

## **APLIKASI JARINGAN SYARAF TIRUAN DALAM MENDETEKSI TINGKAT KEBISINGAN RUANGAN DENGAN METODE LEARNING VECTOR QUANTIZATION**

Eza Rahmanita

*Jurusan Teknik Informatika  
Universitas Trunojoyo*

*Email : eza@trunojoyo.ac.id*

---

### **ABSTRAK**

Semakin majunya perkembangan teknologi dewasa ini membuat orang semakin berpikir maju pula sehingga timbullah keinginan bagaimana agar segala sesuatu dapat dikerjakan dengan mudah dan praktis

Sistem pengidentifikasi kebisingan suara telah dirancang dan disimulasikan dalam Tugas Akhir ini dengan memanfaatkan konsep Jaringan Syaraf Tiruan (JST). Sistem dibangun terdiri atas dua komponen utama yaitu bagian pemrosesan awal dan bagian pengklasifikasi tingkat kebisingan. Bagian pemrosesan awal membentuk pola dari suara atau sinyal di lingkungan sekitar dengan memanfaatkan metode Fast Fourier Transform (FFT) sebagai inti dari bagian pemrosesan awal tersebut. Bagian pengklasifikasi tingkat kebisingan berintikan jaringan syaraf tiruan.

Arsitektur JST yang digunakan adalah Learning Vector Quantization (LVQ). Penelitian memperlihatkan 3 level kebisingan suara dari suatu tempat yaitu tingkat rendah, sedang dan tinggi. Tingkat pengklasifikasian tersebut diharapkan mewakili seluruh tingkat kebisingan dengan menggunakan parameter JST yaitu learning rate = 0,03 dengan iterasi (epoch) = 1000

**Kata kunci :** *Jaringan Syaraf Tiruan, Learning Vector Quantization, FFT*

---

### **I. PENDAHULUAN**

Perkembangan ilmu pengetahuan dan teknologi yang ditandai dengan berkembangnya teknologi komputer yang semakin pesat, membuat komputer menjadi salah satu alternatif alat bantu yang sangat diandalkan manusia untuk menyelesaikan pekerjaan. Komputer mengerti tugas yang harus dilaksanakan berdasarkan urutan sekumpulan instruksi yang diberikan oleh pemakai. Seiring dengan perkembangan teknologi secara umum, maka muncul permasalahan yang sulit untuk dipecahkan bila dilakukan berdasarkan urutan sekelompok instruksi yang diberikan, misalkan: dalam bidang kedokteran, robotik, pengenalan pola dan sebagainya.

Manusia mulai berusaha untuk mengembangkan dan membuat komputer lebih cerdas, yang mampu berfikir sendiri. Serta melihat sifat-sifat fisiologis dari otak manusia maka dikembangkan suatu metode baru yang disebut Jaringan Syaraf Tiruan (*Artificial Neural Network*), yaitu suatu metode pengolahan informasi yang mampu untuk meniru cara kerja otak manusia dalam berfikir guna menyelesaikan suatu permasalahan. Komputer dibuat agar dapat mengerti tugas yang dijalankan dengan metode belajar (*learning*) sehingga tidak lagi berdasarkan urutan kumpulan instruksi yang diberikan oleh pemakai.

Banyak aplikasi Jaringan Syaraf Tiruan diterapkan diberbagai bidang. Salah satunya digunakan untuk mendeteksi tingkat kebisingan. Misalnya di ruang makan dan di ruang mesin pada suatu perusahaan kebisingannya sangat berbeda. Jika seseorang menginformasikan (memanggil, memberitahukan, dll) pegawai dalam ruangan-ruangan tersebut tidak mungkin memberikan tingkat volume speaker yang sama, karena jika dilakukan mungkin terjadi ketidaksesuaian antara volume speaker dengan tingkat kebisingan ruangan tersebut. Oleh karena itu diperlukan suatu aplikasi yang dapat mendeteksi tingkat kebisingan agar volume speaker sesuai dengan ruangan.

Untuk mengatasi masalah tersebut banyak metode dari Jaringan Syaraf Tiruan yang sering digunakan seperti: *Backpropagation*, *Kohonen*, *Maxnet*, *Mexican Hat* dan masih banyak lagi yang lain. Metode yang akan diterapkan dalam kasus ini adalah *Learning Vector Quantization (LVQ)* yaitu untuk mengolah suatu input dari *Digital Signal Processing (DSP)* untuk diolah oleh metode LVQ sehingga menjadi suatu ukuran kebisingan. LVQ mempunyai keistimewaan-keistimewaan diantaranya pada sistemnya yang relatif lebih kecil dibandingkan dengan metode lain (*Backpropagation*, *Multilayer Perceptreon*) sehingga prosesnya relatif lebih cepat.

Dalam tugas akhir ini akan dirancang suatu aplikasi yang mampu mendeteksi tingkat kebisingan menggunakan metode LVQ.

## 2. JARINGAN SYARAF TIRUAN

Awal Jaringan Syaraf Tiruan mulai dikembangkan pada tahun 1943 oleh Warren Mc Culloch dan Walter Piths yang bertujuan untuk pemodelan mesin-mesin yang mampu untuk memperbaiki kesalahan. Setelah itu mengalami masa reses untuk sementara dan pada suatu awal 90 dimulai penyelidikan yang lebih bersifat spesifik. Spesifik yang artinya memiliki konsep dasar yang mirip dengan otak dalam sel syaraf manusia.

Sampai saat ini otak manusia merupakan salah satu benda yang sangat komplikatif dan dalam mempelajari otak ini masih banyak hal-hal yang belum diketahui. Meskipun demikian dari hasil riset yang ada telah didapatkan pengertian mengenai operasi dasar otak pada tingkat rendah. Dia menjalankan fungsinya otak memiliki 10 milyar unit dasar yang biasa disebut *neuron*. Tiap-tiap neuron dihubungkan dengan 10.000 *neuron* yang lain. *Neuron* itu sendiri merupakan suatu unit dasar dari otak dan sistem analog yang berdiri sendiri. Ada dua macam *neuron* dalam manusia, tipe pertama ialah proses lokal antara *neuron* itu sendiri, yang kedua *neuron* yang menghubungkan otak dengan otot mata atau sensor lain serta menghubungkan dengan sensor otak.

Cara kerja *neuron* ini belum sepenuhnya diketahui tetapi pada dasarnya *neuron* ini menerima semua input yang datang untuk dijumlahkan. Jika pada suatu saat cukup banyak input yang diterima maka *neuron* tersebut akan aktif, sebaliknya jika belum cukup banyak maka *neuron* tidak akan aktif.

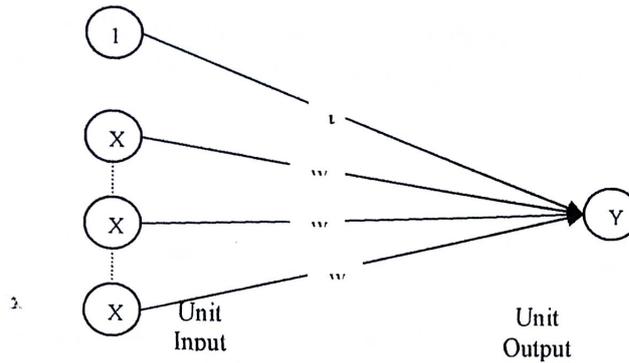
*Neuron* selaku komponen utama dari otak manusia memiliki kemampuan dasar;

- Mampu bereaksi terhadap rangsangan.
- Mampu mempropagasi sinyal aktifasi yang dihasilkannya.
- Mampu merangsang sel yang lain.
- Badan sel mengandung *nukleus* yang merupakan inti dari sel syaraf.
- Benang-benang perpanjangan sel yang terdiri dari satu akson dan beberapa *dendrit*.

Pemodelan jaringan syaraf tiruan secara matematis didasari pada model biologis dari neuron. JST terkadang disebut juga *artificial neuron*, meski lebih sering disebut dengan *nodes* atau *unit*. Sistem JST telah banyak diaplikasikan dalam berbagai bidang teknologi, termasuk dalam mendeteksi tingkat kebisingan ruangan. Keunggulan dari pengimplementasian sistem JST ini adalah karena :

- JST secara teoritis mengimplementasikan sistem komputasi secara paralel

- JST mempunyai sifat *adaptif*
- JST memiliki struktur yang kokoh dan tingkat toleransi yang tinggi
- JST memiliki efisien yang tinggi terutama untuk implementasinya dalam sistem non-linier



Gambar 1. Arsitektur jaringan syaraf tiruan

Setiap *input*  $x$  dimodelkan anak panah serta masing-masing mempunyai sebuah kuantitas yang disimbolkan dengan  $w$ , yang biasa dinamakan bobot (*weight*). Setiap elemen pemroses memiliki banyak masukan, namun hanya memiliki output tunggal. Setiap unit menghasilkan nilai *net-input* yang berdasarkan dengan harga dari semua koneksi masukannya. Dengan tidak memperhitungkan adanya koneksi khusus, harga *net-input* dapat dihitung secara sederhana sebagai hasil penjumlahan dari semua harga masukan yang dikalikan dengan harga bobot yang bersesuaian. Hal ini dapat ditulis dengan persamaan sebagai berikut ;

$$net_i = \sum_i w_i x_i$$

Formulasi vektor merupakan hal yang sangat berguna dan membantu menggambarkan pada beberapa tipe arsitektur JST. *Input Vector*, *Output Vector* dan *Weight Vector* merupakan istilah-istilah yang sering dipergunakan.

### 2.1 Struktur Dasar Kohonen

Struktur utama JST *Kohonen* terdiri dari dua layer, layer pertama merupakan layer masukan, sedangkan layer kedua adalah layer kompetitif yang biasanya diatur dalam bentuk dua dimensi. Pola interkoneksi yang ada merupakan interkoneksi penuh, dimana setiap elemen pemroses dari layer masukan berhubungan dengan setiap elemen pemroses dari layer kompetitif.

Layer masukan akan meneruskan sinyal masukan yang diterima setiap unit masukan. Unit masukan akan mengambil harga yang berkorespondensi dengan pola sinyal masukan dan akan meneruskannya ke setiap unit kompetitif yang pada gilirannya melakukan penjumlahan dari setiap masukan dan bersaing dalam menentukan unit pemenang yang tunggal.

### 2.2 Learning Vector Quantization (LVQ)<sup>1</sup>

*Learning Vector Quantization* (LVQ) adalah JST untuk penggolongan pola dilatihkan dengan satu atau lebih variasi dalam *Kohonen Learning*.

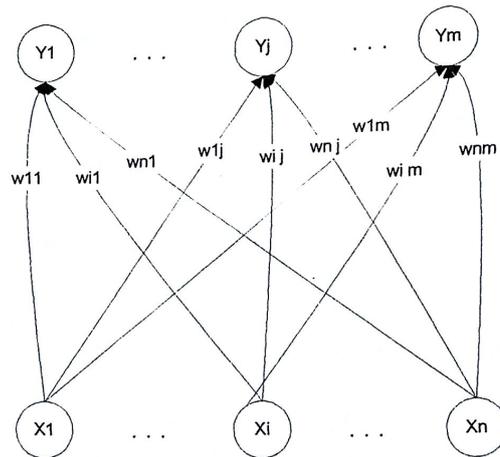
### 2.3 Algoritma Learning Vector Quantization

Aturan dasar proses belajar LVQ adalah sebagai berikut :

- Hanya neuron pemenang yang diperbaharui oleh vektor referensinya.

<sup>1</sup> Teuvo Kohonen, January 1996, "*LVQ\_PAK: The Learning Vector Quantization Program Package*", Helsinki University of Technology

- Pembaharuan vektor referensi dilakukan baik terhadap hasil klasifikasi yang salah maupun yang benar.
  - Proses pembaharuan (*updating*) sesuai dengan yang digunakan pada proses identifikasi.
- Ada beberapa macam algoritma LVQ yang telah dikembangkan, beberapa akan dijelaskan dalam makalah ini akan tetapi dalam tugas akhir ini yang digunakan adalah LVQ1 mengingat kesederhanaan dan kemudahan dalam menggunakannya.



Gambar 2. Arsitektur metode Learning Vector Quantization

### 2.3.1 LVQ1 [Kohonen 1990b]

Konsep utama algoritma ini adalah :

“Apabila vektor masukan dan neuron referensi diklasifikasikan dalam kelas yang sama, maka neuron referensi akan diperbarui dengan melakukan pergeseran posisinya mendekati vektor masukan. Sebaliknya, jika vektor masukan dan neuron referensi tidak diklasifikasikan dalam kelas yang sama, maka neuron referensi akan diperbarui dengan pergeseran posisinya menjauhi vektor masukan.”

Tujuannya adalah untuk menemukan unit keluaran yang paling dekat jaraknya dengan vektor masukan. Proses LVQ1 dimulai dengan proses inialisasi parameter-parameter bersifat global dan inialisasi neuron yang merupakan otak dari JST. Inialisasi matriks *neuron* pada dasarnya merupakan proses pengambilan informasi mengenai unit pemenang dari matrik *neuron*.

$$c = \min \|x - y_c\|$$

Definisikan pendekatan  $y_i$  ke  $x$  dinotasikan dengan  $y_c$ . Tahap berikutnya adalah perhitungan yang dipilih berdasarkan kondisi-kondisi tertentu. Awalnya diisyaratkan kondisi dimana posisi vektor masukan neuron referensi baru pada lokasi neuron masukan yang tadi.

Pemeriksaan syarat kondisi kedua dilakukan dengan membandingkan kelas yang direpresentasikan oleh vektor masukan dan oleh *neuron* referensi. Jika tidak sama, maka akan dilakukan tiga hal berikut ini :

1. Posisi *neuron* referensi digeser menjauhi posisi vektor masukan.
2. Besarnya jari-jari keterpengaruhannya dikurangi dengan perhitungan tertentu.
3. Dibentuk *neuron* referensi yang baru pada lokasi vektor masukan dengan kelas yang direpresentasikannya sama dengan informasi kelas yang dibawa oleh vektor masukan.

Jika kelas yang direpresentasikan oleh keduanya sama, maka akan dilakukan pemeriksaan untuk kondisi terakhir. Pemeriksaan ini dilakukan dengan melihat apakah jarak vektor masukan terhadap vektor referensi sama dengan nol atau tidak.

Adapun pernyataan apabila ditulis dalam bentuk algoritma adalah sebagai berikut:

$$y_c(t+1) = y_c(t) + \alpha(t)[x(t) - y_c(t)]$$

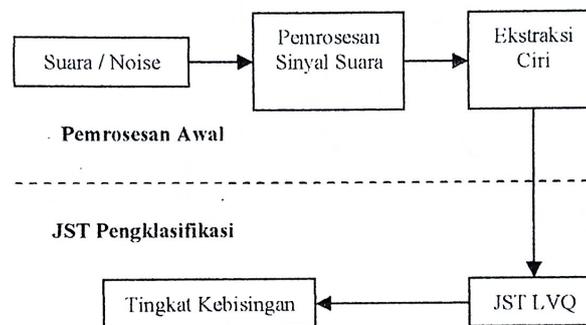
$$y_c(t+1) = y_c(t) - \alpha(t)[x(t) - y_c(t)]$$

$$y_i(t+1) = y_i(t)$$

Dalam menggunakan algoritma LVQ ini *learning rate* yang disarankan kurang dari 0,1

### 3. PERANCANGAN SISTEM

Pada tugas akhir kali ini, sistem yang akan diimplementasikan untuk mendeteksi suara (kebisingan) yang kemudian mengkonversi sinyal tersebut menjadi suatu ukuran kebisingan. Secara sederhana blok diagram dapat dilihat dalam gambar 3.1 yang merupakan sistem implementasi dalam tugas akhir ini.



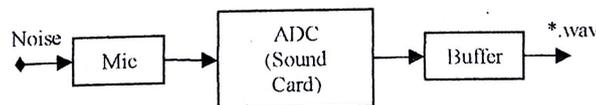
Gambar 3. Arsitektur sistem yang akan digunakan

Pada gambar diatas terlihat bahwa sistem yang dipakai untuk merealisasikan sangat kompleks. Secara singkat sistem ini terdiri dari 3 bagian dimana tiap-tiap bagian mempunyai pola maupun karakteristik yang berbeda-beda yaitu antara lain :

1. Analog Digital Converter (ADC).
2. Digital Signal Processing (DSP)
3. Neural Network, dalam hal ini diwakili dengan LVQ

#### 3.1 Analog Digital Converter

Pada bagian ini, sinyal suara yang diinputkan melalui *microphone* yang berfungsi untuk mengubah sinyal suara analog menjadi sinyal kontinu berupa tegangan analog, hal ini terlihat seperti pada gambar 4.



Gambar 4. Diagram sinyal suara menjadi sinyal digital

Setelah sinyal analog terbentuk berupa tegangan, maka dilakukanlah proses konversi yaitu mengubah sinyal analog menjadi sinyal digital yang berupa diskrit. Dalam proses ADC hal yang perlu diperhatikan adalah mengenai frekuensi sampling karena jika salah melakukan proses pencuplikan dengan sembarang

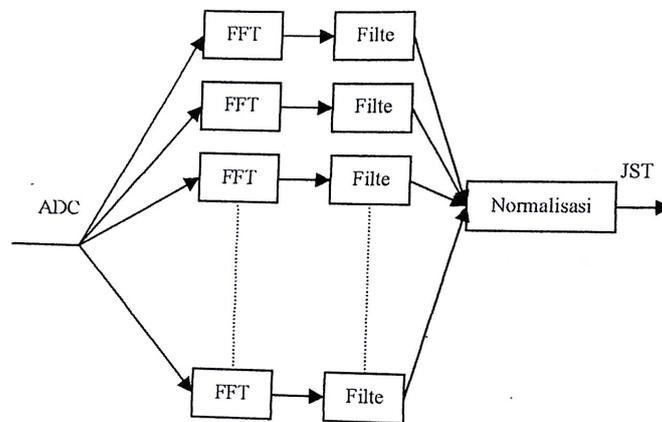
frekuensi maka akan terjadi *Aliasing* (kehilangan data dalam pencuplikan). Dalam hal ini pencuplikan ditetapkan sebesar 20ms sebab dianggap *stationer*<sup>2</sup>.

Pada gambar terdapat blok diagram buffer data, bagian ini berfungsi sebagai penampung data berupa sinyal suara untuk menghindari kehilangan data akibat kecepatan pencuplikan data dengan peralatan untuk mengambil data tidak sama. Pada umumnya peralatan ADC mempunyai kecepatan lebih rendah dibandingkan dengan peralatan pengambil data, untuk itu diperlukan media komunikasi antara kedua peralatan tersebut, salah satunya dengan memberi sinyal *acknowledgement*<sup>3</sup>, jadi jika akan mengambil data digital berupa sinyal suara terlebih dahulu mengirim sinyal ACK. Guna untuk menanyakan apakah data sudah siap diambil.

### 3.2 Digital Signal Processing (DSP)

Pada blok DSP merupakan proses yang sangat penting untuk menentukan keberhasilan dan keefektifan suatu JST dalam mengenali kebisingan. Hasil dari DSP inilah yang digunakan sebagai masukan sistem JST.

Berdasarkan hal diatas, maka dalam tugas akhir ini akan memanfaatkan analisa sinyal yaitu *Fast Fourier Transform* sebagai pengestraksi ciri dari sinyal suara.



Gambar 5. Metode *Mel-scaled Ceptral Coefficients* (Ekstraksi ciri)

Dengan metode *Mel-scale Ceptral Coefficients* (MSCC)<sup>4</sup>, FFT dapat dimodifikasi sedemikian hingga dibagi menjadi beberapa bagian yang pada akhirnya mendapatkan sampel sebanyak 13230 didapat dari frekuensi sampling 11025Hz selama 1,2dt. Proses ini sekaligus mengubah sinyal suara berbasis *Time Domain* menjadi sinyal suara yang berbasis *Frequency Domain* tujuannya adalah memudahkan dalam menganalisa. Dalam tugas akhir ini, digunakan FFT dengan pertimbangan kecepatan dan kemudahan dalam realisasinya.

Oleh karena keluaran dari DSP mempunyai nilai terlalu besar maka dilakukan normalisasi dengan menggunakan *Theorema Gaussian*<sup>5</sup> dengan persamaan berikut;

$$g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

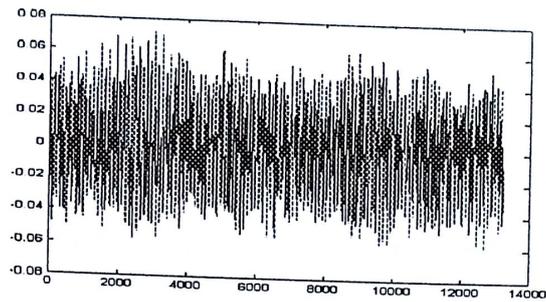
<sup>2</sup> Ville Makinen, "Front-end Feature Extraxtion with Mel-Scaled Ceptral Coefficients", Laboratory of Computational Engineering Helsinki University of Technology

<sup>3</sup> Yoyok Suryoko, "Pengenalan Suara Bawah Air Menggunakan Metode Modifikasi Jaringan Syaraf Tiruan Kohonen Fuature Map", Jurusan Teknik Elektro, Fakultas Teknik Industri, ITS

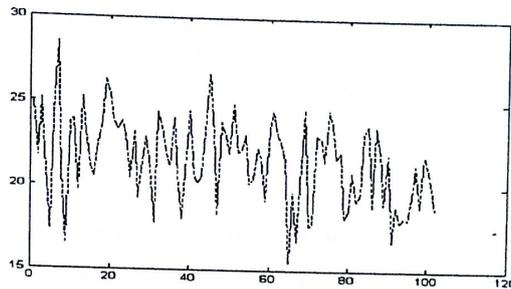
<sup>4</sup> Ville Makinen, "Front-end Feature Extraxtion with Mel-Scaled Ceptral Coefficients", Laboratory of Computational Engineering Helsinki University of Technology

<sup>5</sup> Boaz Porat, "A Course in Digital Signal Processing", John Wiley & Sons, Inc

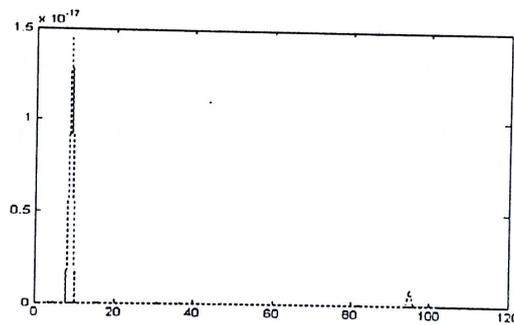
Dengan persamaan diatas diharapkan hasil dari poses DSP menjadi  $0 \leq g(x) \leq 1$  sehingga nilai tersebut siap untuk diproses dalam JST.



(a)



(b)



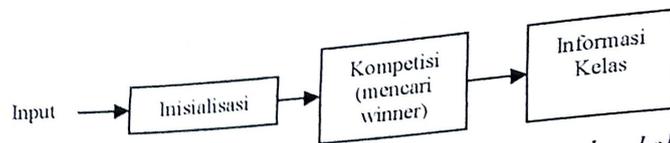
(c)

Gambar 6. (a) Sinyal suara sampel, (b) Sinyal setelah proses Msc (FFT), (c) Sinyal setelah normalisasi (Gaussian)

### 3.3 Neural Network (Learning Vector Quantization)

Seperti yang telah dijelaskan bahwa *Learning Vector Quantization* adalah sebuah metode klasifikasi dimana masing-masing output digambarkan pada sebuah kelas dan kategori.

Pada bagian ini *Neural Network* berfungsi sebagai pengukur kebisingan. Arsitektur yang digunakan dalam tugas akhir ini, *Neural Network* dengan dua layer yaitu *input layer* dan *output layer*. Sebelum digunakan, LVQ mengalami proses yang dinamakan *learning* (belajar). Proses belajar yang dilakukan hampir sama dengan metode-metode yang lain yang tujuan akhirnya JST dapat mengenali kebisingan yang telah dilatihkan (sifat adaptif). Proses belajar pada LVQ biasa disebut dengan *supervised* karena membutuhkan suatu output (target). Setelah dilatihkan JST dapat diaplikasikan untuk mendeteksi kebisingan dengan jalan menggunakan vektor referensi (bobot) hasil dari pembelajaran.



Gambar 7. Diagram blok JST LVQ pada pengenalan kebisingan

Pada tahap aplikasi (simulasi) JST hanya menerima input yang kemudian di olah sehingga menghasilkan output, hal ini biasa disebut *unsupervised* oleh karena JST tidak lagi membutuhkan target. Hasil dari simulasi inilah yang diperlukan untuk membandingkan dengan target yang telah dilatihkan sehingga diketahui *output* tersebut masuk dalam tingkat kebisingan yang telah dilatihkan.

#### 4. HASIL PERCOBAAN

Percobaan dilakukan secara *offline*, sehingga data yang telah dilatihkan dipergunakan kembali sebagai simulasi pengenalan. Tidak menutup kemungkinan program ini dipergunakan sebagai *online (realtime)* apabila didapat data yang sesuai keadaan sekitar dengan melakukan pengambilan data yang selanjutnya melakukan pembelajaran (*training*).

##### 4.1 Hardware dan Software

Dalam percobaan ini digunakan seperangkat komputer dengan spesifikasi; Processor AMD K6III-500Mhz, Memory 128MB, Sound Card FM801 (Forte Media) Yamaha, Microphone dengan redaman 15 dB

Dalam pengambilan sampel digunakan tape *recorder* kemudian ditransfer kedalam komputer menggunakan sebuah perangkat lunak yakni *Sound Forge 5.0* dengan frekuensi 11025Hz selama 1,2dt sehingga di dapat sampel sebanyak 13230 dalam bentuk *sequence* data. Sedangkan pemrograman yang digunakan dalam membuat aplikasi tugas akhir adalah *Mathlab 6.0* dimana sebagian fungsi-fungsi yang digunakan dalam tugas akhir ini telah tersedia meskipun sebagian harus disesuaikan dengan arsitektur jaringan.

##### 4.2 Bagian JST LVQ

Bagian ini menerima hasil analisa dari *Fast Fouier Transform* dan melakukan klasifikasi terhadapnya. Dalam proses pembelajaran terdapat 30 sampel yang terdiri dari 10 sampel untuk kebisingan tingkat rendah, 10 untuk tingkat sedang dan 10 untuk tingkat tinggi yang dilatihkan dalam suatu proses selama 1000 epoch dengan *learning rate* yang berbeda.

##### 4.3.1 Perubahan Jumlah Parameter Weight dan Learning Rate

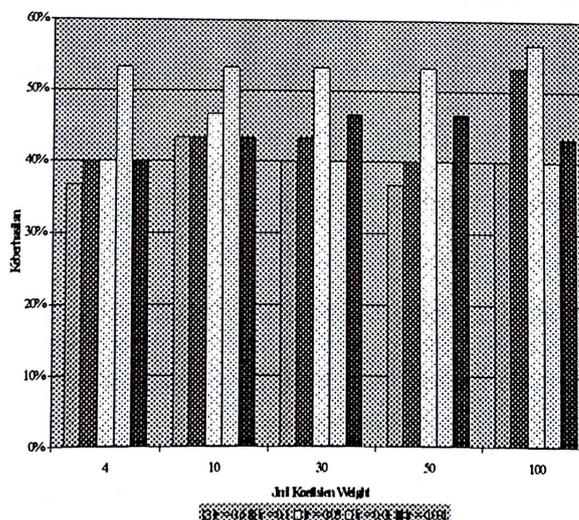
*Leaning rate*<sup>6</sup> adalah parameter yang mengontrol perubahan *weight* dalam proses *training*. Berdasarkan hal tersebut, secara umum laju pengenalan akan berubah sesuai dengan besar kecilnya *leaning rate*. *Learning rate* yang dianjurkan kurang dari 0.1<sup>7</sup>, maka akan dicoba dengan berbagai varian *learning rate* dan jumlah parameter *weight* sehingga diketahui berapa besar unjuk kerja pengenalan JST. Dalam percobaan akan dicoba beberapa parameter *leaning rate* yaitu 0,6; 0,1; 0,01; 0,05 dan 0,001 serta jumlah *weight* 4, 10, 30, 50 dan 100.

Tabel 1. Unjuk kerja pengenalan kebisingan dengan Perubahan jumlah kefisien weight dan Learning rate

<sup>6</sup> Fausett, Laurene, "Fundamental Of Neural Network", Prantise Hall International, Inc

<sup>7</sup> Teuvo Kohonen, January 1996, "LVQ\_PAK: The Learning Vector Quantization Program Package", Helsinki University of Technology

Jumlah Koefisien Weight	Parameter learning rate (k)	Keberhasilan (%)
4	0.6	37%
	0.1	40%
	0.05	40%
	0.01	53%
	0.001	40%
10	0.6	43%
	0.1	43%
	0.05	47%
	0.01	53%
	0.001	43%
30	0.6	40%
	0.1	43%
	0.05	53%
	0.01	40%
	0.001	47%
50	0.6	37%
	0.1	40%
	0.05	53%
	0.01	40%
	0.001	47%
100	0.6	40%
	0.1	53%
	0.05	57%
	0.01	40%
	0.001	43%



Gambar 8. Unjuk kerja pengenalan kebisingan dengan Perubahan jumlah koefisien weight dan learning rate dalam bentuk bar chart

Dengan jumlah epoch 1000, proses *learning* membutuhkan waktu rata-rata selama 8 menit 7 detik dengan menggunakan pada sebuah CPU AMD K6-III 500Mhz, Memory 128Mb.

## 5. KESIMPULAN

Dari analisis yang telah dilakukan atas data hasil penelitian, maka pada akhirnya penulis mencoba untuk menarik kesimpulan secara keseluruhan atas penelitian tugas akhir yang telah dilakukan sebagai berikut :

1. Sistem pengenalan kebisingan dengan harga-harga parameter, inisial leaning rate  $\alpha_0 = 0,03$  dan jumlah epoch = 1000 menghasilkan kemampuan klasifikasi hingga 83.3% untuk suara kebisingan (kriteria pertama) dan 90% untuk suara kebisingan (kriteria kedua) yang dilatihkan.
2. Kestabilan sistem ternyata sangat dipengaruhi oleh jumlah *weight* yang disediakan dalam JST. Semakin banyak *weight* yang disediakan oleh JST maka semakin stabil pula sistemnya.
3. Aplikasi dalam tugas akhir ini dapat dijalankan secara *real time* dengan catatan :
  - ✓ Waktu dan tempat pengambilan sample untuk pembelajaran harus sama dengan waktu dan tempat aplikasi dijalankan.
  - ✓ Sample yang telah dilatihkan hanya berlaku ditempat pengambilan sample tersebut dan tidak untuk tempat lain, apabila dilakukan ditempat lain maka tidak menutup kemungkinan program kurang sesuai dalam mengeluarkan *output* (tingkat kebisingan) bahkan dapat juga kondisi tidak dikenal.

## 6. REFERENSI

- Boaz Porat, "A Course in Digital Signal Processing", John Wiley & Sons, Inc
- Dayhoff, Judith E, "Neural Network Architectures An Introduction", Van Nostrand Reinhold, New York.
- David J. DeFatta, Joseph G. Lucas, William S. Hodgkiss, "Digital Signal Processing : A System Design Approach", John Wiley & Sons, Inc
- Edward W. Kemen, Bonnie S. Heck, "Fundamental Of Signal and Systems Using Matlab", Prantise Hall Upper Saddle River, New Jersey 07458
- Fausett, Laurene, "Fundamental Of Neural Network", Prantise Hall International, Inc.
- Gillian Ewers, "A Simple Approach to Digital Signal Processing", Wiley Interscience and Texas Instruments.
- Howard Demuth, Mark Beale, "Neural Network Toolbox for Use with Matlab", The MathWork, Inc
- John G. Proakis, Dimitris G. Monolakis, "Digital Signal Processing; Pinciples, Algorithms, and Application 3e", Prantise Hall International, Inc
- Magnus Norgaard, "Neural Network Based System Identification Toolbox 1.1 for Use with Matlab", Technical Unversity of Denmark
- Mathlab, "Creating Graphical User Interfaces", The MathWork Inc
- Nicolaos B. Karayiannis, "Soft Learning Vector Quantization and Clustering Algorithms Based on Ordered Weighted Aggregation Operator" in IEEE Transaction on Neural Network, Vol 11, no 5, pp 1093-1105, Sept 2000
- Teuvo Kohonen, Jussi Hynninen, Jari Kangas, Jorma Laaksonen and Tari Tarkkola, "LVQ\_PAK: The Learning Vector Quantization Program Package", Helsinki University of Technology, Faculty of Information Technology, Finland
- Ville Makinen, "Front-end Feature Extraxtion with Mel-Scaled Ceptral Coefficients", Laboratory of Computational Engineering Helsinki University of Technology
- Yoyok Suryoko, "Pengenalan Suara Bawah Air Menggunakan Metode Modifikasi Jaringan Syaraf Tiruan Kohonen Fuature Map", Jurusan Teknik Elektro, Fakultas Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya