

ISSN 0853-6732

J U R N A L

# TEKNIK KOMPUTER

Vol. 12, No. 1, Februari 2004

**Simulasi Sistem Navigasi pada *Mobile Robot***

Iman H. Kartowisastro; Ari Pandi;  
Salfikar Alfarizi Abbas; Yufran Madaharjaya

**Pengenalan Objek Sederhana Menggunakan Logika *Fuzzy***

Ario Witjakso; Kamal Umar Muhammad; Yoko

**Media Kontrol dengan Teknologi SMS**

Eddy Santosa Jaya; Ronny Diarsa; Susanto

**Implementasi Kompresi Data dengan *Huffman Coding*  
ke dalam FPGA**

Lukas Tanutama; David Kurniawan; Agnes; Cun Cun

**Perancangan *Mobile Robot* dengan Sistem Kontrol PC  
melalui Media Komunikasi RF**

Dimas Bram Prayogo; Wiedjaja; Ferdinan; Nordin

**Pengontrolan Suhu dengan *Fuzzy Logic***

Aditya Tri Wicaksono; Ario Witjakso;  
Herry Andayana Suwardji; Krisna Wijaya

***Frequency Recognition* untuk Prototipe Sistem Pengontrolan  
Gerbang dengan TMS320C54X DSP dan AT89C51**

Satrio Dewanto; Jeffry; Heriyanto; Apriyanto

**Pengenalan Nada Suara Gitar dan Piano  
dengan *Back Propagation Neural Network***

Wiedjaja; Pangbun; Abram I.S.; Wahab



J. Teknik Komputer	Vol. 12	No. 1	Hlm. 1-95	Jakarta Februari 2004	ISSN 0853-6732
-----------------------	---------	-------	-----------	--------------------------	-------------------

**JURNAL  
TEKNIK KOMPUTER**

Volume 12 Nomor 1, Februari 2004

Jurnal Ilmiah  
Bidang Ilmu Komputer  
Universitas Bina Nusantara

**Pelindung:**

Dr. Th. Widia Soerjaningsih

**Penanggung Jawab:**

Prof. Dr. Gerardus Polla, M.App.Sc.

**Ketua Penyunting:**

Iman H. Kartowisastro, Ph.D.

**Penyunting Pelaksana:**

Drs. Satrio Dewanto, M.Eng.

Ir. Lukas Tanutama, M.M.

Fransiscus A. Halim, S.Kom., M.M.

Jurike V. Moniaga, S.Kom.

Dr. Ir. Haryono Soeparno, M.Sc.

**Mitra Bestari:**

Prof. Dr. Ir. R. J. Widodo, M.Sc.

(Jurusan Teknik Elektro ITB)

Sardjono Trihatmo, Dipl. Ing.

(P3TIE, BPP Teknologi)

**Penyunting Bahasa:**

Dra. Endang Ernawati, M.Lib.

Titik Rahayu S., S.S.

Agung Pramanto, S.S.

**Sekretariat:**

Hery H.M., S.Kom.

Tri Septi Wulansari

Harkat Christian Zamasi

**Alamat Redaksi:**

Subbiro Publikasi Ilmiah

Bidang Ilmu Komputer

Biro Penelitian dan Publikasi

Universitas Bina Nusantara

Jl. Kebon Jeruk Raya No. 27,

Kemanggisan, Jakarta 11530

Telp. (021) 5345830, 5350660

ext. 1706, Fax. (021) 5300244

ISSN 0853-6732

**BINUS UNIVERSITY**

**DAFTAR ISI**

Kata Pengantar.....	iii
<b>Iman H. Kartowisastro; Ari Pandi; Salfikar Alfarizi Abbas; Yufran Madaharjaya</b> Simulasi Sistem Navigasi pada <i>Mobile Robot</i> .....	1-14
<b>Ario Witjakso; Kamal Umar Muhammad; Yoko</b> Pengenalan Objek Sederhana Menggunakan Logika <i>Fuzzy</i> .....	15-28
<b>Eddy Santosa Jaya; Ronny Diarsa; Susanto</b> Media Kontrol dengan Teknologi SMS.....	29-43
<b>Lukas Tanutama; David Kurniawan; Agnes; Cun Cun</b> Implementasi Kompresi Data dengan Huffman <i>Coding</i> ke dalam FPGA.....	44-53
<b>Dimas Bram Prayogo; Wiedjaja; Ferdinan; Nordin</b> Perancangan <i>Mobile Robot</i> dengan Sistem Kontrol PC melalui Media Komunikasi RF.....	54-62
<b>Aditya Tri Wicaksono; Ario Witjakso; Herry Andayana Suwardji; Krisna Wijaya</b> Pengontrolan Suhu dengan <i>Fuzzy Logic</i> .....	63-73
<b>Satrio Dewanto; Jeffry; Heriyanto; Apriyanto</b> <i>Frequency Recognition</i> untuk Prototipe Sistem Pengontrolan Gerbang dengan TMS320C54X DSP dan AT89C51.....	74-84
<b>Wiedjaja; Pangbun; Abram I.S.; Wahab</b> Pengenalan Nada Suara Gitar dan Piano dengan <i>Back Propagation Neural Network</i> .....	85-95

ation. New York:  
30. Canada: John  
axwell Macmillan  
Recognition. New

# Pengenalan Nada Suara Gitar dan Piano dengan *Back Propagation Neural Network*

Wiedjaja; Pangbun; Abram I.S.; Wahab

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Bina Nusantara, Jakarta

---

---

## ABSTRACT

*The purposes of this research is to recognize music sounds tone from instrument such as guitar and piano. The input signals will be modified from time domain to frequency domain using Fast Fourier Transform (FFT). Then, the score from FFT will be processed to get significant point that represents the nature of input signals. Those significant points are inputed into Neural Network Back Propagation to know what tone are the input signal. Recognition by Back Propagation Neural Network is expected to be identified tone from guitar and piano.*

*Keywords: sound, tone, guitar, piano, Back Propagation Neural Network*

## ABSTRAK

*Tujuan penelitian ialah untuk mengenal nada suara musik yang dikeluarkan dari alat musik seperti gitar dan piano. Sinyal yang di-input akan diubah dari domain waktu ke domain frekuensi dengan Fast Fourier Transform (FFT). Lalu, nilai yang didapat dari FFT akan diproses untuk mendapatkan titik penting yang mewakili sifat dari sinyal input. Titik penting tersebut dijadikan input ke Jaringan Syaraf Tiruan (Back Propagation Neural Network) untuk mengetahui nada apakah sinyal input tersebut. Pengenalan oleh neural network itu diharapkan dapat mengidentifikasi nada dari alat musik gitar dan piano.*

*Kata kunci: nada, suara, gitar, piano, jaringan syaraf tiruan*

---

---

## PENDAHULUAN

Sejak pertama kali diciptakan, komputer telah mengalami banyak perubahan baik dari segi arsitektur maupun dari segi perangkat lunak. Komputer yang awalnya digunakan manusia untuk membantunya dalam proses perhitungan saja, saat ini telah mengalami perluasan fungsi. Para ahli dari berbagai bidang ilmu pengetahuan, baik psikologi, biologi, sampai komputer, menyelidiki serta mempelajari cara kerja sistem syaraf manusia yang luar biasa. Otak manusia dengan jaringan syarafnya yang kompleks memiliki kemampuan untuk mempelajari, mengingat, dan mengembangkan sesuatu. Berdasarkan penelitian tersebut, para ahli dalam bidang teknologi komputer mencoba merancang suatu sistem kecerdasan buatan yang dapat memiliki kemampuan, seperti sistem syaraf dalam skala dan pendekatan yang berbeda. Oleh karena itu, lahirlah yang dikenal dengan jaringan syaraf tiruan. Dengan adanya jaringan syaraf tiruan itu, keterbatasan metode komputasi biasa pada masalah kompleks dapat teratasi.

### **Pengertian File WAV (Waveform Audio File Format)**

*File WAV (Waveform Audio File Format)* merupakan format *file digital audio* yang disimpan dalam bentuk *digital* dengan ekstensi WAV. *File* itu menyimpan amplitudo dan frekuensi suara sehingga membutuhkan tempat penyimpanan yang besar. *File Wav* itu dikembangkan oleh *Microsoft* dengan format PCM. *File WAV* dibagi 3 bagian dalam menyimpan informasi, masing-masing bagian disebut *chunk*.

1. *Chunk* yang pertama adalah *chunk RIFF* dan *chunk* itu menyatakan bahwa *file* tersebut adalah *file WAV*.
2. *Chunk* yang kedua adalah *chunk FORMAT*. *Chunk* itu berisi parameter atau sifat *file WAV*, misalnya parameter *samplerate*.
3. *Chunk* yang ketiga adalah *chunk DATA*. *Chunk* itu berisi data dari sinyal sesungguhnya.

### **Fast Fourier Transform (FFT)**

*Fast Fourier Transform (FFT)* adalah algoritma yang menghitung DFT secara cepat, secara khusus untuk mempersiapkan data yang akan digunakan pada *neural network*. Dengan kata lain, transformasi Fourier berfungsi sebagai alat bantu untuk menganalisis karakteristik sinyal dalam hal frekuensi (*frequency domain*). Juga, merupakan suatu urutan langkah yang digunakan untuk membandingkan DFT (*Diskret Fourier Transform*) dan inversinya. DFT bermanfaat sebagai metode numerik yang menghitung transformasi Fourier dari suatu fungsi kontinu. Dengan FFT maka jumlah perhitungan dapat dikurangi secara dramatis. Penemuan FFT telah membuat perhitungan digital terhadap spektrum frekuensi dapat diwujudkan dalam praktik.

### **Mencari Frekuensi Utama**

Setelah didapatkan hasil FFT yang berupa spektrum dalam *frekuensi domain*, tidak semua titik dari *frekuensi domain* itu memiliki nilai yang diperlukan. Bahkan, sebagian besar titik tersebut adalah titik tidak berguna. Untuk efisiensi dan penghematan waktu bagi *neural network* untuk *learning*, alangkah baiknya jika yang dipelajari (*training/learning*) adalah titik utamanya saja yang tentu jumlahnya jauh lebih sedikit dari jumlah seluruh titik frekuensi. Untuk mendapatkan titik utama atau frekuensi utama, ada beberapa algoritma yang dapat digunakan, diantaranya sebagai berikut.

1. Mencari titik utama berdasarkan nilai rata-rata frekuensi yang penting saja dalam tiap perulangan atau iterasi.
2. Mencari titik utama berdasarkan nilai rata-rata frekuensi ditambah sepuluh persen nilai rata-rata frekuensi pertama untuk tiap perulangan.
3. Mencari titik utama berdasarkan dua metode di atas. Metode itu akan mencari titik utama dengan mengambil keputusan yang terbaik diantara dua metode di atas.

### Pengertian Jaringan Syaraf Tiruan – Propagasi Balik

Jaringan Syaraf Tiruan-Propagasi Balik atau *Back Propagation* adalah jaringan syaraf tiruan yang berkoneksi penuh, berlapis, dan berumpan maju (*feedforward*) yang merupakan salah satu jaringan penting dalam perkembangan Jaringan Syaraf Tiruan. Aplikasi Propagasi Balik itu dapat dicontohkan sebagai berikut: lengan robot, pemrosesan citra, pemrosesan sintesis pengucapan, pengenalan nada, dan lain-lain. Tujuan propagasi balik itu untuk mendapat keseimbangan antara kemampuan bereaksi terhadap pola masukan saat pelatihan dengan pola masukan saat dijalankan. Dengan kata lain, berusaha untuk meminimalkan perbedaan antara nilai keluaran dari jaringan dengan nilai keluaran yang diharapkan (sasaran saat pelatihan dilakukan dengan meminimalkan jumlah kesalahan kuadrat dari perhitungan keluaran jaringan). Propagasi Balik mempergunakan teknik pelatihan dengan pengawasan dan jaringan yang berlapis banyak, sangat bermanfaat dalam memecahkan masalah dalam cakupan yang luas. Metode pelatihan ini meliputi 3 (tiga) tahapan berikut.

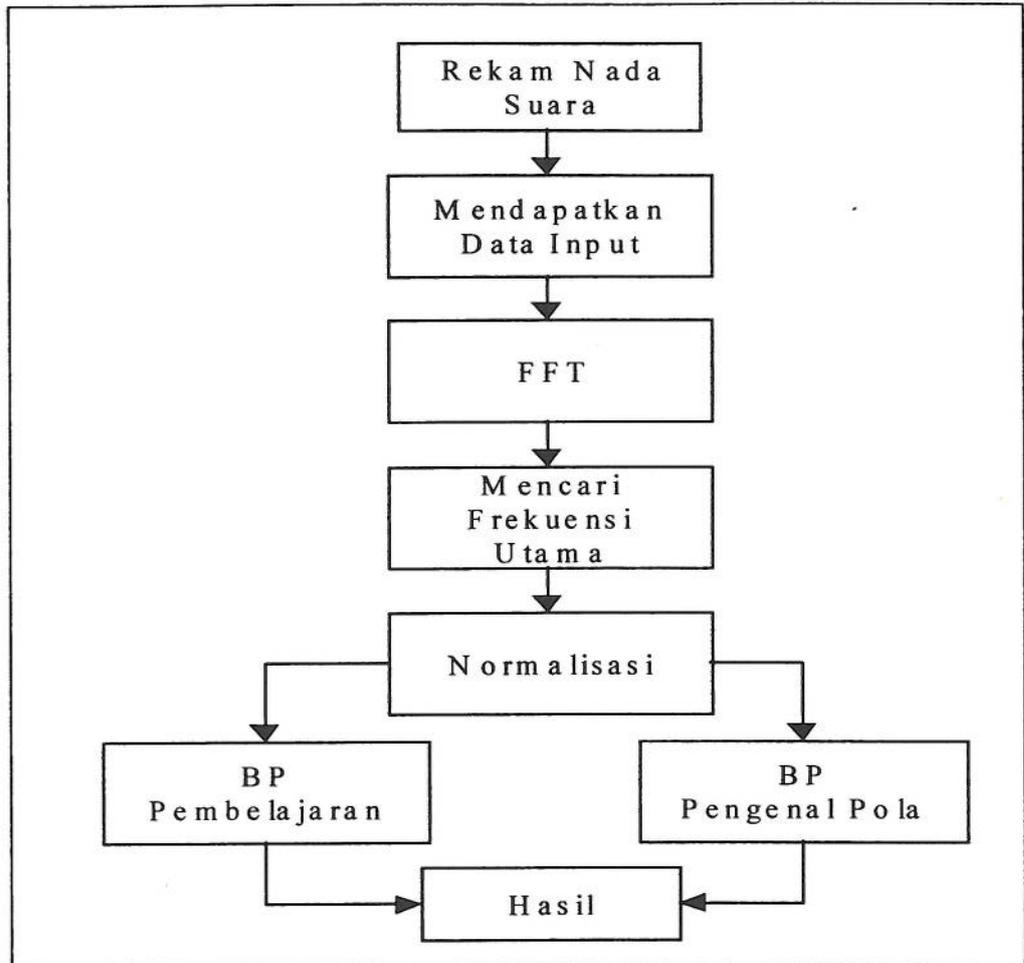
1. Umpan maju dengan nilai pola masukan pelatihan.
2. Perhitungan dan umpan balik kesalahan yang terjadi pada jaringan.
3. Penyesuaian terhadap bobot koneksi.

Setelah proses pelatihan sesuai dilakukan, aplikasi hanya diterapkan pada tahapan perhitungan umpan maju.

## PEMBAHASAN

### Cara Kerja Program

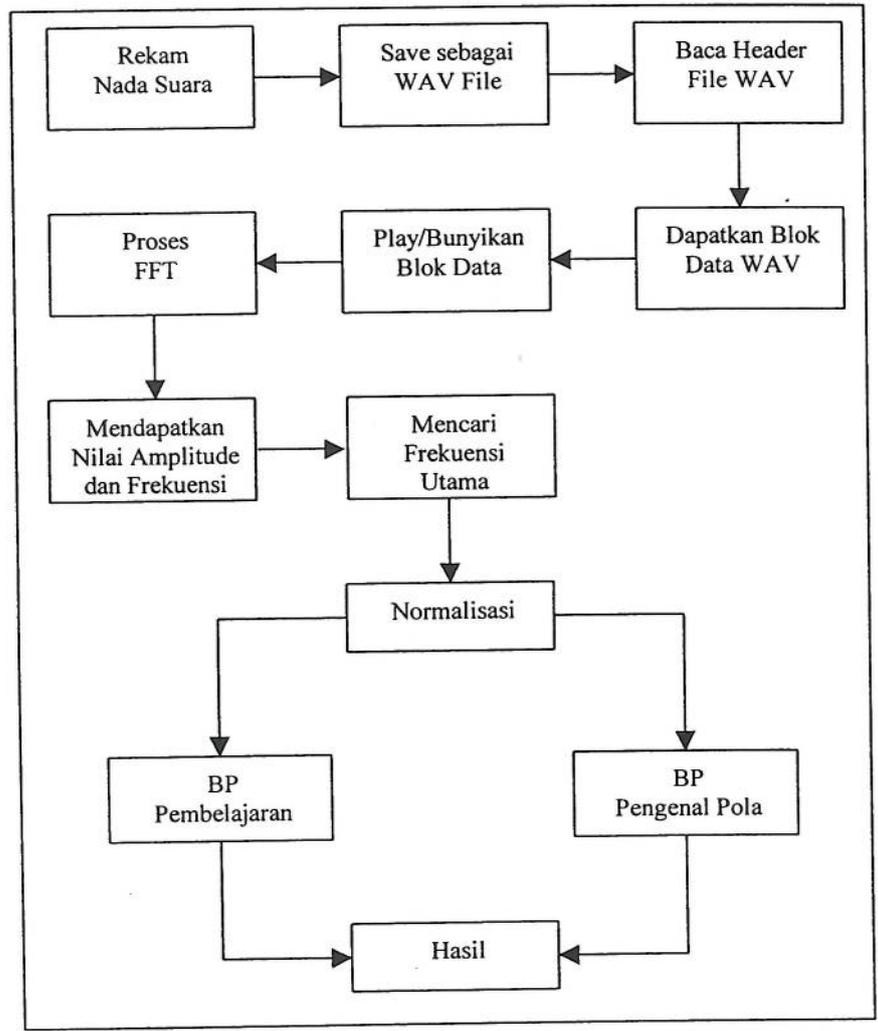
Urutan cara kerja perangkat lunak pengenalan nada ini, dapat dilihat pada Gambar 1. Langkah pertama yang dilakukan adalah merekam nada suara gitar dan piano melalui *jack line in* atau *jack mic* pada *soundcard*. Pada *soundcard*, terjadi proses perubahan dari nada yang diterima berupa sinyal analog diubah menjadi sinyal digital. Data tersebut diolah lagi agar dapat diterima menjadi *input* dari FFT (*Fast Fourier Transform*). Dengan *Fast Fourier Transform* akan didapatkan data berupa titik frekuensi lalu setelah dicari titik frekuensi utama, langkah selanjutnya proses pembelajaran atau pengenalan nada oleh Jaringan Propagasi Balik dapat dilakukan. Untuk lebih jelasnya, langkah proses dapat dilihat pada blok diagram di bawah ini.



Gambar 1 Proses Pengenalan Nada

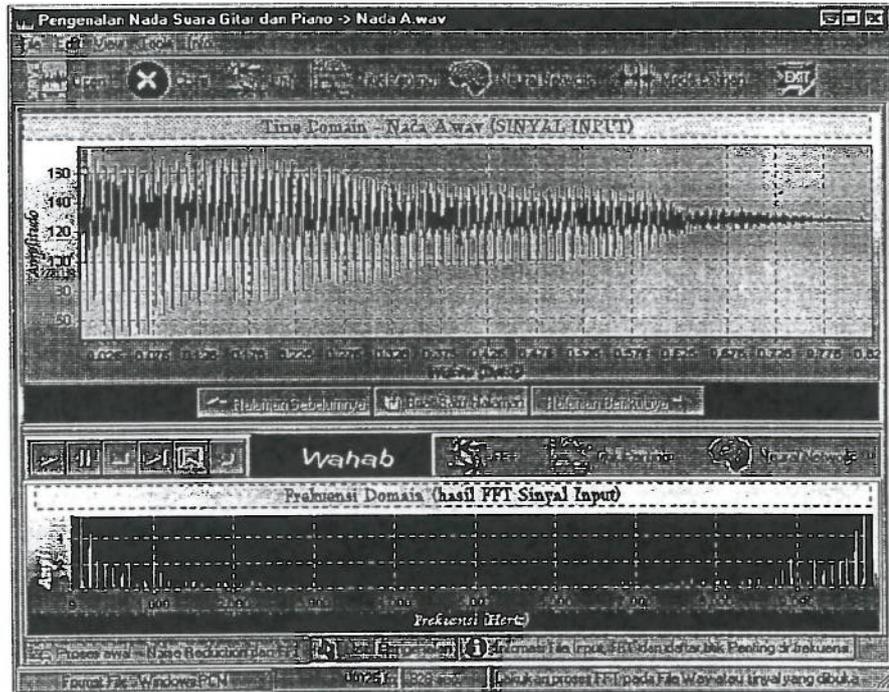
Untuk lebih jelas, dari Gambar 1 dapat dilihat dalam Gambar 2 berikut yang lebih rinci.

P o l a



Gambar 2 Gambaran Lengkap Proses Pengenalan Nada

yang lebih rinci.



Gambar 3 Form Utama Program Pengenal Nada Gitar dan Piano

## Analisis dan Hasil Pengenalan

### Analisis Pelatihan *Back Propagation* untuk Mencari *Setting* yang Optimal

Tabel 1 (Konfigurasi 100-100-1) Jumlah *Neuron* pada *Output Layer* adalah 1

Alpha	Teachrate	Momentum	N-LD	Epoch	MaxErr	MidErr	Lamanya
<b>Pengaruh Alpha terhadap Error</b>							
Dapat dilihat bahwa semakin besar nilai <i>alpha</i> maka <i>error</i> akan semakin besar dan waktu proses pembelajaran akan bertambah sedikit lebih lama.							
0.1	0.1	0.9	100	500	0.15245	0.02210	2:53.697
0.5	0.1	0.9	100	500	0.18701	0.03782	2:58.938
1	0.1	0.9	100	500	0.20589	0.03953	3:1.103



**Pengaruh Teachrate terhadap Error**

Dapat dilihat bahwa semakin besar nilai *teachrate* maka *error* akan semakin kecil dan waktu proses pembelajaran akan bertambah sedikit lebih cepat.

0.1	0.1	0.9	100	5000	0.00974	0.00107	30:2.969
0.1	0.9	0.9	100	5000	0.00080	0.00004	28:58.64

**Pengaruh Momentum terhadap Error**

Dapat dilihat bahwa semakin besar nilai *momentum* maka *error* akan semakin kecil.

0.1	0.1	0.1	100	500	0.15063	0.04813	2:53.119
0.1	0.1	0.9	100	500	0.03517	0.00726	2:17.810
0.1	0.1	1	100	500	0.05541	0.00410	3:3.524

**Pengaruh Jumlah Neuron pada Lapisan Dalam terhadap Error**

Dapat dilihat bahwa semakin besar nilai Jumlah Neuron pada Lapisan Dalam maka *maxerror* akan semakin kecil namun *miderror* sedikit lebih besar. Waktu proses pembelajaran akan bertambah secara linear sesuai dengan jumlah neuron.

1	0.1	0.9	100	3500	0.24813	0.03867	20:24.812
1	0.1	0.9	200	3500	0.22934	0.03900	41:55.919

**Pengaruh Jumlah Epoch terhadap Error**

Terlihat jelas bahwa semakin besar nilai Jumlah *Epoch* maka *error* akan semakin kecil dan waktu proses pembelajaran semakin lama.

1	0.1	0.9	100	500	0.20589	0.03953	3:1.103
1	0.1	0.9	100	1500	0.21767	0.03913	8:52.58
1	0.1	0.9	100	2500	0.20783	0.03836	15:0.309
1	0.1	0.9	100	3500	0.24813	0.03867	20:24.812
1	0.1	0.9	100	5000	0.20024	0.03786	29:19.932
1	0.1	0.9	100	10000	0.21650	0.03770	57:49.109

h 1

MaxErr	Lamanya
0.210	2:53.697
0.1782	2:58.938
0.1953	3:1.103

an waktu proses

Tabel 2 (Konfigurasi 100-100-14)

Alpha	Teachrate	Momentum	N-LD	Epoch	MaxErr<	MidErr<	Lamanya
0.1	0.9	0.9	100	1000	0.50187	0.04525	0:08:49.261
0.1	0.9	0.9	100	2500	0.50084	0.03805	0:22:59.228
0.1	0.9	0.9	100	5000	0.50021	0.03783	0:46:20.659
0.1	0.9	0.9	100	10000	0.50012	0.03679	1:37:02.245
0.1	0.9	0.9	100	34000	0.26135	0.03211	6:13:20.585
0.1	0.4	0.9	100	5000	0.50059	0.03765	0:51:06.929
0.1	0.9	0.4	100	10000	0.50072	0.03810	1:33:41.737

Tabel 3 Persentase Pengenalan Keseluruhan Nada

No	Alpha	Teachrate	Momentum	N-LD	Epoch	Errorrate	Persentase Pengenalan
Di bawah ini Train 5 <i>Input</i> untuk Tiap Nada							
1	0.1	0.9	0.9	100	5000	0.00020	56.73 %
2	0.1	0.9	0.5	100	5000	0.00182	56.19 %
Di bawah ini Train 8 <i>Input</i> untuk Tiap Nada							
3	0.1	0.99	0.99	100	5000	0.03526	91.45 %
4	0.1	0.9	0.1	100	5000	0.04360	82.88 %
5	0.1	0.9	0.9	100	5000	0.03703	98.08 %
6	0.1	0.1	0.1	100	5000	0.33890	15.39 %
7	0.1	0.99	0.1	100	5000	0.04629	93.41 %
8	0.1	0.99	0.1	100	12000	0.03829	95.78 %
Pengaruh <i>Error Rate</i> terhadap Persentase Pengenalan							
9	0.1	0.9	0.9	100	1000	0.04525	96.01 %
10	0.1	0.9	0.9	100	2500	0.03805	97.13 %
11	0.1	0.9	0.9	100	5000	0.03783	98.08 %
12	0.1	0.9	0.9	100	10000	0.03679	98.43 %
13	0.1	0.9	0.9	100	34000	0.03211	98.90 %
14	0.1	0.4	0.9	100	5000	0.03765	97.04 %
15	0.1	0.9	0.4	100	10000	0.03810	97.23 %

Persentase pengenalan pada tabel tersebut didapat dengan mencari nilai rata-rata untuk tiap sel yang berwarna biru pada tabel hasil pengenalan 1 sampai 15 di lampiran A. Pada tahap pertama, digunakan 5 *input* untuk tiap nada sebagai *input* pada pembelajaran *back propagation* dan setelah dites ternyata kemampuannya tidaklah menghasilkan pengenalan yang memuaskan. Hal itu dapat dilihat pada hasil pengenalan Tabel 3 di atas. Pada baris ke-1 dan ke-2 memiliki

persentase pengenalan 56.73% dan 56.19%. Langkah selanjutnya, untuk memperoleh kemampuan pengenalan yang lebih baik maka digunakan 8 *input* untuk tiap nada sebagai *input* pembelajaran *back propagation*. Hasil dari proses pembelajaran dan pengenalan nada dapat dilihat pada Tabel 3 pada baris ke-3 sampai baris ke-8.

Setelah diperhatikan, pada tabel 3 antara baris nomor 3 sampai 8 yang memiliki *error rate* terkecil, yaitu dengan parameter pembelajaran seperti pada baris ke-3. Ternyata, setelah dites, hasil pengenalan nada lebih baik dibanding hasil pada baris ke-1 dan ke-2 yang menggunakan 5 *input* untuk tiap nada. Namun, jika diperhatikan lebih teliti lagi, ternyata hasil pengenalan pada baris ke-3 itu tidaklah sebaik hasil pengenalan pada baris ke-5 yang memiliki parameter pembelajaran seperti di bawah ini.

1.  $\alpha=0.1$ ,  $teachrate=0.9$ ,  $momentum=0.9$ ,  $epochcount=5000$ , lapisan dalam=100

Jadi, *setting* yang paling optimal dalam pengenalan nada adalah seperti yang tertera pada baris ke-5 di tabel 3, walaupun *error rate*-nya lebih besar dari *error rate* pada baris ke-3 namun kemampuannya lebih baik. Baris ke-9 sampai ke-15 membuktikan bahwa *error rate* yang semakin kecil membuat persentase pengenalan semakin bagus. Perhatikan pada baris ke-13 yang memiliki persentase pengenalan terbesar, yaitu sebesar 98.90 %. Namun, perlu diperhatikan bahwa pada baris ke-9 sampai ke-13 merupakan hasil pelatihan dengan parameter pembelajaran yang sama, hanya yang diubah adalah jumlah *epoch* untuk mendapatkan *error rate* yang lebih kecil.

2.  $\alpha=0.1$ ,  $teachrate=0.9$ ,  $momentum=0.9$

Jadi, terbukti bahwa *error rate* yang semakin kecil membuat persentase pengenalan menjadi semakin besar/bagus. Namun, hal itu dapat dicapai jika dilatih dengan nilai  $\alpha$ ,  $teachrate$ , dan  $momentum$  yang tetap, yang diubah hanya jumlah *epoch*. Jika dilatih dengan *setting* pelatihan yang berbeda namun dengan *error rate* yang mendekati sama maka akan didapat hasil seperti pada pada baris 11 dan 14. Pada baris ke-14, didapat nilai *error rate* sebesar 0.03765 yang ternyata lebih kecil dari baris ke-11, yaitu sebesar 0.03783. Namun, persentase pengenalan baris ke-14 lebih buruk/kecil dari baris ke-11. Padahal, seharusnya jika didapat *error rate* yang kecil maka persentase pengenalan akan semakin bagus/besar. Bukti konkret lainnya dapat dilihat pada baris ke-1 yang memiliki *error rate* yang kecil sekali, yaitu 0.00020. Namun, sangat disayangkan sekali hasil pengenalannya tidaklah memuaskan. Lihat saja jika dibandingkan pada baris nomor 3 sampai 15, hampir semua pada baris itu memiliki persentase pengenalan yang lebih besar dari baris ke-1. Jadi, hal itu membuktikan bahwa *error rate* yang kecil belum tentu menghasilkan persentase pengenalan bagus. Semua itu tergantung kepada *setting* parameter pelatihan  $\alpha$ ,  $momentum$ , dan  $teachrate$ , serta jumlah *input* untuk tiap nada.

Sekarang, kenapa *setting* yang optimal dipilih pada baris ke-5 dan ke-11, yaitu karena efisiensi waktu dan efisiensi pengenalan. Coba saja jika dibandingkan pada baris ke-11 dan ke-13, berapa persentase perbedaan antara keduanya? Pada baris ke-11, didapat persentase pengenalan sebesar 98.08 % dan pada baris ke-13 sebesar 98.90 %. Jika dikurangkan, keduanya hanya memiliki selisih 0.82 %. Bukankah itu nilai yang kecil jika dibandingkan dengan lamanya waktu yang diperlukan saat proses pelatihan. Perhatikan pada Tabel 2, bahwa untuk *train* dengan jumlah *epoch* sebesar 34000 memerlukan waktu sampai 6 jam 13 menit sedangkan untuk jumlah *epoch* 5000 hanya memerlukan waktu *train* selama 46 menit saja. Itu artinya untuk *epoch* 5000 memiliki kecepatan pelatihan sekitar 8 kali lebih cepat dari yang *epoch* 34000 namun memiliki persentase pengenalan yang tidak kalah bagusnya, yaitu sebesar 98.08 %.

err<	Lamanya
525	0:08:49.261
305	0:22:59.228
783	0:46:20.659
579	1:37:02.245
211	6:13:20.585
765	0:51:06.929
110	1:33:41.737

rate	Persentase Pengenalan
20	56.73 %
82	56.19 %
26	91.45 %
60	82.88 %
03	98.08 %
90	15.39 %
29	93.41 %
29	95.78 %
25	96.01 %
05	97.13 %
79	98.43 %
11	97.04 %
10	97.23 %

rata-rata untuk  
1 A. Pada tahap  
*back propagation*  
ng memuaskan.  
1 ke-2 memiliki

## PENUTUP

### Simpulan

1. *Back propagation neural network* dapat digunakan untuk mengenal nada suara alat musik gitar dan piano.
2. Semakin besar nilai *teachrate*, *momentum*, ataupun jumlah *epoch* maka *error rate* akan semakin kecil sedangkan untuk nilai *alpha*, *error rate* akan semakin besar.
3. Parameter yang optimal adalah parameter yang menghasilkan efisiensi, dalam hal lamanya proses training dan efisiensi dalam pengenalan nada.
4. Parameter pembelajaran yang paling optimal adalah dengan nilai *alpha*=0.1, *teachrate*=0.9, *momentum*=0.9, *epochcount*=5000, dan satu lapisan dalam dengan jumlah neuron sebanyak 100 buah.
5. *Error rate* yang semakin kecil belum tentu menghasilkan pengenalan yang semakin bagus. Semuanya itu tergantung pada *setting* yang tepat untuk parameter pelatihan, seperti *alpha*, *momentum*, dan *teachrate*, serta jumlah *input* untuk tiap nada.

### DAFTAR PUSTAKA

- Ari, Soekarno. 2002. *Buku Pintar Musik*. Jakarta: Inovasi.
- Blum, Adam. 1992. *Neural Networks In C++: an Object-Oriented Framework for Building Connection Systems*. John Wiley & Sons.
- Defatta, D.J., J.G. Lucas, and W.S. Hodgkiss. 1988. *Digital Signal Processing: A System Design Approach*. Canada: John Wiley & Sons inc.
- Fausett, Laurence. 1994. *Fundamental of Neural Networks: Architectures, Algorithms, and Applications*. Prentice hall.
- Kaye, A Solapung. 2002. *Gitar Tunggal: Metode Dasar Gitar Klasik*. Intermedia.
- Kosko, Bart. 1992. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs: Prentice hall International, Inc.
- \_\_\_\_\_. 1992. *Neural Networks for Signal Processing*. Englewood Cliffs: Prentice Hall International, Inc.
- Proakis John G. and Dimitris G. Manolakis. 1995. *Digital Signal Processing: Principles, Algorithms, and Application 3e*. New Jersey: Prentice Hall, Inc.
- Rao, V.B and H.V. Rao. 1993. *C++ Neural Networks and Fuzzy Logic*. New Yorks: Management Information Source, Inc,
- <http://www.ora.com/centers/gff/formats/micriff/index.htm>

<http://premium.microsoft.com/msdn/library/tools/dnmult/d1/newwave.htm>

<http://www.lightlink.com/tjweber/StripWav/WAVE.htm>

<http://www.technology.niagarac.on.ca/courses/comp630/WavFileFormat.html>

musik gitar

rate akan

ial lamanya

chrate=0.9,  
n sebanyak

akin bagus.  
perti *alpha*.

or *Building*

tem *Design*

ithms, and

pproach to

entice Hall

*Principles,*

management

004: 85-95

*Pengenalan Nada Suara Gitar dan Piano... (Wiedjaja; dkk.)*