

LANDASAN TEORI *THINNING*

oleh :

Rosa Ariani Sukanto / 23507024



PROGRAM STUDI INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2008

Hanya Boleh Digunakan untuk Referensi

BAB I

LANDASAN TEORI

1.1 Geometri

Geometri adalah bagian dari matematika yang menitik beratkan pada ukuran, bentuk, posisi relatif bangun, dan properti dari ruang. Geometri terkait dengan panjang, area, dan volume. Bagian geometri yang akan digunakan untuk pengenalan tulisan tangan ini adalah garis mendatar, garis vertikal, dan garis miring [3]. Ciri-ciri geometris yang akan digunakan untuk mengklasifikasikan karakter adalah sebagai berikut:

Fitur Geometri	Keterangan
	garis vertikal yang dilabeli dengan label angka 1
	garis mendatar yang dilabeli dengan label angka 2
	garis miring ke kiri yang dilabeli dengan label angka 3
	garis miring ke kanan yang dilabeli dengan label angka 4

1.2 Citra

Citra atau gambar atau *image* merupakan sesuatu yang menggambarkan objek dan biasanya dua dimensi [4]. Citra merupakan suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra dapat dikelompokkan menjadi dua yaitu citra tampak dimana objek yang ada di dalamnya direpresentasikan dengan jelas dan citra tak tampak dimana citra direpresentasikan dalam sebuah fungsi yang biasanya merupakan fungsi matematika. Citra digital merupakan citra yang telah disimpan dalam bentuk *file* sehingga dapat diolah dengan menggunakan komputer.

1.2.1 Gambar Biner

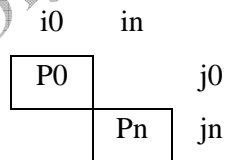
Gambar biner (*binary image*) adalah gambar dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas, biasanya 0 dan 1 dimana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta/objek (*foreground*) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih [5].

Background merupakan kumpulan komponen yang terkoneksi yang bernilai 0 pada gambar biner [6] sedangkan *foreground* merupakan kumpulan komponen yang terkoneksi atau tidak dan bernilai 1 pada gambar biner [6].

1.2.2 Piksel

Pada citra digital sebuah piksel adalah bagian terkecil informasi dari sebuah citra [4]. Piksel biasanya direpresentasikan dengan titik-titik yang membentuk sebuah citra.

Piksel-piksel yang terkoneksi (*connected pixels*) atau piksel-piksel ketetanggaan dapat diilustrasikan sebagai berikut, sebuah piksel P_0 pada koordinat i_0, j_0 dikatakan terkoneksi dengan piksel P_n pada koordinat i_n, j_n jika dan hanya jika ada jalur antara P_0 dan P_n yang merupakan urutan titik [6], seperti pada gambar berikut:



Ada beberapa jenis piksel ketetanggaan, antara lain 4-ketetanggaan, 6-ketetanggaan, dan 8-ketetanggaan, namun yang banyak digunakan adalah 4-ketetanggaan dan 8-ketetanggaan. Dalam tesis ini akan digunakan 8-ketetanggaan. Kumpulan piksel yang dikatakan sebagai 8-ketetanggaan (*8-connected*) adalah ketika sebuah piksel pada koordinat i, j memiliki 8 piksel tetangga yaitu sebagai berikut:

tetangga	tetangga	tetangga
$i-1, j-1$	$i, j-1$	$i+1, j-1$
tetangga	i, j	tetangga

$i-1, j$		$i+1, j$
tetangga	tetangga	tetangga
$i-1, j+1$	$i, j+1$	$i+1, j+1$

Relasi ketetanggaan dari sebuah piksel yang telah didefinisikan dapat dijadikan representasi geometri secara diskrit. [6]

1.3 Preprocessing

Apa itu preprocessing? Apa tujuannya, dalam tesis ini apa saja yang dilakukan?

Preprocessing adalah proses yang memproses sebuah masukan menjadi sebuah keluaran yang akan menjadi masukan bagi proses lain atau proses selanjutnya [4]. Pada tesis ini proses *preprocessing* meliputi proses-proses sebagai berikut:

- *Grey Scalling / Thresholding*

Proses *Grey Scalling* mengubah gambar berwarna menjadi hitam putih dengan mengubah warna setiap komponen RGB gambar menjadi bernilai sama. Proses *Thresholding* mengubah gambar menjadi gambar biner (*binary image*) dimana ditentukan sebuah nilai level *threshold* kemudian piksel yang memiliki nilai di bawah level *threshold* diset menjadi nilai warna putih (0 pada nilai biner) dan nilai di atas level *threshold* diset menjadi nilai warna hitam (1 pada nilai biner). Proses *threshold* digunakan untuk mengekstrak *foreground* (tinta) dari *background* (kertas) dan menjadikan gambar menjadi biner [1].

- *Smoothing*

untuk meminimalisir *noise*. Proses *smoothing* menggunakan metode *stentiford boundary* untuk menghilangkan detail dan *noise* dengan mengecek konektivitas dengan piksel yang bernilai 1 dan menghitung tetangga piksel yang bernilai 0 jika piksel yang sedang dicek memenuhi syarat tidak boleh dihapus maka piksel akan ditandai sebagai piksel yang tidak boleh dihapus [2].

- *Scaling*

menskala gambar karakter tulisan tangan menjadi lebih kecil agar proses lainnya lebih cepat dan tidak terpengaruh besar gambar karakter (pada bagian inti karakter).

- *Stroke Thinning*

untuk mengecilkan garis (stroke) tulisan tangan proses ini menggunakan metode hybrid dimana menggunakan campuran tiga buah metode yang saling mendukung untuk proses thinning yaitu algoritma thinning Zhang-Suen yang mengecek ketetanggaan dan konektivitas 8-arah piksel, Stentiford Acute Angle Emphasis untuk menandai piksel pada bagian luar yang tidak boleh dihapus, dan algoritma holt untuk menghapus hasil thinning yang tidak diperlukan (percabangan yang tidak perlu) [2].

Proses *preprocessing* pada tesis ini menggunakan piksel ketetanggaan tidak dapat dipisahkan dari apa yang disebut *Neighborhood Operation*. *Neighbourhood operation* adalah suatu cara dimana mengkombinasikan area kecil piksel atau *neighborhood* untuk menggenerasi piksel keluaran [7]. Inti dari operasi ini adalah *convolution mask* atau *kernel* dengan menggunakan persamaan *convolution*, ditunjukkan dengan M pada gambar berikut, dimana memiliki label elemen $m_{i,j}$. *Kernel* berupa sebuah *array* angka yang memiliki dimensi vertikal dan horizontal (m x n)

Persamaan *convolution*

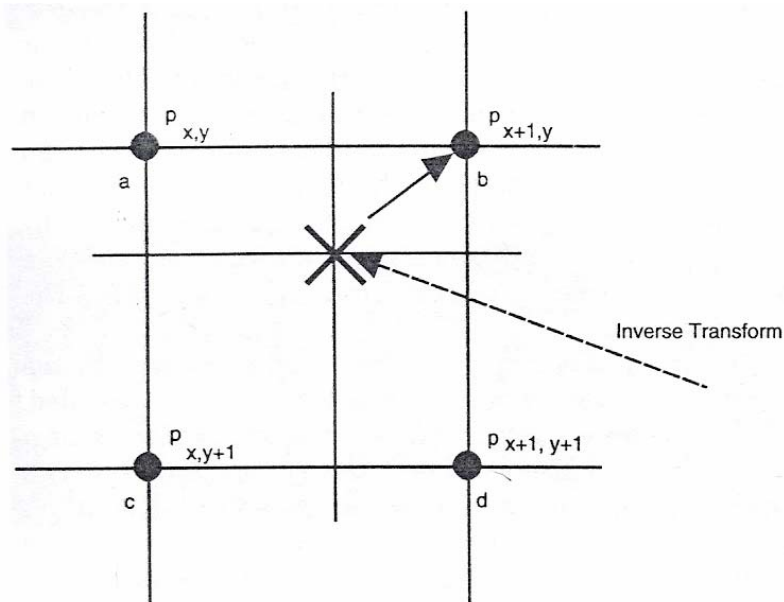
$$c_{x,y} = \frac{\sum_{i=1}^m \sum_{j=1}^n p_{i,j} * m_{i,j}}{\sum_{i=1}^m \sum_{j=1}^n m_{i,j}}$$

1.3.1 Transformasi Geometri

Geometric Transformation menggunakan *sampling* sebagai dasar. *Geometric Transformation* meliputi rotasi, *scaling*, dan translasi [2].

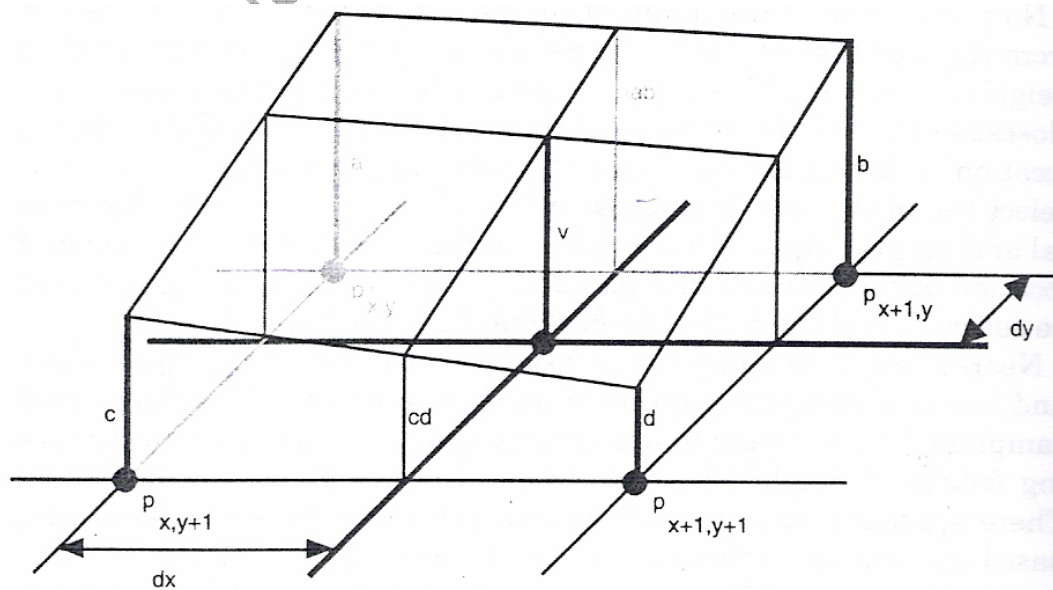
1.3.1.1 Nearest Neighbor Sampling

Menghitung jarak transformasi ke empat tetangga dan diambil piksel dengan jarak terdekat. Seperti pada gambar berikut:



1.3.1.2 Bilinier interpolation

Jika dx adalah jarak sub-piksel pada gambar di atas dari piksel $P_{x,y}$ ke lokasi yang dituju pada dimensi horizontal, jarak proporsional pada titik a dan b (c dan d juga) maka intensitas interpolasi disebut sebagai ab dan cd , sedangkan dy adalah jarak sub-piksel dari piksel $P_{x,y}$ ke lokasi yang dituju pada dimensi vertikal. Sehingga bilinier interpolasinya dapat dihitung dari ab dan cd dan dapat menjadi hasil piksel output.



1.3.1.3 Rotasi

Rotasi memiliki dua properti operasi yang harus didefinisikan yaitu arah perputaran dimana nilai positif berarti berlawanan arah dengan jarum jam dan nilai negatif untuk arah yang searah jarum jam, titik tengah perputaran. Untuk perputaran pada titik *origin* (0,0) maka persamaan transformasinya adalah sebagai berikut:

$$x_s = [x_d \times \cos(a)] - [y_d \times \sin(a)]$$

dan

$$y_s = [x_d \times \sin(a)] + [y_d \times \cos(a)]$$

dimana

a : sudut perputaran

xd : koordinat x dari titik awal

yd : koordinat y titik awal

xs : koordinat x titik akhir

ys : koordinat y titik akhir

Untuk perputaran pada titik tengah lingkaran, maka persamaannya menjadi:

$$x_s = [x_d \times \cos(a)] - [y_d \times \sin(a)] + [x_c \times (1 - \cos(a))] + [y_c \times \sin(a)]$$

$$y_s = [x_d \times \sin(a)] + [y_d \times \cos(a)] - [x_c \times \sin(a)] + [y_c \times (1 - \cos(a))]$$

dimana

xc : koordinat x dari titik tengah lingkaran

yc : koordinat y dari titik tengah lingkaran

1.3.1.4 Scaling

Scaling adalah fungsi yang mengubah ukuran suatu gambar dimana *scaling* biasanya cenderung merupakan sebutan untuk perbesaran dan *shrink* cenderung merupakan sebutan untuk memperkecil ukuran gambar. Persamaan *inverse transform* untuk *scaling* dengan acuan titik *origin* (0,0) adalah sebagai berikut:

$$x_s = x_d / \text{scale } x$$

$$y_s = y_d / \text{scale } y$$

dimana

x_d : koordinat x dari titik awal

y_d : koordinat y titik awal

x_s : koordinat x titik akhir

y_s : koordinat y titik akhir

scale x : faktor skala x

scale y : faktor skala y

Faktor skala jika lebih besar dari 1.0 maka gambar akan diperbesar sedangkan jika faktor skala lebih kecil dari 1.0 maka gambar akan diperkecil.

Persamaan *scaling* dari titik tengah gambar adalah sebagai berikut:

$$x_s = x_d / \text{scale } x - [(x_c / \text{scale } x) - x_c]$$

$$y_s = y_d / \text{scale } y - [(y_c / \text{scale } y) - y_c]$$

dimana

x_c : koordinat x dari titik tengah lingkaran

y_c : koordinat y dari titik tengah lingkaran

1.3.2 Thinning

Thinning adalah operasi morfologi yang digunakan untuk menghapus piksel *foreground* yang terpilih dari gambar biner, biasanya digunakan untuk proses mencari tulang dari sebuah objek (*skeletonization*) [5].

1.3.2.1 Konsep dasar dari Grey-level segmentation

Grey-level segmentation atau *thresholding* adalah konversi antara gambar *grey-level* dan gambar bilevel (hitam putih). *Thresholding* biasa digunakan sebelum dilakukan proses *thinning*. Hal yang harus dilakukan untuk proses *thresholding* adalah memilih sebuah nilai *threshold* dimana piksel yang bernilai dibawah nilai *threshold* akan diset menjadi 0 dan piksel yang bernilai di atas nilai *threshold* diset menjadi 1 [2].

Iterative Selection* untuk mencari nilai *threshold

Iterative selection (Ridler 1978) adalah proses dimana dibuat sebuah nilai *threshold* awal sebagai inisialisasi dan terus diperbaiki seiring dengan proses penelusuran gambar. Nilai *threshold* awal adalah nilai rata-rata *grey-level* (dianggap sebagai T_b), rata-rata dari nilai piksel yang lebih besar atau sama dengan T_b dianggap sebagai T_o maka nilai perkiraan untuk nilai *threshold* adalah $(T_b + T_o) / 2$. Hal ini terus diulang sampai tidak ada perubahan nilai [2].

1.3.2.2 Jumlah konektivitas

Jumlah konektivitas adalah perkiraan dari beberapa objek piksel memiliki kemungkinan terkoneksi. Perkiraan konektivitas itu adalah sebagai berikut (Yokoi 1973):

<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	0	1	1	1	1	1	1	1	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table>	1	0	1	1	1	1	1	0	1	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> </table>	1	0	1	0	1	0	0	1	0	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table>	1	0	1	0	1	0	1	0	1	<table border="1"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	1	0	0	0	0
1	0	1																																															
1	1	1																																															
1	1	1																																															
1	0	1																																															
1	1	1																																															
1	0	1																																															
1	0	1																																															
0	1	0																																															
0	1	0																																															
1	0	1																																															
0	1	0																																															
1	0	1																																															
0	0	0																																															
0	1	0																																															
0	0	0																																															
(a)	(b)	(c)	(d)	(e)																																													
piksel yang berada di tengah tidak	jika piksel yang berada di tengah dihapus maka	jumlah konektivitasnya adalah 3	jumlah konektivitasnya adalah 4,	jumlah konektivitasnya adalah 0																																													

terkoneksi dengan region manapun, maka bisa dihapus, jumlah konektivitasnya adalah 1	dua buah piksel yang dihubungkannya akan menjadi tidak terhubung, jumlah konektivitasnya adalah 2		merupakan nilai maksimum jumlah konektivitas	
--	---	--	--	--

dan persamaannya adalah sebagai berikut:

$$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2})$$

dimana N_k adalah warna piksel dari salah satu dari 8 piksel ketetanggaan yang terlibat dan $S = \{1, 3, 5, 7\}$. N_1 adalah nilai warna piksel sebelah kanan piksel tengah dan terus diteruskan dengan piksel selanjutnya searah jarum jam dari piksel tengah [2].

1.3.2.3 Permasalahan Thinning yang kerap muncul

Necking

Dimana bagian tipis yang berada di perpotongan dari dua garis ditipiskan menjadi sebuah segmen garis.

Tailing

Dimana terdapat garis ekstra pada sebuah segmen

Spurious Projection (line fuzz)

Dimana ada beberapa garis ekstra

Stentiford mengusulkan sebuah proses untuk mengatasi permasalahan-permasalahan ini. *Spurious projection* dapat dicegah dengan melakukan proses *smoothing* (jika memiliki ketetanggaan dua atau kurang dari dua, dan memiliki jumlah konektivitas kurang dari dua maka piksel akan dihapus). *Necking* dapat diatasi dengan menggunakan proses *acute angle emphasis* dimana piksel yang dekat dengan

pertemuan dua garis akan diset menjadi 0 jika merupakan awal dari sebuah sudut yang tajam. Jika piksel yang di tengah sesuai dengan pola-pola berikut, maka tandai piksel sebagai piksel yang dihapus.

<table border="1"> <tbody> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> </tbody> </table> <p>D1</p>	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1		<table border="1"> <tbody> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> </tbody> </table> <p>D2</p>	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1		<table border="1"> <tbody> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> </tbody> </table> <p>D3</p>	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1		<table border="1"> <tbody> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> </tbody> </table> <p>D4</p>	1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1		<table border="1"> <tbody> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> </tbody> </table> <p>D5</p>	1	1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1		1	1	1	
1	1	0	1	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	0	0	1	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	1	0	0	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	0	0	1	1																																																																																																																													
1	0	0	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	1	0	0	1																																																																																																																													
1	1	0	0	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
	1	1	1																																																																																																																														
<table border="1"> <tbody> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </tbody> </table> <p>U1</p>		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	<table border="1"> <tbody> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </tbody> </table> <p>U2</p>		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	<table border="1"> <tbody> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> </tbody> </table> <p>U3</p>		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	<table border="1"> <tbody> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </tbody> </table> <p>U4</p>		1	1	1		1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1	<table border="1"> <tbody> <tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> </tbody> </table> <p>U5</p>		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1
	1	1	1																																																																																																																														
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	1	0	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	0	0	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	1	0	0	1																																																																																																																													
	1	1	1																																																																																																																														
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	0	0	1	1																																																																																																																													
1	0	0	1	1																																																																																																																													
	1	1	1																																																																																																																														
1	1	1	1	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	1	0	0	1																																																																																																																													
1	1	0	0	1																																																																																																																													

Semua usulan stentiford merupakan suatu usaha untuk memperbaiki semua permasalahan yang muncul pada proses *thinning* namun usulan ini juga belum bisa berlaku untuk semua kasus bentuk karakter.

Salah satu algoritma *thinning* yang muncul adalah algoritma Zhang Suen (Zhang 1984). Algoritma ini menggunakan metode paralel yang berarti bahwa nilai baru untuk sebuah piksel dapat dihitung dengan menggunakan nilai piksel dari iterasi sebelumnya. Algoritma ini dibagi menjadi dua buah sub-iterasi dimana jika piksel telah ditandai menjadi piksel yang akan dihapus maka jika kondisi berikut dipenuhi:

1. jumlah konektivitasnya adalah 1.
2. mempunyai paling sedikit dua ketetangga dan tidak lebih dari enam
3. paling sedikit salah satu dari $I(i, j+1)$, $I(i-1, j)$, dan $I(i, j-1)$ bernilai 0 (warna latar belakang)

4. paling sedikit salah satu dari $I(i-1, j)$, $I(i+1, j)$, dan $I(i, j-1)$ bernilai 0

Pada akhir sub-iterasi ini maka piksel yang ditandai menjadi piksel yang akan dihapus, benar-benar dihapus. Pada sub-iterasi berikutnya dilakukan hal yang sama kecuali langkah 3 dan 4. Jika pada akhir iterasi sudah tidak ada lagi piksel yang akan dihapus, maka proses selesai.

Proses *thinning* Zhang Suen masih memiliki beberapa kelemahan yaitu masalah *necking* dan *spurious projection* untuk beberapa kasus, oleh karena itu Holt (Holt 1987) mengusulkan untuk menjadikan dua buah sub-iterasi ditulis sebagai ekspresi logika dengan menggunakan 3 x 3 ketetanggaan. Sub-iterasi yang pertama dapat ditulis sebagai berikut:

$$v(C) \wedge (\sim \text{edge}(C) \vee (v(E) \wedge v(S) \wedge (v(N) \vee v(W))))$$

dimana

C: center (piksel $I(i,j)$)

E: East (piksel $I(i+1, j)$)

S: South (piksel $I(i, j+1)$)

N: North (piksel $I(i, j-1)$)

W: West (piksel $I(i-1, j)$)

v: fungsi yang menyatakan nilai logika piksel (true untuk piksel bernilai 1 dan false untuk piksel bernilai 0 (warna latar belakang))

edge: bernilai true jika C berada di pinggir objek (mempunyai ketetanggaan antara 2 sampai 6 ketetanggaan dan jumlah konektivitasnya adalah 1)

Sub-iterasi kedua dapat ditulis sebagai berikut:

$$v(C) \wedge (\sim \text{edge}(C) \vee (v(W) \wedge v(N) \wedge (v(S) \vee v(E))))$$

Holt menggabungkan kedua persamaan dengan maksud agar operasi dapat berjalan paralel, berikut adalah persamaan untuk piksel yang tidak perlu dihapus:

$$v(C) \wedge (\sim \text{edge}(C) \vee$$

$$\begin{aligned}
 &(\text{edge}(E) \wedge v(N) \wedge v(S)) \vee \\
 &(\text{edge}(S) \wedge v(W) \wedge v(E)) \vee \\
 &(\text{edge}(E) \wedge \text{edge}(SE) \wedge \text{edge}(S))
 \end{aligned}$$

Holt juga mengusulkan untuk adanya proses *staircase removal* karena pada kenyataannya di beberapa kasus setelah proses *thinning* dilakukan, masih ada beberapa piksel yang perlu dihapus. Piksel-piksel yang harus dihapus untuk proses *staircase removal* adalah yang memenuhi pola-pola berikut:

0	1	x	x	1	0	0	x	x	x	x	0
1	1	x	x	1	1	x	1	1	1	1	x
x	x	0	0	x	x	x	1	0	0	1	x

Untuk mencegah adanya lubang baru maka dibuat sebuah aturan dimana satu dari yang bernilai x harus bernilai 0. Untuk iterasi *staircase removal* yang memiliki kecondongan arah ke utara maka persamaan untuk piksel yang tidak perlu dihapus dalam iterasi *staircase removal* adalah sebagai berikut:

$$\begin{aligned}
 &v(C) \wedge \sim(v(N) \wedge \\
 &((v(E) \wedge \sim v(NE) \wedge \sim v(SW) \wedge (\sim v(W) \vee \sim v(S)) \vee \\
 & (v(W) \wedge \sim v(NW) \wedge \sim v(SE) \wedge (\sim v(E) \vee \sim v(S))))))
 \end{aligned}$$

Untuk persamaan yang memiliki kecondongan arah ke selatan maka persamaan untuk piksel yang tidak perlu dihapus dalam iterasi *staircase removal* sama dengan persamaan di atas dengan penggantian N dengan S.

Algoritma *prunning* adalah sebuah teknik yang digunakan pada pemrosesan gambar digital dengan menggunakan morfologi matematika. Biasa digunakan sebagai

komplemen dari algoritma *thinning* untuk menghapus piksel-piksel yang tidak diinginkan [4].

1.4 Segmentasi

Segmentasi merupakan proses membagi citra digital menjadi beberapa bagian [4]. Tujuan dari proses segmentasi adalah untuk mempermudah atau menyederhanakan representasi dari citra menjadi sesuatu yang lebih berarti dan mudah dianalisa. Metode yang digunakan dalam tesis ini adalah dengan menggunakan histogram berbasis frekuensi warna yang termasuk *background* atau *foreground* secara vertikal maupun secara horizontal.

1.4.1 Histogram

Histogram merupakan grafik tabulasi frekuensi dari sesuatu yang dibuat histogramnya [4]. Histogram menampilkan proporsi dari setiap kategori yang ada dalam histogram yang dibuat.

Line Segmentation

melakukan segmentasi per baris teks dengan menggunakan histogram horizontal dimana local minima dianggap sebagai garis pembatas antar baris teks.

Word Segmentation

melakukan segmentasi per kata pada baris teks menggunakan histogram vertikal dimana local minima dianggap sebagai pemisah antar kata. Untuk membedakan dengan histogram vertikal yang digunakan untuk memecah karakter maka dilakukan pembelajaran rasio rata-rata antara spasi antar kata dengan main body agar tidak keliru dengan spasi antar karakter (pada tulisan tangan tegak (discrete)).

Character Segmentation

melakukan segmentasi karakter per kata dengan menggunakan histogram vertikal yang dimodifikasi untuk setiap kata.

1.5 Feature Extraction

Feature Extraction adalah proses transformasi data masukan menjadi kumpulan fitur untuk mengambil informasi yang relevan dari data masukan dengan tujuan untuk mengambil representasi minimal dari data masukan [4]. Fitur gambar menyediakan cara untuk mendeskripsikan properti gambar [8]. Fitur geometris merupakan kunci untuk mendeskripsikan struktur gambar [8].

- ukuran rata-rata tinggi dan lebar karakter

ukuran diambil dari perata-rataan setiap karakter yang dimasukkan sebagai pembelajaran

- ascenders dan descenders

merupakan pembagian tulisan menjadi tiga buah area yaitu bagian atas (ascenders), bagian tengah (main body), bagian bawah (descenders). Kemudian setiap area diambil fiturnya dengan menggunakan histogram untuk membedakan karakter yang akan dikenali.

- permodelan stroke

permodelan stroke menggunakan rangkaian stroke (garis tulisan) untuk mengenali karakter. Rangkaian stroke merupakan kumpulan titik-titik yang diberi label angka berdasarkan arah titik tetangga berikutnya yang disimpan di dalam list yang kemudian dicek polanya. Label yang diberikan adalah sebagai berikut:

- angka 1 untuk arah ke atas atau kebawah (garis vertikal)
- angka 2 untuk arah ke samping (garis horizontal)
- angka 3 untuk arah ke kanan atas atau bawah (garis miring hadap kanan)
- angka 4 untuk arah ke kiri bawah atau atas (garis miring hadap kiri).

Rangkaian label dapat lebih dari satu untuk menggambarkan fitur geometrinya.

Hanya Boleh Digunakan untuk Referensi

DAFTAR REFERENSI

- [1] Guillevic, Didier (1995). Unconstrained Handwriting Recognition Applied to The Processing of Bank Cheques. <http://www.cenparmi.concordia.ca/~didier/thesis/thesisDidier.ps.gz>.
- [2] Parker, J.R. (1997). Algorithms for Image Processing and Computer Vision. Wiley Computer Publishing : Canada.
- [3] Wu, Hung-wu (2007). On Handwriting Recognition with Elementary Geometric and Algorithmic Methods. http://www.artificialintelligenceage.com/handwriting_recognition.pdf.
- [4] <http://en.wikipedia.org>
- [5] Fisher, R., S. Perkins, A. Wilker, dan E. Wolfart (2003). <http://homepages.inf.ed.ac.uk/rbf/HIPR2/>.
- [6] Acharya , Tinku dan Ajoy K. Ray (2005). Image Processing: Principles and Applications. John Wiley and Sons, Inc. New Jersey.
- [7] Burdick, Howard E. (2000). Digital Imaging: Theory and Applications. McGraw-Hill-Singapore
- [8] Ritter, Gerhard X. dan Joseph N. Wilson (1996). Handbook of Computer Vision Algorithms in Image Algebra. CRC Press.

DAFTAR PUSTAKA

- [1] Arica, Nafiz dan Fatos T. Yarman-Vural (2001). An Overview of Character Recognition Focused on Off-line Handwriting. http://www.ceng.metu.edu.tr/~nafiz/papers/SMC_2001.pdf.
- [2] Drissman, Avi (1997). Handwriting Recognition System: An Overview. <http://www.drissman.com/avi/school/HandwritingRecognition.pdf>.
- [3] Fisher, R. dkk (2003). <http://homepages.inf.ed.ac.uk/rbf/HIPR2/>.
- [4] Koerich, A. L. dkk (2003). Large Vocabulary Off-line Handwriting Recognition : A Survey. http://www.ppgia.pucpr.br/~alekoe/Papers/Koerich_PAA_2003.pdf.
- [5] Rao, P. Shankar dan J. Aditya (2000). Handwriting Recognition - "offline" Approach. <http://stanford.edu/~adityaj/handwriting.pdf>.
- [6] Timar, Gergely dkk (2002). Analogic Preprocessing and Segmentation Algorithms for Off-line Handwriting Recognition. http://lab.analogic.sztaki.hu/publications/rcs/CSC_2002_HWCharRec.pdf.
- [7] Toselli, Alejandro dkk (2004). Spontaneous Handwriting Recognition and Classification. http://www.dsic.upv.es/~ajuan/research/2004/Juan04_08c.pdf.
- [8] Yanikoglu, Berrin A. dan Peter A. Sandon (1993). Off-line Cursive Handwriting Recognition Using Style Parameters. <http://www.cs.dartmouth.edu/reports/TR93-192.pdf>.