

## ANALISIS DAN IMPLEMENTASI NEOCOGNITRON NEURAL NETWORK DALAM IDENTIFIKASI KARAKTER NUMERIK

Hendra Winata<sup>1</sup>, Suyanto<sup>2</sup>, Adiwijaya<sup>3</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

### Abstrak

Suatu image karakter numerik memiliki sangat banyak kemungkinan pola untuk setiap kategorinya. Di samping itu, kemungkinan adanya noise, deformasi, dan pergeseran dalam setiap polanya sangat tinggi. Oleh sebab itu, hampir mustahil untuk melakukan identifikasi kategori suatu karakter numerik dengan algoritma konvensional. Pada tugas akhir ini dikembangkan suatu aplikasi untuk melakukan proses identifikasi image karakter numerik dengan menggunakan proses learning dari suatu jenis jaringan syaraf tiruan yang awalnya dirancang khusus untuk melakukan identifikasi terhadap pola image 2D yaitu neocognitron. Data yang digunakan adalah image karakter numerik(full feature) dari beberapa relawan dengan total jumlah 6000 pola (600 pola untuk setiap kategori) sebagai data testing serta local feature dari data Fukushima pada model supervised neocognitron tahun 1988 sebagai data training. Berdasarkan proses testing, diperoleh akurasi terbaik sebesar 70,05%.

**Kata Kunci :** Kata kunci : Jaringan Syaraf Tiruan, Neocognitron, learning, pola, image 2D, karakter numerik.

### Abstract

Numeric character image has many possible pattern for each category. Beside that, possibility to have noise, deformation, and translation in each pattern is very high. As impact, it is impossible to make an application to identify numeric character category with conventional algorithm. This final project is develop to identify numeric character with learning process one kind of artificial neural network designed special for 2D image pattern identification called neocognitron. Data set used in this system are images from handwritten character numeric(full feature) with total pattern is 6000 patterns(600 patterns for each category). created by volunteers as testing data and local feature from supervised neocognitron model by Fukushima in 1988 as training data. As the result of experiment, the highest accuracy is 70,05%.

**Keywords :** Keyword : Artificial Neural Network, Neocognitron, learning, pattern, 2D image, numeric character.

Telkom  
University

# 1. Pendahuluan

## 1.1 Latar Belakang

*Image* karakter numerik memiliki banyak peranan dalam berbagai bidang kehidupan manusia. Beberapa peranan penting yang dimaksud diantaranya yaitu dalam identifikasi suatu plat kendaraan bermotor dan pengenalan tulisan tangan yang melibatkan karakter numerik. Untuk melakukan identifikasi terhadap *image* karakter numerik tidaklah mudah sebab jumlah kemungkinan pola yang dimiliki sangat banyak. Perbedaan pola tersebut diantaranya dipengaruhi oleh *noise*, deformasi, dan pergeseran. Oleh sebab itu, diperlukan metode yang mampu melakukan generalisasi dan toleransi berbagai pola yang dimiliki oleh *image* karakter numerik tersebut.

*Pattern recognition* seperti identifikasi *image* karakter numerik merupakan salah satu jenis permasalahan yang dapat diselesaikan menggunakan *artificial neural network (ANN)* atau jaringan syaraf tiruan (JST) yang termasuk teknik *learning* dalam bidang *artificial intelligence* seperti *perceptron*, *MLP (Multi Layer Perceptron)*, *Kohonen* atau *Self Organization Map (SOM)*, dan *neocognitron*. Di samping menggunakan JST, *optical character recognition (OCR)* dan *Feature Point Extraction* juga dapat digunakan untuk melakukan identifikasi *image* karakter numerik.

Selain metode tersebut di atas, sebenarnya telah ditemukan suatu model JST yang didesain khusus untuk menangani permasalahan *2D (2 Dimension) visual pattern recognition* bernama *neocognitron* oleh Fukushima pada tahun 1979[6]. *Neocognitron* yang merupakan *hierarchical multilayered neural network* mengadopsi sistem visual yang ada pada mamalia dimana terjadi beberapa tahap *feature extraction* dengan *feature* awal berasal dari *simple local feature* yang dimiliki oleh data *input*. Sistem *visual pathway* mamalia diadopsi sebab kehandalannya dalam melakukan *feature extraction* terhadap *pattern* yang dimiliki objek-objek citra *2D* yang ditangkapnya [2]. Namun ironis, *neocognitron* dengan kehandalan yang dimiliki justru masih kurang populer. Hal tersebut salah satunya disebabkan oleh kompleksitas arsitektur yang dimiliki *neocognitron*[6].

Untuk memudahkan proses analisis dan pembuktian kehandalan *neocognitron*, maka dirancang suatu sistem identifikasi *image* karakter numerik. Berdasarkan *output* yang dihasilkan sistem saat proses *testing*, dapat dilakukan pengukuran akurasi dari *neocognitron* yang dibangun. *Neocognitron* dapat dilatih dengan metode *supervised* maupun *unsupervised learning*, namun pada tugas akhir ini dilakukan analisis terhadap *neocognitron* yang menggunakan metode *supervised learning* dengan struktur dan *training pattern* berdasarkan model yang telah dibangun oleh Fukushima [3].

## 1.2 Perumusan Masalah

Beberapa permasalahan dalam tugas akhir ini dapat didefinisikan sebagai berikut :

1. Bagaimana cara kerja *neocognitron* dalam melakukan identifikasi terhadap *image* karakter numerik?

2. Bagaimana pengaruh parameter *selectivity*, ukuran *connection area*, dan *fixed weight* di setiap *stage* untuk menghasilkan *neocognitron* yang memiliki akurasi tinggi dalam mengidentifikasi *image* karakter numerik, beserta argumentasinya?

### 1.3 Tujuan

Tujuan yang hendak dicapai dalam tugas akhir ini adalah:

1. Menganalisis cara kerja *neocognitron* dalam mengidentifikasi *image* karakter numerik.
2. Mengimplementasikan *neocognitron* dalam mengidentifikasi *image* karakter numerik.
3. Menganalisis pengaruh parameter ukuran *connection area*, *selectivity* dan *fixed weight* di setiap *stage* untuk menghasilkan *neocognitron* yang memiliki akurasi tinggi dalam mengidentifikasi *image* karakter numerik.

### 1.4 Batasan Masalah

Dalam implementasi tugas akhir ini dibatasi oleh beberapa hal, sebagai berikut :

1. Data yang digunakan berupa *image* karakter numerik dengan dua warna yaitu hitam dan putih dimana nilai *pixel* warna hitam adalah 1 sedangkan nilai *pixel* warna putih adalah 0.
2. Jenis *neocognitron* yang digunakan adalah *neocognitron* dengan metode pelatihan *supervised*.
3. Parameter performansi dari sistem ini adalah akurasi yang dinyatakan dalam tingkat akurasi antara hasil identifikasi *neocognitron* dengan karakter numerik target yang telah didefinisikan sebelumnya.

### 1.5 Metode Penyelesaian Masalah

Metode penelitian yang digunakan untuk memecahkan permasalahan dalam Tugas Akhir ini terdiri dari 5 tahap, yaitu:

1. Tahap Pengumpulan Data dan Studi Literatur  
Pada tahap ini, dilakukan pengumpulan data, yaitu kumpulan pola karakter numerik yang bervariasi sebagai data *training* dan *testing*. Selain itu, pada tahap ini juga dilakukan pendalaman materi terhadap konsep dan teori *neocognitron* baik melalui konsultasi kepada para pakarnya, maupun studi literatur-literatur terkait dan pustaka *online*.
2. Tahap Perancangan Sistem  
Pada tahap ini, dilakukan perancangan sistem secara umum, kemudian menjabarkannya secara lebih rinci. Data *training* dan *testing* yang digunakan untuk melatih dan menguji program juga dirancang pada tahap ini.
3. Tahap Implementasi  
Pada tahap ini, dilakukan implementasi dari rancangan sistem yang telah dibuat sebelumnya menjadi sistem identifikasi *image* karakter numerik. *Neocognitron* digunakan untuk proses identifikasi *image* karakter numerik melalui proses *learning* yang terdiri dari beberapa tahap *feature extraction*.
4. Tahap Pengujian Sistem  
Pada tahap ini, dilakukan *testing* program yang sudah diimplementasikan menggunakan *tools* MATLAB 7.6. *Testing* dilakukan dengan menggunakan

data *testing* berupa pattern karakter numerik sesuai pada batasan masalah, yang belum pernah diidentifikasi oleh *neocognitron* yang sudah dipersiapkan sebelumnya, sehingga diketahui tingkat akurasi.

5. Tahap Analisis Hasil Pengujian  
Berdasarkan tahap pengujian sistem yang dilakukan sebelumnya, dilakukan analisis terhadap *output* yang dihasilkan dengan target yang telah ditentukan, dikaitkan dengan proses *learning* dan parameter-parameter terkait yang mempengaruhi. Berdasarkan analisis tersebut, diambil kesimpulan cara kerja dan parameter terkait yang mempengaruhi akurasi dari *neocognitron*.
6. Tahap Pembuatan Laporan  
Pada tahap ini, dilakukan penyusunan laporan akhir dan pengumpulan dokumentasi dengan mengikuti kaidah penulisan yang benar dan sesuai dengan ketentuan-ketentuan atau sistematika yang telah ditetapkan oleh institusi.

## 1.6 Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika penulisan sebagai berikut :

### **BAB I Pendahuluan**

Bab ini menguraikan tugas akhir ini secara umum, meliputi latar belakang masalah, perumusan masalah, tujuan, batasan masalah, dan metode yang digunakan.

### **BAB II Landasan Teori**

Bab ini membahas mengenai uraian teori yang berhubungan dengan jaringan syaraf tiruan *supervised neocognitron* dan *feature extraction*.

### **BAB III Analisis Perancangan dan Implementasi**

Bab ini berisi analisis kebutuhan dari sistem dan masalah-masalah yang ada di dalamnya. Hasil analisis ini dituangkan ke dalam suatu sistem pemodelan secara terstruktur. Berdasarkan tahap analisis kemudian dilanjutkan ke tahap perancangan dan implementasi.

### **BAB IV Pengujian dan Analisis Hasil Percobaan**

Bab ini membahas mengenai pengujian hasil implementasi yang telah dilakukan pada bab sebelumnya. Pengujian dilakukan dengan membandingkan hasil implementasi dengan data aslinya. Tahap Pengujian dilanjutkan dengan tahap analisis hasil pengujian.

### **BAB V Kesimpulan dan Saran**

Berisi kesimpulan dari penulisan Tugas Akhir ini dan saran-saran yang diperlukan untuk pengembangan lebih lanjut.

## 4. Pengujian dan Analisis

### 4.1 Pengujian Sistem

Pengujian sistem dilakukan terkait dengan proses observasi sesuai dengan skenario yang dirancang untuk memperoleh parameter *neocognitron* yang memiliki akurasi tinggi sebagai *neocognitron* yang optimal.

#### 4.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian adalah :

1. Menganalisis pengaruh dari parameter *connection area*, *selectivity*, dan *fixed weight* dalam menghasilkan *neocognitron* yang memiliki akurasi tinggi dalam mengidentifikasi *image* karakter numerik yang tidak terpengaruh oleh pergeseran, deformasi maupun *noise* pada suatu karakter numerik.
2. Menganalisis *neocognitron* yang telah dibangun dalam melakukan identifikasi *image* karakter numerik.
3. Menganalisis performansi *neocognitron* dalam melakukan identifikasi *image* karakter numerik.

#### 4.1.2 Strategi Pengujian

Untuk mencapai tujuan pengujian yang telah didefinisikan sebelumnya, dibutuhkan suatu strategi pengujian dalam bentuk skenario observasi untuk menghasilkan jaringan *neocognitron* yang optimal.

##### 4.1.2.1 Observasi Parameter *Neocognitron*

Nilai – nilai parameter yang ada pada Tabel 3-1, 3-3, dan 3-4 adalah nilai standar selama observasi dilakukan (angka yang dicetak **tebal**).

Tabel 4-1 : Daerah Observasi Parameter

PARAMETER	STAGE 1		STAGE 2		STAGE 3		STAGE 4	
	$U_{S1}$	$U_{C1}$	$U_{S2}$	$U_{C2}$	$U_{S3}$	$U_{C3}$	$U_{S4}$	$U_{C4}$
Ukuran <i>Connection Area</i> ( $A_i, D_i$ )	<b>9</b>	25 49 81	9 25 49 81	9 25 <b>49</b> 81	9 25 49 81	9 25 49 81	9 25 49	<b>9</b>
<i>Selectivity</i> ( $r$ )	0,1; 0,2; 0,3 ...; <b>1,7</b> ; ...; 3,8; 3,9; 4,0		0,1; 0,2; 0,3 ...; 2,0; ...; 3,8; 3,9; <b>4,0</b>		0,1; 0,2; 0,3 ...; <b>1,5</b> ; ...; 3,8; 3,9; 4,0		0,1; 0,2; 0,3 ...; <b>1,0</b> ; ...; 3,8; 3,9; 4,0	
<i>Fixed Weight</i> ( <i>S-cell Mask</i> ) <i>c-weight</i>	0,1; 0,2; 0,3 ...; <b>0,9</b> ; ...; 1,8; 1,9; 2,0		0,1; 0,2; 0,3 ...; <b>0,9</b> ; ...; 1,8; 1,9; 2,0		0,1; 0,2; 0,3 ...; <b>0,9</b> ; ...; 1,8; 1,9; 2,0		0,1; 0,2; 0,3 ...; <b>0,8</b> ; ...; 1,8; 1,9; 2,0	
<i>Fixed Weight</i> ( <i>C-cell Mask1</i> ) <i>d-weight</i>	0,1; 0,2; 0,3 ...; 1,7; ...; 3,8; 3,9; <b>4,0</b>		0,1; 0,2; 0,3 ...; 1,7; ...; 3,8; 3,9; <b>4,0</b>		0,1; 0,2; 0,3 ...; <b>2,5</b> ; ...; 3,8; 3,9; 4,0		0,1; 0,2; 0,3 ...; <b>1,0</b> ; ...; 3,8; 3,9; 4,0	
<i>Fixed Weight</i>	0,1; 0,2; 0,3		0,1; 0,2; 0,3		0,1; 0,2; 0,3		0,1; 0,2; 0,3	

(C-cell Mask2) <i>d-weight</i>	...; <b>0,9</b> ; ... ; 1,8 ; 1,9 ; 2,0	...; <b>0,8</b> ; ... ; 1,8 ; 1,9 ; 2,0	...; <b>0,7</b> ; ... ; 1,8 ; 1,9 ; 2,0	...; <b>1,0</b> ; ... ; 1,8 ; 1,9 ; 2,0
-----------------------------------	--	--	--	--

Selama proses *testing* dilakukan observasi terhadap beberapa nilai untuk setiap parameter yang ada pada Tabel 4-1, bukan kombinasi nilai dari parameter-parameter tersebut. Hal ini disebabkan oleh kemungkinan kombinasi nilai setiap parameter yang ada sangat kompleks, maka untuk memudahkan proses observasi terhadap pengaruh setiap parameter tersebut untuk menghasilkan *neocognitron* yang optimal, maka di setiap skenario observasi, hanya ada satu parameter yang dimodifikasi dengan *range* nilai sesuai dengan Tabel 4-1, sedangkan nilai parameter yang lain tetap sesuai nilai standar.

Untuk lebih jelasnya dapat dilihat pada Tabel 4-2 yang menggambarkan 3 skenario perubahan nilai hanya dilakukan pada  $D_1$  dengan nilai parameter yang lain seperti *learning coefficient* dan parameter arsitektur jaringan merupakan nilai standar yang dapat dilihat pada Tabel 3-1 dan 3-4. Khusus untuk ukuran *connection area*  $U_{S1}(A_1)$  dan  $U_{C4}(D_4)$  tetap karena untuk  $A_1$  mengikuti ukuran *receptive field* setiap *cell* dalam  $U_{S1}$  sedangkan untuk  $D_4$  mengikuti ukuran *cell plane* pada  $U_{S4}$  yang hanya berukuran 9 (3x3). Adapun ukuran *connection area*  $U_{S4}(A_4)$ , hanya memiliki 3 skenario nilai karena ukuran *cell plane* di  $U_{S3}$  hanyalah 49, yang artinya ukuran maksimal yang memungkinkan untuk nilai  $A_4$  hanyalah 49.

Tabel 4-2 : Skenario dengan Perubahan Nilai pada  $D_1$

No.	PARAMETER	STAGE 1		STAGE 2		STAGE 3		STAGE 4	
		$U_{S1}$	$U_{C1}$	$U_{S2}$	$U_{C2}$	$U_{S3}$	$U_{C3}$	$U_{S4}$	$U_{C4}$
1	$A_1$ dan $D_1$	9	9	25	49	25	25	25	9
	<i>Selectivity</i> ( $r$ )	1,7		4,0		1,5		1,0	
	<i>S-cell Mask</i> ( $\gamma_1$ )	0,9		0,9		0,9		0,8	
	<i>C-cell Mask1</i> ( $\delta_1$ )	4,0		4,0		2,5		1,0	
	<i>C-cell Mask2</i> ( $\delta_1$ )	0,9		0,9		0,7		1,0	
2	$A_1$ dan $D_1$	9	25	25	49	25	25	25	9
	<i>Selectivity</i> ( $r$ )	1,7		4,0		1,5		1,0	
	<i>S-cell Mask</i> ( $\gamma_1$ )	0,9		0,9		0,9		0,8	
	<i>C-cell Mask1</i> ( $\delta_1$ )	4,0		4,0		2,5		1,0	
	<i>C-cell Mask2</i> ( $\delta_1$ )	0,9		0,9		0,7		1,0	
3	$A_1$ dan $D_1$	9	49	25	49	25	25	25	9
	<i>Selectivity</i> ( $r$ )	1,7		4,0		1,5		1,0	
	<i>S-cell Mask</i> ( $\gamma_1$ )	0,9		0,9		0,9		0,8	
	<i>C-cell Mask1</i> ( $\delta_1$ )	4,0		4,0		2,5		1,0	

	<i>C-cell Mask2</i> ( $\delta_1$ )	0,9	0,9	0,7	1,0
--	------------------------------------	-----	-----	-----	-----

Berdasarkan Tabel 4-3, data yang digunakan selama proses observasi berlangsung adalah *data training\_data* dengan bentuk data dapat dilihat pada lampiran A [2,4]. Selama proses *training* berlangsung, tidak ada perhitungan *error* yang dilakukan sebab yang terjadi hanyalah proses perhitungan maju (*forward*) dari *input layer U<sub>0</sub>* pada *stage 0* hingga *S-layer* pada *stage* maksimum untuk mendapatkan nilai – nilai *modified weight* (bobot ‘a’ dan bobot ‘b’) di setiap *stage*. Seperti yang telah dijelaskan pada bagian landasan teori, *modified weight* hanyalah dimiliki oleh *S-cell* sehingga tidak perlu dilakukan perhitungan hingga *output layer* (*C-layer* pada *stage* maksimum) dan inilah yang menyebabkan pada proses *training* tidak dilakukan perhitungan *error* sama sekali di samping bentuk *data training* yang digunakan pada *neocognitron* sangatlah jauh berbeda dengan *data testing* karena hanya merupakan *local – local feature*, bukan *full feature* (karakter numerik yang utuh).

Berdasarkan karakteristik proses *training neocognitron* yang telah dijelaskan sebelumnya, untuk mengetahui *neocognitron* yang optimal yaitu *neocognitron* yang memiliki akurasi tertinggi maka dilakukan proses *testing* untuk setiap arsitektur yang telah melalui proses *training* dengan menggunakan data *testing* untuk proses observasi (*testing\_data\_for\_observation*), dapat dilihat pada Tabel 4-3. Data *testing* untuk proses observasi berbeda dengan data *testing* untuk proses *testing*. Data *testing* untuk proses observasi ditujukan untuk perhitungan akurasi selama proses observasi berlangsung sedangkan data *testing* ditujukan untuk proses perhitungan akurasi selama proses *testing* berlangsung dengan menggunakan sebuah *neocognitron* optimal yang diperoleh dari proses observasi. Meskipun data yang digunakan berbeda dan dengan tujuan berbeda, proses yang terjadi antara kedua proses *testing* tersebut dan bentuk datanya yang berupa *full feature* sama saja.

Tabel 4-3 : Alokasi Data untuk Setiap Proses

Proses	Data yang Digunakan
<i>Training</i>	<i>training_data</i>
<i>Testing_for_observation</i>	<i>testing_data_for_observation</i>
<i>Testing</i>	<i>testing_data</i>

#### 4.1.2.2 Pengujian *Neocognitron* yang Dihasilkan

Untuk menguji akurasi *neocognitron* dalam mengidentifikasi karakter numerik, maka dilakukan proses *testing* dengan menggunakan data *testing\_data* yang berupa 6000 digit karakter numerik 0 sampai 9 dengan 600 pola digit untuk setiap kelasnya. Setiap digit disimpan dalam suatu *file \*.pat* dan kemudian digabungkan untuk menjadi satu *file \*.mat testing\_data*. Jumlahnya jauh lebih banyak dibandingkan dengan data *testing\_data\_for\_observation* karena data pada data *testing\_data\_for\_observation* merupakan data acuan untuk membuat data *training\_data* [2,4] sebagai acuan untuk mencari jaringan *neocognitron* yang optimal sedangkan *testing\_data* dibuat oleh beberapa relawan sebagai acuan akurasi jaringan *neocognitron* optimal yang diperoleh saat proses observasi.

Berdasarkan proses *testing* yang dilakukan dengan menggunakan data *testing\_data*, diperoleh nilai akurasi yang merupakan perbandingan antara jumlah nilai *output* yang benar dari hasil identifikasi jaringan *neocognitron* dengan targetnya masing – masing dengan jumlah data *testing\_data*. Rumus perhitungannya dapat dilihat pada persamaan 4.1.

$$\text{akurasi (\%)} = \frac{p}{n} \times 100 \% \quad (4.1)$$

dimana:

$n$  = jumlah data *testing*

$p$  = jumlah pola yang benar

Persamaan akurasi yang digunakan pada persamaan 4.1 bertujuan agar nilai akurasi yang dihasilkan oleh *neocognitron* sebanding dengan tingkat *noise* dan deformasi yang tinggi yang diberikan pada saat proses *testing*, baik itu pada data *testing\_data\_for\_observation* untuk perhitungan akurasi saat proses *training* maupun pada data *testing\_data* untuk perhitungan akurasi saat proses *testing*. Di samping itu, kemampuan sistem identifikasi dalam hal ini *neocognitron* untuk tidak mengenali suatu karakter yang memang jauh atau bahkan bukan termasuk karakter numerik harus mempengaruhi akurasi yang dihasilkan dimana manusia sekalipun sangat sulit untuk menentukan kelas dari karakter numerik tersebut.

## 4.2 Analisis Hasil Pengujian

Berdasarkan proses pengujian yang telah dilakukan dengan menggunakan strategi pengujian yang telah dirancang, maka dilakukan analisis terhadap hasil yang diperoleh dari proses pengujian sebelumnya. Hal ini bertujuan untuk menunjukkan pengaruh strategi pengujian yang telah dirancang terhadap pencapaian tujuan proses pengujian yang telah didefinisikan sebelumnya.

### 4.2.1 Analisis Parameter *Connection Area*

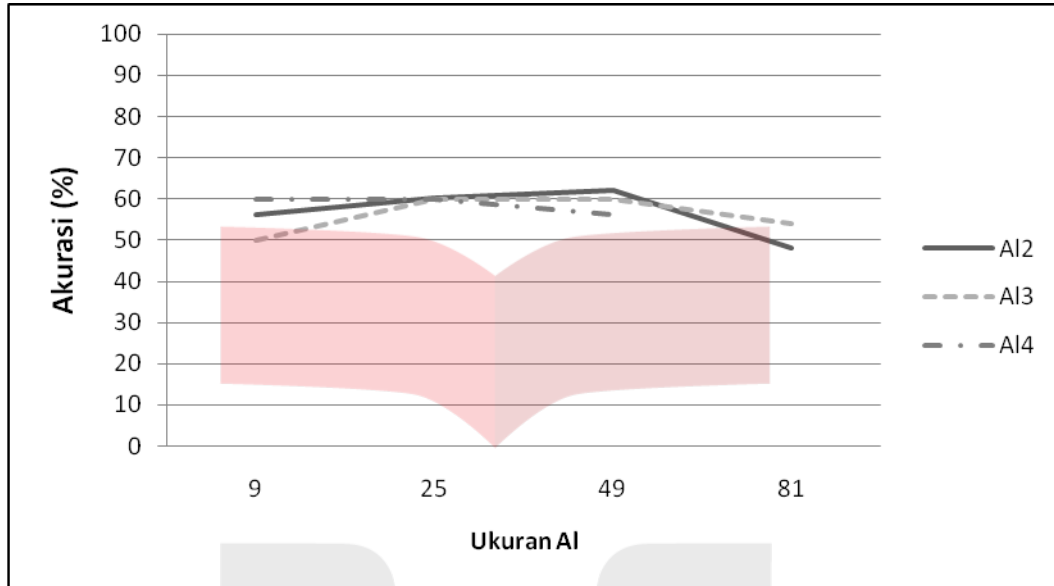
Pada *neocognitron* terdapat dua jenis *connection area* yang berdasarkan jenis *cell* yang memiliki *connection area* tersebut yaitu  $A_l$  yang dimiliki oleh *S-cell* dan *V-cell*,  $D_l$  yang dimiliki oleh *C-cell*. Berdasarkan Gambar 4-1 terlihat bahwa perubahan nilai pada  $A_l$  di setiap *stage* nya memiliki pengaruh pada akurasi yang dihasilkan. Akurasi yang dimaksud di sini adalah nilai akurasi dengan menggunakan data *testing\_data\_for\_observation*. Namun, jika dibandingkan dengan pengaruh  $D_l$  terhadap akurasi yang dihasilkan pada Gambar 4-2, maka dapat dikatakan bahwa perubahan nilai pada  $D_l$  dapat menyebabkan perubahan nilai akurasi yang signifikan. Hal ini terjadi karena  $A_l$  sangat dipengaruhi oleh *training pattern* yang diberikan saat proses *training* berlangsung sehingga proses perubahan nilainya tidak berpengaruh secara signifikan sedangkan  $D_l$  tidak dipengaruhi secara langsung oleh *training pattern*.

Meskipun demikian, di samping pengaruh *training pattern* pada lampiran A, nilai  $A_l$  di setiap *stage* juga dipengaruhi oleh deformasi setiap *feature – feature local* yang diekstrak di setiap *stage*. Jika deformasinya semakin besar maka  $A_l$  berukuran besar dan sebaliknya.

Secara empiris, pengaruh  $A_l$  terhadap akurasi terlihat jelas pada persamaan 2.1 yang berhubungan dengan besar perubahan nilai bobot ' $a$ '. Di samping itu, persamaan 2.6 menunjukkan peranan  $A_l$  pada nilai *output S-cell* ( $U_{Sl}$ ) sedangkan persamaan 2.8 menunjukkan peranan  $A_l$  pada nilai *output V-cell* ( $U_{Vl}$ ). Adapun

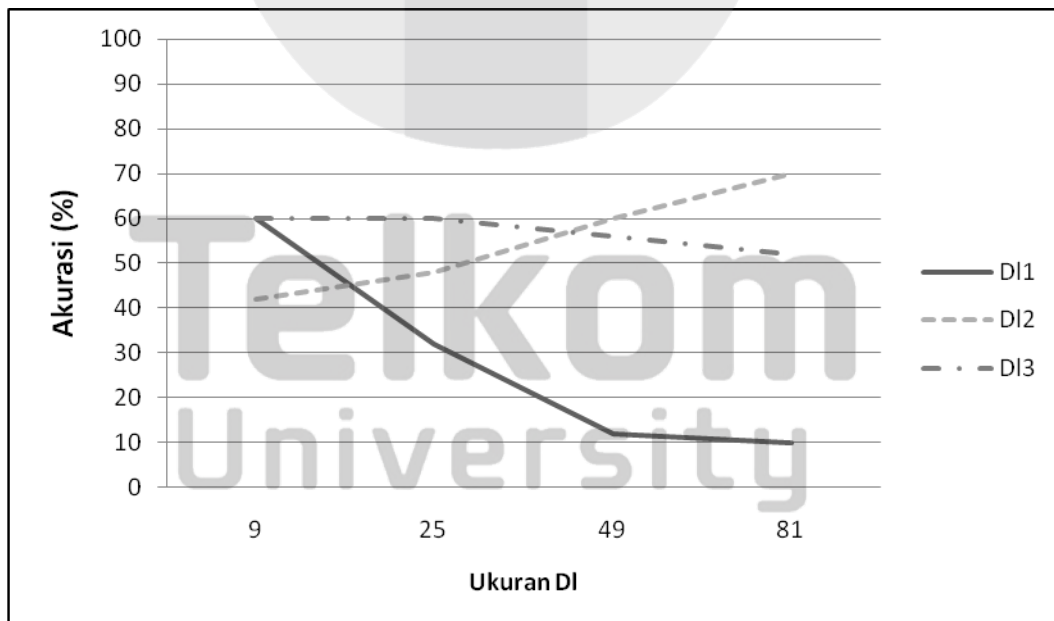


pengaruh  $D_i$  secara empiris terhadap akurasi dapat terlihat jelas pada saat penentuan *bobot* 'd' yang berpengaruh besar pada nilai *output C-cell* ( $U_{Ci}$ ).



Gambar 4-1 : Pengaruh  $A_i$  terhadap Akurasi Sistem

Deformasi *local – local feature* juga memiliki pengaruh yang sama terhadap  $D_i$ . Hal ini dapat dipahami jika dihubungkan dengan fungsi *C-cell* yang memiliki peranan sebagai *cell* yang melakukan toleransi terhadap deformasi maupun pergeseran *local feature* pada *S-cell* pada *cell plane* tertentu yang terhubung dengan *C-cell* yang bersangkutan.



Gambar 4-2 : Pengaruh  $D_i$  terhadap Akurasi Sistem.

Gambar 4-1 menunjukkan bahwa untuk setiap nilai  $A_i$  di setiap *stage* nya kecuali nilai  $A_i$  yang merupakan ukuran *connection area* untuk *S-cell* di *stage* 1

yang dipengaruhi langsung oleh *training pattern* pada  $U_{S1}$  (lampiran A.1), diperoleh nilai optimal yang menghasilkan akurasi tertinggi pada skenario perubahan nilai di setiap *stage* sesuai yang tertera pada Tabel 4-4.

Tabel 4-4 : Nilai  $A_i$  yang Optimal di Setiap Stage

Skenario Stage ke-	$A_1$	$A_2$	$A_3$	$A_4$	Akurasi (%)
2	9	49	25	25	62
3	9	25	25	25	60
4	9	25	25	25	60

Adapun Gambar 4-2 menunjukkan bahwa untuk setiap nilai  $D_i$  di setiap *stage* kecuali nilai  $D_4$  yang merupakan ukuran *connection area* untuk *C-cell* di *stage* 4 yang dipengaruhi langsung oleh *cell plane*  $U_{S4}$  yang besarnya adalah 9 (3x3), diperoleh nilai optimal yang menghasilkan akurasi tertinggi pada skenario perubahan nilai di setiap *stage* sesuai yang tertera pada Tabel 4-5.

Tabel 4-5 : Nilai  $D_i$  yang Optimal di Setiap Stage

Skenario Stage ke-	$D_1$	$D_2$	$D_3$	$D_4$	Akurasi (%)
1	9	49	25	9	60
2	9	81	25	9	70
3	9	49	25	9	60

Jika skenario untuk semua parameter  $A_i$  dan  $D_i$  digabungkan untuk setiap *stage* nya seperti yang ada pada Gambar 4-1 dan 4-2, maka dapat dikatakan bahwa nilai  $D_i$  yang optimal dicapai pada  $D_2$  dengan nilai sebesar 81 (9x9) yang menghasilkan akurasi 70% dimana nilai  $A_i$  untuk semua *stage* dan  $D_i$  untuk *stage* yang lain memiliki nilai standar.

Informasi lain yang dapat diperoleh dari Gambar 4-1 dan 4-2 serta dari Tabel 4-4 dan 4-5 adalah bahwa untuk setiap *stage*, nilai – nilai  $A_i$  dan  $D_i$  yang menghasilkan akurasi optimal berbeda – beda. Seperti yang telah dijelaskan sebelumnya, bahwa hal ini terjadi sebagai akibat perbedaan dari karakteristik *local – local feature* termasuk deformasi setiap *local feature* di setiap *stage*-nya.

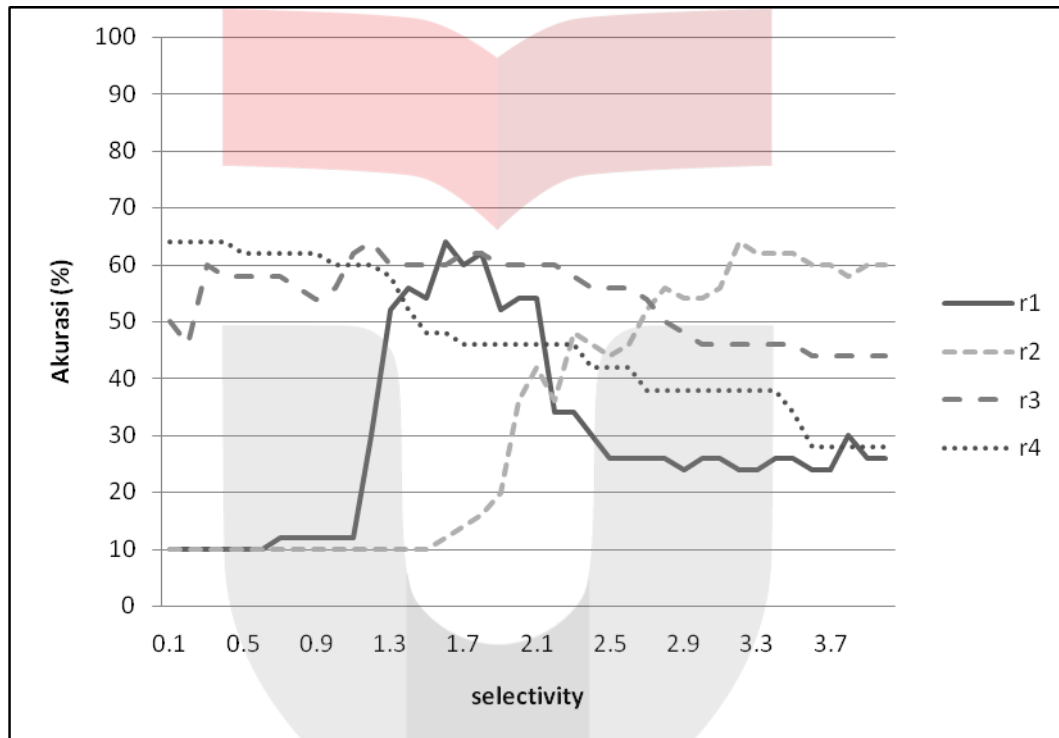
#### 4.2.2 Analisis Parameter Selectivity

Nilai parameter *selectivity* juga sangat dipengaruhi oleh karakteristik dari *training pattern* yaitu *local – local feature* yang diekstraksi oleh *S-layer* pada setiap *stage* nya. Oleh sebab itu, sama halnya dengan karakteristik nilai dari parameter  $A_i$  dan  $D_i$ , karakteristik nilai *selectivity* untuk setiap *stage* berbeda – beda.

Semakin besar nilai *selectivity*( $r$ ) maka semakin kecil toleransi terhadap *noise* dan deformasi serta pergeseran yang terjadi pada *local feature*. Sebagai contoh pada lampiran A.1 yang menunjukkan *training pattern* untuk setiap *cell plane* pada *S-layer* dalam *stage* 1, nilai optimal yang diperoleh berdasarkan Tabel 4-4 diperoleh nilai sebesar 1,6. Nilai tersebut tidak terlalu besar (dibandingkan dengan maksimum nilai pada skenario yang besarnya 4,0) karena *S-cell* pada *cell plane* 1 yang dilatih dengan *training pattern*  $U_{S1.0}$  pada lampiran A.1, tentu

diharapkan tetap memberikan respon terhadap *local feature* dalam *training pattern*  $U_{S1.1}$  dan  $U_{S1.2}$  serta  $U_{S1.10}$  dan  $U_{S1.11}$  pada lampiran A.1, meskipun nilai responnya lebih rendah jika dibandingkan dengan *feature* dalam *training pattern*  $U_{S1.0}$  pada lampiran A.1. Secara empiris, pengaruh nilai *selectivity* terhadap akurasi dapat dilihat pada persamaan 2.6 dimana nilai *selectivity* mempengaruhi nilai *output* sebuah *S-cell*( $U_{Si}$ ).

Berdasarkan Gambar 4-3, diketahui bahwa perubahan nilai *selectivity* juga sangat berpengaruh terhadap perubahan nilai akurasi yang dihasilkan.



Gambar 4-3 : Pengaruh *Selectivity* terhadap Akurasi Sistem

Gambar 4-3 menunjukkan bahwa untuk setiap nilai  $r_i$  di setiap *stage*, diperoleh nilai optimal yang menghasilkan akurasi tertinggi pada skenario perubahan nilai di setiap *stage* sesuai yang tertera pada Tabel 4-4.

Tabel 4-6 : Nilai  $r_i$  yang Optimal di Setiap *Stage*

Skenario <i>Stage</i> ke-	$r_1$	$r_2$	$r_3$	$r_4$	Akurasi (%)
1	1,6	<b>4,0</b>	<b>1,5</b>	<b>1,0</b>	64
2	<b>1,7</b>	3,2	<b>1,5</b>	<b>1,0</b>	64
3	<b>1,7</b>	<b>4,0</b>	1,2	<b>1,0</b>	64
4	<b>1,7</b>	<b>4,0</b>	<b>1,5</b>	0,1	64

Jika skenario untuk semua parameter  $r_i$  untuk semua *stage* digabungkan pada Tabel 4-4, maka dapat dikatakan bahwa nilai  $r_i$  yang optimal menghasilkan akurasi 64% untuk keempat skenario tersebut.

### 4.2.3 Analisis Parameter *Fixed Weight*

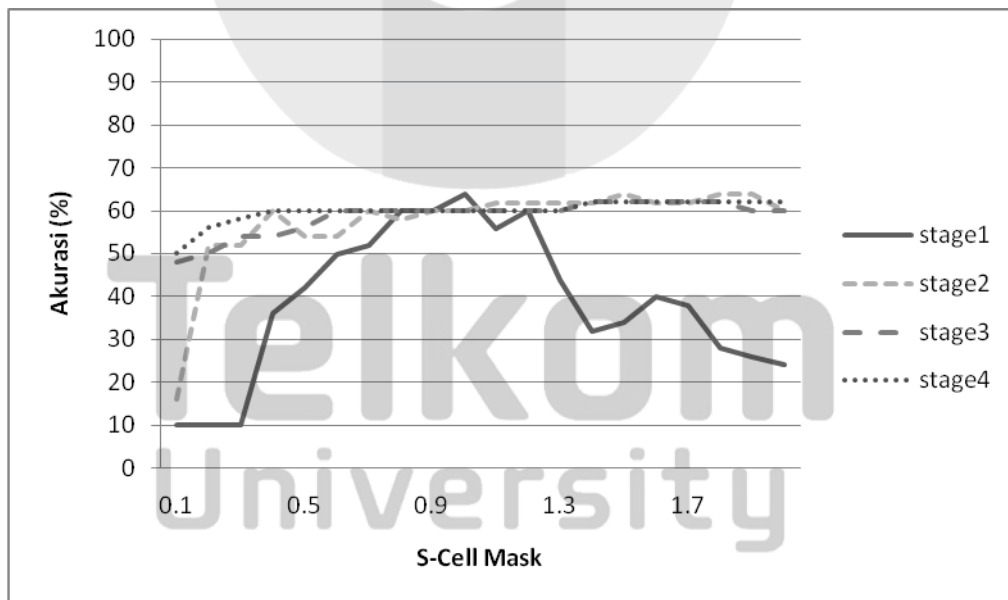
Parameter *fixed weight* yaitu bobot 'c' (*c-weights*) dan bobot 'd' (*d-weights*) dipengaruhi oleh variabel *S-cell mask* ( $\gamma_1$ ) untuk bobot 'c', *C-cell mask1* ( $\delta_1$ ) dan *C-cell mask2* ( $\delta_2$ ) untuk bobot 'd'. Berdasarkan Gambar 4-4, 4-5, dan 4-6 diketahui bahwa *fixed weight* dalam hal ini bobot 'c' dan bobot 'd' berpengaruh terhadap akurasi *neocognitron* yang dihasilkan. Secara empiris, hal ini terkait dengan peranan bobot 'c' dalam menentukan nilai *output* yang dimiliki oleh sebuah *V-cell* yang dapat dilihat pada persamaan 2.8 dan bobot 'd' dalam menentukan nilai *output* yang dimiliki oleh sebuah *C-cell* yang dapat dilihat pada persamaan 2.9.

Di samping itu, bobot 'c' juga menentukan besar perubahan yang terjadi pada bobot 'a' selama proses *training* seperti yang terlihat pada persamaan 2.1. Secara tidak langsung juga, pengaruh bobot 'c' dapat terlihat pada persamaan 2.2 dimana nilai  $U_{VI}$  yang dipengaruhi oleh bobot 'c' juga menentukan besar perubahan nilai bobot 'b' pada proses *training*.

Adapun peranan bobot 'd' secara tidak langsung adalah pada persamaan 2.1 dimana  $U_{CI}$  yang dipengaruhi oleh bobot 'd' menentukan besar perubahan nilai pada bobot 'a' saat proses *training*.

#### 4.2.3.1 Bobot 'c'

Berdasarkan Gambar 4-4, diketahui bahwa perubahan nilai *S-cell Mask* juga sangat berpengaruh terhadap perubahan nilai akurasi yang dihasilkan. Hal ini diakibatkan oleh parameter ini menentukan nilai bobot 'c' yang merupakan salah satu bobot tetap (*fixed weight*).



Gambar 4-4 : Pengaruh *S-cell Mask* terhadap Akurasi Sistem

Gambar 4-4 menunjukkan bahwa untuk setiap nilai  $\gamma_1$  di setiap *stage*, diperoleh nilai optimal yang menghasilkan akurasi tertinggi pada skenario perubahan nilai di setiap *stage* sesuai yang tertera pada Tabel 4-7.

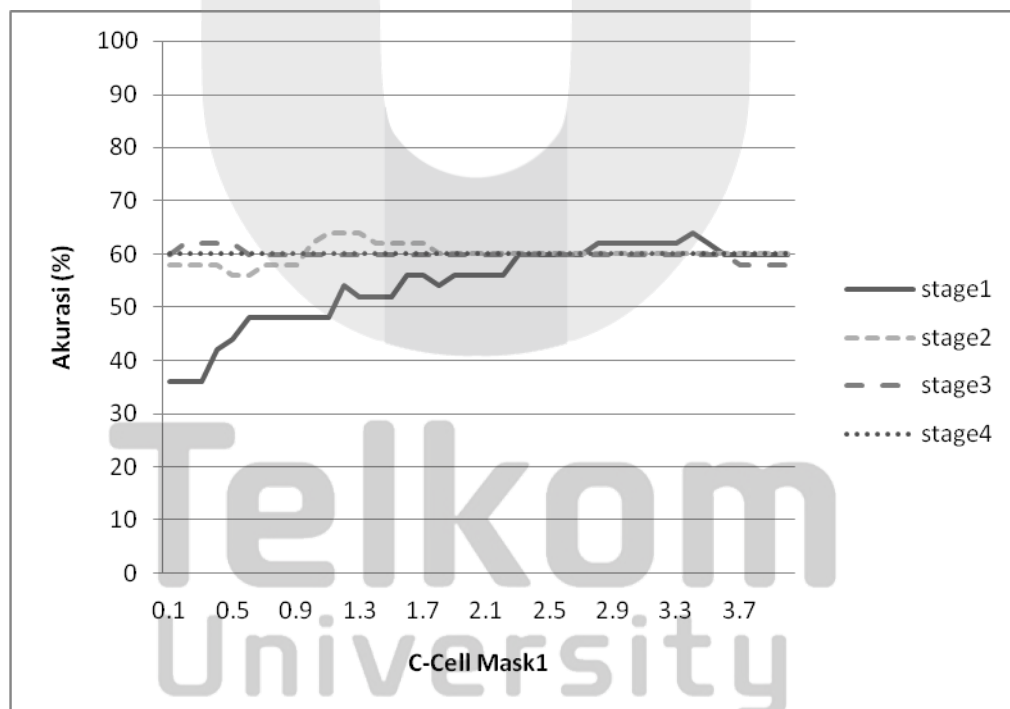
Tabel 4-7 : Nilai  $\gamma_i$  yang Optimal di Setiap Stage

Skenario Stage ke-	$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	Akurasi (%)
1	1,2	0,9	0,9	0,8	60
2	0,9	1,8	0,9	0,8	64
3	0,9	0,9	1,8	0,8	62
4	0,9	0,9	0,9	1,8	62

Jika skenario untuk semua parameter  $\gamma_i$  untuk semua stage digabungkan pada Tabel 4-7, maka dapat dikatakan bahwa nilai  $\gamma_i$  yang optimal dicapai pada  $\gamma_2$  dengan nilai sebesar 1,8 yang menghasilkan akurasi 64% dimana nilai  $\gamma_i$  untuk stage yang lain memiliki nilai standar.

#### 4.2.3.2 Bobot 'd'

Berdasarkan Gambar 4-5 dan 4-6, diketahui bahwa perubahan nilai *C-cell Mask1* dan *C-cell Mask2* juga sangat berpengaruh terhadap perubahan nilai akurasi dari *neocognitron* yang dihasilkan. Hal ini diakibatkan oleh kedua parameter ini menentukan nilai bobot 'd' yang merupakan salah satu bobot tetap (*fixed weight*).



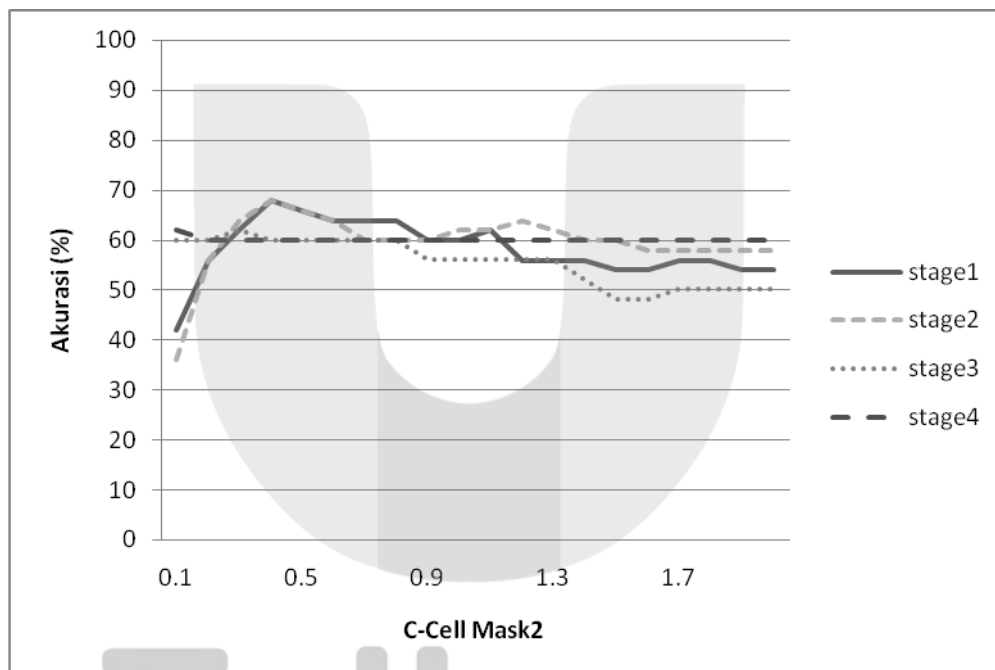
Gambar 4-5 : Pengaruh *C-cell Mask1* terhadap Akurasi Sistem

Gambar 4-5 menunjukkan bahwa untuk setiap nilai  $\delta_i$  di setiap stage, diperoleh nilai optimal yang menghasilkan akurasi tertinggi pada skenario perubahan nilai di setiap stage sesuai yang tertera pada Tabel 4-8.

Tabel 4-8 : Nilai  $\gamma_i$  yang Optimal di Setiap Stage

Skenario Stage ke-	$\hat{\delta}_1$	$\hat{\delta}_2$	$\hat{\delta}_3$	$\hat{\delta}_4$	Akurasi (%)
1	3,4	4,0	2,5	1,0	64
2	4,0	1,1	1,5	1,0	64
3	4,0	4,0	0,2	1,0	62
4	4,0	4,0	2,5	1,0	50

Jika skenario untuk semua parameter  $\hat{\delta}_i$  untuk semua stage digabungkan pada Tabel 4-8, maka dapat dikatakan bahwa nilai  $\gamma_i$  yang optimal dicapai pada  $\hat{\delta}_1$  dengan nilai sebesar 3,4 dimana nilai  $\hat{\delta}_i$  untuk stage yang lain memiliki nilai standar dan  $\hat{\delta}_2$  dengan nilai sebesar 1,1 yang menghasilkan akurasi 64% dimana nilai  $\hat{\delta}_i$  untuk stage yang lain memiliki nilai standar.



Gambar 4-6 : Pengaruh C-cell Mask2 terhadap Akurasi Sistem

Gambar 4-6 menunjukkan bahwa untuk setiap nilai  $\delta_i$  di setiap stage, diperoleh nilai optimal yang menghasilkan akurasi tertinggi pada skenario perubahan nilai di setiap stage sesuai yang tertera pada Tabel 4-9.

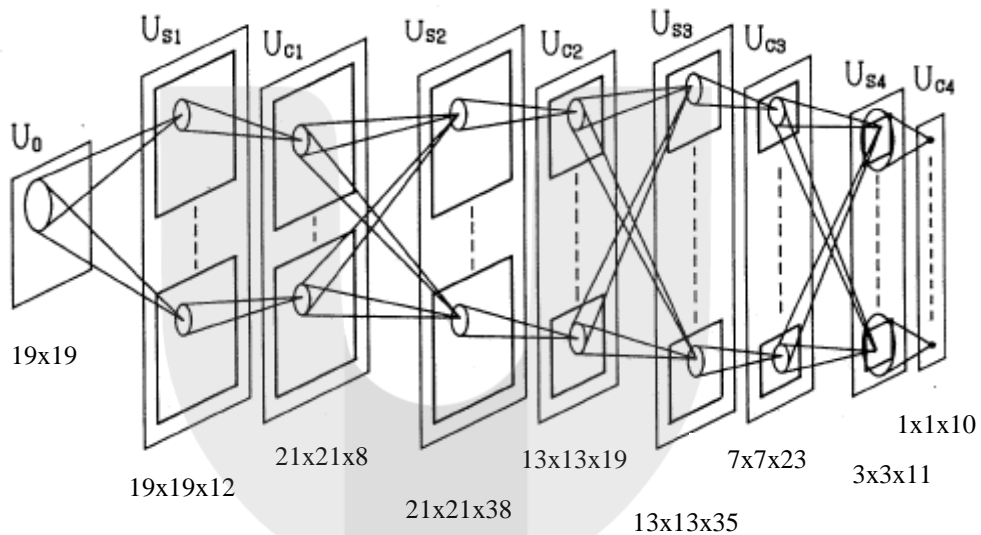
Tabel 4-9 : Nilai  $\delta_i$  yang Optimal di Setiap Stage

Skenario Stage ke-	$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	Akurasi (%)
1	0,4	0,8	0,7	1,0	68
2	0,9	0,4	0,7	1,0	68
3	0,9	0,8	0,3	1,0	62
4	0,9	0,8	0,7	0,1	62

Jika skenario untuk semua parameter  $\delta_i$  untuk semua *stage* digabungkan pada Tabel 4-8, maka dapat dikatakan bahwa nilai  $\gamma_i$  yang optimal dicapai pada  $\delta_1$  dengan nilai sebesar 0,4 dimana nilai  $\delta_i$  untuk *stage* yang lain memiliki nilai standar. dan  $\delta_2$  dengan nilai sebesar 0,4 yang menghasilkan akurasi 68% dimana nilai  $\delta_i$  untuk *stage* yang lain memiliki nilai standar.

#### 4.2.4 Analisis Neocognitron yang Dibangun

Seperti yang telah dikemukakan sebelumnya, bentuk arsitektur *neocognitron* yang diperoleh tentunya sangat dipengaruhi oleh *training pattern* yang disediakan pada lampiran A untuk proses *training neocognitron* karena jenis *neocognitron* yang dipakai pada tugas akhir ini adalah *supervised neocognitron*.



Gambar 4-7 : Arsitektur Neocognitron yang Digunakan [3]

Parameter struktural yang dipengaruhi secara langsung oleh *training pattern* adalah ukuran dan jumlah *cell plane* serta *connection area* pada *S-layer* dan *C-layer*, nilai keterhubungan antar *C-cell plane* dan *S-cell plane*, dan ukuran *input layer*. Adapun parameter non-struktural seperti *learning coefficient*, *S-cell* dan *C-cell mask* (*C-cell mask1* dan *C-cell mask2*) untuk mendapatkan nilai *fixed weight* mengikuti model *supervised neocognitron* yang telah dibuat oleh Fukushima [4,8].

Namun, khusus untuk beberapa nilai *connection area* pada *S-cell* ( $A_i$ ) dan *C-cell* ( $D_i$ ) serta nilai *selectivity* ( $r$ ) diperoleh dengan percobaan dengan memilih nilai akurasi yang tertinggi dari ketiga grafik dalam Gambar 4-1, 4-2, dan 4-3. Tabel 4-4 dan 4-5 menunjukkan daftar semua nilai parameter struktural dan non-struktural dari jaringan syaraf tiruan *neocognitron* yang dibangun.

Nilai yang diperoleh tentunya tidak jauh berbeda dengan yang diperoleh pada model *supervised neocognitron* Fukushima [4,8] karena nilai – nilai tersebut merupakan acuan nilai standar yang digunakan pada percobaan dalam tugas akhir

ini dimana tentunya Fukushima memperoleh nilai – nilai tersebut dengan ruang percobaan yang sangat besar yang memakan waktu yang sangat lama.

Tabel 4-10 : Parameter Struktural dari *Neocognitron* yang Dibangun

Layer	Jumlah cell plane	Ukuran cell plane	Ukuran connection area
<i>Input(U<sub>0</sub>)</i>	1	19 x 19	-
U <sub>S1</sub>	12	19 x 19	3 x 3
U <sub>C1</sub>	8	21 x 21	3 x 3
U <sub>S2</sub>	38	21 x 21	5 x 5
U <sub>C2</sub>	19	13 x 13	9 x 9
U <sub>S3</sub>	35	13 x 13	5 x 5
U <sub>C3</sub>	23	7 x 7	5 x 5
U <sub>S4</sub>	11	3 x 3	5 x 5
U <sub>C4</sub>	10	1 x 1	3 x 3

Tabel 4-11 : Parameter *Non-Struktural* dari *Neocognitron* yang Dibangun

Parameter	Stage 1	Stage 2	Stage 3	Stage 4
<i>S-cell Mask (γ<sub>i</sub>)</i>	0,9	0,9	0,9	0,8
<i>C-cell Mask 1 (^δ<sub>i</sub>)</i>	4,0	4,0	2,5	1,0
<i>C-cell Mask 2 (δ<sub>i</sub>)</i>	0,9	0,8	0,7	1,0
<i>Learning coefficient (q<sub>i</sub>)</i>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>	10 <sup>4</sup>
<i>Selectivity (r<sub>i</sub>)</i>	1,7	4,0	1,5	1,0

Hasil yang diperoleh pada Tabel 4-7 dan 4-8 merupakan *neocognitron* yang memiliki akurasi maksimum yaitu sebesar 70% saat proses *testing\_for\_observation* dengan menggunakan data *testing\_data\_for\_observation* dimana jumlah *output* jaringan yang benar = 35, salah = 10, dan tidak teridentifikasi = 5 (persamaan 4.1). Dengan kata lain, arsitektur tersebut diperoleh setelah membandingkan nilai akurasi yang diperoleh pada Tabel 4-4,4-5 dan 4-6 yang menunjukkan nilai akurasi optimal pada masing – masing skenario yaitu skenario *connection area A<sub>i</sub>* di *stage 2,3,4* dan *D<sub>i</sub>* di *stage 1,2,3* serta *selectivity* di setiap *stage*.

#### 4.2.5 Analisis Performansi *Neocognitron* dalam Identifikasi *Image* Karakter Numerik

Berdasarkan hasil observasi, diperoleh jaringan *neocognitron* yang optimal seperti yang tertera pada Tabel 4-7 dan 4-8. Jaringan *neocognitron* ini selanjutnya melalui proses *testing* dengan menggunakan data *testing\_data*. Berdasarkan proses *testing* ini diperoleh akurasi jaringan *neocognitron* tersebut sebesar 74.05% dengan menggunakan persamaan 4.1 dimana jumlah *output* jaringan yang benar = 4443, salah = 458, dan tidak teridentifikasi = 1099.

Sesuai dengan *training pattern* yang digunakan pada lampiran A yang mempengaruhi bentuk arsitektur *neocognitron* yang dibangun, cuma dibutuhkan 4 *stage* dan hanya dibutuhkan *local – local feature* yang mungkin dimiliki oleh semua karakter numerik untuk mengenali karakter numerik meskipun telah



mengalami deformasi, *noise*, maupun pergeseran. Di samping itu, *neocognitron* yang dibangun juga telah mampu untuk menolak karakter yang tidak termasuk karakter numerik.

Meskipun nilai akurasi yang dihasilkan dari proses *testing* hanya sebesar 80,025%, hal tersebut masih dapat ditoleransi karena *pattern* yang disediakan pada saat proses *testing* yaitu data *testing\_data* memang memiliki tingkat *noise*, deformasi, dan pergeseran yang cukup tinggi. Di samping itu, jumlah data *testing\_data* yang sangat besar yaitu 6000 pola data dimana 600 pola untuk setiap kategorinya jika dibandingkan dengan jumlah data *learning training\_data* dan *testing\_data\_for\_observation* yang jauh lebih sedikit juga dapat menggambarkan kehandalan dari *neocognitron* dalam hal identifikasi *image* karakter numerik.

Dengan banyaknya kemungkinan bentuk *pattern* dari suatu karakter numerik ditambah dengan karakter *non* numerik, hal di atas merupakan salah satu bukti bahwa *neocognitron* sudah layak dikatakan sebagai suatu sistem yang mampu melakukan identifikasi terhadap karakter numerik, bahkan jika karakter numerik tersebut telah mengalami *noise*, deformasi, dan pergeseran.



## Daftar Pustaka

- [1] Brown, Eric . Character Recognition by Feature Point Extraction. <http://www.ccs.neu.edu/home/feneric/charrec.html>. Diakses pada tanggal 30 April 2009.
- [2] Fukushima. 2007. Neocognitron. Scholarpedia,2(1):1717. <http://www.scholarpedia.org/article/Neocognitron>. Diakses pada tanggal 15 April 2009.
- [3] Fukushima: "Neocognitron: A hierarchical neural network capable of visual pattern recognition", *Neural Networks*, 1[9], pp. 119-130 (1988).
- [4] Fukushima: "Neocognitron for handwritten karakter numerik recognition", *Neurocomputing*, 51, pp. 161-180 (April 2003).
- [5] Fukushima, K., Neocognitron capable of incremental learning. *Portal ACM*, Volume 17 , Issue 1, pp. 37-46 (January 2004).
- [6] Fukushima, K., & Miyake, S. 1982. Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6), 455-469.
- [7] Fukushima, K., & Wake, N. 1991. Handwritten Alphanumeric Character Recognition by the Neocognitron. Reprinted from *IEEE TRANSACTION ON NEURAL NETWORKS* Vol.2, No.3, May 1991.
- [8] Korjoukuv, Iliia. 2004. A Cognitive Model of Human Drawing. Master of Science Thesis University of Amsterdam.
- [9] Kusmayati, D. 1998. Neocognitron. Buku Tugas Akhir Sekolah Tinggi Teknologi Surabaya.
- [10] Lovel, David. 1994. The Neocognitron As a System For Handwritten Character Recognition: Limitations and Improvements. Dissertation book, Department Electrical and Computer Engineering University of Queensland.
- [11] Lovel, David et al. An Evaluation of The Neocognitron. <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=BB392638BCD1454C790343F6C435BD55?doi=10.1.1.51.9257&rep=rep1&type=pdf>.
- [12] Lovel, David et al. Improving The Performance of The Neocognitron. Department of Electrical and Computer Engineering University of Queensland.
- [13] Minarik, Gabriel . 2008. Neocognitron v1.0.0 BETA. <http://neocognitron.euweb.cz>. Diakses pada tanggal 10 April 2009.
- [14] Neocognitron Neural Network. <http://www.learnartificialneuralnetworks.com/neocognitron-neural-network.html>. Diakses pada tanggal 24 April 2009.
- [15] Ooyen, van et al. 1993. Pattern Recognition in the *Neocognitron* is Improved by Neuronal Adaptation. *Biol. Cybern*, 70, pp. 47-53 (1993).
- [16] Shi, Daming et al. 1999. Neocognitron's parameter tuning by genetic algorithms. *International Journal of Neural Systems*, Vol. 9, No. 6 (December, 1999) 497-509.
- [17] Suyanto. 2007. Artificial Intelligence, Searching, Reasoning, Planning, and Learning. Bandung: Informatika.

- [18] Tran, Michael et al. 2000. Neocognitron for Rotated Pattern Recognition. <http://www.csse.monash.edu.au/~rdp/research/Presentations/cata-2000a.ppt>. Diakses pada tanggal 29 April 2009.
- [19] Velínský, Tomáš. 2000. Neocognitron Tutorial. <http://neuron.felk.cvut.cz/courseware/data/chapter/velinsky2000/en/>. Diakses pada tanggal 20 April 2009.

