

# PENERAPAN KENDALI JARINGAN NEURAL PADA SISTEM GERAKAN LURUS PRESISI

Ilham Budi Santoso<sup>(1)</sup> dan Zainal Abidin<sup>(2)</sup>

<sup>(1)</sup>Mahasiswa Pasca Sarjana Teknik Sistem Mekanika Jurusan Teknik Mesin ITB  
Jl. Ganesha 10 Bandung

<sup>(2)</sup>Jurusan Teknik Mesin ITB, Jl. Ganesha 10 Bandung

## Ringkasan

*Makalah ini menguraikan tentang penerapan jaringan neural sebagai pengendali adaptif pada sistem gerakan lurus presisi. Sebagai pengendali adaptif pada sistem gerakan lurus presisi jaringan neural belajar dalam dua tahap. Tahap pertama adalah pembelajaran secara general di mana jaringan neural melakukan identifikasi terhadap fungsi invers sistem. Pada proses belajar tahap kedua jaringan neural berfungsi sebagai pengendali adaptif yang belajar secara online. Tahap ini dikenal sebagai pembelajaran secara khusus. Melalui dua tahap proses belajar tersebut terlihat bahwa pengendali mampu bekerja pada daerah operasi yang cukup lebar dengan intervensi operator yang cukup kecil tanpa harus mengorbankan prestasi kendalinya.*

**Kata kunci :** mekanisme gerakan lurus presisi, jaringan neural, pengendali invers, pembelajaran general, pembelajaran khusus, output matching.

## Abstract

*This paper presents an empirical result on the application of neural network controller in a precision straight motion system. As a controller, neural networks learn in two phase. In the first learning phase, neural network was doing inverse system identification. It is called general learning. In the second learning phase, neural network was applied as an adaptif controller which doing online learning. It is called special learning. Through these two learning phases, neural network shows a good performance for wide range operation with a minimum operator/human intervention.*

**Keywords:** Straight motion precision mechanism, neural networks, invers controller, general learning, special learning, output matching.

## 1. Pendahuluan

Pemosisian pada mekanisme gerakan presisi adalah salah satu aspek terpenting pada penelitian mesin-mesin presisi. Tingkat ketelitian pada pemosisian presisi ditentukan oleh dua hal yaitu pengendali dan ketelitian geometrik dari komponen-komponen mekanik. Tingkat ketelitian geometrik komponen mekanik dibatasi oleh kemampuan proses produksi. Selain itu, tingkat ketelitian geometrik ini dapat turun akibat distorsi geometrik akibat gaya pengencangan saat perakitan komponen mekanik. Akibat keterbatasan ketelitian geometrik tersebut pengendali memiliki peran yang sangat penting dalam mengkompensasi ketidaktelitian tersebut, sehingga prestasi pemosisian dapat dipertahankan. Selain itu, pengendali harus mampu pula mengatasi berbagai fenomena tak linier seperti *stick slip* dan *backlash*.

Penerapan pengendali klasik seperti PID (*Proportional Integral Derivative*) pada mekanisme gerakan lurus presisi menunjukkan bahwa prestasi pemosisian yang baik dan tegar

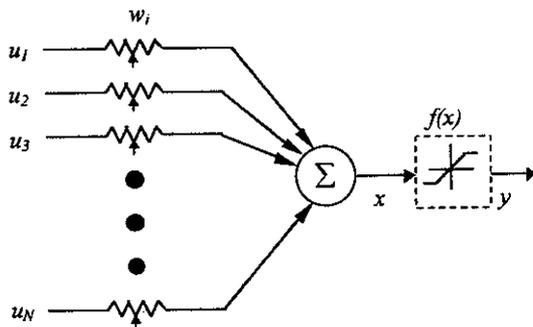
hanya dapat diperoleh pada rentang operasi yang cukup kecil [1],[2]. Agar prestasi pemosisian dapat dipertahankan pada daerah operasi yang cukup lebar diperlukan suatu pengendali yang mampu beradaptasi terhadap berbagai ketidakpastian akibat ketidaklinieran perilaku dinamik mekanisme gerakan lurus presisi.

Sementara itu, banyak penelitian yang melaporkan keberhasilan jaringan neural tiruan (*Artificial Neural Networks*) sebagai pengendali pada sistem-sistem tak linier seperti pada sistem tangki ganda [3], sistem kendali pada truck [4], sistem kendali pada gantry crane [5], dan lain-lain. Selain itu, pengendali jaringan neural juga tegar terhadap gangguan derau [6]. Dengan melihat kemampuan jaringan neural tersebut, pada penelitian ini jaringan neural digunakan sebagai pengendali pada mekanisme gerakan lurus presisi. Penerapan jaringan neural sebagai pengendali adaptif pada mekanisme gerakan presisi diharapkan dapat memperlebar daerah operasi pemosisian presisi dengan sesedikit mungkin intervensi dari operator tanpa harus mengorbankan prestasi pemosisiannya.

## 2. Jaringan Neural Umpan Maju dengan Metode Belajar Rambat Mundur

### 2.1 Jaringan Neural Umpan Maju [7,8,9]

Jaringan neural merupakan suatu jaringan yang tersusun atas unit pengolah data. Unit pengolah data, yang biasa disebut dengan nodal, merupakan model fungsional dari sel *neuron* pada sistem otak manusia atau sistem otak *vertebrata* pada umumnya. Skematik dari model fungsional *neuron* tersebut ditunjukkan pada Gambar 1.



Gambar 1. Nodal yang merupakan model fungsional dari sel neural

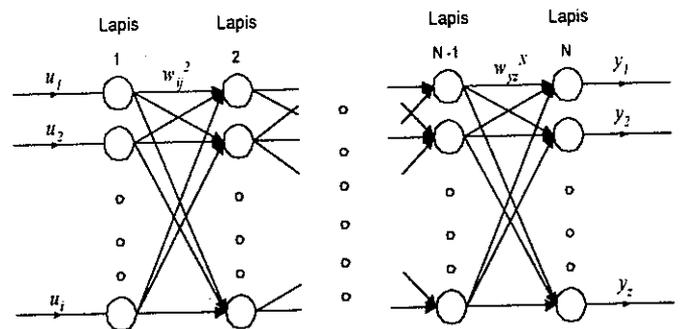
Struktur model tersebut dikembangkan oleh McCulloch dan Pitts. Sedangkan kemampuan model untuk beradaptasi didasarkan metode belajar Hebbian.

Unit pengolah atau nodal memiliki sejumlah masukan dan memiliki keluaran tunggal. Semua masukan pada nodal dikalikan dengan suatu nilai pembobot, dimana nilai pembobot tersebut dapat berubah secara adaptif. Hasil perkalian tersebut kemudian dijumlahkan dan dimasukkan dalam suatu fungsi aktivasi tertentu. Secara matematik unit pengolah data dapat dinyatakan sebagai persamaan berikut :

$$y = f(x) = f\left(\sum_{i=1}^N w_i u_i\right)$$

dimana  $f(\cdot)$ , menyatakan fungsi aktivasi nodal,  $u_i$  untuk  $i=1..N$ , adalah masukan nodal,  $w_i$  adalah nilai pembobot untuk  $i=1..N$ , dan  $y$  adalah keluaran sistem.

Pada arsitektur jaringan umpan maju (*feedforward networks*), nodal disusun menjadi beberapa lapis. Setiap nodal pada suatu lapis terhubung dengan setiap nodal pada lapis di depannya, sedangkan nodal-nodal yang terletak pada satu lapis tidak saling terhubung. Skema dari arsitektur jaringan neural tiruan umpan maju ini ditunjukkan pada Gambar 2. Pada gambar tersebut Lapisan nodal pada arsitektur jaringan umpan maju dapat dikelompokkan menjadi tiga macam lapisan.



Gambar 2 Skema dari jaringan neural umpan maju yang tersusun atas beberapa lapis

Pertama adalah lapis masukan yang merupakan lapisan terluar (Lapis 1) dari jaringan neural. yang Lapisan ini merupakan penghubung antara jaringan neural dengan masukan dari luar. Keluaran dari lapis ini terhubung dengan semua nodal pada lapisan berikutnya (Lapis 2) melalui nilai pembobot tertentu.

Kelompok kedua adalah lapisan keluaran (*output layer*). Lapisan ini adalah lapisan terluar (Lapis  $N$ ) dimana jaringan memberikan keluarannya kepada lingkungan.

Kelompok ketiga adalah lapisan tersembunyi (*hidden layer*). Lapisan ini adalah lapisan yang terletak antara lapisan masukan dan lapisan keluaran. Lapisan tersembunyi dapat tersusun atas satu lapis atau lebih. Masukan pada setiap lapisan tersembunyi adalah keluaran dari lapisan sebelumnya dan keluaran setiap lapisan tersembunyi merupakan masukan bagi lapisan di depannya.

Secara matematis keluaran dari jaringan neural tiruan umpan maju pada Gambar 2 di atas dapat dinyatakan sebagai persamaan berikut [7][8][9] :

$$y_z = f\left(\sum_1^y w_{yz}^{n-1} \cdot f_y\left(\dots\left(\sum_1^j w_{jk}^2 \cdot f_j\left(\sum_1^i w_{ij}^1 \cdot u_i\right)\right)\dots\right)\right)$$

Agar hubungan antara masukan jaringan,  $u_{ij}$  dan keluaran jaringan neural,  $y_z$ , dapat mengikuti suatu pola tertentu maka jaringan melakukan proses adaptasi terhadap nilai-nilai pembobot pada jaringan. Proses adaptasi ini biasa dikenal sebagai proses 'belajar' dari jaringan neural tiruan. Proses belajar tersebut mengikuti suatu algoritme tertentu. Salah satu algoritme belajar yang adalah algoritme belajar rambat mundur (*backpropagation*). Algoritme belajar tersebut banyak diterapkan pada jaringan neural karena kesederhanaan algoritmenya sehingga sangat memungkinkan untuk diterapkan pada sistem *real-time*.

## 2.2 Algoritme Belajar Rambat Mundur

Algoritme belajar rambat mundur membagi proses belajar jaringan menjadi tiga tahap. Tiga tahap tersebut dilakukan secara iteratif hingga jaringan menghasilkan perilaku yang diinginkan.

Tahap pertama adalah tahap umpan maju (*feedforward*), yaitu jaringan diberikan suatu masukan,  $U$  ( $u_1, u_2, \dots, u_i$ ), sehingga jaringan menghasilkan keluaran  $Y$  ( $y_1, y_2, \dots, y_z$ ), seperti yang telah diuraikan pada sub bab 2.1. Setelah keluaran jaringan diperoleh, tahap selanjutnya adalah tahap penentuan nilai galat (*error*),  $E$ . Galat tersebut diperoleh dengan cara membandingkan keluaran jaringan  $Y$  dengan nilai keluaran yang diinginkan  $Y_d$  ( $y_{d1}, y_{d2}, \dots, y_{dz}$ ), yaitu :

$$E(t) = Y(t) - Y_d(t)$$

Pada tahap ketiga dilakukan adaptasi bobot jaringan berdasarkan nilai galat yang telah diperoleh. Tujuan dari adaptasi bobot ini adalah memperkecil nilai galat pada iterasi berikutnya sehingga nilai galatnya pada suatu saat akan menuju ke nol. Secara matematik tujuan dari adaptasi bobot ini, yaitu :

$$\lim_{t \rightarrow \infty} E(t) = \lim_{t \rightarrow \infty} |Y(t) - Y_d(t)| = 0$$

Berdasarkan metode kuadrat terkecil (*least square*), dapat didefinisikan kriteria performansi dari jaringan adalah :

$$J = \frac{1}{2} \|E(t)\|^2$$

Semakin kecil nilai  $J$  maka performansi jaringan semakin baik.

Karena tujuan dari adaptasi bobot adalah minimisasi dari nilai galat total,  $E$ , maka besar perubahan bobot jaringan disesuaikan dengan besar sumbangan tiap-tiap nilai pembobot terhadap nilai galat total yang terjadi. Oleh karena itu perubahan tiap-tiap nilai pembobot pada jaringan,  $w_{ij}^N$ , sebanding dengan besar gradient pembobot terhadap performansi jaringan,  $J$ .

Gradient tersebut dinyatakan dalam persamaan berikut :

$$\frac{\partial J}{\partial w_{ij}^N} = E(t) \frac{\partial E(t)}{\partial w_{ij}^N}$$

dimana  $w_{ij}^N$  adalah nilai pembobot yang menghubungkan nodal ke- $i$  pada lapisan ke- $(N-1)$  ke nodal ke- $j$  pada lapisan ke- $N$ . Dengan menggunakan nilai gradient di atas maka nilai pembobot untuk iterasi berikutnya adalah [7][8][9] :

$$w_{ij}^N(t+1) = w_{ij}^N(t) + \Delta w_{ij}^N(t) = w_{ij}^N(t) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}^N(t)}$$

dengan  $\eta$  adalah konstanta pembelajaran yang besarnya adalah  $\{\eta \in [0,1] : \eta \in Real\}$ . Pada persamaan di atas nilai  $\Delta w_{ij}^N$  bernilai negatif karena tujuan dari proses belajar ini adalah minimisasi dari nilai  $E$ .

Pada beberapa kasus, penggunaan metode belajar di atas memerlukan waktu yang cukup lama (jumlah iterasi yang cukup besar) untuk mencapai nilai  $J$  minimum. Untuk mempercepat proses belajar itu ditambahkan momentum pada perubahan nilai pembobot. Suku momentum tersebut berupa  $\Delta w_{ij}^N$  dari iterasi sebelumnya. Sehingga besar bobot pada iterasi berikutnya menjadi [7][8][9] :

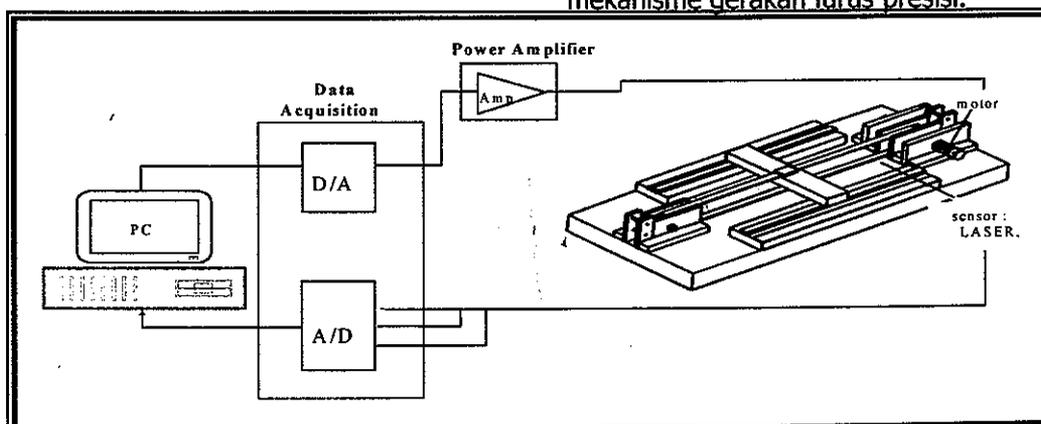
$$w_{ij}^N(t+1) = w_{ij}^N(t) + \Delta w_{ij}^N(t) + \alpha \cdot \Delta w_{ij}^N(t-1)$$

dimana  $\alpha$  adalah konstanta momentum yang besarnya  $\{\alpha \in [0,1] : \alpha \in Real\}$ .

## 3. Sistem Gerakan Lurus Presisi dan Beberapa Perilaku Dinamikanya

### 3.1 Sistem Gerakan Lurus Presisi

Secara skematik sistem gerakan lurus presisi ini ditunjukkan pada Gambar 3. Pada Gambar tersebut terlihat bahwa sistem gerakan lurus presisi terdiri dari aktuator, sensor posisi, kartu akuisisi data, penguat daya, komputer PC, dan mekanisme gerakan lurus presisi.



Gambar 3 Skema dari sistem gerakan lurus presisi

Aktuator yang digunakan pada sistem ini berupa motor DC dengan magnet permanen. Pemilihan motor DC ini bertujuan untuk memudahkan strategi pengendalian karena kecepatan putar motor DC sebanding dengan tegangan listrik yang diberikan. Motor DC ini dilengkapi dengan roda gigi reduksi untuk menurunkan putaran sekaligus untuk meningkatkan torsi. Sensor posisi yang digunakan pada sistem ini berupa LASER yang memiliki resolusi pembacaan sebesar  $0.125 \mu\text{m}$  dengan rentang pengukuran maksimum sebesar 10 mm. Kartu akuisisi data memiliki 16 kanal masukan ADC (*Analog to Digital Converter*) dan 2 kanal keluaran DAC (*Digital to Analog Converter*) masing-masing memiliki kecermatan 12 bits. Kartu ini terpasang pada ISA slot komputer PC dengan processor Intel 166 MHz dan RAM sebesar 32 Mbytes.

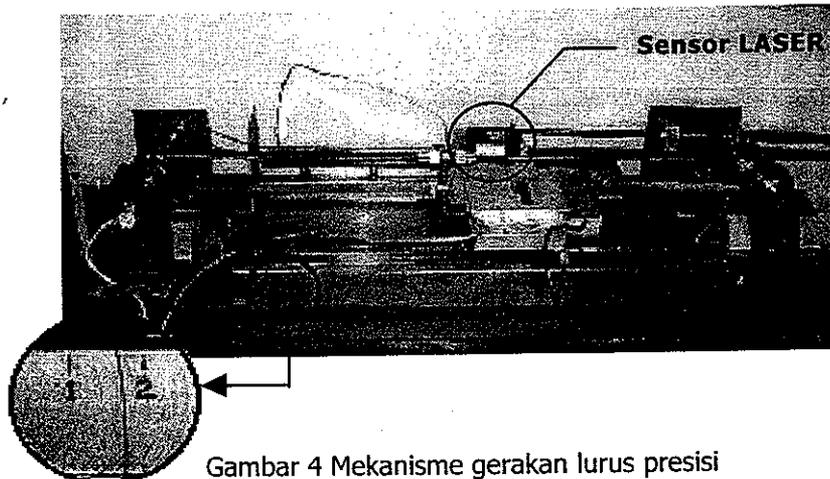
Mekanisme gerakan lurus presisi berupa sebuah meja yang dapat meluncur di atas sepasang *linear motion*. Gerakan meja tersebut dihasilkan oleh motor dan diteruskan oleh mekanisme sabuk dan puli. Adapun posisi meja diamati oleh sensor posisi. Informasi numerik yang dihasilkan oleh sensor posisi dikirimkan ke komputer melalui kartu akuisisi data sebagai informasi masukan bagi algoritme kendali. Aksi kendali yang dihasilkan oleh algoritme kendali selanjutnya dikirimkan ke aktuator setelah dikuatkan oleh penguat daya. Gambar 4 berikut menunjukkan mekanisme gerakan lurus presisi ini.

Pada Gambar 4 terlihat bahwa pada struktur pendukung *linear motion* terdapat skala pengukuran. Skala tersebut digunakan sebagai titik-titik referensi operasi pada saat pengujian prestasi pengendalian.

### 3.2 Beberapa perilaku tak linier pada sistem gerakan lurus presisi

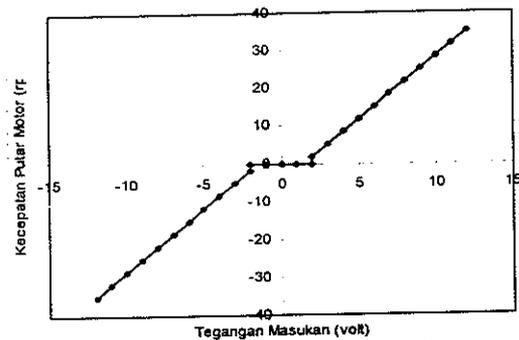
#### A. Dead zone pada motor penggerak

*Dead zone* terjadi akibat torsi motor belum cukup mampu untuk mengatasi gaya gesek statik pada stator.



Gambar 4 Mekanisme gerakan lurus presisi

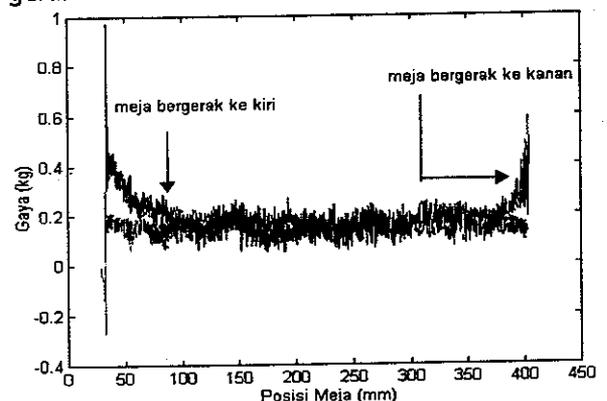
*Dead zone* ini terjadi pada daerah tegangan masukan  $\pm 0.15$  volt. Namun akibat adanya gaya gesek statik pada *linear motion*, *dead zone* motor ini diperlebar hingga  $\pm 1.75$  volt. Karakteristik kecepatan putar motor DC dalam keadaan terpasang pada mekanisme gerakan lurus presisi diperlihatkan pada Gambar 5.



Gambar 5. Karakteristik kecepatan motor terhadap tegangan masukan dalam keadaan terakit.

#### B. Variasi besar gaya gesek dinamik sepanjang *linear motion* [2]

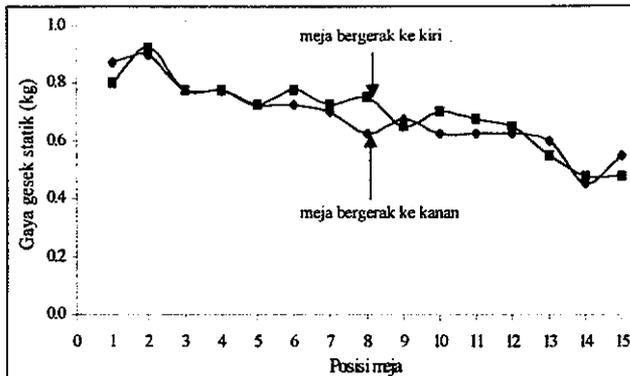
Penelitian yang dilakukan oleh Sibenordy [2] menunjukkan bahwa besar gaya gesek dinamik di sepanjang *linear motion* tidaklah konstan. Gambar 6 berikut menunjukkan variasi besar gaya gesek dinamik di sepanjang *linear motion* untuk arah gerakan ke kanan dan ke kiri.



Gambar 6 Variasi gaya gesek dinamik di sepanjang *linear motion* [2]

### C. Variasi besar gaya gesek dinamik sepanjang linear motion [2]

Sibenordy [2] juga mengemukakan bahwa gaya gesek statik di sepanjang linear motion tidaklah konstan. Fenomena ini ditunjukkan pada Gambar 7 berikut. Karena gaya gesek statik di sepanjang linear motion tidak konstan maka *dead zone* pada motor penggerak pun menjadi tidak konstan untuk posisi di sepanjang *linear motion*.



Gambar 7 Variasi gaya gesek statik di sepanjang linear motion [2]

Selain perilaku di atas, adapula perilaku yang belum diamati secara khusus namun sangat mempengaruhi prestasi pengendali yaitu sifat elastisitas sabuk.

### 4. Penerapan pengendali neural

Pada penelitian ini pengendali jaringan neural digunakan sebagai pengendali invers langsung. [8][9][10]. Pengendali invers menggunakan jaringan neural untuk mengidentifikasi fungsi yang merupakan invers dari sistem atur. Jika sistem atur didefinisikan sebagai fungsi  $f[.]$ , sebagai berikut:

$$y(k+1) = f[y(k), \dots, u(k), u(k-1), \dots],$$

dimana  $y(\cdot)$  dan  $u(\cdot)$  adalah keluaran dan masukan sistem atur maka fungsi pengendali invers,  $g[.]$ , didefinisikan sebagai berikut [10]:

$$r(k) = y(k+1) = f[y(k), \dots, g[r(k), y(k), \dots, u(k-1), \dots], u(k-1), \dots]$$

Untuk membuat pengendali tersebut dibangun jaringan neural dengan 1 buah lapis tersembunyi dengan konfigurasi 12 buah nodal pada lapis masukan, 6 nodal pada lapis tersembunyi, 1 nodal pada lapis keluaran. Fungsi aktivasi yang digunakan pada jaringan ini berupa fungsi sigmoid bipolar.

Untuk menerapkan pengendali jaringan neural ini dilakukan dua tahap pembelajaran tahap pertama adalah pembelajaran general dan tahap kedua adalah pembelajaran khusus [11]. Pada pembelajaran general jaringan belajar dilakukan

secara *off-line* guna melakukan identifikasi fungsi invers dari mekanisme gerakan presisi. Pada pembelajaran khusus jaringan neural berfungsi sebagai pengatur sambil terus beradaptasi untuk menghasilkan prestasi kendali terbaik

#### 4.1 Pembelajaran general

Tujuan pembelajaran ini adalah untuk mengidentifikasi fungsi invers dari mekanisme gerakan lurus presisi. Hasil identifikasi ini akan digunakan untuk inialisasi jaringan neural pada tahap belajar khusus. Dengan demikian, pada saat jaringan neural berfungsi sebagai pengendali, tidak diperlukan pembelajaran yang cukup berat dan osilasi yang berlebihan dapat dihindari. Osilasi yang berlebihan ini harus dihindari karena dapat berakibat gerakan meja keluar dari rentang pembacaan sensor posisi.

Kriteria performansi dari pembelajaran general ini dinyatakan dalam berikut :

$$J[w(k)] = 0.5(u(k) - u_{nn}(k))^2 \quad (1)$$

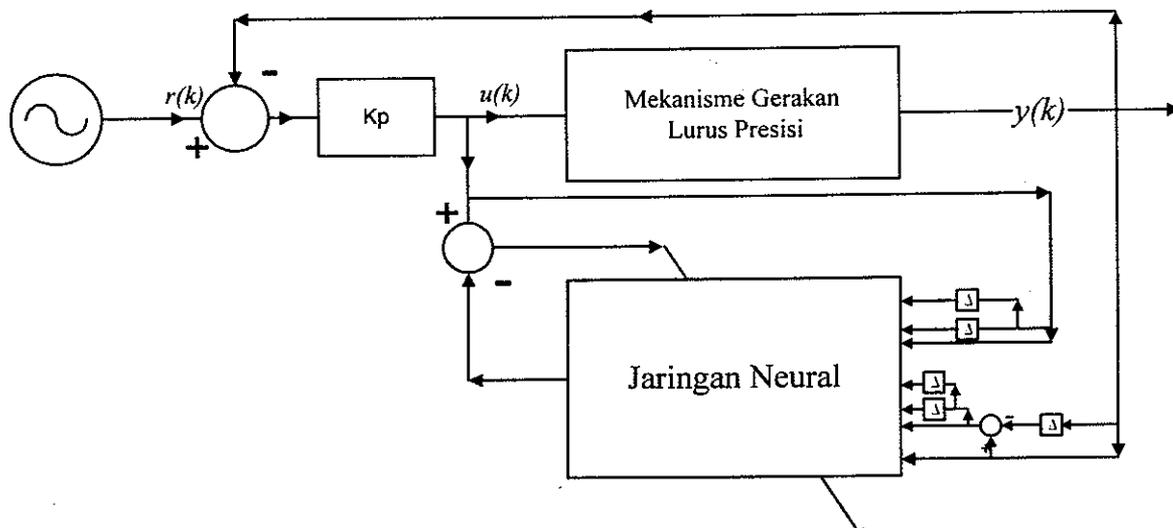
dengan  $w$  adalah bobot dari jaringan neural,  $u(k)$  adalah sinyal masukan pada mekanisme presisi pada saat  $t$ , dan  $u_{nn}(k)$  keluaran pengendali pada saat  $t$ . Tujuan dari pembelajaran ini adalah meminimumkan kriteria performansi pada persamaan 1. Gambar 8 menunjukkan diagram blok dari proses pembelajaran general ini

Pada Gambar 8 tampak bahwa sistem menerima sinyal masukan  $r(k)$ . Untuk menghindari terjadinya ketidakstabilan yang dapat berakibat gerakan meja keluar dari rentang pembacaan sensor posisi, pengendali proporsional digunakan pada proses belajar ini. Pengendali proporsional tersebut memiliki konstanta proporsional  $Kp$ . Keluaran dari pengendali proporsional,  $u(k)$ , yang berupa tegangan bagi motor DC, diperoleh dengan mengalikan galat antara  $r(k)$  dan posisi meja aktual,  $y(k)$ , dengan  $Kp$ . Hasil perkalian tersebut digunakan sebagai referensi bagi proses belajar jaringan neural.

Berdasarkan prinsip kerja mekanisme gerakan lurus presisi dapat diketahui bahwa kecepatan gerakan meja merupakan fungsi dari besar tegangan masukan serta posisi awal meja, yang dapat dinyatakan sebagai persamaan berikut :

$$v(k) = y(k) - y(k-1) = f(u(k), y(k)) \quad (2)$$

Dengan menggunakan algoritme belajar rambat mundur dengan tujuan meminimumkan kriteria performansi pada persamaan (1) diharapkan jaringan neural dapat memprakirakan fungsi  $g[.]$  pada persamaan (3).



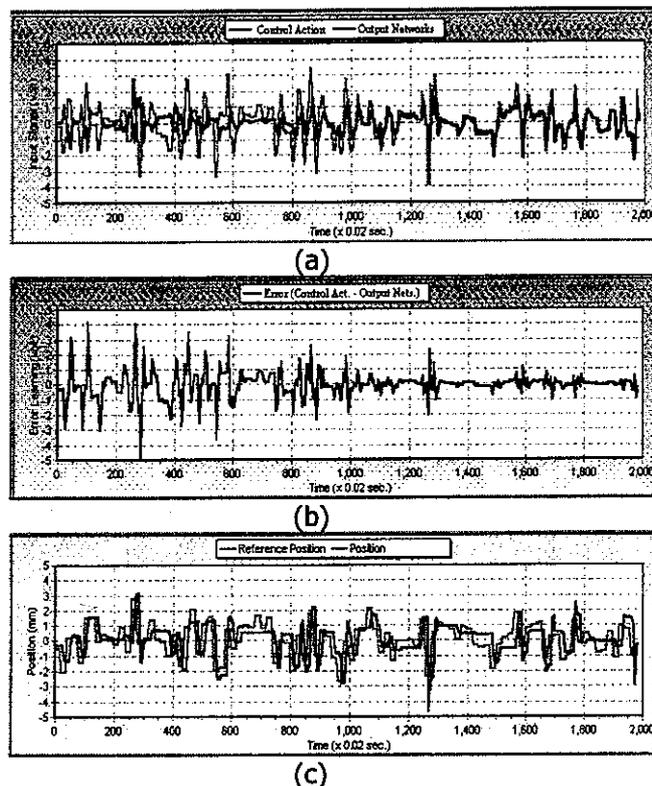
Gambar 8 Diagram blok dari proses pembelajaran general.

Gambar 9 menunjukkan hasil pelatihan jaringan neural. Pada proses pelatihan ini digunakan nilai konstanta belajar,  $\eta$ , sebesar 0.0005 dan konstanta momentum,  $\alpha$ , sebesar 0.5.

Pada Gambar 9a. ditampilkan sinyal masukan ke mekanisme presisi yang berupa tegangan motor,  $u(k)$ , dan sinyal keluaran dari jaringan neural,  $u_{nn}(k)$ , yang merupakan prediksi dari sinyal masukan. Pada gambar tersebut terlihat bahwa  $u_{nn}(k)$  berusaha untuk mengikuti  $u(k)$ . Gambar 9b menampilkan galat antara sinyal masukan aktual,  $u(k)$ , dan sinyal masukan prediksi,  $u_{nn}(k)$ . Dari gambar tersebut terlihat bahwa besar galat yang terjadi semakin mengecil dengan semakin bertambahnya waktu. Hal ini menunjukkan bahwa jaringan dapat belajar cukup baik. Sedangkan Gambar 9c menampilkan sinyal referensi,  $r(k)$ , dan posisi aktual meja,  $y(k)$ .

Untuk melihat kesahihan model invers yang diperoleh dari proses belajar general ini, perlu dilakukan validasi. Untuk melakukan validasi tersebut, sinyal referensi  $r(k)$ , diganti dengan jenis sinyal yang lain serta memilih nilai  $\eta$  dan  $\alpha$  sama dengan nol yang berarti bahwa jaringan neural dalam kondisi tidak belajar

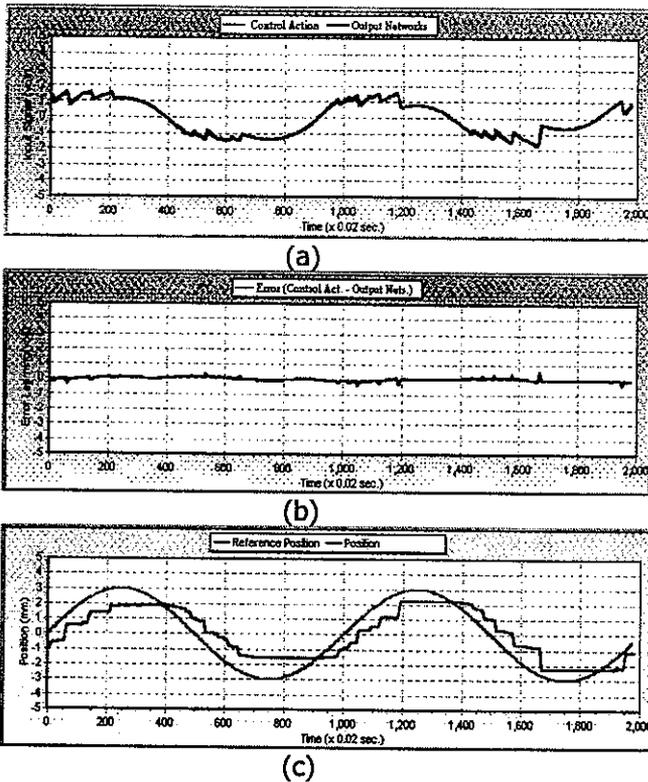
Hasil proses validasi dengan sinyal referensi sinus ditunjukkan pada Gambar 10. Pada Gambar 10a tampak bahwa sinyal aksi kendali dan keluaran jaringan neural saling berhimpit, serta pada Gambar 10b tampak bahwa galat antara kedua sinyal tersebut relatif kecil. Hal ini menunjukkan jaringan neural dapat memperoleh model invers sistem gerakan lurus presisi dengan cukup baik. Selanjutnya jaringan neural yang telah dilatih pada proses belajar general ini akan digunakan sebagai pengendali pada proses pembelajaran khusus



Gambar 9 Proses pembelajaran general dengan sinyal referensi random  $\eta$  sama dengan 0.0005 dan  $\alpha$  sama dengan 0.5  
 (a). Aksi kendali,  $u(k)$ , dan keluaran jaringan neural  $u_{nn}(k)$  (b) Galat antara  $u(k)$  dan  $u_{nn}(k)$  (c). Posisi referensi dan posisi aktual

#### 4.2 Pembelajaran secara khusus

Pada tahap pembelajaran ini, jaringan neural bekerja sebagai pengendali invers langsung.

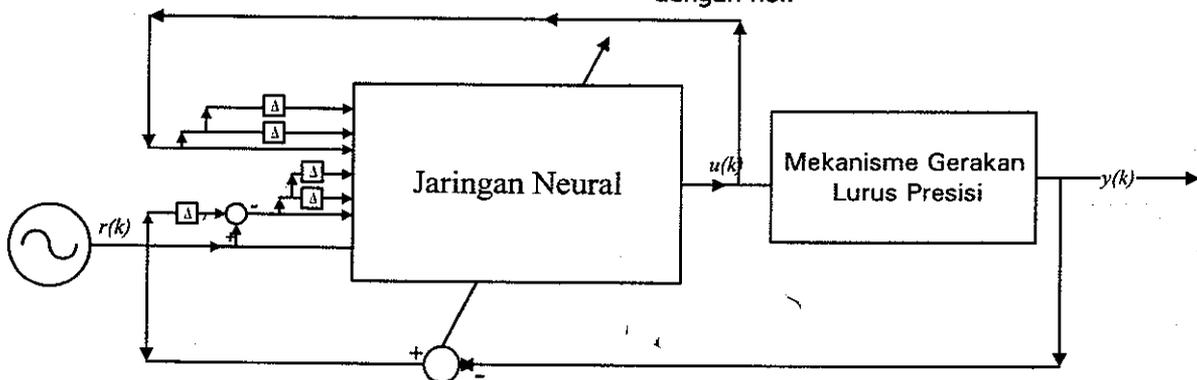


Gambar 10 Proses pembelajaran general dengan sinyal referensi sinus  $\eta$  sama dengan 0.0 dan  $\alpha$  sama dengan 0.0 (a). Aksi kendali,  $u(k)$ , dan keluaran jaringan neural  $u_{nn}(k)$  (b) Galat antara  $u(k)$  dan  $u_{nn}(k)$  (c). Referensi posisi dan posisi aktual

Kriteria performansi yang digunakan pada tahap pembelajaran ini berdasarkan metode penyamaan keluaran (*output matching*) [10], yang dapat dinyatakan dalam persamaan berikut ini

$$J(w) = 0.5(r(k) - y(k))^2 \quad (4)$$

dengan  $r(k)$  adalah posisi meja yang diinginkan,  $y(k)$  adalah posisi aktual meja dan  $w$  adalah pembobot jaringan neural.



Gambar 11 Diagram blok sistem kendali gerakan lurus presisi

Dengan menggunakan kriteria performansi pada persamaan 4, jaringan neural diupayakan untuk menghasilkan aksi kontrol  $u(k)$ , sedemikian sehingga nilai  $J(w)$  minimum. Untuk itu jaringan neural disusun berdasarkan persamaan berikut ini:

$$u(k) = g[r(k), v(k), v(k-1), \dots, v(k-n), u(k-1), u(k-2), \dots, u(k-m)]$$

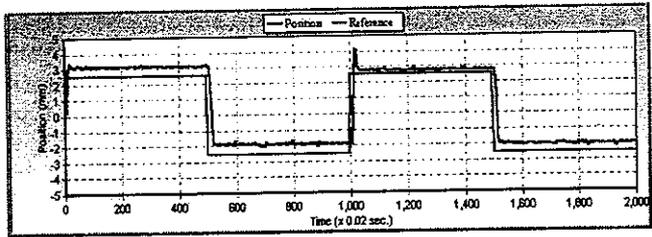
Berdasarkan persamaan di atas disusun diagram blok sistem kendali mekanisme gerakan lurus presisi seperti pada Gambar 11. Untuk melihat prestasi dari penerapan pengendali neural ini dilakukan serangkaian pengujian berikut.

### A. Jaringan tidak belajar

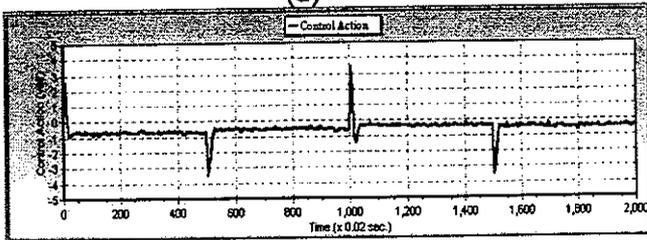
Pengujian ini dilakukan untuk melihat seberapa jauh keberhasilan proses pembelajaran general. Oleh karena itu pada tahap ini konstanta belajar dipilih sama dengan 0 demikian pula dengan konstanta momentum, dengan demikian tidak terjadi proses adaptasi bobot pada pengujian ini. Gambar 12 dan Gambar 13 menunjukkan hasil pemosisian dengan sinyal referensi  $r(k)$ , berupa sinyal signum dan sinyal langkah pada daerah operasi di sekitar titik 1.

Dari kedua gambar tersebut tampak bahwa meskipun kesalahan tunak (*steady state error*) yang terjadi masih cukup besar, namun posisi meja dapat mengikuti sinyal referensi yang diberikan. Hal ini memperkuat pernyataan bahwa dari proses pembelajaran general, jaringan neural telah berhasil mengidentifikasi secara garis besar fungsi invers dari mekanisme gerakan presisi.

Untuk meningkatkan prestasi kendali dari pengendali jaringan neural maka jaringan melakukan proses belajar selama proses pengendalian dengan cara memilih nilai konstanta belajar dan konstanta momentum tidak sama dengan nol.

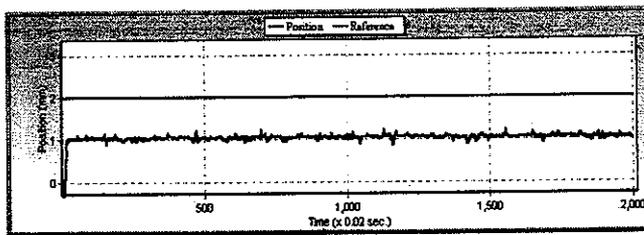


(a)

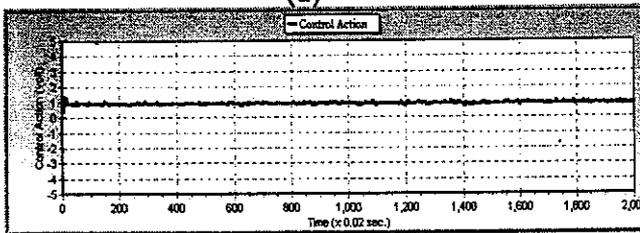


(b)

Gambar 12 Penerapan Pengendali jaringan neural dengan nilai  $\eta$  dan  $\alpha$  sama dengan nol dengan sinyal signum sebagai referensi (a) Sinyal referensi dan posisi aktual meja (b) Aksi kendali jaringan neural.



(a)



(b)

Gambar 13 Penerapan Pengendali jaringan neural dengan nilai  $\eta$  dan  $\alpha$  sama dengan nol dengan sinyal langkah dari 0 ke 2 mm sebagai referensi. (a) Sinyal referensi dan sinyal posisi aktual meja (b) Aksi kendali jaringan neural

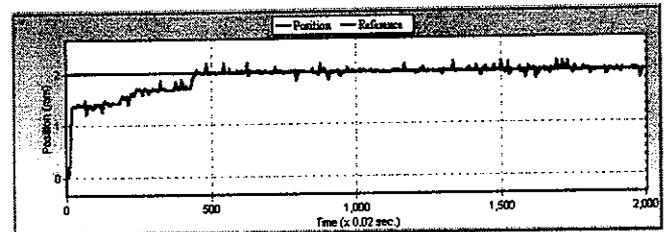
## B. Jaringan belajar secara on-line

Pada tahap ini jaringan belajar secara on-line dengan nilai konstanta belajar sebesar,  $\eta$ , 0.0015 dan konstanta momentum,  $\alpha$ , sebesar 0.5. Gambar 14 dan Gambar 15 menunjukkan hasil pemosisian untuk arah gerakan ke kanan dan ke kiri di daerah operasi sekitar titik 1. Gambar 14a dan Gambar 15a menunjukkan sinyal referensi langkah dan posisi aktual meja. Gambar 14b dan Gambar 15b menunjukkan besar aksi kendali dari pengendali jaringan neural. Sedangkan Gambar 14c dan Gambar 15c menunjukkan besar galat

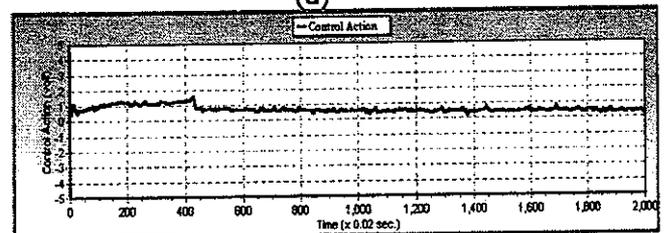
yang terjadi antara posisi referensi,  $y_d(\cdot)$ , dan posisi aktual meja,  $y(\cdot)$ .

Dari pengujian ini menunjukkan bahwa dengan pembelajaran secara *on line* prestasi kendali jaringan neural meningkat jauh lebih baik hal ini ditunjukkan oleh kesalahan tunaknya berada di bawah 20  $\mu\text{m}$ . Selain itu dari pengujian ini juga diperlihatkan bahwa pengendali dapat beradaptasi dengan perubahan perilaku dinamik sistem. Hal ini ditunjukkan oleh prestasi kendali jaringan neural yang cukup baik untuk arah gerakan ke kiri dan ke kanan. Padahal penelitian yang dilakukan oleh Sibenordy [2] menunjukkan bahwa sistem memiliki perilaku yang sangat berbeda untuk gerakan ke kiri dan ke kanan.

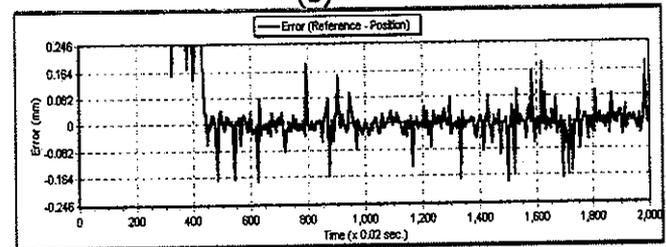
Untuk melihat kemampuan pengendali jaringan neural beradaptasi pada rentang operasi yang cukup lebar dilakukan pengujian yang serupa dengan pengujian di atas namun untuk daerah operasi di sekitar titik 10. Pada pengujian ini semua parameter jaringan sama dengan pengujian sebelumnya yaitu  $\eta=0.0015$  dan  $\alpha = 0.5$ . Hasil pengujian untuk arah gerakan ke kanan dan ke kiri ditunjukkan pada Gambar 16 dan 17.



(a)

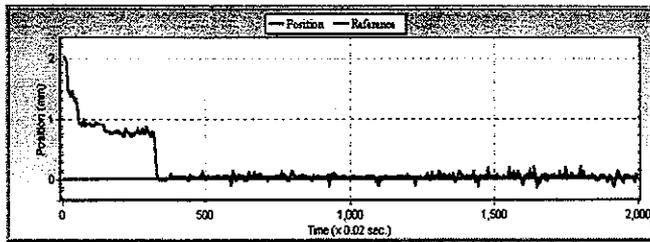


(b)

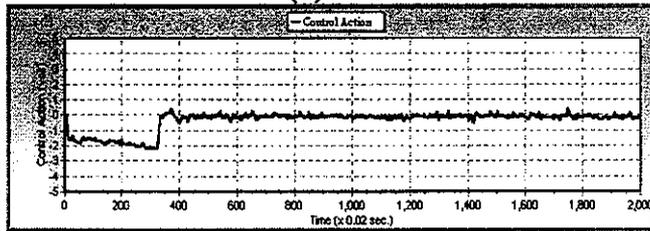


(c)

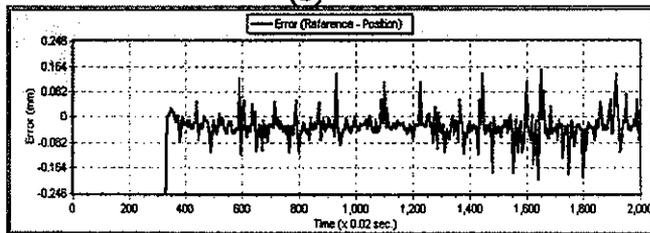
Gambar 14 Penerapan pengendali jaringan neural langkah dari 0 ke 2 mm sebagai referensi dengan daerah operasi di sekitar titik 1. (a) Sinyal referensi dan posisi aktual meja (b) Aksi kendali jaringan neural (c) Galat antara sinyal referensi dan posisi aktual meja



(a)



(b)



(c)

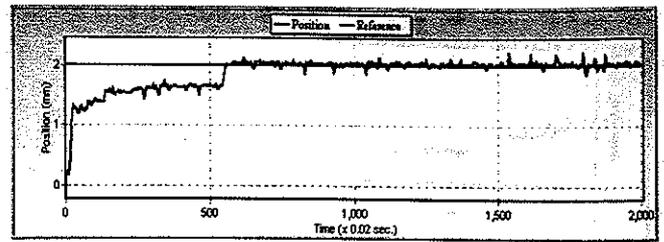
Gambar 15 Penerapan pengendali jaringan neural langkah dari 0 ke 2 mm sebagai referensi (a) Sinyal referensi dan sinyal posisi aktual meja. (b) Aksi kendali jaringan neural (c) Galat antara sinyal referensi dan posisi aktual meja

Dari pengujian yang dilakukan pada daerah operasi di sekitar titik 10 menunjukkan bahwa pengendali mampu melakukan pemosisian dengan cukup baik seperti halnya pada daerah operasi di sekitar titik 1. Padahal perilaku dinamik pada kedua daerah sangat berbeda [2]. Hal ini menunjukkan bahwa penerapan pengendali jaringan neural telah berhasil memperluas daerah operasi pemosisian tanpa harus campur tangan dari operator.

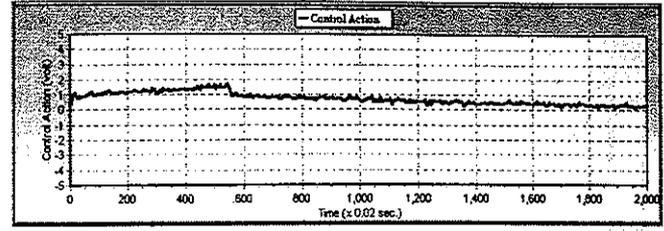
Dari hasil-hasil pengujian ini terlihat adanya lonjakan-lonjakan (*spike*) pada sinyal posisi meja. Munculnya lonjakan-lonjakan tersebut diduga berasal dari derau (*noise*) pengukuran mengingat secara fisik gerakan meja seperti itu sangat kecil kemungkinannya.

## 5. Kesimpulan

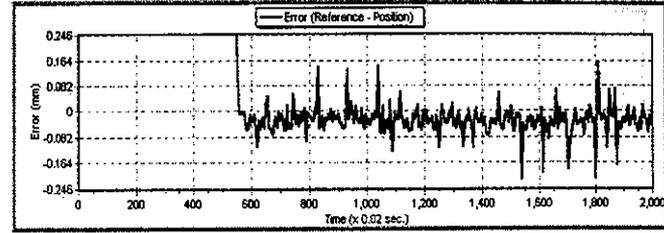
Penelitian ini menunjukkan bahwa pengendali neural dapat berfungsi dengan baik sebagai pengendali pada sistem gerakan lurus presisi. Hal ini ditunjukkan oleh prestasinya yang relatif konstan untuk pemosisian dalam arah yang berbeda serta daerah operasi yang berbeda.



(a)



(b)



(c)

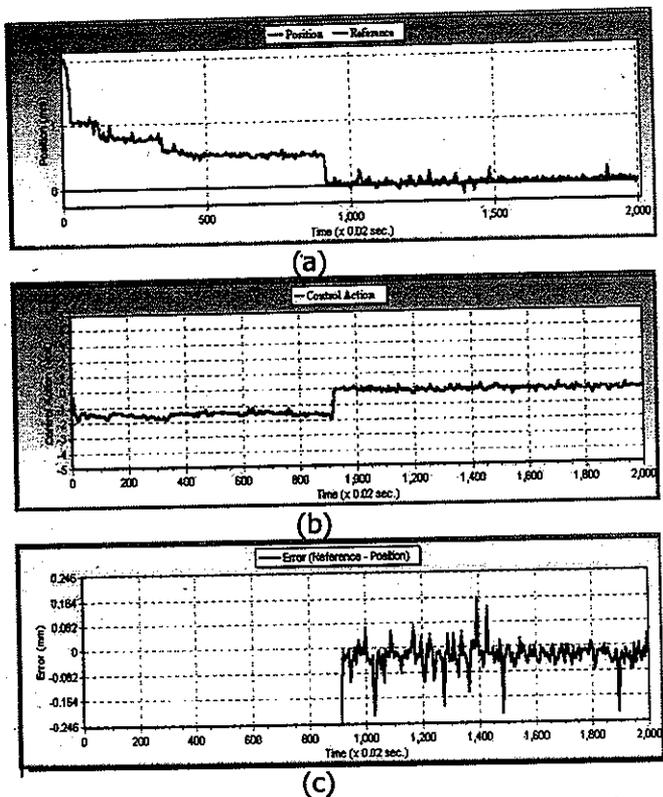
Gambar 16 Penerapan pengendali jaringan neural langkah dari 0 ke 2 mm sebagai referensi di daerah operasi sekitar titik 10. (a) Sinyal referensi dan sinyal posisi aktual meja (b) Aksi kendali jaringan neural (c) Galat antara sinyal referensi dan posisi aktual meja

## 6. Ucapan terima kasih

Makalah ini merupakan hasil dari Penelitian Hibah Tim yang berjudul *Dynamic Modeling and Control of A High Precision Straight-Motion Mechanism* yang didanai oleh Direktorat Jenderal Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan melalui kontrak nomor : 017/HTTP-III/URGE/1997.

## 7. Daftar Pustaka

1. Santoso, I.B., Abidin, Z., 1998, "Penerapan pengendali fuzzy pada sistem gerakan lurus presisi", *Jurnal Teknik Mesin* Vol XIII - no.3 - September 1998
2. Sibenordy, 1998, "Pemodelan Dinamik dan Penerapan Teknik Kendali PID Digital Pada Mekanisme Gerakan Lurus Presisi Tinggi", *Jurusan Teknik Mesin ITB*.
3. Abidin, Z., and Santoso, I.B., 1996, "Implementation of Neural Network Controllers to a MIMO unidirectional Nonlinear Plant", *Proceedings of ICARV'96*, Singapore.



Gambar 17 Penerapan pengendali jaringan neural langkah dari 0 ke 2 mm sebagai referensi pada daerah operasi di sekitar titik 10. (a) Sinyal referensi dan sinyal posisi aktual meja (b) Aksi kendali jaringan neural (c) Galat antara sinyal referensi dan posisi aktual meja

4. D.H. Nguyen and B. Widrow, 1990, "Neural Networks for Self-Learning Control Systems", IEEE Control Systems Magazine.
5. Lee, Hyukjoon., Park, Yongseok., Mehrotra, Kishan., Mohan, Chilukuri., Ranka, Sanjay., "Nonlinear System Identification Using Recurrent Networks"
6. Santoso, I.B., Abidin, Z., 1996, "Ketegaran Pengendali neuro terhadap derau", Proceeding ETM, Bandung.
7. Fausett, Laurene., 1994, "Fundamental of Neural Networks", Prentice-Hall Inc.
8. Jang, J.-S. R., Sun, C.-T., Mizutani, E., 1997, "Neuro-Fuzzy and Soft Computing", Prentice Hall.
9. Lin, Chin-Teng., Lee, George C.S., 1996, "Neural Fuzzy Sistem", Prentice Hall Inc.
10. Andersen, H. C. A., 1998, "The Controller Output Error Method", Department of Computer Science dan Electrical Engineering - University of Queensland, Australia.
11. Norgaard, M., 1997, "Neural network based control system design", Departemen of Automation, Technical University of Denmark.