PERANCANGAN PROGRAM APLIKASI PENGENALAN TEKS MENGGUNAKAN *FUZZY LOGIC*

Ngarap Im Manik
Jurusan Matematika FST BINUS University,
Jln.K.H Sjahdan No.9 Palmerah, Jakarta 11480, Indonesia
e-mail : manik@binus.edu

Abstract

This paper discusses application program design for character recoqnition with size font that medley to maximize accuration. Image processing method usually with used that is grayscaling, thresholding, filtering, segmentation, and streching. To the effect of method that is subject to be update image so as simple. This paper approach character recoqnition with use fuzzy logic. By marks sense this approaching, fonts on text can be recognized one about one bases characteristic that its proprietary. Output programs that resulting as text that readily been processed more and of examination that is done, letters recoqnition system one by one have accuration as 96,67%. on standard document recoqnition, increase that accuration resulting as big as 96,13%.
***Keywords** : Character recoqnition, fuzzy logic.*

1. PENDAHULUAN

Teknologi yang terus berkembang membuat sistem komputerisasi bergerak dengan cepat, namun hal ini tidak seimbang dengan kemampuan manusia memindahkan data secara manual ke dalam komputer untuk dapat diolah lebih lanjut. Suatu sistem dikembangkan untuk menjawab permasalahan tersebut. Sistem tersebut dinamakan *Optical Character Recognition* (*OCR*). *OCR* merupakan aplikasi dari teknologi pengenalan teks, yaitu suatu teknologi yang mampu mengenali teks pada citra digital dan mengalihkannya pada dokumen digital. Dalam perkembangannya, aplikasi *OCR* seringkali digunakan pada berbagai jenis dokumen, dimana beberapa dokumen memiliki ukuran *font* yang berbeda-beda satu dengan yang lainnya. Hal ini menyebabkan aplikasi *OCR* yang ada menjadi kurang maksimal dalam mengenali teks. Oleh karena itu, diperlukan sebuah program yang dapat mengenali teks dengan ukuran *font* yang bervariasi yang menghasilkan tingkat akurasi yang tinggi.(Zand, 2008; Sulaiman, 2007).

Dalam merancang suatu program pengenalan teks, pada tahap pengolahan citra digunakan beberapa metode, antara lain: *grayscaling*, *thresholding*, *filtering*, *segmentation*, dan *stretching*. Sedangkan pada tahap pengenalan pola digunakan pendekatan *fuzzy logic*. Pendekatan ini mampu memisahkan komponen-komponen penyusun sebuah citra ke dalam bentuk nilai keanggotaan, dan diproses dengan aturan-aturan yang ada. Setelah proses berhasil maka pola tersebut diterima sebagai keanggotaan huruf.

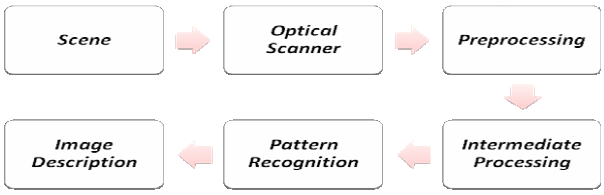
Tujuan dari perancangan program ini adalah menciptakan sistem pengenalan teks dengan tingkat akurasi tinggi pada ukuran *font* yang bervariasi dan mengembangkan pengetahuan mengenai *fuzzy logic* di bidang teknologi informasi. Sedangkan manfaat yang diharapkan adalah dengan adanya aplikasi ini, diharapkan setiap orang dapat lebih mudah mengolah informasi tanpa harus memasukkan data dengan cara mengetik, sehingga pekerjaan berlangsung dengan cepat. Dan memberi kemudahan bagi pengguna dengan mempercepat pengolahan data teks yaitu pada teks dengan ukuran *font* yang bervariasi.

2. TINJAUAN PUSTAKA

2.1 Pengenalan Teks

Teknologi pengenalan teks merupakan teknologi yang mampu mengenali teks pada citra digital dan mengalihkannya pada dokumen digital. Aplikasi dari teknologi pengenalan teks ini dikenal dengan nama *Optical Character Recognition* (*OCR*). *OCR* sendiri digunakan untuk mengenali teks hasil cetakan mesin *(Machine-Printed Text)*. *OCR* dipatenkan pada tahun 1929 di Jerman oleh Gustav Tauschek. Pada saat itu, diterapkan pada mesin yang menggunakan alat optik (sekarang ini umumnya kita menggunakan alat optik berupa *scanner*). Saat ini konsep dasar dari *OCR* banyak digunakan di beberapa aplikasi pengenalan teks.

Berikut ilustrasi dari proses *OCR*:



Gambar 1. Proses pada *OCR* (Tri, 2007, p5)

Prinsip kerja dari aplikasi *OCR* adalah sebagai berikut.

1. Memasukkan dokumen berisi teks (teks cetakan mesin) ke dalam alat optik (*scanner*) sehingga didapat sebuah *file* citra.
2. *File* citra tersebut diproses menggunakan perangkat lunak aplikasi pengenalan teks, di mana perangkat ini melakukan proses pengenalan terhadap karakter-karakter yang ada pada *file* citra tersebut.
3. Keluaran dari perangkat lunak aplikasi pengenalan teks ini berupa file teks yang berisi karakter-karakter yang telah dikenali dan siap untuk diolah lebih lanjut.

Oleh karena itu, tingkat keberhasilan dari perangkat lunak aplikasi pengenalan teks ini sangat bergantung dari sejumlah faktor berikut.(Gunawan T, 2005)

1. Kualitas gambar teks yang ada pada dokumen yang dibaca serta tingkat kompleksitasnya (ukuran, format teks, warna, latar belakang).
2. Kualitas alat optik yang dipakai (*scanner*).
3. Kualitas perangkat lunak aplikasi pengenalan teks itu sendiri.

Dalam penelitian mengenai pengenalan teks digunakan beberapa pendekatan, yaitu:

1. Pendekatan statistik (*statistical approach*)
2. Pendekatan sintaktik (*syntactic approach*)
3. Pendekatan *neural network,* dan
4. Pendekatan *fuzzy logic*

2.2 Fuzzy Logic

Fuzzy logic merupakan suatu metode yang digunakan dalam proses pengambilan keputusan dengan cara memetakan suatu ruang *input* ke dalam ruang *output*. Sistem ini diperkenalkan pertama kali oleh Prof. Lotfi Zadeh dari Universitas California, Barkeley tahun 1962, dimana pada saat itu *boolean logic* hanya mengenal dua keadaan yaitu : ya/tidak, *ON/OFF*, *High/Low* atau hanya mempunyai logika 0 dan 1 saja. Sedangkan kondisi nyata di alam ini bukan hanya ya (1, *high*, *on*) atau tidak (0, *low*, *off*) tetapi seluruh kemungkinan diantara 0 dan 1, sehingga untuk mengenal kondisi ini kita tidak dapat menggunakan *boolean logic* tetapi dengan menggunakan *fuzzy logic.*(Yohanes, 2002)

Proses Sistem *Fuzzy*

Pada sistem *fuzzy* terdapat tiga proses sebagai berikut.

Fuzzification

Proses ini berfungsi untuk mengubah masukan-masukan berupa nilai analog atau yang nilai kebenarannya bersifat pasti (*crisp input*) menjadi nilai *fuzzy*, yang digunakan sebagai *fuzzy input*. *Fuzzy input* ini berupa nilai *linguistic* yang semantiknya ditentukan berdasarkan fungsi keanggotaan tertentu. Jika terdapat suatu nilai analog yang menjadi *input* pada proses *fuzzy* maka *input* tersebut dimasukkan pada batas *scope/domain* sehingga didapatkan suatu nilai fungsi keanggotaan. Nilai fungsi keanggotaan inilah yang menentukan proses pengambilan keputusan selanjutnya. Salah satu metode yang digunakan oleh proses ini dalam pengenalan teks adalah *feature extraction*. Metode ini digunakan untuk mendapatkan karakteristik dari suatu citra dengan melihat bentuk dasar objek pada citra tersebut. Tujuan metode ini adalah untuk melakukan pengukuran terhadap hal-hal yang membedakan pola masukkan sehingga objek pada citra yang satu dan yang lain dapat dibedakan.

Rule Evaluation

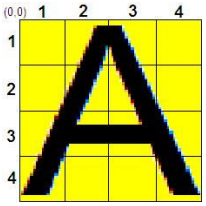
Proses ini digunakan untuk mencari nilai *fuzzy output* dari *fuzzy input*. Jika terdapat suatu nilai *fuzzy input* dari proses *fuzzification* nilai tersebut akan dimasukkan ke dalam *rule* yang telah dibuat untuk mendapatkan nilai *fuzzy output*. Pada proses inilah suatu sistem dapat dikatakan pintar atau tidak. Jika *rule* yang dibuat tidak pintar maka sistem yang dikontrol menjadi kacau dan objek yang seharusnya dikenali menjadi tidak dapat dibaca. Dengan *rule* yang ada diperoleh nilai *fuzzy* yang digunakan dalam membantu pengambilan keputusan.

Defuzzification

Pada tahap inilah pengambilan keputusan dilakukan. Nilai yang didapatkan berupa nilai *crisp*, yaitu 0 atau 1. Proses *defuzzification* melakukan suatu fungsi *output* yang memproses nilai *fuzzy* yang berasal dari *rule evaluation* sehingga keputusan akhir dapat dilakukan. Fungsi *output* ini dilakukan dengan mencocokan nilai-nilai yang ada dengan *threshold* yang ditentukan. Sebuah *input* pada proses *fuzzy* akan diterima sebagai anggota himpunan *fuzzy* jika memiliki nilai keanggotaan yang melewati batas *threshold* yang ada.

2.3 Pendekatan *Fuzzy Logic* pada Pengenalan Teks

Pendekatan *fuzzy logic* merupakan salah satu cara pendekatan yang digunakan dalam pengenalan pola. Pada pendekatan ini digunakan metode pemisahan karakter menjadi daerah-daerah kecil yang disebut *box-method*. Hal ini terlihat seperti yang terlihat pada gambar berikut.



Gambar 2. *Box-Method*

Dari *box* hasil pembagian tersebut akan didapat suatu nilai yang digunakan sebagai nilai *input* fungsi keanggotaan pada proses *fuzzy*. Nilai-nilai yang didapatkan berasal dari nilai koordinat *pixel* terdapat warna hitam (*i*, *j*), dimana pada pojok kiri atas masing-masing *box* memiliki koordinat *pixel* (*0*,*0*). Nilai *vector distance* masing-masing *pixel* diperoleh dengan rumus:

$$d_{kb} = \sqrt{i^2 + j^2}$$

Nilai *vector distance* tersebut kemudian dihitung sebagai normalisasi dari total semua *pixel* masing-masing *box* dengan rumus:

$$\gamma_b = \frac{1}{n}\sum_{k=1}^{n} d_{kb}$$

Adapun *n* merupakan jumlah *pixel* dalam *box* dan *b* merupakan nomor *box*. Nilai ini digunakan sebagai nilai fungsi keanggotaan pada sistem *fuzzy*.

Dalam sistem *fuzzy*, dihitung nilai *mean* (*m*) dan *variance* (σ^2) menggunakan rumus:

$$m_i = \frac{1}{N_i}\sum_{j=1}^{N_i} f_{ij}$$

$$\sigma_i^2 = \frac{1}{N}\sum_{j=1}^{N_i}(f_{ij} - m_i)^2$$

Dimana N_i adalah jumlah sampel dalam *cluster* ke-*i* dan f_{ij} merupakan nilai fungsi keanggotaan dari masing-masing box dengan karakter ke-*j*. Setelah didapatkan nilai *mean* dan *variance*, maka nilai tersebut dapat dibandingkan dengan nilai fungsi keanggotaan masing-masing *box* yang dicari dengan rumus:

$$\mu_{xi} = e^{\frac{-(x_i - m_i)^2}{\sigma_i^2}}$$

Fungsi keanggotaan (μ_{xi}) ini menyatakan tingkat kesesuaian pola antara karakter yang akan dikenali dengan *knowledge base* yang ada. Dimana *x* merupakan nilai fungsi keanggotaan dari karakter yang tidak diketahui.

Pada *library* dengan jumlah yang terbatas, proses *fuzzy* kurang bekerja dengan baik. Hal ini disebabkan karena beberapa aturan *fuzzy* memiliki nilai *variance* yang terlalu kecil dan juga nilai *variance* yang terlalu besar. Untuk menanggulangi hal ini, fungsi keanggotaan tersebut dibagi menjadi dua persamaan berbeda, dengan rumus:

$$\mu_{xi} = e^{\frac{-(x_i - m_i)^2}{\sigma_i^2}} \quad \text{untuk } \sigma_i^2 \geq 1$$

$$\mu_{xi} = e^{-(x_i - m_i)^2 \sigma_i^2} \quad \text{untuk } \sigma_i^2 < 1$$

Selanjutnya dicari nilai μ_{xi} masing-masing karakter dengan mencari nilai rata-rata semua *box*, dengan rumus:

$$\mu_{av}(r) = \frac{1}{c} \sum_{j=1}^{c} e^{\frac{-(x_j - m_j(r))^2}{\sigma_j^2(r)}}$$

Adapun *c* berarti jumlah total semua *box* dan *r* = merupakan penomoran karakter-karakter yang ada pada *knowledge base*. Sebuah karakter dapat dikenali dengan mencari nilai $\mu_{av}(r)$ yang paling besar atau mendekati angka 1 (satu) (Fernando H, 2003; Kosko, 2005).

3. METODE PENELITIAN

2.1 Analisis Masalah

Pada proses pengenalan teks, seringkali terjadi berbagai masalah. Masalah tersebut disebabkan karena banyaknya ragam ukuran *font*. Keragaman ukuran ini *font* mengakibatkan banyaknya keragaman warna, ukuran, dan pola huruf. Selain daripada itu, pada gambar tertentu memiliki *noise* berupa bintik berwarna hitam ataupun berwarna putih. Hal-hal tersebut menyulitkan pembacaan huruf-huruf yang ada.

2.2 Analisis Metode

Ada beberapa metode yang umum digunakan pada sistem pengenalan teks. Pada pengolahan citra digunakan metode *grayscaling*, *thresholding*, *filtering*, *segmentation*, dan *stretching*. Sedangkan pada pengenalan pola huruf digunakan aturan *fuzzy*.

Aturan *fuzzy* pada pengenalan teks digunakan untuk melakukan pengenalan huruf. Pada aturan *fuzzy*, pola huruf hasil pelatihan disimpan sebagai *library*. *Library* tersebut digunakan pada saat mencocokan pola huruf. Jika huruf yang dikenali sesuai dengan huruf pada *library*, maka huruf dikenali. Pada aturan *fuzzy* juga ditentukan besar nilai toleransi perbedaan pola. Niai toleransi dibentuk dari perbedaan pola-pola huruf yang tersimpan. Jika ciri huruf yang akan dikenali sesuai dengan batas toleransi pola huruf tertentu, maka ciri huruf tersebut masih dipertimbangkan sebagai anggota huruf tersebut.

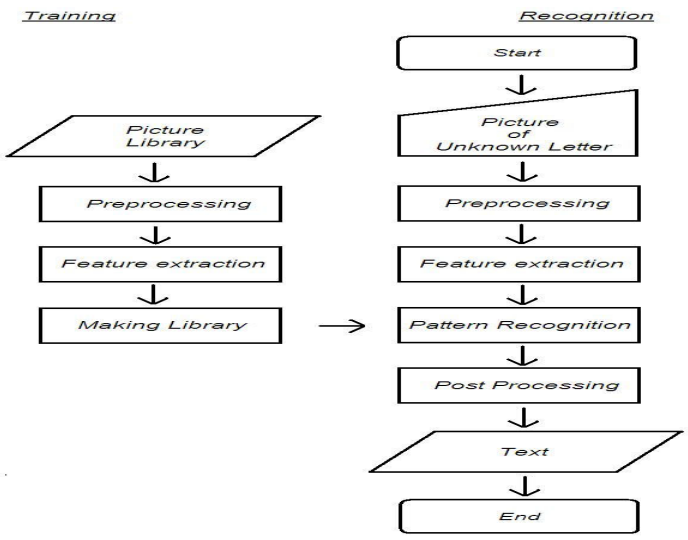
2.3 Analisis Ruang Lingkup

Pada perancangan program pengenalan teks ini memiliki batasan-batasan tersendiri. Program ini ditujukan untuk pengenalan huruf cetak. Ukuran *font* yang digunakan antara 8 hingga 72. Ukuran ini digunakan sebagai standarisasi ukuran *font* pada program-program pada umumnya. Jenis *font* yang digunakan adalah *arial* dengan *style* normal. Huruf *arial* dipilih karena huruf ini cukup sering digunakan. Selain itu, huruf ini memiliki bentuk yang sederhana. Pembatasan berukutnya adalah pada resolusi citra digunakan sebesar 96 *pixel*/*inch*. Ukuran tersebut biasa digunakan pada citra digital murni, bukan pada citra yang ditangkap oleh alat optik.

2.4 Pemecahan Masalah

Menghadapi permasalahan-permasalahan yang terjadi, untuk merancang program pengenalan teks digunakan metode-metode yang saling mendukung. Pada tahap pertama, citra yang akan dikenali diolah menggunakan *grayscaling*. Pada tahap ini komponen warna citra telah disederhanakan menjadi keabuan. Setelah itu, citra melewati proses *thresholding,* sehingga pada citra hanya akan terdapat warna hitam untuk huruf yang akan dikenali dan warna putih untuk latar belakang (Fruceci, 2008). Proses berikutnya yaitu mereduksi *noise* dengan *filtering*. Pada tahapan berikutnya, huruf-huruf pada citra dipisahkan satu per satu dengan proses *segmentation*. Untuk menghasilkan *output* yang teratur rapi, pada proses ini juga dilakukan pengukuran tinggi huruf dan jarak spasi. Pengukuran ini akan menghasilkan suatu informasi yang digunakan untuk membedakan apakah huruf tersebut termasuk huruf besar atau kecil dan juga apakah huruf-huruf tersebut memiliki jarak spasi satu sama lain. Setelah proses tersebut, dilakukan proses *streching* untuk menyesuaikan ukuran huruf. Huruf-huruf yang telah dinormalisasi siap untuk dikenali menggunakan aturan *fuzzy*. Dari metode-metode yang digunakan, diharapkan bahwa masalah-masalah yang terjadi pada sistem pengenalan teks dapat ditanggulangi. Sehingga teks *output* program yang dihasilkan sesuai atau mendekati teks asli yang terdapat pada citra.

Struktur sistem pengenalan teks secara garis besar dapat dilihat pada gambar berikut.



Gambar 3. Rancangan Sistem Pengenalan Teks

4. HASIL & PEMBAHASAN

Hasil

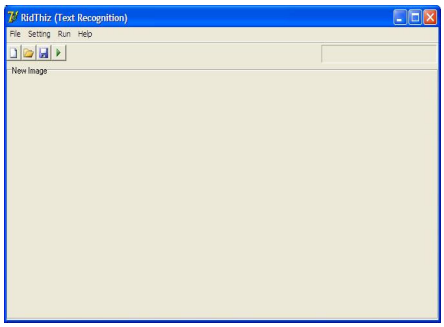
Hasil rancangan program ini dapat diaplikasikan dengan spesifikasi perangkat keras sebagai berikut : Prosesor *Intel Atom* 1,6 *GHz,* Memori *RAM* 1 *GHz, VGA Card OnBoard, Hardisk* 120 *GB, Monitor Keyboard, Mouse* dan dengan spesifikasi piranti lunak yang digunakan *Microsoft Windows XP, Borland Delphi* 7.0 dan *MsPaint.*(Fadlisyah, 2008).

Menu Utama

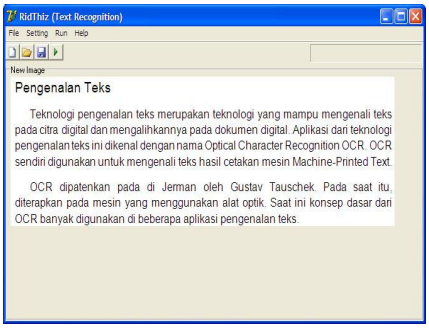
Menu Utama merupakan bagian utama program ini. Dalam menu utama ada beberapa pilihan menu yang dapat dipilih oleh pengguna, yaitu:

- *File* : *New*, *Open Image*, *Save Text As*, *Exit*
- *Setting* : *Font*, *Labeling*, *Fast Recognize*
- *Run* : *Training*, *Recognize*
- *Help* : *Program Help*, *About*

Jika program tersebut di Run maka beberapa tampilan menu program seperti yang ditunjukkan berikut ini :



Gambar 4.Tampilan Menu Utama



Gambar 5 Tampilan Setelah *Open Image*

Dalam melakukan evaluasi program, pada pengujian pertama dilakukan beberapa konfigurasi yang berasal dari pengaturan menu "*Training*". Pengaturan tersebut berupa "*Type*" (*Font Size Type*), Nilai "*Box*" (*Matrix Box*), dan Nilai "*Pixel Per Box*" (*Pixel Per Box*). Setelah konfigurasi selesai, program ini diuji untuk mengenali huruf satu per satu. Pengujian lain yang dilakukan adalah untuk mengenali teks pada citra berupa dokumen standar.

Pengujian terakhir yang dilakukan adalah pengujian pengaruh penggunaan *labeling* dalam sistem ini. Berikut hasil pengujian terhadap sistem pengenalan teks serta pembahasan hasil evaluasi tersebut.

Pengujian Nilai "*Box*" dan "*Pixel Per Box*"

Pengujian ini dilakukan untuk mencari nilai "*Box*" dan nilai "*Pixel Per Box*" yang paling optimal. Pengujian ini dilakukan menggunakan sampel citra berisi huruf besar (A-Z) dan huruf kecil (a-z) dengan *Font* = "12" dan *Type* = "*All*". Berikut ini data hasil pengujian nilai "*Box*" dan "*Pixel Per Box*".

Tabel 1. Hasil Pengujian Nilai "*Box*" dan "*Pixel Per Box*"

Box	Pixel	Pixel	Known	Rate
3	10	30	44	84,62
4	8	32	48	92,31
5	**8**	**42**	**50**	**96,15**
6	6	36	48	92,31
7	6	40	49	94,23
8	4	32	42	80,77
9	4	36	47	90,38
10	3	30	47	90,38

Keterangan:
Nilai "Pixel" = Box * Pixel Per Box
Nilai "Pixel Per Box" dilakukan dengan sampling sebanyak 1 (satu) kali. Sampling dilakukan dengan perkiraan nilai "Pixel" yang tidak terlalu kecil dan juga tidak terlalu besar (berkisar 30-45).
Dari tabel hasil pengujian di atas menunjukkan tingkat akurasi terbesar diperoleh pada *Box* = "5' dan *Pixel Per Box* = "8", yakni sebesar 96,15%.

Pengujian Pengenalan Huruf

Pengujian ini dilakukan untuk mengenali huruf satu per satu. Tujuan dari pengujian ini adalah untuk memperoleh tingkat akurasi yang paling optimal dari **sistem *fuzzy*** yang ada. Sampel pengujian menggunakan citra pada kondisi terbaik yaitu citra berisi huruf besar (A-Z) dan huruf kecil (a-z). Setiap huruf dipisahkan dengan spasi. *Font* yang digunakan adalah 8 sampai 72. Sedangkan pengaturan sistem *training* digunakan *Type* = "*All*", *Box* = "5", dan *Pixel Per Box* = "8". Berikut ini data hasil pengujian pengenalan huruf. Hasil pengujian menunjukkan tingkat akurasi rata-rata sebesar 96,67%.

Pengujian Dokumen Standar

Pengujian ini menggunakan sampel berupa citra berisi teks yang terdiri dari 3 (tiga) macam dokumen. Masing-masing dokumen memiliki 3 (tiga) ukuran *font* yang berbeda. Sedangkan ukuran *font* yang digunakan untuk pengujian ini adalah 10, 16, dan 24. Pada pengaturan sistem training digunakan *Type* = "*All*", *Box* = "5", dan *Pixel Per Box* = "8".
Dari hasil keluaran dapat disimpulkan bahwa pengujian pada dokumen standar menunjukkan tingkat akurasi sebesar 96,13%.

Pengujian *Labeling*

Pengujian ini bertujuan untuk melihat seberapa besar pengaruh *labeling* pada proses *segmentation* dalam sistem pengenalan teks. Pengaruh yang diujikan tersebut berupa waktu proses dan jumlah pemisahan huruf. Sampel yang digunakan pada pengujian ini sama dengan sampel yang digunakan pada dokumen standar yakni sebanyak 3 (tiga) buah sampel. Berikut tabel hasil pengujian *labeling*.

Tabel 2. Hasil Pengujian *Segmentation*

	Font Size "10"				*Font Size* "16"				*Font Size* "24"			
	Non		*With*		*Non*		*With*		*Non*		*With*	
Citr	*T*	*E*	*T*	*E*	*T*	*E*	*T*	*E*	*T*	*E*	*T*	*E*
Sa	1	3	1	1	1	1	1	0	1	0	2	0
Sa	1	4	1	3	1	3	1	0	1	1	2	0
Sa	8	1	8	5	1	6	1	1	1	0	1	0
Tot	2	2	3	9	3	1	4	1	4	1	6	0

Tabel 3. Hasil Pengujian Pengaruh *Labeling*

	Time (%)			*Error* (%)		
	"10"	"16"	"24"	"10"	"16"	"24"
Total	7,82	14,92	27,49	0,86	0,70	0,08
Average	**16,74**			**0,55**		

Keterangan:
Tanda kutip ("") = Ukuran *font*
T = Time = Waktu Proses ; *Average* = Nilai rata-rata
E = Error = Jumlah pemisahan huruf yang tidak berhasil dilakukan
Non L = Non Labeling = Proses *segmentation* tanpa *labeling*
With L = With Labeling = Proses *segmentation* dengan *labeling*

Hasil pengujian di atas menunjukkan bahwa pada pengujian waktu proses, *segmentation* dengan *labeling* memerlukan waktu proses lebih lama daripada *segmentation* tanpa *labeling*, yakni sebesar 16,74%. Sedangkan pada pengujian jumlah pemisahan huruf, *segmentation labeling* melakukan pemisahan huruf lebih banyak daripada *segmentation* tanpa *labeling*, yakni sebesar 0,55%.

Pembahasan

Hasil pengujian memperlihatkan bahwa program ini mampu mengenali huruf satu per satu dengan tingkat akurasi yang cukup tinggi, yakni sebesar 96,67%. Hal ini berarti bahwa sistem *fuzzy* yang digunakan berjalan dengan baik. Salah satu keunggulan dari sistem *fuzzy* dalam pengenalan teks adalah pada tahapan *feature extraction.* Tahapan ini mampu membedakan huruf dari ciri-ciri yang dimilikinya. Pada pengujian dokumen standar juga memberikan hasil yang baik, yaitu dengan tingkat akurasi sebesar 96,13%. Hasil pengujian pada dokumen tersebut sangat tergantung pada berbagai faktor, antara lain: sistem *fuzzy* yang digunakan, pembedaan huruf besar dan kecil, serta huruf-huruf yang terhubung. Beberapa huruf yang telah tidak terbaca oleh sistem *fuzzy* sangat berpengaruh terhadap pembacaan keseluruhan dokumen. Hal ini disebabkan karena beberapa dokumen memiliki huruf-huruf tertentu yang sering digunakan, sedangkan huruf-huruf lain jarang digunakan.

Salah satu kelemahan dari program ini adalah pada pembacaan huruf-huruf yang terhubung satu sama lain. Hal ini disebabkan karena sistem pengenalan teks akan membaca huruf yang terhubung tersebut sebagai satu kesatuan (satu buah huruf). Hal ini akan mempengaruhi *input* dari sistem *fuzzy*. Proses *segmentation* yang digunakan pada program ini belum mampu memisahkan huruf yang saling terhubung. Pada pengujian *labeling*, sistem memperlihatkan bahwa penggunaan *labeling* memerlukan waktu yang lebih lama dalam pengenalan teks, yakni dengan perbedaan waktu proses sebesar 16,74%. Sedangkan untuk pemisahan jumlah huruf, penggunaan *labeling* mampu memisahkan karakter lebih banyak daripada tanpa *labeling*, yakni dengan perbedaan 0,55%. Semakin besar ukuran *font* maka perbedaan waktu proses semakin besar, akan tetapi perbedaan jumlah pemisahan huruf menjadi semakin kecil. Sebaliknya, semakin kecil ukuran *font* maka perbedaan waktu proses menjadi semakin kecil, akan tetapi perbedaan jumlah pemisahan huruf semakin besar. Nilai perbedaan yang kecil pada pemisahan huruf disebabkan karena hampir semua karakter mampu dipisahkan dengan cara biasa. Sedangkan *labeling* hanya ditujukan untuk huruf-huruf yang menempel. Hal ini berarti bahwa dari berbagai kemungkinan yang ada pada sebuah dokumen, presentase huruf yang menempel sangatlah kecil.

5. KESIMPULAN

Dari keseluruhan perancangan program, dapat ditarik kesimpulan bahwa pendekatan *fuzzy logic* dapat digunakan dalam pengenalan teks dan hasil lain yang diperoleh bahwa :

1. Penentuan nilai *threshold* berpengaruh terhadap pemisahan huruf. Nilai *threshold* terbaik yang digunakan adalah 178.
2. Penentuan segmen "huruf besar atau kecil" dilakukan berdasarkan tinggi baris. Nilai batas yang digunakan adalah 0,65*tinggi baris.
3. Penentuan segmen "spasi+huruf" dilakukan berdasarkan tinggi baris. Nilai batas yang digunakan adalah 0,26*tinggi baris.
4. Pengelompokan pola huruf untuk *library* memiliki pengaruh terhadap sistem pengenalan teks. Pada program ini, pola huruf yang dengan *font* "9,11,16,26,48" lebih tepat digunakan untuk *picture library* "*All*", pola huruf yang *font* "14,20,26,36" lebih tepat digunakan untuk *picture library* "*Large*", dan pola huruf yang terdiri atas *font* "8,9,10,11,12 ,20,72" lebih tepat digunakan *picture library* "*Small*".

5. Penggunaan *labeling* berpengaruh terhadap sistem pengenalan teks. Pada pengenalan dokumen standar dengan *labeling*, sistem mampu melakukan pemisahan huruf 0,55% lebih banyak daripada tanpa *labeling*. Akan tetapi, proses *labeling* memakan waktu lebih lama, yakni sebesar 16,74%.
6. Sistem *fuzzy* berpengaruh terhadap tingkat akurasi pengenalan teks. Hasil pengaturan optimal pada sistem *fuzzy* diperoleh pada "*Box*" = 5 dengan jumlah "*Pixel Per Box"* sebanyak 8.

6. UCAPAN TERIMAKASIH

Dalam kesempatan ini penulis mengucapkan terima kasih kepada **Faisal** alumni jurusan Matematika FST Binus University yang telah membantu penulis dalam hal pembuatan program komputer pada percobaan penelitian ini sehingga dapat diselesaikan sesuai dengan rencana.

7. DAFTAR PUSTAKA

Fadlisyah, Taufiq, Zulfikar, Fauzan. (2008). *Pengolahan Citra Menggunakan Delphi*. Edisi pertama. Graha Ilmu, Yogyakarta.

Fernando, H. (2003). Handwriting Digit Recogntion With Fuzzy Logic. *Jurnal Teknik Elektro*. Vol 3(2), pp84-87.

Frucci, M., Baja GSD. (2008). From Segmentation to Binarization of Gray-level Images. *Journal of Pattern Recognition Research*. Vol 3(1), pp1-13.

Kosko, B. (2005). *Fuzzy Engineering*. International Edition. Prentice Hall, Inc., New Jersey.

Sulaiman, SN., Alias, MF., Isa, NAM., Rahman, MFA. (2007). An Expert Image Processing System on Template Matching. *IJCSNS International Journal of Computer Science and Network Security*. Vol 7(7), pp234-238.

Vaughan, T. (2004). *Multimedia: Making It Work*. Sixth Edition. McGraw-Hill Technology Education, New-York.

Yohanes TDS., Thiang, Suntono Chandra. (2002). Aplikasi Sistem Neuro-Fuzzy untuk Pengenalan Kata. *Jurnal Teknik Elektro*. Vol 2(2), pp73-77.

Zand, M., Nilchi, AN., Monadjemi, SA. (2008). Recognition-based Segmentation in Persian Character Recognition. *International Journal of Computer and Information Science and Engineering*. Vol 2(1), pp14-18.