Available online at TRANSMISI Website http://ejournal.undip.ac.id/index.php/transmisi

TRANSMISI, 13 (3), 2011, 82-86

Research Article

Aplikasi Pengenalan Ucapan dengan Ekstraksi *Mel-Frequency Cepstrum Coefficients* (MFCC) Melalui Jaringan Syaraf Tiruan (JST) *Learning Vector Quantization* (LVQ) untuk Mengoperasikan Kursor Komputer

Angga Setiawan[1], Achmad Hidayatno[2], R. Rizal Isnanto[2]

1. *Mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang*
2. *Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang*

Abstract

During this time, computer cursor operation was done by pressing and moving the mouse. So, this is less flexible for computer user that require movement in operating a computer, since to use mouse comfortably someone has to sit. Moreover, physical completeness is required for mouse operating, so that for someone who has physical disabilities feels difficult to operate it. Therefore, it is required to develop a system that provides a better comfort and flexibility not only for the healthy user computer but also for the user computer who has physical disabilities. In this final project, computer cursor operation program via voice is created. With this program, someone will have more flexibility when operating the computer cursor and also people with physical disabilities is enabled to communicate with computer. Voice recognition is a technology that is apllied in this program, with the feature extraction process used MFCC (Mel-Frequency Cepstrum Coefficients) method. As for the recognitions process used artificial neural network type LVQ (Learning Vector Quantization). Voice is passed through a microphone and then it is analyzed by MFCC to produce MFCC coefficients. These coefficients are used as input vector for LVQ neural network and used as data to train the network until it has the classification capability. Programming language that is used in creating this software is Delphi programming language. Based on the result of the testing program, it is found that the success percentage rate of voice recognition with training data, that is data which is derived from databases that have been recorded and trained into the program which amounts to 240 data, is 88,89 %. While in the testing with test data, that is data which is derived from the real time sayings of respondents which is amounts to 240 data, it is found that the success percentage rate of voice recognition is 83,99 %.

Keyword *: voice recognition, computer cursor, MFCC, LVQ*

I. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya teknologi, komunikasi yang dilakukan oleh manusia tidak hanya terbatas pada komunikasi antara manusia dengan manusia tetapi juga sudah berkembang komunikasi antara manusia dengan mesin (komputer). Komunikasi yang dilakukan antara manusia dengan komputer dilakukan dengan bantuan alat seperti *mouse*, *keyboard*, mikrofon, dan sebagainya.

Akan tetapi komunikasi antara manusia dengan komputer tidak bisa dinikmati oleh semua orang karena untuk melakukannya diperlukan kelengkapan dan kesehatan fisik manusia. Hal ini menyebabkan para penyandang cacat fisik sulit untuk melakukan komunikasi dengan komputer.

Oleh karenanya, diperlukan adanya inovasi teknologi yang memungkinkan para penyandang cacat fisik untuk melakukan komunikasi dengan komputer. Dalam Penelitian ini, dibuat sebuah metode komunikasi dengan komputer melalui suara. Secara spesifik, metode ini memungkinkan manusia untuk mengoperasikan kursor komputer melalui suaranya.

Metode yang dibuat ini merupakan salah satu aplikasi dari pengenalan ucapan (*voice recognition*), yakni sebuah pengembangan sistem yang memungkinkan komputer untuk dapat menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali ucapan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan pola tertentu yang tersimpan dalam perangkat.

Metode ekstraksi ciri yang digunakan dalam penelitian ini adalah MFCC (*Mel-Frequency Cepstrum Coefficients*) sedangkan untuk proses pembelajaran sistem digunakan Jaringan Syaraf Tiruan tipe LVQ (*Learning Vector Quantization*). Bahasa pemrograman yang digunakan dalam Penelitian ini adalah bahasa pemrograman Delphi.

1.2 Tujuan

Tujuan dari Penelitian ini adalah membuat suatu program aplikasi dari pengenalan ucapan dengan ekstraksi *Mel-Frequency Cepstrum Coefficients* (MFCC) melalui Jaringan Syaraf Tiruan (JST) *Learning Vector Quantization* (LVQ) untuk mengoperasikan kursor komputer.

1.3 Batasan Masalah

Untuk menyederhanakan pembahasan pada Penelitian ini, masalah dibatasi sebagai berikut :

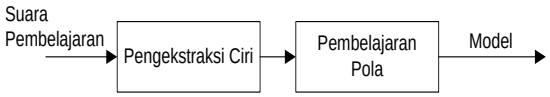
1. Data masukan (pada basis data) berupa sinyal suara yang diambil dari 6 orang responden (3 pria dan 3 wanita).
2. Pengoperasian kursor komputer terbatas hanya pada gerakan ke kanan, kiri, bawah, atas , klik kiri, *double click*, dan klik kanan.

3. Derau yang turut terekam pada proses perekaman diabaikan.
4. Aplikasi yang dibuat hanya dijalankan pada sistem operasi Microsoft Windows dan tidak membahas seluk beluk sistem operasinya.
5. Jenis bahasa pemrograman yang digunakan adalah bahasa pemrograman Delphi versi 7.

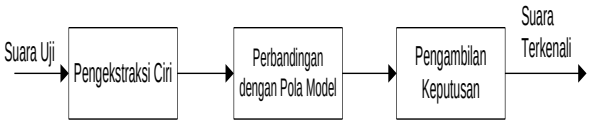
II. DASAR TEORI

2.1 Pengenalan Suara

Pengenalan suara merupakan salah satu upaya untuk dapat mengenali atau mengidentifikasi suara sehingga dapat dimanfaatkan untuk berbagai aplikasi. Secara umum tahap pengenalan suara dibagi menjadi dua bagian, yakni tahap pembelajaran pola dan tahap pengenalan suara melalui perbandingan pola. Blok diagram pembelajaran pola dan pengenalan suara ditunjukan pada Gambar 1.



(a) Blok diagram pembelajaran pola.



(b) Blok diagram pengenalan suara.

Gambar 1 Blok diagram pembelajaran pola dan pengenalan suara

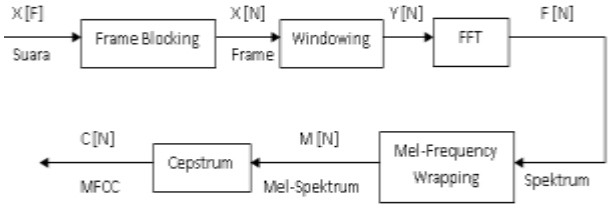
Berikut ini merupakan penjelasan dari masing-masing blok :

1. Pengekstraksi Ciri
   Bagian ini merupakan proses mendapatkan sederetan besaran pada bagian sinyal masukan untuk menetapkan pola pembelajaran atau pola uji. Untuk sinyal suara, ciri-ciri besaran biasanya merupakan keluaran dari beberapa bentuk teknik analisis spektrum seperti LPC (*Linear Predictive Coding*) atau MFCC (*Mel-Frequency Cepstrum Coefficients*).
2. Pembelajaran Pola
   Satu atau lebih pola uji yang berhubungan dengan bunyi suara dari kelas yang sama, digunakan untuk membuat pola representatif dari ciri-ciri kelas tersebut. Hasilnya yang biasa disebut dengan pola referensi, dapat menjadi sebuah model yang mempunyai karakteristik bentuk statistik dari ciri-ciri pola referensi.
3. Perbandingan dengan Pola Model
   Pola uji yang dikenali, dibandingkan dengan setiap kelas pola referensi. Kesamaan besaran antara pola uji dengan setiap pola referensi akan dihitung.
4. Pengambilan Keputusan
   Bagian ini merupakan proses menentukan kelas pola referensi mana yang paling cocok untuk pola uji berdasarkan klasifikasi pola.

2.2 *Mel-Frequency Cepstrum Coefficients* (MFCC)

MFCC didasarkan atas variasi *bandwidth* kritis terhadap frekuensi pada telinga menusia yang merupakan filter yang bekerja secara linier pada frekuensi rendah dan bekerja secara logaritmik pada frekuensi tinggi. Filter ini digunakan untuk menangkap karakteristik fonetis penting dari sinyal ucapan. Untuk meniru kondisi telinga, karakteristik ini digambarkan dalam skala mel-frekuensi, yang merupakan frekuensi linier di bawah 1000 Hz dan frekuensi logaritmik di atas 1000 Hz.

Biasanya frekuensi pencuplikan yang digunakan diatas 10000 Hz agar dapat meminimalkan efek *aliasing* pada konversi analog-digital. Diagram blok dari pemroses MFCC dapat dilihat pada Gambar 2.



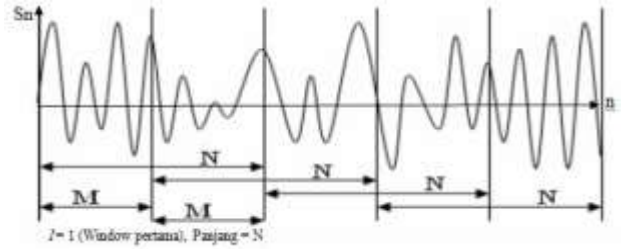
Gambar 2 Diagram blok pemroses MFCC

Untuk lebih jelasnya, masing-masing proses pada diagram pemroses MFCC akan diuraikan berikut ini

2.2.1 *Frame Blocking*

Pada langkah ini, sinyal ucapan yang terdiri dari S sampel (X(S)) dibagi menjadi beberapa *frame* yang berisi N sampel, masing-masing *frame* dipisahkan oleh M (M<N). *Frame* pertama berisi sampel N pertama. *Frame* kedua dimulai M sampel setelah permulaaan *frame* pertama, sehingga *frame* kedua ini *overlap* terhadap *frame* pertama sebanyak N-M sampel. Seterusnya, *frame* ketiga dimulai M sampel setelah *frame* kedua (juga *overlap* sebanyak N-M sampel terhadap *frame* kedua). Proses ini berlanjut sampai seluruh suara tercakup dalam *frame*. Hasil dari proses ini adalah matriks dengan N baris dan beberapa kolom sinyal X[N].

Proses ini tampak pada Gambar 3, *Sn* adalah nilai sampel yang dihasilkan, dan *n* menunjukkan urutan sampel yang akan diproses.



Gambar 3 Proses *frame blocking*

2.2.2 *Windowing*

Langkah selanjutnya adalah *windowing* setiap *frame* untuk meminimalisir diskontinuitas sinyal pada permulaan dan akhir setiap *frame*. Konsepnya adalah meruncingkan sinyal ke angka nol pada permulaan dan akhir setiap *frame*. Bila *window* didefinisikan sebagai w(n), $0 \le n \le N-1$, dengan N adalah jumlah sampel dalam tiap *frame*, maka hasil dari proses ini adalah sinyal :

$$y(n) = x(n)w(n), 0 \le n \le N-1$$

dengan $y(n)$ = sinyal hasil *windowing* sampel ke-n
$x(n)$ = nilai sampel ke-n
$w(n)$ = nilai *window* ke-n
N = jumlah sampel dalam *frame*

Secara khusus (dalam masalah ini), secara empiris, digunakan *hamming window*, yang mempunyai bentuk,

$$w(n) = 0{,}54 + 0{,}46\cos\left(\frac{2\pi n}{N-1}\right), 0 \le n \le N-1$$

2.2.3 *Fast Fourier Transform* (FFT)

Proses selanjutnya adalah Alihragam Fourier Cepat (*Fast Fourier Transform*), yang mengkonversi setiap *frame* yang berisi N sampel dari ranah waktu ke ranah frekuensi.

FFT adalah sebuah algoritma cepat untuk implementasi *Discrete Fourier Transform* (DFT) yang dioperasikan pada sebuah sinyal waktu-diskret yang terdiri dari N sampel sebagai berikut :

$$f(n) = \sum_{K=0}^{N-1} y_k e^{-2\pi jkn/N}, n = 0,1,2,\ldots,N-1$$

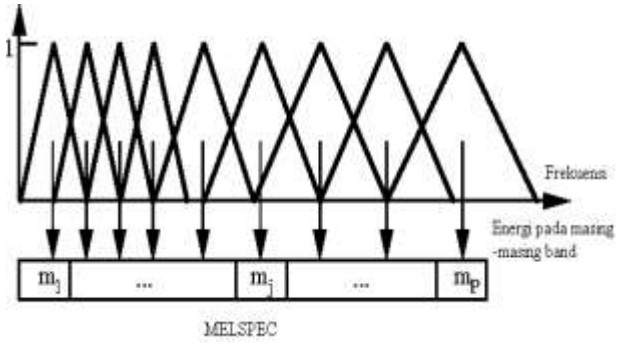
2.2.4 *Mel-Frequency Wrapping*

Studi psikofisik telah menunjukkan bahwa persepsi manusia tentang frekuensi suara untuk sinyal ucapan tidak mengikuti skala linier. Jadi, untuk setiap nada dengan frekuensi sesungguhnya f, dalam Hz, sebuah pola diukur dalam sebuah skala yang disebut 'mel'. Skala 'mel frekuensi' adalah skala frekuensi linier di bawah 1000 Hz dan skala logaritmik di atas 1000 Hz.

Skala ini didefinisikan oleh Stanley Smith, John Volkman dan Edwin Newman sebagai :

$$mel(f) = 2595 * \log_{10}(1 + \frac{f}{700})$$

Sebuah pendekatan untuk simulasi spektrum dalam skala mel adalah dengan menggunakan *filter bank* yang diletakkan secara seragam dalam skala mel yang ditunjukkan pada Gambar 4.



Gambar 4 Contoh *mel-spaced filter bank*

Bila spektrum F[N] adalah masukan proses ini, maka keluarannya adalah spektrum M[N] yang merupakan spektrum F[N] termodifikasi yang berisi *power output* dari filter-filter ini. Koefisien spektrum mel dinyatakan dengan K, dan secara khusus ditentukan berharga 20.

Dalam *mel-frequency wrapping*, sinyal hasil FFT dikelompokkan ke dalam berkas filter triangular ini. Maksud pengelompokan di sini adalah setiap nilai FFT dikalikan terhadap *gain filter* yang bersesuaian dan hasilnya dijumlahkan. Maka setiap kelompok mengandung sejumlah bobot energi sinyal sebagaimana dinyatakan sebagai $m_1 \ldots m_p$ seperti tampak pada Gambar 4.

2.2.5 Cepstrum

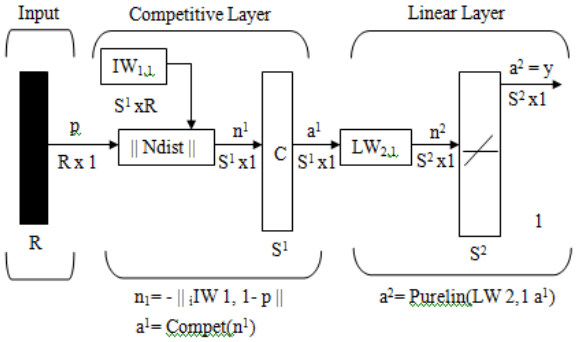
Cepstrum adalah sebutan kebalikan untuk *spectrum*. *Cepstrum* biasa digunakan untuk mendapatkan informasi dari suatu sinyal suara yang diucapkan oleh manusia.Pada langkah terakhir ini, spektrum log mel dikonversi menjadi *cepstrum* menggunakan *Discrete Cosine Transform* (DCT). Hasil dari proses ini dinamakan *Mel-Frequency Cepstrum Coefficients* (MFCC).

MFCC ini adalah hasil alihragam cosinus dari logaritma *short-term power spectrum* yang dinyatakan dalam skala mel-frekuensi. Bila *mel power spectrum coefficients* dinotasikan sebagai S_k, k = 1,2…..K , Minh N.Do mendefinisikan koefisien dari MFCC ($\hat{c}_n$) sebagai :

$$cn = \sum_{k=1}^{K} (\log Sk)\cos[n(k - \frac{1}{2})\frac{\pi}{K}], n = 1,2,\ldots,K$$

2.3. Jaringan Syaraf Tiruan *Learning Vector Quantization* (LVQ)

Learning Vector Quantization (LVQ) adalah suatu metode untuk melakukan pembelajaran atau pelatihan pada lapisan kompetitif yang terawasi. LVQ belajar mengklasifikasikan vektor masukan ke kelas target yang ditentukan oleh pengguna. Arsitektur jaringan LVQ ditunjukkan pada gambar berikut ini.



Gambar 5 Arsitektur jaringan LVQ

Jaringan LVQ terdiri dari dua lapisan tersembunyi yaitu lapisan kompetitif dan lapisan linear. Lapisan kompetitif disebut juga *Self Organizing Map* (SOM). Disebut lapisan kompetitif karena *neuron-neuron* berkompetisi dengan algoritma kompetisi yang akan menghasilkan *neuron* pemenang (*winning neuron*).

Pada jaringan LVQ, bias pada lapisan kompetitif dihilangkan. Lapis linear mengalihragamkan kelas-kelas pada lapisan kompetitif ke klasifikasi target yang ditentukan oleh pengguna.

2.3.1 Algoritma Pembelajaran LVQ

Algoritma ini akan mengubah bobot satu *neuron* yang paling dekat dengan vektor masukan. Misal vektor masukan **x** = (x_1, x_2,….., x_n), keluaran vektor bobot *neuron* ke-j adalah $\mathbf{w_j}$ = (w_{1j}, w_{2j},…., w_{nj}), $\mathbf{C_j}$ = kelas yang diwakili *neuron* ke-j, **T** = kelas yang benar untuk masukan **x**, dan jarak euclidean antara vektor masukan dan vektor bobot dinyatakan :

$$d(j) = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}$$

dengan $x - w_j = ((x_i - w_{ij}), (x_2 - w_{2ij}).., (x_n - w_{nj}))$,
maka perubahan bobot *neuron* dilakukan dengan langkah-langkah berikut :

Langkah 0 : Inisialisasi vektor bobot dan laju pembelajaran α

Langkah 1 : Jika kondisi untuk berhenti salah, laksanakan langkah 2 sampai 6

Langkah 2 : Untuk tiap vektor masukan **x**, laksanakan langkah 3 dan 4

Langkah 3 : Hitung nilai j sehingga **d(j)** minimum

Langkah 4 : Mengubah bobot *neuron* ke-j sebagai berikut :
Jika $\mathbf{T = C_j}$ maka

$$w_j(t+1) = w_j(t) + \alpha(x - w_j(t))$$

yaitu mendekatkan vektor bobot **w** ke vektor masukan **x**

Jika $\mathbf{T} \neq \mathbf{C_j}$ maka

$$w_j(t+1) = w_j(t) - \alpha(x - w_j(t))$$

yaitu menjauhkan vektor bobot **w** ke vektor masukan **x**

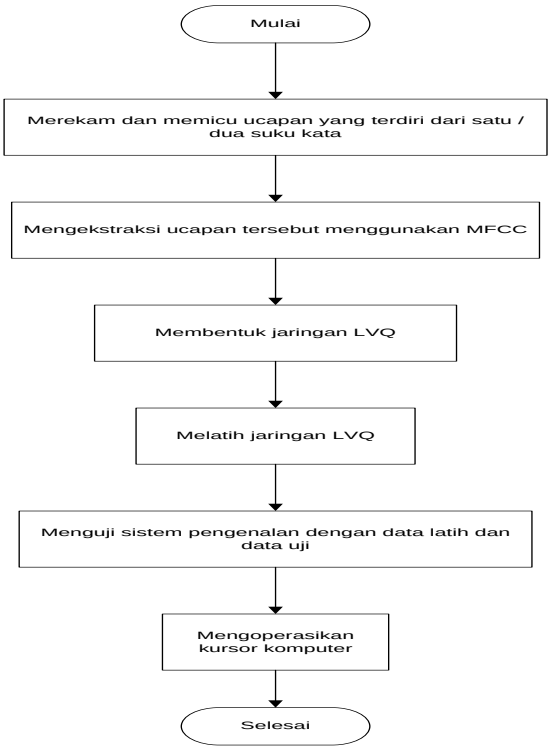
Langkah 5 : Mengurangi nilai laju pembelajaran α

Langkah 6 : Mengecek kondisi untuk berhenti : Jumlah iterasi atau laju pembelajaran mencapai nilai yang sangat kecil.

III. PERANCANGAN PROGRAM

3.1 Gambaran Umum

Secara umum pembuatan program aplikasi dapat dilihat pada Gambar 6.



Gambar 6 Alur pembuatan program aplikasi pengenalan suara untuk mengoperasikan kursor komputer

3.2 Perekaman dan Pemicuan Ucapan

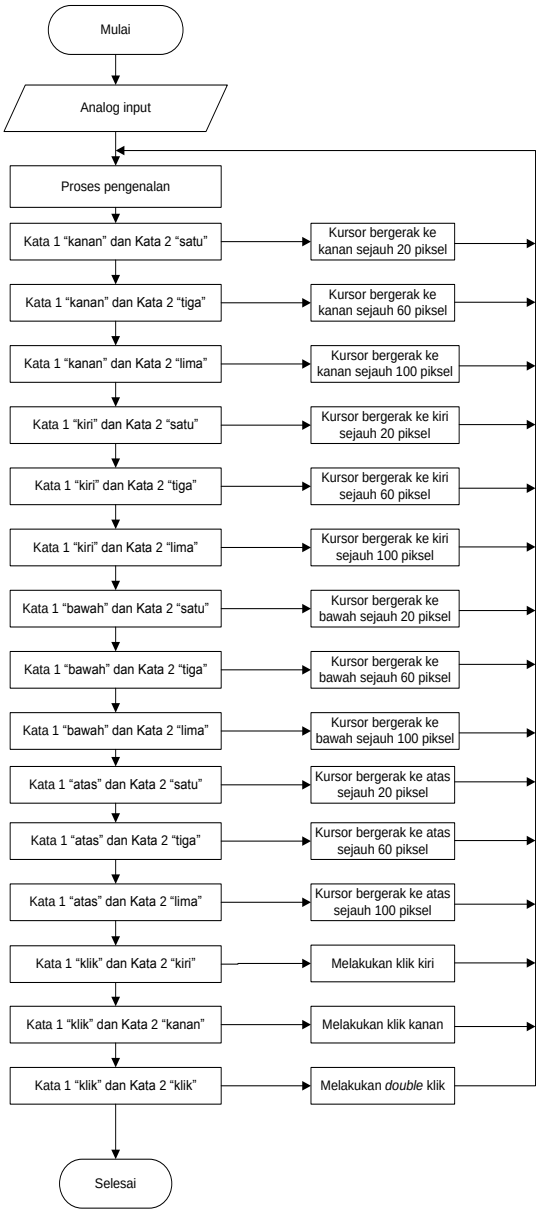
Ucapan yang akan dijadikan objek dalam pembuatan program direkam sekaligus dipicu sebagai data masukan bagi proses pengenalan dan pembentukan jaringan LVQ. Data masukan ucapan diperoleh melalui mikrofon. Sinyal tersebut dengan frekuensi pencuplikan (*frequency sampling*) sebesar 11025 Hz, resolusi delapan bit dan waktu pemicuan sebanyak satu detik (11025 sampel).

Ucapan-ucapan yang akan dikenali ada 8 kata yaitu, “kanan”, “kiri”, “bawah”, “atas”, “satu”, “tiga”, “lima”, dan “klik”. Untuk setiap ucapan diambil sampel ucapan sebanyak enam orang dan setiap orang mengucapkan sebanyak 5 kali dalam setiap kata tersebut.

Ketika program eksekusi mulai dijalankan maka proses perekaman terhadap sinyal masukan berupa suara dilakukan secara terus menerus sekaligus ditampilkan bentuk sinyalnya secara waktu-nyata *(real-time)*. Jika ada sinyal masukan dengan nilai amplitudo sebesar 0,7 dalam satuan ternormalisasi maka proses pemicuan dimulai.

3.3 Proses Menjalankan Aplikasi

Proses menjalankan aplikasi dapat dilihat pada Gambar 7.



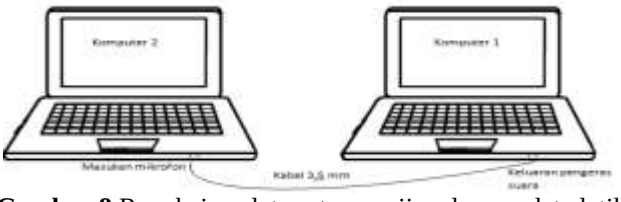
Gambar 7 Alur program utama aplikasi pengenalan suara untuk mengoperasikan kursor komputer

IV. PENGUJIAN DAN ANALISIS

Pengujian program terbagi menjadi dua bagian yakni, pengujian dengan data latih dan pengujian dengan data uji.

4.1 Pengujian Dengan Data Latih

Pengujian dengan data latih dilakukan dengan menggunakan dua komputer yang dirangkai seperti pada Gambar 8



Gambar 8 Rangkaian alat saat pengujian dengan data latih

Komputer pertama digunakan untuk memainkan data latih sedangkan komputer kedua digunakan untuk mengenali data latih yang dimainkan oleh komputer pertama. Data latih

berasal dari basisdata suara yang telah direkam dan dilatihkan ke program. Sedangkan basisdata suara itu sendiri berasal dari ucapan responden yang direkam sebanyak 5 kali untuk setiap target kata, sehingga jumlah dari data latih sebanyak = 5 x 6 (responden) x 8 (target kata) = 240 data.

Pengujian dilakukan dengan menggunakan 15 instruksi, yang mana masing-masing instruksi tersebut merupakan gabungan dari 2 kata yang dikenali oleh program. Pengujian dikatakan berhasil apabila instruksi tersebut muncul di layar monitor dan komputer mengoperasikan perintah yang ada pada instruksi tersebut. Pengujian dilakukan sebanyak 5 kali untuk masing-masing instruksi.

4.2 Pengujian Dengan Data Uji

Berbeda dengan pengujian data latih, pengujian dengan data uji dilakukan hanya dengan menggunakan satu komputer dan data yang diujikan berasal dari responden yang mengucapkan kata secara waktu-nyata (*real-time*). Setiap responden mengucapkan 15 instruksi, yang merupakan gabungan dari dua kata yang dikenali oleh program, yang mana masing-masing instruksi tersebut diucapkan sebanyak 5 kali.

Pengujian dikatakan berhasil apabila instruksi yang diucapkan muncul di layar monitor dan komputer mengoperasikan perintah yang ada pada instruksi tersebut.

Pada Tabel 1 ditunjukkan persentase keberhasilan pengujian program dengan data latih dan data uji.

Tabel 1 Persentase keberhasilan pengenalan ucapan dengan data uji dan data latih

No	Instruksi yang diucapkan	%Keberhasilan dengan data latih	%Keberhasilan dengan data uji
1	"kanan-satu"	100 %	90 %
2	"kiri-satu"	83,33 %	76,66 %
3	"bawah-satu"	83,33 %	76,66 %
4	"atas-satu"	83,33 %	83,33 %
5	"kanan-tiga"	100 %	90 %
6	"kiri-tiga"	83,33 %	76,66 %
7	"bawah-tiga"	83,33 %	80 %
8	"atas-tiga"	83,33 %	83,33 %
9	"kanan-lima"	100 %	90 %
10	"kiri-lima"	83,33 %	76,66 %
11	"bawah-lima"	83,33 %	80 %
12	"atas-lima"	83,33 %	83,33 %
13	"klik-kanan"	100 %	93,33 %
14	"klik-kiri"	83,33 %	80 %
15	"klik-klik"	100 %	100 %
	% Rata-rata keberhasilan total	88, 89 %	83, 99 %

4.3 Faktor yang Berpotensi Mempengaruhi Tingkat Pengenalan Pengujian Data

Faktor-faktor yang mempengaruhi tingkat pengenalan ucapan pada program adalah sebagai berikut.

1. Kondisi Lingkungan
2. Kondisi dan intonasi suara responden
3. Letak Mikrofon
4. Cara perekaman sinyal suara
5. Kondisi Peralatan

V. PENUTUP

5.1 Kesimpulan

Dari pembahasan hasil aplikasi pengenalan suara yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut.

1. Keluaran dari MFCC adalah koefisien ciri yang berisi nilai-nilai yang mewakili sinyal ucapan.
2. Algoritma LVQ pada program ini digunakan untuk mengklasifikasikan masukan ke kelas target yang ditentukan.
3. Rata-rata persentase keberhasilan pengenalan suara program dengan menggunakan data latih adalah sebesar 88,89 %.
4. Rata-rata persentase keberhasilan pengenalan suara program dengan menggunakan data uji adalah sebesar 83,99 %.

5.2 Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah sebagai berikut.

1. Perlu dilakukan penambahan variasi ucapan, agar instruksi untuk mengoperasikan kursor komputer bisa lebih banyak seperti melakukan penambahan kata ”serong” untuk menggerakkan kursor komputer secara diagonal.
2. Perlu dilakukan penelitian lanjutan dengan menggunakan algoritma lain dalam proses ekstraksi ciri, misalnya dengan menggunakan algoritma LPC (*Linear Predictive Coding*).
3. Pada proses pembelajaran jaringan perlu dilakukan penelitian lanjutan dengan menggunakan Jaringan Syaraf Tiruan tipe lainnya, misalnya Jaringan Syaraf Tiruan Perambatan Balik..

Daftar Pustaka

[1] Bahri, K. S. dan W. Sjachriyanto, *Teknik Pemrograman Delphi*, Edisi Revisi, Informatika Bandung, Bandung, 2008.

[2] Chaedar, Z. A., *Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) Untuk Menjalankan Program Komputer*, Penelitian S-1, Universitas Diponegoro, Semarang, 2005.

[3] Developments,U.,*BASS*,http:// www.un4seen.com/, Agustus 2011.

[4] Gajic, Z., *Advanced Mouse Processing*, http://delphi.about.com/ od/ vclusing/ a/mouseadvanced.htm, Juli 2011.

[5] Ganchev, T., N. Fakotakis dan G. Kokkinakis, *Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task*, SPECOM Journal, Vol. 1, pp. 191-194, October 2005.

[6] Haykin, S., *Neural Networks*, Macmilian College Publishing Company.Inc, New York, 1994.

[7] Siang, J. J., *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*, Penerbit Andi, Yogyakarta, 2005.

[8] ---, *AHoFFT*, http:// read.pudn.com/ downloads65/ ebook/ 232427/ reconstruction/AHoFFT.pas .htm, Agustus 2011.