

PENGENDALIAN MOTOR DC MENGGUNAKAN JARINGAN SYARAF TIRUAN *BACKPROPAGATION*

Wahyudi, Sorikhi, dan Iwan Setiawan.

Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang
e-mail : wahyuditinom@yahoo.com.

Abstrak

Jaringan syaraf backpropagation merupakan jaringan syaraf yang telah umum digunakan dalam beragam aplikasi kontrol. Dari segi arsitektur, backpropagation merupakan jaringan syaraf perceptron dengan banyak layer (multilayer), sehingga jaringan syaraf ini dikenal juga sebagai jaringan syaraf perceptron multilayer. Nama backpropagation diambil dari proses pembelajaran yang melibatkan penghantaran balik sinyal error dari output layer menuju layer sebelumnya hingga layer sebelum input layer. Jaringan syaraf backpropagation digunakan sebagai kontroler dalam pengaturan kecepatan motor DC. Skema yang digunakan dalam pengontrolan adalah fix stabilizing controller. Dalam skema ini, jaringan syaraf berperan sebagai controller dimana sinyal training yang digunakan berasal dari sinyal keluaran penguat proporsional. Pengujian dilakukan pada pengaruh learning, jumlah hidden layer, dan penguat proporsional terhadap respon transien plant. Hasil pengujian menunjukkan bahwa jaringan dengan satu hidden layer lebih cepat mencapai kondisi tunak dibandingkan jaringan dengan dua hidden layer. Nilai penguat proporsional dimana respon plant masih stabil adalah 2.

Kata kunci: backpropagation, fix stabilizing controller, plant motor DC.

1. PENDAHULUAN

Pada saat ini banyak teknik kontrol alternatif, seperti *neural network* (jaringan syaraf tiruan), adaptif, serta *fuzzy* kontrol sebagai alternatif teknik kontrol klasik. Perkembangan jaringan syaraf tiruan (JST) telah menginspirasi munculnya teknik kontrol alternatif yang baik dan efisien. JST mempunyai kemampuan mempelajari kondisi dinamis sistem dengan mengestimasi fungsi *input-output* sistem. Dengan menggunakan JST, akan didapat suatu kontroler tanpa harus memodelkan *plant* secara matematik serta penalaan parameter kontroler. Pada teknik kontrol klasik, seperti PID, membutuhkan pemodelan *plant* secara matematik dan penalaan parameter dari kontroler harus dilakukan terlebih dahulu sebelum mengimplementasikan kontrolernya. Namun demikian teknik kendali klasik masih banyak diterapkan pada dunia industri, karena telah terbukti handal dalam menyelesaikan permasalahan kontrol dan para pengguna masih enggan beralih ke teknik kontrol alternatif. Pada paper ini dibahas salah satu kontrol alternatif dengan menggunakan jaringan syaraf tiruan backpropagation untuk mengendalikan motor DC. Pembahasan dilakukan dengan mengamati pengaruh learning rate, jumlah hidden layer, dan penguat proporsional pada skema kendali fix stabilizing controller.

2. TINJAUAN PUSTAKA

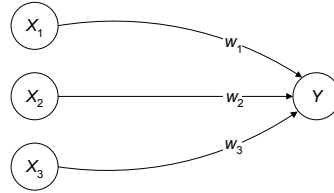
Jaringan syaraf tiruan adalah sistem pengolahan informasi yang mempunyai karakteristik unjuk kerja tertentu sebagaimana jaringan syaraf biologis^[2]. Jaringan syaraf tiruan didasarkan pada beberapa asumsi :

1. Pengolahan informasi terjadi pada neuron.
2. Sinyal-sinyal dilewatkan antar neuron melalui rantai penghubung.
3. Masing-masing rantai penghubung akan mengalikan sinyal yang ditransmisikan.
4. Masing-masing neuron menggunakan fungsi aktivasi pada jaringan masukannya untuk menentukan sinyal keluaran.

Jaringan syaraf tiruan dicirikan oleh hubungan pola antar neuron, metode untuk menentukan bobot, dan fungsi aktivasinya. Gambar 1 menunjukkan sebuah neuron *Y* yang menerima input dari neuron *X*₁, *X*₂, dan *X*₃. Bobot pada hubungan dari *X*₁, *X*₂, dan *X*₃ ke neuron *Y* adalah *w*₁, *w*₂, dan *w*₃. Sinyal masukan jaringan, *y*_{in}, ke neuron *Y* adalah jumlah dari sinyal terboboti (*weighted signal*) dari *X*₁, *X*₂, dan *X*₃, yaitu : $y_{in} = w_1x_1 + w_2x_2 + w_3x_3$. Aktivasi *y* dari neuron *Y* diberikan oleh suatu fungsi dari jaringan masukannya, $y = f(y_{in})$, sebagai contohnya adalah fungsi *sigmoid* biner (sebuah kurva berbentuk *S*) yang memenuhi persamaan berikut.

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Fungsi non linier digunakan pada jaringan *multilayer* karena hasil pengumpanan sinyal melalui dua atau lebih *layer* dengan menggunakan fungsi linier akan diperoleh hasil yang tidak berbeda dengan apa yang diperoleh jika menggunakan *single layer* (lapis tunggal).



Gambar 1 Neuron sederhana.

Pelatihan dengan menggunakan jaringan syaraf *backpropagation* melibatkan tiga tahapan: *feedforward* pola pelatihan masukan, *backpropagation* terhadap *error*, serta penyesuaian bobot. Algoritma pelatihan *backpropagation* adalah sebagai berikut.

- Langkah 0. Inisialisasi bobot.
- Langkah 1. Ketika kondisi berhenti salah, lakukan langkah 2-9.
- Langkah 2. Untuk masing-masing pasangan *training*, lakukan langkah 3-8.

Feedforward:

- Langkah 3. Masing-masing neuron masukan ($X_i, i = 1, \dots, n$) menerima sinyal masukan x_i dan menghantarkan sinyal ini ke semua *unit* dalam *layer* di atasnya (*unit-unit* tersembunyi).
- Langkah 4. Masing-masing neuron tersembunyi ($Z_j, j = 1, \dots, n$) menjumlahkan sinyal masukan terboboti,

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij},$$

hitung fungsi aktivasi untuk menghitung sinyal keluaran,

$$z_j = f(z_in_j),$$

lalu mengirim sinyal ini ke semua *unit* pada *layer* di atasnya (*unit* keluaran).

- Langkah 5. Masing-masing *unit* keluaran ($Y_k, k = 1, \dots, m$) menjumlahkan sinyal masukan terboboti,

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

serta menghitung fungsi aktivasinya untuk menghitung sinyal keluaran,

$$y_k = f(y_in_k).$$

Backpropagation of error:

- Langkah 6. Masing-masing *unit* keluaran ($Y_k, k = 1, \dots, m$) menerima pola target terhadap pola masukan, kemudian menghitung informasi *error*nya,

$$\delta_k = (t_k - y_k) f'(y_in_k),$$

menghitung koreksi bobot,

$$\Delta w_{jk} = \alpha \delta_k z_j,$$

menghitung koreksi *bias*,

$$\Delta w_{0k} = \alpha \delta_k,$$

dan mengirim δ_k ke *unit* di *layer* dibawahnya.

- Langkah 7. Masing-masing *unit* tersembunyi ($Z_j, j = 1, \dots, p$) menjumlahkan masukan *deltanya* (dari *unit* pada *layer* di atasnya),

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk},$$

kalikan dengan turunan fungsi aktivasi untuk menghitung informasi *error*,

$$\delta_j = \delta_in_j f'(z_in_j),$$

menghitung koreksi bobot,

$$\Delta v_{ij} = \alpha \delta_j x_i,$$

kemudian menghitung koreksi *bias*,

$$\Delta v_{0j} = \alpha \delta_j.$$

Update bobot dan bias:

- Langkah 8. Masing-masing *unit* keluaran ($Y_k, k = 1, \dots, m$) *update bias* dan bobotnya ($j = 0, \dots, p$):

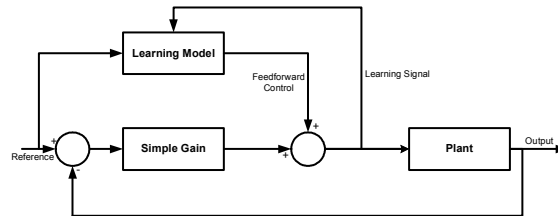
$$W_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

Masing-masing *unit* tersembunyi ($Z_j, j = 1, \dots, p$) melakukan *update bias* dan bobotnya ($i = 0, \dots, n$):

$$V_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} .$$

Langkah 9. Uji syarat henti.

Salah satu skema *direct learning control* yang paling sederhana adalah sebagaimana terlihat pada Gambar 2^[3]. Kontroler linier didesain demikian hingga sistem loop tertutup menjadi stabil dalam setiap daerah operasi dan sinyal dari kontroler tersebut menjadi sinyal *training* bagi modul pembelajaran.

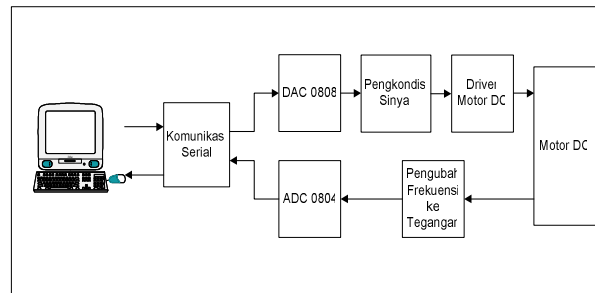


Gambar 2 Fixed Stabilizing Controller^{[3][8]}.

Pada skema ini jaringan syaraf tiruan berperan sebagai model inverse dinamis dari *plant*. Model ini menggunakan sinyal referensi dan keluaran aktual *plant* sebagai masukan serta *error* kontrol umpan balik sebagai algoritma adaptasi. Ketika jaringan syaraf tiruan telah terlatih, maka sinyal *error* akan mengalami konvergensi menuju nol. Dengan demikian kontroler jaringan syaraf tiruan akan berlatih mengambil peran kontroler umpan balik.

3. METODE PENELITIAN

Secara umum sistem yang dibuat ditunjukkan pada Gambar 3.



Gambar 3 Diagram blok perancangan sistem.

Penjelasan dari masing – masing blok adalah sebagai berikut.

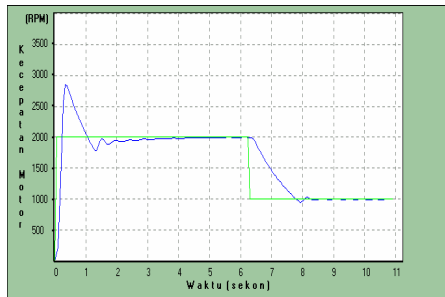
1. DAC 0808 berfungsi untuk mengubah data digital berupa masukan referensi kecepatan motor dari komputer menjadi data analog berupa tegangan.
2. Pengkondisi sinyal berfungsi untuk menguatkan tegangan keluaran dari DAC. Keluaran DAC maksimal adalah 5 V, sedangkan tegangan maksimal yang dibutuhkan adalah 13 V sehingga perlu dikuatkan sebesar 2,6 kali.
3. *Driver* motor DC untuk menggerakkan motor DC diperlukan arus yang cukup besar, sedangkan *output* pengkondisi sinyal tidak memiliki arus yang cukup untuk menggerakkan motor DC. Oleh karena itu, diperlukan sebuah penggerak motor DC yang tidak lain adalah penguat arus untuk menguatkan tegangan kontrol dari pengkondisi sinyal.
4. Motor DC yang digunakan dilengkapi dengan optokopler beserta piringan berkode yang terpasang pada poros motor.
5. Pengubah frekuensi ke tegangan menggunakan LM2917N sebagai komponen utama yang berfungsi mengubah data frekuensi dari optokopler menjadi tegangan yang sesuai dengan besarnya frekuensi masukan.
6. ADC 0804 digunakan untuk mengubah besaran analog dari pengubah frekuensi ke tegangan menjadi data digital sehingga dapat diolah di komputer.
7. Komunikasi serial, mikrokontroler AT89S51 diposisikan sebagai sarana komunikasi serial, yang mengatur aliran data dari ADC ke komputer atau dari komputer ke DAC.
8. Komputer digunakan sebagai unit kontroler untuk mengaplikasikan algoritma jaringan syaraf tiruan *backpropagation*. Disamping itu, komputer juga digunakan untuk menampilkan grafik respon dan menyimpan data ke dalam file untuk keperluan analisis.

4. HASIL DAN PEMBAHASAN

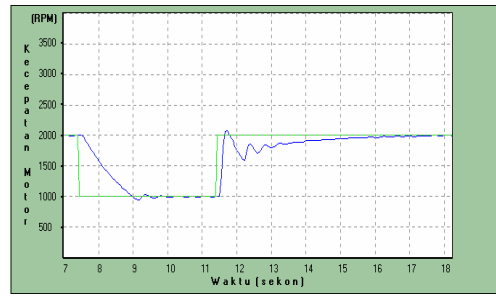
Pengujian dilakukan untuk mengetahui adanya pengaruh *learning rate*, jumlah *hidden layer*, dan penguat proporsional terhadap respon transien *plant*.

4.1 Pengaruh *Learning Rate* terhadap Respon Transien *Plant*

Untuk mengetahui pengaruh nilai *learning rate* pada unjuk kerja jaringan syaraf dilakukan dengan cara mengubah nilai *learning rate* yang berbeda, sementara parameter lainnya diberikan dengan nilai tetap. Pada Gambar 4 dan Gambar 5 diperlihatkan hasil pengujian untuk 2 *learning rate* yang berbeda. Pada Gambar 4.a diperlihatkan respon *plant* mengalami *overshoot* sebesar 45 % dan waktu yang dibutuhkan untuk mencapai kondisi tunak pada kecepatan referensi 2000 RPM sekitar 5,2 detik. Gambar 4.a menunjukkan respon transien *plant* pada saat awal dan pada saat kecepatan referensi diturunkan, sedang pada Gambar 4.b ditunjukkan respon *plant* ketika kecepatan motor dinaikkan dari 1000 RPM menjadi 2000 RPM, respon *plant* mengalami *overshoot* 5 % dengan waktu penetapan sekitar 6,5 detik.



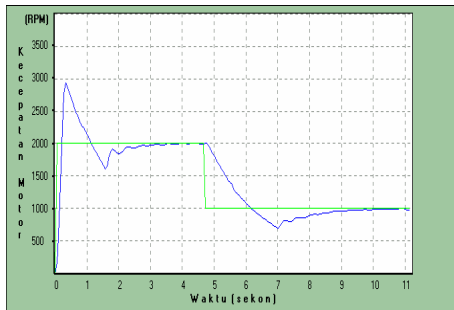
a. Referensi awal dan berubah turun.



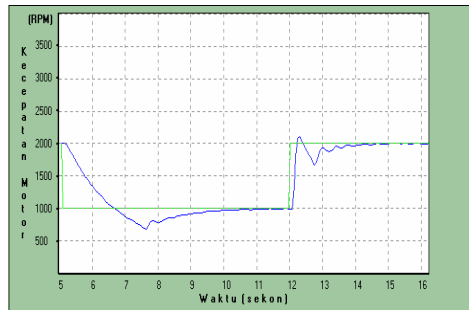
b. Referensi berubah naik.

Gambar 4 Respon transien *plant* dengan nilai *learning rate* = 0,3.

Gambar 5 menunjukkan perubahan respon transien *plant* dengan nilai *learning rate* = 0,7. Pada Gambar 5.a respon *plant* mengalami *overshoot* sebesar 50 % dan waktu yang dibutuhkan untuk mencapai kondisi tunak sekitar 3,8 detik. Gambar 5.b menunjukkan respon sistem ketika kecepatan motor dinaikkan dari 1000 RPM menjadi 2000 RPM, respon *plant* mengalami *overshoot* 10 % dengan waktu penetapan sekitar 2,7 detik.



a. Referensi awal dan berubah turun.



b. Referensi berubah naik.

Gambar 5 Respon transien *plant* dengan nilai *learning rate* = 0,7.

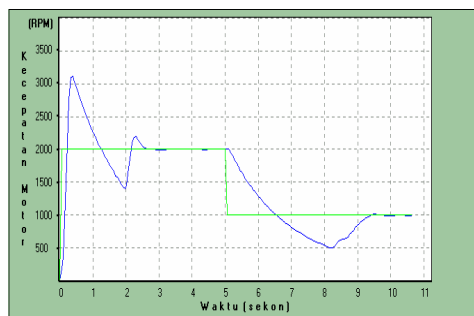
Tabel 1 Pengaruh nilai *learning rate* terhadap respon *plant*.

Parameter unjuk kerja	Respon awal <i>plant</i> mencapai referensi		Respon <i>plant</i> dari kecepatan 2000 RPM ke 1000 RPM		Respon <i>plant</i> dari kecepatan 1000 RPM ke 2000 RPM	
	<i>Overshoot</i> maksimum	Waktu penetapan	<i>Undershoot</i> maksimum	Waktu penetapan	<i>Overshoot</i> maksimum	Waktu penetapan
<i>Learning rate</i> = 0,3	45 %	5,2 detik	5 %	2,25 detik	5 %	6,5 detik
<i>Learning rate</i> = 0,7	50 %	3,8 detik	30 %	5,3 detik	10 %	2,7 detik

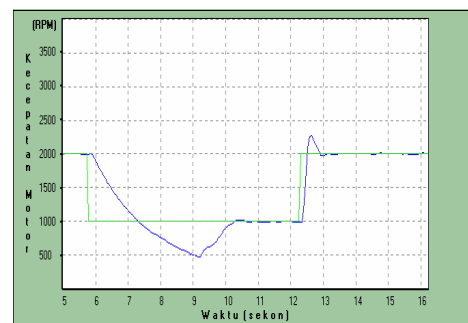
Berdasarkan Tabel 1 dan gambar respon transien *plant*, penambahan nilai *learning rate* akan berpengaruh terhadap kecepatan respon transien *plant*, semakin besar nilai *learning rate* semakin cepat *plant* mencapai kondisi tunak. Penambahan nilai *learning rate* juga akan menambah *overshoot plant* awal serta terjadinya *undershoot* yang lebih besar.

4.2 Pengaruh Jumlah *Hidden Layer* terhadap Respon Transien *Plant*

Dalam beberapa referensi, penentuan jumlah *hidden layer* yang diperlukan dalam menyelesaikan suatu kasus tidak dijelaskan secara mendetail. Dalam hal ini tidak ada panduan yang jelas mengenai berapakah jumlah *hidden layer* yang paling optimal untuk memecahkan suatu permasalahan. Ada yang mengatakan bahwa satu *hidden layer* sudah mencukupi, namun dua *hidden layer* akan menjadikan fungsi jaringan syaraf sebagai aproksimator menjadi lebih baik^[2]. Ada pula yang menyatakan bahwa pemilihan jumlah *hidden layer* lebih ditentukan secara empiris daripada dengan perhitungan matematis^[6]. Pada paper ini jumlah *hidden layer* yang dibandingkan adalah 1 dan 2. Gambar 6 menunjukkan respon transien *plant* pada jaringan syaraf dengan satu *hidden layer*. Respon *plant* mengalami *overshoot* sebesar 55 % dan waktu yang dibutuhkan untuk mencapai kondisi tunak pada kecepatan 2000 RPM sekitar 2,5 detik. Pada Gambar 6.b diperlihatkan respon sistem ketika kecepatan motor dinaikkan dari 1000 RPM menjadi 2000 RPM, respon *plant* mengalami *overshoot* 12,5 % dengan waktu penetapan yang dibutuhkan sekitar 1 detik.



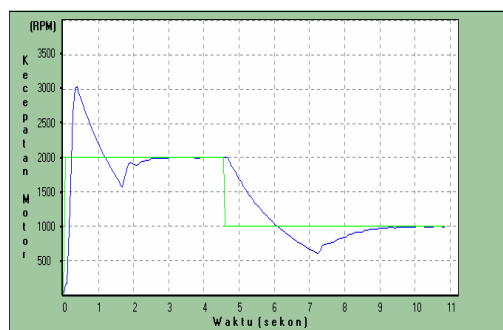
a. Referensi awal dan berubah turun.



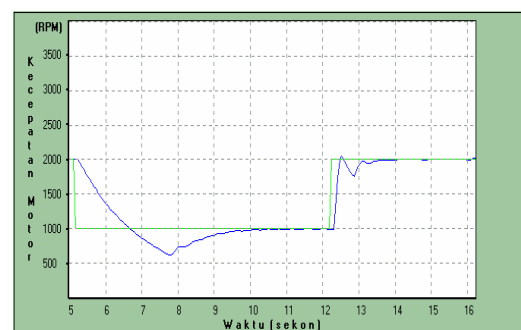
b. Referensi berubah naik.

Gambar 6 Respon transien *plant* dengan jumlah *hidden layer* = 1.

Pada Gambar 7 menunjukkan perubahan respon transien *plant* dengan dua *hidden layer*. Pada saat awal, respon *plant* mengalami *overshoot* sebesar 49,9 % dan waktu yang dibutuhkan untuk mencapai kondisi tunak pada kecepatan 2000 RPM sekitar 2,8 detik. Pada Gambar 7.b diperlihatkan saat kecepatan motor dinaikkan dari 1000 RPM menjadi 2000 RPM, respon *plant* mengalami *overshoot* 2,5 % dengan waktu penetapan yang dibutuhkan sekitar 1,6 detik.



a. Referensi awal dan berubah turun.



b. Referensi berubah naik.

Gambar 7 Respon transien *plant* dengan jumlah *hidden layer* = 2.

Tabel 2 Pengaruh jumlah *hidden layer* terhadap respon *plant*.

Parameter unjuk kerja	Respon awal <i>plant</i>		Respon <i>plant</i> dari kecepatan 2000 RPM ke 1000 RPM		Respon <i>plant</i> dari kecepatan 1000 RPM ke 2000 RPM	
	<i>Overshoot</i> maksimum	Waktu penetapan	<i>Undershoot</i> maksimum	Waktu penetapan	<i>Overshoot</i> maksimum	Waktu penetapan
<i>Hidden layer</i> = 1	55 %	2,5 detik	51 %	4,3 detik	12,5 %	1 detik
<i>Hidden layer</i> = 2	49,9 %	2,8 detik	35 %	5 detik	2,5 %	1,6 detik

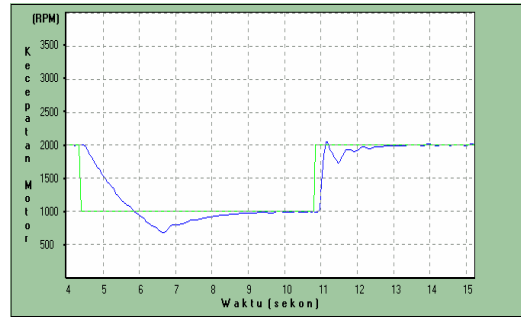
Pada Tabel 2 dapat dilihat bahwa jaringan dengan satu *hidden layer* mempunyai waktu penetapan yang lebih cepat dibandingkan dengan jaringan dengan dua *hidden layer*. Pada jaringan dengan satu *hidden layer*, respon *plant* mengalami *overshoot* yang lebih besar jika dibandingkan jaringan dengan dua *hidden layer*.

4.3 Pengaruh Penguat Proporsional terhadap Respon Transien Plant

Pengujian untuk mengetahui pengaruh nilai penguat proporsional pada unjuk kerja jaringan syaraf dilakukan dengan cara mengubah nilai penguat proporsional yang berbeda, sementara parameter lainnya diberikan dengan nilai tetap. Pada Gambar 8 dan Gambar 9 diperlihatkan hasil pengujian untuk 2 nilai penguat proporsional yang berbeda. Pada Gambar 8.a diperlihatkan respon *plant* mengalami *overshoot* sebesar 49,9 % dan waktu yang dibutuhkan untuk mencapai kondisi tunak pada referensil 2000 RPM sekitar 3,7 detik. Pada Gambar 8.b diperlihatkan respon *plant* pada saat referensi diubah dari 1000 RPM menjadi 2000 RPM, respon memberikan nilai *overshoot* 1 % dengan waktu penetapan sekitar 2,2 detik.



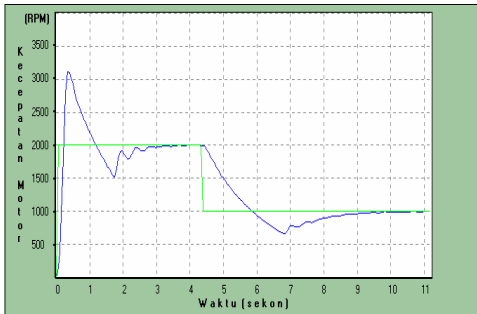
a. Referensi awal dan berubah turun.



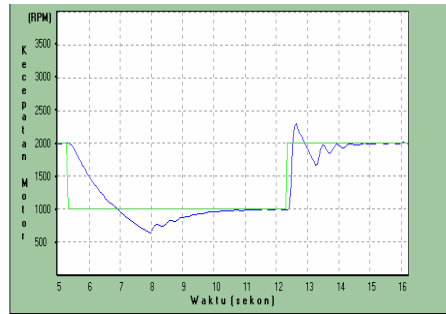
b. Referensi berubah naik.

Gambar 8 Respon transien *plant* dengan penguat proporsional = 1,1.

Pada Gambar 9 diperlihatkan respon *plant* dengan nilai penguat proporsional sebesar 1,5. Pada Gambar 9.a menunjukkan awal *plant* mengalami *overshoot* sebesar 55 % dan waktu yang dibutuhkan untuk mencapai kondisi tunak pada kecepatan referensi sekitar 3,75 detik. Pada Gambar 9.b diperlihatkan saat kecepatan motor dinaikkan dari 1000 RPM menjadi 2000 RPM, respon *plant* mengalami *overshoot* 15 % dengan waktu penetapan sekitar 2,4 detik.



a. Referensi awal dan berubah turun.



b. Referensi berubah naik.

Gambar 9 Respon transien *plant* dengan penguat proporsional = 1,5.

Tabel 3 Pengaruh jumlah penguat proporsional terhadap respon *plant*.

Parameter unjuk kerja	Respon awal <i>plant</i>		Respon <i>plant</i> dari kecepatan 2000 RPM ke 1000 RPM		Respon <i>plant</i> dari kecepatan 1000 RPM ke 2000 RPM	
	<i>Overshoot</i> maksimum	Waktu penetapan	<i>Undershoot</i> maksimum	Waktu penetapan	<i>Overshoot</i> maksimum	Waktu penetapan
Penguat proporsional = 1,1	49,9 %	3,7 detik	30 %	5,2 detik	1 %	2,2 detik
Penguat proporsional = 1,5	55 %	3,75 detik	30 %	4,9 detik	15 %	2,3 detik

Pada Tabel 3 dapat dilihat bahwa jaringan dengan penguat proporsional 1,1 mempunyai waktu penetapan yang lebih cepat dibandingkan dengan jaringan dengan penguat proporsional 1,5. Pada jaringan dengan penguat proporsional 1,1, respon *plant* mengalami *overshoot* yang lebih kecil jika dibandingkan jaringan dengan penguat proporsional 1,5.

5. KESIMPULAN

1. *Learning rate* pada jaringan syaraf *backpropagation* berpengaruh terhadap kecepatan *plant* mencapai kondisi tunak.
2. Jaringan syaraf dengan satu *hidden layer* mempunyai respon transien yang lebih cepat dibandingkan dengan jaringan syaraf dengan dua *hidden layer*.
3. Semakin besar penguat proporsional, kecepatan *plant* mencapai kondisi tunak tidak semakin cepat. Nilai kritis penguat proporsional agar *plant* tidak osilasi adalah 2.
4. Hasil pengujian yang memberikan respon paling baik adalah untuk nilai *learning rate* = 0,75, jumlah *hidden layer* = 1, dan penguat proporsional = 1,1.

6. DAFTAR PUSTAKA

- Fausett, Laurene, 1994, *Fundamentals of Neural Networks: Architecture, Algorithms, and Applications*, Prentice Hall Inc.
- Hagan Martin T., Demuth Howard B., *Neural Network for Control*, IEEE Journal.
- Malvino, Paul Albert, 1996, *Elektronika Komputer Digital: Pengantar Mikrokomputer (terj.)*, Edisi Ke-6, Erlangga.
- Ogata, Katsuhiko, 1991, *Teknik Kontrol Otomatik, Jilid 1*, Erlangga.
- Setiawan, Iwan, *Jaringan Syaraf Tiruan untuk Aplikasi Pemodelan dan Pengendalian*, Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro.
- <http://cortex.snowseed.com/index.html>