

## Desain Kontrol Sistem Eksitasi *Steam Turbine Generator (STG) Pada Combine Cycle Power Plant 120 MW PT.Krakatau Daya Listrik Menggunakan Jaringan Syaraf Tiruan Backpropagation*

Muhamad Otong<sup>1</sup>, Ri Munarto<sup>2</sup>, Dasep Iskandar<sup>3</sup>

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa, Cilegon, Banten.

### Informasi Artikel

Naskah Diterima : 11 Mei 2018

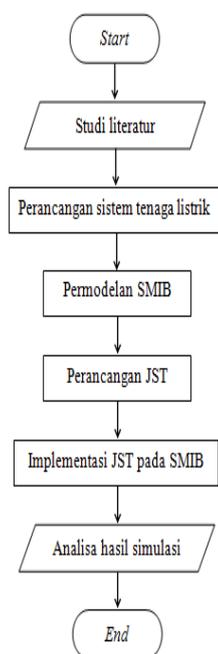
Direvisi : 20 Juni 2018

Disetujui : 24 Juni 2018

\*Korespondensi Penulis :

muhamadotonguntirta@yahoo.com

### Graphical abstract



### Abstract

Excitation System is a vital part of generating electricity in the generator, the function of the excitation system in the generator is to control the output of the generator to remain stable at the changing system load. The excitation control system of the synchronous generator is one of the steps to improve the stability of the power system and to ensure the quality of the electricity generated by the network. In this research used one of the implementation of artificial intelligence technology is Artificial Neural Network designed as controller excitation system generator. The network is designed using a backpropagation learning algorithm, and the Lavenberg-Marquardt training algorithm with a feedforward multilayer network architecture. The result of simulated test and training using Matlab application obtained with presentation of system success equal to 96,387%.

**Keywords:** Excitation System, JST, Lavenberg-Marquardt, Backpropagation.

### Abstrak

Sistem Eksitasi merupakan bagian vital pada proses pembangkitan listrik di generator, fungsi dari sistem eksitasi pada generator adalah untuk mengendalikan output dari generator agar tetap stabil pada beban sistem yang berubah-ubah. Sistem kendali eksitasi generator sinkron adalah salah satu langkah untuk meningkatkan stabilitas sistem tenaga dan untuk menjamin kualitas tenaga listrik yang dihasilkan jaringan. Pada penelitian ini digunakan salah satu implementasi teknologi artificial intelligence, yaitu JST (Jaringan Syaraf Tiruan) yang didesain sebagai pengendali sistem eksitasi generator. Jaringan dirancang menggunakan algoritma pembelajaran backpropagation dan algoritma pelatihan Lavenberg-Marquardt dengan arsitektur jaringan feedforward multilayer. Hasil dari pengujian dan pelatihan yang disimulasikan menggunakan aplikasi matlab diperoleh dengan presentasi keberhasilan sistem sebesar 96,387%.

**Kata kunci:** Sistem Eksitasi, JST, Lavenberg-Marquardt, Backpropagation

© 2018 Penerbit Jurusan Teknik Elektro UNTIRTA Press. All rights reserved

## 1. PENDAHULUAN

Dalam suatu sistem interkoneksi tenaga listrik ada banyak unit pembangkit yang mentransmisikan tenaga ke beban melalui saluran transmisi. Ketika suatu sistem tenaga listrik mengalami gangguan, maka gangguan tersebut akan membatasi kapabilitas jaringan sistem tenaga. Untuk meningkatkan stabilitas sistem tenaga dan untuk menjamin kualitas tenaga listrik yang dihasilkan jaringan, maka diperlukan sebuah teknologi berupa sistem penguatan atau yang lebih sering disebut sebagai sistem kendali eksitasi. Sistem eksitasi ini adalah sebuah teknik penguatan arus medan magnet yang dibangkitkan pada generator dengan menggunakan prinsip elektromagnetis untuk menghasilkan medan magnet penguatan. Eksiter dan AVR adalah bagian dari sistem eksitasi yang dianggap sebagai sumber dari arus medan untuk generator sinkron. Tambahan sinyal kendali

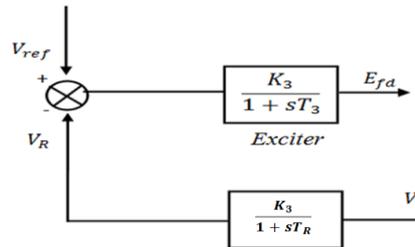
pada sistem eksitasi generator biasanya digunakan untuk meningkatkan stabilitas sistem. Untuk memberikan sinyal kendali pada sistem eksitasi maka digunakan teori kendali *closed-loop feedback*[4].

Fungsi utama sistem eksitasi adalah untuk meregulasi tegangan terminal generator yang dilakukan dengan mengatur tegangan medan yang dipengaruhi tegangan terminal yang bervariasi. Generator merupakan sumber penghasil tenaga listrik pada sistem tenaga. Generator disuplai oleh tenaga dari *prime mover*, sedangkan arus eksitasi disediakan oleh sistem eksitasi. Tegangan eksitasi  $E_{fd}$  dihasilkan dari eksiter dan dikendalikan oleh AVR (*Automatic Voltage Regulator*). Tujuannya untuk menjaga tegangan terminal  $V_t$  sama dengan nilai referensi  $V_{ref}$ [5].

Salah satu implementasi dari teknologi *artificial intelligence* adalah JST (jaringan Syaraf Tiruan) atau ANN (*Artificial Neural Network*). Jaringan syaraf tiruan memiliki potensi besar karena dibangun berdasarkan fondasi matematika yang kuat yang mengikutsertakan alat matematis yang cakap dan mudah dimengerti. Ciri utama yang dimiliki jaringan syaraf tiruan adalah kemampuan untuk belajar. Belajar (*learning*) pada jaringan syaraf tiruan dapat diartikan sebagai proses penyesuaian parameter pembobot interkoneksi yang dimiliki oleh sel. Proses belajar akan dihentikan jika nilai kesalahan atau *error* sudah dianggap cukup kecil untuk semua pasangan data latihan. Jaringan yang sedang melakukan proses belajar disebut berada dalam tahap latihan (*training*). Pada tahap awal pelatihan ini perlu dilakukan terlebih dahulu sebelum melakukan pengujian suatu objek[6]. Pada penelitian kali ini penulis akan membahas tentang Sistem eksitasi *steam turbine generator* (STG) menggunakan jaringan syaraf tiruan algoritma *Backpropagation*, dengan menggunakan parameter-parameter yang diambil dari spesifikasi generator dan eksitasi CAPP PT.Krakatau Daya Listrik yang saat ini sedang mendapatkan perhatian khusus dari divisi *maintenance*.

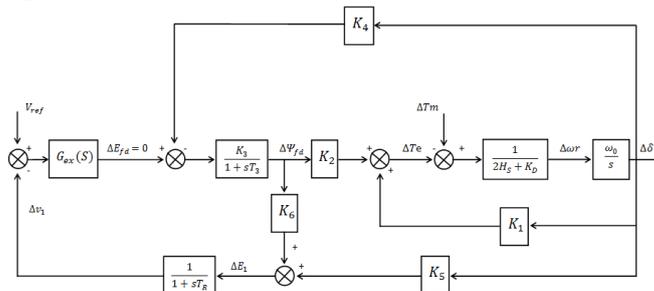
## 2. METODE PENELITIAN

Sistem penguatan atau sistem eksitasi dari sebuah mesin listrik harus dapat memberikan arus searah kepada kumparan generatornya dan dapat menjaga tegangan terminalnya konstan pada nilai yang diinginkan. *Input* untuk sistem eksitasi adalah tegangan referensi dan tegangan dari transduser. Blok *amplifier* memiliki konstanta waktu  $T_A$  dan konstanta *gain*  $K_A$ . Transduser berfungsi sebagai sensor dan mengkonversi besaran-besaran lain seperti mekanik menjadi besaran elektrik sebelum dimasukkan ke dalam sistem eksitasi. Parameter  $T_R$  melambangkan konstanta waktu tegangan terminal transduser.



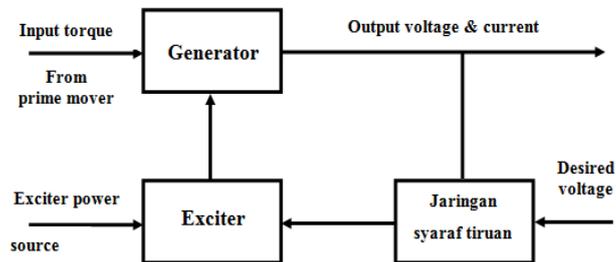
Gambar 1. Sistem Eksitasi dengan *Voltage Transducer*

Blok sistem eksitasi di atas kemudian ditambahkan ke dalam diagram blok yang dapat dilihat pada (Gambar 2). Representasi diagram blok dapat diaplikasikan untuk semua jenis eksiter, dengan  $G_{ex}(s)$  menyatakan fungsi alih dari AVR dan eksiter. (Gambar 2) merupakan diagram blok untuk SMIB dalam bentuk fungsi alih.



Gambar 2. Representasi Diagram Blok dengan Eksiter dan AVR

Generator model lama yang masih menggunakan *Automatic Voltage Regulator* sebagai kontrol eksitasinya membutuhkan perhitungan matematis permodelan generator yang cukup rumit. Pada penelitian kali ini diterapkan jaringan syaraf tiruan sebagai kontrol sistem eksitasi agar perhitungan matematis permodelan generator tidak begitu rumit dan agar mencapai nilai kestabilan generator yang baik.



Gambar 3. Blok Diagram Perancangan Sistem

### 3. HASIL DAN PEMBAHASAN

Yang pertama dilakukan adalah mengatur parameter-parameter SMIB terlebih dahulu sebelum model SMIB yang sudah dirancang dapat disimulasikan. Parameter-parameter yang digunakan dalam simulasi permodelan SMIB diambil dari beberapa referensi jurnal dan metode *trial* dan *error*. Data yang diambil dari jurnal adalah data yang paling cocok dengan rangkaian SMIB yang sudah dirancang dan menghasilkan hasil simulasi yang paling baik. Sedangkan konstanta-konstanta mesin  $K_1$ ,  $K_2$ ,  $K_3$ ,  $K_4$ ,  $K_5$ , dan  $K_6$  serta parameter  $H$ ,  $D$ , dan  $T_d$ , data diambil dari PT.Krakatau Daya Listrik. Konstanta-konstanta mesin  $K_1$  sampai dengan  $K_6$  dicantumkan pada (Tabel 1) berikut:

Tabel 1. Konstanta Mesin Sistem SMIB

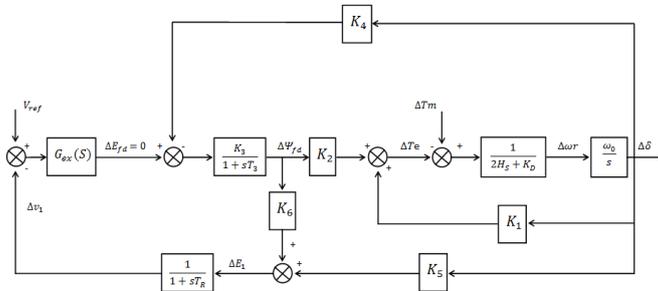
Konstanta Mesin	Nilai Konstanta
$K_1$	2,3983
$K_2$	1,4755
$K_3$	0,1921
$K_4$	1,3095
$K_5$	0,5626
$K_6$	0,0583

Selain data konstanta mesin sistem SMIB maka data-data yang didapat dari jurnal dengan metode *trial* dan *error* pun dicantumkan pada (Tabel 2).  $H$  merupakan konstanta inersia mesin dalam sistem per unit yang diperoleh dari hasil normalisasi momen inersia dan  $D$  adalah koefisien *damping* dalam sistem per unit. Selain itu terdapat juga konstanta-konstanta sistem eksitasi seperti konstanta *gain* dan waktu untuk *amplifier*  $K_A$  dan  $T_A$ , konstanta untuk eksiter  $K_E$  dan  $T_E$ , dan konstanta waktu untuk transduser  $T_R$  dari metode *trial* dan *error*.

Tabel 2. Parameter Performansi SMIB

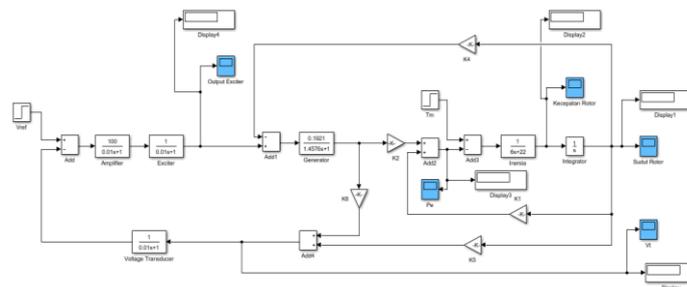
Parameter	Nilai
$H$	3
$D$	22
$T_d$	7,5881
$K_A$	100
$T_A$	0,01
$K_E$	1
$T_E$	0,01
$T_R$	0,01

Sistem eksitasi yang digunakan pada penelitian ini adalah representasi sistem eksitasi *thyristor* yang diklasifikasikan dalam sistem eksitasi tipe ST1A. Model sistem eksitasi yang digunakan telah disederhanakan untuk merepresentasikan model SMIB. Model system eksitasi tipe ST1A yang digunakan merupakan tipe sistem eksitasi dengan *gain* eksiter yang tinggi tanpa menggunakan *feedback*, sehingga dipilih  $K_A$  yang tinggi yaitu 100. Karena semakin tinggi nilai  $K_A$  untuk *gain amplifier* maka akan semakin dekat dengan terminal generator dengan  $V_{ref}$  yang telah ditentukan.



Gambar 4. Rangkaian Simulasi SMIB

Agar diperoleh hasil simulasi berupa tegangan keluaran generator  $V_t$ , daya keluaran generator  $P_e$ , kecepatan rotor  $\omega$ , sudut rotor  $\delta$ , dan keluaran eksiter  $E_{fd}$  maka parameter-parameter model SMIB diatas harus diatur terlebih dahulu.

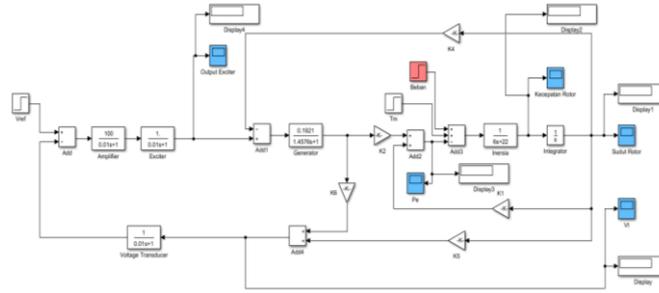


Gambar 5. Rangkaian Simulink Tanpa Beban Menggunakan Kontrol AVR

Masukan atau *input* sistem eksitasi adalah  $V_{ref}$ , yang diatur agar bernilai satu p.u sebagai referensi keluaran tegangan generator yang diinginkan, dan  $\Delta V_t$ , yang merupakan keluaran dari transduser yang bekerja sebagai sensor tegangan terminal generator dan berfungsi mengubah tegangan terminal generator menjadi tegangan DC dan mengubah besaran lain seperti kecepatan dan sudut rotor menjadi besaran elektrik.

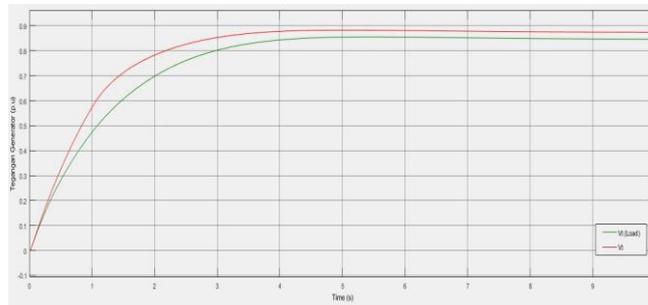
Setelah transduser mengkonversi tegangan DC kemudian dibandingkan dengan tegangan referensi  $V_{ref}$  dan sinyal *error* digunakan untuk mengendalikan arus kumparan medan utama melalui eksiter. Regulator tegangan atau *amplifier* menguatkan sinyal *error* yang dihasilkan oleh transduser kemudian keluaran regulator tegangan digunakan untuk mengendalikan pilot eksitasi dengan mengendalikan kumparan medan generator melalui eksiter.

Untuk pengujian selanjutnya, beban berupa *input step* pada masukan ke inersia mesin ditambahkan ke *simulink* dengan rangkaian SMIB tersebut. Beban *input step* diatur agar bernilai satu p.u.



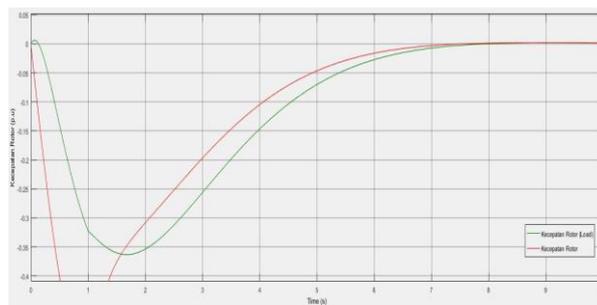
Gambar 6. Rangkaian Simulink dengan Beban Menggunakan Kontrol AVR

Sistem pembangkit diberikan masukan berupa dua sinyal *step* yang masing-masing bernilai satu p.u, untuk masukan sistem eksitasi  $V_{ref}$  dan referensi turbin  $\Delta T_m$ . Hasil tegangan keluaran  $V_t$  yang diperoleh dapat dilihat pada (Gambar 7), dibandingkan dengan tegangan keluaran generator saat sistem diberikan beban *input step* yang juga bernilai satu p.u pada masukan inersia mesinnya.



Gambar 7. Perbandingan Tegangan Keluaran Generator Tanpa Beban dan Dengan Beban

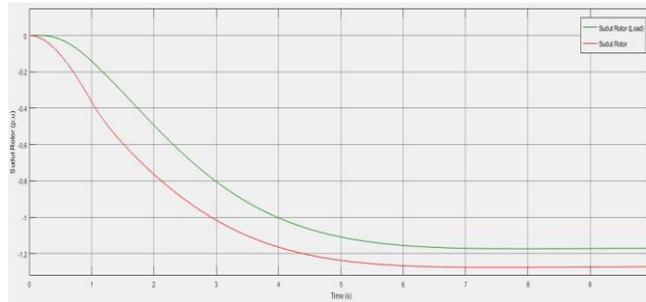
Pada gambar tersebut dapat dilihat bahwa tegangan tidak mencapai nilai yang diinginkan sesuai dengan referensi yaitu satu p.u. Nilai  $V_t$  maksimum yang diperoleh pada saat tanpa beban adalah 0,8928 p.u, sistem mengalami sedikit osilasi sampai sistem stabil pada detik ke-4 dengan nilai  $V_t$  kurang lebih 0,8739 p.u. Ketika generator ditambahkan beban maka menyebabkan tegangan terminal mengalami penurunan sebesar 0,28 p.u menjadi 0,8459 p.u. Tetapi waktu yang dibutuhkan untuk tegangan terminal berada pada titik stabil yaitu selama 4 detik.



Gambar 8. Perbandingan Kecepatan Rotor Tanpa Beban dan Dengan Beban

Gambar di atas merupakan salah satu keluaran sistem lainnya yaitu kecepatan rotor  $\omega$  yang dibandingkan ketika tanpa beban dan dengan diberikan beban, dapat dilihat pada gambar grafik mengalami osilasi selama 8 detik sebelum kecepatan rotor stabil pada nilai kurang lebih 0 p.u. Kestabilan kecepatan rotor ada pada nilai 0 karena pada prinsipnya kecepatan sinkron generator dihasilkan dari kecepatan putar rotor dengan kutub-kutub magnet yang berputar dengan kecepatan yang sama dengan medan putar stator.

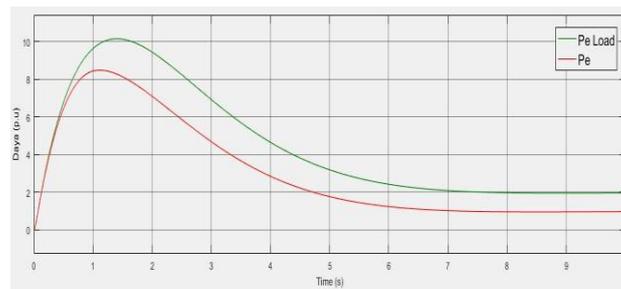
Pada saat generator diberikan beban, maka perubahan yang terjadi dapat dilihat pada osilasi yang dialami oleh kecepatan rotor, osilasi yang dialami oleh kecepatan rotor lebih kecil dibandingkan pada saat generator dijalankan tanpa beban. Waktu yang dibutuhkan kecepatan rotor pada saat generator diberikan beban untuk stabil pada nilai 0 p.u sama yaitu kurang lebih 8 detik.



Gambar 9. Perbandingan Sudut Rotor Tanpa Beban dan Dengan Beban

Gambar 9 yaitu perbandingan sudut rotor ketika generator tanpa beban dan ketika generator diberikan beban. Sudut rotor  $\delta$  dihasilkan dari persamaan kecepatan rotor,  $\Delta\delta = \omega_0\Delta\omega_r$ , dari rumus tersebut dapat dilihat bahwa hubungan sudut rotor  $\delta$  dengan kecepatan rotor  $\omega$  adalah berbanding lurus. Pada saat generator tidak diberikan beban grafik sudut rotor  $\delta$  mengalami penurunan dari *setting* nilai awal sebesar 1 p.u menjadi kurang lebih -1,2692 p.u, sedangkan waktu yang dibutuhkan sudut rotor  $\delta$  untuk mencapai nilai stabil tersebut adalah 7 detik.

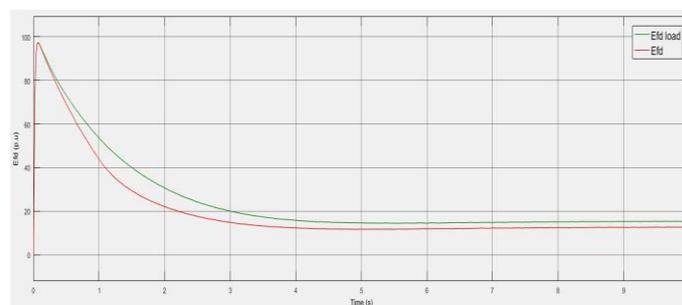
Pada saat generator diberikan beban sudut rotor mengalami kenaikan sebesar kurang lebih 0,1 p.u hingga sudut rotor stabil pada nilai -1,1667 p.u. waktu yang dibutuhkan sudut rotor ketika generator diberikan beban untuk stabil dari *setting* nilai awal sama dengan pada saat generator tidak diberikan beban yaitu selama 7 detik.



Gambar 10. Perbandingan Daya Generator Tanpa Beban dan Dengan Beban

Pada gambar 10, grafik respon daya keluaran generator  $P_e$  mengalami *overshoot* yang cukup besar dengan nilai maksimum 17,54 p.u. kemudian daya mulai mengalami penurunan dan menjadi stabil pada titik detik ke-8 dengan nilai  $P_e$  kurang lebih 1 p.u.

Pada saat generator diberikan beban daya keluaran generator  $P_e$  mengalami *overshoot* yang sedikit lebih besar dibandingkan pada saat generator tidak diberikan beban dan daya mengalami kenaikan sebesar 1 p.u sehingga nilai daya keluaran generator menjadi kurang lebih 2 p.u. waktu yang dibutuhkan untuk daya mencapai nilai yang stabil sama dengan ketika generator tidak diberikan beban yaitu selama 8 detik.



Gambar 11. Perbandingan Keluaran Eksiter Tanpa Beban dan Dengan Beban

Keluaran sistem eksitasi atau keluaran eksiter adalah tegangan medan  $E_{fd}$ . Dapat dilihat pada gambar 11,  $E_{fd}$  mengalami *overshoot* yang sangat tinggi selama 3 detik sampai akhirnya stabil pada detik ke-8 dengan nilai kurang lebih 12 p.u.

Tidak terdapat perbedaan *overshoot*  $E_{fd}$  yang terlalu jauh ketika generator tidak diberikan beban maupun pada saat generator diberikan beban  $E_{fd}$  mengalami kenaikan sebesar 3 p.u sehingga  $E_{fd}$  berada pada nilai kurang lebih 15 p.u. waktu yang dibutuhkan untuk tegangan medan  $E_{fd}$  stabil sama dengan pada saat generator tidak diberikan beban yaitu selama 8 detik.

Pelatihan jaringan yang dilakukan pada penelitian ini menggunakan metode *inverse*. Masukan sistem digunakan sebagai target atau keluaran dari jaringan, dan keluaran sistem akan digunakan sebagai masukan jaringan. Data yang digunakan untuk pelatihan didapatkan dari hasil simulasi SMIB (*single machine infinite bus*) yang telah di rangkai dan disimulasikan pada *simulink*, kemudian diperoleh empat masukan, yaitu  $P_e$ ,  $V_t$ ,  $\omega$ , dan  $\delta$  dengan masing-masing 550 sampel data dan satu keluaran yaitu  $E_{fd}$  yang juga memiliki sampel data yang sama yaitu 550 sampel data.

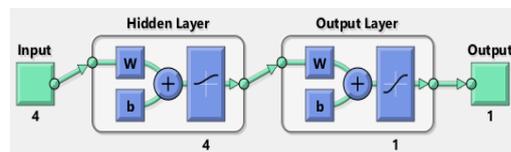
Tabel 3. *Sample Data* untuk Pelatihan

Target	Input			
$E_{fd}$	$V_t$	$P_e$	$\Omega$	$\delta$
0	0	0	0	0
0,0005596	3,64E-10	-1,30E-10	-3,35E-05	-5,61E-10
0,019922174	2,16E-08	2,12E-07	-0,0002009	-2,02E-08
0,503084843	1,66E-06	3,31E-05	-0,0010363	-5,38E-07
3,07969958	2,44E-05	0,000557641	-0,0027006	-3,67E-06
8,472384994	0,000119581	0,002834502	-0,0047759	-1,15E-05
18,57149416	0,000440942	0,010652268	-0,0077388	-3,05E-05
33,68482127	0,00129196	0,031519422	-0,0117436	-7,08E-05
52,25467835	0,003166236	0,077630618	-0,0170314	-0,000150438
70,56596918	0,006636959	0,16303629	-0,0238	-0,000296738

Data-data di atas kemudian dimasukkan ke dalam Matlab untuk dilatih. Jaringan *backpropagation* dapat di bentuk melalui fungsi *newff*, kemudian banyaknya lapisan tersembunyi (*hidden layer*) dan jumlah sel newron di lapisan tersebut ditentukan secara acak sampai mendapatkan hasil pelatihan yang optimal. Fungsi aktivasi yang dapat digunakan untuk *backpropagation* adalah fungsi yang bisa dideferensiasikan seperti *tansig*, *logsig* dan *purelin*.

Jaringan dikelompokkan berdasarkan arsitektur jaringan yaitu banyaknya lapisan tersembunyi, pada pelatihan yang telah dilakukan dicoba untuk membangun jaringan *feedforward* dengan 1 sampai dengan 4 lapisan tersembunyi.

Masing-masing pengelompokkan dilatih dengan fungsi aktivasi yang berbeda-beda secara acak hingga diperoleh hasil pelatihan terbaik. Parameter yang diubah adalah  $\mu$  yang merupakan salah satu parameter karakteristik dari *Lavenberg-Marquardt* (*trainlm*).

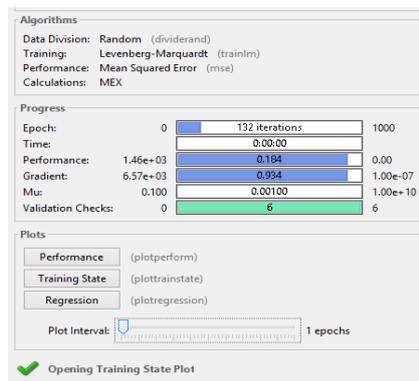


Gambar 12. Arsitektur Jaringan pada Matlab.

Gambar di atas merupakan arsitektur jaringan untuk pelatihan yang terdiri atas empat lapisan *input*, empat lapisan tersembunyi dan satu lapisan *output*. Jumlah sel neuron pada lapisan tersembunyi tidak terlalu berpengaruh jika fungsi aktivasi yang digunakan telah sesuai dengan arsitektur jaringan yang dirancang.

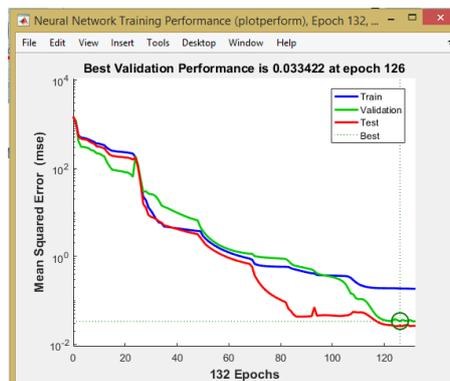
Semakin banyak sel neuron yang digunakan pada lapisan tersembunyi maka dibutuhkan komputasi yang semakin rumit untuk melatih jaringan. Sama halnya dengan banyaknya jumlah lapisan tersembunyi, semakin banyak lapisan tersembunyi maka akan membutuhkan komputasi yang lebih rumit pada saat pelatihan. Tetapi jumlah lapisan tersembunyi dapat membantu jaringan agar

lebih cepat dalam mengenali pola data yang dilatih, dan mengingat pola tersebut pada saat jaringan digunakan untuk pengujian.



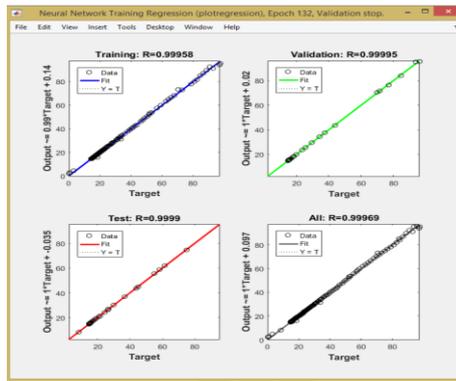
Gambar 13. *nntoolbox* Pelatihan Jaringan.

Dari gambar *nntoolbox* pelatihan di atas, proses pembelajaran berhenti saat *error* validasi mencapai 6 iterasi, yang terjadi pada iterasi pelatihan ke-132. Pada *nntoolbox* terdapat algoritma yang digunakan untuk pembelajaran jaringan, proses pembelajaran yang terdiri atas parameter-parameter pembelajaran dan plot yang berisi grafik-grafik hasil pelatihan.



Gambar 14. *Performance*.

Performansi dari pelatihan dapat di lihat dengan memilih *performance* yang ada pada jendela *nntoolbox* pelatihan jaringan. Grafik yang muncul menampilkan *error* dari pelatihan, validasi dan pengujian. Dari *input* data pelatihan sebanyak 4x550 diambil data sebanyak 10% untuk validasi, data tersebut digunakan untuk memvalidasi bahwa jaringan tersebut memiliki kemampuan untuk bergeneralisasi dan pelatihan akan berhenti jika generalisasi tidak dapat lagi ditingkatkan (jumlah kegagalan melebihi maksimum *error* validasi) atau sebelum jaringan mengalami *overfitting*. *Error* validasi akan bertambah apabila gradien pada iterasi ke- $k$  lebih besar daripada gradien iterasi ke- $(k-1)$ . Validasi terbaik atau kesalahan validasi terjadi pada iterasi ke-126 sebesar 0,033422, yang berarti tidak ada signifikan *overfitting* yang terjadi sampai iterasi tersebut.



Gambar 15. Hasil Regresi Pelatihan.

Pada jendela pelatihan *nttoolbox* juga terdapat hasil regresi dari pelatihan. Nilai regresi  $R$  digunakan untuk mengukur korelasi antara target dan *output*. regresi  $R$  bernilai satu mempunyai arti bahwa target dan *output* memiliki korelasi yang sangat dekat, dan apabila suatu variabel memiliki hubungan atau korelasi dengan variabel lainnya maka variabel tersebut dapat digunakan untuk meramal atau memprediksi variabel lain dengan hubungan serupa. Jaringan yang telah dibuat memiliki hasil regresi yang baik karena mendekati nilai satu. Baik untuk regresi hasil pelatihan, validasi ataupun pengujian. Semakin mendekati nilai satu maka jaringan tersebut dikatakan semakin baik.

Tujuan dilakukan pengujian adalah untuk mengetahui apakah jaringan yang telah dilatih dapat menghasilkan keluaran sesuai dengan target seperti pada waktu pelatihan jika jaringan diberi data masukan yang serupa sebagai masukan jaringan, yang kemudian akan dihitung persentase keberhasilan pengujian.

Pengujian dilakukan dengan data yang diperoleh dari hasil simulasi SMIB pada *simulink* yang telah ditambahkan perubahan beban berupa sinyal *step* pada masukan ke inersia mesin. Seperti halnya data yang digunakan untuk pelatihan, data yang akan digunakan untuk pengujianpun terdiri atas lima elemen. Empat elemen digunakan sebagai masukan ke jaringan syaraf tiruan yaitu,  $V_t$ , daya generator  $P_e$ , kecepatan rotor  $\omega$ , sudut rotor  $\delta$  dan satu elemen atau parameter digunakan sebagai target jaringan syaraf tiruan yaitu  $E_{fd}$ . Contoh data untuk pengujiannya pada tabel 4 berikut:

Tabel 4. Sample Data Untuk Pengujian

Target	Input			
$E_{fd}$	$V_t$	$P_e$	$\Omega$	$\delta$
0	0	0	0	0
0,019922	3,64E-10	-1,30E-10	-3,35E-05	-5,61E-10
0,670996	2,16E-08	2,12E-07	-0,0002009	-2,02E-08
3,427465	1,66E-06	3,31E-05	-0,0010363	-5,38E-07
9,215303	2,44E-05	0,000557641	-0,0027006	-3,67E-06
19,80288	0,000119581	0,002834502	-0,0047759	-1,15E-05
35,34493	0,000440942	0,010652268	-0,0077388	-3,05E-05
54,09172	0,00129196	0,031519422	-0,0117436	-7,08E-05
72,07833	0,003166236	0,077630618	-0,0170314	-0,000150438
85,49182	0,006636959	0,16303629	-0,0238	-0,000296738

Data hasil pengujian jaringan syaraf tiruan diperoleh dalam bentuk matriks seperti pada gambar 5.  $T_t$  adalah target dari pengujian jaringan syaraf tiruan dan  $z$  adalah hasil atau *output* dari pengujian jaringan syaraf tiruan. Untuk cek hasil kecocokan per-data antara target pengujian dan hasil keluaran jaringan dengan menghitung besarnya *error* antara matriks target  $T_t$  dan matriks *output*  $z$ . Matriks yang dihasilkan berukuran 550x2, masing-masing untuk target dan *output*.

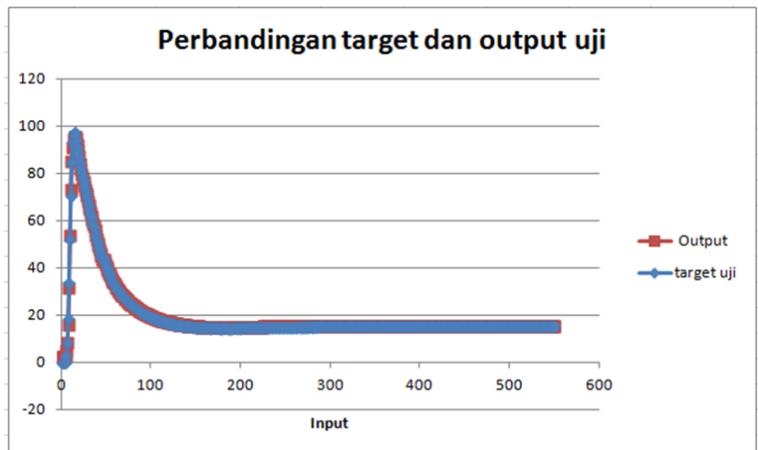
Tabel 5. Sample Data Hasil Pengujian

Data Cek ke-	Target ( $T_t$ )	Output ( $z$ )	Error
1	0	2,103681548	-2,103681548
2	0,0005596	2,125653641	-2,125094041



3	0,019922174	2,238632724	-2,21871055
4	0,503084843	2,887420468	-2,384335625
5	3,07969958	4,685177897	-1,605478317
200	14,70733422	14,68806963	0,019264587
201	14,6831799	14,69966254	-0,016482635
202	14,68275677	14,7121074	-0,029350633
203	14,6956306	14,71848545	-0,022854847
204	14,76274434	14,72471629	0,038028047
205	14,78646354	14,73470494	0,051758602
400	15,31901516	15,35916275	-0,040147587
401	15,41514605	15,35925181	0,055894237
402	15,41106176	15,3588203	0,052241462
403	15,41252111	15,35856787	0,053953237
404	15,42287437	15,35843709	0,064437281
405	15,29299107	15,35820789	-0,065216822
545	15,37008901	15,34682121	0,0232678
546	15,39690132	15,34669808	0,050203242
547	15,40066523	15,3473661	0,05329913
548	15,39341194	15,3469454	0,046466537
549	15,33393215	15,34686271	-0,012930555
550	15,3179752	15,34689416	-0,028918965

Selanjutnya data *output* jaringan dan target yang ditentukan dibuat dalam bentuk grafik untuk mengetahui perbandingannya pada gambar 16.



Gambar 16. Grafik Perbandingan Target dan *Output* Uji

Dari grafik tersebut dapat dilihat bahwa warna biru mewakili target dari pengujian dan warna merah mewakili keluaran dari jaringan. Warna merah sebagai *output* jaringan hampir keseluruhan sesuai dan menyerupai garis warna biru atau berada pada warna biru. Hal itu berarti jaringan yang diperoleh dari hasil pengujian tersebut telah dapat mengenali pola data serupa yang dijadikan *input* dan target.

Dari tabel data hasil pengujian diatas maka dapat dihitung persentase keberhasilan pengujian menggunakan persamaan sebagai berikut:

$$Persentase = \frac{Output - Error}{Target - Error} \times 100\%$$

Nilai target, output, dan error yang digunakan adalah rata-rata dari nilai target, output, dan error data hasil pengujian. Nilai rata-rata didapat dari hasil jumlah nilai atau data target, output, dan error dibagi banyaknya jumlah data. Maka di peroleh hasil data sebagai berikut:

$$\text{Rata - rata target} = \frac{11086,98}{550} = 20,15815$$

$$\text{Rata - rata output} = \frac{10700,44}{550} = 19,45534$$

$$\text{Rata - rata error} = \frac{386,5444}{550} = 0,702808$$

Selanjutnya masukkan nilai rata-rata target, output, dan error ke dalam persamaan untuk menghitung persentase keberhasilan. Maka diperoleh:

$$\text{Presentase} = \frac{19,45534 - 0,702808}{20,15815 - 0,702808} \times 100\%$$
$$\text{Presentase} = 96,387\%$$

Jadi, dari hasil perhitungan diperoleh persentase keberhasilan pengujian sebesar 96,387%.

#### 4. KESIMPULAN

##### 4.1 Kesimpulan

Berdasarkan hasil penelitian dan penulisan skripsi ini, maka dapat di buat kesimpulan sebagai berikut:

1. Pada saat generator tidak diberikan beban maka nilai  $V_t$ ,  $\delta$ ,  $P_e$ ,  $\omega$  dan  $E_{fd}$  berturut-turut adalah 0,8739 p.u, 0 p.u, 1 p.u, dan 12 p.u. Sedangkan waktu yang dibutuhkan agar semua parameter sistem berada pada titik stabil adalah 8 detik.
2. Persentase keberhasilan pengujian jaringan syaraf tiruan sebesar 96,387% dengan rata-rata *error* sebesar 0,702808 jaringan syaraf tiruan yang telah dirancang mampu mengenali pola data masukan dengan persentase keberhasilan yang cukup besar, hal ini berarti jaringan dapat mengatur sistem eksitasi pada SMIB yang telah disimulasikan untuk berada pada nilai target yang telah ditentukan.

##### 4.2 Saran

Saran untuk pengembangan penelitian selanjutnya:

1. Mengganti model *plant* yang digunakan untuk membuktikan apakah jaringan mampu beradaptasi dengan data *plant* yang berbeda.
2. Gunakan konstanta mesin dengan model generator yang berbeda.
3. Jaringan dirancang dengan menggunakan algoritma pembelajaran yang lain agar dapat dibandingkan dan diketahui yang mana yang hasilnya lebih bagus.

#### REFERENSI

- [1] Anderson Paul M, A. A. Fuad. *Power System Control and Stability*.
- [2] Kundur Prabha. *Power System Stability and Control*.
- [3] AL-Qatamin, Moath H. *An Optimal State Feedback Controller Based Neural Networks for Synchronous Generators*, *International Journal of Electrical and Computer Engineering*, Vol. 3, No. 4, Agustus 2013, pp-561-567.
- [4] Memon, Aslam P. Uqaili, M Aslam. Memon, Zubair A, dan Tanwani, Naresh K. *Suitable Feedforward Artificial Neural Network Automatic Voltage Regulator for Excitation Control System*, *Universal Journal of Electrical and Electronic Engineering* 2(2): 45-51, 2014.
- [5] Hassan, Fadhil A. dan Jawad, Helen J. *Artificial Neural Network Control of the Synchronous Generator AVR with Unbalanced Load Operating Conditions*, *Eng. & Tech. Journal*, Vol. 28, No.17, 2010.
- [6] Fauset, Laurene. 1994. *Fundamentals of Neural Networks Architectures, Algorithms and Applications*. USA: Prentice-Hal, Inc.

- [7] ANSI/IEEE Std 421.1, IEEE Standard Definition for Excitation Systems for Synchronous Machines, American National Standards Institute and The Institute of Electrical and Electronics Engineers, USA, 1986.
- [8] Wijaya, Mochtar. 2001 Dasar-Dasar Mesin Listrik. Jakarta: Djambatan
- [9] Angga (2011). Sistem Penguatan Generator, Fundamental Training PT. Krakatau Daya Listrik.

