

## SISTEM PENGENALAN SUARA DENGAN MENGGUNAKAN METODE TEMPLATE MATCHING UNTUK AUTOMATISASI PENGATURAN AUDIO PLAYER

Natasha Herma Devita<sup>1</sup>, Koredianto Usman<sup>2</sup>, Suryo Adhi Wibowo<sup>3</sup>

<sup>1</sup>Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

---

### Abstrak

Saat ini pengaturan audio player masih secara manual yaitu dengan mengklik fungsi-fungsi tombol yang ada pada tampilannya. Cara ini dianggap masih belum efektif dan praktis karena saat mengakses program ini harus mengabaikan pekerjaan utama. Oleh karena itu dicari suatu cara untuk dapat dengan mudah dan praktis melakukan pengaturan audio player pada komputer yaitu dengan mengimplementasikannya berdasarkan command suara. Dengan begitu diharapkan suara dapat menjadi suatu komando yang praktis untuk kendali jarak jauh.

Pada tugas akhir ini dibangun suatu sistem yang mampu mengenali suara masukan dan menganalisis ketepatan system dalam menjalankan Audio Player sesuai dengan yang diperintahkan. Apabila cocok maka suara ini akan dapat dengan praktis mengatur jalannya suatu lagu tanpa perlu menyentuhnya dengan tangan. Dalam hal ini digunakan metode template matching untuk mengenali atau mengklasifikasi pola suara input dengan pola suara yang sebelumnya telah disimpan dalam database. Sistem mengekstraksi perintah suara yang berbeda-beda menjadi dua macam ciri yaitu Zero Crossing dan Magnitude Average yang kemudian menjadi masukan untuk metode Template Matching ini.

Hasil implementasi sistem ini yaitu dapat mengenali command berdasarkan suara manusia dalam menjalankan Audio Player dengan tingkat akurasi mencapai 90 % dan hasilnya dapat ditampilkan dalam waktu komputasi tidak lebih dari 1,4 detik. Sehingga dengan begitu sistem ini cukup baik dari segi ketepatan dan kecepatan dalam melakukan pengaturan audio player secara otomatis.

Kata Kunci : template matching, audio player, zero crossing, Magnitude Average

---

### Abstract

Nowadays controlling Audio Player still manually, by clicking the button functions that exist in user interface. This method is considered not effective and partical yet because when you access this program you should ignore the main job. Therefore should be a way to be easily and practically make the adjustment on the Audio Player to implement in a computer based on voice command. So that the sound is expected to be a practice command for remote control.

In this final project will be built a system to be able recognize voice input and analyze the accuracy of the system while running in Audio Player according the command. If the command match, it will be able regulate the song practically without having to touch it by hand. In this case the Template Matching used to identify or classify the input voice pattern which is the pattern have been stored in database. The system will be extracts voice command into two different kinds of characteristics there are Zero Crossing and Magnitude Average then become the input for this method of Template Matching.

The result of the implementation of this system can be recognize a command based on the human voice in running the Audio Player with the highest accuracy rate is 90 % and the results can be displayed less then 1,4 seconds. So that the system is good enough in terms of accuracy and speed in performing on Audio Player controls automatically.

Keywords : Template Matching, Audio Player, Zero Crossing, Magnitude

---

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini komputer hampir dapat dijumpai di setiap tempat seperti kantor perusahaan, sekolah, atau bahkan rumah tangga. Perkembangan teknologi komputer yang pesat, khususnya di bidang perangkat lunak, membuat komputer menjadi semakin menjadi perangkat yang *user friendly* dan telah menjadikannya suatu kebutuhan tersendiri bagi masyarakat. Salah satu program aplikasi yang sering diakses dan menjadi kebutuhan utama adalah *audio player* karena hampir setiap orang tanpa sadar melakukan pekerjaan di depan komputer sambil mendengarkan musik. Saat ini pengaturan *audio player* masih secara manual yaitu dengan mengklik fungsi-fungsi tombol yang ada pada tampilannya. Cara ini dianggap masih belum efektif dan praktis karena saat mengakses program ini harus mengabaikan pekerjaan utama. Dalam perangkat kecil seperti ponsel telah dirancang implementasi pengaturan *audio player* dengan istilah “*shake*”, yaitu hanya dengan menggoyangkan ponsel maka lagu yang sedang dimainkan dapat diganti seketika tanpa perlu membuka aplikasi *audio player* secara langsung. Namun pada komputer justru cara “*shake*” ini tidaklah memungkinkan dengan ukuran komputer sebagai perangkat besar. Oleh karena itu dicari suatu cara untuk dapat dengan mudah dan praktis melakukan pengaturan *audio player* pada komputer yaitu dengan mengimplementasikannya berdasarkan *command* suara.

### 1.2 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

1. Merancang dan melakukan implementasi program aplikasi yang memudahkan kita dalam melakukan pengaturan *audio player* melalui pengenalan suara dengan menggunakan metode *template matching*.
2. Menganalisis performansi sistem berdasarkan parameter tingkat akurasi dan waktu komputasi.

### 1.3 Rumusan Masalah

Permasalahan yang diangkat dalam menyelesaikan tugas akhir ini adalah :

1. Bagaimana membuat suatu sistem yang dapat mengenali suara manusia dalam melakukan pengaturan *audio player*.
2. Bagaimana proses sistem dalam mengenali dan mengklasifikasi pola sinyal suara menggunakan metode *template matching*.
3. Bagaimana performansi sistem dalam kondisi tanpa *noise* dan *bernoise* ditinjau dari ketepatan sistem dalam mengenali *command* suara dan kecepatan sistem untuk menghasilkan *output*.
4. Bagaimana performansi sistem dalam mengenali dan mendeteksi perintah dari individu awam dan individu terlatih yang tidak memiliki sampel di dalam *database* sistem.

### 1.4 Batasan Masalah

Sejumlah permasalahan yang dihadapi dalam tugas akhir ini akan dibatasi ruang lingkup pembahasannya, yaitu:

1. Pengenalan pola suara *input* dari sistem menggunakan metode *template matching*.
2. Aplikasi yang dibuat menangani fungsi “*play*”, “*pause*”, “*resume*”, “*stop*”, “*next*”, dan “*previous*”.
3. Bahasa pemrograman yang digunakan adalah MATLAB 7.4.0 (R2007a).
4. Parameter yang dianalisis adalah waktu komputasi dan akurasi antara *input* sinyal suara dengan *output* dari bahasa pemrograman.
5. Format suara yang digunakan adalah \*.wav dengan frekuensi *sampling* 8000 Hz.
6. Sistem yang dihasilkan bersifat *private* hanya untuk orang tertentu.

### 1.5 Metodologi Penelitian

Pembuatan tugas akhir ini terbagi menjadi beberapa tahap pengerjaan, yang tertera sebagai berikut:

#### A. Studi teori literatur

Studi teori literatur ini yaitu mempelajari referensi yang berhubungan dengan *voice recognition*, *FFT*, *windowing*, *template matching*, dan cara menggunakan matlab untuk mendukung pembuatan aplikasi ini. Bahan – bahan untuk referensi bisa diperoleh dari buku-buku atau dari artikel – artikel di internet dan berbagai buku panduan.

#### B. Studi praktek

Studi praktek ini berkaitan dengan pembelajaran awal penggunaan teknologi sebelum benar-benar digunakan untuk implementasi aplikasi.

#### C. Pengimplementasian

Pada tahap pengimplementasian ini aplikasi sudah mulai dibuat dengan mengacu pada hasil dari tahap-tahap sebelumnya.

#### D. Uji coba dan evaluasi

Pada tahap ini tahap implementasi sudah selesai dilaksanakan dan saatnya untuk melakukan uji coba terhadap sistem yang baru dibuat. Uji coba ini dimaksudkan untuk menguji jalannya sistem apakah sudah sesuai dengan tujuan yang diinginkan. Jika terjadi kesalahan pada jalannya aplikasi agar dapat divalidasi secepatnya untuk diperbaiki.

#### E. Penyusunan laporan

Pada tahap terakhir ini dibuat suatu dokumen sebagai laporan pengerjaan tugas akhir. Didalam dokumen ini berisi segala hal yang berkaitan dengan teori pendukung dan tahap-tahap detail pembuatan aplikasi. Dokumen ini dibuat dengan harapan bisa bermanfaat untuk pihak lain yang membutuhkan.

### 1.6 Sistematika Penulisan

Tugas akhir ini disusun menurut sistematika penulisan terdiri atas lima bab yang disusun sebagai berikut:

#### **BAB I           Pendahuluan**

Berisi latar belakang masalah, tujuan penulisan, perumusan masalah, batasan masalah, metodologi penulisan dan sistematika penulisan.

## **BAB II Landasan Teori**

Berisi tentang teori yang mendukung dan mendasari penulisan tugas akhir ini, yaitu teori tentang aplikasi pengenalan suara, *windowing*, *fast fourier transform*, *templates matching*, dan teori-teori penunjang lain.

## **BAB III Perancangan dan Implementasi Perangkat Lunak**

Berisi urutan proses model perancangan tahap pengolahan awal dan proses klasifikasi dengan *template matching*.

## **BAB IV Analisis dan Pengujian Sistem**

Berisi penjelasan tentang hasil penelitian dan pembahasan mengenai hasil penelitian tersebut.

## **BAB V Kesimpulan dan Saran**

Berisi kesimpulan dari analisa yang telah dilakukan dan saran untuk pengembangan lebih lanjut.

Telkom  
University

## BAB IV

### ANALISIS DAN PENGUJIAN SISTEM

#### 4.1 Analisis Sistem

Sistem yang dibangun dalam Tugas Akhir ini bertujuan untuk mendeteksi suara manusia dalam melakukan pengaturan *audio player*. Dalam proses pendeteksian ini digunakan metode *Template Matching* yang berfungsi untuk mencocokkan matriks nada uji dengan matriks nada latih yang terdapat pada *database*. Ekstraksi ciri yang digunakan berupa ZC ( *Zero Crossing* ) dan MA ( *Magnitude Average* ) untuk mendapatkan ciri sinyal dari domain waktu dan domain frekuensi.

##### 4.1.1 Analisis Perangkat Keras dan Perangkat Lunak

Berikut adalah spesifikasi *hardware* dan *software* yang diperlukan dan digunakan dalam pengambilan data dan pembuatan program aplikasi pengenalan suara untuk pengaturan *audio player*.

1) **Hardware:**

a) *Notebook* dengan spesifikasi sebagai berikut:

- Acer Aspire 2920
- *Processor*: Intel Core 2 Duo T5550 1.83 GHz
- Sistem Operasi: Windows XP Professional
- *Memory*: 2 GB RAM
- *Harddisk*: 160 GB

b) *External Sound Card* Saffire 6 USB



**Gambar 4.1** *Sound Card* Saffire 6 USB

c) *Dynamic Cardioid Microphone Shure RS 45*



**Gambar 4.2** *Dynamic Cardioid Microphone Shure RS 45*

- d) *Microphone Logitech*
- e) TV LG 21 inch MEZ41323511
- f) *Basic Stereo Speakers SP-S105*

2) **Software:**

- a) MATLAB 7.8.0 R2007a
- b) *Software Cool Edit Pro 2.1*

#### 4.1.2 Analisis Masukan dan Keluaran

Data masukan untuk sistem ini berupa suara manusia yang langsung diakuisisi di *software* matlab 2007a dan diakses melalui GUI sistem. Sedangkan keluaran dari sistem ini adalah eksekusi nyata penggunaan *audio player* beserta waktu komputasinya. Sistem dikatakan dapat berjalan dengan baik apabila hasil deteksi atau eksekusi yang dijalankan sesuai dengan apa yang diperintahkan.

#### 4.2 Analisis Data Hasil Pengujian Sistem

Untuk mengetahui performansi sistem yang telah dirancang, maka dilakukan pengujian terhadap sistem dengan beberapa skenario utama pengujian yaitu:

1. Pengujian dan analisis pengaruh *threshold* terhadap akurasi *output* sistem.
2. Pengujian dan analisis pengaruh ukuran *frame* terhadap akurasi *output* sistem.
3. Pengujian dan analisis pengaruh noise terhadap akurasi *output* sistem.
4. Pengujian dan analisis pengujian terhadap individu awam dan individu terlatih terhadap akurasi *output* sistem.

#### 4.2.1 Pengaruh *threshold* terhadap akurasi *output* sistem.

Dalam skenario ini dilakukan pengujian lima nilai *threshold* yang digunakan dalam proses *cropping* dengan melakukan sepuluh kali percobaan untuk setiap *command*. Tujuan pemberian *threshold* itu sendiri adalah untuk memisahkan sinyal suara *input* dengan sinyal diam ataupun sinyal *noise*. Pada pengujian ini *threshold* yang digunakan berdasarkan standar deviasi sinyal *input*. *Threshold* yang akan diuji dan dianalisis adalah nilai hasil standar deviasi itu sendiri, 0.8 kali standar deviasi, 1.2 kali standar deviasi, 1.4 kali standar deviasi, dan 2 kali standar deviasi. Pengujian berada dalam kondisi ideal, artinya tidak adanya *noise* yang mempengaruhi sinyal demi menghitung performansi yang maksimal dalam penggunaan sistem ini.

##### 4.2.1.1 Pengujian Performansi Sistem dengan *Threshold* = 0.8 std

Berikut merupakan tabel hasil pengujian sistem dengan menggunakan *threshold* sebesar 0.8 kali standar deviasi sinyal *input*.

**Tabel 4.1** Tabel Akurasi dan Waktu Komputasi Sistem dengan thr = 0.8std

No.	<i>Command</i>	Akurasi (%)	Waktu komputasi rata - rata (s)
1	<i>Play</i>	60	1.18163667
2	<i>Pause</i>	100	1.126398
3	<i>Resume</i>	50	1.13899
4	<i>Stop</i>	100	1.13696
5	<i>Next</i>	100	1.231247
6	<i>Previous</i>	60	1.17345167
Rata – rata total		78.33	1.164781

##### 4.2.1.2 Pengujian Performansi Sistem dengan *Threshold* = std

Berikut merupakan tabel hasil pengujian sistem dengan menggunakan *threshold* sebesar standar deviasi sinyal *input*.

**Tabel 4.2** Tabel Akurasi dan Waktu Komputasi Sistem dengan thr = std

No.	Command	Akurasi (%)	Waktu komputasi rata - rata (s)
1	Play	60	1.17047
2	Pause	100	1.122274
3	Resume	50	1.13095
4	Stop	100	1.131729
5	Next	100	1.273917
6	Previous	100	1.215754
Rata – rata total		85	1.174182

**4.2.1.3 Pengujian Performansi Sistem dengan Threshold = 1.2std**

Berikut merupakan tabel hasil pengujian sistem dengan menggunakan *threshold* sebesar 1.2 kali standar deviasi sinyal *input*.

**Tabel 4.3** Tabel Akurasi dan Waktu Komputasi Sistem dengan thr = 1.2 std

No.	Command	Akurasi (%)	Waktu komputasi rata - rata (s)
1	Play	70	1.162806
2	Pause	100	1.139087
3	Resume	60	1.12913
4	Stop	100	1.142613
5	Next	100	1.233556
6	Previous	90	1.15817
Rata – rata total		86.67	1.160894

**4.2.1.4 Pengujian Performansi Sistem dengan Threshold = 1.4 std**

Berikut merupakan tabel hasil pengujian sistem dengan menggunakan *threshold* sebesar 1.4 kali standar deviasi sinyal *input*

**Tabel 4.4** Tabel Akurasi dan Waktu Komputasi Sistem dengan thr = 1.4 std

No.	Command	Akurasi (%)	Waktu komputasi rata - rata (s)
1	Play	80	1.314376
2	Pause	100	1.246581
3	Resume	30	1.29133
4	Stop	100	1.264431
5	Next	100	1.366553
6	Previous	80	1.336373
Rata – rata total		81.6667	1.160894

**4.2.1.5 Pengujian Performansi Sistem dengan Threshold = 2 std**

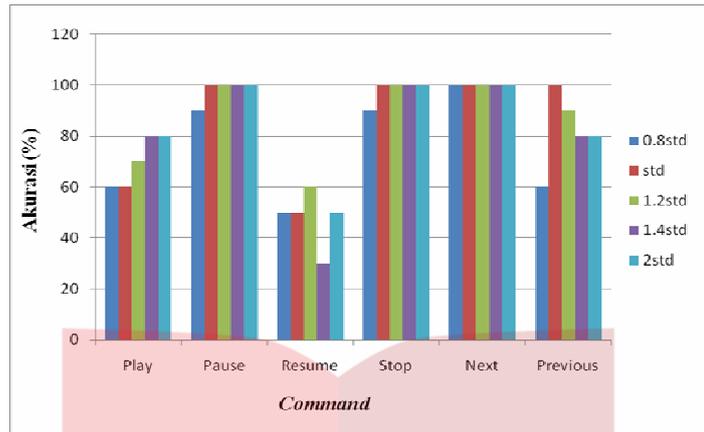
Berikut merupakan tabel hasil pengujian sistem dengan menggunakan *threshold* sebesar 2 kali standar deviasi sinyal *input*.

**Tabel 4.5** Tabel Akurasi dan Waktu Komputasi Sistem dengan thr = 2 std

No.	Command	Akurasi (%)	Waktu komputasi rata - rata (s)
1	Play	80	1.136406
2	Pause	100	1.098846
3	Resume	50	1.118818
4	Stop	100	1.105887
5	Next	100	1.147889
6	Previous	80	1.154254
Rata – rata total		85	1.127017

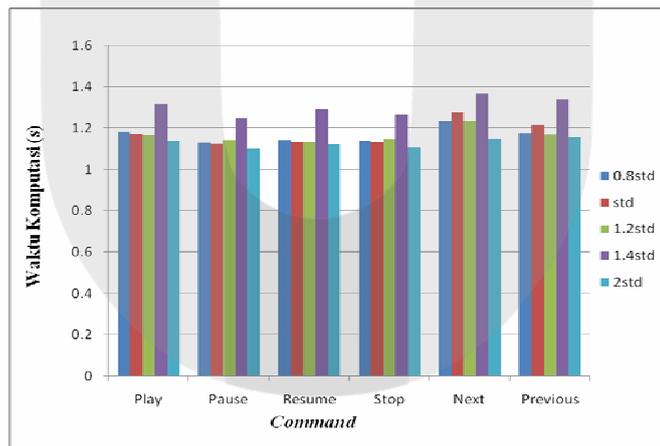
**4.2.1.6 Perbandingan pengujian berdasarkan nilai Threshold**

Dari hasil pengujian performansi sistem berdasarkan pengaruh *threshold* didapat grafik seperti berikut:



Gambar 4.3 grafik akurasi sistem berdasarkan nilai *threshold*

Berdasarkan grafik diatas dapat disimpulkan bahwa akurasi paling baik ketika diberi *threshold* sebesar 1.2 kali standar deviasi sinyal input. Hal ini dapat ditunjukkan dengan keunggulan sistem dalam mengenali hampir dari seluruh command dibandingkan dengan nilai *threshold* lainnya. Dan berikut adalah grafik waktu komputasi dari kelima pengujian.



Gambar 4.4 grafik waktu komputasi sistem berdasarkan nilai *threshold*

#### 4.2.2 Pengaruh ukuran *frame* terhadap akurasi *output* sistem.

Pada pengujian ini dilakukan dalam tiga skenario, yaitu pengujian sistem dengan ukuran *frame* 10 ms, 20 ms, dan 30 ms. Pembagian data kedalam *frame* – *frame* tertentu bertujuan untuk mempermudah sistem dalam melakukan komputasi sehingga sistem tidak perlu menghitung data per sampel namun dengan menghitung per *frame*.

#### 4.2.2.1 Pengujian Performansi Sistem dengan Ukuran *Frame* 10 ms

Pada pengujian ini dilakukan berdasarkan panjang waktu 10 ms yaitu setara dengan 10 x 8000 Hz = 80 sampel per frame. Berikut merupakan tabel hasil pengujian sistem yang telah dilakukan.

**Tabel 4.6** Tabel Akurasi dan Waktu Komputasi Sistem dengan ukuran *frame* 10 ms

No.	<i>Command</i>	Akurasi (%)	Waktu komputasi rata - rata (s)
1	<i>Play</i>	80	1.187253
2	<i>Pause</i>	100	1.10469
3	<i>Resume</i>	70	1.132149
4	<i>Stop</i>	100	1.11567
5	<i>Next</i>	100	1.19241
6	<i>Previous</i>	90	1.183251
Rata – rata total		90	1.15257

#### 4.2.2.2 Pengujian Performansi Sistem dengan Ukuran *Frame* 20 ms

Pada pengujian ini dilakukan berdasarkan panjang waktu 20 ms yaitu setara dengan 20 x 8000 Hz = 160 sampel per frame. Berikut merupakan tabel hasil pengujian sistem yang telah dilakukan.

**Tabel 4.7** Tabel Akurasi dan Waktu Komputasi Sistem dengan ukuran *frame* 20 ms

No.	<i>Command</i>	Akurasi (%)	Waktu komputasi rata - rata (s)
1	<i>Play</i>	80	1.157525
2	<i>Pause</i>	100	1.10266
3	<i>Resume</i>	60	1.119083
4	<i>Stop</i>	100	1.105932
5	<i>Next</i>	100	1.15597
6	<i>Previous</i>	70	1.156943
Rata – rata total		85	1.133019

### 4.2.2.3 Pengujian Performansi Sistem dengan Ukuran *Frame* 30 ms

Pada pengujian ini dilakukan berdasarkan panjang waktu 30 ms yaitu setara dengan  $10 \times 8000 \text{ Hz} = 240$  sampel per *frame*. Pengujian ini telah dilakukan sebelumnya ketika menghitung nilai *threshold* terbaik. Oleh karena itu hasil yang digunakan mengacu pada tabel 4.5.

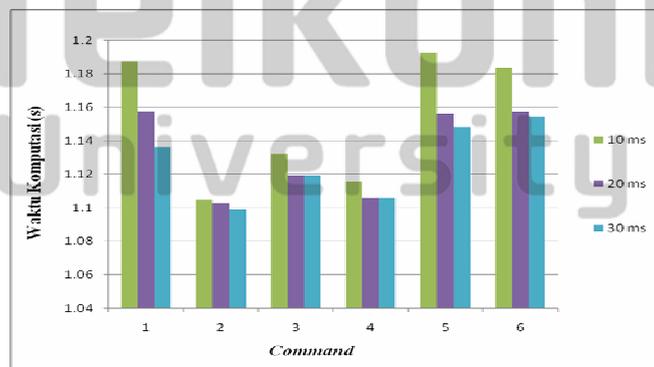
### 4.2.2.4 Perbandingan Pengujian Berdasarkan Pengaruh Ukuran *Frame*

Dari hasil pengujian performansi sistem berdasarkan pengaruh ukuran *frame* didapat grafik seperti berikut:



Gambar 4.5 grafik akurasi sistem berdasarkan ukuran *frame*

Berdasarkan grafik diatas dapat disimpulkan bahwa akurasi paling baik ketika diberi ukuran *frame* sebesar 10 ms atau setara dengan 80 sampel untuk tiap *frame*.. Hal ini dapat ditunjukkan dengan keunggulan sistem dalam mengenali hampir dari seluruh *command* dibandingkan dengan ukuran *frame* lainnya. Keunggulan ini dapat diperoleh karena dengan panjang data 6960 dan ukuran frame 80 sampel per *frame* menghasilkan *frame* yang lebih banyak dibandingkan dengan ukuran sampel lain, sehingga ciri yang dihasilkan pun lebih banyak dan spesifik. Berikut adalah grafik waktu komputasi dari kelima pengujian.



Gambar 4.6 grafik waktu komputasi sistem berdasarkan ukuran *frame*

### 4.2.3 Pengaruh *noise* terhadap akurasi *output* sistem.

Pada pengujian ini dilakukan dalam tiga skenario, yaitu pengujian sistem dengan sinyal input tanpa *noise*, sinyal input dengan *domestic noise*, dan sinyal input dengan *feedback noise*. Masing masing *noise* diberi lima skala volume yang sebanding untuk dianalisis pengaruhnya terhadap performansi sistem.

#### 4.2.1 Pengujian Performansi Sistem dengan *Domestic Noise*

Pada pengujian kali ini akan dilihat ketahanan sistem terhadap sinyal yang telah diberi *noise*. *Noise* yang digunakan berupa *domestic noise*, yaitu *noise* umum yang berasal dari lingkungan sekitar tempat *audio player* ini dijalankan. Pada kali ini *noise* yang digunakan berasal dari TV dengan lima skala volume yang berbeda untuk membandingkan performansinya.

Skala volume tersebut adalah 8, 10, 12, 14 dan 16 yang berasal dari TV LG MEZ41323511 dengan jarak 30 cm dari laptop.

Tabel 4.8 Tabel Akurasi Sistem ber*Noise Domestic*

No.	Skala Volume	Akurasi (%)						Rata-rata (%)
		<i>Play</i>	<i>Pause</i>	<i>Resume</i>	<i>Stop</i>	<i>Next</i>	<i>Previous</i>	
1	1	100	100	75	100	100	100	95.8333
2	2	100	100	75	100	100	100	95.8333
3	3	75	100	75	100	100	100	91.6667
4	4	75	100	75	100	100	75	87.5
5	5	50	100	50	100	100	100	83.3333
Rata-rata		80	100	70	100	100	95	90.8333

Tabel 4.9 Tabel Waktu Komputasi Sistem ber*Noise Domestic*

No.	Skala Volume	Waktu Komputasi (s)						Rata-rata (s)
		<i>Play</i>	<i>Pause</i>	<i>Resume</i>	<i>Stop</i>	<i>Next</i>	<i>Previous</i>	
1	1	1.1668	1.1219	1.1413	1.1413	1.1787	1.1728	1.15381
2	2	1.17727	1.1215	1.13513	1.1351	1.1764	1.1633	1.15145
3	3	1.17125	1.1242	1.1383	1.1383	1.1743	1.1654	1.15196
4	4	1.1885	1.1356	1.1321	1.1321	1.1802	1.1797	1.15803
5	5	1.1847	1.1232	1.1374	1.1374	1.1699	1.1735	1.15435
Rata-rata		1.1777	1.1253	1.13685	1.1368	1.1759	1.1709	1.15392

### 4.2.2 Pengujian Performansi Sistem dengan *Feedback Noise*

Pada pengujian kali ini akan dilihat ketahanan sistem terhadap sinyal yang memiliki *feedback noise*. Yaitu *noise* yang berasal dari sistem *audio player* ini sendiri ketika menjalankan lagu yang dimainkan. *Noise* yang digunakan juga diatur kedalam lima skala volume yang berbeda yang berasal dari *Basic Stereo Speakers SP-S105* dengan jarak 30 cm dari laptop. Skala tersebut ditentukan untuk mendapatkan volume yang sama dengan volume TV yang dijadikan parameter pengujian sebelumnya.

**Tabel 4.10** Tabel Akurasi Sistem ber*Noise Feedback*

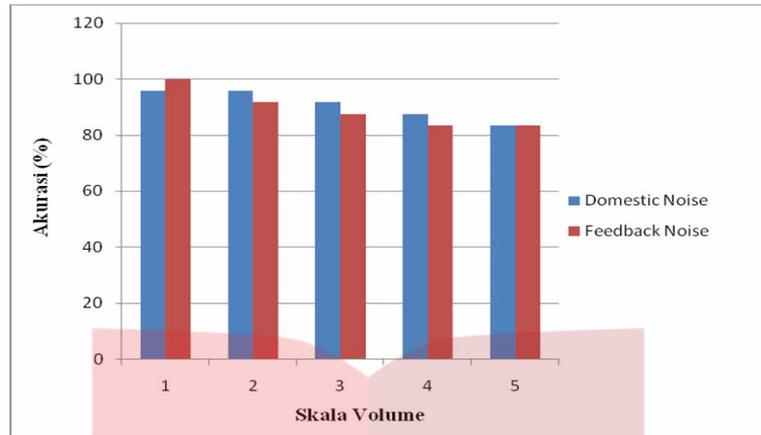
No.	Skala Volume	Akurasi (%)						Rata-rata (%)
		<i>Play</i>	<i>Pause</i>	<i>Resume</i>	<i>Stop</i>	<i>Next</i>	<i>Previous</i>	
1	1	100	100	100	100	100	100	100
2	2	100	100	75	100	100	75	91.6667
3	3	75	100	50	100	100	100	87.5
4	4	50	100	50	100	100	100	83.3333
5	5	100	75	50	100	100	75	83.3333
Rata-rata		85	95	65	100	100	90	89.1667

**Tabel 4.11** Tabel Waktu Komputasi Sistem ber*Noise Feedback*

No.	Skala Volume	Waktu Komputasi (s)						Rata-rata (s)
		<i>Play</i>	<i>Pause</i>	<i>Resume</i>	<i>Stop</i>	<i>Next</i>	<i>Previous</i>	
1	1	1.1657	1.134	1.13182	1.14	1.17	1.16495	1.15071
2	2	1.1728	1.122	1.12308	1.13	1.18	1.1675	1.14853
3	3	1.1684	1.122	1.1233	1.14	1.17	1.1643	1.1471
4	4	1.1714	1.122	1.12287	1.13	1.16	1.176	1.14626
5	5	1.1627	1.122	1.1232	1.13	1.17	1.17	1.146
Rata-rata		1.1682	1.124	1.12485	1.13	1.17	1.16855	1.14772

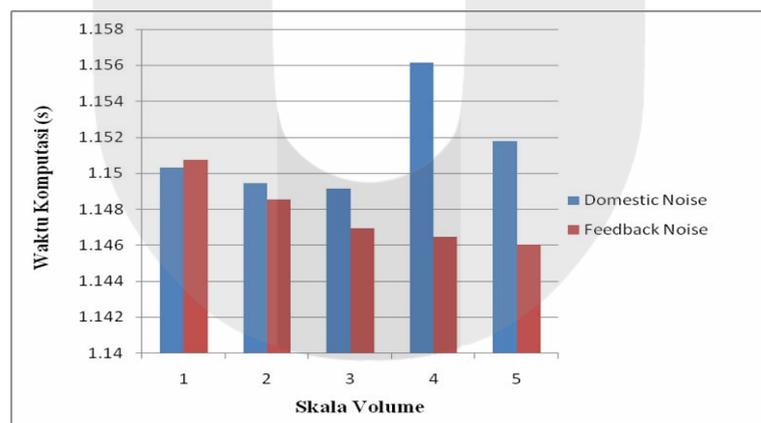
### 4.2.3 Perbandingan Pengujian Berdasarkan Pengaruh *Noise*

Dari hasil pengujian performansi sistem berdasarkan penambahan *domestic noise* dan *feedback noise* didapat grafik seperti berikut:



Gambar 4.7 grafik akurasi sistem berdasarkan skala volume *domestic noise* dan *feedback noise*

Dari gambar 4.7 dapat disimpulkan bahwa semakin besar skala volume maka *noise* yang dihasilkan juga menjadi semakin besar. Hal ini mempengaruhi tingkat akurasi sistem, karena semakin besar *noise* semakin kecil pula tingkat akurasi sistem yang dihasilkan. Dan berikut grafik waktu komputasi sistem berdasarkan pengaruh *noise*.



Gambar 4.8 grafik waktu komputasi sistem berdasarkan skala volume *domestic noise* dan *feedback noise*

#### 4.2.4 Pengaruh Beda Individu Terhadap Akurasi *Output* Sistem

Pada pengujian kali ini akan dibagi kedalam dua skenario yaitu pengujian performansi sistem menggunakan individu awam dan individu terlatih. Kedua tipe individu ini merupakan individu – individu yang sampel suaranya tidak ada dalam database dan diuji untuk melihat kehandalan sistem dalam mengenali *command* masukan. Masing – masing pengujian dilakukan oleh sepuluh individu berbeda dengan tiga kali percobaan untuk setiap *command*, kemudian dilakukan perhitungan dan analisis terhadap kedua skenario tersebut. Berikut ini adalah tabel

informasi individu yang melakukan pengujian terhadap sistem pengenalan suara untuk pengaturan *audio player* :

**Tabel 4.12** Informasi Individu Penguji

No.	Nama	Jenis Kelamin	Usia (Tahun)
1.	Rusdan	L	23
2.	Antony	L	23
3.	Iqbal	L	25
4.	Ari	L	22
5.	Ali	L	21
6.	Resti	P	21
7.	Dennis	P	21
8.	Nissa	P	20
9.	Aca	P	21
10.	Deby	P	21

Kesepuluh individu pada tabel 4.12 awalnya melakukan pengujian sebagai individu awam dengan tidak mengetahui bagaimana cara mengucapkan tiap *command* dalam mengatur sistem *audio player* ini. Setelah itu baru diajarkan dan dilatih cara pengucapannya untuk kemudian dilakukan pengujian ulang sebagai individu terlatih.

#### 4.2.4.1 Pengujian Performansi Sistem Berdasarkan Individu Awam

Pengujian ini dilakukan untuk mengetahui performansi sistem terhadap individu yang belum terlatih. Tujuan dilakukan pengujian ini adalah untuk menguji respon yang diberikan ketika individu baru memberikan *command* terhadap *audio player* ini, di mana pada dasarnya *audio player* disetting untuk dipergunakan oleh individu tertentu saja(individu terlatih)

**Tabel 4.13** Tabel Akurasi dan Waktu Komputasi Sistem berdasarkan individu awam

No.	<i>Command</i>	Akurasi (%)	Waktu komputasi rata - rata (s)
1	Play	16.67	1.2530
2	Pause	33.33	1.2198
3	Resume	6	1.2287
4	Stop	86.67	1.2362
5	Next	16.67	1.2502

6	Previous	13	1.2547
Rata – rata total		28.60	1.2396

#### 4.2.4.2 Pengujian Performansi Sistem Berdasarkan Individu Terlatih

Pengujian ini dilakukan untuk mengetahui performansi sistem terhadap individu yang sudah terlatih namun sampel suaranya tidak ada dalam database. Tujuan dilakukan pengujian ini untuk mengetahui kehandalan sistem terhadap kesesuaian *command* yang diberikan individu terlatih dengan respon yang dihasilkan oleh *audio player*.

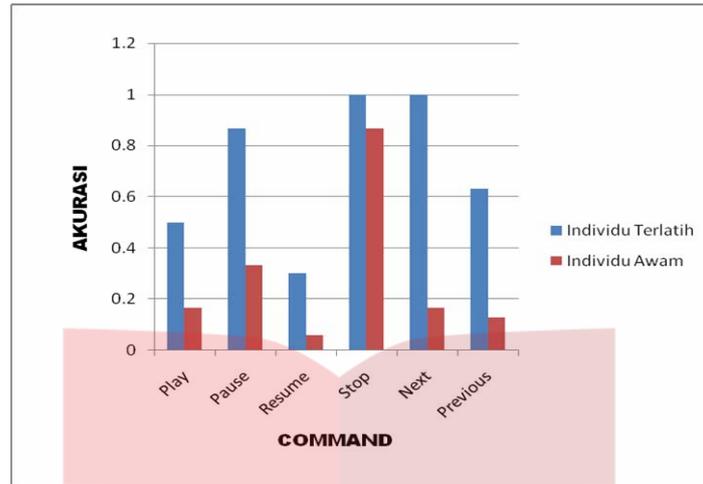
**Tabel 4.14** Tabel Akurasi dan Waktu Komputasi Sistem berdasarkan individu terlatih

No.	<i>Command</i>	Akurasi (%)	Waktu komputasi rata - rata (s)
1	<i>Play</i>	50	1.2563
2	<i>Pause</i>	86.67	1.2196
3	<i>Resume</i>	30	1.2310
4	<i>Stop</i>	100	1.2363
5	<i>Next</i>	100	1.2501
6	<i>Previous</i>	63	1.2554
Rata – rata total		71.60	1.2414

#### 4.2.4.3 Perbandingan Penggunaan Beda Individu pada Akurasi *Output* Sistem

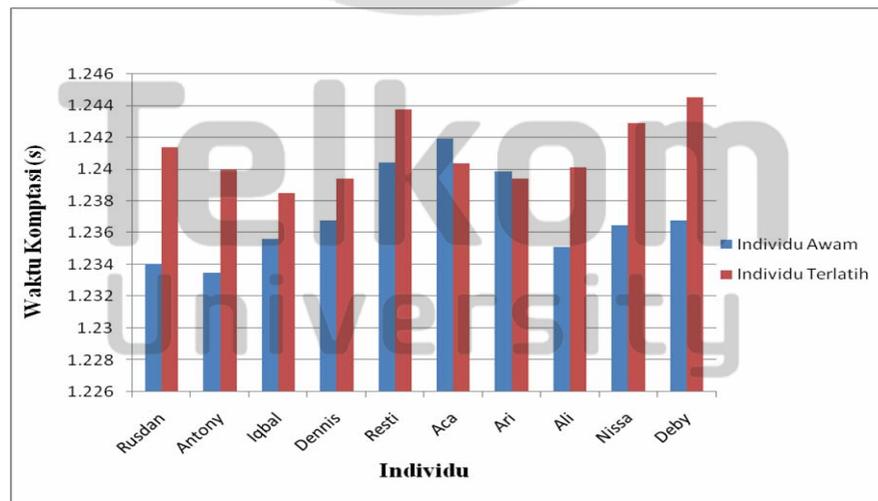
Berikut merupakan grafik yang diperoleh dari hasil pengujian performansi sistem berdasarkan penggunaan beda individu, yaitu individu awam dan individu terlatih.





Gambar 4.9 Grafik akurasi sistem berdasarkan pengaruh beda individu

Dari gambar 4.9 dapat dilihat bahwa sistem ini cukup baik untuk sistem yang *private*, dalam arti hanya komunitas tertentu saja yang bisa mengaksesnya. Terbukti dari perbedaan tingkat akurasi yang signifikan antara pengujian dengan individu awam dan pengujian dengan individu terlatih. Ketika melakukan pengujian, individu awam belum mengetahui bagaimana cara yang benar dalam mengujinya sehingga sistem tidak dapat dieksekusi dengan tepat, oleh karena itu tingkat akurasi sistem sangat kecil. Namun ketika melakukan pengujian dengan individu terlatih, walaupun tidak memberi sampel suara dalam *database*, sistem masih dapat mendeteksi dengan benar karena pengujian dilakukan oleh orang yang telah dilatih dahulu sebelumnya. Maka dari itu tingkat akurasi yang dihasilkan dari pengujian individu terlatih cukup besar. Dan berikut ini grafik waktu komputasi sistem berdasarkan pengaruh beda individu.

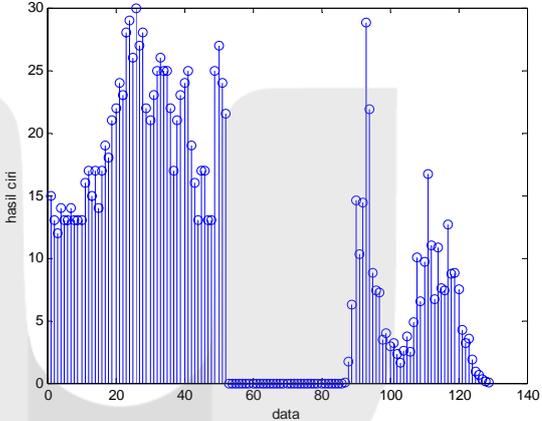
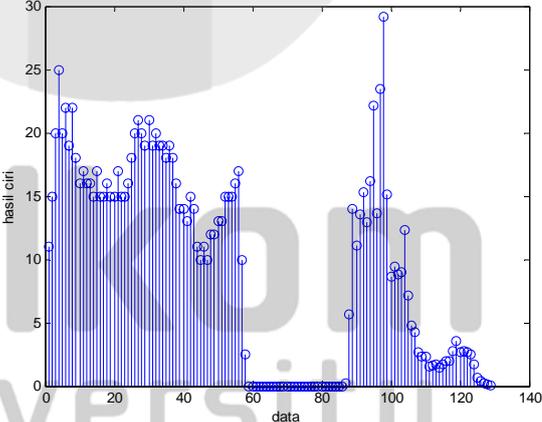


Gambar 4.10 Grafik waktu komputasi sistem berdasarkan pengaruh beda individu

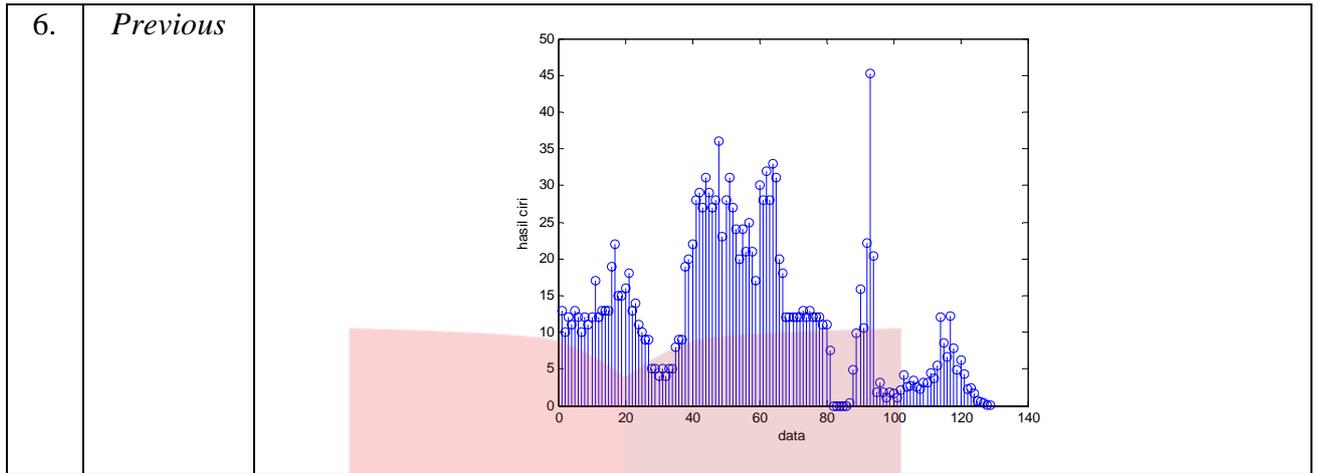
### 4.3 Analisis Ekstraksi Ciri

Dari pengujian yang telah dilakukan dapat disimpulkan bahwa akurasi yang dihasilkan oleh *command* “*resume*” merupakan akurasi terkecil untuk setiap pengujian. Sedangkan akurasi terbesar untuk setiap pengujian dihasilkan oleh *command* “*stop*”. Hal ini dapat terjadi karena “*stop*” menghasilkan ciri yang identik sehingga sistem dapat mengenali *command* ini dengan mudah. Sebaliknya *command* “*resume*” menghasilkan ciri yang kurang identik sehingga sistem masih sulit mengenali *command* ini. Berikut merupakan tabel hasil ciri untuk setiap *command*.

**Tabel 4.15** Tabel Hasil Ciri untuk Tiap *Command*

No.	<i>Command</i>	Hasil Ciri
1.	<i>Play</i>	
2.	<i>Pause</i>	

<p>3.</p>	<p><i>Resume</i></p>	
<p>4.</p>	<p><i>Stop</i></p>	
<p>5.</p>	<p><i>Next</i></p>	



Dari tabel 4.15 dapat dilihat bahwa ciri yang dihasilkan oleh *command* “*stop*” merupakan ciri yang paling berbeda dari semua *command*. Artinya *command* “*stop*” ini memang menghasilkan ciri yang identik sehingga sistem dapat dengan mudah mengenali *command* ini. Sedangkan untuk *command* “*resume*” ciri yang dihasilkan mirip dengan *command* “*play*” dan “*pause*” sehingga sistem sulit membedakan *command* yang dimaksud ketika cara pengucapan yang dihasilkan sedikit berbeda dengan *database*. Untuk memperjelas analisis berikut merupakan tabel untuk mengetahui kedekatan ciri “*resume*” dan “*stop*” terhadap *command* lain berdasarkan *euclidian distance*.

**Tabel 4.16** Tabel Kedekatan Ciri untuk Tiap *Command*

<i>Command</i>	<i>Play</i>	<i>Pause</i>	<i>Resume</i>	<i>Stop</i>	<i>Next</i>	<i>Previous</i>
<i>Resume</i>	9.3945	9.2799	-	16.1767	15.0107	16.8199
<i>Stop</i>	15.3581	12.9299	16.1767	-	16.7956	17.7903

## DAFTAR PUSTAKA

- [1] Suman, Maret 2011, sinyal suara, lembar suman, <http://lembarsuman.blogspot.com/2009/02/sinyal-suara.html>, diakses 10 April 2011.
- [2] Chrismanto, Antonius Rachmat. 2006. IM 2023 Multimedia, <http://lecturer.ukdw.ac.id/anton/download/multimedia3.pdf>. diakses tanggal 4 Desember 2011.
- [3] Setiawan, Arif. “Analisis Klasifikasi Suara Berdasarkan Gender dengan Format Wav Menggunakan Algoritma *K-Means*”. <http://www.scribd.com/doc/52494069/ANALISIS-KLASIFIKASI-SUARA-BERDASARKAN-GENDER>, diakses 8 Januari 2012.
- [4] Wahyudi, Happy Zuly. 2010. Konsep dasar audio, <http://digitalview.weebly.com/konsep-dasar-audio.html>. diakses tanggal 4 Desember 2011.
- [5] Adhi, Pribadi Mumpuni. 2011. “Analisis Karakteristik Akustik Suara Manusia”. Jurnal Praktikum. Institut Teknologi Bandung: Bandung.
- [6] Basuki, Achmad. “Aplikasi Pengolahan Suara untuk *Request* Lagu”. Politeknik Elektronika Negeri Surabaya. Institut Teknologi Sepuluh November: Surabaya.
- [7] Arman, Arry Akhmad. 2011. “Proses Pembentukan dan Karakteristik Sinyal Ucapan”. Departemen Teknik Elektro. Institut Teknologi Bandung: Bandung.
- [8] Rodiyansyah, Sandi Fajar. 2010. “Spectrogram dan Analisis Kemiripan Sinyal Suara dengan Pendekatan *Euclidian Distance*”. Magister Ilmu Komputer. Universitas Gadjah Mada: Yogyakarta.
- [9] Putra, Roba Laba. 2011. “Aplikasi Pengenalan Suara untuk *Request* Lagu Menggunakan Jaringan Syaraf Tiruan *Back Propagation* (JST-BP)”. Tugas akhir. IT Telkom: Bandung.
- [10] Aprilia, Bahagia. 2011. “Desain dan Implementasi Pengenalan Individu Melalui Sinyal Suara Menggunakan Metode *Learning Vector Quantization*”. Tugas Akhir. IT TELKOM: Bandung.

- [11] Heryantina, Citra Ayu. 2011. “Desain dan Analisis Sistem Aplikasi *Music Player* Berbasis Pengolahan Citra Digital Secara *Realtime*”. Tugas Akhir. IT TELKOM: Bandung.
- [12] Fajar, Galih Ahmad. 2011. “Pengenalan dan Analisis Kualitas Penalaan Nada Tunggal Piano Secara *Real time* Menggunakan Metode *JST-SOM*”. Tugas Akhir. IT Telkom: Bandung.
- [13] Putra, Andhika Bandung. 2006. “*Speech Recognition* Menggunakan Gabor Wavelet dan Jaringan Saraf Tiruan *Backpropagation* Untuk Sistem Keamanan Berbasis Suara”. Tugas Akhir. IT Telkom: Bandung.
- [14] Dokumen non personal, Template Matching, <http://www.scribd.com/doc/26870986/Template-Matching>, diakses 20 April 2011.

