***PENGONTROLAN KECEPATAN MOTOR DC BERBASIS PENGENALAN SUARA MENGGUNAKAN METODE LINIER PREDICTIVE CODING (LPC) DAN ADAPTIVE NEURO FUZZY INFERENCE SYSTEMS (ANFIS)***

Muhammad Akil1,2,Ingrid Nurtanio1, Rhiza Samsoe’oed Sadjad 1

1 Jurusan Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin, Makassar

2 Jurusan Teknik Mekatronika, Politeknik Bosowa, Makassar

e-mail: akilmuhammad1989@yahoo.co.id, *ingrid @unhas.ac.id*, [*rhiza@unhas.ac.id*](mailto:rhiza@unhas.ac.id).

Abstrak **– Pengenalan suara merupakan objek penelitian yang menarik, disamping dapat digunakan sebagai fungsi pengenalan suara juga dapat digunakan untuk pengontrolan. Suara yang ingin digunakan sebagai perintah pada pengontrolan terlebih dahulu harus diekstrak untuk menghasilkan ciri tertentu. Terdapat beberapa metode ekstraksi ciri suara namun dalam penelitian ini digunakan *Linier Predictive Coding (LPC).* Metode LPC digunakan untuk menghasilkan ciri berdasarkan inputan perintah suara sebanyak 5 jenis kata seperti ”nyala”, ”mati”, ”lambat”, ”sedang”, dan ”cepat”. Pengambilan database hasil ekstraksi ciri suara dilakukan dengan melakukan rekaman secara berulang sebanyak 5 kali dari masing-masing instruksi suara. Database tersebut kemudian dilakukan proses training menggunakan ANFIS dengan jumlah iterasi sebesar 50 dan menghasilkan kesalahan training sebesar 0,00011963 . Hasil dari training data awal menunjukkan bahwa antara plotting data training dengan keluaran ANFIS menunjukkan nilai yang berimpitan, sehingga ANFIS dapat memetakan nilai masukan terhadap keluaran dengan baik. Hasil uji coba data training dengan memasukkan perintah suara menunjukkan bahwa keluaran ANFIS sesuai dengan jenis instruksi suara yang diberikan. Keluaran ini dapat dijadikan sebagai masukan rangkaian kontroller Arduino melalui komunikasi serial, sehingga masukan dapat diproses oleh rangkaian kontrol untuk mengeluarkan sinyal PWM ke rangkaian driver motor DC. Berdasarkan pengujian alat rangkaian kontrol dapat mengeluarkan nilai PWM sesuai dengan masukan perintah suara.**

***Kata Kunci:*** *Pengontrolan, Pengenalan Suara, LPC, ANFIS, PWM.*

1. **PENDAHULUAN**

Suara manusia merupakan suatu karunia yang sangat berharga, dengan adanya suara seseorang mudah berkomunikasi dengan orang lain sehingga proses diskusi atau bertukar pikiran dapat terpenuhi. Suara tersebut memiliki keunikan dengan nada yang berbeda-beda. Secara umum, proses terbentuknya suara berasal dari pembentukan aliran udara dari paru-paru kemudian termodulasi sehingga mengeluarkan bunyi yang spesifik[1]. Dalam bidang biometrika, perbedaan bunyi atau suara dapat dijadikan sebagai referensi untuk mengidentifikasi seseorang berdasarkan ciri-ciri fisiologis dan tingkah laku yang dimiliki individu tersebut[2].

Sinyal suara yang keluar dari tenggorokan belum dapat dijadikan sebagai masukan dalam sistem control, sebab sinyal tersebut masih berbentuk gelombang audio sesuai besar kecilnya tingkatan nada. Sehingga, perlu dilakukan pembelajaran terhadap fitur atau ciri dari suatu sinyal. Ada beberapa metode dalam melakukan ekstraksi ciri sinyal suara diantaranya adalah menggunakan metode *LPC* *(Linear Predictive Coding)* [2,3,5,7,8] dan *MFCC* *(Mel-Frequency Cepstrum Coeffisients)*[8] kemudian dilanjutkan dengan beberapa metode pengenalan pola suara hasil ekstraksi diantaranya *Adaptive Neuro-Fuzzy Inference Systems* (*ANFIS*)[6], *Artificial Neural Network (ANN)*[3] dan lain-lain.

Dengan berkembangnya teknologi sekarang ini, pengenalan suara dapat dijadikan sebagai ciri yang dapat dibentuk dengan menggunakan metode kecerdasan buatan agar menghasilkan keluaran yang dapat digunakan untuk mengendalikan sesuatu misalnya kendali gerakan robot lengan secara *real-time* yang menggunakan *Neuro Fuzzy* [2], mengendalikan pergerakan robot mobil dengan metode ANN *(Artificial Neural Network)* [3], dan sebagai isyarat untuk memainkan *music player* dengan metode Jaringan Syaraf Tiruan Propagasi Balik [6].

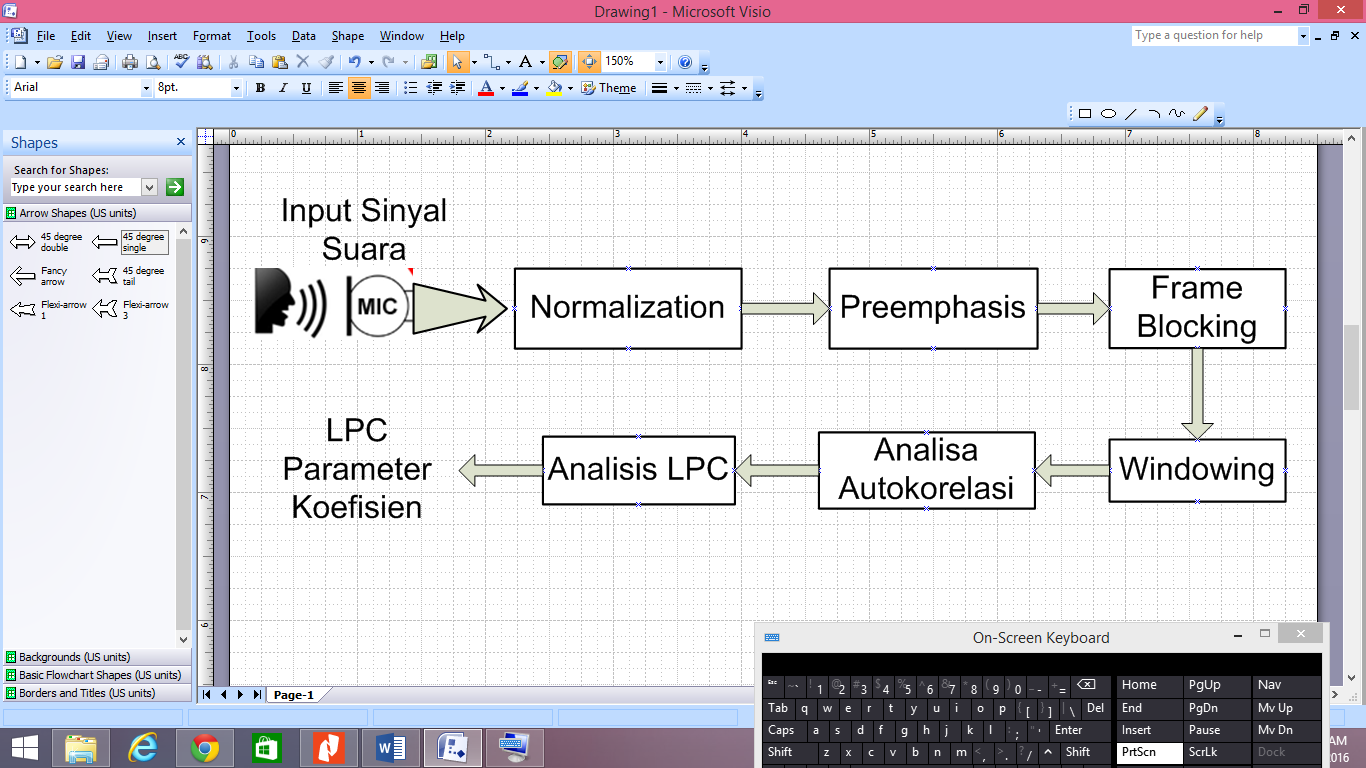
Berdasarkan penelitian-penelitian sebelumnya, peneliti dapat menyimpulkan bahwa penerapan sistem pengenalan suara masih terfokus pada pengontrolan *On-Off* saja, sehingga kami ingin melakukan penelitian tentang Pengontrolan Kecepatan Motor DC Berbasis Pengenalan Suara Menggunakan Metode *Linear Predictive Coding* (*LPC*) sebagai ekstraksi fitur sinyal suara dan dilanjutkan dengan metode belajar menggunakan *Adaptive Neuro-Fuzzy Inference Systems* (*ANFIS*).

Tujuan dari penulisan ini adalah (1) Untuk mengontrol kecepatan motor DC menggunakan pengenalan suara dengan metode *LPC* dan *ANFIS, (2)* Untuk mengetahui perbedaan ciri hasil ekstraksi pengenalan suara menggunakan metode *Linear Predictive Coding* (*LPC*) dan (3) Untuk mendapatkan pola pengenalan suara dengan menggunakan *Adaptive Neuro Fuzzy Inference Systems* (*ANFIS*). Tulisan ini memuat tentang metode pengenalan suara menggunakan *Linier Predictive Coding (LPC)* dan *ANFIS* sebagai pengenalan pola ciri suara, juga dibahas tentang pengendalian kecepatan motor DC menggunakan teknik *Pulse Width Modulation (PWM)*.

1. **TINJAUAN PUSTAKA**
2. **Ekstraksi Ciri Menggunakan *LPC* *(Linear Predictive Coding)***

Ekstraksi fitur dalam *Automatic Speech Recognition* adalah perhitungan urutan vektor-vektor fitur yang menyediakan representasi sinyal suara yang diberikan. Biasanya hal ini dilakukan dalam tiga tahapan [8]. Tahapan pertama disebut analisis suara yang melakukan analisis *spectra* dari sinyal suara dan menghasilkan fitur mentah yang menggambarkan *spectrum* interval ucapan pendek secara keseluruhan. Tahapan kedua mengkompile vector fitur yang diperluas yang terdiri dari static dan dinamik fitur. Tahapan terakhir mengubah vector fitur yang telah diperluas kedalam beberapa vector lengkap dan kuat yang akan disuplai ke pengenal.

Metode *LPC* digunakan untuk mengekstraksi fitur dari suara asli dengan kualitas yang baik dan efsien untuk digunakan dalam perhitungan [1]. *Linear Predictive Coding* *(LPC)* melakukan analisis dengan cara memperkirakan formant, memisahkan formant dari sinyal, yang dinamakan proses inverse filtering, lalu mengestimasi intensitas dan frekuensi dari sinyal wicara yang tersisa, yang disebut residu. Karena sinyal wicara bervariasi seiring waktu, estimasi tersebut dilakukan untuk setiap potongan kecil dari sinyal, yang dinamakan frame. Prosedur untuk mendapatkan koefisien *LPC* diperlihatkan pada gambar 1 [9].



Gambar 1. Blok diagram metode LPC

Langkah-langkah perhitungan *LPC* adalah sebagai berikut [5]:

1. *Pre-Emphasis*

Pertama-tama analisis dari sampel ucapan dilakukan dengan melewatkan suara tersebut melalui sebuah filter dengan tujuan untuk mendapatkan bentuk spectral frekuensi sinyal wicara yang lebih halus dan membuatnya kurang rentang terhadap efek presisi. Koefisien dari filter tersebut harus diantara 0.9 dan 1. Dimana bentuk spectral yang relative bernilai tinggi untuk daerah rendah dan cenderung turun secara tajam untuk daerah frekuensi di atas 2000 H. *Filter preemphasis* didasari oleh hubungan input/output dalam domain waktu yang dinyatakan dalam persamaan beda seperti berikut [1]:

(1)

Di mana a merupakan konstanta *filter pre-emphasis*, biasanya bernilai 0,9 < a < 1,0.

1. *Frame Blocking*

Setelah tahap *preemphasis*, hasil sinyal suara dibagi menjadi beberapa *frame* yang terdiri dari M sampel setiap 20 sampai 40 detik. Pada proses ini, sinyal suara disegmentasi menjadi beberapa *frame* yang saling tumpang tindih (*overlap*). Hal ini dilakukan agar tidak ada sedikitpun sinyal yang hilang (*deletion*). Proses ini akan berlanjut sampai seluruh sinyal sudah masuk ke dalam satu atau lebih *frame*.

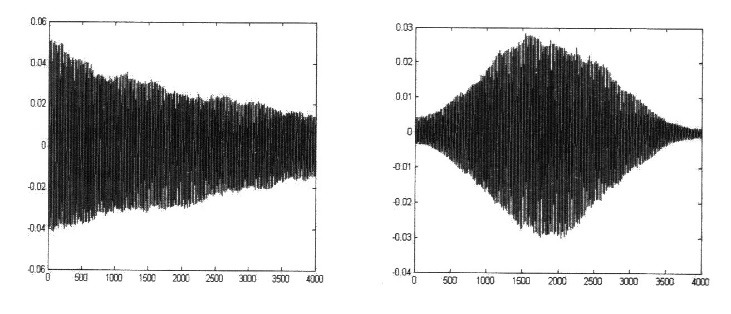
1. *Windowing*

Sinyal analog yang sudah diubah menjadi sinyal digital dibaca *frame* demi *frame* dan pada setiap *frame*-nya dilakukan *windowing* dengan fungsi *window* tertentu. Proses *windowing* bertujuan untuk meminimalisasi ketidak berlanjutan sinyal pada awal dan akhir setiap *frame*. Jika kita defenisikan *window* sebagai , di mana adalah jumlah sampel pada setiap *frame*-nya, maka hasil dari *windowing* adalah sinyal [6]:

(2)

Dimana biasanya menggunakan *window hamming* yang memiliki bentuk:

(3)



|  |  |
| --- | --- |
| 1. Sebelum *windowing* | 1. Setelah *windowing* |
| Gambar 2. Ilustrasi proses windowing [6] | |

1. *Analisis* *Autokorelasi*

Analisis *autokorelasi* terhadap setiap *frame* hasil *windowing* dengan persamaan [1]:

(4)

Dimana setiap adalah order dari *LPC*. Oder *LPC* yang sering digunakan adalah berkisar antara 8 hingga 16.

1. Analisis *LPC*

Langkah selanjutnya adalah analisis *LPC*, yang mana mengkoversikan setiap frame dari p+1 *autokorelasi* menjadi himpunan “Parameter *LPC*”.

untuk . Nantinya himpunan ini menjadi koefisien *LPC*, ataupun transformasi *LPC* yang lain. Metode formal untuk mengubah dari koefisien *autokorelasi* menjadi himpunan parameter *LPC* disebut sebagai metode *Durbin* dan dapat dibentuk mejadi algoritma sebagai berikut [1]:

(5)

(6)

(7)

(8)

(9)

Dengan adalah hasil *autokorelasi*, adalah *error*, adalah koefisien pantulan, adalah koefisien prediski untuk .

1. Konversi Parameter *LPC* menjadi Koefisien *LPC*

Mengubah parameter *LPC* ke koefisien *cepstral* untuk mendapatkan kinerja yang lebih baik dan tahan terhadap *noise*, yaitu dengan persamaan [1]:

(10)

(11)

Koefisien *cepstral* ini adalah koefisien dari representasi ada *spectrum logaritmis*.

1. ***ANFIS (Adaptive Neuro-Fuzzy Inference Systems)***

*ANFIS* adalah sebuah metode untuk menala aturan dasar sistem *Fuzzy* dengan sebuah algoritma pembelajaran berbasis kumpulan data *training* yang berada dalam *Artificial Neural Network* [10][2]. *ANFIS* memiliki keuntungan lebih cepat dan akurat secara signifikan dibandingkan banyak sistem jaringan syaraf yang lainnya [10]. *ANFIS* yang berhubungan dengan model *Fuzzy* type Sugeno memiliki 2 input dan output tunggal seperti diperlihatkan pada gambar 5. Secara spesifik *ANFIS* hanya mendukung sistem type Sugeno. Berdasarkan sistem ini kita dapat [11]:

* Memasukkan data (*training*, ujicoba dan pengecekan)
* Menghasilkan model *FIS* awal atau memasukkan model *FIS* awal.

Aturan set urutan pertama sistem *Fuzzy* Sugeno sebagai berikut:

Aturan :

F:\KULIAH S2 ELEKTRO UNHAS\Semester 3\Gambar\Drawing7.emf

Gambar 3. Struktur ANFIS

Dalam system *ANFIS* sebagaimana gambar di atas, terdapat lima lapisan proses dengan fungsi dan persamaan masing-masing lapisannya dijelaskan sebagai berikut [1]:

***Layer* 1: *Fuzzyvication***

Keluaran dari node i pada layer 1 dinotasikan sebagai . setiap node pada layer 1 bersifat adaptif dengan output:

(12)

(13)

Keterangan:

* dan adalah nilai-nilai input node
* dan adalah himpunan *fuzzy*.

Jadi masing-masing node pada layer 1 berfungsi membangkitkan derajat keanggotaan.

***Layer* 2: *Product***

Setiap node pada layer ini berfungsi untuk menghitung kekuatan aktivasi *(firing strength)* pada setiap *rule* sebagai *product* dari semua input yang masuk atau sebagai operator *t-norm* *(triangular-norm)*:

(14)

Sehingga

(15)

(16)

Output pada lapisan ini bertindak sebagai fungsi bobot *(weight)*.

***Layer* 3: Normalisasi**

Setiap node pada lapisan ini bersifat nonadaptif, fungsinya hanya untuk menghitung rasio antara *firing strength* pada rule ke- terhadap total *firing strength* dari semua rule:

(17)

***Layer* 4: *Defuzzyfication***

Pada lapisan ini setiap node bersifat adaptif dengan fungsi:

(18)

Keterangan:

* adalah output pada layer 3
* adalah himpunan parameter pada *fuzzy* model Sugeno orde pertama.

***Layer* 5: Total Output**

Pada tahapan ini seluruh sinyal yang masuk (keluaran pada layer 4) akan dijumlahkan sehingga diberikan satu node tunggal yang dilambangkan dengan .

(19)

Seluruh lapisan tersebut akan membangun suatu *adaptive-networks* yang fungsinya sama dengan *fuzzy* model Sugeno orde pertama.

1. ***Pulse Width Modulation (PWM)***

Salah satu teknik pengontrolan kecepatan motor DC adalah dengan menggunakan *PWM (Pulse Width Modulation).* Sinyal PWM dibangkitkan dengan menggunakan fitur timer pada arduino atau mikrokontroller lainnya. PWM merupakan teknik yang sangat efisien dalam menyediakan supplay tegangan antara *on* dan *off* secara penuh [15]. Pada mikrokontroller terdapat 3 jenis timer yang dapat digunakan untuk membangkitkan nilai PWM, masing-masing timer terhubung dengan Pin tertentu pada arduino (Pin yang suppoort PWM).

Pemberian nilai PWM dapat dilakukan dengan memberikan nilai analog pada pin yang telah diberikan tanda khusus untuk mengeluarkan sinyal PWM. Nilai tersebut dari 0 sampai 255 karena register PWM yang digunakan adalah 8 bit. Nilai 0 mengisyaratkan bahwa tidak ada tegangan yang dikeluarkan oleh Pin PWM Arduino, jika nilai tersebut dinaikkan maka Pin PWM akan mengeluarkan tegangan rata-rata hingga tegangan maksimum. Agar lebih jelasnya dapat dilihat pada gambar 4.

C:\Users\Tosh\Desktop\Graphic1.emf

Gambar 4. Keluaran sinyal PWM

1. **METODE**
2. **Blok Diagram**

Blok diagram system secara keseluruhan dapat dilihat pada gambar 5.

F:\KULIAH S2 ELEKTRO UNHAS\Semester 3\Gambar\Drawing6.emf

Gambar 5. Blok Diagram System Pengontrolan Kecepatan Motor DC Berbasis Pengenalan Suara

Berdasarkan blok diagram, tahapan pertama yang dilakukan adalah melakukan rekaman suara dengan menggunakan microphone. Suara yang telah direkam kemudian dilakukan ekstraksi ciri dengan menggunakan metode LPC, hasil ekstraksi ciri tersebut dibuatkan database agar dapat dijadikan input pada metode pengenalan suara. Keluaran pengenalan suara akan dihubungkan ke bagian kontroller melalui jalur komunikasi serial, pada bagian ini controller akan mengidentifikasi jenis inputan yang masuk dan akan diproses sehingga menghasilkan keluaran berupa sinyal PWM. Sinyal PWM kemudian diumpankan ke driver motor DC sehingga motor DC dapat berputar dengan kecepatan sesuai nilai PWM.

1. **Flowchart**

Secara sistematis urutan atau langkah-langkah yang dilakukan dalam penelitian dapat dilihat pada gambar 6. Proses perekaman hingga menghasilkan ciri menggunakan LPC. Sedangkan metode pengenalan pola atau ciri digunakan ANFIS. Pada tahapan ini data dilatih dan diujicoba sampai menghasilkan kesalahan minimum.

F:\KULIAH S2 ELEKTRO UNHAS\Semester 3\Gambar\Flowchart 1.emf

Gambar 6. Flowchart System

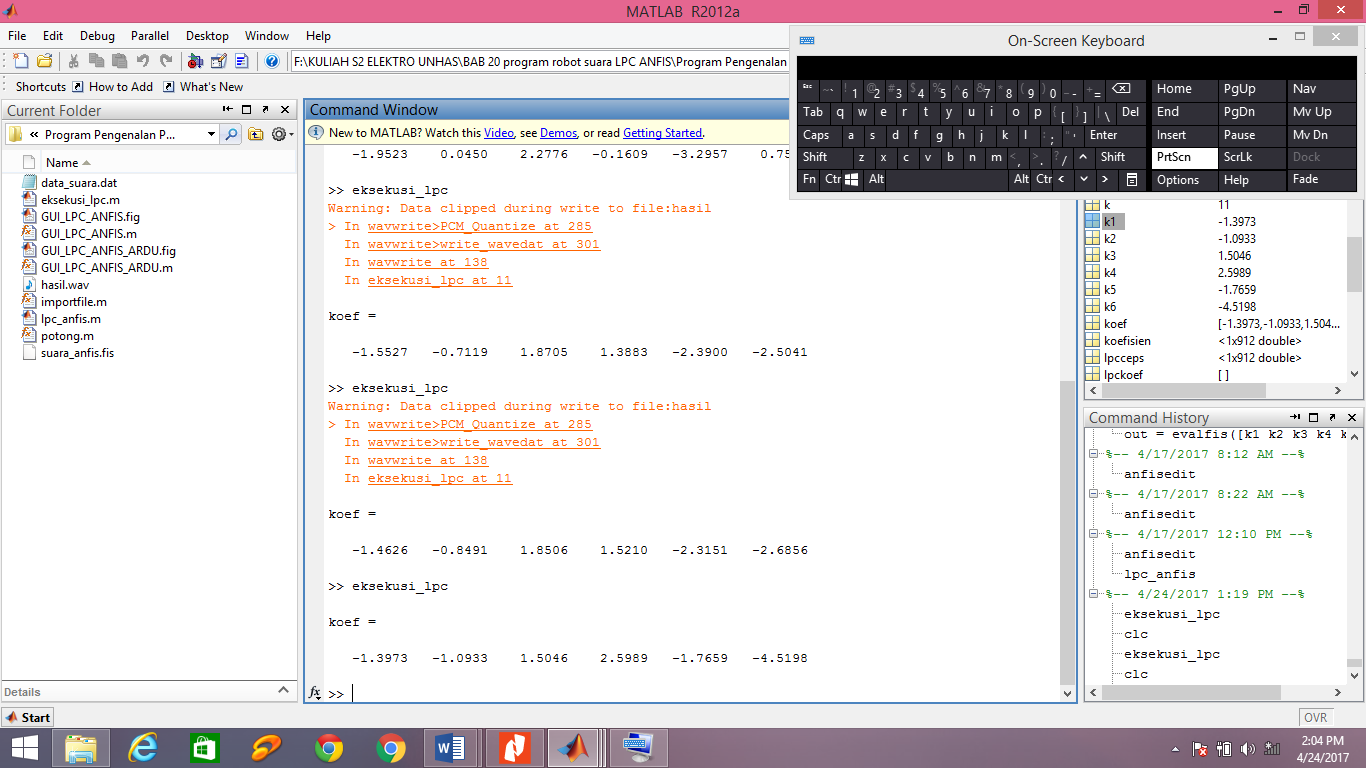
1. **HASIL DAN PEMBAHASAN**

## **Proses Perekaman Suara dan Ekstraksi Ciri LPC**

Tahapan awal yang dilakukan adalah dengan merekam sinyal suara yang akan dijadikan sebagai suara perintah. Terdapat 5 jenis perintah suara yaitu “ Nyala”, “Lambat”, “Sedang”, “Cepat”, dan “Mati”. Masing–masing perintah suara akan dilakukan proses perekaman secara berulang-ulang sebanyak 5 kali sehingga total rekaman sebanyak 15 buah.

Untuk menghasilkan ciri suara maka sinyal suara yang telah direkam akan dilakukan beberapa proses seperti yang telah dibahas pada bab sebelumnya yaitu tahapan-tahapan perhitungan metode LPC. Jumlah ciri yang dihasilkan dari setiap instruksi suara yang telah direkam adalah sebanyak 6 ciri.

Berikut adalah hasil ekstraksi ciri suara “Lambat” dengan menggunakan metode LPC.



Gambar 7. Hasil ekstaksi ciri suara “Lambat” dengan 6 koefisien

F:\KULIAH S2 ELEKTRO UNHAS\Semester 3\Gambar\Sinyal_Suara.emf

Gambar 8. Sinyal Suara Instruksi “Lambat”

F:\KULIAH S2 ELEKTRO UNHAS\Semester 3\Gambar\Suara_Pre_Emphasiis.emf

Gambar 9. Sinyal Suara Pre-Emphasis

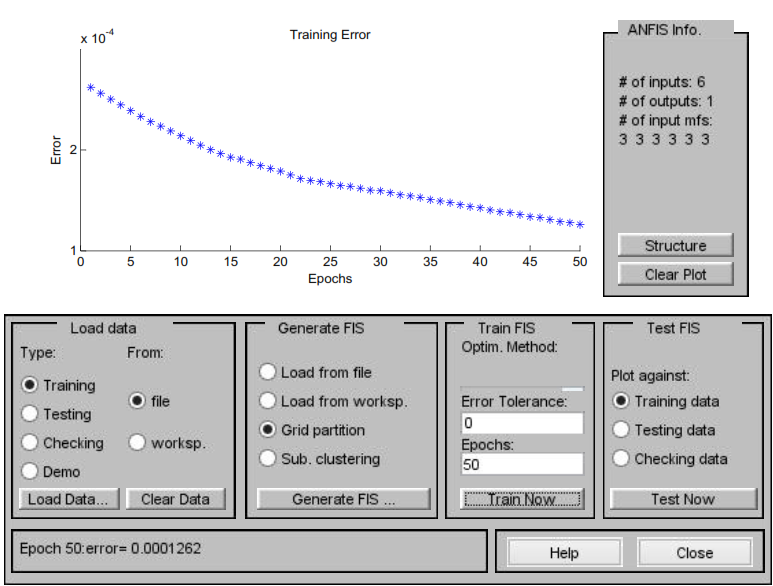
Hasil rekaman ciri suara yang telah dilakukan secara berulang disimpan dalam sebuah database. Masing-masing instruksi suara diberikan nilai target yang berbeda-beda agar ANFIS dapat memetakan pola berdasarkan sinyal suara inputan. Sinyal suara “Nyala” diberikan nilai taget “1”, “Lambat” diberikan nilai target “2”, “Sedang” diberikan nilai target “3”, “Cepat” diberikan nilai target “4” dan “Mati” dengan target “5”.

Tabel 1. Database ciri sinyal suara dan targetnya

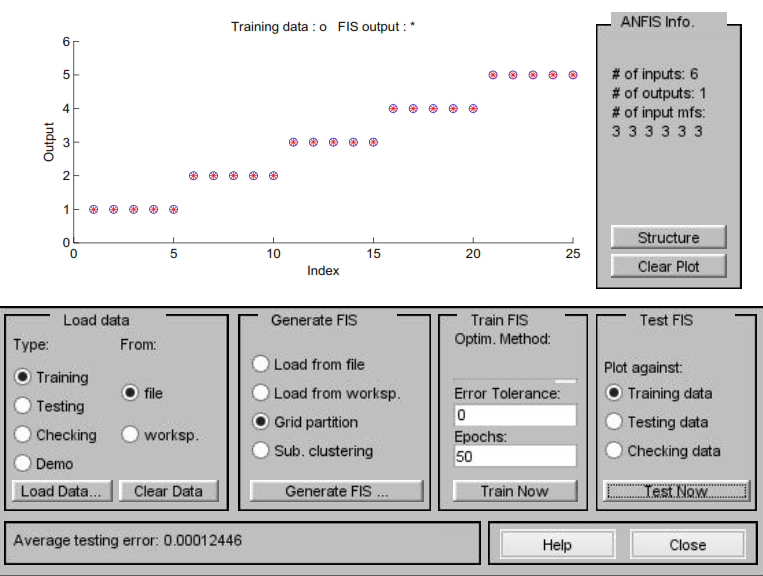
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ciri 1 | Ciri 2 | Ciri 3 | Ciri 4 | Ciri 5 | Ciri 6 | Target |
| -1.7677 | -0.1021 | 1.6886 | 0.5121 | -1.8426 | -1.7719 | 1 |
| -1.6023 | -0.4476 | 1.9782 | 0.6927 | -2.3422 | -1.4263 | 1 |
| -1.4706 | -0.6969 | 2.1161 | 0.4289 | -2.1715 | -0.8736 | 1 |
| -1.4669 | -0.3786 | 1.4647 | 0.6932 | -1.2341 | -1.6367 | 1 |
| -1.8475 | -0.5192 | 3.4149 | 0.1234 | -5.5467 | -0.4700 | 1 |
| -1.9268 | -0.0228 | 2.2992 | 0.1266 | -3.5794 | 0.2417 | 2 |
| -1.9403 | -0.1134 | 2.6546 | -0.1136 | -3.9561 | 0.4861 | 2 |
| -1.9031 | 0.0491 | 1.9926 | -0.1322 | -2.4741 | 0.6009 | 2 |
| -1.4234 | -0.4422 | 1.0431 | 1.1871 | -1.0936 | -1.6250 | 2 |
| -1.6134 | -0.1516 | 1.3231 | 0.6505 | -1.7152 | -0.8273 | 2 |
| -1.9292 | 0.4036 | 1.5264 | -0.7654 | -1.2847 | 1.4278 | 3 |
| -1.9142 | 0.2327 | 1.9155 | -0.6792 | -2.1662 | 1.3777 | 3 |
| -1.1542 | -1.0435 | 0.7742 | 1.3284 | 0.2779 | -0.9242 | 3 |
| -1.7457 | -0.2297 | 1.6724 | 0.3809 | -1.4830 | -0.4979 | 3 |
| -1.7362 | 0.0500 | 1.3486 | 0.0621 | -1.4437 | 0.1584 | 3 |
| -2.0382 | 0.2790 | 1.9352 | 0.1780 | -3.0410 | -0.0875 | 4 |
| -1.6658 | 0.0152 | 1.5230 | -0.0143 | -1.7191 | 0.2726 | 4 |
| -1.4411 | -0.3502 | 1.2465 | 0.8021 | -1.4006 | -1.1076 | 4 |
| -1.3236 | -0.5604 | 0.8890 | 1.2725 | -0.8888 | -1.4343 | 4 |
| -1.9617 | 0.0151 | 2.5053 | -0.3193 | -3.5598 | 0.9157 | 4 |
| -0.9764 | 0.0329 | -0.2511 | -0.4143 | 0.1425 | 0.6363 | 5 |
| -0.9651 | -0.1267 | 0.0767 | -0.6665 | 0.2460 | 0.7129 | 5 |
| -0.5480 | -0.2493 | -0.6381 | -0.3819 | 0.1689 | 0.5791 | 5 |
| -1.3828 | 0.3527 | 0.0325 | -0.5897 | 0.5769 | 0.5592 | 5 |
| -1.3811 | 0.2215 | 0.1857 | -0.3169 | 0.0431 | 0.4620 | 5 |

## **Training data dengan ANFIS**

ANFIS digunakan untuk melakukan training database bedasarkan target yang telah dibuat. Jenis FIS yang digunakan adalah sugeno. Setelah memasukkan file database yang akan ditraining, secara otomatis ANFIS editor mendefinisikan bahwa terdapat 6 jenis input dan 1 output yang digunakan sebagai target. Jumlah keanggotaan masing-masing input diberikan 3 kondisi sehingga secara keseluruhan terdapat 18 jumlah anggota fungsi. Sebelum melakukan training data maka tentukan terlebih dahulu metode taining FIS dan jumlah *epoch*, metode optimasi training FIS yang digunakan adalah *Hybrid* sedangkan jumlah epoch sebesar 50. Epoch ini sangat menentukan apakah data yang ditraining memiliki error yang sangat kecil (mendekati nol).



Gambar 10. Training database dengan jumlah epoch 50



Gambar 11. Hasil uji coba data training

Berdasarkan gambar 10 dapat dilihat bahwa jumlah rata-rata *error* yang dihasilkan setelah dilakukan training data awal dengan jumlah epoch 50 adalah 0,00011963. Selanjutnya data yang sudah dilatih tadi akan diujicoba dan hasilnya dapat dilihat pada gambar 11. Hasil plot antara data training dan *output* FIS diperlihatkan pada gambar 11.

## **Pengujian System**

1. **Ujicoba pengenalan suara**

Setelah melakukan pelatihan dan tes ujicoba pada database dimana *error* yang muncul sangat kecil maka tahapan selanjutnya adalah dengan melakukan ujicoba menggunakan data baru dengan jenis perintah suara yang sama. Ujicoba dilakukan dengan memasukkan perintah suara baru baik dari responden dalam database maupun yang diluar database.

Tabel 2. Ujicoba responden dalam database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Jenis Perintah Suara** | ***Output Program*** | ***Output***  ***Terdeteksi*** | **Target** | ***Error*** |
| 1 | Nyala | 0 – 1,25 | 1,0783 | 1 | -0,783 |
| 2 | Lambat | 1,25 – 2,25 | 1,5251 | 2 | 0,4749 |
| 3 | Sedang | 2,25 – 3,25 | 2,6547 | 3 | 0,3453 |
| 4 | Cepat | 3,25 – 4,25 | 3,3082 | 4 | 0,6918 |
| 5 | Mati | 4,25 – 5,25 | 5,0478 | 5 | -0,478 |
|  |  |  | Total Error | | 0,251 |
|  |  |  | Rata-rata Error | | 0,0502 |

Tabel 3. Ujicoba responden diluar database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Jenis Perintah Suara** | ***Output Program*** | ***Output***  ***Terdeteksi*** | **Target** | ***Error*** |
| 1 | Nyala | 0 – 1,25 | 0,6025 | 1 | 0,3975 |
| 2 | Lambat | 1,25 – 2,25 | 2,0617 | 2 | 0,0617 |
| 3 | Sedang | 2,25 – 3,25 | 3,1359 | 3 | -0,1359 |
| 4 | Cepat | 3,25 – 4,25 | 3,6391 | 4 | 0,3609 |
| 5 | Mati | 4,25 – 5,25 | 4,5498 | 5 | 0,4502 |
|  |  |  | Total Error | | 1,1344 |
|  |  |  | Rata-rata Error | | 0,22688 |

1. **Ujicoba rangkaian kontrol**

Rangkaian kontrol kecepatan motor DC menggunakan arduino dengan sinyal PWM sebagai representasi cepat atau lambatnya putaran motor. Berdasarkan hasil uji coba rangkaian kontrol dengan inputan suara, sinyal PWM yang keluar dari arduino sesuai dengan masukan perintah suara. Apabila diberikan instruksi suara ”Lambat“ maka arduino akan menerima data karakter secara serial dari komputer, data tersebut akan diproses dan hasilnya akan keluar sinyal PWM dengan nilai 125 dengan duty cycle sebesar 49%.

Untuk mendapatkan nilai duty cycle pada keluaran sinyal PWM yang lainnya menggunakan rumus yang sama dan hasilnya dapat dilihat pada tabel 4.

Tabel 4. Keluaran Sinyal PWM berdasarkan inputan perintah suara

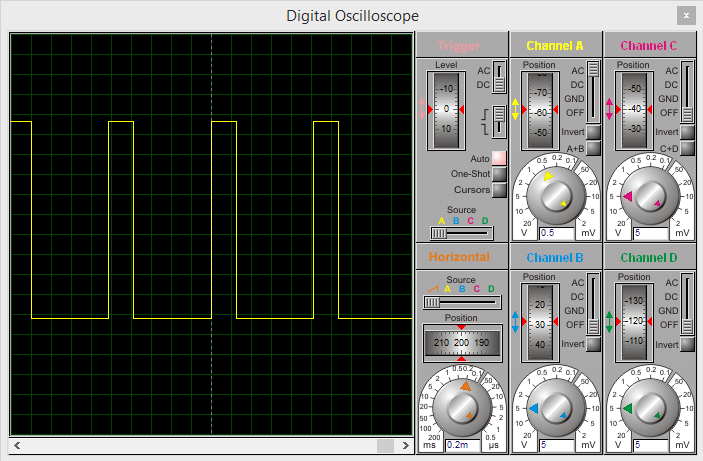
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Jenis Perintah Suara** | ***Output Program*** | **Sinyal PWM** | **Duty Cycle** |
| 1 | Nyala | 0 – 1,25 | 62,5 | 24,5% |
| 2 | Lambat | 1,25 – 2,25 | 125 | 49% |
| 3 | Sedang | 2,25 – 3,25 | 187,5 | 73,5% |
| 4 | Cepat | 3,25 – 4,25 | 250 | 98% |
| 5 | Mati | 4,25 – 5,25 | 0 | 0 |

Rangkaian kontrol kecepatan motor DC dapat dilihat pada gambar 12.

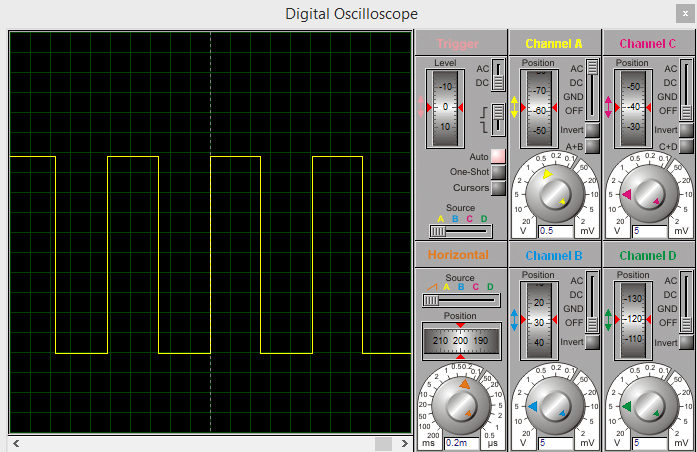
F:\KULIAH S2 ELEKTRO UNHAS\Semester 4\Skema Tesis.EMF

Gambar 12. Skema Rangkaian Kontrol

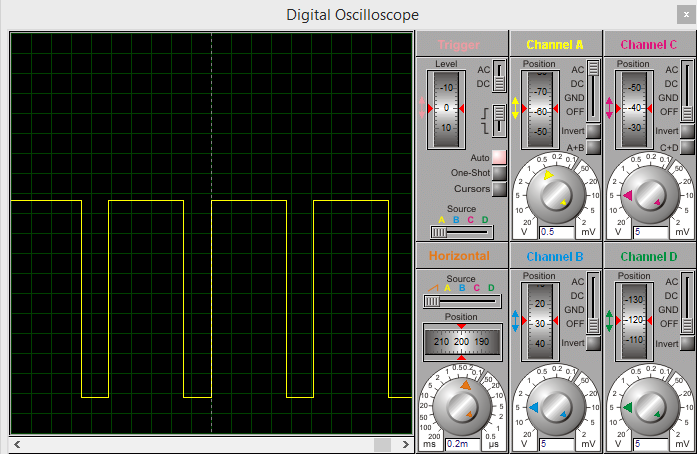
## Setelah melakukan simulasi rangkaian kontrol, didapatkan bentuk sinyal PWM untuk masing-masing keluaran seperti ditunjukkan pada gambar 13, 14, 15, 16 dan 17.



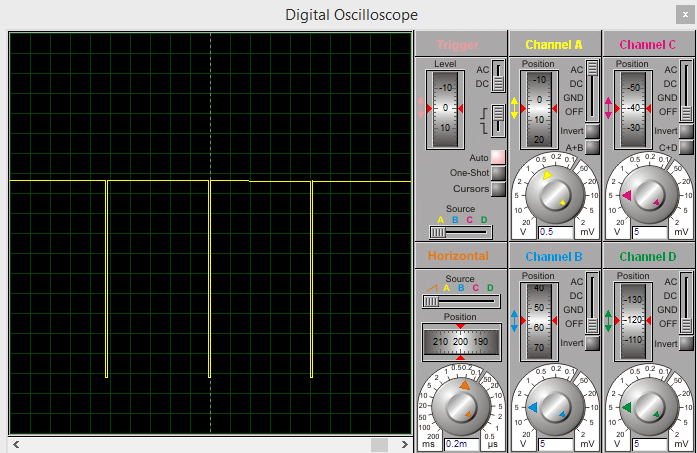
Gambar 13. Bentuk sinyal PWM dengan duty cycle 24,5%



Gambar 14. Bentuk sinyal PWM dengan duty cycle 49%



Gambar 15. Bentuk sinyal PWM dengan duty cycle 73,5%



Gambar 16. Bentuk sinyal PWM dengan duty cycle 98%

1. **KESIMPULAN**

Secara keseluruhan tujuan dari penelitian ini dapat tercapai dari hasil yang didapatkan pada pembahasan sebelumnya yaitu ekstraksi ciri dari hasil rekaman menggunakan metode LPC mengeluarkan ciri yang unik untuk masing-masing instruksi suara. Masing-masing ciri dapat dikenali dengan baik oleh ANFIS sebagai instrumen pengenalan pola suara. Dari hasil uji coba yang berasal dari responden menghasilkan error rata-rata 0,0502 sedangkan diluar responden menghasilkan error rata-rata 0,22688. Sedangkan pada uji coba rangkaian control dengan menggunakan komunikasi serial antara komputer dengan arduino, data dapat terkirim sesuai sinyal inputan yang masuk dan mengeluarkan sinyal PWM yang tepat agar dapat menjalankan motor DC dengan kecepatan berubah-ubah seiring sinyal masukan suara. Sehingga dapat disimpulkan rangkaian tersebut bekerja 100%.

**DAFTAR PUSTAKA**

[1] Sanjaya, M. 2016. Robot Cerdas Berbasis Speech Recognition menggunakan Matlab dan Arduino. Yogyakarta: Andi.

[2] Ronando, E. dan Isa, IM. 2012. Pengenalan Ucapan Kata Sebagai Pengendali Gerakan Robot Lengan Secara *Real-Time* dengan Metode *Linear Predictive Coding-Neuro Fuzzy*. Jurnal Sains dan Seni ITS Vol. 1, No. 1. PP A51-A56.

[3] S. Wijoyo, “*Speech Recognition Using Linear Predictive Coding and Artificial Neural Network for Controlling Movement of Mobile Robot*” *2011 Int. Conf. Inf. Electron. Eng.*, vol. 6, pp. 179–183, 2011.

[4] Kurniawan, P. dkk. 2012. Desain dan Analisis Sistem Aplikasi Music Player Berbasis Pengenalan Suara Menggunakan Metode Jaringan Saraf Tiruan Back Propagation (JST-BP). Tugas Akhir. Universitas Telkom.

[5] Gupta, H. dkk. 2016. *LPC and LPCC Method of Feature Extraction in Speech Recognition System. International Conference* - *Cloud System and Big Data Engineering (Confluence). IEEE Conference Publications. Page* 498-502

[6] Sanjaya, M. and Salleh. Z. 2014. Implementasi Pengenalan Pola Suara Menggunakan *Mel-Frequency Cepstrum Coefficients* (*MFCC*) dan *Adaptive Neuro-Fuzzy Inferense System* (*ANFIS*) Sebagai Kontrol Lampu Otomatis. Al-HAZEN *Jurnal of Physics* Volume 1, No.1, page 43-54

[7] Thiang and Wanto. 2010. *Speech Recognition Using LPC and HMM Applied for Controlling Movement of Mobile Robot*. *Seminar Nasional Teknologi Informasi.*

[8] Dave. N. 2013. *Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. International Journal of Advanced Research in Education & Technology (*IJARET*).* Volume 1, Issue VI.

[9] Yusnita M.A. 2011. *Malaysian English Accents Identification using LPC and Formant Analysis. IEEE International Conference on Control System, Computing and Engineering*. Page 472-476.

[10] Pandey, B. dkk. 2010. *Multilingual Speaker Recognition Using ANFIS. 2nd International Conference on Signal Processing Systems (ICSPS) IEEE*. Page V3-714 – V3-718

[11] Helmi, N. and Helmi B.Hoda. 2008. *Speech Recognition with Fuzzy Neural Network for Discrete Words. Fourth International Conference on Natural Computation IEEE. Page* *265-269.*

[12]Thiang and Saputra, H. 2005. Sistem Pengenalan Kata dengan Menggunakan *Linear Predictive Coding* dan *Nearest Neighbor Classifier.* Jurnal Teknik Elektro volume 5, Nomor 2. Page 19-24.

[13] Kumar, P. Sharma, Lakshmikantha, B.R dan Shanmukha, K. Sundar. 2011. *Real Time Control of DC Motor Drive using Speech Recognition. International Conference on Power Electronics 2010 (IICPE2010)) IEEE*. *Page 1-5*

[14] Rai, N. dan Rai, B. 2014. *An Ann Based Mobile Robot Control Through Voice Command Recognition Using Nepali Language. International Journal of Applied Control, Electrical and Electronics Engineering (IJACEEE) Vol 2, No.4.*

[15] Kumar, Ravi KS. dkk. 2015. “Microprocessor Based Closed Loop Speed Control of DC Motor Using PWM”. International Conference on Control,lnstrumentation, Communication and Computational Technologies (ICCICCT). IEEE.