

# JARINGAN SARAF TIRUAN UNTUK IDENTIFIKASI POLA KODE DERAU PALSU

Eka Saputra L2F096585  
Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro

## Abstrak

Jaringan saraf tiruan merupakan suatu metode yang salah satunya banyak digunakan untuk pengenalan pola dan pemetaan input-output, dimana metode ini adalah suatu instrumentasi cerdas yang dibuat berdasarkan kinerja dari otak manusia yang dapat memberikan solusi pengenalan pola walaupun pola yang diberikan cacat.

Dengan semakin berkembangnya teknologi kini semakin banyak model pelatihan jaringan saraf tiruan yang diciptakan untuk dapat memperbaiki dan menyempurnakan model yang sudah ada, didalam tugas akhir ini digunakan dua buah model pelatihan yaitu model *back propagation* dan model *spiked random* yang dapat dilihat kemampuan kedua model ini dalam pelatihan jaringan dan pengujiannya untuk mengidentifikasi pola kode derau palsu, dimana dilakukan simulasi dengan pengenalan secara *off-line* dari pola-pola masukan yang disajikan dalam bentuk biner sesuai dengan pola kode derau palsu, dengan masukan pola-pola tersebut jaringan saraf tiruan dapat dilatih untuk memberikan tanggapan sesuai pola target keluaran berupa inisial delay waktu.

Pada proses pelatihan untuk pola masukan kode 7 bit dan 15 bit, jaringan saraf *spiked random* dapat mencapai nilai MSE hingga  $4,1898e-09$  dan  $1,4005e-05$  untuk 100 iterasi, sedangkan MSE untuk *back propagation* mencapai  $0,003718$  dan  $0,0022392$ . Dalam tahap pengujian, hasil yang baik dapat dicapai jika MSE sudah mencapai nilai sama atau kurang dari  $2e-08$ .

## 1. Pendahuluan

Penelitian terhadap model jaringan saraf tiruan dimulai dari ilmu saraf otak. Dari penelitian tersebut diketahui bahwa pada jaringan saraf terdapat suatu unit pemroses yang dinamakan neuron dan hubungan antar neuron yang dinamakan sinaptik. Sinaptik melakukan pembangkitan (*excitatory*) atau penghambatan (*inhibitory*) terhadap sebuah neuron dari neuron lain, dan pada neuron dilakukan pemrosesan terhadap sinaptiknya. Selanjutnya model ini menjadi baku dengan penggambaran berikut : sinaptik adalah hubungan yang berbobot, neuron melakukan operasi penjumlahan sinyal masukan dari neuron lain melalui sinaptik, dan melakukan fungsi aktivasi terhadap jumlah sinyal masukan tersebut. Ingatan dinyatakan dalam bobot sinaptik, yang nilainya dapat berubah dan diperbaiki.

Dari Kemampuan belajar pada Jaringan Saraf Tiruan inilah yang merupakan kelebihan yang tidak dipunyai oleh metode konvensional. Jaringan Saraf Tiruan dengan algoritmanya dapat dilatih untuk mengenali dan mengidentifikasi pola-pola. Pada tugas akhir ini, model Jaringan Saraf Tiruan mempunyai kemampuan untuk mengenal pola-pola masukan yang disajikan dalam bentuk matriks dimana nanti akan tetap menghasilkan suatu keluaran yang diinginkan walaupun masukannya telah sedikit diubah.

Tujuan yang hendak dicapai dalam tugas akhir ini adalah merencanakan dan merancang suatu simulator untuk jaringan saraf tiruan yang dapat

digunakan untuk identifikasi pola kode derau-palsu komunikasi spektrum tersebar.

## 2. Pembatasan Masalah

Karena kompleksnya permasalahan yang terdapat pada suatu sistem jaringan saraf, maka perlu adanya batasan-batasan untuk menyederhanakan permasalahan, yaitu :

- Jaringan saraf tiruan yang digunakan adalah model jaringan saraf tiruan umpan maju dengan metode pembelajaran *back propagation* dan *spiked random*.
- Jenis pelatihan jaringan untuk identifikasi yang digunakan bersifat *off-line*.
- Arsitektur jaringan yang dipakai terdiri dari tiga *layer* dengan penentuan jumlah neuron dalam *hidden layer* secara coba-coba.
- Sebagai contoh peangaplikasiannya yaitu pada sisi penerima sistem komunikasi spektrum tersebar sebagai pengenalan kode derau-palsu.
- Pola yang dihadirkan berupa data bit biner (0,1) atau (-1,1) sesuai dengan pola kode derau-palsu untuk 7 bit kode dan 15 bit kode pada komunikasi spektrum tersebar yang digambarkan dalam bentuk matriks.
- Keluaran yang ditampilkan juga dalam bentuk data bit sebagai tanggapan atau respon dari data yang masuk.
- Simulasi menggunakan program Bantu Matlab.

## 2. Dasar Teori

Sebuah Jaringan saraf tiruan adalah suatu sistem pemrosesan informasi yang memiliki karakteristik unjuk kerja tertentu dalam hubungan dengan jaringan saraf biologis. Jaringan saraf tiruan dikembangkan sebagai generalisasi model matematis dari pengenalan insani atau saraf biologis, berdasarkan asumsi bahwa :

1. Pengenalan informasi terjadi pada beberapa elemen sederhana yang disebut neuron.
2. Isyarat yang lewat diantara neuron-neuron melalui rantai penghubung.
3. Tiap rantai penghubung mempunyai bobot penghubung yang dalam jaringan saraf merupakan pengali/penguat sinyal yang ditransmisikan.
4. Tiap neuron menerapkan suatu fungsi aktivasi ke masukan jaringan penjumlah bobot sinyal masukan untuk menentukan sinyal keluaran.

Suatu jaringan saraf tiruan dinyatakan oleh :

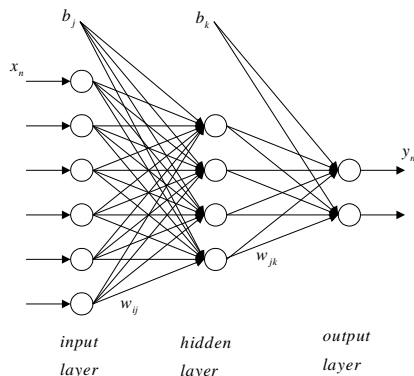
1. Pola-pola hubungan antar neuron yang disebut arsitektur jaringan saraf.
2. Metode penentuan bobotnya pada penghubung
3. Fungsi aktivasinya.

## 3. Jaringan Saraf Tiruan

Algoritma model pelatihan yang digunakan memakai dua buah model pelatihan :

### 3.1 Jaringan Saraf Back Propagation

Dalam pelatihan jaringan untuk model ini melalui tahapan dari *layer* ke *layer* begitu juga dengan perbaikan bobotnya melalui cara propagasi mundur dari *layer* ke *layer*. Error pada output menentukan perhitungan error pada *hidden layer* yang digunakan sebagai dasar penyesuaian bobot antara *input layer* dengan *hidden layer*.



Gambar 3.1 Arsitektur Jaringan Saraf Tiruan PropagasiMundur

Proses iterasi sendiri akan berhenti hingga error mencapai nilai minimum. Pemrosesan data dan pelatihan pola-pola masukannya melalui beberapa tahap:

### 1. Pelatihan umpan maju pola-pola input

Selama proses umpan maju setiap neuron input menerima sebuah sinyal dan memancarkan sinyal ini ke neuron-neuron pada *hidden layer*, dimana setiap neuron *hidden layer* melakukan perhitungan terhadap fungsi aktivasinya dan mengirimkan sinyal-sinyalnya ke setiap neuron pada output dimana keluarannya nanti merupakan respon jaringan tersebut dari setiap pola input yang diberikan.

$$v_j(n) = \sum_{i=1}^N w_{ji} z_i(n) \quad (3.1)$$

$$z_j(n) = f(v_j(n)) \quad (3.2)$$

dimana :

- $z_i(n)$  = keluaran dari *neuron* i pada input ke-n,
- $w_{ji}$  = bobot antara *neuron* j dan *neuron* i
- N = jumlah neuron pada layer dimana *neuron* i terletak.

### 2. Perhitungan Error yang bersangkutan

Selama pelatihan setiap output neuron membandingkan aktivasi output yang dihasilkan dengan nilai target yang diinginkan untuk menentukan error yang berhubungan dengan pola-pola pada unit tersebut. Berdasarkan pada error ini besaran koreksi kesalahan dapat dihitung dan digunakan untuk mendistribusibalikkan error yang terdapat pada output neuron ke semua neuron pada *layer* sebelumnya (*hidden neuron* yang berhubungan dengan output neuron). Koreksi kesalahan digunakan untuk memperbaiki bobot-bobot antara *output layer* dan *hidden layer*. Dengan cara yang sama besaran koreksi kesalahan juga dihitung pada setiap *hidden neuron*, tetapi koreksi kesalahan pada *hidden neuron* tidak digunakan untuk mempropagasibalikkan error *input layer*, tetapi digunakan untuk memperbaiki bobot-bobot antara *hidden layer* dan *input layer*.

$$e_k(n) = d_k(n) - y_k(n) \quad (3.3)$$

$$E = \frac{1}{2N} \sum_{n=1}^N \sum_{k=1}^M e_k^2(n) \quad (3.4)$$

dimana :

- M = jumlah neuron pada output layer
- N = jumlah pasangan input-output
- K = *neuron* ke-k pada output layer

### 3. Perbaikan Bobot

Setelah semua besaran koreksi kesalahan ditentukan, lalu bobot-bobot untuk semua *layer* disesuaikan. Penyesuaian terhadap bobot dari *hidden neuron* ke *output neuron* berdasar pada besaran koreksi kesalahan dan aktivasi dari *hidden neuron*, sedangkan penyesuaian bobot dari *input neuron* ke *hidden neuron* berdasar pada besaran koreksi kesalahan dan aktivasi dari *input neuron*. Setelah pelatihan selesai, aplikasi dari jaringan

hanya membutuhkan proses umpan maju untuk menghasilkan output.

$$w_{ji}(n+1) = w_{ji}(n) + \mu \delta_j(n) z_i(n) \quad (3.5)$$

dimana  $\delta$  didapat jika neuron k pada output layer, maka :

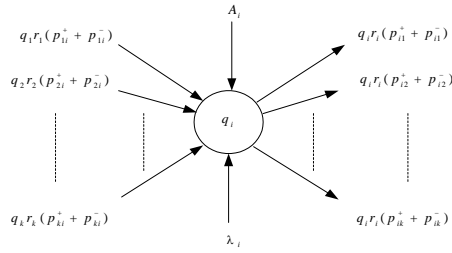
$$\delta_k(n) = e_k(n) f'(v_k(n)) \quad (3.6)$$

sedangkan jika neuron j adalah hidden layer, maka :

$$\delta_j(n) = \delta_k(n) w_{kj} f''(v_j(n)) \quad (3.7)$$

### 3.2 Jaringan Saraf Spiked Random

Model jaringan saraf tiruan ini pertama kali diperkenalkan oleh E. Gelenbe pada tahun 1989, dimana suatu neuron mendapatkan sinyal yang berupa sinyal negatif dan positif, dimana sinyal positif menggambarkan *excitation* dan sinyal negatif menggambarkan *inhibition*. Tiap keadaan suatu neuron merupakan integer non-negatif yang akan naik jika sinyal eksitasi menghampirinya dan akan menurun jika yang datang adalah sinyal inhibitory.



Gambar 3.2 Penggambaran Neuron Dalam Jaringan Saraf Model Spiked Random

Dimana :

$A_i$  = laju datang sinyal *excitatory* dari luar

$\lambda_i$  = laju datang sinyal *inhibitory* dari luar

$q_j r_j p_{ji}^+$  = sinyal *excitatory* dari neuron j

$q_j r_j p_{ji}^-$  = sinyal *inhibitory* dari neuron j

$q_i$  = kemungkinan suatu neuron dibangkitkan

Sinyal *excitatory* diinterpretasikan sebagai sinyal “+1” pada neuron penerima, sedangkan sinyal *inhibitory* diinterpretasikan sebagai sinyal “-1”. Suatu neuron i seperti pada Gambar 3.2 yang mengeluarkan sinyal apakah itu sinyal *excitatory* atau *inhibitory* kemungkinan besar akan kehilangan satu unit potential. Neuron i akan menembak atau firing (menjadi terbangkitkan) jika potentialnya positif. Lalu sinyal akan dikeluarkan pada laju  $r_i$ .

Sinyal akan dikirim ke beberapa neuron j dengan kemungkinan  $p_{ij}^+$  adalah sinyal *excitatory*, atau dengan kemungkinan  $p_{ij}^-$  adalah sinyal *inhibitory*. Suatu neuron juga dapat mengirimkan sinyal keluar jaringan dengan kemungkinan  $d_i$  dengan  $d_i + \sum_{j=1}^n (p_{ij}^+ + p_{ij}^-) = 1$ . Dimisalkan  $w_{ij}^+ = r_i p_{ij}^+$  dan  $w_{ij}^- = r_i p_{ij}^-$ , disini  $w$  adalah bobot dari neuron,

walaupun mereka secara spesifik menggambarkan laju dari sinyal *excitatory* dan *inhibitory* dimana mereka bernilai non-negatif. Sinyal *excitatory* dan *inhibitory* yang datang dari luar jaringan masing-masing datang ke neuron i pada laju  $A_i$  dan  $\lambda_i$ .

Salah satu hal yang penting dalam pemodelan suatu jaringan saraf tiruan adalah fungsi aktivasi dari tiap neuron, dimana merupakan suatu fungsi non-linier yang mengubah sinyal yang datang dalam bentuk *excitatory* ataupun *inhibitory* menjadi sebuah keluaran yang dibatasi pada suatu range tertentu. Dalam tugas akhir ini dimisalkan suatu fungsi aktivasi sebagai  $q_i$ , dimana  $q_i(t)$  adalah kemungkinan neuron i dibangkitkan pada waktu t, yang digambarkan sebagai berikut :

$$q_i = \lim_{t \rightarrow \infty} q_i(t) \quad , i = 1, 2, \dots, n \quad (3.8)$$

Untuk fungsi aktivasinya sendiri adalah :

$$q_i = \frac{\lambda_i^+}{(r_i + \lambda_i^-)} \quad (3.9)$$

dimana  $\lambda_i^+$  dan  $\lambda_i^-$  untuk  $i = 1, \dots, n$  merupakan persamaan dari system non-linier yang simultan :

$$\lambda_i^+ = A_i + \sum_{j=1}^n q_j w_{ji}^+ \quad (3.9)$$

$$\lambda_i^- = \lambda_i + \sum_{j=1}^n q_j w_{ji}^- \quad (3.10)$$

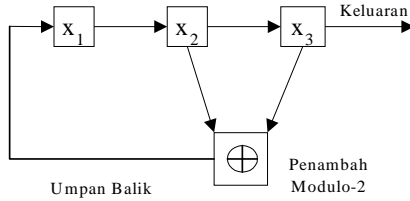
$$r_i = \sum_{j=1}^n (w_{ji}^+ + w_{ji}^-) \quad (3.11)$$

### 3.3 Pola Kode Derau Palsu

Walaupun referensi pada pengirim di komunikasi spektrum tersebar dapat menggunakan sinyal kode yang random untuk penyebaran dan penciutannya dikarenakan sinyal kode dan data yang dimodulasi secara simultan ditransmisikan lewat spektrum yang berbeda, tetapi referensi pada penerima tidak dapat menggunakan sinyal kode yang random karena kode harus juga dibangkitkan pada sisi penerima.

Untuk itu kode derau palsu tidak sepenuhnya random, tetapi merupakan sinyal periodik yang harus disinkronisasikan antara pengirim dan penerima. Dalam tugas akhir ini akan disimulasikan dua buah pola kode yaitu untuk pembangkit kode dengan 3 register geser dan 4 register geser. Untuk pembangkit kode dengan 3 register geser akan dihasilkan  $2^3 - 1 = 7$  bit kode. Pada Gambar 2.8 dapat dilihat contoh dari register geser umpan balik linier yang terbuat dari 3 tingkat register, penambah modulo-2 dan umpan balik dari penambah ke input register. Pengoperasian register geser diatur oleh pulsa clock. Pada tiap pulsa clock isi dari tiap tingkatan digeser satu tingkat ke kanan, juga untuk

tiap pulsa clock isi dari tingkatan  $x_2$  dan  $x_3$  dimodulo-2 dan hasilnya di umpan balikkan ke tingkatan  $x_1$ , sedangkan output dari register geser ini adalah keluaran dari  $x_3$ .



Gambar 3.1 Register Geser umpan balik linier 3 tahapan

Sebagai contoh dimisalkan  $x_1$  diisi dengan satu dan tingkatan lainnya diisi dengan nol, sehingga kondisi awal register menjadi 1 0 0. Dari Gambar 3.1 dapat dilihat pergeserannya adalah :

1 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0

Karena keadaan akhir adalah 1 0 0, sama dengan keadaan awal maka dapat dilihat bahwa baris ini akan berulang setelah 7 pulsa clock. Baris keluaran adalah dengan melihat isi pada tingkatan  $x_3$  pada tiap pulsa clock, yaitu :

0 0 1 0 1 1 1

Untuk mengubahnya menjadi *binary sequences*, diperlukan untuk mengidentifikasi +1 dengan +1 dan 0 dengan -1, sehingga deret diatas dapat ditulis kembali menjadi :

-1 -1 1 -1 1 1 1

Berikut ini tabel kondisi awal dan keluarannya untuk LFSR dengan  $n = 3$  tahapan register geser :

Tabel 2.1 Kondisi keluaran LFSR dengan  $n = 3$

Kondisi Awal	Keluaran
0 0 1	1 0 0 1 0 1 1
0 1 0	0 1 0 1 1 1 0
0 1 1	1 1 0 0 1 0 1
1 0 0	0 0 1 0 1 1 1
1 0 1	1 0 1 1 1 0 0
1 1 0	0 1 1 1 0 0 1
1 1 1	1 1 1 0 0 1 0

Saat deret ini diubah menjadi fungsi waktu, fungsi waktu ini akan berulang dengan sendirinya setiap periodik setiap  $7t_1$  detik. Dapat dilihat juga pada Tabel 2.1 semua kombinasi dari kondisi awal register geser, kecuali untuk semua nilai nol.

#### 4. Perancangan Dan Pengujian

Untuk perbandingan hasil simulasi dari kedua metode jaringan saraf, disediakan dua data pelatihan yang terdiri dari masing-masing pola masukan dan pola keluaran. Untuk pelatihan yang pertama pola input yang digunakan untuk pelatihan jaringan didasarkan pada pola kode derau palsu untuk 7 bit kode, yaitu :

- Pola 1 : -1 1 -1 -1 1 1 1
- Pola 2 : 1 -1 1 -1 -1 1 1
- Pola 3 : 1 1 -1 1 -1 -1 1
- Pola 4 : 1 1 1 -1 1 -1 -1
- Pola 5 : -1 1 1 1 -1 1 -1
- Pola 6 : -1 -1 1 1 1 -1 1
- Pola 7 : 1 -1 -1 1 1 1 -1

Untuk keluarannya dibuat suatu pola baru yang menandakan delay dari tiap pola yang masuk :

TARGET =

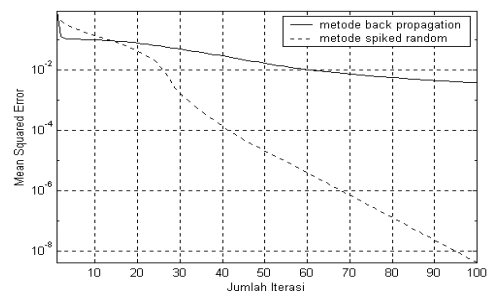
```
[ 0.5 0.5 0.5 0.5 0.5 0.5 1
  0.5 0.5 0.5 0.5 0.5 1 0.5
  0.5 0.5 0.5 0.5 1 0.5 0.5
  0.5 0.5 0.5 1 0.5 0.5 0.5
  0.5 0.5 1 0.5 0.5 0.5 0.5
  0.5 1 0.5 0.5 0.5 0.5 0.5
  1 0.5 0.5 0.5 0.5 0.5 0.5 ];
```

Inisialisasi parameter yang digunakan untuk tiap jaringan disiapkan pada nilai parameter yang sama untuk tiap metode, begitu juga dengan struktur jaringan sarafnya, yang terdiri dari :

- Input layer dengan 7 buah neuron (sesuai dengan pola masukan yang dipakaikan pada jaringan).
- Satu buah hidden layer dengan 8 buah neuron, dimana jumlah neuron dalam hidden layer ini ditentukan setelah dilakukan percobaan untuk mencari jumlah hidden neuron yang tepat.
- Output layer dengan 7 buah neuron (sesuai dengan pola keluaran untuk mencari tanggapan dari jaringan)

Tiap layer dari jaringan saraf tersebut saling berhubungan secara penuh (*fully connected*), sedangkan nilai laju ajar (*learning rate*) yang dipakai adalah 0,1 dengan nilai awal bobot random antara 0 hingga 0,1, serta nilai momentum yang biasa dipakai dalam metode *back propagation* adalah nol dalam arti untuk perbaikan bobotnya tidak memakai nilai momentum.

Dari hasil pelatihan jaringan berdasar parameter yang ada, berikut ini diperlihatkan grafik hasil pelatihan untuk 100 iterasi.



Gambar 3.3 Grafik pelatihan jaringan yang standar untuk 100 iterasi terhadap MSE untuk pola 7 bit kode

Dari Gambar 3.3 dapat dilihat bahwa pelatihan jaringan saraf model *spiked random* dapat lebih cepat mencapai nilai *mean squared error* (MSE) yang minimal dibandingkan dengan metode *back propagation* untuk jumlah iterasi yang sama. Untuk 100 kali iterasi metode *spiked random* dapat mencapai MSE  $4,1898e-09$ , sedangkan metode *back propagation* hanya mencapai MSE  $0,003718$ . Hasil pelatihan selengkapnya dapat dilihat pada Tabel 3.2 yang menampilkan pencapaian nilai *mean squared error* untuk tiap 10 iterasi hingga iterasi ke 100. Waktu yang tercatat pada tabel tersebut adalah waktu dalam hitungan detik yang dicapai oleh jaringan saraf dalam melakukan pelatihan untuk tiap iterasinya (waktu bisa semakin cepat tergantung PC yang digunakan).

Tabel 3.2 Hasil pelatihan jaringan untuk pola 7 bit

Iter ke	Metode Back Propagation		Metode Spiked Random	
	MSE	Waktu (det.)	MSE	Waktu (detik)
10	0,9512	6,1	0,13343	49,26
20	0,06111	11,21	0,041437	98,27
30	0,05231	16,73	0,001642	147,48
40	0,02834	21,88	0,000137	196,71
50	0,01627	26,1	2,103e-05	245,86
60	0,01002	37,3	3,963e-06	295,42
70	0,00707	38,62	7,283e-07	344,22
80	0,00544	44,21	1,312e-07	393,88
90	0,00430	50,1	2,352e-08	442,73
100	0,00371	55,42	4,189e-09	491,87

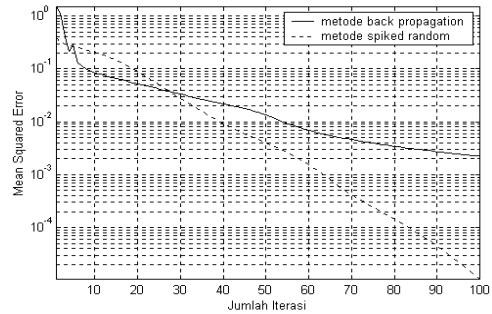
Waktu tiap iterasi pada metode *back propagation* memang jauh lebih cepat tetapi itu tidak dapat dijadikan patokan, yang menjadi patokan adalah laju konvergensi error untuk tiap iterasi. Untuk dapat mencapai nilai MSE seperti yang dicapai metode *spiked random* untuk 100 iterasi yaitu  $4,1898e-09$ , metode *back propagation* memerlukan hingga sekitar 25000 iterasi untuk mencapai nilai MSE tersebut dengan nilai parameter yang sama.

Pada pola kode 15 bit Tiap neuron dalam input layer akan dipakai sebagai masukan dari bit kode yang datang, sehingga jumlahnya disesuaikan dengan jumlah bit kode tersebut begitu juga dengan jumlah neuron dalam output layer, struktur jaringan yang dipakai juga menggunakan tiga buah layer yang selengkapnya dapat dijabarkan sebagai berikut

- Input layer dengan 15 neuron sebagai penerima bit kode yang masuk.
- Satu hidden layer yang terdiri dari 20 neuron.
- Output layer dengan 15 neuron sesuai dengan pola keluaran yang ditargetkan.

Pelatihan untuk masing-masing metode yang digunakan juga masih standar dengan nilai *learning rate* 0,1 dan inisialisasi bobot random 0 hingga 0,1. Dari hasil pelatihan jaringan berdasar parameter

yang ada, pada Gambar 3.4 memperlihatkan grafik hasil pelatihan untuk 100 iterasi.



Gambar 3.4 Grafik pelatihan jaringan yang standar untuk jumlah iterasi terhadap MSE untuk pola 15 bit

Dari grafik pelatihan pada Gambar 3.4 dapat dilihat bahwa pelatihan jaringan saraf model *spiked random* dapat lebih cepat mencapai nilai *mean squared error* (MSE) yang minimal dibandingkan dengan metode *back propagation* untuk jumlah iterasi yang sama. Untuk 100 kali iterasi metode *spiked random* dapat mencapai MSE  $1,4005e-05$ , sedangkan metode *back propagation* hanya mencapai MSE  $0,0022392$ . Tetapi jika dibandingkan dengan pola 7 bit kode metode *spiked random* mengalami kemunduran untuk laju konvergensi MSE, sedangkan untuk metode *back propagation* pelatihan semakin cepat yang berarti untuk data input yang semakin banyak laju konvergensi error bisa meningkat.

Hasil pelatihan selengkapnya dapat dilihat pada Tabel 3.3 sebagai perbandingan untuk kedua metode tersebut, metode *spiked random* mengalami penambahan waktu pelatihan untuk tiap iterasi karena perhitungannya menggunakan perhitungan untuk tiap pola yang masuk.

Tabel 3.3 Hasil pelatihan jaringan untuk pola 15 bit

Iter ke	Metode Back Propagation		Metode Spiked Random	
	MSE	Waktu (detik)	MSE	Waktu (detik)
10	0,0843	7,89	0,21276	970,24
20	0,05213	16,54	0,09765	1920,4
30	0,03434	23,43	0,02513	2897,6
40	0,02013	32,45	0,00856	3586,7
50	0,013265	41,43	0,001642	4892,4
60	0,006854	49,54	0,000745	5832,6
70	0,004432	55,32	0,000137	6821,7
80	0,003421	63,77	4,165e-4	7715,6
90	0,00276	72,78	6,067e-5	8609,9
100	0,002239	80,41	1,400e-5	9488,1

Setelah jaringan menjalani tahap pelatihan, maka dilakukan pengujian jaringan terhadap pola-pola masukan, dalam tugas akhir ini dilakukan pengujian terhadap empat macam pola masukan untuk masing-masing pola kode 7 bit dan 15 bit,

yaitu pola masukan yang sesuai dengan pola pelatihan, pola masukan dengan urutan pola yang telah diacak dan dua data pola masukan yang diberi noise. Dalam tahap pengujian, hasil yang baik dengan akurasi ketepatan hingga 100 % dapat dicapai jika MSE sudah mencapai nilai sama atau kurang dari  $2e-07$  untuk pola pengujian yang sama dengan masukan pola pelatihan, begitu juga dengan pola masukan yang urutannya diacak, jaringan dapat mengidentifikasi dengan baik dari tiap pola yang masuk, sedangkan untuk pola masukan yang diberi noise tanggapan jaringan tidak sempurna dengan semakin banyak noise makin buruk tanggapan keluarannya, walaupun untuk pola yang diberi noise  $\pm 30$  % dari pola pelatihan jaringan masih dapat mengklasifikasikannya ke dalam target keluaran dengan baik.

## 5. Kesimpulan

Setelah melakukan pembuatan simulasi, melakukan pelatihan dan pengujian jaringan terhadap pola kode derau palsu, maka penulis dapat menarik beberapa kesimpulan sebagai berikut :

1. Jaringan saraf tiruan dengan metode perbaikan bobot *spiked random* dan *back propagation* dapat digunakan untuk identifikasi pola kode derau palsu yang disajikan dalam bentuk biner.
2. Identifikasi pola kode derau palsu untuk pola 7 bit kode, setelah pelatihan jaringan menjalani 100 iterasi metode *spiked random* mampu mencapai MSE hingga  $4,1898e-09$  sedangkan metode *back propagation* mencapai MSE  $0,003718$ .
3. Identifikasi pola kode derau palsu untuk pola 15 bit kode, setelah pelatihan jaringan menjalani 100 iterasi metode *spiked random* mampu mencapai MSE hingga  $1,4005e-05$  sedangkan metode *back propagation* mencapai MSE  $0,0022392$ .
4. Kemampuan generalisasi dari jaringan saraf tiruan pada pemetaan pola input-output untuk pola kode derau palsu dengan metode perbaikan bobot *back propagation* maupun *spiked random* cenderung sama apabila kedua metode telah mencapai nilai error yang sama.
5. Pada metode *back propagation* nilai *learning rate* yang besar dapat mempercepat pelatihan jaringan untuk konvergensi error tetapi jika terlalu besar maka akan sering terjebak dalam local minima yang justru dapat memperlambat pelatihan.
6. Penambahan nilai momentum pada metode *back propagation* berguna untuk mengimbangi nilai *learning rate*, dimana nilai *learning rate* ( $\eta$ ) yang mendekati nol dengan nilai momentum ( $\alpha$ ) yang mendekati 1 cenderung akan mempercepat pelatihan walaupun efeknya kecil,

sedangkan nilai  $\eta$  yang mendekati 1 dengan memakai nilai  $\alpha$  yang semakin besar akan dapat menjaga kestabilan dari pelatihan jaringan.

7. Untuk masukan pola 7 bit kode pada metode *back propagation* semua nilai *learning rate* dapat dipakai dengan menggunakan nilai momentum konstan yang tepat, sedangkan untuk masukan pola 15 bit kode nilai *learning rate* yang dapat dipakai adalah hingga 0,1 karena jika terlalu besar laju konvergensi error tidak dapat berjalan dengan baik.
8. Untuk masukan pola 7 bit kode maupun 15 bit kode pada metode *spiked random* nilai *learning rate* yang tepat adalah 0,1 dimana pelatihan jaringan dapat berjalan dengan baik dan lebih cepat.

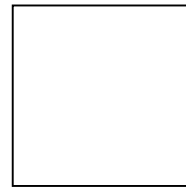
## Saran

1. Pelatihan Jaringan saraf tiruan pada tugas akhir ini dilakukan secara *off-line* sehingga walaupun jaringan dapat memberikan tanggapan yang baik terhadap pola yang masuk tetapi jika pola masukan yang diberikan mengalami perubahan atau terdapat gangguan tanggapan yang diberikan tidak begitu sempurna, sehingga akan lebih baik jika pelatihan jaringan dilakukan secara *on-line*.
2. Penelitian lebih lanjut dapat dilakukan dengan metode yang lain, seperti pengembangan dari metode *spiked random* dalam bentuk bipolar GNN yang memiliki dua jenis neuron dalam arsitekturnya yang mempunyai peran yang berlawanan. Hasil dari identifikasi pola kode derau palsu juga dapat lebih dikembangkan dengan mencoba merancang jaringan saraf tiruan yang dapat mengidentifikasi pola tanpa mengetahui jumlah bit pola yang dikirim oleh pengirim.

## DAFTAR PUSTAKA

1. Alexander, Fegor & Morton, Helm, *An Introduction to Neural computing*, Chapman & Hall, UK, 1991.
2. B.Muller and J.Reinhardt, *Neural Networks*, Spring Verlag, Berlin, 1990.
3. Cichoki and R. Unbehaven, *Neural Networks for optimisation and signal processing*, John Wiley & Sons, Inc ; New York, NY, 1993.
4. C. Cramer, E. Gelenbe and H. Bakircioglu, *Low Bit Rate Video Compression with Neural Network and Temporal Subsampling*, Proceeding of the IEEE, Vol. 84, No. 10 pp. 1529-1543, October 1996.
5. Dayhoff, J.E, *Neural Networks Architecture*, Van Nostrad Reinhold, New York, 1990.
6. Erol Gelenbe, Zhi-Hong Mao and Yan-Da Li, *Function approximation with RNN*, IEEE

- Transaction on Neural Networks, Vol. 10, No. 1, January 1999.
7. Fauseet, Laurence, *Fundamental of Neural Networks., Architecture , Algorithms and Applications*, Prentice-Hall Inc, USA, 1994.
  8. Freeman J.A & Shapura P.M, *Neural Networks : Algorithms, Applications and Programming Technique*, Addison Wesley Publishing Company Inc, USA,1994.
  9. Haylen, Simon, *Neural Networks A Comprehensive Foundation*, MacMillan College Publishing co Inc, USA, 1994.
  10. Herbert Taub and Donald L. Schilling, *Principles of Communication System*, second edition McGraw-Hill, Inc.
  11. Hermawanto, W , Jaringan Saraf Tiruan Back Propagation Untuk Diagnosa Penyakit Dan Kelainan Pada Fungsi Hati, Tugas Akhir Teknik Elektro, Undip, Semarang, 1999.
  12. Huch Nielsen, *Neurocomputing : Picking The Human Brain*, IEEE Spectrum, March '88.
  13. Kosho , Bort, *Neural Network And Fuzzy System*, Prentice-Hall Inc, USA, 1992.
  14. Narendra, K. S., K. Parthasarathy, *Identification and control of dynamical systems using neural networks*, IEEE Trans. Neural Networks, 1, 4-27, 1990.
  15. Nelson, Illingworth, *A practical Guide to Neural Nets*, Texas Instruments, 1993.
  16. Newsgroup comp.ai.neural-nets, *Capabilities of Artificial Neural Networks and Neural-Network-like structures*.  
URL: <ftp://ftp.sas.com/pub/neural/FAQ.html>
  17. Cooper, D.McGillem, *Modern Communications and Spread Spectrum*, McGraw-Hill, Electrical Engineering Series.
  18. R.Lipmann, *An Introduction to Computing With Neural Networks*, IEEE Magazine pp 4-22 April 1987.
  19. Sandi Setiawan, *Artificial Intelligence*, Penerbit ANDI Offset Yogyakarta.
  20. Suparman, *Mengenal Artificial Intelligence*, Penerbit ANDI Offset Yogyakarta.
  21. Warwick, G.W.Irwin, K.J.Hunt, *Neural Networks for control and Systems*, IEEE Control Engineering Series, 1992.



Eka Saputra, Lahir 14 September 1978, saat ini sedang menyelesaikan pendidikan Strata 1 di Teknik Elektro Universitas Diponegoro Semarang. Konsentrasi yang diambil adalah Kontrol

Bidang yang diminati adalah teknologi informasi, jaringan saraf , system kontrol dan pemrosesan sinyal.

Disetujui Oleh  
Dosen Pembimbing

Sumardi ST, MT  
NIP. 132 125 670