

Klasifikasi Suara Manusia Dengan Menggunakan Jaringan Syaraf Tiruan Dan Algoritma Levenberg-Marquadt

Gunawan Abdillah

Dosen tetap jurusan informatika F-MIPA UNJANI

Abstrak. Teknologi pengenalan suara saat ini telah mengalami perkembangan terutama dalam hal *speech processing*. *Speech processing* merupakan suatu cara untuk mengekstrak informasi yang diinginkan dari sebuah sinyal suara. Penelitian ini membahas sistem klasifikasi suara manusia *male* dan *female*. Mengekstrak ciri dari sinyal suara setiap frame pada domain waktu dan frequency domain sangat membantu untuk menyederhanakan dan mempercepat perhitungan. Adapun fitur-fitur untuk suara atau audio antara lain *Short Time Energy*, *Zero Crossing Rate*, *Spectral Centroid* dan lain-lain. Hasil pengujian sistem menunjukkan bahwa klasifikasi suara manusia dengan menggunakan jaringan syaraf tiruan *backpropagation* dan algoritma *Levenberg-Marquadt* untuk perubahan matriks bobotnya sangat baik dan cepat karena kompleksitas perhitungan yang tidak terlalu tinggi. Database sample suara sebanyak 40 buah dengan data test sebanyak 4 suara. Output dari sistem adalah hasil klasifikasi yang telah dikenali dengan nilai kemiripan 0,5 sebagai *male* dan $< 0,5$ sebagai *female*.

Kata Kunci : *Feature Extraction*, Klasifikasi, *Backpropagation*, Algoritma *Levenberg Marquadt*

Latar Belakang

Teknologi pengenalan suara saat ini telah mengalami perkembangan yang cukup pesat. Banyak aplikasi-aplikasi yang telah dihasilkan seiring dengan metode dan proses dari pengenalan suara tersebut. Dari perkembangan teknologi pengolahan sinyal suara ini didapatkan ide untuk membuat aplikasi yang dapat digunakan untuk membedakan atau mengklasifikasikan suara manusia.

Clustering merupakan salah satu dari aplikasi proses *digital signal processing* (pengolahan sinyal digital) termasuk didalamnya *Speech Recognition*. Pada penelitian ini, dibuat sebuah program yang dapat meng-*cluster* sejumlah sample suara dengan fitur-fitur suara yang direpresentasikan secara numerik. Implementasi sistem klasifikasi suara manusia ini menggunakan MATLAB. Pada MATLAB tersedia toolbox-toolbox yang berguna untuk membuat aplikasi kluster suara manusia, yaitu *Audio Signal Processing* dan *Neural Net*.

Data *training* diperoleh dari file-file suara statis berformat wave. Data *training* tersebut akan diekstrak cirinya untuk didapatkan *feature*-nya. *Feature* inilah yang akan digunakan untuk klasifikasi suara manusia berdasarkan gender pria (*male*) dan wanita (*female*). Untuk *training* data suara hasil ekstraksi *feature* digunakan jaringan syaraf tiruan *backpropagation*. Adapun perubahan bobotnya menggunakan algoritma *Levenberg-Marquadt*.

Tinjauan Pustaka

1 *Speech Processing*

Suara adalah suatu sinyal yang sangat dipengaruhi oleh frekuensi dan merupakan bentuk sinyal diskrit yang sangat dipengaruhi oleh waktu. *Speech processing* mengekstrak informasi-

yang diinginkan dari sebuah sinyal suara. Untuk memproses sebuah sinyal dengan sebuah komputer digital, sinyal harus dihadirkan dalam bentuk digital sehingga sinyal tersebut dapat digunakan oleh sebuah komputer digital.

Awalnya, gelombang suara akustik diubah ke sebuah sinyal digital sesuai untuk *voice processing*. Sebuah microphone atau telephone handset dapat digunakan untuk merubah gelombang akustik ke dalam sebuah sinyal analog. Sinyal analog ini dikondisikan dengan *antialiasing filtering*. *Antialiasing filter* membatasi *bandwidth* sinyal menjadi kira-kira *Nyquist rate* sebelum sampling. Sinyal analog terkondisikan kemudian diubah ke dalam bentuk sebuah sinyal digital oleh sebuah *analog-to-digital (A/D) converter*.

2 FFT (Fast Fourier Transform)

Transformasi *fourier* adalah suatu metode yang sangat efisien untuk menyelesaikan transformasi *fourier* diskrit yang banyak dipakai untuk keperluan analisa sinyal seperti pemfilteran, analisa korelasi, dan analisa spectrum. *Diskrit Fourier Transformasi (DFT)* adalah deretan yang terdefinisi pada kawasan frekuensi diskrit yang merepresentasikan Transformasi *Fourier* terhadap suatu deretan terhingga (*finite duration sequence*). *DFT* berperan penting untuk implementasi algoritma suatu varitas pengolahan sinyal, karena efisien untuk komputasi berbagai aplikasi.

Fast fourier Transformation atau transformasi *Fourier* cepat, merupakan proses lanjutan dari *DFT (Diskrit Fourier Transformation)*. Transformasi *Fourier* ini dilakukan untuk mentransformasikan sinyal dari domain waktu ke domain frekuensi. Hal ini bertujuan agar sinyal dapat diproses dalam spectral substraksi.

FFT adalah bentuk khusus dari persamaan integral *fourier* :

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \quad k = 0,1,2,\dots,N-1$$

Dalam proses *fast fourier transform* akan menghasilkan dua buah nilai yaitu nilai real dan nilai imajiner. Dan untuk proses selanjutnya dibutuhkan nilai real saja untuk itu diperlukan proses mutlak log.

3 Feature Extraction

Mengekstrak ciri dari sinyal suara setiap frame pada *time domain* dan *frequency domain* sangat membantu untuk menyederhanakan dan mempercepat perhitungan serta mempermudah interpretasi secara fisik. Ciri-ciri ini termasuk level standar deviasi dan rata-rata standar deviasi. Adapun fitur-fitur untuk suara atau audio, antara lain *Short Time Energy*, *Zero Crossing Rate*, *Spectral Centroid*, *Spectral Flux*, *Bandwith*, *Cepstral Coefficient*, *Delta Spectrum*, *Root Mean Square*, *High Feature Value Ratio*, *Low Feature Value Ratio*, *Spectral Spread*, *Spectral RollOff* dan lain-lain.

3.1 Spectral Centroid

Menyeimbangkan titik ukuran spectrum dari bentuk asosiasi spectral dengan *spectral brightness*. Nilai *centroid* yang tinggi menunjukkan frekuensi yang tinggi.

$$C = \frac{\sum_{n=1}^N M_t[n] \cdot n}{\sum_{n=1}^N M_t[n]}$$

3.2 Zero Crossing Rate

Zero crossing merupakan sample berurutan pada sebuah sinyal digital yang memiliki perbedaan tanda, ukuran dari noise sebuah sinyal pada fitur *time domain*.

$$ZCR = \frac{1}{N} \sum_{n=2}^N |\text{sign}(x(n)) - \text{sign}(x(n-1))|$$

$$\text{sign}(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$$

3.3 Short Time energy

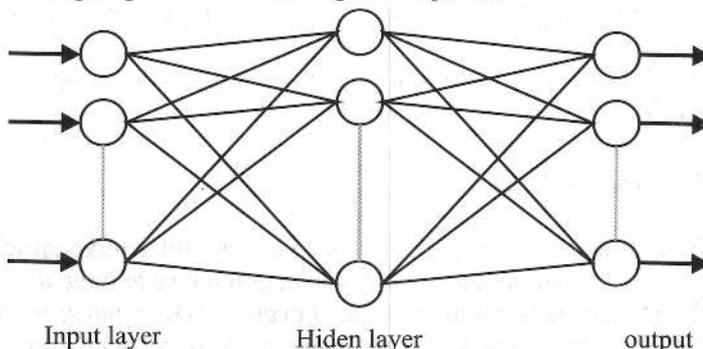
$$STE = \frac{1}{N} \sum_{n=1}^N x(n)^2$$

4 Backpropagation Neural Network

Backpropagation merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metode belajar terbimbing. Pada jaringan diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan pada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola.

Algoritma pelatihan jaringan saraf backpropagation terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan.

Jaringan backpropagation terdiri atas tiga lapisan atau lebih unit pengolahan. Gambar 1 menunjukkan jaringan backpropagation dengan tiga lapisan pengolahan, bagian kiri sebagai masukan, bagian tengah disebut sebagai lapisan tersembunyi dan bagian kanan disebut lapisan keluaran. Ketiga lapisan ini terhubung secara penuh.



Gambar 1. Struktur Jaringan Syaraf

Perambatan maju dimulai dengan memberikan pola masukan ke lapisan masukan. Pola masukan ini merupakan nilai aktivasi unit-unit masukan. Dengan melakukan perambatan maju dihitung nilai aktivasi pada unit-unit di lapisan berikutnya. Pada setiap lapisan, tiap unit pengolah melakukan penjumlahan berbobot dan menerapkan fungsi sigmoid untuk menghitung keluarannya.

Untuk menghitung nilai penjumlahan berbobot digunakan rumus :

$$S_j = \sum_{i=0}^n a_i \cdot w_{ji}$$

Dengan :

a_i = masukan yang berasal unit i

w_{ji} = bobot sambungan dari unit i ke unit j

Setelah nilai S_j dihitung, fungsi sigmoid diterapkan pada S_j untuk membentuk $f(S_j)$. Fungsi sigmoid ini mempunyai persamaan :

$$f(S_j) = \frac{1}{1 + e^{-S_j}}$$

Hasil perhitungan $f(S_j)$ ini merupakan nilai aktivasi pada unit pengeolah j . Nilai ini dikirimkan keseluruhan keluaran unit j . Setelah perambatan maju dikerjakan maka jaringan siap melakukan perambatan mundur.

Jaringan perambatan mundur dilatih dengan metode belajar terbimbing. Pada metode ini jaringan diberi sekumpulan pasangan pola yang terdiri dari pola masukan dan pola yang diinginkan (*target*). Pelatihan dilakukan berulang-ulang sehingga dihasilkan jaringan yang memberikan tanggapan yang benar terhadap semua masukannya.

Pembelajaran atau *learning process* merupakan sarana pelatihan untuk mendapatkan nilai bobot yang sesuai pada setiap node yang membentuk jaringan syaraf tiruan (ANN). Data akan dibagi menjadi 2 bagian yang pertama untuk digunakan sebagai proses pembelajaran yang disebut dengan data training dan yang bagian kedua untuk proses pengujian yang disebut data testing. Untuk kasus di atas pola pembelajaran dan pengujiannya dapat diuraikan sbb:

- ❖ Memasukan sejumlah data berupa nilai angka pada node input dan output
- ❖ Menggunakan algoritma pembelajaran back propagation dan algoritma Levenberg-Marquadt untuk melakukan update nilai bobot pada node di lapisan hiddennya
- ❖ Bila proses belajar sudah mencapai konvergen, nilai bobot tersebut akan disimpan dan untuk diujikan kembali untuk data yang sama
- ❖ Melakukan pengujian dengan menggunakan data yang berlainan dengan proses pembelajaran

5 Algoritma *Levenberg-Marquadt*

Algoritma *Levenberg-Marquadt* merupakan algoritma yang sudah terkenal sebagai algoritma optimasi. Algoritma ini didapatkan dari hasil turunan gradien sederhana dan metode gradien lainnya yang berhubungan pada suatu masalah. *Levenberg-Marquardt* merupakan metode alternatif yang populer selain metode *Gauss-Newton* untuk mencari minimum dari suatu fungsi $F(x)$ non linier.

$$F(x) = \frac{1}{2} \sum_{i=1}^m [f_i(x)]^2$$

Dimana $x = x_1, x_2, x_3, \dots$ yang merupakan vector dan f_i merupakan fungsi dari R^n ke R . f_i menunjukkan sisa (*residual*) dan diasumsikan m n .

Agar lebih sederhana, maka F direpresentasikan sebagai vector sisa (*residual vector*)

$$f: R^n \rightarrow R^m$$

$$f(x) = (f_1(x), f_2(x), f_3(x), \dots, f_n(x))$$

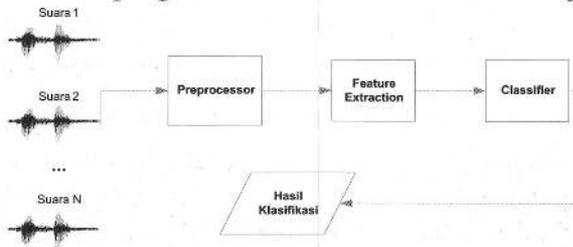
Algoritma *Levenberg-Marquadt* menggunakan matrix Jacobi. Dimisalkan f_i ditandai dengan $J_i(x)$ maka metode *Levenberg-Marquadt* akan mencari langkah yang tepat untuk memberikan solusi p dari persamaan (*equation*).

$$(J_k^T J_k + \lambda_k I) p_k = -J_k^T f_k$$

Di mana nilai scalar λ_k bukan negatif dan I merupakan matriks identitas.

Hasil Dan Pembahasan

Sistem kerja perangkat lunak pengklasifikasian suara manusia ditunjukkan pada gambar 2 berikut:

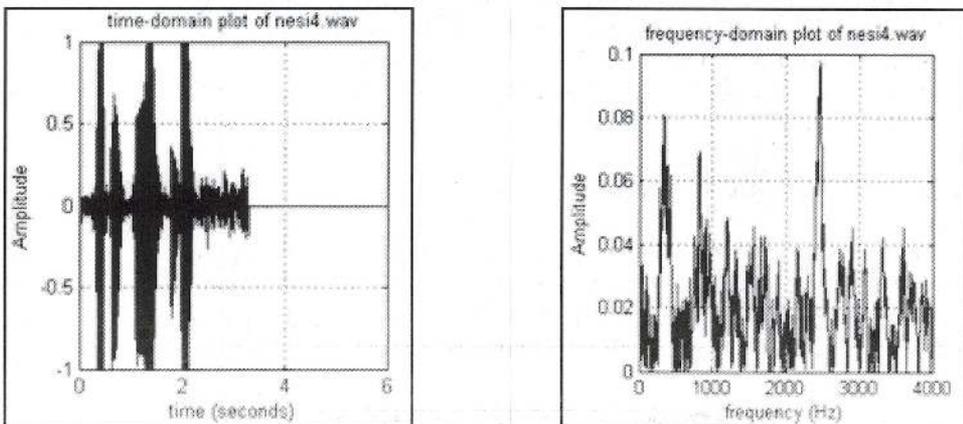


Gambar 2. Diagram Blok Klasifikasi Suara Manusia

Preprocessor

Pada bagian pre-processor, diperlukan perangkat-perangkat pendukung untuk melakukan pengolahan suara. Perangkat multimedia pendukung, yaitu sound card, speaker active dan microphone. Suara-suara yang diambil dari perangkat pendukung ini disimpan dalam format file wave. File berformat wave merupakan inputan bagi program ini.

Hasil dari preprocessor :



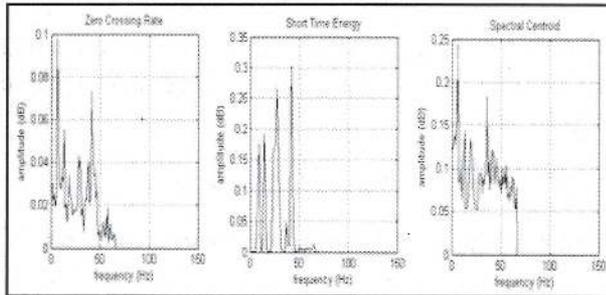
Gambar 3. Time Dan Frequency domain dari file nesi4.wav

Ekstraksi Ciri

Feature Extraction pada jurnal ini menggunakan 3 ciri, dengan dua fitur *Time Domain* dan satu fitur *Frequency Domain*. Adapun ciri atau fitur yang digunakan pada feature extraction ialah

- ❖ Fitur Time Domain: *Zero Crossing Rate* dan *Short Time Energy*
- ❖ Fitur Frequency Domain: *Spectral centroid*.

Adapun tampilan grafiknya dapat dilihat sebagai berikut:



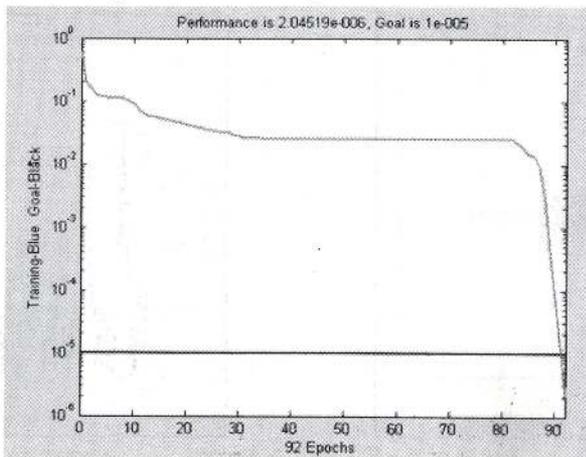
Gambar 4. Feature suara "nesi4.wav"

Ketiga ciri tersebut dihitung untuk semua sample suara yang dalam hal ini terdapat 40 sample suara. Masing-masing ciri didapatkan standar deviasi dan rata-rata sehingga membentuk ciri data suara 6 dimensi.

Percobaan dilakukan dengan melakukan *training* data untuk 40 sample suara yang diekstrak cirinya. Keseluruhan *feature* dari data *training*, dijadikan data input pada jaringan syaraf tiruan *backpropagation* dengan perubahan bobot matriksnya menggunakan Algoritma *Levenberg-Marquadt*. Jaringan syaraf tiruan *backpropagation* dirancang untuk 6 input, 6 unit layer tersembunyi (*hidden layer*), 1 unit output dan fungsi aktivasi *sigmoid*. Maksimum iterasi (*epoch*) yang dilakukan adalah 1000 kali dengan *learning rate* 0,1.

Jaringan syaraf tiruan ini ditentukan secara *supervised* (pelatihan terbimbing) untuk membedakan suara pria (*male*) dan suara wanita (*female*). Sebagai targetnya untuk *male* adalah 1 dan *female* adalah 0. klasifikasi output yang dihasilkan ialah untuk *male* 0,5 dan *female* < 0,5.

Hasil pelatihan mengalami konvergensi tercepat hanya melalui 92 *epoch*. Adapun grafik pada saat pelatihan jaringan konvergensi adalah sebagai berikut:



Gambar 5. Grafik Konvergensi terbaik

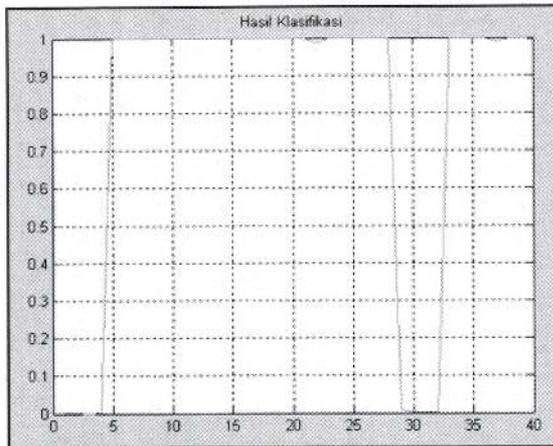
Sedangkan hasil pelatihan dengan 40 data sample suara adalah sebagai berikut:

NO	Nama File	std ZCR	mean ZCR	std STE	mean STE	std SC	mean SC	Hasil	Cluster	Keterangan
1	nesi1.wav	0,023730	0,028200	0,053839	0,030018	0,041820	0,109316	0,0000	Female	Berhasil
2	nesi2.wav	0,023089	0,027556	0,060161	0,034566	0,045273	0,106809	0,0001	Female	Berhasil
3	nesi3.wav	0,011613	0,022843	0,122127	0,084569	0,023433	0,087313	0,0000	Female	Berhasil
4	nesi4.wav	0,017669	0,023440	0,078219	0,048713	0,032818	0,092258	0,0018	Female	Berhasil
5	ednofri1.wav	0,033071	0,026682	0,046726	0,026043	0,041055	0,096020	1,0000	Male	Berhasil
6	ednofri2.wav	0,039708	0,030886	0,048695	0,027065	0,050672	0,105170	0,9986	Male	Berhasil
7	ednofri3.wav	0,026037	0,024439	0,047040	0,025282	0,041596	0,097089	1,0000	Male	Berhasil
8	ednofri4.wav	0,030507	0,024865	0,073055	0,036762	0,035571	0,094021	1,0000	Male	Berhasil
9	david1.wav	0,027513	0,021344	0,049431	0,027118	0,035595	0,085672	1,0000	Male	Berhasil
10	david2.wav	0,025937	0,025611	0,075767	0,036803	0,036567	0,098210	1,0000	Male	Berhasil
11	david3.wav	0,024548	0,025686	0,095128	0,042284	0,035063	0,103253	1,0000	Male	Berhasil
12	david4.wav	0,023269	0,025393	0,066233	0,035101	0,039709	0,098462	1,0000	Male	Berhasil
13	sofyan1.wav	0,015088	0,025246	0,130075	0,065757	0,035347	0,100256	0,9985	Male	Berhasil
14	sofyan2.wav	0,020387	0,021691	0,128442	0,064249	0,032297	0,094949	1,0000	Male	Berhasil
15	sofyan3.wav	0,031652	0,025949	0,117375	0,063370	0,039712	0,098799	1,0000	Male	Berhasil
16	sofyan4.wav	0,019919	0,022289	0,137326	0,068748	0,036137	0,100233	1,0000	Male	Berhasil
17	gunawan1.wav	0,048244	0,033842	0,056490	0,029438	0,052182	0,113318	0,9984	Male	Berhasil
18	gunawan2.wav	0,038587	0,025922	0,079396	0,045345	0,046280	0,093854	1,0000	Male	Berhasil
19	gunawan3.wav	0,038189	0,025810	0,072721	0,038659	0,046374	0,099303	1,0000	Male	Berhasil
20	gunawan4.wav	0,044965	0,029239	0,092387	0,045855	0,050673	0,101057	1,0000	Male	Berhasil
21	herlambang1.wav	0,021168	0,029672	0,079291	0,034514	0,034983	0,103224	0,9999	Male	Berhasil
22	herlambang2.wav	0,012413	0,023890	0,105226	0,052168	0,030433	0,098875	0,9874	Male	Berhasil
23	herlambang3.wav	0,012300	0,021648	0,065936	0,032585	0,033322	0,095829	1,0000	Male	Berhasil
24	herlambang4.wav	0,018107	0,026754	0,112719	0,054228	0,028632	0,102674	1,0000	Male	Berhasil
25	hendy1.wav	0,010022	0,022079	0,072746	0,031692	0,034424	0,103500	1,0000	Male	Berhasil
26	hendy2.wav	0,024552	0,022782	0,067975	0,028431	0,034017	0,098336	1,0000	Male	Berhasil
27	hendy3.wav	0,022379	0,022064	0,121424	0,060475	0,030819	0,091675	1,0000	Male	Berhasil
28	hendy4.wav	0,028329	0,028174	0,115847	0,050776	0,040408	0,107519	1,0000	Male	Berhasil
29	retno1.wav	0,028382	0,029657	0,086674	0,034091	0,038329	0,118073	0,0062	Female	Berhasil
30	retno2.wav	0,014153	0,027709	0,081679	0,043718	0,032235	0,115709	0,0000	Female	Berhasil
31	retno3.wav	0,011600	0,026482	0,086195	0,042591	0,029399	0,110554	0,0004	Female	Berhasil
32	retno4.wav	0,016262	0,031870	0,056717	0,026421	0,034404	0,122323	0,0000	Female	Berhasil
33	susapto1.wav	0,044199	0,026211	0,129537	0,079286	0,054906	0,097207	0,9999	Male	Berhasil
34	susapto2.wav	0,028959	0,026486	0,052396	0,028694	0,045678	0,103824	0,9993	Male	Berhasil
35	susapto3.wav	0,033152	0,025824	0,077810	0,037894	0,045402	0,105760	1,0000	Male	Berhasil
36	susapto4.wav	0,012231	0,021400	0,044695	0,021365	0,034944	0,096401	1,0000	Male	Berhasil
37	arif1.wav	0,018911	0,025345	0,044096	0,023708	0,037532	0,105399	0,9894	Male	Berhasil
38	arif2.wav	0,022112	0,024091	0,070131	0,041114	0,037771	0,098052	0,9999	Male	Berhasil
39	arif3.wav	0,020303	0,024131	0,047895	0,024834	0,033079	0,098503	1,0000	Male	Berhasil
40	arif4.wav	0,019052	0,023597	0,066474	0,038902	0,031472	0,092382	0,9943	Male	Berhasil

Keterangan:
ZCR = Zero Crossing Rate

STE = Short Time Energy
SC = Spectral Centroid

Jaringan mengenali 100% dari data yang dilatihkan, ditunjukkan oleh hasil keluaran yang ditampilkan oleh komputer sesuai dengan target yang telah ditentukan. Adapun grafik hasil keluaran dari 40 data sample suara sebagai berikut:



Gambar 6. Grafik hasil klasifikasi

Kemudian jaringan diuji dengan 4 data baru untuk menentukan seberapa besar jaringan mampu mengklasifikasikan suara. Berikut hasil pengujiannya:

Tabel 2. Tabel Hasil Pengujian dari 4 sample suara baru

NO	Nama File	std ZCR	Mean ZCR	std STE	Mean STE	std SC	Mean SC	Hasil	Cluster	Keterangan
1	Test_arif.wav	0,01910	0,02360	0,06650	0,03890	0,03150	0,0924	0,9998	Male	Berhasil
2	Test_nesi.wav	0,02370	0,02820	0,05380	0,03000	0,04180	0,1093	1,270E-05	Female	Berhasil
3	Test_ednofri.wav	0,03310	0,02670	0,04670	0,02600	0,04110	0,096	1,0000	Male	Berhasil
4	Test_retno.wav	0,02840	0,02970	0,08670	0,03410	0,03830	0,1181	0,0030	Female	Berhasil

Hasil pengujian menunjukkan bahwa empat data test tersebut sesuai dengan target.

Kesimpulan

1. Sistem klasifikasi suara menggunakan jaringan syaraf tiruan *Backpropagation* dan algoritma *Levenberg-Marquadt* berjalan sangat baik dan cepat karena kompleksitas perhitungan yang tidak terlalu tinggi.
2. Keunggulan sistem jaringan syaraf tiruan antara lain pada proses cepat, akurat dan minimalisasi kesalahan.
3. Sebagai langkah pengembangan agar didapatkan hasil yang optimal, jaringan perlu dilatihkan dengan data berjumlah banyak dan bervariasi sehingga tingkat akurasi meningkat.

Daftar Pustaka

- 1 Berghen, Frank Vanden, *Levenberg-Marquardt algorithms vs Trust Region algorithms*, IRIDIA, University Libre de Bruxelles, November 12, 2004
- 2 Deller, Proakis and Hansen (1993), *Discrete-Time Processing of Speech Signals*, Prentice Hall
- 3 Duane Hanselman, Bruce Littlefield, 2002, *MATLAB, Bahasa Komputasi Teknis*, Pearson Education Asia dan Penerbit Andi, Yogyakarta.
- 4 Fausett, Lauren., *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*, Englewood Cliffs, NJ : Prentice Hall, 1994.
- 5 Giannakopoulos, Theodoros (2006), *Some Basic Audio Features*, Department of Informatics and Telecommunications University of Athens, Greece, www.di.uoa.gr/~tyiannak
- 6 Gill, P. R.; Murray, W.; and Wright, M. H. *The Levenberg-Marquardt Method.*, in *Practical Optimization*. London: Academic Press, pp. 136-137, 1981.
- 7 Hermawan, Arief (2006), *Jaringan Saraf Tiruan Teori dan Aplikasi*, Andi Offset, Yogyakarta.
- 8 Kristian Sandberg, 2001, *An overview of Fourier analysis for signal processing*, Dept. of Applied Mathematics, University of Colorado at Boulder, <http://amath.colorado.edu/courses/3310/2001fall/Improc/Webpages/fourier/fourier.css>.
- 9 Levenberg, K. *A Method for the Solution of Certain Problems in Least Squares*. Quart. Appl. Math. Vol. 2, pp. 164-168, 1944.
- 10 Marquardt, D. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. SIAM J. Appl. Math. Vol. 11, pp. 431-441, 1963.
- 11 Rabiner dan Juang (1993), *Fundamentals of Speech Recognition*, Prentice Hall
- 12 Siang, Jong Jek, Drs., M.Sc. (2005), *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, Andi Offset, Yogyakarta.
- 13 *Speech Coding in MATLAB*, <http://www.eas.asu.edu/~speech/education/educ1.html>