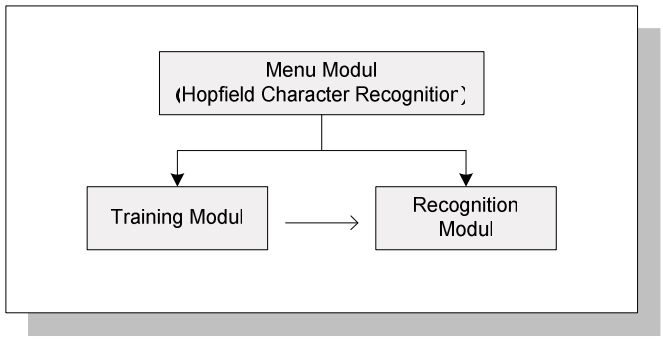
BAB IV
PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini akan dijelaskan mengenai implementasi, hasil pengujian perangkat lunak termasuk di dalamnya terdapat proses pelatihan, proses pengenalan, dilanjutkan dengan tahap analisis untuk menganalisis hasil pengujian.

4.1 Implementasi

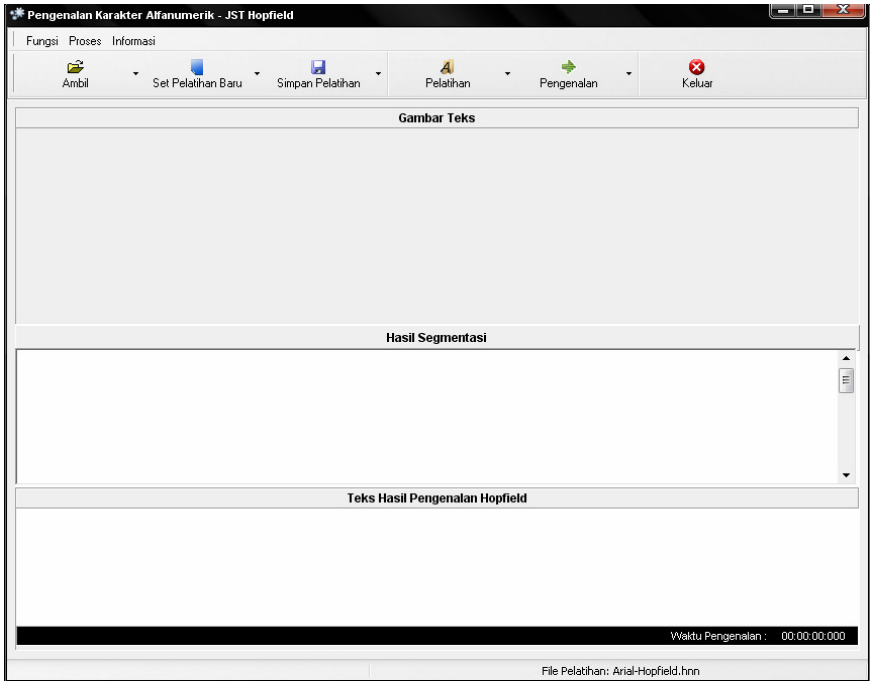
Tujuan dari tahap implementasi ini adalah untuk mengimplementasikan hasil analis dan pemodelan sistem pada bab III, sehingga didapatkan sebuah perangkat lunak untuk melakukan pengenalan karakter dari setiap kata dalam paragraf. Perangkat lunak ini diimplementasikan ke dalam tiga bagian, yaitu bagian menu utama (Modul Menu), bagian proses pelatihan (*Training Module*) dan bagian proses pengenalan (*Recognition Module*). Berikut diagram dari ketiga modul tersebut:



Gambar 4.1 Modul Perangkat Lunak

4.1.1 Modul Menu

Modul ini berisi *menu* utama yang disediakan agar pengguna dapat memilih apakah ingin menjalankan proses pelatihan atau proses pengenalan, dalam modul ini juga terdapat proses untuk mengubah-ubah parameter pengenalan. Sebelum proses pengenalan, harus dilakukan proses pelatihan terlebih dahulu.



Gambar 4.2 Antar Muka Menu Utama

4.1.2 Modul Pelatihan (*Training)*

Dalam modul pelatihan terdapat tiga proses utama yaitu proses *input* pola (*input pattern)*, set parameter pelatihan dan proses pelatihan jaringan. Di mana pada proses *input pattern* digunakan untuk menyimpan pola-pola dan nilai vektor dari setiap karakter yang akan digunakan sebagai masukkan dari proses pelatihan jaringan. Sedemikian sehingga didapatkan arsitektur jaringan yang memiliki parameter seperti nilai bobot. Hasil proses pelatihan berupa JST yang sudah terlatih, disimpan terlebih dahulu untuk selanjutnya digunakan dalam modul pengenalan.



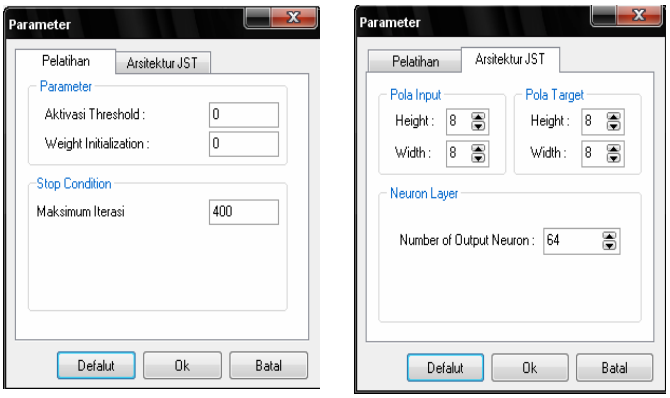
Gambar 4.3 Antar Muka Modul Pelatihan

4.1.3 Modul Pengenalan (*Recognition*)

Pada modul pengenalan yang bisa dikatakan sebagai modul utama, karena di sini akan kita lakukan pengujian terhadap hasil runtutan proses pengujian terhadap arsitektur JST yang didapatkan dari proses pelatihan sebelumnya yang disebut sebagai proses pengenalan (*recognition*) di mana akan dihasilkan output karakter dari citra masukan dari pengguna. Pada program ini modul pengenalan menyatu dengan modul menu utama.

4.1.4 Modul Parameter Untuk Proses Pelatihan (*Training)*

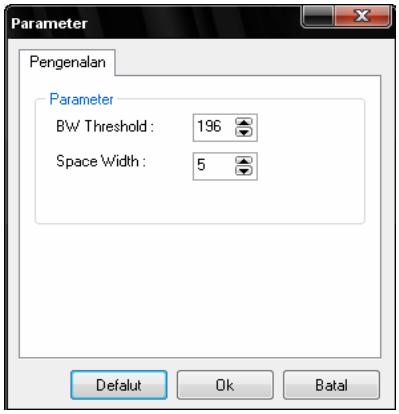
Pada modul ini berisi parameter-parameter yang digunakan pada proses pelatihan JST *Hopfield* dan parameter untuk arsitektur JST *Hopfield.* Pada parameter pelatihan digunakan untuk mengubah-ubah nilai dari *Aktivasi Threshold* (batas ambang pada saat proses aktivasi), *Weight Initialization* (Inisialisasi matriks bobot awal), Maximum Iterasi (batas maksimum iterasi tiap pola pelatihan). Sedangkan untuk parameter arsitektur JST digunakan untuk merubah karakteristik pola *input* dan *output* serta jumlah *neuron* yang dipakai.



Gambar 4.4 Parameter Dalam Proses Pelatihan

4.1.5 Modul Parameter Untuk Proses Pengenalan

Pada modul ini digunakan untuk mengubah parameter hanya pada saat proses pengenalan saja. Pengubahan parameter ini dilakukan untuk memperoleh hasil pengenalan yang lebih baik apabila karakter yang akan dikenali merupakan karakter berwarna.



Gambar 4.5 Parameter Untuk Proses Pengenalan

Parameter *BW Threshold* digunakan sebagai nilai batas untuk memisahkan antara *background* dengan karakter yang akan dikenali. Pengubahan parameter ini sangat berpengaruh apabila karakter yang akan dikenali berupa teks berwarna ataupun gambar yang kurang jelas menurut sistem. *Space Width* digunakan untuk mengubah lebar jarak antar kata.

4.1.6 Spesifikasi Perangkat Lunak Pada Tahap Implementasi

Sistem ini diimplementasikan dengan menggunakan perangkat lunak pendukung sebagai berikut :

1. Sistem Operasi Microsoft Windows XP Professional
2. Borland Delphi 7
3. Paint
4. Adobe Photosop 7

4.2 Kriteria Pengujian dan Analisis Pengujian

Dalam sebuah pengujian diperlukan suatu batasan tentang kriteria pengujiannya itu sendiri maupun analisis yang akan dilakukan, berikut beberapa kriteria yang dipakai:

1. Kriteria dalam pengujian sistem :
    - Jenis huruf yang digunakan untuk *input* pelatihan dan pengenalan adalah *Arial.* Untuk pelatihan menggunakan *font Arial* 16pt.
    - Ukuran *font* yang digunakan untuk pengujian yaitu 14pt, 16pt dan 18pt normal.
    - Citra *input* berupa *file* *.*bmp* yang berisi rangkaian karakter yang berasal dari *print screen file* Microsoft Word.

- Input rangkaian karakter yang digunakan tanpa tanda baca atau simbol, seperti titik (.) , koma (,)….dan seterusnya.

2. Kriteria dalam analisis sistem :

   Analisis akan dilakukan terhadap *output* pengenalan dengan menghitung seberapa besar persentase *output* karakter pengenalan yang sesuai (*valid)* dengan *input* yang diberikan dan menganalisis hal-hal yang mempengaruhi kinerja sistem. *Valid* dalam arti karakter hasil pengenalan cocok atau sama dengan karakter yang diinputkan berdasarkan perspektif pengamatan mata manusia. Serta akan dijelaskan perhitungan-perhitungan matematisnya berdasarkan ilustrasi yang diberikan.

4.3 Pengujian dan Analisis Proses

4.3.1 Pengujian dan Analisis Proses Pelatihan (*Training)*

Pada proses ini sistem melakukan proses pelatihan di mana huruf dan angka dilatihkan tiap karakter sesuai dengan nilai *input* vektornya. Jenis karakter yang dilatihkan adalah *Arial* 16pt. Berikut nilai-nilai parameter yang digunakan dalam proses pelatihan :

Weight Initialization : $W_{ji} = 0$

Maksimum Iterasi tiap pola : 400

Sedangkan parameter yang digunakan sebagai arsitektur JST adalah sebagai berikut :

Pola *Input* : Tinggi = 8, Lebar = 8

Pola *Output* : Tinggi = 8, Lebar = 8

Jumlah Neuron : 64

Berikut tabel hasil pelatihan yang telah dilakukan oleh sistem :

Tabel 4.1 Proses Pelatihan

Karakter	Iterasi tiap pola pelatihan	Jumlah iterasi saat pola ke-n	Nilai Energi
0	400	400	-2.130
1	400	800	-0.257
2	400	1200	-2.708
3	400	1600	-2.823
4	400	2000	-4.396
5	400	2400	-0.601
6	400	2800	-1.613
7	400	3200	-0.936
8	400	3600	-0.329
9	400	4000	-2.231
a	400	4400	-2.384
b	400	4800	-1.504
c	400	5200	-2.547

Karakter	Iterasi tiap pola pelatihan	Jumlah iterasi saat pola ke-n	Nilai Energi
d	400	5600	-0.355
e	400	6000	-1.269
f	400	6400	-5.096
g	400	6800	-0.937
h	400	7200	-1.147
i	400	7600	-0.389
j	400	8000	-0.540
k	400	8400	-2.640
l	400	8800	-0.294
m	400	9200	-4.464
n	400	9600	-7.463
o	400	10000	-4.337
p	400	10400	-4.039
q	400	10800	-2.761
r	400	11200	-1.517
s	400	11600	-1.217
t	400	12000	-1.191
u	400	12400	-1.554
v	400	12800	-1.102
w	400	13200	-3.291
x	400	13600	-0.087
y	400	14000	-1.479
z	400	14400	-2.640
A	400	14800	-3.478
B	400	15200	-0.310
C	400	15600	-3.391
D	400	16000	-0.753
E	400	16400	-3.142
F	400	16800	-0.262
G	400	17200	-1.213
H	400	17600	-3.903
I	400	18000	-0.294
J	400	18400	-2.356
K	400	18800	-3.767
L	400	19200	-2.887
M	400	19600	-0.758
N	400	20000	-1.593
O	400	20400	-3.047
P	400	20800	-2.067
Q	400	21200	-1.087
R	400	21600	-3.815
S	400	22000	-0.575
T	400	22400	-2.153
U	400	22800	-4.491
V	400	23200	-1.772
W	400	23600	-0.469
X	400	24000	-5.419
Y	400	24400	-4.753
Z	400	24800	-0.637

Proses yang terjadi pada pelatihan adalah citra masukan tersebut diubah ke dalam citra biner dengan menggunakan metode *grayscaling* dan *thresholding*. Intensitas tiap-tiap *pixel* citra yang diubah ke dalam citra biner yang kemudian dirubah ke dalam bentuk *bipolar* 1 dan -1 dalam sebuah matrik vektor yang sudah disiapkan, kemudian vektor *input* di *encode* untuk menghasilkan sebuah input yang digunakan dalam JST. Pada proses pelatihan ini setiap pola akan melakukan 400 kali iterasi, dengan harapan selama proses iterasi tersebut jaringan bisa mencapai kondisi stabil/konvergen dalam arti jaringan bisa memanggil ulang sebuah pola tertentu. Nilai energi pada JST *Hopfield*

menandakan bahwa jaringan akan stabil pada nilai minimum sehingga nilai energi pada JST *Hopfield* akan selalu ≤ 0, yang didapat dari setiap pembaharuan *neuron*-nya. Setiap *neuron* akan di-*update* secara asinkron hingga pola dipanggil secara stabil.

4.3.2 Pengujian dan Analisis Sistem *Character Recognition*

Jaringan yang telah siap untuk dipakai dilakukan pengujian untuk mengenali karakter dari setiap kata pada citra dokumen. Dalam beberapa pengujian pengubahan parameter *BW Threshold* sangat diperlukan agar mendapatkan hasil pengenalan yang lebih baik. Berikut beberapa pengujian yang dilakukan :

1. Untuk pengujian pertama dilakukan dengan gambar yang berisi karakter yang merepresentasikan setiap huruf dan angka dengan ukuran 14pt, 16pt dan 18pt. Berikut tabel hasil pengujian pertama :

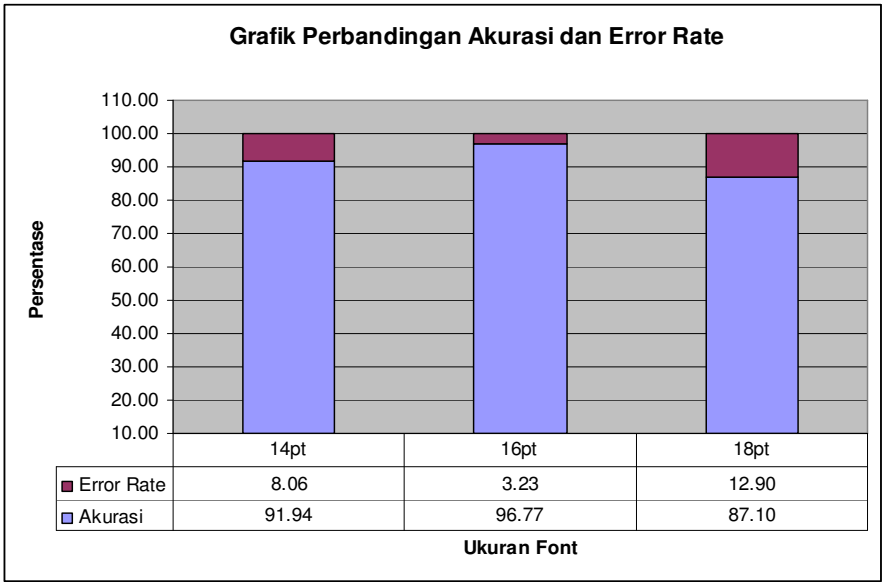
Tabel 4.2 Hasil Pengujian Per-karakter 14pt, 16pt dan 18pt

Karakter	Output Pengenalan					
	14pt	Valid	16pt	Valid	18pt	Valid
0	0	✔	0	✔	0	✔
1	1	✔	1	✔	1	✔
2	2	✔	2	✔	2	✔
3	3	✔	3	✔	3	✔
4	4	✔	4	✔	4	✔
5	5	✔	5	✔	5	✔
6	6	✔	6	✔	6	✔
7	7	✔	7	✔	7	✔
8	8	✔	8	✔	8	✔
9	9	✔	9	✔	9	✔
a	a	✔	a	✔	a	✔
b	b	✔	b	✔	b	✔
c	c	✔	c	✔	c	✔
d	d	✔	d	✔	d	✔
e	e	✔	e	✔	e	✔
f	t	✘	f	✔	f	✔
g	g	✔	g	✔	g	✔
h	h	✔	h	✔	h	✔
i	i	✔	i	✔	i	✔
j	j	✔	j	✔	j	✔
k	k	✔	k	✔	k	✔
l	l	✘	l	✘	l	✘
m	m	✔	m	✔	m	✔
n	n	✔	n	✔	n	✔

Karakter	Output Pengenalan					
	14pt	Valid	16pt	Valid	18pt	Valid
o	o	✔	o	✔	c	✘
p	p	✔	p	✔	p	✔
q	q	✔	q	✔	q	✔
r	r	✔	r	✔	I	✘
s	s	✔	s	✔	s	✔
t	t	✔	t	✔	f	✘
u	u	✔	u	✔	U	✘
v	V	✘	v	✔	V	✘
w	w	✔	w	✔	w	✔
x	x	✔	x	✔	x	✔
y	y	✔	y	✔	y	✔
z	z	✔	z	✔	Z	✘
A	A	✔	A	✔	A	✔
B	B	✔	B	✔	B	✔
C	C	✔	C	✔	C	✔
D	D	✔	D	✔	D	✔
E	E	✔	E	✔	E	✔
F	F	✔	F	✔	F	✔
G	G	✔	G	✔	G	✔
H	H	✔	H	✔	H	✔
I	I	✔	I	✔	I	✔
J	J	✔	J	✔	J	✔
K	K	✔	K	✔	K	✔
L	L	✔	L	✔	L	✔
M	M	✔	M	✔	V	✘
N	N	✔	N	✔	N	✔
O	O	✔	O	✔	O	✔
P	P	✔	P	✔	P	✔
Q	O	✘	Q	✔	Q	✔
R	R	✔	R	✘	R	✔
S	S	✔	S	✔	S	✔
T	T	✔	T	✔	T	✔
U	U	✔	U	✔	U	✔
V	v	✘	V	✔	V	✔
W	W	✔	w	✘	W	✔
X	X	✔	X	✔	X	✔
Y	Y	✔	Y	✔	Y	✔
Z	Z	✔	Z	✔	Z	✔

Tabel 4.3 Presentase Hasil Pengujian Per-karakter 14pt, 16pt dan 18pt

Ukuran Huruf	Keterangan			
	Jumlah Karakter	Jumlah Valid	Error Rate (%)	Akurasi (%)
14pt	62	57	8.06	91.94
16pt	62	60	3.23	96.77
18pt	62	54	12.90	87.10



Gambar 4.6 Grafik Perbandingan Akurasi dan *Error Rate* Pengenalan Tiap Karakter.

Dari tabel 4.2, pengujian rata-rata kesalahan pengenalan terdapat pada karakter yang memiliki kemiripan bentuk, seperti :

Tabel 4.4 Karakter Yang Memiliki Beberapa Kemiripan Bentuk

Karakter	Karakter hasil pengenalan		
	14pt	16pt	18pt
f	t	***valid***	***valid***
l (L kecil)	I (i besar)	I (i besar)	I (i besar)
o	***valid***	***valid***	c
r	***valid***	***valid***	I (i besar)
t	***valid***	***valid***	f
u (kecil)	***valid***	***valid***	U (besar)
v (kecil)	V (besar)	***valid***	V (besar)
z (kecil)	***valid***	***valid***	Z (besar)
M (besar)	***valid***	***valid***	V (besar)
Q (besar)	O (besar)	***valid***	***valid***
V (besar)	v (kecil)	***valid***	***valid***
W (besar)	***valid***	w (kecil)	***valid***

Dalam pengujian ini dilakukan dengan tujuan apakah sistem bisa mengenali dengan baik tiap pola karakter dengan membandingkan terhadap ukuran karakter yang sama dengan yang dilatihkan. Dari tabel 4.4 terlihat untuk beberapa karakter untuk ukuran *font* tertentu sistem tidak bisa mengenali karakter tersebut dengan benar, ini dimungkinkan disebabkan oleh proses normalisasi yang dilakukan oleh sistem, karena pada proses normalisasi *(stretch)* semua karakter diseragamkan ukurannya menjadi 8 x 8, hal inilah yang kemudian menjadikan sistem salah dalam mengenali pola yang dimaksud dikarenakan terjadinya perubahan nilai vektor *input* suatu karakter akibat proses normalisasi tersebut sehingga jaringan stabil pada pola lain. Untuk karakter l (L kecil) semua pengujian menunjukan hasil yang sama "l" apabila dilihat sistem telah berhasil mengenali karakter tersebut, akan tetapi apabila kita coba merubah karakter hasil pengenalan tersebut ke dalam jenis *font* yang lain maka terlihat bahwa karakter tersebut bukan yang dimaksud, hal ini dimungkinkan karena vektor pola tersebut memiliki kesamaan bentuk.

Pada sistem yang dibuat didesain dengan asumsi gambar yang digunakan sebagai input memiliki *noise* yang sekecil mungkin. Dalam pengambilan suatu *input* dari hasil *print screen* kadang membuat gambar menjadi kurang tajam dari objek aslinya yang bisa dianggap *noise* karena bisa menurunkan akurasi JST dalam melakukan pengenalan, walaupun dalam perspektif penglihatan mata manusia masih bisa terbaca sebagai karakter itu sendiri tetapi tidak dengan sistem. Akibat adanya *noise* tersebut yang memungkinkan sistem tidak bisa mengenali karakter yang dimaksud dengan benar sehingga sistem akan mengambil dan menampilkan suatu pola yang memiliki kedekatan antara vektor input dengan pola target.

Tabel 4.5 Karakter Yang Tidak Dikenali Sistem Dikarenakan Kesalahan Proses *Thresholding*

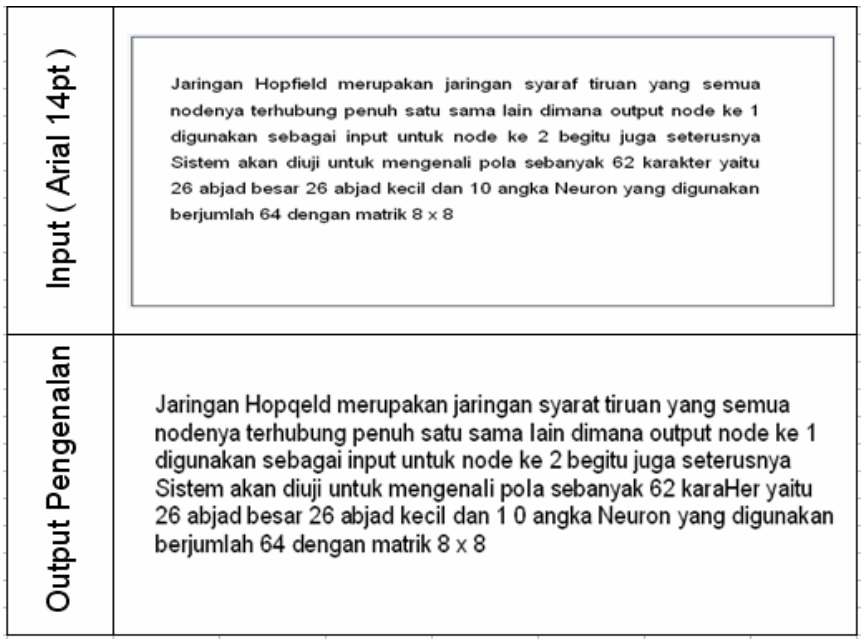
Karakter	Karakter hasil pengenalan		
	14pt	16pt	18pt
t	***valid***	***valid***	f
Q (besar)	O (besar)	***valid***	***valid***
o (kecil)	***valid***	***valid***	c

Akibat adanya pola input yang kurang sempurna seperti ketajaman gambar, objek karakter yang terlalu besar maupun terlalu kecil ukurannya. Untuk ketajaman

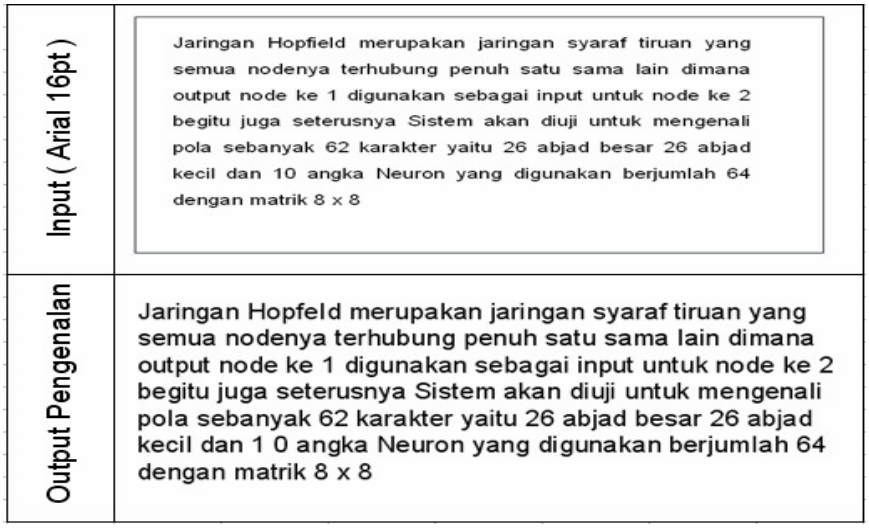
gambar akan mempengaruhi dalam proses pemisahan *background* dengan objek yaitu proses *grayscaling dan thresholding* di mana kadang nilai *BW Threshold* pada objek melebihi nilai *default BW Threshold* (196) sehingga objek tersebut yang seharusnya merepresentsikan suatu karakter tertentu akan terbaca sebagai *background.* Pada dasarnya walaupun *input* pengenalan berupa *image* hitam – putih tetapi *image* tersebut bukanlah *image* yang hanya tersusun atas warna hitam dan putih saja melainkan kombinasi dari representasi warna merah, hijau dan biru atau RGB (*Red, Green, Blue).* Sedangkan ukuran suatu karakter nantinya sangat berpengaruh dalam proses normalisasinya.

Secara persentase akurasi dalam pengenalan, persentase dalam mengenali karakter dengan ukuran 16pt paling besar yaitu 96,77 % sedangkan 14pt = 91.94% dan 18pt = 87,10 %. Akurasi yang didapat pada ukuran 16pt ini lebih dikarenakan ukuran pola tersebut sama dengan ukuran pola yang digunakan dalam pelatihan.

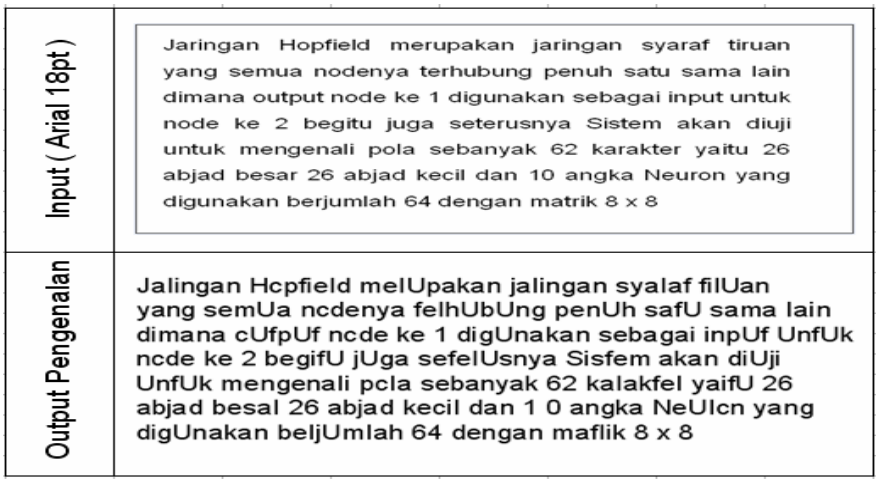
2. Untuk pengujian kedua dilakukan dengan gambar yang berisi rangkaian kata dalam paragraf, hanya dibedakan ukuran *font* . Berikut hasil pengujian untuk semua ukuran huruf :



Gambar 4.7 Pengujian Rangkaian Kata Dalam Paragraf *Arial* 14pt



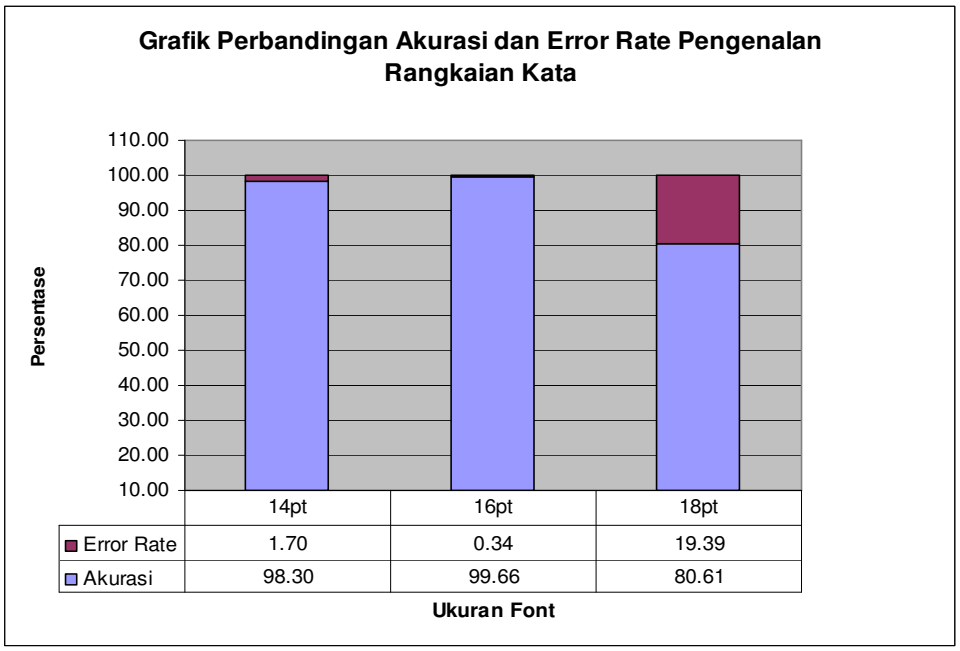
Gambar 4.8 Pengujian Rangkaian Kata Dalam Paragraf *Arial* 16pt



Gambar 4.9 Pengujian Rangkaian Kata Dalam Paragraf *Arial* 18pt

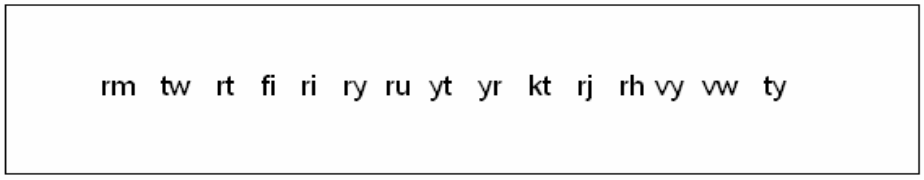
Tabel 4.6 Hasil Percobaan Pengenalan Rangkaian Kata

CITRA UJI JST	CITRA - 1	CITRA - 2	CITRA - 3
Ukuran Font	14pt	16pt	18pt
Nama File	paragrap_14pt_1	paragrap_16pt_1	paragrap_18pt_1
Ukuran File	687 x 333	687 x 333	687 x 333
Jumlah Kata	58	58	58
Jumlah Baris	6	7	7
Jumlah Karakter (Tanpa Spasi)	294	294	294
Jumlah Pengenalan Karakter Valid	289	293	237
Jumlah Karakter yang Tidak Dikenali	5	1	57
Error Rate (%)	**1.70**	**0.34**	**19.39**
Akurasi (%)	**98.30**	**99.66**	**80.61**



Gambar 4.10 Grafik Perbandingan Akurasi dan *Error Rate* Pengenalan Rangkaian Kata

Selain disebabkan adanya kemiripan bentuk, kesalahan pengenalan juga disebabkan karena proses segmentasi yang kurang baik pada dua karakter yang tersambung sangat tipis sehingga seharusnya terdapat dua karakter akan tetapi sistem menganggap itu sebagai satu karakter. Berikut akan diuji beberapa kombinasi karakter yang memungkinkan akan terjadi kesalahan pengenalan diakibatkan kesalahan pada proses segmentasi. Pengambilan kombinasi karakter ini diambil secara *random* . Hasil pengenalan untuk kombinasi karakter tersebut kemungkinan akan berbeda untuk masing-masing ukuran *font*.



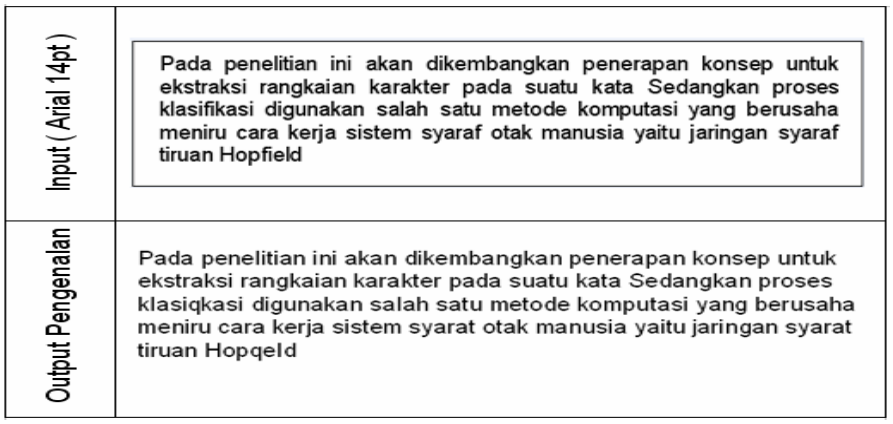
Gambar 4.11 Kombinasi Karakter Yang Memungkinkan Terjadinya Kesalahan Pengenalan Akibat Proses Segmentasi

Tabel 4.7 Hasil Percobaan Terhadap Karakter Yang Mengalami Kesalahan Segmentasi

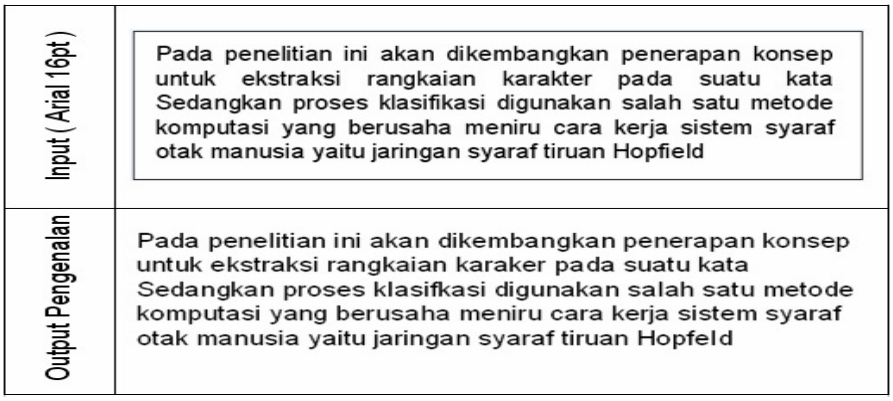
Rangkaian karakter	Output Pengenalan		
	14pt	16pt	18pt
rm	***valid***	***valid***	lm
tw	Y (besar)	j (kecil)	Y (besar)
rt	h	h	h
fi	q	f	***valid***
ri	***valid***	***valid***	li
ry	D	***valid***	ly
ru	***valid***	***valid***	lU
yt	y (kecil)	***valid***	yf
yr	***valid***	***valid***	yl
kt	H	k (kecil)	kf
rj	***valid***	***valid***	lj
rh	***valid***	***valid***	lh
vy	W (besar)	***valid***	Vy
vw	w (kecil)	w (kecil)	w (kecil)
ty	j (kecil)	***valid***	fy

Keberhasilan sistem dalam mengenali suatu karakter sangat berpengaruh dari hasil segmentasi yang baik. Pada tabel 4.7 terlihat beberapa rangkaian karakter yang tidak bisa dikenali oleh sistem dikarenakan rangkaian karakter tersebut tidak tersegmentasi dengan baik hal ini disebabkan karena karakter tersebut tersambung dengan tipis atau jarak antara kedua karakter tesebut kurang dari parameter yang digunakan untuk memisahkan antar karakter. Tersambung di sini dalam arti pada suatu koordinat *pixel p(x,y)* ada bagian dari kedua karakter tersebut yang berhimpitan koordinatnya oleh karena itu sistem tidak bisa memisahkan kedua karakter tersebut sebagai dua karakter dikarenakan tidak memenuhi parameter yang ada sehingga sistem menganggapnya sebagai satu karakter. Semakin besar ukuran dari suatu *font* semakin kecil kemungkinan terjadinya kesalahan pada proses segmentasi. Kemungkinan terbesar kesalahan pengenalan pada ukuran *font* yang lebih besar adalah kesalahan pada saat proses normalisasi yang menyebabkan JST stabil pada pola lain. Untuk ukuran *font* yang lebih kecil kemungkinan kesalahan terjadi pada proses segmentasi. Pada percobaan ini terlihat sistem cukup baik dalam mengenali suatu rangkaian kata yaitu 98.30%

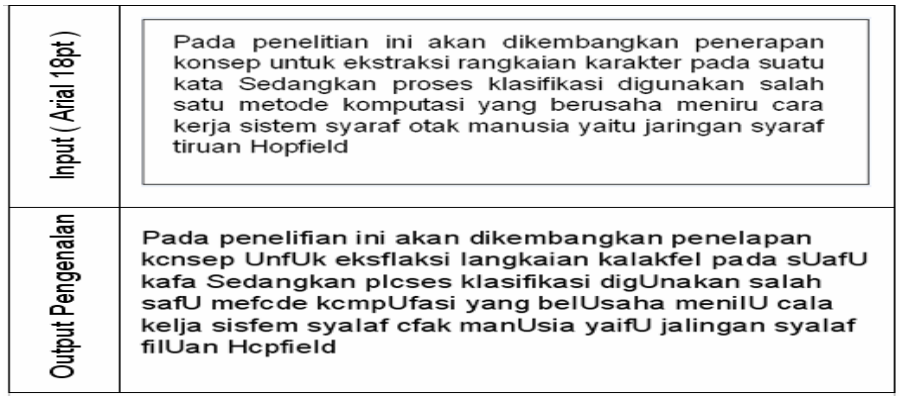
untuk 14pt, 99.66% untuk 16pt dan 80.61% untuk 18pt. Berikut beberapa pengujian lainnya untuk pengenalan terhadap rangkaian karakter :



Gambar 4.12 Pengujian II Rangkaian Kata Dalam Paragraf *Arial* 14pt



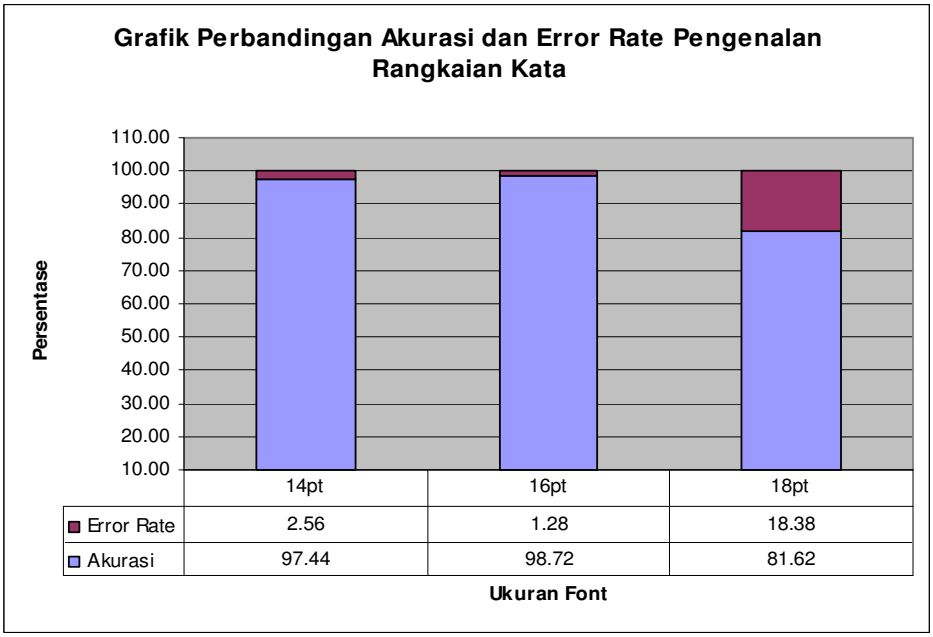
Gambar 4.13 Pengujian II Rangkaian Kata Dalam Paragraf *Arial* 16pt



Gambar 4.14 Pengujian II Rangkaian Kata Dalam Paragraf *Arial* 18pt

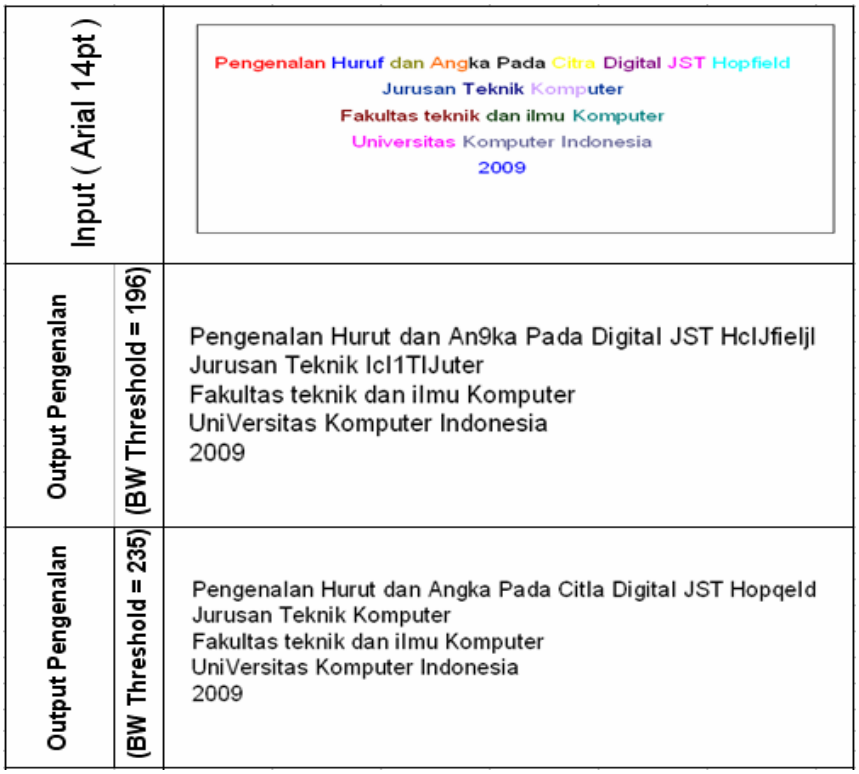
Tabel 4.8 Hasil Percobaan II Pada Rangkaian Kata

CITRA UJI JST	CITRA - 1	CITRA - 2	CITRA - 3
Ukuran Font	14pt	16pt	18pt
Nama File	paragrap_14pt_2	paragrap_16pt_2	paragrap_18pt_2
Ukuran File	620 x 138	620 x 138	620 x 138
Jumlah Kata	36	36	36
Jumlah Baris	5	5	6
Jumlah Karakter (Tanpa Spasi)	234	234	234
Jumlah Pengenalan Karakter Valid	228	231	191
Jumlah Karakter yang Tidak Dikenali	6	3	43
Error Rate (%)	**2.56**	**1.28**	**18.38**
Akurasi (%)	**97.44**	**98.72**	**81.62**

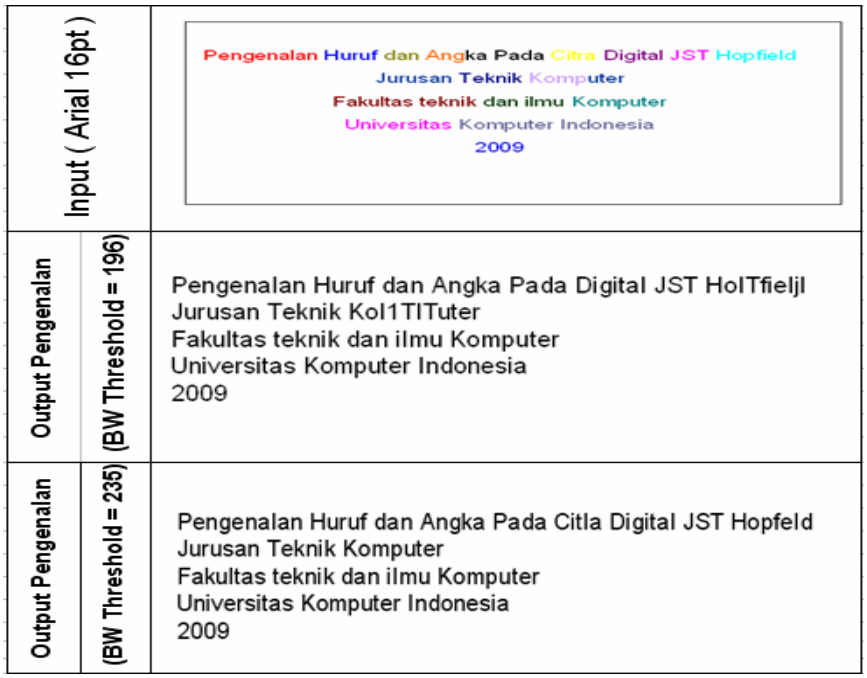


Gambar 4.15 Grafik Perbandingan Akurasi dan Error Rate Pengenalan Rangkaian Kata Pengujian II

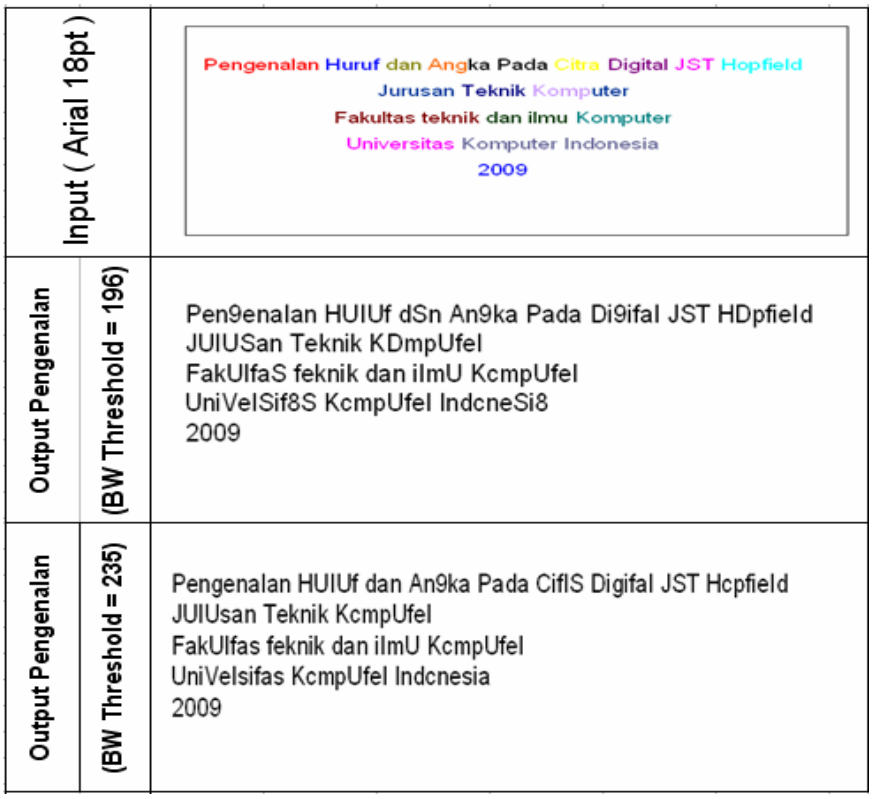
3. Untuk Pengujian ketiga dilakukan dengan memberikan masukan berupa citra karakter berwarna dengan ukuran font 14pt, 16pt dan 18pt. Pengujian dilakukan untuk melihat seberapa besar pengaruh pengubahan parameter *BW Threshold* terhadap *output* pengenalan.



Gambar 4.16 Pegujian Hasil Pengenalan Rankaian Kata Berwarna Terhadap Pengaruh Perubahan *BW Threshold Arial* 14pt



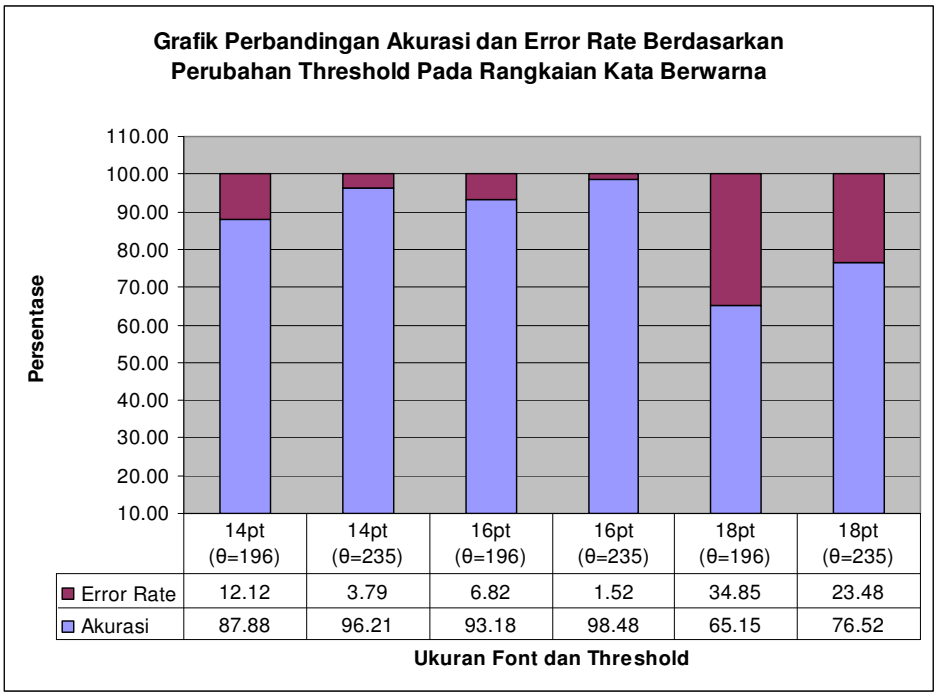
Gambar 4.17 Pegujian Hasil Pengenalan Rankaian Kata Berwarna Terhadap Pengaruh Perubahan *BW Threshold Arial* 16pt



Gambar 4.18 Pegujian Hasil Pengenalan Rankaian Kata Berwarna Terhadap Pengaruh Perubahan *BW Threshold Arial* 18pt

Tabel 4.9 Pengaruh Perubahan *BW Threshold* Terhadap *Output* Pengenalan Rangkaian Kata Berwarna

CITRA UJI JST	CITRA - 1	CITRA - 1	CITRA - 2	CITRA - 2	CITRA - 3	CITRA - 3
Ukuran Font	14pt	14pt	16pt	16pt	18pt	18pt
BW Threshold	196	235	196	235	196	235
Nama File	warna_14pt	warna_14pt	warna_16pt	warna_16pt	warna_18pt	warna_18pt
Ukuran File	563 x 257	563 x 257	605 x 257	605 x 257	676 x 267	676 x 267
Jumlah Kata	21	21	21	21	21	21
Jumlah Baris	5	5	5	5	5	5
Jumlah Karakter (Tanpa Spasi)	132	132	132	132	132	132
Jumlah Pengenalan Karakter Valid	116	127	123	130	86	99
Jumlah Karakter yang Tidak Dikenali	16	5	9	2	46	33
Error Rate (%)	**12.12**	**3.79**	**6.82**	**1.52**	**34.85**	**23.48**
Akurasi (%)	**87.88**	**96.21**	**93.18**	**98.48**	**65.15**	**76.52**



Gambar 4.14 Grafik Perbandingan Akurasi dan *Error Rate* Pengenalan Karakter Berwarna Bedasarkan Perubahan *BW Threshold*

Pada pengujian ketiga sistem dicoba untuk mengenali masukan citra karakter berwarna. Dalam pengujian ini merubah-rubah parameter *BW Threshold* sangat diperlukan, *BW Threshold default* untuk sistem ini adalah 196. Seperti yang sudah dijelaskan sebelumnya pengaruh *BW Threshold* itu sendiri dikarenakan nilai ini dipakai sebagai batas ambang untuk memisahkan antara karakter dengan *background* berdasarkan intensitas tiap-tiap *pixel* karakter pada citra. Dari tabel diatas terlihat bahwa pada posisi *BW Threshold default* sistem hanya mampu mengenali *input* yang diberikan sebesar 87.88% untuk 14pt , 93.18% untuk 16pt dan 65.15% untuk 18pt, ketika parameter *BW Threshold* dirubah 235 terjadi kenaikan yang cukup signifikan terhadap hasil pengenalan yaitu 96.21% untuk 14pt , 98.48% untuk 16pt dan 76.52% untuk 18pt. Dari hasil pengujian untuk *BW Threshold* 196 terlihat sistem tidak bisa mengenali dengan baik dari *input* rangkaian karakter berwarna, dari *output* tersebut juga terdapat kata yang hilang yaitu “Citra” ini dimungkinkan karena sistem menganggapnya sebagai *background* dikarenakan nilai-nilai *pixel* pada kata tersebut melebihi nilai *BW Threshold default* yaitu 196. Dalam hal ini dapat disimpulkan bahwa sistem sangat sulit untuk mengenali pola karakter yang memiliki nilai *pixel* yang

mendekati nilai *pixel* warna putih. Telihat perubahan ketika nilai dari *BW Threshold* dirubah menjadi 235 sistem bisa dikatakan mengenali rangkaian karakter yang di*input*-kan kepadanya walaupun masih terdapat beberapa kesalahan pengenalan, tetapi inilah *output* maksimal yang bisa dihasilkan. Pemilihan nilai *BW Threshold* ini didasarkan pada perspektif penglihatan mata manusia dengan membandingkan antara *input* dengan *output* pengenalan untuk mendapatkan *output* pengenalan yang paling maksimal.