**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Energi listrik merupakan suatu bentuk energi yang memiliki peran sangat vital dalam aktivitas keseharian manusia. Setiap tahun terjadi peningkatan kebutuhan energi yang signifikan dan menuntut pihak penyedia tenaga listrik untuk memberikan *supply* tenaga yang cukup dan berkualitas. Salah satu faktor penting dalam menghasilkan listrik yang berkualitas adalah kestabilan sistem.

Terdapat dua macam kestabilan sistem yaitu kestabilan transien dan kestabilan dinamik. Kestabilan transien adalah kemampuan sisitem mempertahankan kondisi sinkron ketika ada gangguan besar yang terjadi secara mendadak seperti hubung singkat dan pelepasan beban. Sedangkan kestabilan dinamik adalah kemampuan sistem dalam mempertahankan kondisi sinkron ketika terjadi gangguan kecil yang sering terjadi. Salah satu penyebab gangguan kecil adalah perubahan beban.

Perubahan daya pada sisi beban harus diikuti dengan perubahan daya pada pembangkit. Peningkatan permintaan daya beban harus diikuti dengan peningkatan daya pembangkitan. Tanpa penyesuaian, maka kenaikan beban akan mengakibatkan penurunan kecepatan putaran rotor generator yang berdampak pada berubahnya frekuensi dan tegangan keluaran generator pada sistem.

Sistem kelistrikan Lombok terdiri dari beberapa pembangkit listrik yang tersebar. Setiap pembangkit akan terhubung pada suatu bus yang besar yang pada umumnya disebut bus *infinite* dimana pada sistem kelistrikan Lombok adalah bus 20 kV. Suatu sistem pembangkit yang terhubung pada bus *infinite* di sebut *single machine infinite bus* (SMIB). Pada suatu sistem pembangkitan SMIB, perubahan beban akan menyebabkan kecepatan sudut rotor berubah. Perubahan tersebut menyebabkan sistem pengendalinya mengatur kembali kecepatan sudut rotor tersebut kembali ke kecepatan sinkronnya. Oleh karena itu, kondisi perubahan beban tersebut perlu diredam. Peredaman dilakukan untuk mengembalikan kecepatan sudut rotor kembali ke nilai nominalnya akibat adanya gangguan berupa perubahan beban.

*Power Sistem Stabilizer* (PSS) merupakan peralatan tambahan yang dapat memberikan redaman tambahan terhadap osilasi kecepatan sudut rotor. Untuk menghasilkan redaman yang baik, parameter-perameter PSS tersebut harus disesuaikan dengan kondisi sistem tenaga yang akan dipasangkan PSS.

Perkembangan sistem tenaga listrik yang semakin cepat dan masalah yang semakin beragam, menuntut penyelesaian waktu nyata. Oleh karena itu, pada penelitian ini diusulkan menerapkan PSS berbasis jaringan saraf tiruan berulang (*recurrent neural network*, RNN) untuk menggantikan PSS konvensional pada sistem tenaga yang dapat merespon perubahan performa sistem secara langsung. Pengaplikasian RNNPSS dititik beratkan pada kinerja RNNPSS terhadap osilasi perubahan kecepatan sudut dan perbaikan performa sistem dilihat dari *overshoot* dan *settling time*.

* 1. **Rumusan Masalah**

Permasalahan yang dihadapi dalam penelitian ini adalah

1. Bagaimana *power system stabilizer* berbasis jaringan syaraf tiruan berulang (*recurrent neural network*, RNN) digunakan untuk menggantikan *power* *system stabilizer* (PSS) konvensional pada kestabilan dinamik SMIB?
2. Bagaimana respon pemasangan RNNPSS terhadap perubahan beban sistem?
3. Bagaimana perbandingan performa RNNPSS dengan PSS konvensional terhadap perubahan beban sistem?
4. Bagaimana memodelkan sistem kelistrikan Lombok menjadi SIMB yang terhubung ke bus 20 kV?
   1. **Batasan Masalah**

Ruang lingkup masalah yang diteliti dalam penelitian ini adalah sebagai berikut:

1. Penekanan penelitian adalah *power system stabilizer* berbasis jaringan syaraf tiruan berulang (RNNPSS) digunakan untuk menggantikan *power system stabilizer* (PSS) konvensional pada kestabilan dinamik sistem tenaga.
2. Sistem yang digunakan adalah generator tunggal yang terhubung di *infinite* bus.
3. Beban diasumsikan statik, dengan asumsi ini, beban yang terhubung dapat dianalogikan sebagai suatu beban dalam bentuk admitansi yang dalam analisisnya dapat disetarakan dengan admitansi jaringan.
4. Sistem kelistrikan Lombok dimodelkan dalam bentuk SMIB yang terhubung ke bus 20 kV dengan mengambil 1 pembangkit dan 2 saluran.
5. Gangguan yang diberikan pada *plant* berupa ganggguan kecil (perubahan beban).
   1. **Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Penelitian ini ditujukan mengetahui performa sistem setelah diaplikasikan RNNPSS.
2. Untuk mengetahui respon pemasangan RNNPSS terhadap perubahan beban sistem pada SMIB sistem kelistrikan Lombok.
3. Untuk membandingkan unjuk kerja sistem tenaga listrik yang memakai RNNPSS dengan yang tidak menggunakan RNNPSS melalui *overshoot* dan *settling time* sistem.
   1. **Manfaat Penelitian**

Manfaat-manfaat yang diharapkan dengan diadakannya penelitian ini antara lain:

1. Dapat menjadi acuan dalam meningkatkan stabilitas sistem tenaga.
2. Sebagai pembanding pada sistem yang terpasang RNNPSS dengan yang tidak terpasang RNNPSS.
3. Meningkatkan wawasan tentang studi stabilitas melalui penerapan RNNPSS.
   1. **Sistematika Penulisan**

Dalam penyusunan laporan penelitian ini, penulis menggunakan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini secara ringkas dibahas latar belakang penulisan, perumusan masalah, batasan masalah, tujuan penelitian, manfaat yang diharapkan, dan sistematika penulisan.

BAB II DASAR TEORI

Memuat tentang tinjauan pustaka yang menjabarkan hasil penelitian sebelumnya yang berkaitan dengan penelitian ini dan landasan teori yang menjabarkan teori-teori penunjang yang berhubungan dengan penelitian ini.

BAB III METODE PENELITIAN

Memuat tentang metode yang digunakan dalam melakukan penelitian,alat dan bahan yang digunakan pada waktu pelaksanaan penelitian, langkah-langkah yang dilakukan, rencana dan jadwal pelaksanaan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas hasil simulasi beserta analisis respon perubahan deviasi kecepatan sudut, perubahan sudut rotor, dan perubahan tegangan dari *plant* yang dikontrol menggunakan RNNPSS, PSS konvensional, dan sistem tanpa kontrol pada sistem SMIB.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil simulasi, analisa dan pembahasan pada tugas akhir ini. Serta diberikan saran-saran untuk pengembangan berikutnya.

**BAB II**

**DASAR TEORI**

Dasar teori berikut terdiri dari tinjauan pustaka dan landasan teori. Tinjauan pustaka memuat tinjauan terhadap beberapa penelitian sebelumnya yang berhubungan dengan tinjauan stabilitas sistem tenaga. Dasar teori merupakan teori pendukung dalam memahami penelitian ini.

**2.1 Tinjauan Pustaka**

Stabilisator sistem tenaga berbasis jaringan syaraf tiruan berulang pada sistem mesin tunggal. (Aribowo, 2010). Penelitian ini menggunakan jaringan syaraf tiruan berulang (RNN) untuk mendesain stabilisator sistem tenaga karena mempunyai keunggulan bahwa keluarannya tidak hanya bergantung pada masukan saat ini, tetapi juga tergantung pada kondisi masukan waktu balik sebelumnya. RNNPSS dapat menangkap respon dinamik dari sistem tanpa waktu tunda umpan balik eksternal karena neuron berulang mempunyai umpan balik internal. RNNPSS yang digunakan dalam penelitian terdiri dari dua komponen utama yaitu RNN-*identifier* dan RNN-*controller*. RNN-*identifier* berfungsi untuk mengetahui karakteristik dinamika sistem, sedangkan RNN-*controller* berfungsi untuk meredam osilasi frekuensi rendah. Generator dimodelkan dalam bentuk *Philips Heffron,* data pelatihan untuk meredam osilasi kecepatan adalah data keluaran *plant* berupa perubahan kecepatan sudut. Fungsi aktifasi yang digunakan adalah *tansig*, dan *purelin*. Jumlah neuron yang digunakan adalah 5. Fungsi pelatihan yang diterapkan adalah *trainlm*. Struktur jaringan yang digunakan pada pelatihan ini adalah terdiri dari tiga lapis *layer,* yaitu lapis masukan, lapis tersembunyi, dan lapis keluaran. Dari hasil simulasi didapatkan bahwa RNNPSS mampu memperbaiki performansi sistem generator di tempat RNNPSS dipasang. Keberhasilan dari desain stabilisator sistem tenaga jaringan syaraf tiruan berulang (RNNPSS) ini sangat bergantung data dan proses pembelajaran yang benar.

Desain adaptive PSS berbasis *artificial neural network* (ANN) dan PID *controller* menggunakan *imprealist competitive algorithm* (ICA). (Wibisono, dkk, 2011). Penelitian ini diperkenalkan desain PSS baru yang disebut dengan PSS adaptif. PSS adaptif ini memiliki kemampuan untuk mempertimbangkan sifat alami dari sistem tenaga listrik yang sangat tidak linear. PSS adaptif yang menggunakan *neural network* ini digunakan untuk meredam osilasi akibat gangguan pada sistem. *Neural network* yang digunakan terdiri dari dua yaitu *neuro identifier* dan *neuro controller*. *Neuro identifier* bertugas untuk mengindetifikasi perilaku *plant*, sedangkan *neuro controller* berfungsi untuk menghasilkan sinyal kontrol pada *input*. Pelatihan yang digunakan adalah pelatihan *off-line* dan *online.* Pelatihan *off-line* diterapkan pada *neuro identifier* dan *neuro controller* tujuannya adalah untuk menghindari kemungkinan terjadi ketidakstabilan diawal operasi *online* karena sistem yang masih belum terlatih. Karena pada operasi *off-line* ini *neuro controller* belum mampu menghasilkan sinyal kontrol, maka sinyal kontrol yang dihasilkan oleh *neuro controller* digantikan oleh *pseudo random binary sequence* (PRSB). Setelah menjalani proses pelatihan secara *off-line*, maka *neuro identifier* telah memiliki kemampuan untuk meniru perilaku *plant* secara sempurna dan *neuro controller* telah mampu menghasilkan sinyal kontrol yang dapat membuat sistem menjadi stabil terhadap adanya PRBS. *Controller* PID dipakai untuk meningkatkan respon dinamik dan juga untuk menghilangkan ataupun mengeleminasi *steady state* *error*. PID ditambahkan pada sistem eksitasi. Nilai parameter *controller* PID ini dioptimalkan dengan algoritma ICA. Dari hasil simulasi didapat bahwa PSS adaptif berbasis *neural networks* dapat digunakan untuk meredam osilasi sistem tenaga listrik. PSS adaptif ini memberikan respon yang baik jika dibandingkan PSS konvensional. Dengan penambahan PID *controller* yang dioptimalkan dengan ICA terbukti memperkecil nilai *overshoot* hingga 0,08% dan mempercepat *settling time* hingga sebesar 5%.

Dari tinjauan pustaka diatas peneliti melakukan desain penerapan PSS berbasis *recurrent neural network* untuk memperbaiki deviasi kecepatan sudut rotor akibat perubahan beban yang terjadi pada SMIB. Model penerapan RNNPSS pada *plant* dibuat dalam bentuk *simulink* sedangkan untuk pelatihan *recurrent neural network* dibuat dalam bentuk *m.file* pada *software Matlab* versi 7.8.0.347 (R2009a).

**2.2 Landasan Teori**

**2.2.1 Sistem Tenaga Listrik**

Sistem tenaga listrik didukung tiga bagian utama yaitu sistem pembangkitan, jaringan transmisi dan distribusi, dan beban seperti Gambar 2.1.



Gambar 2.1 Sistem *single machine infinite bus*

Gambar 2.1 menunjukan transfer energi listrik dari pusat pembangkit ke bus *infinite* melalui jaringan transmisi dan distribusi. Jika terjadi perubahan permintaan daya pada sisi beban, maka terjadi perubahan pada daya pembangkitan yang berdampak pada perubahan tegangan dan frekuensi. Perubahan ini akan menyebabkan pengendali pada pembangkit akan mengatur kembali besaran-besaran input guna mencapai titik keseimbangan baru. Pengendali tersebut digambarkan seperti Gambar 2.2 yang memperlihatkan proses transfer energi dari energi fosil/air menjadi besaran listrik seperti daya, arus, dan tegangan.



Gambar 2.2 Sistem pembangkit tenaga listrik

Dari Gambar 2.2 dapat dijelaskan bahwa energi yang dibawa oleh bahan bakar menggerakan *prime over* yang akan memutar generator untuk menghasilkan energi listrik. *Governor* adalah pengendali yang berfungsi untuk menstabilkan nilai torsi mekanik yang menjadi masukan bagi generator. Bila pada perubahan kecepatan putar, *governor* berfungsi sebagai umpan-balik untuk mengembalikan kecepatan putar ke kecepatan nominalnya. Sistem eksitasi adalah pengendali pada generator yang berfungsi untuk mengontrol keluaran generator seperti tegangan, dan arus. Bila terjadi perubahan pada besaran-besaran tersebut, maka eksitasi akan berfungsi sebagai umpan-balik untuk mencapai nilai kesetimbangan baru.

**2.2.2 Dinamika Sistem Tenaga Listrik**

Sistem tenaga modern dipresentasikan oleh sebuah sistem interkoneksi yang sangat tergantung pada kontrol untuk memanfaatkan secara optimal sumber daya yang ada. Sumber yang dapat diperbarui dan ekonomi energi listrik merupakan faktor penentu perkembangan Industri yang bisa meningkatkan standar hidup masyarakat. Sejak revolusi Indusri, kebutuhan energi listrik meningkat tajam. Sebagian besar energi dibutuhkan oleh masyarakat modern disuplai dalam bentuk energi listrik. Peningkatan kebutuhan energi listrik yang sebanding dengan keterbatasan antara sumber daya dan lingkungan merupakan tantangan yang harus menimbulkan usaha untuk meningkatkan kemampuan jaringan transmisi dalam pencarian solusi teknologi terbaik. Perkembangan rekayasa kontrol yang sangat cepat cenderung mengatur penyaluran daya pada saluran transmisi daya listrik sesuai kebutuhan dan variasi dari waktu ke waktu menggunakan kontrol terpadu dan optimal.

Dinamika sistem tenaga menjadi faktor penting untuk memenuhi operasi sebuah sistem. Hal itu dipengaruhi oleh komponen-komponen dinamika sistem seperti generator, jaringan transmisi, beban dan peralatan kontrol yang lain.

**2.2.3 Kestabilan Sistem Tenaga Listrik**

Stabilitas sistem tenaga listrik telah menjadi perhatian utama dalam sebuah sistem operasi. Perhatian itu muncul dari fakta bahwa pada kondisi keadaan mantap (*steady state*), kecepatan rata-rata untuk semua generator harus sama. Kondisi tersebut dinamakan pada operasi sinkron dari sebuah sistem yang interkoneksi. Gangguan kecil atau besar pada sistem tenaga berdampak pada operasi sinkron. Dengan demikian diharapkan stabilitas sistem akan menuju ke keadaan mantap dalam waktu singkat setelah gangguan menghilang.

Gangguan dapat dibagi menjadi 2 kategori, yaitu gangguan kecil dan gangguan besar. Gangguan kecil merupakan satu dari elemen sistem dinamik yang dapat di analisa dengan menggunakan persamaan linear (analisis sinyal kecil). Gangguan kecil yang terjadi berupa perubahan beban pada sisi beban atau pembangkit secara acak, pelan, atau bertingkat. Gangguan yang menghasilkan kejutan tiba-tiba pada tegangan bus adalah jenis gangguan besar yang harus dihilangkan secepatnya. Jika tidak dihilangkan secepatnya, gangguan itu akan sangat mempengaruhi kestabilan sistem. Tidak hanya besar gangguan, waktu gangguan juga berpengaruh terhadap kestabilan sistem (Stevenson, 1983).

**2.2.3.1 Kestabilan Dinamik Sistem Tenaga Listrik**

Stabilitas dinamik sistem tenaga listrik adalah kestabilan sistem tenaga listrik akibat perubahan beban kecil. Penyebutan stabilitas dinamik untuk membedakannya dengan stabilitas transient dalam bentuk hubung singkat, tegangan surja, dan abnormal *overload*. Perubahan kecil pada beban (pelanggan) akan menyebabkan perubahan kecepatan rotor dan perubahan tegangan terminal generator. Kecepatan rotor generator akan berayun disekitar kecepatan sinkron dan tegangan generator berayun disekitar harga nominalnya.

Suatu generator dikatakan stabil bila setelah gangguan (perubahan beban), kecepatan rotor kembali ke kecepatan sinkron dan tegangan konvergen ke nilai tertentu sekitar nilai nominal. Dalam sistem tenaga elektromekanik dikenal besaran daya mekanik (*P*m - dari *prime over*) dan daya elektrik (*P*e – dari generator). Dalam studi kestabilan, kedua besaran tersebut digunakan dalam bentuk ekuivalennya yaitu torsi ( torsi mekanik, *T*m dan torsi elektrik, *T*e). Walaupun dalam bentuk aktual daya mekanik dengan torsi mekanik berbeda, tetapi dalam besaran per unit-nya, kedua besaran tersebut memiliki nilai yang sama. Untuk selanjutnya, pembahasan kestabilan akan menggunakan kedua besaran tersebut.

Dalam besaran torsi, dirumuskan hubungan kedua torsi tersebut sebagai berikut:

Dimana

*J* : momen inersia rotor generator

*ωr* : kecepatan sudut rotor

*Ta*  : torsi kecepatan

*Tm*: torsi mekanik (dari *prime over*)

*Te*: torsi elektrik (karena beban generator)

Dari Persamaan (2.1) didapatkan bahwa perbedaan antara torsi yang dibangkitkan dengan torsi yang diserap (digunakan) dapat menyebabkan percepatan atau perlambatan rotor. Hubungan tegangan terhadap kecepatan rotor disebutkan:

(2.2)

Dimana

*E* : tegangan induksi generator

*C* : konstanta

*n*  : kecepatan rotor

*Ø*  : fluks magnetik

Dari Persamaan (2.2) didapatkan bahwa perubahan kecepatan rotor juga menyebabkan perubahan tegangan, dimana diketahui salah satu sebab perubahan kecepatan adalah torsi mekanik dan torsi elektrik yang tidak sebanding.

Kembali pada Persamaan (2.1), dengan penurunan yang panjang yang tidak ditunjukkan pada bagian ini didapatkan dalam Persamaan (2.3):

Dengan,

*H* : konstanta inersia

*KD* : faktor redaman

*δ* : posisi sudut

Dari Persamaan (2.3) terbaca bahwa perubahan kecepatan generator dapat dipengaruhi oleh selisih daya dan torsi mekanik dengan elektrik, yang kemudian akan berdampak pada perubahan kecepatan dan perubahan sudut rotor.

**2.2.3.2 Kestabilan Transien**

Kestabilan transien adalah kemampuan sistem tenaga untuk mencapai kondisi stabil operasi baru yang dapat diterima setelah sistem mengalami gangguan besar. Analisis kestabilan transien menggunakan pendekatan model nonlinear. Kestabilan transien pada sistem tenaga adalah respon *output* yang mencapai kondisi operasi *steady state* yang diizinkan dan sistem yang dapat kembali ke posisi semula pada saat sistem mengalami gangguan. Kestabilan transien merupakan fungsi dari kondisi operasi dan gangguan.

Stevenson mengatakan bahwa secara konsep kestabilan *steady state* dan dinamik adalah sama. Hal penting lain yang harus dicatat adalah walaupun sistem beroperasi dalam kondisi yang tidak stabil secara transien, tetapi kestabilan sinyal kecil pada sistem perlu dijaga setiap waktu. Secara umum, kestabilan tergantung pada beban sistem. Peningkatan beban dapat memicu ketidakstabilan. Hal itu menunjukkan bahwa menjaga kestabilan sistem merupakan hal penting meskipun berada di bawah kondisi beban berat.

**2.2.4 Pemodelan Sistem Tenaga Listrik**

Sistem tenaga listrik terdiri dari bagian-bagian yang kompleks, seperti: pembangkit, sistem transmisi, saluran distribusi, beban yang dinamik dan lain-lain. Antara bagian yang satu dengan bagian yang lain saling berinteraksi. Kondisi beban yang dinamik dapat mempengaruhi putaran generator, tegangan dan variabel *state* sistem yang lain.

**2.2.4.1 Model Linear Dinamika Sebuah Mesin Sinkron**

Dinamika dasar sebuah generator sinkron ditunjukkan pada Gambar 2.3.



Gambar 2.3 Dinamika dasar generator sinkron

Suatu generator agar menghasilkan energi listrik, membutuhkan dua masukan yaitu torsi dan fluksi. Masukan pertama adalah torsi mekanik. Torsi ini berfungsi memutar rotor generator sinkron. Masukan kedua adalah fluksi magnetik dari rangkaian medan. Belitan penghasil fluksi rangkaian medan terletak di rotor generator sinkron. Dengan adanya torsi mekanik dari turbin, medan fluksi yang dihasilkan oleh rangkaian medan akan berputar. Fluksi putar ini akan memotong belitan konduktor pada stator, sehingga timbul tegangan di ujung-ujung belitan stator.

Adanya pembebanan pada generator menimbulkan pengurangan fluksi yang di hasilkan rangkaian medan, sehingga menurunkan tegangan terminal generator. Gejala ini dinamakan reaksi jangkar. Untuk mempertahankan nilai fluksi dan tegangan terminal generator, dipasang a*utomatic voltage regulator* (AVR), suatu perangkat yang membandingkan tegangan yang diinginkan terhadap tegangan aktualnya, kemudian mengatur rangkaian medan agar mendapatkan nilai tegangan dan fluksi yang konstan.

Fenomena-fenomena diatas dapat dinyatakan dengan persamaan matematik. Hasil penurunan tersebut yang sudah dalam bentuk diagram blok fungsi transfer tampak pada Gambar 2.4.

Sumber: (Padiyar,1996)

Gambar 2.4 Diagram blok dinamika gerak mesin

Blok A pada Gambar 2.4 menyatakan dinamika gerak mekanik generator. Fungsi transfer dalam blok ini berdasarkan Persamaan (2.3). Blok B menyatakan proses yang terjadi pada AVR. Blok C menyatakan hubungan tegangan generator dengan tegangan pada rangkaian medan. Reaksi jangkar diwakili oleh umpan balik *K*4, dan pengaruh perubahan tegangan terminal oleh umpan balik *K*5. Sedangkan variabel- variabel *K*1 sampai dengan *K*6 merupakan variabel transfer yang besarnya tergantung dari konfigurasi jaringan dan pembebanan.

Proses gerak mekanik generator sinkron digambarkan oleh diagram blok A pada Gambar 2.4. Bila digambarkan terpisah adalah sebagai berikut:



Gambar 2.5 Diagram blok gerak mekanik dari mesin sinkron

Jika beban sistem berubah, maka terjadi perubahan pada kecepatan rotor *∆ω* dan sudut rotor *∆δ*. Perubahan *∆δ* akan menimbulkan torsi yang dinamakan torsi sinkronisasi *T*s yang sefasa dengan *δ*, dengan *K*1 dinamakan koefisien torsi sinkronisasi.

*∆T*s *= K*1 *∆δ* (2.4)

Perubahan *∆ω* akan menimbulkan torsi yang dinamakan torsi redaman (*TD*) mendahului *T*s dengan sudut sebesar 90°. Hubungan fasa antara *T*D*, T*s*, T*e terlihat pada Gambar 2.6.



Gambar 2.6 Hubungan fasa torsi redaman dan torsi sinkronisasi

Pada Gambar 2.7 terlihat adanya osilasi yang terjadi saat generator berusaha menyesuaikan diri terhadap perubahan beban. Besar nilai frekuensi osilasi dari *∆δ* pada Gambar 2.7 dapat dihitung dengan cara membentuk persamaan karakteristik dari blok Gambar 2.5 yaitu:

Dari Persamaan (2.5) kemudian menghitung frekuensi osilasi natural (*ωn*) yaitu dengan mengasumsikan *D* = 0;

Hubungan nilai *K*1 dan nilai *D* dapat dilihat Gambar 2.6.



Sumber: (Kundur, 1993)

Gambar 2.7 Pengaruh *∆TD* dan *∆TS* terhadap *∆δ*

Kemudian menghitung frekuensi osilasinya dengan menyertakan *D* > 0, maka osilasi akan teredam dengan frekuensi:

Dimana:

Dengan nilai koefisien torsi sinkronisasi dan koefisien torsi redaman analisis kestabilan suatu mesin menjadi lebih mudah karena setiap umpan balik lingkar tertutup selalu menghasilkan torsi sinkronisasi dan torsi redaman tertentu. Jumlah total koefisien torsi sinkronisasi dan torsi redaman dari setiap lingkaran tertutup akan mempunyai nilai positif dan negatif.

**2.2.4.2 *Single Machine Infinite Bus* (SMIB)**

Sistem s*ingle machine infinite bus* (SMIB) me­­­­­­ru­­pakan sistem satu mesin yang menyalurkan daya ke bus tidak terhingga (*infinite bus)*. Dikatakan tidak te­­r­­hingga karena jarak antara mesin dengan beban dianggap sangat jauh, sehingga tegangan di bus tidak terhingga diasumsikan konstan.

.

Gambar 2.8 *Single Machine Infinite Bus*

Sistem tersebut terdiri dari generator sinkron yang dihubungkan dengan *infinite bus* melalui saluran transmisi dengan reaktansi *Xe*. Generator dilengkapi dengan AVR ( *automatic voltage regulator*), *Exciter* dan PSS (*power system stabilizer*).

Dalam mempermudah pemodelan sistem menggunakan model linear. Model *Heffron-Philips* merupakan salah satu model yang umum dipakai dalam pemodelan sistem. Pemodelan *Heffron-Philips* ditunjukkan pada Gambar 2.9.



Gambar 2.9 Model *linear* SMIB

Dari Gambar 2.9 diatas, dapat dilihat bahwa pemodelan sistem SMIB diatas yang sudah terpasang PSS. Masukan dari dari PSS adalah kecepatan sudut yang dianggap sebagai sinyal *error* sedangkan keluarannya adalah tegangan yang di kirim ke eksitasi untuk memperbaiki redaman.

**2.2.5 Masalah Stabilitas Sistem Generator Sinkron**

Permasalahan kestabilan mesin sinkron didasarkan pada torsi redaman dan torsi sinkronisasi yang timbul serta pengaruhnya terhadap kestabilan sistem generator sinkron dapat dilihat pada uraian berikut.

Perubahan torsi elektrik:

Dengan,

*∆Te*total dapat dinyatakan dengan,

Dengan,

A = + α

Diagram berikut ini secara khusus menggambarkan kelakuan dinamik rotor dari generator sinkron.



Gambar 2.10 Blok diagram model dinamika gerak rotor dengan memasukan efek loop reaktif

Dengan fungsi transfer:

Kemudian dalam bentuk umum (standar):

Dari *loop* A menghasilkan torsi sinkronisasi, dari *loop* B menghasilkan torsi redaman. Persamaan karakteristik loop mekanik:

Dengan,

Sehingga akar-akar persamaan karakteristiknya (nilai eigen) adalah:

Dari akar-akar karakteristik (nilai eigen) dapat dilihat sifat kestabilan sistem. Sistem stabil jika bagian *riil* nilai eigennya berharga negatif atau *ζωn* > 0, kondisi ini dipenuhi jika:

(*D*+*B*) > 0 ; torsi redaman (*T*D)bernilai positif

*A* > 0 ; torsi sinkronisasi (*T*s) bernilai positif

Akan timbul masalah jika *T*D dan *T*s sangat kecil atau negatif, dan sudut daya akan bertambah terus tidak mencapai konvergen dan amplitudo osilasi sistem semakin membesar sehinggga sistem tidak stabil. Dapat disimpulkan bahwa:

ζ > 0 maka sistem stabil

ζ < 0 maka sistem tidak stabil

**2.2.6 Pemodelan Mesin Tunggal**

**2.2.6.1 Model Beban Pada Generator**

Penerapan persamaan ayunan mesin sinkron pada analisis sinyal kecil, didapatkan:

 (2.16)

Atau dalam kondisi perubahan kecepatan yang kecil:

 (2.17)

Transformasi Laplace dari Persamaan (2.17) didapatkan:

 (2.18)

Persamaan di atas dapat dinyatakan dalam diagram blok berikut:



Gambar 2.11 Blok diagram inersia generator

Beban pada sistem tenaga listrik terdiri dari berbagai komponen elektris. Untuk beban resistif seperti beban untuk penerangan dan pemanasan, daya listrik yang dibutuhkan tidak begantung pada frekuensi. Motor merupakan beban yang sensitif terhadap frekuensi. Tingkat sensitifitas beban ini terhadap frekuensi dipengaruhi oleh kombinasi dari karakteristik kecepatan-beban dari semua beban yang ditanggung. Karakteristik kecepatan-beban dari keseluruhan beban dirumuskan dengan:

 (2.19)

Dengan  merupakan perubahan beban yang tidak sensitif terhadap frekuensi, dan  merupakan perubahan beban yang bergantung pada frekuensi. D merupakan perubahan beban dalam persen dibagi dengan perubahan frekuensi dalam persen. Model dari pembebanan dalam generator ditunjukan pada Gambar 2.12 berikut:



Gambar 2.12 Blok diagram model pembebanan pada generator

**2.2.6.2 Model *Prime Mover***

Sumber energi mekanik yang digunakan untuk memutar turbin generator secara umum disebut sebagai *prime mover*. Model *prime mover* yang paling sederhana dari turbin uap yang tidak dipanaskan kembali (nonreheat) dapat dinyatakan dengan *single time constant* *TT* yang menghasilkan *transfer function* sebagai berikut:

 (2.20)

**2.2.6.3 Model Governor**

Ketika beban elektrik generator tiba-tiba meningkat, energi listrik yang diperlukan akan melebihi daya mekanik yang diinputkan. Kekurangan daya ini diatasi oleh energi kinetik yang tersimpan pada sistem yang berputar. Pengurangan energi kinetik menyebabkan kecepatan turbin dan frekuensi pada sistem menurun.

Perubahan kecepatan akan disensor oleh governor yang akan bekerja untuk mengatur katup masukan pada turbin sehingga daya mekanik yang dikeluarkan oleh turbin akan berubah. Sistem ini akan membawa kecepatan generator berada pada kondisi *steady state* yang baru. Pemodelan governor ditunjukkan pada Gambar 2.13



Gambar 2.13 Blok diagram dari sistem governor pada turbin uap

**2.2.6.4 Model *Amplifier***

*Amplifier* pada sistem eksitasi bisa jadi merupakan *amplifier* magnetik, *amplifier* berputar (*rotating amplifier*), atau *amplifier* modern yang menggunakan rangkaian elektronik. *Amplifier* dinyatakan dengan sebuah *gain* *K*A dan sebuah *time* konstan *T*A. *transfer function* dari *amplifier* ditunjukkan dalam Persamaan (2.21) berikut:

 (2.22)

Umumnya nilai dari *K*A berada pada range 10-400. Sedangkan *time constant* bernilai sangat kecil, yaitu pada range antara 0,002-0,1 s.

**2.2.6.5 Model Eksitasi**

Terdapat banyak tipe eksitasi. Sistem eksitasi modern menggunakan sumber listrik AC yang dilewatkan pada *rectifier* *solid state* seperti SCR. Tegangan keluaran dari eksitasi merupakan suatu fungsi yang non linear. Hal ini disebabkan oleh adanya efek saturasi pada rangkaian medan. Oleh karena itu, tidak ada hubungan yang sederhana antara tegangan terminal dan tegangan medan pada eksitasi. Terdapat banyak model dengan berbagai derajat kepuasan yang telah dikembangkan dan tersedian pada publikasi IEEE. Pemodelan eksitasi modern yang paling umum digunakan adalah pemodelan linear, yang menggunakan *time* konstan dan mengabaikan saturasi atau sifat nonlinear lainnya. Dalam bentuk yang paling sederhana, *transfer function* dari eksitasi ini dinyatakan oleh satu *time constant* *TE* dan *gain* *KE*.

 (2.23)

**2.2.6.6 Model Generator**

Mesin sinkron menghasilkan emf sebagai fungsi dari kurva magnetisasi, dan tegangan terminal bergantung pada beban generator. Pada model linear, *transfer function* yang menghubungkan tegangan terminal generator dengan tegangan medannya dapat dinyatakan dengan *gain* *KG* dan *time constant* *TG*. dan *transfer function*-nya dapat dinyatakan sebagai berikut:

 (2.24)

**2.2.6.7 Model Sensor**

Tegangan disensor melalui *Potential Transformer* (PT) dan disearahkan dengan menggunakan *bridge rectifier*. Sensor dimodelkan dengan *transfer function* orde satu yang sederhana sebagai berikut:

 (2.25)

Dengan TR merupakan bilangan yang sangat kecil, nilainya berkisar antara 0,01 sampai 0,06 s. Blok AVR dari model yang telah dijabarkan di atas ditunjukkan dalam Gambar 2.14 berikut:

 Gambar 2.14 Blok diagram *automatic voltage regulator*

**2.2.7 *Power System Stabilizer***

PSS merupakan perangkat sistem tenaga tambahan yang terhubung pada sistem kontrol pembangkit melalui port yang telah tersedia pada unit AVR pada sistem eksitasi suatu pembangkit. PSS adalah suatu piranti yang memiliki fungsi transfer umum sebagai berikut:

Kpss adalah konstanta penguatan PSS yang dapat diatur nilainya.

Dengan fungsi alih diatas, PSS merupakan piranti yang dapat diatur besaran dan fasanya. Besaran dapat diatur melalui *K*pss, sedangkan fasanya dapat diatur melalui parameter *T*k. Dengan sifat seperti ini, PSS dapat digunakan untuk membangkitkan torsi redaman positif. Karena torsi redaman sefasa dengan kecepatan, maka untuk menghasilkan torsi redaman, PSS harus menggeser fasa sinyal masukan menjadi sefasa dengan kecepatan. Jenis-jenis masukan sinyal PSS dapat berupa *Δω* atau *Δδ* (Robandi, 2008).

**2.2.7.1 Metode Penalaan PSS Ideal**

Misalnya suatu PSS dengan masukan daya (*∆P*) dipasang pada mesin dari suatu sistem. Diagram blok rangkaian PSS seperti pada Gambar 2.15.



Gambar 2.15 Diagram Blok Pemasangan PSS

Agar dapat menghasilkan torsi redaman maka fasa sinyal masukan *∆P* harus digeser searah dengan fasa kecepatan. Untuk menggeser sinyal masukan *∆P*, PSS harus pula harus mengkompensasi fasa terbelakang yang dihasilkan oleh rangakaian eksitasi.



Gambar 2.16 Hubungan fasa *∆P, ∆T* dan Sistem Eksitasi

Dari Gambar 2.16 *∆P* tertinggal sebesar 90° dari sumbu *∆T*D positif sedang sistem eksitasi menghasilkan fasa terbelakang sebesar . Jadi agar *∆P* sefasa dengan *∆T*D, PSS harus menghasilkan fasa sebesar *β*1 dan *β*2. Bila PSS diatur untuk menghasilkan pergeseran fasa sebesar *β*2 maka dikatakan metode penalaan fasanya adalah metode penalaan fasa non minimum.

Jika didefinisikan :

*U*G : Sinyal keluaran PSS ke AVR mesin

*∆P*: perubahan daya mesin

*G*F : Fungsi alih rangkaian medan.

*GA:* Fungsi alihAVR

Maka sinyal keluaran PSS dapat dituliskan:

*U*G *= GF(*S*)∆P* (2.27)

Blok diagram model PSS ditunjukkan pada Gambar 2.17



Gambar 2.17 PSS dalam bentuk model *linear*

Berikut penjelasan masing-masing blok pada Gambar 2.17

1. Blok *Gain*

*Gain* berfungsi untuk mengatur besaran penguatan agar diperoleh besaran torsi sesuai dengan yang diinginkan.

1. Blok *Washout*

*Washout filter* berfungsi untuk menyediakan bias *steady state* *output* PSS yang akan memodifikasi tegangan terminal generator. PSS diharapkan hanya dapat merespon variasi transien dari sinyal kecepatan rotor generator *∆ω* dan tidak untuk sinyal DC *offset. Washout* *filter* bekerja sebagai *high pass filter* yang melewatkan semua frekuensi yang diinginkan. Jika hanya mode lokal yang diinginkan, nilai *T*w dapat dipilih dalam range 1 sampai 2. Tetapi, jika mode interarea juga ingin diredam, maka nilai *T*w harus dipilih dalam interval 10 sampai 20.

1. Blok *Lead/Lag*.

*Lead-Lag* berfungsi sebagai penghasil karakteristik *phase-lead* yang sesuai untuk mengkompensasi *phasa-lag* antara masukan eksitasi dan torsi generator.

1. *Limiter*

*Limiter* berfungsi untuk menghindari *stabilizer* dari perubahan tegangan yang tajam dan tetap mempertahankan pengatur tegangan pada kondisi yang baik selama terjadi gangguan.

**2.2.8 Jaringan Syaraf Tiruan**

Jaringan syaraf adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.

Otak manusia berisi berjuta-juta sel syaraf yang bertugas untuk memproses informasi. Tiap-tiap sel bekerja seperti suatu prosesor sederhana. Masing-masing sel tersebut saling berinteraksi sehingga mendukung kemampuan kerja otak manusia.

Sel-1

(Neuron-1)

Sel-2

(Neuron-2)

Inti Sel

(Neucleus)

Axon

Dendrit

Synapsis

Dendrit

Gambar 2.18 Susunan syaraf manusia.

Gambar 2.18 menunjukkan susunan syaraf pada manusia. Setiap sel syaraf (neuron) akan memiliki satu inti sel, inti sel ini nanti yang akan bertukan untuk melakukan pemrosesan informasi. Informasi yang datang akan diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini akan menjadi masukan bagi neuron lain yang mana antar dendrit kedua sel tersebut dipertemukan dengan synapsis. Informasi yang dikirimkan antar neuron ini berupa rangsangan yang dilewatkan melalui dendrit. Informasi yang datang dan diterima oleh dendrit akan dijumlahkan dan dikirim melalui axon ke dendrit akhir yang bersentuhan dengan dendrit dari neuron yang lain. Informasi ini akan diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering dikenal dengan nama nilai ambang (*threshold*). Pada kasus ini, neuron tersebut dikatakan teraktivasi. Hubungan antar neuron terjadi secara adaptif, artinya struktur hubungan tersebut terjadi secara dinamis. Otak manusia selalu memiliki kemampuan untuk belajar dengan melakukan adaptasi.

**2.2.8.1 Model Jaringan Syaraf Tiruan**

Jaringan syaraf tiruan terdiri dari beberapa neuron dan terdapat hubungan antara neuron-neuron tersebut. Neuron-neuron akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Pada Gambar 2.19 menunjukkan struktur neuron dari jaringan syaraf tiruan.



Gambar 2.19 Struktur neuron jaringan syaraf tiruan

Neuron buatan ini sebenarnya mirip dengan neuron biologis. Neuron-neuron buatan bekerja dengan cara yang sama pula dengan neuron biologis. Informasi (disebut dengan: *input*) dikirim ke neuron dengan bobot kedatangan tertentu. *Input* diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian dibandingkan dengan nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input melewati suatu nilai ambang tertentu, maka neuron akan diaktifkan, tetapi jika tidak maka neuron tidak diaktifkan. Apabila neuron diaktifkan, maka neuron akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya, demikian seterusnya.

Pada jaringan syaraf, neuron-neuron dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (neuron *layer*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan masukan dan lapisan keluaran). Informasi yang diberikan pada jaringan syaraf dirambatkan ke lapisan-lapisan, mulai dari lapisan masukan sampai ke lapisan keluaran melalui lapisan yang lainnya, yang sering disebut dengan lapisan tersembunyi (Siang, 2004).

**2.2.8.2 Arsitektur Jaringan**

Didalam jaringan syaraf tiruan, neuron-neuron dikelompkkan dalam lapisan-lapisan. Umumnya, neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama. Faktor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Pada setiap lapisan yang sama, neuron-neuron akan memiliki fungsi aktivasi yang sama.

Ada beberapa arsitektur jaringan syaraf, antara lain:

1. Jaringan dengan lapisan tunggal (*single layer net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.

1. Jaringan dengan banyak lapisan (*multilayer net*)

Jaringan dengan banyak lapisan memiliki 1 (satu) atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output* (memiliki 1 atau lebih lapisan tersembunyi). Umumnya, ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan. Jaringan dengan lapisan banyak ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal dengan pembelajaran yang lebih rumit.

1. Jaringan dengan lapisan kompetitif (*competitive layer net*)

Hubungan antara neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur.

**2.2.8.3 Proses Pembelajaran**

Jaringan syaraf tiruan mencoba untuk mensimulasikan kemampuan otak manusia untuk belajar. Tidak seperti model biologis, jaringan syaraf memiliki struktur yang tidak dapat diubah, dibangun oleh sejumlah neuron, dan memilki nilai tertentu yang menunjukkan seberapa besar koneksi antar neuron (yang dikenal dengan bobot). Perubahan yang terjadi selama proses pembelajaran adalah perubahan nilai bobot. Nilai bobot akan bertambah, jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, sebaliknya jika informasi tidak disampaikan oleh suatu neuron ke neuron yang lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi. Pada saat pembelajaran dilakukan pada masukan yang berbeda, maka nilai bobot akan diubah secara dinamis sehingga mencapai suatu nilai yang cukup seimbang. Apabila nilai ini telah tercapai mengindikasikan bahwa tiap-tiap masukan telah berhubungan dengan *output* yang diharapkan.

Untuk mendapatkan tingkat kecerdasan yang diinginkan maka jaringan syaraf tiruan harus melalui proses pembelajaran. Pembelajaran (*learning*) adalah proses yang melibatkan serangkaian nilai masukan menjadi masukan jaringan secara berurutan dan bobot jaringan disesuaikan sehingga akan diperoleh nilai-nilai yang sama dengan nilai keluarannya. Ide dasar jaringan syaraf tiruan adalah metode belajar. Jaringan syaraf tiruan membagi metode belajar menjadi dua macam, yaitu:

1. Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika keluaran yang diharapkan telah diketahui sebelumnya. Pada proses pembelajaran, satu pola masukan akan diberikan ke satu neuron pada lapisan masukan. Pola ini akan dirambatkan disepanjang jaringan syaraf hingga sampai ke neuron lapisan keluaran. Lapisan keluaran ini akan membagi pola keluaran yang kemudian dicocokkan dengan keluaran targetnya. Apabila terjadi perbedaan antara pola keluaran hasil pembelajaran dengan pola target, maka disini akan muncul *error*. Apabila nilai *error* ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

1. Pembelajaran tak terawasi (*unsupervised learning*)

Pada metode pembelajaran yang tak terawasi ini tidak memerlukan target keluaran. Pada metode ini tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Tujuan dari pembelajaran ini adalah mengelompokkan unit-unit yang yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola.

Untuk dapat menyelasaikan suatu permasalahan, JST memerlukan prosedur belajar, yaitu bagaimana sebuah konfigurasi JST dapat dilatih untuk mempelajari data historis yang ada. Dengan pelatihan ini, pengetahuan yang terdapat pada data dapat diserap dan direpresentasikan oleh harga-harga bobot koneksi. Jaringan syaraf tiruan terdiri dari sejumlah simpul (*node*) yang merupakan elemen pemroses. Setiap simpul tersebut memodelkan sebuah sel syaraf biologis (neuron). Hubungan antar simpul dicapai melalui bobot koneksi. Bobot koneksi menentukan apakah sinyal yang mengalir bersifat peredam (*inhibitory connection*) atau perangsangan (*excitatory connection*). Bobot koneksi yang bersifat peredam dapat dinyatakan, misalnya oleh bilangan negatif, sedangkan yang bersifat merangsang oleh bilangan positif. Selain ditentukan oleh karakteristik bobot koneksinya, besarnya sinyal yang keluar dari sebuah simpul juga ditentukan oleh fungsi aktivasi (*activation function*) yang digunakannya. Artinya, pemilihan fungsi akivasi menentukan derajat aktif dari sebuah simpul. Pada Gambar 2.20 dibawah ini, memperlihatkan contoh sebuah simpul yang ke-*j* sebagai elemen pemroses. Biasanya masukan bersih simpul ke-*j* adalah : net\_*input* = *XiWIj • Xi* adalah besarnya masukan dari simpul yang ke-*i* (*i* = 1, 2, 3,.. *n*) sedangkan *WIj* adalah besarnya bobot koneksi dari simpul yang ke-*i* ke simpul *j.* Bila masukan bersih cukup kuat untuk mengaktifkan simpul *j,* maka besarnya keluaran dari simpul tersebut adalah *output* = *f (net\_input).*

# 

Gambar 2.20 Simpul (neuron) sebagai elemen pemroses

Pemilihan jenis fungsi aktivasi menentukan bentuk sinyal keluaran dari simpul (*neuron*) dan harus disesuaikan dengan kebutuhan. Pada Gambar 2.21 dibawah ini memperlihatkan dua jenis fungsi aktivasi yang sering digunakan. Pada Gambar 2.21 (a) bentuk sinyal keluarannya adalah biner (0 atau 1), sedangkan pada Gambar 2.21 (b) sinyal kelurannya adalah kontinyu.



Gambar 2.21 Fungsi aktivasi: (a) fungsi pembatas (*tresholded function*) (b) fungsi sigmoid (*sigmoid function*)

* + - 1. ***Recurrent Neural Network* (RNN)**

Jaringan yang mengakomodasi keluran jaringan untuk menjadi masukan pada jaringan itu lagi dalam menghasilkan keluaran jaringan berikutnya disebut sebagai *recurrent neural network*.

Jaringan Elman termasuk salah satu dari bentuk *recurrent neural network*, yang mana jaringan Elman ini terdiri atas *n* buah lapisan tersembunyi. Lapisan pertama pada jaringan memiliki bobot-bobot yang diperoleh dari lapisan masukan. Pada setiap lapisan jaringan akan menerima bobot dari lapisan sebelumnya. Pada setiap lapisan jaringan, kecuali lapisan terakhir juga memiliki satu bobot *recurrent* dan untuk semua lapisan memiliki bias.



Gambar 2.22 *Recurrent neural network*

Jaringan Elman biasanya menggunakan fungsi aktivasi *tansig* untuk lapisan tersembunyi, dan *purelin* untuk lapisan keluaran. Dalam hal ini, neuron pada jaringan ditetapkan cukup banyak, demikian pula jumlah lapisan tersembunyi juga disesuaikan dengan kompleks permasalahan.

*Delay* yang terjadi pada hubungan antara lapisan masukan dengan lapisan tersembunyi pertama pada waktu (*t-*1) dapat digunakan untuk saat (*t*). sehingga apabila kita memiliki 2 jaringan *recurrent* dengan bobot-bobot awal yang sama dan diberikan pada masukan yang yang sama, bias jadi akan menghasilkan keluaran yang berbeda.

Kelemahan *feed forward* *neural network* adalah terbatasnya pada pelatihan fungsi statis dan keluaran hanya tergantung pada kondisi masukan saat ini, sehingga tidak mampu kalau ada pola perubahan pola data masukan. Kelemahan tersebut mendorong dikembangkannya menjadi *recurrent neural network* (RNN). RNN adalah *neural network* dengan fasilitas umpan balik menuju neuron itu sendiri maupun neuron yang lain, sehingga aliran informasi dari masukan mempunyai arah jamak (*multidirectional*). Kondisi ini dimaksudkan untuk menampung kejadian lampau diikutkan pada proses komputasi.

Sturuktur *power system* dan model RNNPSS yang dipasang paralel antara RNN-*identifier* dan RNN-*controller* adalah seperti Gambar 2.23 dan Gambar 2.24.

Gambar 2.23 Sturuktur *power* sistem



Sumber: (Chen, 2009)

Gambar 2.24 Model RNNPSS

1. RNN *Identifier* (RNN-*i*)

Skema dari RNN *identifier* menggunakan model *forward*, yang strukturnya ditunjukkan pada Gambar 2.24 RNN ini membaca keluaran dari *plant* berupa kecepatan sudut *ω(t)* dan berusaha untuk meniru gelombang dari keluaran *plant* dengan cara membandingkan keluaran *plant* dengan keluaran RNN*-i* untuk dilakukan proses pembelajaran untuk memperkecil galat. RNN*-i* mempunyai dua masukan, yaitu *∆ω* dan *V*PSS*. ∆ω* adalah keluaran *plant* dan *V*PSS adalah keluaran PSS. Sebagai masukan awal RNN-*i* menggunakan keluaran PSS konvensional untuk proses pelatihan. Secara matematis dapat di tulis dengan Persamaan (2.42).

Masukan RNN-*i* adalah *output* dari *plant* dan tegangan keluaran PSS

Dengan,

*ω* : deviasi kecepatan anguler terhadap kecepatan sinkron

: output dari PSS



Gambar 2.25 Sturuktur RNN-*i* dengan satu *input*

1. RNN *Controller* (RNN-*c*)

Skema dari RNN *Controller* menggunakan model forward, yang strukturnya ditunjukkan pada Gambar 2.25 RNN ini membaca keluaran dari *plant* yaitu *ω(t)* dan berusaha untuk meniru gelombang dari keluaran *plant* dengan cara membandingkan keluaran *plant* dengan *output*nya untuk dilakukan proses pembelajaran untuk memperkecil galat. RNN-*c* mempunyai masukan *∆ω* yang telah dipasang dengan PSS untuk proses pelatihan*.* Secara matematis dapat di tulis dengan Persamaan (2.45).



Gambar 2.26 Sturuktur RNN-*c* dengan satu *input*

**2.2.8.5 Pelatihan Dengan Teknik Heuristik**

Teknik ini merupakan dari suatu analisis kinerja pada algoritma *steepest* (*gradient*) *descent* standard yang telah disediakan oleh *Matlab.*

**2.2.8.5.1 *Gradient Descent* Dengan Momentum dan *Adaptive* *Learning Rate* (*traingdx*)**

Fungsi ini akan memperbaiki bobot-bobot berdasarkan *gradient descent* dengan laju pembelajaran (*learning rate*) yang bersifat adaptif. Algoritma *gradient descent* dengan laju pembelajaran adaptif, dasarnya sama dengan algoritma *gradient descent* standar dengan beberapa perubahan. Pertama-tama dihitung terlebih dahulu nilai keluaran jaringan dan *error* pelatihan. Pada setiap *epoch*, bobot-bobot baru dihitung dengan menggunakan laju pembelajaran (*learning rate*) yang ada. Kemudian dihitung kembali keluaran jaringan dan dan *error* pelatihan. Jika *error* pelatihan yang baru ini melebihi *error* pelatihan lama dengan kenaikan melebihi maksimum kenaikan kerja (max\_perf\_inc), maka bobot-bobot baru tersebut akan diabaikan, sekaligus nilai laju pembelajaran akan dikurangi dengan cara mengalikannya dengan lr\_dec.Sebaliknya, apabila *error* pelatihan baru lebih kecil daripada *error* pelatihan lama, maka nilai-nilai bobot akan dipertahankan, sekaligus nilai laju pembelajaran akan dinaikkan dengan cara mengalikannnya dengan lr\_inc.

Dengan cara ini, apabila laju pembelajaran terlalu tinggi dan mengarah ke ketidakstabilan, maka laju pembelajaran akan diturunkan. Sebaliknya jika laju pembelajaran terlalu kecil untuk menuju konvergenan, maka laju pembelajaran akan dinaikkan. Dengan demikian, maka algoritma pembelajaran akan tetap terjaga pada kondisi stabil.

Parameter-parameter yang perlu diset untuk pembelajaran ini adalah fungsi pelatihan, menjadi *traingdx*. Parameter-parameter lain yang perlu diset adalah: (kusumadewi, 2004)

* Maksimum *epoch* (iterasi)

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch*. Jumlah *epoch* yang digunakan dalam penelitian ini adalah 2000.

* Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan. Nilai untuk kinerja tujuan yang digunakan adalah 1×10-9.

* Laju pembelajaran (*learning rate*)

*Learning rate* adalah laju pembelajaran. Nilai *default* untuk laju pembelajaran adalah 0,01

* Rasio untuk menaikkan laju pembelajaran

Rasio ini berguna sebagai faktor pengali untuk menaikkan laju pembelajran apabila laju pembelajaran yang ada terlalu rendah untuk mencapai konvergen. Nilai *default* untuk rasio kenaikkan laju pembelajaran adalah 1,05.

* Rasio untuk menurunkan laju pembelajaran

Rasio ini berguna sebagai faktor pengali untuk menurunkan laju pembelajaran apabila laju pembelajaran yang ada terlalu tinggi dan menuju ke ketidakstabilan. Nilai *default* untuk rasio penurunan laju pembelajaran adalah 0,7.

# BAB III

# METODOLOGI PENELITIAN

**3.1 Metodelogi Penelitian**

Dalam tugas akhir ini dilakukan simulasi dan membandingkan antara sistem tenaga listrik SMIB yang menggunakan *recurrent neural network power system stabilizer* (RNNPSS), PSS konvensional, dan sistem tanpa PSS*.* Gambar *single line* diagram SMIB dapat dilihat pada Gambar 3.1.

# 

Gambar 3.1 *Single machine infinite bus*

**3.2 Alat dan Data Penelitian**

**3.2.1 Alat Penelitian**

Penelitian ini menggunakan seperangkat komputer yang memuat program aplikasi *Matlab* versi 7.8.0.347 (*R*2009*a*) yang digunakan untuk melakukan studi stabilitas dinamik sistem tenaga listrik akibat gangguan kecil dan pelatihan RNN.

Untuk menjalankan *software* *Matlab* 7.8.0.347 (*R*2009*a*), diperlukan komputer dengan spesifikasi minimum sebagai berikut:

1. Minimal processor Intel Pentium III.
2. Memory minimum 128 MB RAM, disarankan menggunakan 512 MB RAM untuk kinerja yang lebih cepat.
3. Minimal operasi *Microsoft Windows* 98. Program ini tidak berjalan pada *Windows* 95

**3.2.2 Data Penelitian**

**A. Sumber Data**

Penelitian ini menggunakan pemodelan mesin tunggal yang terdapat pada buku “ *Power System Control*” karya Hadi Saadat dan data generator serta saluran pada sistem kelistrikan Lombok dalam bentuk SMIB dengan mengambil salah satu pembangkit dan 2 saluran yang terhubung ke *infinite* bus .

**B. Jenis Data**

1. Data paremeter mesin pada buku “ *Power System Control*” karya Hadi Saadat.

Tabel 3.1 Parameter mesin pada buku “ *Power System Control*” karya Hadi Saadat.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nomor | Bagian sistem | Parameter | Nilai | Satuan |
| 1 | Turbin | *K*T | 1 | pu |
| *T*T | 0,5 | s |
| 2 | *Governor* | *K*g | 1 | pu |
| *T*g | 0,2 | s |
| 3 | Generator | *K*G | 0,8 | pu |
| *T*G | 1,4 | s |
| 4 | *Amplifier* | *K*A | 10 | pu |
| *T*A | 0,1 | s |
| 5 | *Exciter* | *K*G | 1 | pu |
| *T*G | 0,4 | s |
| 6 | Sensor | *K*R | 1 | pu |
| *T*R | 0,05 | s |
| 7 | *Inertia* | *H* | 5 | pu |
| 8 | *Regulation* | *R* | 0,05 | pu |

Sumber: buku “*Power system Control”*  karya Haadi Saadat, pp. 568

2 . Sistem kelistrikan Lombok



Gambar 3.2 Penyederhanaan *single line* diagram sistem kelistrikan Lombok

Tabel 3.2 Parameter sistem kelistrikan Lombok

|  |  |  |  |
| --- | --- | --- | --- |
| Nomor | Parameter | Nilai | Satuan |
| 1 | Rating daya (MVA) | 4,3 | MVA |
| 2 | Rating Tegangan (kV) | 6,3 | kV |
| 3 | *Line* 1 | 0,179 + *j*0,423 | pu |
| 4 | *Line* 2 | 0,182 + *j*0,135 | pu |
| 5 | *Line 3* | 0,7144 − *j*0,2331 | pu |
| 6 | *Line 4* | 0,2394 + *j*0,1284 | pu |
| 5 | *X*T | 0,1 | pu |
| 6 | *X’*d | 0,0600 | pu |
| 7 | *X*d | 0,7746 | pu |
| 8 | *X*q | 1,64 | pu |
| 9 | *M* | 2,4 | pu |
| 10 | *T’*d0 | 5,6 | s |
| 11 | *K*gu | 20 | pu |
| 12 | *T*gu | 1 | s |
| 13 | *D*T | 0,1 | s |
| 14 | *R* | 1 | pu |
| 15 | *D* | 5 | pu |

Sumber: PT. PLN (Persero) sektor Lombok yang diolah oleh Praja (2008)

Tabel 3.3 Parameter eksitasi sistem kelistrikan Lombok

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generator | *K*A | *T*A | *K*f | *T*f |
| PLTD Taman | 52 | 0,01 | 0,114 | 0,6 |

Keterangan: Kapasitas generator @ < 10 MVA

Sumber: Parameter dan sampel data IEEE Sistem Eksitasi tipe ST1A pada *software* ETAP

Model sistem kelistrikan Lombok masih menggunakan model lama dengan mengggunakan sistem untuk saluran 20 kV. Pada Tabel 3.2 parameter generator dan saluran sistem kelistrikan Lombok, menggunakan parameter generator pembangkit Taman. Pada Gambar 3.2 nilai saluran di ambil pada saluran Taman yang terhubung ke busAmpenan (*infinite*) yang terdiri dari 4 saluran dengan pemodelan sistem sudah disederhanakan dalam bentuk SMIB seperti yang terdapat pada Gambar 3.3. Sedangkan untuk sistem eksitasi mengacu pada pemodelan sistem eksitasi standar IEEE yaitu tipe ST1 untuk generator dengan kapasitas pembangkitan dibawah 10 MVA.



Gambar 3.3 SMIB pembangkit Taman yang terhubung ke bus *infinite*

**3.3 Langkah- Langkah Penelitian**

Tahapan penelitian dibagi dalam 2 bagian yaitu: pelatihan *recurrent neural network* (RNN) secara *off line* yang pada penelitian ini disebut tahapan pre kontrol dan simulasi stabilitas SMIBsetelah pemasangan RNNPSS.

**3.3.1 Pelatihan *Neuro Identifier* Dan *Neuro Controller* Saat Pre Konrol**

Untuk melatih *recurrent neural network* (RNN) dilaksanakan menurut langkah-langkah berikut

1. Menyimpan data hasil *running* SMIB yang telah di pasangi PSS sebagai data *input* dan target dalam pelatihan.

Pada pelatihan ini data dari *neuro identifier* adalah: keluaran *plant* berupa dan  yang didapat dari keluaran PSS konvensional. Keluaran dari *neuro identifier* ini adalah hasil identifikasi terhadap perubahan kecepatan pada *plant*, yang disebut . Target pelatihan *neuro identifier* ini adalah perubahan kecepatan sudut pada *plant* . Sedangkan *input* dari *neuro controller* pada fase ini adalah: dan *Δω(k ­* 3*)*. Keluaran dari *neuro controller* ini adalah sinyal kontrol yang menggantikan VPSS.

1. Menentukan metode pelatihan dalam meniru keluaran *plant* yang pada penelitian ini menggunakan *tansig* pada *hidden layer* dan *purelin* pada keluaran.
2. Menentukan jumlah neuron pada yang akan digunakan dalam pelatihan.

Jumlah neuron pada pelatihan ini adalah 5 neuron dengan 6 *input layer*, 2 *hidden layer* dan 1 *output layer* pada pelatihan *neuro identifier* sedangkan *neuro controller* terdiri dari 3 *input layer*, 2 *hidden layer*, dan 1 *output layer*.

1. Menentukan jumlah *epoch*, *goal*, dan *learning rate* yang digunakan dalam pelatihan.

Jumlah *epoch* pada pelatihan ini adalah 2000 *goal* yang ingin didapat adalah 1×10-9 dengan *learning rate* sebesar 0,1.

1. Memplot grafik hasil pelatihan

Grafik hasil pelatihan ini adalah *input* dan target dari hasil pelatihan.

1. Meng-*update* bobot jaringan dan bobot bias apabila *error* yang di dapat masih terlalu besar.

Apabila *error* yang didapat terlalu besar maka diakukan *up date* bobot. *Up date* bobot ini bertujuan untuk menghasilkan bobot bias dan jaringan yang baru untuk digunakan pada pelatihan selanjutnya agar *error* bias diperbaiki.

1. Memplot grafik hasil *update* bobot jaringan dan bias pada *recurrent neural network* (RNN).

**3.3.2 Simulasi Stabilitas SMIB Setelah Pemasangan RNNPSS**

1. Menentukan parameter-parameter sistem yang digunakan dalam pemodelan.

2. Menghitung nilai *K*1 sampa *K*6 yang akan digunakan pada *plant*.

Untuk menghitung nilai parameter *K*1 sampai *K*6 diperlukan data-data berupa hasil perhitungan tegangan internal bus generator, reaktansi generator dan perhitungan saluran.

3. Menentukan nilai PSS yang akan digunakan pada *plant* .

4. Memodelkan sistem SMIB yang pada penelitian pemodelan terdiri dari *plant* tanpa PSS, *plant* dengan PSS dan *plant* dengan RNNPSS.

5. Menarik kesimpulan.

**3.4 Diagram Alir Pelatihan *Neuro Identifier* Dan *Neuro* *Controller***



Gambar 3.4 Diagram alir pelatihan *neuro identifier* dan *controller*

**3.5 Diagram Alir Penerapan RNNPSS Pada SMIB**



Gambar 3.5 Diagram alir penerapan RNNPSS pada SMIB

**BAB IV**

**HASIL DAN PEMBAHASAN**

Bab 4 ini berisi tentang simulasi dan analisis hasil simulasi dari pemodelan sistem. Simulasi ini dilakukan untuk menguji kinerja dari PSS berbasis *recurrent neural network*. Simulasi dilakukan pada sistem SMIB yang terdapat pada buku “*Power System Control*” karya Hadi Saadat, pp. 568 dan sistem kelistrikan Lombok dalam bentuk SMIB. Hasil simulasi ditunjukkan dengan membandingkan kinerja antara sistem tanpa kontrol, sistem dikontrol menggunakan PSS, dan sistem dikontrol menggunakan PSS berbasis *recurrent neural network* (Akbar, dkk).

Data pelatihan untuk meredam osilasi kecepatan sudut adalah keluaran *plant* berupa kecepatan dengan variasi gangguan antara 0,02 sampai 0,7 pu. Fungsi aktivasi yang digunakan adalah *tansig* dan *purelin*. Jumlah neuron yang digunakan adalah 5. Sturuktur jaringan yang digunakan pada pelatihan ini adalah terdiri dari empat lapis, yaitu lapis masukan, dua lapis *hidden* layer dan lapisan keluaran. Setelah dilakukan proses pemetaan, maka langkah selanjutnya adalah memasang RNNPSS di sistem.

**4.1 Desain Kontroler dengan *Neural Network***

Desain *Power System Stabilizer* berbasis *Recurrent Neural Network* (PSSRNN) terdiri dari dua buah *neural network* yaitu *neuro controller* dan *neuro identifier*. Struktur dari *neuro identifier* dan *neuro controller* dapat dilihat pada Gambar 4.1

Gambar 4.1 Sturktur *neuro identifier* dan *neuro controller* RNNPSS

Gambar 4.1 di atas menunjukkan jalur *recurrent neural network*  untuk melakukan *update* bobot dari *neuro controller* dan *neuro identifier.*

Karena dua proses pelatihan ditunjukkan dalam satu gambar, maka perlu untuk dijabarkan mengenai definisi dari *time* step. Sinyal dan , keduanya *disampling* dengan periode waktu *time* step *k*, tetapi  bukanlah respon dari sinyal kontrol . Efek dari sinyal kontrol  baru dapat dilihat pada saat RNNPSS diterapkan secara post-kontrol.

**4.1.1 *Neuro Identifier***

*Neuro identifier* dimodelkan dengan keluaran berupa  pada *time* k+1 bergantung pada *n­* nilai terakhir dari keluaran dan masukan. Perumusan dari keluaran *neuro identifier* diberikan dalam Persamaan (4.1).

 (4.1)

Dengan y(k) dan u(k) menunjukkan kaluaran dan masukan dari *plant* *time* step k.

Untuk simulasi yang dilakukan pada tugas akhir ini, dan merupakan perubahan kecepatan () dari *plant*, *output* *neuro controller* adalah dan estimasi keluaran *plant* oleh *neuro identifier* yang disebut sebagai .

Satu alasan mengapa hanya dipilih tiga *time* step adalah karena model sistem yang berorde tiga dari sistem sudah cukup untuk melakukan simulasi model *dinamik stability*. Alasan yang lain adalah semakin banyak *time delay* yang digunakan, maka akan memerlukan waktu komputasi yang lama pula, dan pada penelitian yang telah dilakukan sebelumnya, sudah dapat dibuktikan bahwa tiga *time delay* pun sudah cukup untuk menyelesaikan permasalahan serupa.

*Neuro identifier* merupakan jaringan syaraf tiruan yang dilatih menggunakan algoritma *Elman recurrent neural network.* Jumlah dari *input* layer, *hidden* *layer*, dan *output* *layer* adalah 6, 5, dan 1, dengan *hidden layer* terdiri dari 2 *layer*.

**4.1.2 Neuro *Controller***

*Neuro controller* juga merupakan *network* *multilayer* yang dilatih menggunakan algoritma *Elman recurrent neural network*. Jumlah neuron pada *layer* *input, hidden*, dan *output* berturut-turut adalah 3, 5, dan 1, dengan *hidden* *layer* terdiri dari 2 *layer*. *Input* pada *neuro controller* merupakan perubahan kecepatan dan dua nilai sebelumnya. Keluaran dari *neuro controller* ini adalah sinyal kontrol yang disebut sebagai Vpss.

**4.1.3 Proses Pelatihan**

Proses pelatihan dari RNNPSS disebut sebagai fase pre-kontrol. Pada pelatihan ini dilakukan *up date* bobot dari *neural network* untuk mendapatkan bobot yang terbaik sebelum di pasang pada sistem. Pre-kontrol merupakan proses yang sangat penting karena dapat berfungsi untuk mencegah kemungkinan ketidak-stabilan pada awal proses post-kontrol.

**4.1.4 Fase Pre-Kontrol**

Pada fase ini, dilakukan pelatihan *neural network* baik itu *neuro identifier* maupun *neuro controller.* Pelatihan ini memungkinkan *neural network* untuk mempelajari segala kemungkinan perilaku dinamik dari *plant*. *Error* yang ingin dicapai dalam pelatihan ini adalah 1×10-9 dengan *epoch* sebanyak 2000 dan laju pembelajaran sebesar 0,1.

**4.1.5 Proses Pelatihan *Neuro Identifier* Fase Pre-Kontrol**

Proses pelatihan dari *neuro identifier* selama fase pre-kontrol ditunjukkan pada Gambar 4.2. Masukan dari *neuro identifier* pada fase ini adalah:  . Keluaran dari *neuro identifier* ini adalah hasil identifikasi terhadap perubahan kecepatan pada *plant*, yang disebut . Target pelatihan *neuro identifier* ini adalah perubahan kecepatan pada *plant* .

Tabel 4.1 Data target dan *output* jaringan pada pelatihan *identifier*

|  |  |  |
| --- | --- | --- |
| Target | Output jaringan | MSE |
| 0,00199 | 0,002308 | 0,0032 |
| 0,002956 | 0,00321 | 0,00025 |
| 0,003965 | 0,003973 | 0.000008 |
| 0,005506 | 0,00501 | 0,000496 |
| 0,005904 | 0,006458 | 0,00055 |
| 0,007781 | 0,007608 | 0,000173 |
| 0,009558 | 0,009428 | 0,00031 |
| 0,011193 | 0,011151 | 0,000423 |
| 0,012644 | 0,012614 | 0,0003 |
| 0,013874 | 0,014054 | 0,0002 |

Tabel 4.1 diatas menunjukan 10 sampel data target dengan *output* jaringan *identifier* yang telah dilatih yang diambil pada *net* ke-4. Dari data diatas dilihat bahwa hasil pelatihan *neuro identifier* hasilnya tidak terlalu jauh berdasarkan *error* yang didapat dengan target yang diinginkan. Peniruan target (dalam hal ini adalah kecepatan sudut *plant*) dengan baik akan membuat *neuro identifier* semakin baik dalam meniru perilaku *plant* pada saat post kontrol untuk itu pelatihan awal ini sangat penting dalam membentuk *neuro identifier*.

Hasil pelatihan *neuro identifier* untuk simulasi SMIB dapat dilihat pada Gambar 4.3 sampai Gambar 4.6.



Gambar 4.2 Sturuktur *neuro identifier* saat pre-kontrol



Gambar 4.3 Hasil pelatihan ke-1 *neuro identifier*

Gambar 4.3 merupakan hasil pelatihan ke-1 *neuro identifier,* pada pelatihan ini *input* merupakan matrik 1×6 dengan target matrik 1×1 jumlah data pada setiap matrik adalah 214. *Error* yang diingkan pada pelatihan ini adalah 0 dengan jumlah iterasi 2000. Dari hasil pelatihan *neuro identifier* belum mampu meniru perilaku *plant* dengan baik, *error* yang dihasilkan pada pelatihan ini adalah 2,48 X 10-8 pada iterasi ke 2000 sehingga perlu pelatihan lanjutan dengan meng-*update* bobot bias dan jaringan.



Gambar 4.4 Hasil pelatihan ke-2 *neuro identifier*

Pada pelatihan ke-2 dengan data yang sama *neuro identifier* hanya mampu memberikan *error* sebesar 1,29 X 10-8  pada iterasi ke 2000. Pada kondisi ini *neuro identifier* belum mampu meniru perilaku *plant* secara sempurna sehingga perlu dilatih lagi dengan meng-update bobot dan jaringannya.



Gambar 4.5 Hasil pelatihan ke-3 *neuro identifier*

Pada pelatihan ke-3 *neuro identifier* sudah mulai mampu meniru perilaku *plant* dengan baik, hal ini dapat dilihat dengan *error* yang diberikan yaitu sebesar 9,92 X 10-9  pada itersi ke 2000. Namun pada Gambar 4.5 dapat kita lihat bahwa masih terdapat riak-riak pada hasil identifikasi *neuro identifier* sehingga perlu di latih lagi dengan meng-*update* bobot bias dan jaringan.

Gambar 4.6 Hasil pelatihan ke-4 *neuro identifier*

Dari hasil identifikasi yang ditunjukkan pada Gambar 4.6 dapat dilihat bahwa *neuro identifier* dapat melakukan identifikasi pada perilaku *plant* dengan sangat baik. Pelatihan dilakukan sebanyak 4 kali dengan meng-*update* bobot yang terdapat pada jaringan untuk menghasilkan pelatihan secara sempurna. *Error* terkecil yang terjadi pada proses pelatihan ini mencapai 9,61x10-9.

Bobot bias dan *error* pada pelatihan *neuro identifier* saat pre-kontrol dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data Pelatihan *Neuro Identifier*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pelatihan | Bobot Bias Input | Bobot Bias Lapisan | Rata-rata *Error* | Iterasi |
| ke-1 | 1,427020596 | -1,44273324 | 2,48 X 10-8 | 2000 |
| -0,644723726 | -0,88110675 |
| -0,235599249 | 0,015398558 |
| 0,830576962 | -0,9100378 |
| 1,575850215 | -1,48813815 |
| ke-2 | 1,428332562 | -1,46380724 | 1,29 X 10-8 | 2000 |
| -0,666401928 | -0,88191609 |
| -0,248133529 | 0,030123069 |
| 0,846745169 | -0,90410819 |
| 1,570699189 | -1,47834111 |
| ke-3 | 1,442534938 | -1,47039804 | 9,92 X 10-9 | 2000 |
| -0,679916528 | -0,87605979 |
| -0,250253534 | 0,0341588 |
| 0,850167756 | -0,90313835 |
| 1,568945446 | -1,46200141 |
| ke- 4 | 1,453376129 | -1,46783579 | 9,61 X 10-9 | 2000 |
| -0,684503155 | -0,8681567 |
| -0,25309273 | 0,03004471 |
| 0,848488855 | -0,90339034 |
| 1,567660498 | -1,44487819 |

Dari Tabel 4.2 dapat dilihat bahwa dengan dilakukan pelatihan beberapa kali maka bobot lapisan dan bias akan meng-update nilainya untuk mendapatkan performa yang baik. Sehingga *error* yang dihasilkan akan cenderung semakin kecil. Dengan semakin kecilnya *error* yang di dapat maka *neural network* akan semakin baik meniru perilaku masukan yang dalam hal ini adalah keluaran *plant*.

**4.1.6 Proses Pelatihan *Neuro Controller* Fase Pre-Kontrol**

Proses pelatihan dari *neuro controller* selama fase pre-kontrol ditunjukkan pada Gambar 4.7. *Input* dari *neuro controller* pada fase ini adalah: dan *Δω(k ­* 3*)*. Keluaran dari *neuro controoler* ini adalah sinyal kontrol yang menggantikan VPSS. Target pelatihan *neuro controller* ini adalah perubahan VPSS. Proses pelatihan dari *neuro controller* ditunjukkan pada Gambar 4.7 berikut.

Gambar 4.7 Sturuktur *neuro controller* saat fase pre-kontrol

Pada tahapan ini, input dari *neuro controller* adalah dan . Keluaran dari *neuro controller* ini disebut . Sinyal keluaran dari *neuro controller* ini selanjutnya akan dimasukan dalam *neuro identifier*. Sinyal kontrol yang diinginkan dikirim melalui *neuro identifier* dengan membandingkannya dengan respon sistem yang diinginkan. Pada fase pelatihan *neuro controller* ini, nilai bobot dari *neuro identifier* tidak mengalami perubahan. Hasil pelatihan *neuro controller* pada simulasiSMIB ditunjukan pada Gambar 4.8 sampai Gambar 4.11.

Tabel 4.3 Data target dan *output* jaringan *neuro controller*

|  |  |  |
| --- | --- | --- |
| Target | Output jaringan | MSE |
| 0,00226 | 0,002433 | 0,00017 |
| 0,002987 | 0,003155 | 0,00017 |
| 0,004265 | 0,004342 | 0.00077 |
| 0,005935 | 0,005576 | 0,00036 |
| 0,008015 | 0,007837 | 0,00018 |
| 0,008809 | 0,009488 | 0,00068 |
| 0,01155 | 0,010946 | 0,00060 |
| 0,014096 | 0,014126 | 0,00003 |
| 0,01638 | 0,016478 | 0,00098 |
| 0,018344 | 0,018618 | 0,00027 |

Tabel 4.3 diatas menunjukan 10 sampel data target dengan *output* jaringan *neuro controller* yang telah dilatih yang diambil pada *net* ke-4. Dari data diatas dilihat bahwa hasil pelatihan *neuro controller* hasilnya tidak terlalu jauh dengan target yang diinginkan. Keluaran pada *neuro controller* ini adalah sinyal kontrol yang akan menggantikan *VPSS* untuk di kirim ke *neuro identifier* dan *plant*. *Neuro controller* ini dilatih sebanyak 4 kali dengan meng-update bobot bias dan jaringan.



Gambar 4.8 Hasil pelatihan ke-1 *neuro controller*

Pada pelatihan ke-1 *neuro controller* menghasilkan *error* sebesar 5,44 X 10-8 pada iterasi ke 2000. Untuk menghasilkan sinyal kontrol yang lebih baik maka perlu dilakukan pelatihan kembali agar *error* yang dihasilkan bisa diperbaiki. *Net* yang dihasilkan pada pelatihan ke-1 digunakan sebagai jaringan pelatihan selanjutnya.



Gambar 4.9 Hasil pelatihan ke-2 *neuro controller*

Pada pelatihan ke-2 *neuro controller* sudah mulai mampu meniru *input* yang diberikan. Pada pelatihan ini *error* yang dihasilkan sebesar 3,44 X 10-8 pada iterasi ke 2000. Namun pada Gambar 4.9 diatas dapat kita lihat bahwa masih terdapat riak-riak pada sinyal target sehingga perlu dilakukan pelatihan ulang agar menghasilkan sinyal kontor yang lebih baik.



Gambar 4.10 Hasil pelatihan ke-3 *neuro controller*

Dengan meng-*update net* pada pelatihan ke-2 sinyal kontrol pada pelatihan ke-3 dapat menurunkan *error* sebesar 1,36X 10-8 dari *error* sebelumnya yaitu sebesar 3,44 X 10-8. *Error* yang dihasikan belum mampu untuk membuat *neuro controller* menghasilkan sinyal kontrol yang baik sehingga perlu dilakukan pelatihan selanjutnya untuk memperbaiki *error*.



Gambar 4.11 Hasil pelatihan ke-4 *neuro controller*

Pada pelatihan ke-4 *neuro controller* mampu menghasilkan sinyal kontrol yang baik. *Error* yang didapat pada pelatihan ini sebesar 1,21 X 10-8 pada iterasi ke 2000. Karena *neuro controller* sudah mampu menghasilkan sinyal kontrol dengan baik maka pealtihan dihentikan sampai pelatihan ke-4.

Bobot bias dan error pada pelatihan *neuro controller* saat pre-kontrol dapat dilihat pada Tabel 4.4.

Tabel 4.4 Data pelatihan *neuro controller* saat pre-kontrol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pelatihan | Bobot bias input | Bobot bias lapisan | Rata rata e*rror* | Iterasi |
| Ke-1 | -1,386079431 | -1,645983901 | 5,44 X 10-8 | 2000 |
| -0,887435911 | -0,707806143 |
| -0,013426412 | 0,18644349 |
| -0.267880447 | -0,860675148 |
| 1,679043927 | 1,552785435 |
| Ke-2 | -1,360939898 | -1,646441315 | 3,44 X 10-8 | 2000 |
| -0,67967092 | -0,652725935 |
| -0,012213659 | 0,183715641 |
| -0,266247884 | -0,856795243 |
| 1,671691221 | 1,555472288 |
| Ke-3 | -1,371879751 | -1,646964035 | 1,36X 10-8 | 2000 |
| -0,609449293 | -0,588550464 |
| -0,019958323 | 0,173138991 |
| -0,23675673 | -0,780964162 |
| 1,648005323 | 1,554811828 |
| Ke-4 | -1,361035708 | -1,647114279 | 1,21 X 10-8 | 2000 |
| -0,590038321 | -0,56001442 |
| -0,027136389 | 0,1722238 |
| -0,202881547 | -0,750168864 |
| 1,624297054 | 1,546569849 |

Dari Tabel 4.4 dapat dilihat bahwa dengan dilakukan pelatihan beberapa kali maka bobot lapisan dan bias *neuro controller* akan meng-*update* nilainya untuk mendapatkan performa yang baik. Sehingga *error* yang dihasilkan akan cenderung semakin kecil. Dengan semakin kecilnya *error* yang di dapat maka *neural network* akan semakin baik memberikan sinyal kontrol dalam meredam gangguan kecil.

**4.1.7 Fase Post-Kontrol**

Pada bagian ini, sinyal kontrol yang dihasilkan oleh *neuro controller* mulai diterapkan pada *plant*. Bagian ini adalah bagian yang sangat penting dimana *neuro identifie*r dan *neuro controller* digunakan untuk memperbaiki deviasi kecepatan sudut sistem akibat adanya gangguan. Model *neuro controller* dan *neuro identifier* diterapkan pada *plant* dapat dilihat pada Gambar 4.12.



Gambar 4.12 Penerapan *neuro identifier* dan *neuro controller* saat fase post-kontrol

**4.2 Menghitung parameter-parameter *K*1 sampai *K*6**

Perhitungan *K*1 sampai *K*6 hanya dilakukan pada SMIB kelistrikan Lombok sedangkan untuk SMIB validasi tidak dilakukan perhitungan nilai *K*1 sampai *K*6 karena sudah diberikan langsung dalam pemodelannya. Untuk menghitung nilai parameter *K*1 sampai *K*6 diperlukan data-data berupa hasil perhitungan tegangan internal bus generator, reaktansi generator, perhitungan saluran dan daya beban.

Besarnya daya beban yang diberikan yaitu sebesar *P* = 0,9 dan *Q* = 0,3. Langkah – langkah perhitungannya sebagai berikut:

1. Menghitung arus yang mengalir ke *infinite* bus
2. Menghitung reaktansi transien antara tegangan terminal dan infinite bus sebelum gangguan
3. Menghitung tegangan internal transient
4. Menghitung sudut kerja awal
5. Menghitung nilai tegangan dan arus di sumbu *q* dan *d*
6. Menghitung nilai *K*1
7. Menghitung nilai *K*2
8. Menghitung nilai *K*3
9. Menghitung nilai *K*4
10. Menghitung nilai *K*5
11. Menghitung nilai *K*6

Dari hasil perhitungan nilai *K*1 sampai *K*6 dapat dilihat pada table berikut:

Tabel 4.5 Parameter *K*1 sampai *K*6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Daya beban | *K*1 | *K*2 | *K*3 | *K*4 | *K*5 | *K*6 |
| *P* = 0,9 dan *Q* =0,3 | 1,8612 | 0,9107 | 0,8911 | 0,5002 | −0,1268 | 1,0571 |

**4.3 Simulasi SMIB pada buku karya Haadi Saadat**

SMIB yang dipakai dalam simulasi ini merujuk pada pemodelan linearSMIB yang terdapat pada buku “*Power System Control*” karya Hadi Saadat pp, 568.

Dari perumusan blok diagram yang dijabarkan pada bab 2, maka dapat dibentuk suatu Model linear SMIB seperti yang ada pada referensi Haadi Saadat yang akan digunakan dalam simulasi.



Gambar 4.13. SMIB pada buku karya Hadi Saadat

**4.3.1 Simulasi dengan step bernilai positif (penambahan beban)**

Performansi sistem dilihat dari grafik keluaran dari perubahan kecepatan dan perubahan sudut rotor serta perubahan tegangan. Grafik perubahan deviasi kecepatan sudut, perubahan sudut rotor, dan perubahan tegangan ditunjukkan berturut-turut pada Gambar 4.14 - 4.17, Gambar 4.18., dan Gambar 4.19

Tabel 4.6 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan step 0,02 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | −0,0271 | 9,6349 |
| PSS Konvensional | −0,0260 | 8,8644 |
| RNNPSS | −0,0254 | 6,0036 |

Dari Tabel 4.6 RNNPSS memberikan performa yang terbaik dibandingkan dengan metode kontrol dengan PSS konvensional dan sistem tanpa kontrol.

Gambar 4.14 Respon perubahan *∆ω* dengan step 0,02 pu

Perubahan step sebesar 0,02 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.14. RNNPSS dapat menurunkan *overshoot* sebesar −0,0254 rad/s (6,27%) dari kondisi semula yaitu sebesar −0,0271 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar −0,0260 rad/s (4,05%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 6,0036 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,8644 s dan 9,6349 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 2,31%, 6,27% dan 4,05% sedangkan *settling time* sebesar 32,27%, 37,69%, dan 7,99% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.6.

Tabel 4.7 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan step 0,03 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | −0,0276 | 9,6313 |
| PSS konvensional | −0,0265 | 8,8653 |
| RNNPSS | −0,0260 | 6,0026 |

Dari Tabel 4.7 RNNPSS memberikan performa yang terbaik dibandingkan dengan metode kontrol dengan PSS konvensional dan sistem tanpa kontrol.

Gambar 4.15 Respon perubahan *∆ω* dengan step 0,03 pu

Perubahan step sebesar 0,03 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.15. RNNPSS dapat menurunkan *overshoot* sebesar −0,0260 rad/s (5,80%) dari kondisi semula yaitu sebesar −0,0276 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar −0,0265 rad/s (3,99%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 6,0026 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,8653 s dan 9,6313 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 1,88%, 5,80% dan 3,99% sedangkan *settling time* sebesar 32,29%, 37,68%, dan 7,95% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.7

Tabel 4.8 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan step 0,25 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | −0,0206 | 9,3219 |
| PSS Konvensional | −0,0199 | 8,6745 |
| RNNPSS | −0,0182 | 5,8039 |

Dari Tabel 4.8 RNNPSS memberikan performa yang terbaik dibandingkan dengan metode kontrol dengan PSS konvensional dan sistem tanpa kontrol.

Gambar 4.16 Respon perubahan *∆ω* dengan step 0,25 pu

Perubahan step sebesar 0,25 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.16. RNNPSS dapat menurunkan *overshoot* sebesar −0,0182 rad/s (11,65%) dari kondisi semula yaitu sebesar −0,0206 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar −0,0199 rad/s (3,40%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 5,8039 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,6745 s dan 9,3219 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 8,54%, 11,65% dan 3,40% sedangkan *settling time* sebesar 33,09%, 37,74%, dan 6,94% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.8.

Tabel 4.9 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan step 0,4 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | −0,0275 | 9,3219 |
| PSS Konvensional | −0,0265 | 8,6745 |
| RNNPSS | −0,0258 | 5,7178 |

Dari Tabel 4.9 RNNPSS memberikan performa yang terbaik dibandingkan dengan metode kontrol dengan PSS konvensional dan sistem tanpa kontrol.

Gambar 4.17 Respon perubahan *∆ω* dengan step 0,4 pu

Perubahan step sebesar 0,4 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.17. RNNPSS dapat menurunkan *overshoot* sebesar −0,0258 rad/s (6,18%) dari kondisi semula yaitu sebesar −0,0275 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar −0,0265 rad/s (3,64%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 5,7178 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,6745 s dan 9,3219 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 2,64%, 6,18% dan 3,64% sedangkan *settling time* sebesar 34,08%, 38,66%, dan 6,94% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.9.

Tabel 4.10. Perbandingan respon perubahan *∆δ* pada sistem SMIBdengan step 0,02 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (pu) | *Settling time* (s) |
| Tanpa Kontrol | −0,0454 | 9,1847 |
| PSS Konvensional | −0,0455 | 8,6026 |
| RNNPSS | −0,0454 | 9,1678 |



Gambar 4.18 Perubahan *∆δ* dengan step 0,02 pu

Perubahan step sebesar 0,02 pu diberikan pada sistem sehingga menyebabkan perubahan sudut rotor pada sistem. Dari Gambar 4.18. Dapat dilihat bahwa *overshoot* sudut rotor ketika dipasang RNNPSS sebesar −0,0454 pu lebih baik dari pada PSS konvensional yang memiliki *overshoot* sebesar −0,0455 pu sedangakan sistem tanpa kontrol memiliki nilai *overshoot* sebesar −0,0454 pu. *Settling time* tercepat diberikan oleh PSS konvensional yaitu sebesar 8,6026 s, sedangkan RNNPSS dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 9,1678 s dan 9,1847 s. Lebih jelasnya dapat dilihat pada Tabel 4.10.

Tabel 4.11 Perbandingan respon perubahan tegangan pada sistem SMIBdengan step 0,02 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (pu) | *Settling time* (s) |
| Tanpa Kontrol | 1,0379 | 3,9253 |
| PSS Konvensional | 0,9981 | 7,0907 |
| RNNPSS | 1,0335 | 4,0958 |



Gambar 4.19 Perubahan tegangan dengan step 0,02 pu

Perubahan step sebesar 0,02 pu diberikan pada sistem sehingga menyebabkan perubahan tegangan pada sistem. Dari Gambar 4.19 dapat dilihat bahwa RNNPSS dapat menurunkan *overshoot* tegangan sebesar 1,0335 pu (0,80%) dari kondisi semula yaitu sebesar 1,0379 pu masih lebih baik PSS konvensional yaitu sebesar 0,9981 pu (0,20%). *Settling time* tercepat diberikan oleh sistem tanpa PSS yaitu sebesar 3,9253 s, sedangkan RNNPSS dan PSS konvensional hanya mampu memberikan *settling time* sebesar 4,0958 s dan 7,0907 s. Lebih jelasnya dapat dilihat pada Tabel 4.11

**4.3.2 Simulasi dengan step bernilai negatif (pengurangan beban)**

Tabel 4.12 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan step −0,02 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,0271 | 9,6349 |
| PSS Konvensional | 0,0260 | 8,8644 |
| RNNPSS | 0,0111 | 7,2633 |

Dari Tabel 4.12 RNNPSS memberikan performa yang terbaik dibandingkan dengan metode kontrol dengan PSS konvensional dan sistem tanpa kontrol.

Gambar 4.20 Respon perubahan *∆ω* dengan step −0,02 pu

Perubahan step sebesar −0,02 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.20. RNNPSS dapat menurunkan *overshoot* sebesar 0,0111 rad/s (59,04%) dari kondisi semula yaitu sebesar 0,0271 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0,0260 rad/s (4,06%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 7,2633 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,8644 s dan 9,6349 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 57,31%, 59,04% dan 4,06% sedangkan *settling time* sebesar 18,06%, 24,61%, dan 7,99% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.12.

Tabel 4.13 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan step −0,05 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,0287 | 9,6244 |
| PSS Konvensional | 0,0275 | 8,8667 |
| RNNPSS | 0,0126 | 5,6664 |



Gambar 4.21 Respon perubahan *∆ω* dengan step −0,05 pu

Perubahan step sebesar −0,05 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.21. RNNPSS dapat menurunkan *overshoot* sebesar 0,0126 rad/s (56,10%) dari kondisi semula yaitu sebesar 0,0287 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0,0275 rad/s (4,18%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 5,6664 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,8667 s dan 9,6244 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 54,18%, 56,10% dan 4,18% sedangkan *settling time* sebesar 36,09%, 41,12%, dan 7,87% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.13.

Tabel 4.14 Perbandingan respon perubahan *∆ω* pada sistem SMIBdengan step −0,5 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,0554 | 9,5167 |
| PSS Konvensional | 0,0530 | 8,8536 |
| RNNPSS | 0,0392 | 5,7580 |



Gambar 4.22 Respon perubahan *∆ω* dengan step −0,5 pu

Perubahan step sebesar −0,5 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.22. RNNPSS dapat menurunkan *overshoot* sebesar 0,0392 rad/s (29,24%) dari kondisi semula yaitu sebesar 0,0554 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0,0530 rad/s (4,33%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 5,7580 s, sedangkan PSS konvensional dan tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,8536 s dan 9,5167 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 26,04%, 29,24% dan 4,33% sedangkan *settling time* sebesar 34,96%, 39,50%, dan 6,97% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.14.

Tabel 4.15 Perbandingan respon perubahan *∆ω* pada sistem SMIBdengan step −0,7 pu

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,0683 | 9,4882 |
| PSS Konvensional | 0,0657 | 8,8368 |
| RNNPSS | 0,0525 | 5,7870 |



Gambar 4.23 Respon perubahan *∆ω* dengan step −0,7 pu

Perubahan step sebesar −0,7 pu diberikan pada sistem sehingga menyebabkan perubahan deviasi kecepatan sudut pada sistem. Dari Gambar 4.24. RNNPSS dapat menurunkan *overshoot* sebesar 0,0525 rad/s (23,13%) dari kondisi semula yaitu sebesar 0,0683 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0,0657 rad/s (3,81%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 5,7870 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 8,8368 s dan 9,4882 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 20,09%, 23,13% dan 3,81% sedangkan *settling time* sebesar 34,51%, 39,01%, dan 6,87% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.15.

**4.4 Simulasi SMIB Sistem Kelistrikan Lombok**

Setelah dilakukan penyederhanaan sistem kelistrikan Lombok menjadi SMIB dan perumusan blok diagram yang dijabarkan pada bab 2, maka dapat dibentuk suatu Model linear SMIB seperti yang akan digunakan dalam simulasi. Model linear SMIB sistem kelistrikan Lombok dapat dilihihat pada Gambar 4.24.

Gambar 4.24. SMIB sistem kelistrikan Lombok

Tabel 4.16 Perbandingan respon perubahan *∆ω* pada sistem SMIBdengan beban *P* = 0,9 pu, *Q* = 0,3 pu dan *Pf* = 0,95

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,2368 | 5,5354 |
| PSS Konvensional | 0,2328 | 5,4726 |
| RNNPSS | 0,2184 | 3,5067 |



Gambar 4.25 Respon perubahan *∆ω* dengan *P* = 0,9 pu, *Q* 0.3 pu, dan *Pf* = 0,95

Pada Gambar 4.25 dengan gangguan perubahan beban sebesar *P* = 0,9 pu, *Q* = 0,3 pu dan *Pf* = 0,95 terjadi pada sistem. RNNPSS dapat menurunkan *overshoot* sebesar 0,2184 rad/s (9,29%) dari kondisi semula yaitu sebesar 0,2368 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0,2328 rad/s (1,69%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 3,5067 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 5,4726 s dan 5,5354 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 6,19%, 9,29% dan 1,69% sedangkan *settling time* sebesar 35,92%, 36,65%, dan 1,13% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.16.

Tabel 4.17 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan beban *P* = 0,75 pu , *Q* = 0,35 pu, dan *Pf* = 0,90

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,2305 | 5,4615 |
| PSS Konvensional | 0,2271 | 5,4101 |
| RNNPSS | 0,2189 | 4,5065 |



Gambar 4.26 Respon perubahan *∆ω* dengan *P* = 0,75 pu, *Q* = 0,35 pu, dan *Pf* = 0,90

Pada Gambar 4.29 dengan gangguan perubahan beban sebesar *P* = 0,75 pu, *Q* = 0,35 pu dan *Pf* = 0,90 terjadi pada sistem. Dari hasil dapat dilihat RNNPSS dapat menurunkan *overshoot* sebesar 0,2189 rad/s (5,03%) dari kondisi semula yaitu sebesar 0,2305 rad/s masih kurang baik dari PSS konvensional yang mampu menurunkan sebesar 0,2271 rad/s (1,48%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 4,5065 s, sedangkan PSS konvensional dan sistem tanpa kontrol memberikan *settling time* sebesar 5,4101 s dan 5,4615 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 3,61%, 5,03% dan 1,48% sedangkan *settling time* sebesar 16,70%, 17,49%, dan 0,94% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.17.

Tabel 4.18 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan beban *P* = 0,85 pu , *Q* = 0,24 pu, dan *Pf* = 0,96

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0.2795 | 5.1444 |
| PSS Konvensional | 0.2717 | 5.0796 |
| RNNPSS | 0.2601 | 3.8453 |



Gambar 4.27 Respon perubahan *∆ω* dengan *P* = 0,85 pu ,*Q* = 0,24 pu, dan *Pf* = 0,96

Gambar 4.27 dengan gangguan perubahan beban yang sama yaitu sebesar *P* = 0,85 pu, *Q* = 0,24 pu dan *Pf* = 0,96 terjadi pada sistem. Dari hasil dapat dilihat RNNPSS dapat menurunkan *overshoot* sebesar 0.2601 rad/s (6,94%) dari kondisi semula yaitu sebesar 0.2795 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0.2717 rad/s (2,79%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 3.8453 s, sedangkan PSS konvensional dan sistem tanpa kontrol memberikan *settling time* sebesar 5.0796 s dan 5.1444 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* berturut-turut sebesar 4,27%, 6,94% dan 2,79% sedangkan *settling time* sebesar 24,30%, 25,25%, dan 1,26% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.18.

Tabel 4.19 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan *P* = 0,8 pu, *Q* = 0,19 pu dan *Pf* = 0,97

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | −0,2724 | 5,0721 |
| PSS Konvensional | −0,2657 | 5,0155 |
| RNNPSS | −0,2598 | 4,8993 |



Gambar 4.28 Respon perubahan *∆ω* dengan *P* = 0,8 pu, *Q* = 0,19 pu, dan *Pf* = 0,97

Perubahan beban sebesar *P* = 0,8 pu, *Q* = 0,19 pu dan *Pf* = 0,97 terjadi pada sistem. Dari Gambar 4.28. RNNPSS dapat menurunkan *overshoot* sebesar −0,2598 rad/s (5,24%) dari kondisi semula yaitu sebesar −0,2724 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar −0,2657 rad/s (2,46 %). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 4,8993 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 5,0155 s dan 5,0721 s. Dari perbandingan hasil simulasi terdapat perubahan nilai signifikan masih kurang dengan *overshoot* berturut-turut sebesar 2,22%, 5,24% dan 2,46% sedangkan *settling time* sebesar 2,32%, 3,41%, dan 1,12% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.19.

Tabel 4.20 Perbandingan respon perubahan *∆ω* pada sistem SMIB dengan *P* = 0,9 pu, *Q* = 0,1 pu dan *Pf* = 0,99

|  |  |  |
| --- | --- | --- |
| Metode Kontrol | *Overshoot* (rad/s) | *Settling time* (s) |
| Tanpa Kontrol | 0,3609 | 5,8628 |
| PSS Konvensional | 0,3428 | 5,7539 |
| RNNPSS | 0,3398 | 4,4047 |



Gambar 4.29 Respon perubahan *∆ω* dengan *P* = 0,9 pu *Q* = 0,1 pu dan

*Pf* = 0,99

Perubahan beban sebesar *P* = 0,9 pu, *Q* = 0,1 pu dan *Pf* = 0,99 terjadi pada sistem. Dari Gambar 4.29. RNNPSS dapat menurunkan *overshoot* sebesar 0,3398 rad/s (5,85%) dari kondisi semula yaitu sebesar 0,3609 rad/s dan lebih baik dari PSS konvensional yang hanya mampu menurunkan sebesar 0,3428 rad/s (5,02%). *Settling time* tercepat diberikan oleh RNNPSS yaitu sebesar 4,4047 s, sedangkan PSS konvensional dan sistem tanpa kontrol hanya mampu memberikan *settling time* sebesar 5,7539 s dan 5,8628 s. Dari perbandingan hasil simulasi terdapat perubahan signifikan nilai *overshoot* sebesar 0,88%, 5,85% dan 5,02% sedangkan *settling time* sebesar 23,45%, 24,87%, dan 1,86% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS. Untuk lebih jelasnya dapat dilihat pada Tabel 4.20.

**BAB V**

**PENUTUP**

**5.1 Kesimpulan**

Dari hasil simulasi dan pembahasan yang dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dilihat dari besarnya *overshoot* dan *settling time* penerapan RNNPSS pada sistem SMIB dapat digunakan untuk meredam gangguan akibat perubahan beban.
2. Dari hasil simulasi data validasi terjadi perubahan signifikan nilai *overshoot* sebesar 22,12%, 25,21% dan 3,97% sedangkan *settling time* sebesar 31,91%, 36,98% dan 7,45% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS.
3. Pada SMIB sistem kelistrikan Lombok dengan perubahan beban yang faktor daya (cos *φ*) di atas 0,85 RNNPSS mampu memberikan respon perubahan deviasi kecepatan sudut lebih baik dengan perubahan signifikan nilai *overshoot* sebesar 2,80%, 4,80% dan 2,06% sedangkan *settling time* sebesar 19,25%, 20,08% dan 1,03% antara RNNPSS terhadap PSS konvensional, RNNPSS terhadap tanpa PSS dan PSS konvensional terhadap tanpa PSS.

**5.2 Saran**

Berdasarkan proses penelitian yang telah dilakukan maka disampaikan saran-saran berikut:

1. Peredaman gangguan dapat dilakukan dengan menggunakan algoritma cerdas yang lain untuk mendapatkan hasil yang lebih baik.
2. Pada penelitian selanjutnya diharapkan penerapan RNNPSS digunakan pada sistem multi mesin dengan 2 pembangkit atau lebih.
3. Karena keterbatasan data sistem eksitasi sistem pembangkit maka digunakan data pendekatan yang lazim digunakan pada pembangkit sejenis. Dalam penelitian selanjutnya data parameter eksitasi tersebut disesuaikan dengan data lapangan.