

APLIKASI SPEECH RECOGNITION BAHASA INDONESIA DENGAN METODE MEL-FREQUENCY CEPSTRAL COEFFICIENT DAN LINEAR VECTOR QUANTIZATION UNTUK PENGENDALIAN GERAK ROBOT

Anggoro Wicaksono, Sukmawati NE, Satriyo Adhy, Sutikno

Jurusan Ilmu Komputer/Informatika

Abstrak

JST (Jaringan Saraf Tiruan) banyak digunakan untuk aplikasi pengenalan pola, termasuk pengenalan pola ucapan yang digunakan sebagai perintah untuk menggerakkan robot. Metode JST yang dapat digunakan salah satunya adalah LVQ (Linear Vector Quantization). Pengenalan ucapan dengan jaringan saraf tiruan dapat didahului dengan menggunakan beberapa metode ekstraksi ciri, diantaranya adalah dengan menggunakan metode MFCC. *Mel-Frequency Cepstral Coefficient* (MFCC) adalah suatu metode dalam pemrosesan sinyal untuk memperoleh ciri dari tiap pola suara. Dalam aplikasi ini pengenalan ucapan meliputi lima kata yaitu maju, mundur, kanan, kiri, dan berhenti. Input suara berupa file *.wav dengan sample rate 44100 Hz. Ekstraksi ciri menggunakan *Mel-Frequency Cepstral Coefficient* (MFCC) sebelum dilakukan pengenalan menggunakan JST LVQ.

Kata kunci : Pengenalan ucapan, MFCC, JST, LVQ, Robot

1. PENDAHULUAN

Proses pengenalan suara oleh manusia mulai terbentuk sejak balita yaitu ketika sudah dapat mendengar dan mampu mengeluarkan suara. Proses ini tanpa disadari dilakukan melalui proses pembelajaran, yaitu belajar mengenal ucapan yang didengar. Pada manusia tidaklah begitu sulit untuk mengenali ucapan yang didengar, karena manusia mempunyai sistem informasi yang mampu mengenali pola dengan sangat baik.

Seiring perkembangan teknologi, manusia mulai menggunakan teknologi untuk melakukan pengenalan suara. Tetapi saat ini belum banyak perangkat teknologi yang mampu mengenali bahasa lisan manusia secara tepat. Hal ini disebabkan sulitnya perangkat teknologi untuk menangkap pesan lisan, lalu menerjemahkannya dan akhirnya menjalankan perintah yang terkandung dalam pesan lisan tersebut.

Salah satu penggunaan *speech recognition* yang mulai dikembangkan adalah penggunaan *speech recognition* untuk mengendalikan gerak robot. Teknologi pengendalian robot sudah memasuki tahap dimana perintah berupa suara. Sehingga tidak diperlukan lagi *remote control* untuk mengendalikan robot, tetapi hanya dengan menggunakan perintah suara robot sudah bisa dikendalikan.

Dalam penggunaan *speech recognition* untuk mengendalikan gerak robot, dibutuhkan suatu algoritma yang dapat digunakan untuk melakukan ekstraksi ciri dan pengenalan suara yang memungkinkan untuk mendapatkan tingkat akurasi yang cukup tinggi. Selain itu algoritma tersebut harus memiliki kecepatan yang cukup tinggi dalam mengenali perintah suara, sehingga robot dapat segera bergerak setelah diperintah melalui suara.

Mel-Frequency Cepstral Coefficient (MFCC) merupakan salah satu algoritma yang sering digunakan untuk ekstraksi ciri sebelum dilakukan pengenalan pola atau *pattern recognition*. MFCC dapat menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang dikandung dalam sinyal suara [2].

Linear Vector Quantization (LVQ) merupakan salah satu metode dalam *pattern recognition* yang cukup banyak mendapat perhatian. LVQ memiliki kemampuan untuk mengklasifikasikan *vector* masukan ke kelas target yang telah ditentukan. Pembelajaran dengan cara ini disebut dengan pembelajaran terarah (*supervised learning*) [5].

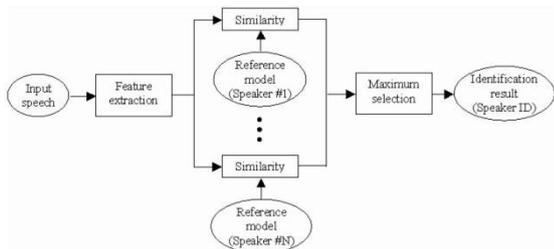
Penelitian ini mengusulkan bagaimana membuat suatu aplikasi *speech recognition* bahasa Indonesia dengan jaringan saraf tiruan *linear vector quantization* untuk pengendalian gerak robot.

2. TINJAUAN PUSTAKA

Pengenalan Suara

Pengenalan suara merupakan bahan penelitian yang menarik dalam beberapa tahun ini, meskipun masih terdapat beberapa masalah yang belum dapat diselesaikan dalam penelitian tersebut. Pengenalan suara pada dasarnya dibagi menjadi identifikasi suara dan verifikasi suara. Identifikasi suara adalah kegiatan memetakan sinyal suara dari sumber yang tidak dikenal ke dalam *database* yang berisi suara – suara yang sudah dikenali. Sedangkan verifikasi suara adalah kegiatan menentukan apakah suara yang dimasukkan cocok dengan suara yang telah diperkirakan sebelumnya. Proses identifikasi dapat dibedakan menjadi *text-dependent* dan *text-independent*. *Text-dependent* membutuhkan inputan tertentu yang lebih spesifik seperti *password* atau

kode pin Sedangkan *text-independent* mengharuskan sistem untuk dapat mengidentifikasi inputan yang lebih bebas. Proses pengenalan suara dari proses input sampai teridentifikasi ditunjukkan pada gambar 1 [1].



Gambar 1. Proses Pengenalan Suara [1]

Pada gambar 1 dijelaskan tentang proses pengenalan suara yang terdiri dari :

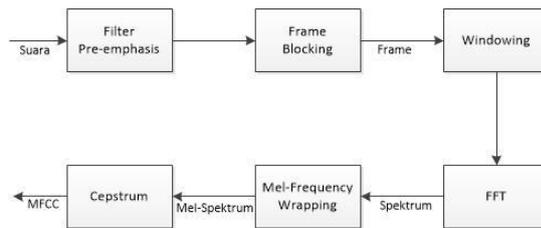
1. *Input speech* merupakan suara dimasukkan ke dalam sistem yang kemudian akan dikelola oleh sistem
2. *Feature extraction* merupakan bagian dimana hasil inputan suara diekstraksi cirinya dengan metode tertentu dengan tujuan untuk membersihkan inputan dari *noise*
3. *Similarity* merupakan bagian dimana hasil inputan yang sudah di bersihkan kemudian dibandingkan dengan data yang sudah ada menggunakan metode tertentu sehingga menghasilkan bobot – bobot tertentu
4. *Maximum selection* merupakan perbandingan dari bobot – bobot yang telah dihasilkan pada tahap sebelumnya untuk dicari bobot yang terbesar
5. *Identification result* merupakan tahap akhir dari sistem yang memberikan hasil dari pengenalan suara berupa suara yang cocok dengan data.

Mel-Frequency Cepstrum Coefficients (MFCC)

MFCC merupakan salah satu metode yang mengkonversikan sinyal suara menjadi beberapa paramater. MFCC memiliki beberapa keunggulan, yaitu:

1. Mampu menangkap karakteristik suara yang sangat penting bagi pengantar suara, atau dengan kata lain dapat menangkap informasi-informasi penting yang terkandung dalam sinyal suara.
2. Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang dikandungnya.
3. Mereplikasi organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara.

Ekstraksi ciri menggunakan MFCC juga merupakan adaptasi dari sistem pendengaran manusia, dimana sinyal suara akan difilter secara linear untuk frekuensi rendah (dibawah 1000 Hz) dan secara logaritmik untuk frekuensi tinggi (diatas 1000 Hz). Blok diagram untuk MFCC ditunjukkan pada gambar 2 [2].



Gambar 2. Blok diagram MFCC

Filter Pre-Emphasis

Proses pemfilteran sinyal suara diperlukan setelah proses perekaman atau sampling. Tujuan dari pemfilteran adalah untuk mendapatkan bentuk spectral frekuensi sinyal suara yang lebih halus. Filter pre-emphasis didasari oleh hubungan input/output dalam domain waktu yang dinyatakan dalam persamaan [7]:

$$y(n)=x(n)-ax(n-1)$$

Keterangan :

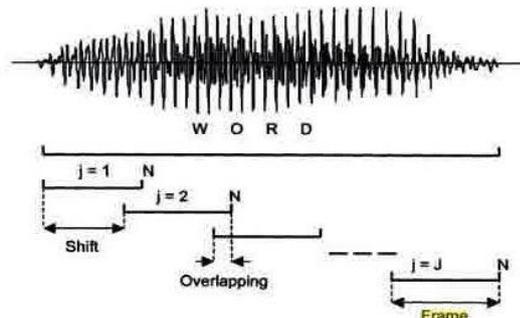
- y = hasil perhitungan sinyal suara
- n = nomer urutan sinyal suara
- x = masukan sinyal suara
- a = konstanta filter pre-emphasis, biasanya bernilai $0,9 \leq a \leq 1,0$

Proses pemfilteran dengan pre-emphasis mempunyai dua keuntungan utama, yaitu [8]:

1. Sinyal suara secara alami memiliki kemiringan spektral yang negatif berkisar 20 dB/decade karena karakter fisik dari pembentukan ucapan. Filter pre-emphasis berfungsi untuk mengimbangi kemiringan alami spektral sebelum dilakukan analisis, sehingga dapat meningkatkan efisiensi dari analisis.
2. Pendengaran manusia lebih sensitif dengan spektrum yang berkisar di atas 1kHz. Filter pre-emphasis menguatkan spektrum di area di atas 1 kHz tersebut.

Frame Blocking

Frame Blocking adalah suatu proses di mana sinyal suara dibagi menjadi beberapa potongan yang nantinya dapat memudahkan dalam perhitungan dan analisa suara. Setiap potongan-potongan dari sinyal suara disebut frame. Satu frame terdiri dari beberapa sampel tergantung tiap detik suara akan disampel dan berapa besar frekuensi samplangnya [9].



Gambar 3. Frame Blocking [8]

Berdasarkan pada gambar 3, sinyal suara dibagi menjadi beberapa frame dan saling bertumpang tindih (*overlapping*). *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap frame. Panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30% - 50 % dari panjang frame [10].

Frame pada *frame blocking* pada umumnya memiliki panjang antara 10 – 30ms. Panjang *frame* yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Di satu sisi, ukuran dari *frame* harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik. Tetapi di sisi lain ukuran *frame* juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik [10].

Proses Windowing

Proses framing dapat menyebabkan terjadinya kebocoran spektral atau *aliasing*. Efek ini dapat terjadi karena rendahnya jumlah *sample rate*, atau karena proses *frame blocking* dimana menyebabkan sinyal menjadi *discontinue*. Untuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses *farming* harus melewati proses *windowing*.

Fungsi *window* yang paling sering digunakan dalam aplikasi *speech recognition* adalah *Hamming Window*. Fungsi ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB) selain itu *noise* yang dihasilkan juga tidak telalu besar (kurang lebih 1,36 BINS) [15].

Fungsi *Hamming Window* adalah sebagai berikut:

$$w_{ham}(n) = 0.54 - 0.46 \cos \frac{2\pi n}{M-1}$$

Keterangan:

$w_{ham}(n)$ = keluaran dari proses *Hamming Window*

n = 0, 1, 2, ..., $M - 1$

M = panjang *frame*

Fast Fourier Transform (FFT)

FFT merupakan salah satu metode untuk transformasi sinyal suara menjadi sinyal frekuensi. Artinya proses perekaman suara disimpan dalam bentuk digital berupa gelombang spektrum suara berbasis frekuensi. Hasil dari proses FFT menghasilkan pendeteksian gelombang frekuensi domain dalam bentuk diskrit [9].

FFT merupakan turunan dari persamaan *Discrete Fourier Transform* (DFT) dimana jumlah perhitungan digital pada DFT dapat dikurangi secara signifikan. Sehingga dengan adanya FFT maka perhitungan digital terhadap spektrum-spektrum frekuensi dapat diwujudkan secara sederhana dalam implementasinya. Pinsip dasar FFT adalah menguraikan penghitungan N-titik DFT menjadi penghitungan DFT dengan ukuran yang lebih kecil dan memanfaatkan periodisitas dan simetri dari

bilangan kompleks. FFT dapat dituliskan dalam bentuk sinusoidal sebagai berikut [9]:

$$x(k) = \frac{1}{N} \sum_{n=1}^N x(n) \cos \left(\frac{2\pi kn}{N} \right) - j \sin \left(\frac{2\pi kn}{N} \right)$$

Keterangan:

N = koefisien sinus dan cosinus pada $2\pi kn$

k = indeks dari frekuensi pada frekuensi ke- N

n = indeks waktu

$x(k)$ = nilai dari spektrum ke- k (domain frekuensi)

$x(n)$ = nilai sinyal pada domain waktu

Mel-Frequency Wrapping

Mel-Frequency Wrapping biasanya dilakukan menggunakan *Filterbank*. *Filterbank* adalah salah satu bentuk dari filter yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari *frequency band* tertentu dalam sinyal suara. Pada MFCC, *Filterbank* diterapkan dalam domain frekuensi. Berikut ini adalah rumus yang digunakan dalam perhitungan *filterbanks* [2].

Keterangan:

$$y[i] = \sum_{j=1}^N S[j] H_i[j]$$

Keterangan :

$y[i]$ = hasil dari *mel-frequency wrapping*

$S[j]$ = *magnitude spectrum* pada frekuensi j

$H_i[j]$ = koefisien *filterbank* pada frekuensi ($1 \leq i \leq M$)

M = jumlah channel dalam *filterbank*

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) [3].

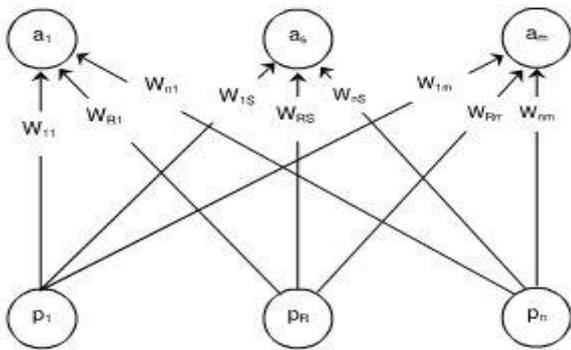
MFCC ini adalah hasil alihragam cosinus dari logaritma *short-term power spectrum* yang dinyatakan dalam skala mel-frekuensi. Bila *mel power spectrum coefficients* dinotasikan sebagai S_k , $k = 1, 2, \dots, K$, Minh N.Do mendefinisikan koefisien dari MFCC (c_n) sebagai [3] :

$$c_n = \sum_{k=1}^K (\log S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], n = 1, 2, \dots, K$$

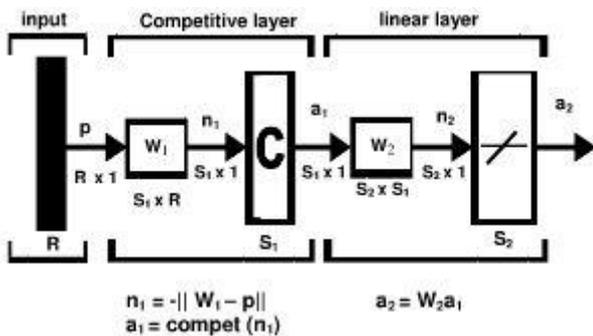
Jaringan Saraf Tiruan Linear Vector Quantization (LVQ)

LVQ merupakan salah satu jaringan saraf tiruan dengan pembelajaran terbimbing (*supervised learning*). LVQ mengklasifikasikan *input* secara berkelompok ke dalam kelas yang sudah didefinisikan melalui jaringan yang telah dilatih. Dengan kata lain LVQ mendapatkan *n input* dan mengelompokkan ke dalam *m output*[4].

Pada dasarnya arsitektur LVQ sama dengan *self-organizing map* kohonen (tanpa struktur topologi yang diasumsikan dengan *layer output*). Sebagai tambahan setiap *layer output* mempunyai kelas yang diketahui [5].



Gambar 4. Arsitektur LVQ sederhana[5]



Gambar 5. Arsitektur LVQ dengan *competitive layer* dan *linear layer*[5]

Pada gambar 4 menunjukkan arsitektur LVQ sederhana, dimana hanya terdapat *layer input*, bobot dan *layer output*. Pada gambar 5 menunjukkan arsitektur LVQ yang terdiri dari sebuah *layer input*, sebuah *competitive layer*, dan sebuah *linear layer*. *Competitive layer* mengklasifikasikan vektor *input* ke dalam sejumlah *cluster* berdasarkan jarak yang terdapat diantara masing – masing vektor masukannya. Selanjutnya *linear layer* memetakan kelas yang didapatkan oleh *competitive layer* ke dalam kelas yang telah didefinisikan sebelumnya oleh pengguna, dalam layer ini menggunakan fungsi aktivasi *linear* dengan tujuan agar *input* sebanding dengan *outputnya*[5].

Algoritma LVQ bertujuan akhir mencari nilai bobot yang sesuai untuk mengelompokkan vektor – vektor ke dalam kelas tujuan yang telah di inisialisasi

pada saat pembentukan jaringan LVQ. Sedangkan algoritma pengujiannya adalah menghitung nilai *output* (kelas vektor) yang terdekat dengan vektor *input*, atau dapat disamakan dengan proses pengklasifikasian. Algoritma pembelajaran LVQ adalah sebagai berikut [5] :

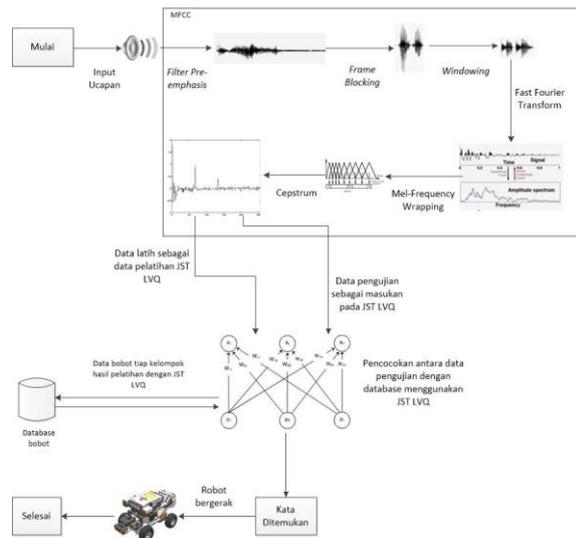
1. Inisialisasi vektor referensi ; inisialisasi rating pembelajaran α (0)
2. Ketika kondisi berhenti adalah *false*, lakukan langkah 2 sampai 6
3. Untuk setiap *input* pelatihan vektor x lakukan langkah 3 – 4
4. Temukan j hingga $\|p - W_j\|$ minimum
5. Perbaharui W_s sebagai berikut :
 Jika $T = C_s$, maka
 W_s (baru) = W_s (lama) + $\alpha[p - W_s$ (lama)];
 Jika $T \neq C_s$, maka
 W_s (baru) = W_s (lama) – $\alpha[p - W_s$ (lama)];
6. Kurang rating pelatihan
7. Tes kondisi berhenti, yaitu kondisi yang mungkin menetapkan sebuah jumlah tetap dari iterasi atau rating pembelajaran mencapai nilai kecil yang cukup.

Keterangan :

- p : vektor pelatihan (*input*) ($p_1, \dots, p_R, \dots, p_n$)
- T : kategori yang tepat atau kelas untuk vektor pelatihan
- W_s : bobot vektor untuk unit *output* ke- s ($W_{11}, \dots, W_{R1}, \dots, W_{n1}$)
- C_s : kategori atau kelas yang ditampilkan oleh unit *output* ke- s
- $\|p - W_j\|$: jarak *Euclidean* antara vektor *input* dan bobot vektor untuk *layer output* ke- s .

Arsitektur Sistem

Secara umum model sistem yang akan dilakukan untuk menangani masalah yang diangkat dalam penelitian ini dapat dilihat pada gambar di bawah ini:



Gambar 7 Arsitektur Sistem

3. GARIS BESAR PENYELESAIAN MASALAH

Berdasarkan arsitektur sistem pada gambar 7, Aplikasi ini dapat dikelompokkan menjadi empat kelompok yaitu input ucapan, *feature extraction*, *pattern matching*, dan gerak robot.

1. Input Ucapan

Input suara dilakukan di dalam ruangan tertutup dan suasana yang sunyi. Perekaman suara dalam format .wav dengan frequency sampling (fs) 44100 Hz, 16 bit dengan kanal mono. Perekaman suara diambil dari 5 orang berbeda.

2. Pre-Processing

Pre-Processing bertujuan untuk mengolah suara agar dapat diambil karakteristiknya dan menghilangkan noise untuk mempermudah proses pada tahap ekstraksi ciri dan pengenalan pola. Pada tahap pre-processing terdapat empat proses yang dilakukan yaitu :

3. Feature Extraction

Proses *feature extraction* menggunakan *Mel-Frequency Cepstral Coefficient* (MFCC). Dalam MFCC memiliki beberapa tahapan yaitu :

a. Filter Pre-emphasis

Pada tahap ini sinyal suara yang telah direkam dilakukan penghilangan noise dengan filter pre-emphasis.

b. Frame Blocking

Setelah dilakukan penghilangan noise dengan filter pre-emphasis, sinyal suara selanjutnya dipotong dengan menggunakan frame blocking sehingga menjadi banyak frame dengan durasi masing-masing frame sama. Framing dilakukan setiap 30 ms dengan overlapping sebesar 10 ms.

c. Proses Windowing

Setelah dilakukan frame blocking dilakukan proses windowing dengan menggunakan Hamming Window. Proses windowing dilakukan untuk mengurangi discontinuitas sinyal pada awal dan akhir frame akibat dari dilakukannya frame blocking.

d. Fast Fourier Transform (FFT)

Proses FFT dilakukan setelah proses windowing, pada proses ini sinyal suara hasil perekaman yang berdomain waktu diubah menjadi domain frekuensi.

e. Mel-Frequency Wrapping

Sinyal suara setelah mengalami FFT selanjutnya dilakukan filter menggunakan *filterbank* sehingga diketahui ukuran dari energi dan *frequency band* tertentu dalam sinyal suara.

f. Cepstrum

Sinyal suara kemudian mengalami serangkaian proses sehingga didapatkan informasi atau ciri dari suatu sinyal suara.

4. Pattern Matching

Proses Pattern Matching atau pencocokan pola dilakukan dengan menggunakan Jaringan Saraf Tiruan Linear Vector Quantization (LVQ). Matriks proyeksi yang telah dihasilkan oleh LDA dimasukkan ke dalam layer input pada jaringan JST LVQ. Kemudian dilakukan pembobotan serta pencocokan dengan data dari database yang berupa bobot dari masing – masing kelompok suara dan memberikan hasil apakah data input cocok dengan salah satu kelompok data di dalam database.

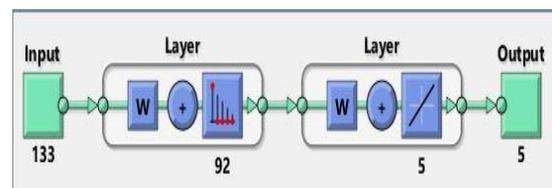
5. Gerak Robot

Gerak robot pada Aplikasi Speech Recognition Bahasa Indonesia dengan Jaringan Saraf Tiruan Linear Vector Quantization untuk Pengendalian Gerak Robot hanya berupa gerakan maju, mundur, kanan, kiri dan berhenti. Setelah sinyal suara mengalami berbagai proses dari input ucapan hingga pattern matching, apabila hasil input cocok dengan salah satu data pada database, maka robot bergerak sesuai dengan hasil output dari pattern matching. Apabila hasil input tidak cocok, maka robot tidak akan melakukan perubahan gerak.

4. HASIL EKSPERIMEN

Data pelatihan yang digunakan untuk melakukan eksperimen berupa lima orang laki-laki dewasa dan masing-masing mengucapkan kata kanan, kiri, maju, mundur dan berhenti sebanyak dua kali. Sehingga didapatkan 50 data suara sebagai data pelatihan jaringan LVQ. Untuk data uji berupa lima orang yang berbeda dari data pelatihan.

Jaringan LVQ yang digunakan memiliki 133 input, 1 *hidden layer* yang terdiri dari 92 *hidden neuron* dan memiliki 5 output. Jumlah *hidden neuron* ditentukan melalui rumus $2/3 * (\text{jumlah input} + \text{jumlah output})$. Jaringan ditunjukkan pada gambar 8.



Gambar 8 Jaringan LVQ yang Digunakan

Pada penelitian ini digunakan *learning rate* sebesar 0,01 dan menggunakan berbagai *max epoch* pada saat pelatihan sebesar 10000. Hasil dari pengujian menggunakan data latih dan data uji pada masing-masing perintah adalah sebagai berikut:

Tabel 1. Pengujian

No	Instruksi yang diucapkan	% keberhasilan dengan data latih	% keberhasilan dengan data uji
1	Kanan	70%	40%
2	Kiri	40%	40%
3	Maju	70%	40%
4	Mundur	100%	40%
5	Berhenti	80%	40%
Total		72%	40%

5. KESIMPULAN

Berdasarkan pengujian pada Tabel 1 diambil beberapa kesimpulan sebagai berikut:

1. Kata yang paling mudah dikenali adalah “mundur” dengan tingkat presetase pengenalan dengan data latih adalah 100% dan pengenalan dengan data uji adalah 40%
2. Rata – rata tingkat keberhasilan pengenalan data latih adalah sebesar 72%
3. Rata – rata tingkat keberhasilan pengenalan data uji adalah sebesar 40%

Saran

Untuk penelitian selanjutnya dapat merubah variabel seperti jumlah *hidden layer*, *hidden neuron*,

learning rate, dan *max epoch* sehingga dapat ditemukan akurasi yang lebih tinggi.

6. DAFTAR PUSTAKA

- [1] Molgaard, Lasse L., & Kasper W Jorgensen, 2005, *Speaker Recognition*, Special Course IMM•DTU.
- [2] Putra, D. dan Resmawan, A. 2011. “*Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW*”. Tugas Akhir Universitas Udayana. Bali.
- [3] Setiawan, Angga, *Aplikasi Pengenalan Ucapan Dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) Untuk Mengoperasikan Kursor Komputer*. Tugas Akhir Universitas Diponegoro. Semarang
- [4] Putra, Darma., 2010, *Pengolahan Citra Digital*, Yogyakarta: C.V Andi Offset.
- [5] QuickBooks Docstoc. (14 Oktober 2011). *LVQ (Learning Vector Quantization)*. Diperoleh 21 Maret 2014, dari <http://www.docstoc.com/docs/99267042/Learning-Vector-Quantization---DOC>
- [6] Doddington, G.R., 1989, *Phonetically Sensitive Discriminants for Improved Speech Recognition*, Proc. ICASSP, pp. 556-559, Glasgow, Scotland
- [7] Santoso, T. B. & Huda, *Pemfilteran Sinyal Wicara*. Dalam: *Modul 4 Praktikum Pengolahan Informasi Wicara* hal: 22-35.
- [8] Sigmund, M, 2003, *Voice Recognition by Computer*, Tectum Verlag DE.
- [9] Rianto, J, 2011, *Perangkat Lunak Pengenalan Suara (Voice Recognition) Untuk Absensi Karyawan Dengan Mrnggunakan Metode Dynamic Time Warping (DTW)*, Bandung: UNIKOM.