggunakan tangan. Adalah terlalu sukar untuk memperoleh penyambungan dua aksara pada garis tapak seperti yang dipaparkan dalam Rajah 9 berikut:



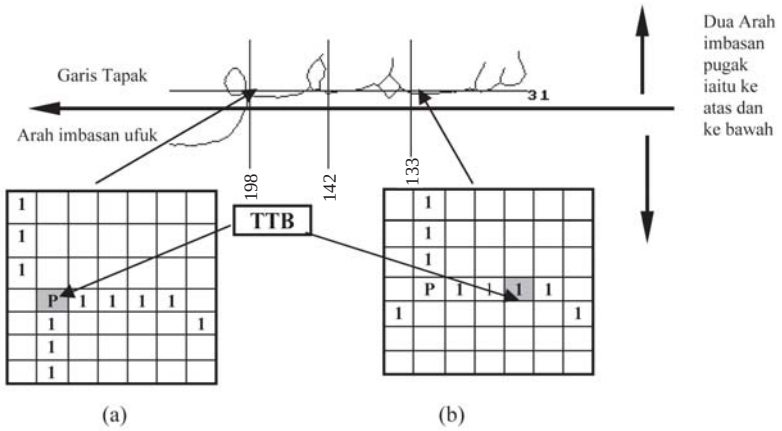
RAJAH 9. Satu contoh perkataan bahasa Malaysia yang ditulis dalam tulisan Jawi (disebut 'syampu' atau „شمڤو“)

Titik tembereng tidak sering berlaku pada garis tapak malah kadang kala boleh berlaku di atas atau di bawah garis tapak yang dikirakan. Garis tapak yang diperoleh daripada kaedah histogram berlaku pada kedudukan piksel ke 31, manakala titik tembereng berlaku pada kedudukan 83, 142, dan 198 yang diperoleh daripada algoritma cadangan yang akan dijelaskan dalam bahagian berikut.

Daripada sifat-sifat data yang terkumpul ketika proses penyingkiran hingar dan penipisan imej teks Jawi, jelaslah bahawa hampir keseluruhan aksara mempunyai berbagai bentuk yang berlainan bergantung kepada posisi dalam satu subperkataan atau perkataan. Adalah tidak mustahil dapat menyaring kelas aksara yang betul. Satu cara untuk memperoleh kelas yang betul adalah dengan menganalisa keseluruhan data ujian pada semua titik tembereng. Dengan mengekalkan penggunaan kaedah UPH untuk mencari garis tapak maka garis tapak yang berpotensi dapat ditentukan. Berasaskan garis tapak ini maka carian titik tembereng TTB dapat dilakukan dengan mengimbas imej teks Jawi menggunakan satu templat bersaiz 7x7. Titik *P*, iaitu pusat piksel, ditandakan sebagai titik calon untuk ditemberengkan. Carian adalah berdasarkan penyemakan piksel hitam dalam kehubungan 8-jiran, iaitu mencari fitur-fitur global dan tempatan. Berdasarkan garis tapak yang ditemui dan titik tengah *P*, semua jiran titik tengah tadi akan direkodkan dan dianalisis.

Bushofa dan Spann (1997) telah memperkenalkan kaedah PTT ini dengan penekanan kepada teks bercetak dan tidak menjadi masalah yang besar untuk memperoleh titik tembereng kerana titik tembereng sering berlaku pada garis

tapak. Seperti yang telah dijelaskan di atas, tidak semua titik tembereng berlaku pada garis tapak bagi teks tulisan tangan. Jadi apa yang perlu dibuat adalah setelah memperoleh garis tapak yang berpotensi, satu lagi tugas yang perlu ditambah adalah mengimbas imej secara memugak dalam lengkongan garis tapak tersebut sama ada ke atas atau ke bawah, sehingga piksel hitam yang paling berpotensi ditemui (lihat Rajah 10). Piksel ini akan dijadikan pusat piksel dan dijadikan calon untuk TTB jika jiran-jirannya memenuhi syarat-syarat tertentu. Terdapat dua keadaan titik tembereng seperti yang dipaparkan dalam Rajah 10(a) dan Rajah 10(b).



Rajah 10. Satu imbasan imej Jawi secara mengufuk dibuat di sepanjang garis tapak dan kemudian satu lagi imbasan secara memugak dibuat untuk mencari titik pusat piksel *P* sama ada ke atas atau ke bawah sehingga menemui piksel yang paling berpotensi, TTB, seperti yang dipaparkan dalam templet bersaiz 7x7. (a) Titik TTB berlaku pada pusat *P*. (b) Titik TTB berlaku bukan pada titik *P*

Untuk kasus yang pertama (lihat Rajah 10(a)), titik tembereng TTB diperoleh secara terus manakala untuk kasus yang kedua (Rajah 10(b)) pula perlu menjalani pemeriksaan histogram unjuran memugak. Pemeriksaan histogram bermula daripada titik *P* menganjur ke kanan sehingga menemui nilai histogram minimum tempatan. Titik ini akan dijadikan sebagai titik tembereng TTB.

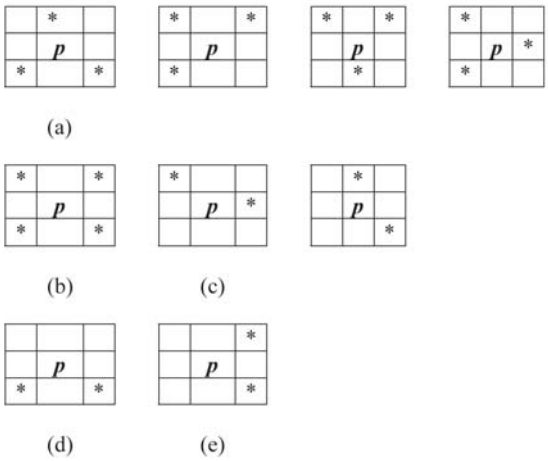
SIFAT-SIFAT TITIK TEMBERENG DATA UJIAN

Semua imej teks jawi yang diperoleh ketika prapemprosesan akan disimpan dalam fail pangkalan data. Sebanyak 280 perkataan telah diambil daripada 28 orang penulis. Terdapat enam jenis cantuman di antara dua aksara seperti yang ditunjukkan dalam Rajah 11. Majoriti adalah berbentuk (a).

Corak cantuman aksara boleh dikesan dengan memeriksa piksel jiran menggunakan hubungan 8-jiran berdasarkan kepada konfigurasi piksel dalam

Rajah 11. Jenis corak cantuman utama bagi aksara Jawi tulisan tangan. (a) simpang tiga yang mendatar, (b) simpang tiga dengan sedikit dongakan, (c) simpang empat, (d) dongakan mengufuk tanpa peralihan arah, (e) dongakan mengufuk dengan peralihan arah, (f) dongakan memugak ke bawah

tetingkap 3x3 berpusatkan pada titik yang diberi, iaitu *p*. Di antara konfigurasi piksel yang terbentuk daripada corak cantuman dalam Rajah 11 diberikan dalam Rajah 12.



RAJAH 12. Contoh Konfigurasi Cantuman Piksel. (a) Cantuman dalam Rajah 11(a) dan (b). (b) Rajah 11(c). (c) Rajah 11(d). (d) Rajah 11(e). (e) Rajah 11(f)

Berdasarkan kepada maklumat Rajah 12 dan Rajah 13 syarat-syarat untuk menembereng titik tembereng dapat dibina, iaitu seperti berikut:
Syarat pertama, iaitu *P1* yang menggambarkan konfigurasi piksel dalam rajah 11(a), Rajah 11(b), serta Rajah 12(a) adalah seperti berikut:

$$(((p1 + p4 + p6) == 3) \| ((p0 + p2 + p6) == 3) \| ((p0 + p2 + p5) >= 1) \| ((p0 + p3 + p6) >= 1))) \quad (4)$$

Syarat kedua, iaitu *P2* pula untuk Rajah 11(c) serta Rajah 12(b) adalah seperti berikut:

$$((p0 + p2 + p4 + p6) == 4). \quad (5)$$

Syarat ketiga, iaitu *P3* untuk Rajah 11(d) serta Rajah 12(c) adalah seperti berikut:

$$(((p0 + p3) == 2) \| ((p1 + p4) == 2)). \quad (6)$$

Syarat keempat, iaitu *P4* untuk Rajah 11(e), Rajah 11(f) serta Rajah 12(d) dan Rajah 12(e) adalah seperti berikut:

$$((p4 + p6) == 2) \| ((p2 + p4) == 2) == 0. \quad (7)$$

Syarat kelima, *P5* adalah seperti berikut:

$$P1 \| P2 \| P3 \| P4. \quad (8)$$

Simbol ‘’||’’ membawa maksud “atau”.

p0	*p1*	*p2*
p7	*p*	*p3*
p6	*p5*	*p4*

RAJAH 13. Tetingkap Uji. *p* menunjukkan piksel pusat

Untuk melaksanakan algoritma ini beberapa maklumat berikut dikekalkan (disari dari kaedah UPH dan PTT) dan dibutirkan:

i. Menentukan garis rujukan atau garis tapak asas. Ia berasaskan kepada analisis ke atas UPH. Kaedah yang digunakan ialah melihat puncak yang mula-mula dijumpai dan kemudian cari garis tapak yang paling bawah dan paling tinggi. Ketebalan garis adalah sama dengan ketebalan garis asas rujukan.
ii. Membina unjuran memugak dengan menjumlahkan piksel hitam.
iii. Langkah seterusnya telah diperbaharui dan ditambah pada algoritma aksara Khella iaitu mencari atau mengenal pasti TTB, iaitu mencari titik mula dan titik akhir sesuatu aksara dengan menggunakan maklumat unjuran mengufuk dan fitur-fitur global atau tempatan. Di akhir langkah ini setiap subperkataan akan ditemberengkan kepada *Aksara Berpotensi* (AB) yang setiap satunya dibatasi oleh dua titik TTB.

Output tatacara penemberengan terubahsuai ini ialah:

a. bilangan aksara yang ditemberengkan;
b. nama fail bagi setiap aksara yang ditemberengkan; dan
c. ketinggian bagi setiap aksara iaitu panjang dan lebarnya.

Dua algoritma ubah suai telah dicadangkan yang akan dijelaskan di bahagian selanjutnya.

PENEMBERENGAN KASUS I

Algoritma cadangan yang pertama, iaitu Algoritma Penemberengan Aksara I untuk menyelesaikan masalah bentuk 4 diberikan seperti berikut:

Algoritma Penemberengan Aksara I
Input: Imej subperkataan atau perkataan.
Output: Aksara-aksara tercerai.
Permulaan algoritma
Langkah 1: Jumlahkan bilangan objek piksel hitam bagi subperkataan atau perkataan semasa secara memugak.
Langkah 2: Cari TTB dan rekodkan berdasarkan syarat-syarat *P1*, *P2*, *P3*, *P4* dan *P5*.
Langkah 3: Berdasarkan UPH yang dikirakan dalam Langkah 1 dan titik fitur global atau tempatan dalam Langkah 2, titik pisahan ditentukan dengan mencari jurang-jurang minimum tempatan yang mula-mula ditemui (setelah dianjakkan ke arah kanan).
Langkah 4: Pisahkan aksara atau AK pada dua TTB yang berturut-turut, iaitu dua titik penggal yang ditemui secara berturut-turut.
Langkah 5: Simpan semua aksara yang telah dipisahkan ke dalam fail.
Tamat algoritma.

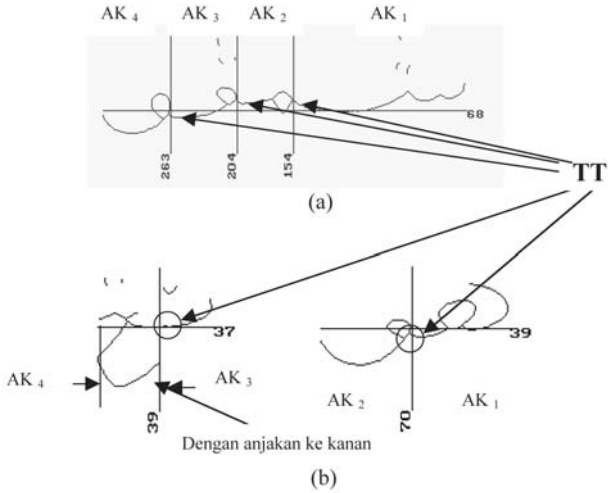
Dua contoh hasil penemberengan menggunakan algoritma di atas diberikan dalam Rajah 14. Rajah 14(a) menunjukkan garis tapak anggaran ialah pada baris 68 dari utara. Sebanyak 3 fitur telah ditemui pada perkataan " شمڤو ". Fitur-fitur ini berlaku pada kedudukan titik 154, 204, dan 263 daripada kanan ke kiri dan dikenal pasti sebagai TTB setelah menyemak nilai jiran histogram di sebelah kanan masing-masing dengan nilai ambang kurang daripada jiran sebelumnya.

Garis pemisahan dibinakan secara memugak pada titik TTB. Contoh kedua (lihat Rajah 14(b)) menunjukkan perkataan hutang terlebih dahulu dipisahkan kepada 2 subperkataan, iaitu (1) subperkataan " هو " dan (2) subperkataan " تغ ", yang dapat dipisahkan dengan jurang. Selanjutnya subperkataan (1) dan (2) diproses menggunakan algoritma cadangan, iaitu mencari fitur-fitur titik simpang, titik silang. Sedikit anjakan ke kanan dilakukan supaya dapat memisahkan aksara dengan betul.

PENEMBERENGAN KASUS II

Untuk mengatasi masalah bentuk 2, algoritma berikut telah dicadangkan:

Algoritma Penemberengan Aksara II
Input: Imej subperkataan atau perkataan.
Output: Aksara-aksara tercerai.



RAJAH 14. Hasil Algoritma Cadangan bagi Rajah 7. (a) Tanpa anjakan ke kanan. (b) Dengan anjakan

Permulaan algoritma

Langkah 1: Jumlahkan bilangan objek piksel hitam bagi subperkataan atau perkataan semasa secara mengufuk.

Langkah 2: Cari TTB dan direkodkan berdasarkan syarat-syarat *P1*, *P2*, *P3*, *P4*, dan *P5*.

Langkah 3: Berdasarkan UPH yang dikirakan dalam Langkah 1 dan titik fitur global atau tempatan dalam Langkah 2, titik pisahan ditentukan dengan mencari jurang-jurang minimum tempatan yang mula-mula ditemui (setelah dianjakkan ke arah kanan).

Langkah 4: Pisahkan aksara AK pada dua TTB yang berturut-turut, iaitu dua titik penggal yang ditemui secara berturut-turut.

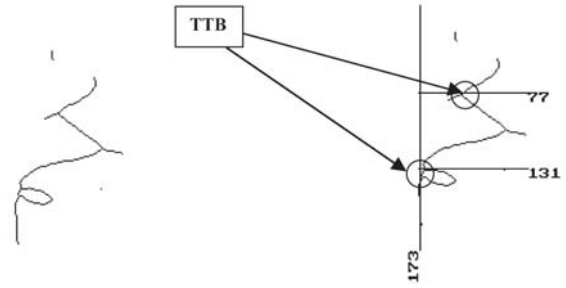
Langkah 5: Simpan semua aksara yang telah dipisahkan dalam fail pangkalan data.

Tamat algoritma.

Hasil daripada algoritma ini digambarkan di dalam Rajah 15. Contoh Perkataan 'Najm' atau ''نجم''yang dipaparkan adalah satu contoh masalah yang paling sukar diselesaikan. Menerusi algoritma cadangan, titik TTB dapat ditentukan. Bermula dengan menentukan histogram secara mengufuk, menentukan garis tengah memugak (dalam contoh yang diberi garis penengah berlaku pada kedudukan 173), dan kemudian mengesan fitur-fitur titik simpang atau titik menyilang, titik perubahan tanda dalam arah kanan ke kiri.

Terdapat dua fitur telah ditemui, iaitu pada kedudukan yang ditandakan dengan bulatan. TTB bagi subperkataan tersebut berlaku pada kedudukan titik 77, iaitu di antara aksara " ن" dan " جـ" dan titik 131, di antara " جـ" dan

“ ـسـ ”. Semakan nilai jiran histogram mengufuk didapati kurang daripada nilai ambang.



(a) (b)

RAJAH 15. Bentuk Ligatur. (a) Perkataan jawi asal (b) Hasil algoritma II

HASIL UJI KAJI

Algoritma penemberengan yang dibincangkan di dalam bahagian sebelumnya dibandingkan menggunakan 140 contoh perkataan teks Jawi untuk setiap algoritma menjadikan sejumlah 420 perkataan semuanya. Hasil perbandingan di antara dua pendekatan iaitu pendekatan cadangan dan pendekatan Khella ditunjukkan dalam Jadual 1, dan ia jelas menunjukkan bahawa algoritma cadangan telah mencapai hasil yang lebih baik berbanding dengan algoritma Khella (1992).

JADUAL 1. Hasil penemberengan algoritma cadangan

Algoritma	Contoh Ujian	Pendekatan Cadangan	Pendekatan Khella
Label	140	90.71%	-
I	140	90.00%	75.7%
II	140	75.71%	-
Purata	420	85.47%	-

Secara puratanya sejumlah 85.47% daripada 420 perkataan dapat ditemberengkan dengan baik. Ralat yang berlaku adalah disebabkan kesalahan daripada sampel penulis, iaitu 10.48% seperti salah kedudukan titik (juzuk sekunder), kesalahan daripada algoritma penipisan, iaitu 3.81% seperti menghasilkan ekor-ekor yang tidak dikehendaki seperti yang pernah dilaporkan oleh Khairuddin (2000), dan kesalahan dalam aturcara, iaitu sebanyak 0.24% seperti gagal mengesan TTB atau terlebih serta terkurang tembereng (seperti aksara “ ـسـ ” dan “ ـصـ ”).

Hasil uji kaji menunjukkan algoritma cadangan lebih baik daripada pendekatan Khella (1992) atas sebab-sebab berikut:

i. Khella tidak menyelesaikan masalah aksara bertindan secara memugak (lihat Rajah 5 dan Rajah 6), dan masalah ligatur (lihat Rajah 15);
ii. Kaedah Khella gagal mengesan titik tembereng bagi aksara bergelung seperti dalam contoh Rajah 8.

Jadual 1 di atas menunjukkan bahawa pendekatan Khella tidak menyelesaikan masalah aksara bertindan secara memugak tetapi dapat diselesaikan oleh algoritma cadangan yang disebut algoritma Label. Algoritma ini dapat memberikan penemberengan yang betul sebanyak 90.71% daripada 140 perkataan atau subperkataan yang diuji.

Khella juga tidak menyelesaikan masalah aksara ligatur, manakala algoritma cadangan yang disebut algoritma II dapat menyelesaikan 75.71% daripada perkataan atau subperkataan yang diujikan.

Hasil uji kaji juga dapat mengesan satu lagi bentuk lazim aksara tulisan tangan yang tidak dapat ditemberengkan dengan betul. Rajah 16 menunjukkan satu contoh subperkataan 'niver' atau " نيور " yang gagal ditembereng oleh algoritma cadangan.

RAJAH 16. Contoh aksara bercantum

Bentuk ini (kalau boleh disebut bentuk lazim ke 5) mudah didapati di dalam mana-mana teks Jawi atau Arab, dan berlaku apabila percantuman di antara satu aksara tunggal iaitu aksara " ﻭ " dengan aksara " ﺭ " (seperti di dalam contoh yang dilorekkan dengan tanda bulatan hitam) membentuk dua aksara yang bercantum, sedangkan sifat aksara-aksara itu tidak boleh bercantum.

KESIMPULAN

Algoritma penemberengan cadangan bagi menemberengkan baris, perkataan, dan aksara Jawi telah menggunakan gabungan tiga algoritma, iaitu algoritma UPH, PKT dan PTT telah dilaksanakan oleh masing-masing Khella(1992), Pitas (1993) dan, Bushofa dan Spann (1997). Beberapa langkah ubah suai telah dilakukan ke atas algoritma Bushofa dan Spann untuk disesuaikan dengan aksara Jawi tulisan tangan bagi menyelesaikan masalah bentuk 1, 2, dan 4.

Hasil uji kaji menunjukkan algoritma cadangan lebih baik berbanding dengan UPH dan merupakan satu lagi sumbangan kepada penyelidikan PTOAJ.

Seterusnya, imej aksara Jawi yang telah ditemberengkan boleh dinormalkan, dilakukan proses penyarian fitur dan seterusnya proses pengecaman.

Penyelidikan akan datang akan memberikan tumpuan kepada masalah bentuk lazim yang kelima. Ada penunjuk yang menyatakan bahawa penemberengan dua aksara bercantum sedemikian boleh diselesaikan di bawah andaian-andaian tertentu seperti yang pernah diselesaikan oleh Shaaban (1996).

Secara keseluruhannya dapat disimpulkan bahawa penemberengan aksara bercetak berbanding dengan aksara tulisan tangan adalah jauh berbeza. Hal ini adalah disebabkan aksara bercetak mempunyai bentuk tidak berubah dan mudah untuk memperoleh *TTB* seperti yang dibuktikan oleh pendekatan Bushofa dan Spann(1997). Misalnya aksara ‘’ن” tengah yang apabila menggunakan pendekatan mereka menghasilkan penemberengan berbentuk “ـنـ” berbanding dengan pendekatan cadangan yang berbentuk “ـنـ”. Untuk menyeragamkan hasil yang diperoleh maka satu lagi modul diperlukan untuk memperoleh bentuk yang sama dan akan ditimbangkan sebagai penyelidikan akan datang.

RUJUKAN

Al-Badr, B. & Mahmoud, S. A. 1995. Survey and Bibliography of Arabic Optical Text Recognition. *Signal Processing Vol. 41: 49-77.*

Al-Badr, B. 1992. On the Recognition of Arabic Documents. Laporan Teknik *# 93-10-1, Jabatan Sains Komputer dan Kejuruteraan, Universiti Washington.* Seattle: Universiti Washington.

Altuwaijri, M. M. Bayoumi, M. A. 1995. A New Recognition System for Multi-Font Arabic Cursive Words. *Pascasidang ICECS'95. Ammam, Jordan, Disember: 17-21.*

Bushofa, B. M. F. & Spann, M. 1997. Segmentation and recognition of Arabic characters by structural classification. *Image and Vision Computing Vol. 15: 167-179.*

Khairuddin Omar & Ramlan Mahmod. 1999. Sejarah perkembangan pengecaman teks optik Arab/Jawi. Siri Laporan Teknik. *Fakulti Teknologi dan SainsMaklumat, UKM, FTSM/Ogos 1999/LT 77.*

Khairuddin Omar. 1999a. Pengecaman teks optik Arab/Jawi: Satu tinjauan. *Pascasidang Seminar Teknologi Komunikasi Canggih dalam Islam 1999 Anjuran Pusat Sains Trengganu dan Jabatan Hal Ehwal Agama Islam Trengganu, pada 21, 22, dan 23 November 1999 di Pusat Sains Trengganu, Kuala Trengganu.*

Khairuddin Omar. 1999b. Ulasan Karya Pendekatan Penemberengan Aksara Arab. Siri Laporan Teknik. *Fakulti Teknologi dan Sains Maklumat, UKM, FTSM/Julai 1999/LT 74.*

Khairuddin Omar. 2000. *Pengecaman Tulisan Tangan Teks Jawi Menggunakan Pengelas Multiaras.* Tesis Doktor Falsafah, Jabatan Sains Komputer, Fakulti Sains Komputer dan Teknologi Maklumat, UPM. Serdang: Universiti Putera Malaysia.

Khella, F. 1992. *Analysis of Hexagonally Sampled Images with Application to Arabic Cursive Text Recognition*. Tesis Sarjana Falsafah, Jabatan Kejuruteraan Elektrik, Universiti Bradford. Bradford: Universiti Bradford.

Pitas, I. 1993. *Digital image processing algorithms*. New Jersey: Prentice Hall.

Romeo-Pakker, K., Miled, H. & Lecourtier, Y. 1995. A new approach for Latin/ Arabic character segmentation. *Pascasidang 3rd International Conference Analysis & Recognition, Montreal, Canada, 14-16, Ogos, 1995, Vol. 2, ms. 874-877.*

Shaaban, Z. 1996. *Algorithms for Off-line Upper-case Hand-written Text Recognition and Its Associated Processes.* Tesis PhD., Fakulti Sains dan Sistem Maklumat, UTM Skudai:Universiti Teknologi Malaysia.

Khairuddin Omar
Jabatan Sains dan Pengurusan Sistem
Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor D. E.
ko@ftsm.ukm.my

Ramlan Mahmod
Jabatan Multimedia
Universiti Putra Malaysia
43400 UPM Serdang, Selangor D. E.
ramlan@fsktm.upm.edu.my

Md. Nasir Sulaiman
Jabatan Sistem Maklumat
Universiti Putra Malaysia
43400 UPM Serdang, Selangor D. E.
nasir@fsktm.upm.edu.my

Abdul Rahman Ramli
Jabatan Kejuruteraan Elektronik dan Komputer
Universiti Putra Malaysia
43400 UPM Serdang, Selangor D. E.
arr@eng.upm.edu.my