



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**APLIKASI PENGENALAN UCAPAN HURUF
HIJAIYAH BERBASIS ANDROID UNTUK
REFRESHABLE BRAILLE DISPLAY
MENGUNAKAN *MEL-FREQUENCY CEPSTRUM
COEFFICIENTS* DAN *NEURAL NETWORK***

NIDA AMALIA
5113100100

Dosen Pembimbing
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

**APLIKASI PENGENALAN UCAPAN HURUF
HIJAIYAH BERBASIS ANDROID UNTUK
REFRESHABLE BRAILLE DISPLAY
MENGUNAKAN *MEL-FREQUENCY CEPSTRUM
COEFFICIENTS* DAN *NEURAL NETWORK***

NIDA AMALIA
5113100100

Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**HIJAIYAH SPEECH RECOGNITION ANDROID
APPLICATION FOR REFRESHABLE BRAILLE
DISPLAY USING MEL-FREQUENCY CEPSTRUM
COEFFICIENT AND NEURAL NETWORK**

NIDA AMALIA
5113100093

Supervisor I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Supervisor II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology and Communication
Sepuluh Nopember Institute of Technology
Surabaya, 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

APLIKASI PENGENALAN UCAPAN HURUF HIJAIYAH BERBASIS ANDROID UNTUK *REFRESHABLE BRAILLE* *DISPLAY* MENGGUNAKAN *MEL-FREQUENCY* *CEPSTRUM COEFFICIENTS* DAN *NEURAL NETWORK*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
NIDA AMALIA
NRP: 5113 100 100

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP. 197512202001122002



(Pembimbing 1)

Dini Adni Navastara, S.Kom.
NIP. 198510172015042001

(Pembimbing 2)

SURABAYA
JULI, 2017

[Halaman ini sengaja dikosongkan]

**APLIKASI PENGENALAN UCAPAN HURUF HIJAIYAH
BERBASIS ANDROID UNTUK *REFRESHABLE BRAILLE*
DISPLAY MENGGUNAKAN *MEL-FREQUENCY*
CEPSTRUM COEFFICIENTS DAN *NEURAL NETWORK***

Nama Mahasiswa : Nida Amalia
NRP : 5113 100 100
Jurusan : Teknik Informatika, FTIf ITS
Dosen Pembimbing 1 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Pada tahun 2010 sebanyak 3.05% atau lebih dari 7.2 juta jiwa penduduk Indonesia mengalami kesulitan melihat. Dari tingginya angka tersebut, maka perlu adanya perhatian khusus terkait kebutuhan mereka, termasuk kebutuhan rohani. Sebagai negara dengan mayoritas penduduk beragama Islam, Al-Qur'an menjadi salah satu kebutuhan yang penting. Sayangnya, sarana untuk belajar Al-Qur'an braille masih sangat minim sehingga banyak penyandang kesulitan melihat yang masih belum bisa membaca Al-Qur'an.

Pada tugas akhir ini akan diusulkan media pembelajaran Al-Qur'an yang dapat digunakan penyandang kesulitan melihat. Media pembelajaran tersebut berupa aplikasi pengenalan suara berbasis Android yang diintegrasikan dengan refreshable braille display sebagai media untuk menampilkan pola braille dari Huruf Hijaiyah. Aplikasi yang akan dibangun memiliki fitur utama pengenalan suara untuk Huruf Hijaiyah dengan menggunakan teknik ekstraksi fitur Mel-Frequency Cepstrum Coefficients (MFCC). Sebelumnya, akan dilakukan preprocessing terlebih dahulu menggunakan Voice Activity Detection (VAD) untuk membedakan segmen suara informasi dengan segmen suara non-informasi. Langkah pertama dalam proses MFCC adalah pre-emphasis filter dimana proses ini akan memperbaiki sinyal dari

noise. Selanjutnya dilakukan tahap frame blocking dan windowing untuk membagi sinyal menjadi beberapa frame dan mengurangi diskontinuitas sinyal yang hasilnya diproses menggunakan Fast Fourier Transform (FFT) sehingga didapatkan sinyal dalam frekuensi domain. Ukuran energi dari setiap frekuensi band dihitung pada proses Filterbank. Langkah terakhir adalah Discrete Cosine Transform (DCT) yang menghasilkan Mel Cepstrum. Pola yang dihasilkan dari MFCC dilatih menggunakan algoritma Neural Network Back-Propogation yang akan mengklasifikasi suara yang diucapkan.

Uji coba yang dilakukan terhadap 290 data suara yang terdiri dari 145 suara laki-laki dan 145 suara perempuan. Hasil uji coba menunjukkan bahwa metode ini dapat memberikan hasil klasifikasi yang akurat dengan akurasi adalah 97.93% untuk data laki-laki, 93.10% untuk data perempuan, dan 90.34% untuk uji coba menggunakan keseluruhan data.

Kata kunci: Pengenalan Ucapan, Mel-Frequency Cepstrum Coefficients, Back Propagation Neural Network, Hijaiyah, Android, Refreshable Braille.

HIJAIYAH SPEECH RECOGNITION ANDROID APPLICATION FOR REFRESHABLE BRAILLE DISPLAY USING MEL-FREQUENCY CEPSTRUM COEFFICIENT AND NEURAL NETWORK

Student Name : Nida Amalia
Registration Number : 5113 100 100
Department : Informatics Engineering, FTIf ITS
First Supervisor : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Second Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

Abstract

In 2010, there are 3.05% or more than 7.2 million blind people in Indonesia. From this high numbers, we have to give special attention for their needs, including spiritual needs. As a country with a majority Muslim population, Qur'an becomes one of the most important needs. Unfortunately, the lack of media to learn braille Qur'an makes many blind people still can not read it.

This final project will propose media to learn Qur'an that can be used by blind people. The media is an Android-based speech recognition application that is integrated with refreshable braille display as a media for displaying braille patterns of Hijaiyah Letters. The main feature of the application is speech recognition for Hijaiyah Letters by using Mel-Frequency Cepstrum Coefficients (MFCC) for feature extraction. Preprocessing will be done first by using Voice Activity Detection (VAD) to distinguish the information and non-information voice segment. The first step in the MFCC process is the pre-emphasis filter that will improve the signal from noise. Next, frame blocking and windowing stage that will divide the signal into multiple frames and reduce the signal discontinuity. Then, the result is processed using Fast Fourier Transform (FFT) to obtain signal in frequency domain. The energy size of each frequency band is calculated on the Filterbank process. The final step is the Discrete Cosine Transform

(DCT) that produces Mel Cepstrum. Patterns generated from MFCC are trained using the Neural Network Back-Propogation algorithm that will classify spoken sounds.

Testing that is held to 290 audio data which contain of 145 man audio data and 145 woman audio data. The result of the test shows that this method can give accurate result with accuracy 97.93% for man audio data, 93.10% for woman audio data, and 90.34 for testing with all data.

Keywords: Speech Recognition, Mel-Frequency Cepstrum Coefficients, Back Propagation Neural Network, Hijaiyah, Android, Refreshable Braille.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**Aplikasi Pengenalan Ucapan Huruf Hijaiyah Berbasis Android untuk *Refreshable Braille Display* Menggunakan *Mel-Frequency Cepstrum Coefficients* dan *Neural Network*”.**

Buku Tugas Akhir ini disusun dengan harapan dapat memberikan manfaat mempermudah penyandang tunanetra supaya bisa lebih mandiri untuk mempelajari Huruf Hijaiyah, baik untuk mengenali huruf maupun pengucapannya. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

Dalam perancangan, pengerjaan, dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. dan Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi ide, nasihat, dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Orang tua penulis Bapak Achmad Syafi'i Iskandar dan Ibu Istifaiyah yang telah memberikan memberikan doa dan senantiasa memberi dukungan, nasihat, dan motivasi kepada penulis dalam mengerjakan Tugas Akhir.
3. Saudara penulis, Mbak Nada dan Mas Oki yang telah memberikan dorongan dan arahan untuk mengerjakan Tugas Akhir ini. Adek bintang serta seluruh keluarga besar yang telah memberikan motivasi kepada penulis.

4. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku kepala Laboratorium KCV yang bersedia meminjamkan salah satu komputer untuk pengerjaan tugas akhir.
5. Wahyu yang selalu siap menjadi rekan kerja dan tempat berbagi cerita bagi penulis dalam suka dan duka.
6. Hamid yang telah setia menjadi tim untuk menyelesaikan alat.
7. Teman-teman Admin Laboratorium KCV yang telah banyak membantu memfasilitasi dan menghibur dalam pengerjaan Tugas Akhir.
8. Titis, Sari, dan mahasiswa angkatan 2013 lain yang telah banyak membantu, menemani dan bekerja sama dengan penulis dalam melewati berbagai mata kuliah dan aktivitas perkuliahan lainnya.
9. Nela, Risma, dan Okta yang selalu menemani begadang, menjadi teman diskusi, dan memberikan semangat kepada penulis.
10. Haqiqi, Ery, Hilman, Luqman, Habib, Dini, Tyon dan teman-teman lain yang telah membantu penulis membuat data untuk Tugas Akhir ini.
11. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga, penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan ke depannya.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

| | |
|---|-------------|
| LEMBAR PENGESAHAN | v |
| Abstrak | vii |
| Abstract..... | ix |
| KATA PENGANTAR..... | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR..... | xvii |
| DAFTAR TABEL | xix |
| DAFTAR KODE SUMBER..... | xxi |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Batasan Masalah | 4 |
| 1.4 Tujuan Tugas Akhir..... | 4 |
| 1.5 Manfaat Tugas Akhir..... | 4 |
| 1.6 Metodologi..... | 5 |
| 1.7 Sistematika Laporan | 6 |
| BAB II DASAR TEORI..... | 9 |
| 2.1 Huruf Hijaiyah..... | 9 |
| 2.2 Huruf Braille..... | 10 |
| 2.3 Refreshable Braille Display | 11 |
| 2.4 Arduino..... | 12 |
| 2.4.1 Arduino Nano..... | 13 |
| 2.5 Android..... | 13 |
| 2.6 Retrofit..... | 14 |
| 2.7 Pengenalan Suara..... | 14 |
| 2.8 Voice Activity Detection (VAD)..... | 15 |
| 2.9 Mel-Frequency Cepstrum Coefficients (MFCC)..... | 16 |
| 2.10 Neural Network | 20 |
| 2.10.1 Algoritma Gradient Descent with Momentum and Adaptive Learning | 23 |
| 2.11 Confusion Matrix..... | 24 |

| | |
|---|-----------|
| BAB III PERANCANGAN SISTEM | 27 |
| 3.1 Data..... | 27 |
| 3.1.1 Data Masukan..... | 27 |
| 3.1.2 Data Proses..... | 28 |
| 3.1.3 Data Keluaran..... | 28 |
| 3.2 Desain Umum Keseluruhan Sistem | 29 |
| 3.2.1 Desain Umum Aplikasi pada Hardware | 30 |
| 3.2.2 Desain Umum Aplikasi pada Android | 31 |
| 3.2.3 Desain Umum Aplikasi pada MATLAB | 32 |
| BAB IV IMPLEMENTASI | 41 |
| 4.1 Lingkungan Implementasi | 41 |
| 4.1.1 Perangkat Keras..... | 41 |
| 4.1.2 Perangkat Lunak..... | 41 |
| 4.2 Implementasi Arduino | 41 |
| 4.2.1 Implementasi Pengaturan Komponen Pada Arduino | 42 |
| 4.2.2 Implementasi Fungsi Register | 42 |
| 4.2.3 Implementasi Konversi Huruf Hijaiyah | 43 |
| 4.2.4 Implementasi Pengiriman Data Braille | 44 |
| 4.2.5 Implementasi Main Program | 45 |
| 4.3 Implementasi Android | 47 |
| 4.3.1 Implementasi Penerimaan Data Braille | 47 |
| 4.3.2 Implementasi Perekaman Suara | 48 |
| 4.3.3 Implementasi Pembuatan File .wav..... | 49 |
| 4.3.4 Implementasi Pengiriman Suara..... | 50 |
| 4.3.5 Implementasi Penerimaan Hasil Klasifikasi..... | 52 |
| 4.4 Implementasi MATLAB..... | 53 |
| 4.4.1 Preprosessing dengan VAD..... | 53 |
| 4.4.2 Ekstraksi Fitur MFCC | 54 |
| 4.4.3 Klasifikasi Neural Network..... | 58 |
| BAB V UJI COBA DAN EVALUASI | 61 |
| 5.1 Lingkungan Uji Coba..... | 61 |
| 5.2 Data Uji Coba | 61 |
| 5.3 Hasil Uji Coba | 61 |

| | | |
|--|--|------------|
| 5.4 | Skenario Uji Coba..... | 69 |
| 5.4.1 | Uji Coba Parameter Frame Duration..... | 69 |
| 5.4.2 | Uji Coba Ukuran Overlap | 70 |
| 5.4.3 | Uji Coba Jumlah Neuron..... | 71 |
| 5.4.4 | Uji Coba Learning Rate | 71 |
| 5.4.5 | Uji Coba Berdasarkan Gender..... | 73 |
| 5.5 | Hasil dan Evaluasi | 73 |
| BAB VI KESIMPULAN DAN SARAN | | 75 |
| 6.1 | Kesimpulan..... | 75 |
| 6.2 | Saran | 75 |
| LAMPIRAN..... | | 81 |
| BIODATA PENULIS..... | | 121 |

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1.1 Al-Qur'an Braille | 2 |
| Gambar 2.1 Penomoran Sel Braille [9] | 10 |
| Gambar 2.2 Huruf Hijaiyah dalam Braille [10] | 11 |
| Gambar 2.3 Refreshable Braille Display | 12 |
| Gambar 2.4 Arduino Nano [14] | 13 |
| Gambar 2.5 Blok Diagram Proses MFCC | 16 |
| Gambar 2.6 Mel Filter Bank | 19 |
| Gambar 3.1 Desain Umum Alur Sistem | 30 |
| Gambar 3.2 Data untuk Arduino | 30 |
| Gambar 3.3 Diagram Alir Sistem pada Refreshable Braille Display | 31 |
| Gambar 3.4 Diagram Alir Sistem pada Android..... | 32 |
| Gambar 3.5 Diagram Alir Sistem pada MATLAB | 33 |
| Gambar 3.6 Diagram Alir Proses VAD | 35 |
| Gambar 3.7 Diagram Alir Proses MFCC..... | 38 |
| Gambar 5.1 Sinyal Suara Huruf Alif | 62 |
| Gambar 5.2 Hasil Pemotongan Sinyal Huruf Alif dengan VAD | 63 |
| Gambar 5.3 Hasil Preemphasis Filter Pada Sinyal Suara Alif | 64 |
| Gambar 5.4 Hasil Frame Blocking Pada Sinyal Suara Alif | 65 |
| Gambar 5.5 Hasil Magnitude Spectrum Pada Sinyal Suara Alif | 66 |
| Gambar 5.6 Filterbank Sinyal Huruf Alif | 67 |
| Gambar 5.7 Window Pada Proses Cepstral Lifter | 67 |
| Gambar 5.8 MFCC Sinyal Huruf Alif | 68 |

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Daftar Huruf Hijaiyah dan Pengucapannya [9]..... | 9 |
| Tabel 2.2 Confusion Matrix | 25 |
| Tabel 5.1 Hasil Uji Coba Parameter Frame Duration | 69 |
| Tabel 5.2 Hasil Uji Coba Parameter Ukuran Overlap..... | 70 |
| Tabel 5.3 Hasil Uji Coba Parameter Jumlah Neuron..... | 71 |
| Tabel 5.4 Hasil Uji Coba Parameter Learning Rate..... | 72 |
| Tabel 5.5 Hasil Uji Coba Parameter Gender..... | 73 |

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

| | |
|---|----|
| Kode Sumber 4.1 Pengaturan Komponen Pada Arduino..... | 42 |
| Kode Sumber 4.2 Fungsi Register | 43 |
| Kode Sumber 4.3 Konversi Data Braille Arduino | 44 |
| Kode Sumber 4.4 Pengiriman Data Braille ke Android..... | 45 |
| Kode Sumber 4.5 <i>Main</i> Program Arduino | 47 |
| Kode Sumber 4.6 Penerimaan Data Braille ke Android | 48 |
| Kode Sumber 4.7 Perekaman Suara..... | 49 |
| Kode Sumber 4.8 Pembuatan File .wav | 50 |
| Kode Sumber 4.9 Pengiriman suara..... | 52 |
| Kode Sumber 4.10 Penerimaan Hasil Klasifikasi Suara ke Aplikasi Android | 53 |
| Kode Sumber 4.11 Implementasi VAD | 54 |
| Kode Sumber 4.12 Proses <i>Preemphasis Filter</i> | 54 |
| Kode Sumber 4.13 Proses <i>Framing</i> | 55 |
| Kode Sumber 4.14 Implementasi Windowing..... | 56 |
| Kode Sumber 4.15 Implementasi Magnitude Spectrum | 56 |
| Kode Sumber 4.16 Implementasi <i>FilterBank</i> | 57 |
| Kode Sumber 4.17 Implementasi Konversi ke Satuan Herzt..... | 57 |
| Kode Sumber 4.18 Implementasi Konversi ke Satuan Mel | 57 |
| Kode Sumber 4.19 Implementasi DCT..... | 58 |
| Kode Sumber 4.20 Implementasi Neural Network | 59 |

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

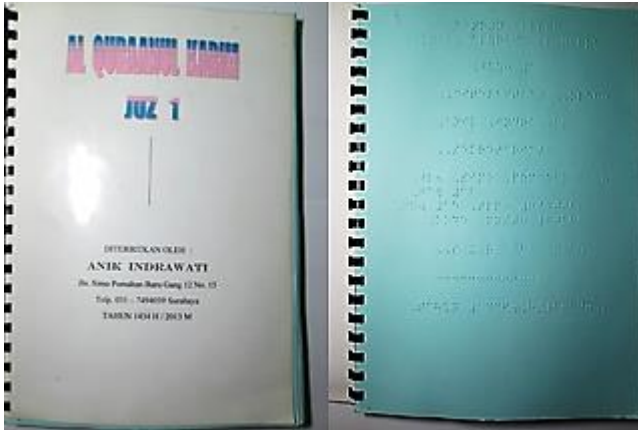
Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Kesulitan melihat merupakan kondisi dimana seseorang tidak dapat melihat dengan baik, meskipun menggunakan kacamata dan dalam kondisi yang cukup cahaya [1]. Berdasarkan data dari Badan Pusat Statistik (BPS) pada tahun 2010, persentase penduduk berumur ≥ 10 tahun di Indonesia yang mengalami kesulitan melihat adalah 3.05% [1]. Padahal, jumlah penduduk Indonesia pada tahun 2010 adalah 237.556.363 orang [2], sehingga dengan kata lain penduduk yang mengalami kesulitan melihat mencapai lebih dari 7.2 juta jiwa. Pada Buletin Jendela Data dan Informasi Kesehatan 2014, diperoleh data dimana kondisi kesulitan melihat menempati posisi dengan jumlah tertinggi dibandingkan dengan disabilitas yang lainnya [3]. Data-data tersebut menunjukkan bahwa tidak sedikit masyarakat Indonesia yang mengalami kesulitan melihat sehingga perlu adanya perhatian khusus terkait kebutuhan mereka.

Diantara kebutuhan dari seorang individu yang harus dipenuhi adalah kebutuhan rohani. Kebutuhan rohani merupakan kebutuhan yang diperlukan untuk pemenuhan jiwa atau rohani [4]. Kebutuhan rohani dapat dipenuhi sesuai dengan agama masing-masing, salah satunya dengan membaca kitab. Seperti yang kita ketahui, mayoritas penduduk Indonesia beragama Islam dimana Al-Qur'an merupakan pedoman bagi para pemeluknya. Saat ini, masyarakat bisa mendapatkan dan belajar membaca Al-Qur'an dengan mudah. Namun, berbeda halnya dengan penyandang kesulitan melihat, hal ini dikarenakan Al-Qur'an yang mereka

gunakan sedikit berbeda karena ditulis menggunakan pola braille yang dapat dilihat pada Gambar 1.1.



Gambar 1.1 Al-Qur'an Braille

Al-Qur'an, baik braille maupun bukan, disusun atas Huruf Hijaiyah. Berbeda dengan huruf latin, Huruf Hijaiyah memiliki pengucapan yang unik sehingga apabila pelafalannya salah, maka juga berdampak pada maknanya. Oleh karena itu, mengenal pembacaan Huruf Hijaiyah merupakan suatu dasar yang penting dalam mempelajari Al-qur'an sehingga dibutuhkan bimbingan khusus supaya dapat membacanya dengan benar. Namun, minimnya sarana untuk belajar Al-Qur'an braille menyebabkan banyak penyandang kesulitan melihat yang masih belum bisa membaca Al-Qur'an braille.

Pada penelitian sebelumnya oleh Budiman Putra, telah didesain dan dibangun aplikasi pembelajaran Al-Qur'an dengan fitur pengenalan suara menggunakan ekstraksi fitur *Mel-Frequency Cepstrum Coefficients* (MFCC) sampai pada tahap *prototype* [5]. Metode MFCC juga digunakan pada penelitian A. N. Wahidah untuk pengenalan ucapan makhraj Huruf Hijaiyah [6]. Namun, aplikasi-aplikasi tersebut dibangun pada media desktop sehingga

tidak bisa digunakan oleh penyandang kesulitan melihat secara mandiri.

Salah satu media yang dapat digunakan oleh penyandang kesulitan melihat adalah *refreshable braille display* yang merupakan perangkat yang dapat menampilkan karakter braille secara dinamis dan bisa berubah sesuai dengan perintah yang diberikan oleh penggunanya [7]. Namun, saat ini belum ada *refreshable braille display* yang memiliki fitur sebagai media pembelajaran Al-Qur'an.

Pada penelitian ini akan diusulkan media pembelajaran Al-Qur'an yang dapat digunakan penyandang kesulitan melihat. Media pembelajaran tersebut berupa aplikasi pengenalan ucapan berbasis Android yang diintegrasikan dengan *refreshable braille display* yang menjadi media untuk menampilkan pola braille dari Huruf Hijaiyah. Komunikasi antara aplikasi dengan alat dilakukan dengan menggunakan media bluetooth. Aplikasi yang akan dibangun memiliki fitur utama pengenalan ucapan untuk Huruf Hijaiyah. Tahapan yang dilakukan untuk proses pengenalan ucapan terdiri tahap pelatihan dan tahapan pengujian dimana sinyal suara akan melalui preprocessing terlebih dahulu dengan *Voice Activity Detection (VAD)*. Kemudian dilakukan ekstraksi fitur menggunakan MFCC. Pola suara yang sudah dibentuk dari proses MFCC selanjutnya akan diolah menggunakan algoritma *Neural Network* untuk dilakukan klasifikasi.

Hasil yang diharapkan dari pengerjaan Tugas Akhir ini adalah berupa aplikasi pengenalan ucapan Huruf Hijaiyah berbasis Android untuk *refreshable braille display* menggunakan MFCC dan *Neural Network*. Dengan adanya media pembelajaran ini diharapkan penyandang kesulitan melihat dapat mempelajari Huruf Hijaiyah sebagai dasar membaca Al-Qur'an secara mandiri.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana memisahkan segmen suara informasi dengan segmen suara non informasi menggunakan *Voice Activity Detection* (VAD)?
2. Bagaimana melakukan ekstraksi fitur menggunakan MFCC pada sinyal suara Huruf Hijaiyah?
3. Bagaimana aplikasi dapat mengenali dan mengklasifikasikan suara yang dilafalkan?
4. Bagaimana evaluasi hasil dari pengenalan ucapan Hiruf Hijaiyah pada aplikasi?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Aplikasi klien dibangun dengan Android Studio menggunakan bahasa Java.
2. Aplikasi server dibangun dengan menggunakan MATLAB R2016b.
3. Sistem *embedded* pada *refreshable braille display* dibangun dengan Arduino IDE menggunakan bahasa C.
4. *Database* berupa audio dari Huruf Hijaiyah yang diambil dari 10 orang responden (5 orang laki-laki dan 5 orang perempuan).
5. Terdapat 29 hasil klasifikasi berupa huruf hijaiyah.

1.4 Tujuan Tugas Akhir

Tujuan dari Tugas Akhir ini adalah membuat aplikasi pengenalan ucapan Huruf Hijaiyah berbasis Android untuk *refreshable braille display* menggunakan MFCC dan *Neural Network*.

1.5 Manfaat Tugas Akhir

Manfaat yang diperoleh dari pembuatan tugas akhir ini yaitu mempermudah penyandang tunanetra supaya bisa lebih mandiri untuk mempelajari Huruf Hijaiyah, baik untuk mengenali huruf maupun pengucapannya.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada studi literatur, dilakukan pengumpulan data dan studi terhadap sejumlah referensi yang diperlukan dalam pengerjaan tugas akhir. Referensi tersebut didapatkan dari beberapa artikel yang dipublikasikan oleh jurnal. Selain dari artikel, studi literatur juga dilakukan melalui pencarian referensi dari internet yang membahas mengenai informasi yang dibutuhkan, seperti proses ekstraksi fitur MFCC dan klasifikasi *Neural Network*. Metode yang diusulkan tugas akhir ini mengacu pada penelitian yang dilakukan oleh Mansour Alsulaiman yang melakukan perbandingan terhadap berbagai metode ekstraksi fitur untuk pengenalan ucapan pada Bahasa Arab menggunakan klasifikasi *Back Propagation Neural Network*. Metode ekstraksi fitur yang diujicobakan pada penelitian tersebut adalah *Linear Prediction Coefficients* (LPC), *LPC Cepstral Coefficients* (LPCEPSTRA), *Linear Prediction Reflection Coefficients* (LPREFC), *Mel-frequency Cepstral Coefficients* (MFCC), *Linear Mel-filter bank channel Outputs* (MELSPEC). Dari hasil percobaan yang dilakukan terhadap pengucapan angka dalam Bahasa Arab, didapatkan hasil terbaik untuk metode ekstraksi fitur MFCC [8].

2. Analisis dan Desain Perangkat Lunak

Pada tahap ini disusun rancang bangun dari perangkat lunak yang dibangun. Pengguna menginputkan suara melalui aplikasi dengan frekuensi *sampling* sebesar 44100. Kemudian, server akan memproses sinyal input dengan melakukan pemisahan sinyal suara informasi dan non-informasi, ekstraksi fitur sinyal, dan klasifikasi. Setelah proses selesai, sistem akan menampilkan kelas hasil klasifikasi sinyal suara.

3. Implementasi Perangkat Lunak

Sistem pengenalan ucapan Huruf Hijaiyah akan dibuat dengan bahasa pemrograman MATLAB menggunakan kakas bantu IDE MATLAB R2016b pada platform *desktop* untuk bagian server dan menggunakan bahasa pemrograman Java menggunakan kakas bantu IDE Android Studio untuk bagian klien. Pada aplikasi Android digunakan *library* Retrofit untuk upload media audio. Sedangkan pada MATLAB digunakan *Toolbox Signal Processing*. Kakas bantu pendukung lain diantaranya Arduino IDE dengan bahasa C untuk pemrograman pada alat dan Microsoft Word sebagai pengolah dokumen.

4. Uji Coba dan Evaluasi

Dalam tahap ini, dilakukan pengujian parameter-parameter yang dibutuhkan pada proses klasifikasi. Hasil klasifikasi dari sinyal dibandingkan dengan *ground truth* terhadap sinyal input untuk dihitung performanya dengan menghitung nilai akurasi, *sensitivity* dan *specificity*.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini berisi tentang analisis dan perancangan desain sistem pengenalan ucapan Huruf Hijaiyah.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi kedepannya.

[Halaman ini sengaja dikosongkan]

BAB II DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai Huruf Hijaiyah sebagai studi kasus, MFCC untuk ekstraksi fitur sinyal suara, dan *Neural Network* untuk klasifikasi pengucapan Huruf Hijaiyah.

2.1 Huruf Hijaiyah

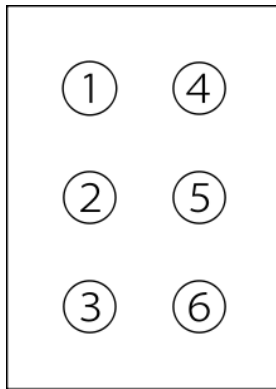
Huruf Hijaiyah merupakan huruf penyusun kata dalam Al Qur'an. Menurut istilah, Huruf Hijaiyah merupakan suara yang berpegang/tertekan kepada makhrajnya (tempat keluarnya huruf). Jumlah Huruf Hijaiyah terdiri atas 29 huruf yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Daftar Huruf Hijaiyah dan Pengucapannya [9]

| Huruf | Pengucapan | Huruf | Pengucapan |
|-------|------------|-------|------------|
| ا | alif | ط | tha |
| ب | ba | ظ | zha |
| ت | ta | ع | ain |
| ث | tsa | غ | ghain |
| ج | jim | ف | fa |
| ح | ḥa | ق | qaf |
| خ | kha | ك | kaf |
| د | dal | ل | lam |
| ذ | dzal | م | mim |
| ر | ra | ن | nun |
| ز | zai | ه | ha |
| س | sin | ء | hamzah |
| ش | syin | و | wau |
| ص | shad | ي | ya |
| ض | dhad | | |

2.2 Huruf Braille

Huruf Braille adalah sejenis sistem tulisan sentuh yang digunakan oleh orang buta. Sistem ini diciptakan oleh seorang Perancis yang bernama Louis Braille. Sistem tulisan Braille mencapai taraf kesempurnaan pada tahun 1834. Huruf-huruf Braille menggunakan kerangka penulisan seperti kartu domino. Satuan dasar dari sistem tulisan ini disebut sel Braille, di mana tiap sel terdiri dari enam titik timbul; tiga baris dengan dua titik. Sel braille mempunyai enam titik timbul yang dinomorkan seperti Gambar 2.1.



Gambar 2.1 Penomoran Sel Braille [9]

Huruf braille terdiri dari beberapa jenis, salah satunya adalah huruf braille versi Hijaiyah. Seperti telah kita ketahui sebelumnya bahwa huruf braille terdiri dari koordinasi enam titik, sehingga Huruf Hijaiyah dalam versi braille juga terdiri dari koordinasi enam titik. Koordinasi titik-titik dalam tersebut dapat dilihat melalui Gambar 2.2.

| | | | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| ا | ب | ت | ث | ج | ح | خ | د | ذ | ر |
| •○ ○○ ○○ | •○ •○ ○○ | ○• •• •○ | •• ○• ○• | ○• •• ○○ | •○ •• ○• | •• ○○ •• | •• ○• ○○ | ○• ○• •• | •• •• •• |
| ز | س | ش | ص | ض | ط | ظ | ع | غ | ف |
| •○ ○• •• | ○• •• •○ | •• ○• ○• | •• •○ •• | •• ○• •○ | ○• •• •• | •• •• •• | •○ •• •• | ○• ○• •• | •• ○• ○○ |
| ق | ك | ل | م | ن | ه | و | ي | ء | |
| •• •• •○ | •○ ○○ •○ | •○ •○ •○ | •• ○○ •○ | •• ○• •○ | •○ •• ○○ | ○• •• ○• | ○• •○ ○○ | ○○ ○○ •○ | |

Gambar 2.2 Huruf Hijaiyah dalam Braille [10]

2.3 Refreshable Braille Display

Refreshable braille display adalah alat elektro-mekanis untuk menampilkan karakter braille, biasanya dengan menggunakan pin berujung bulat yang diangkat melalui lubang di permukaan yang datar [11]. Alat ini akan menampilkan karakter dari Huruf Hijaiyah dalam versi braille. Pola-pola braille yang berupa titik-titik akan ditampilkan melalui alat dalam bentuk pola timbul secara dinamis. *Refreshable braille display* yang digunakan pada Tugas Akhir ini dapat dilihat pada Gambar 2.3. Alat ini memiliki 1 buah sel penampil pola braille dengan kombinasi 6 titik. Pada bagian kanan dan kiri sel terdapat tombol yang berfungsi untuk memindahkan pola-pola braille yang sudah tersimpan pada *database* ke bagian sebelumnya ataupun ke bagian selanjutnya. Data yang ditampilkan pada alat ini juga akan dikirimkan ke perangkat *handphone* melalui komponen Bluetooth yang telah tertanam pada alat.



Gambar 2.3 Refreshable Braille Display

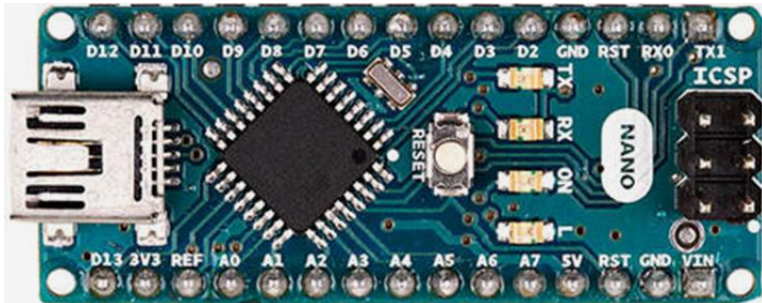
2.4 Arduino

Arduino terdiri atas dua bagian utama: *board* Arduino, yang merupakan bagian *hardware* dimana dibangun perangkat; dan Arduino IDE, bagian *software* yang dijalankan pada komputer. IDE digunakan untuk membuat *sketch* (program) dan diunggah pada *board* Arduino. *Sketch* tersebut yang akan menginstruksi board Arduino tentang apa yang harus dilakukan [12]. Adapun kelebihan Arduino dibandingkan dengan *microcontroller* lain adalah sebagai berikut [13]:

1. Keuntungan terbesar Arduino adalah struktur siap pakai. Koneksi untuk pemrograman atau antarmuka lainnya dapat dilakukan hanya dengan menyambungkan Arduino ke port USB komputer.
2. *Library* dan contoh yang ada di dalam perangkat lunak Arduino cukup lengkap dan mudah digunakan. Selain itu, Arduino memiliki mengonversi satuan secara otomatis sehingga mempermudah *debugging*.
3. Banyak forum di internet yang membahas tentang Arduino. Dengan demikian, permasalahan tentang Arduino akan mudah didapatkan solusinya. Selain itu, situs Arduino sendiri menjelaskan setiap fungsi Arduino.

2.4.1 Arduino Nano

Arduino Nano adalah papan kecil dan lengkap dengan konektivitas dan spesifikasi yang sama dengan UNO dalam bentuk yang lebih kecil. Arduino Nano juga diprogram menggunakan Arduino Software (IDE) [14].



Gambar 2.4 Arduino Nano [14]

2.5 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka [15]. Sistem operasi Android dibagi menjadi 5 *layer*, yaitu sebagai berikut [16]:

1. *Application Layer*: Bagian ini terdiri dari aplikasi pengguna. *Application layer* mencakup aplikasi dasar dari sistem operasi dan aplikasi yang ditambahkan oleh pengguna. Semua aplikasi ditulis dalam bahasa pemrograman Java.
2. *Framework*: Bagian ini merupakan perluasan dari komponen perangkat lunak yang digunakan oleh seluruh aplikasi dalam sistem operasi.
3. *Libraries*: Bagian ini ditulis dalam bahasa pemrograman C dan C++. Sistem operasi dapat mengakses bagian ini melalui *framework*.

4. *Dalvik Virtual Machine* (DVM): Bagian ini merupakan bagian utama dari *environment* sistem. Mesin virtual digunakan untuk memulai *library* utama yang ditulis dalam bahasa pemrograman Java. Tidak seperti mesin virtual dari Java yang berdasarkan *stack*, DVM berbasis pada struktur *registry* dan dirancang untuk *mobile*.
5. *Linux Kernel*: Bagian ini merupakan bagian yang bertindak sebagai *hardware abstraction layer*. Alasan utama penggunaannya adalah untuk mengatur dan memproses memori, model keamanan, sistem jaringan, dan pengembangan sistem.

2.6 Retrofit

Retrofit merupakan *library* android yang dibuat oleh Square yang digunakan sebagai REST Client pada Android. Retrofit memungkinkan tidak perlu lagi untuk membuat *method-method* sendiri untuk menggunakan REST Client API dari *backend*. *Library* ini menyediakan *framework* yang baik untuk otentikasi dan berinteraksi dengan API dengan mengirimkan *request* menggunakan HTTP. Retrofit mem-*parsing* JSON *response* dari Web API menjadi Plain Old Java Object (POJO) sehingga akan memudahkan menggunakan nilai yang berada dalam *response* karena sudah otomatis terkonversi menjadi obyek [17].

2.7 Pengenalan Suara

Pengenalan lafal/ucapan adalah suatu proses untuk mengenali ucapan dari seseorang. Secara umum, tahap pengenalan ucapan dibagi menjadi dua bagian, yakni tahap pembelajaran pola yang meliputi pengekstraksi ciri dan pembelajaran pola serta tahap pengenalan suara yang meliputi pengekstraksi ciri, pembelajaran pola, perbandingan pola dengan model, dan pengambilan keputusan [18]. Terdapat lima pendekatan yang dapat dilakukan untuk melakukan pengenalan ucapan, yaitu sebagai berikut [19]:

1. *Template Based Approach*. Pada pendekatan ini, input ucapan yang ada dibandingkan dengan *database* dari pola ucapan yang ada untuk mendapatkan hasil yang paling cocok.
2. *Knowledge Based Approach*. Pendekatan ini mengenali ucapan dengan proses pembelajaran pada sistem. Fitur-fitur pada ucapan diambil dan kemudian dilatih pada sistem untuk menghasilkan *rule* secara otomatis dari sampel.
3. *Neural Network Based Approach*. *Neural Network* memiliki kemampuan untuk menyelesaikan pengenalan yang rumit, namun tidak sebaik *Hidden Markov Model* (HMM) untuk kosakata dengan jumlah yang besar. Pendekatan ini dapat menyelesaikan data kualitas rendah, data dengan *noise*, dan pengenalan pembicara. Tipe ini dapat memperoleh akurasi yang lebih baik dari pada HMM jika ukuran data latih dan kosakata terbatas.
4. *Dynamic Time Warping Based Approach*. Pendekatan ini menghitung kemiripan antara dua sekuens yang mungkin berbeda dalam waktu atau kecepatan. Teknik ini berguna untuk pengenalan kata terisolasi dan dapat dimodifikasi untuk mengenali kata-kata yang terhubung juga.
5. *Statistical Based Approach*. Pada pendekatan ini, variasi ucapan dimodelkan secara statistik menggunakan metode pelatihan.

2.8 *Voice Activity Detection (VAD)*

Voice Activity Detection (VAD) merupakan algoritma yang dapat mendeteksi ada atau tidaknya ucapan. Masalah penting dalam aplikasi pengolahan ucapan adalah penentuan periode ucapan aktif dalam sinyal audio yang diberikan. Ucapan dapat dicirikan oleh sinyal yang diskontinu karena informasi diambil hanya ketika seseorang sedang berbicara. Daerah dimana informasi suara ada disebut sebagai segmen '*voice-active*' dan jeda antara berbicara disebut segmen '*voice-inactive*' atau '*silence*'. Penentuan kelas dari segmen audio didasarkan pada vektor observasi. Hal ini umumnya disebut dengan fitur vektor. Energi sinyal merupakan

komponen dasar fitur vektor [20]. Energi sinyal dapat memberikan representasi untuk perubahan amplitudo sinyal suara terhadap waktu [21]. Secara umum, energi dari *frame* suatu sinyal ditunjukkan pada Persamaan (2.1).

$$E_n = \frac{1}{N} \sum_{i=1}^N [x(i)]^2, \quad 1 \leq i \leq N \quad (2.1)$$

Dimana, E_n adalah energi sinyal, $x(i)$ adalah sinyal suara pada detik ke- i dan N adalah banyaknya sampel dalam *frame*.

2.9 Mel-Frequency Cepstrum Coefficients (MFCC)

Salah satu teknik ekstraksi ciri yang telah luas dipakai pada pemrosesan sinyal suara adalah MFCC. Penggunaan teknik ini pada sistem pemrosesan sinyal memberikan pengenalan yang lebih baik dibandingkan dengan metode lain yang sudah ada [22]. MFCC merupakan metode yang paling dikenal dan paling populer. Tujuan utama dari proses MFCC adalah untuk meniru perilaku telinga manusia. Gambar 2.5 menunjukkan tahapan dari MFCC.



Gambar 2.5 Blok Diagram Proses MFCC

Teknik ekstraksi fitur MFCC pada dasarnya terdiri dari *windowing*, *Discrete Fourier Transform* (DFT), mengambil log dari nilai dan *warping* frekuensi dari skala Mel, yang diikuti dengan mengaplikasikan inverse dari *Discrete Cosine Transform* (DCT). Deskripsi detail dari langkah-langkah ekstraksi fitur MFCC adalah sebagai berikut [23]:

1. *Pre-Emphasis*: *Pre-Emphasis* mengacu untuk melakukan *filter* yang menekankan pada frekuensi tinggi. Tujuannya adalah untuk menyeimbangkan spektrum dari suara yang memiliki penurunan pada daerah frekuensi tinggi. Saat mengeluarkan suara dari glotis, terjadi penurunan kurang

lebih -12 dB/oktaf. Namun, ketika energi akustik keluar dari bibir, hal ini menyebabkan sekitar +6 dB/oktaf ditambahkan pada spektrum. Sebagai hasilnya, ketika sinyal suara direkam dengan menggunakan mikropon dari suatu jarak terjadi penurunan sekitar -6 dB/oktaf dibandingkan dengan spektrum yang benar. Untuk itu, *pre-emphasis* menghilangkan beberapa efek glotis dari parameter saluran suara. Persamaan (2.2) menunjukkan *pre-emphasis filter* yang umum digunakan.

$$H(z) = 1 - bz^{-1} \quad (2.2)$$

Dimana, b adalah kontrol dari penurunan *filter* dan biasanya memiliki nilai antara 0.4 dan 0.1

2. *Frame blocking* dan *windowing*: Sinyal ucapan adalah sinyal yang *quasi-stationary*. Untuk karakteristik akustik yang stabil, ucapan harus diuji pada periode waktu yang pendek. Untuk itu analisa ucapan harus selalu dilakukan pada segmen yang pendek dimana sinyal suara diasumsikan stasioner. Pengukuran spektral jangka pendek biasanya dilakukan pada 20 ms *window*, dan dimajukan setiap 10 ms. Hal ini dilakukan supaya karakteristik dari ucapan individu memungkinkan untuk dilacak. Pemilihan analisa tiap 20 ms *windows* dilakukan supaya menghasilkan resolusi spektral yang baik. Tujuan dari analisa yang *overlap* ini supaya ucapan dari sekuen input dapat dipusatkan pada beberapa *frame*. Setiap *frame* diaplikasikan untuk meruncingkan sinyal pada batas yang *frame*. Secara umum, Hanning atau Hamming *window* digunakan untuk ini. Hal ini digunakan untuk meningkatkan harmonik, menghaluskan tepi, dan untuk mengurangi efek dari tepi saat mengambil DFT dari sinyal.
3. Spektrum DFT: Setiap *windowed frame* dikonversi kedalam nilai spektrum dengan mengaplikasikan DFT menggunakan Persamaan (2.3).

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}; 0 \leq k \leq N-1 \quad (2.3)$$

Dimana, N adalah banyaknya titik yang digunakan untuk mengkomputasikan DFT.

4. Spektrum Mel: Spektrum Mel dikomputasikan dengan melewati transformasi Fourier dari sinyal melewati *band-pass filter* yang dikenal sebagai *mel-filter bank*. Mel adalah unit dari pengukuran yang didasarkan pada frekuensi yang dirasakan oleh telinga manusia. Unit tersebut tidak sesuai secara linier dengan frekuensi fisik dari suatu bunyi, karena sistem pendengaran manusia tidak merasakan nada secara linier. Skala mel berada di sekitar frekuensi linier < 1 kHz dan logaritmik > 1 kHz. Perkiraan dari skala mel dari frekuensi fisik ditunjukkan oleh Persamaan (2.4).

$$f_{mel} = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (2.4)$$

Dimana, f menotasikan frekuensi fisik dalam Hz, dan f_{mel} menotasikan frekuensi yang dirasakan.

Filter bank bisa diimplementasikan pada domain waktu dan domain frekuensi. Untuk komputasi MFCC, *filter bank* secara umum diimplementasikan pada domain frekuensi seperti yang dapat dilihat pada Gambar 2.6. Pusat dari frekuensi dari filter normalnya berada pada sumbu frekuensi. Namun, supaya dapat meniru telinga manusia, sumbu yang berdasar fungsi non-linier pada Persamaan (2.4) diimplementasikan.

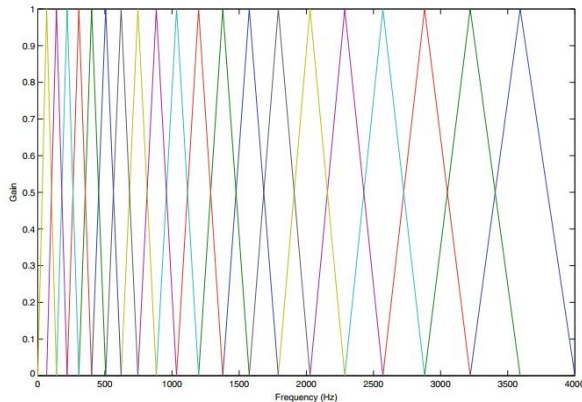
Spektrum mel dari nilai spektrum $X(k)$ dikomputasikan dengan mengalikan nilai spektrum dengan setiap *filter* segitiga mel yang diberi bobot menggunakan Persamaan (2.5).

$$s(m) = \sum_{k=0}^{N-1} [|X(k)|^2 H_m(k)]; 0 \leq m \leq M - 1 \quad (2.5)$$

Dimana, M adalah total dari banyaknya *filter* segitiga mel yang diberi bobot. $H_m(k)$ adalah bobot yang diberikan pada energi spektrum *bin* ke- k yang berkontribusi pada m pita output dan diekspresikan pada Persamaan (2.6).

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{2(k - f(m-1))}{f(m) - f(m-1)}, & f(m-1) \leq k \leq f(m) \\ \frac{2(f(m+1) - k)}{f(m+1) - f(m)}, & f(m) < k \leq f(m+1) \\ 0, & k > f(m+1) \end{cases} \quad (2.6)$$

Dimana, m antara 0 sampai dengan $M - 1$



Gambar 2.6 Mel Filter Bank

5. *Discrete Cosine Transform (DCT)*: Jika saluran suara sudah halus, energi level dari pita yang bertetangga cenderung terkolerasi. DCT diaplikasikan pada koefisien frekuensi mel yang tertransformasi dan menghasilkan *cepstral* koefisien. Sebelum mengkomputasi DCT, spektrum mel biasanya direpresentasikan pada skala log. Hasil pada sinyal dari domain *cepstral* dengan puncak *que-frequency* yang berkorespondensi dengan nada dari sinyal dan jumlah formant yang merepresentasikan puncak *que-frequency*. Kebanyakan informasi dari sinyal direpresentasikan dengan beberapa koefisien MFCC yang pertama sehingga sistem bisa dibuat dengan hanya mengekstraksi koefisien tersebut dengan mengabaikan atau memotong urutan komponen DCT yang lebih tinggi. MFCC dikalkulasikan dengan Persamaan (2.7).

$$c(n) = \sum_{m=0}^{M-1} \log_{10}(s(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad (2.7)$$

$; n = 0, 1, 2, \dots, C - 1$

Dimana, $c(n)$ adalah koefisien *cepstral* dan C adalah banyaknya MFCC. Biasanya MFCC hanya menggunakan 8-13 koefisien *cepstral*. Koefisien ke-0 sering diabaikan karena merepresentasikan energi log dari sinyal input, dimana hanya membawa sedikit informasi spesifik.

2.10 Neural Network

Neural Network terdiri dari elemen-elemen sederhana yang meniru sistem saraf biologis manusia. Karakteristik yang paling menguntungkan dari *Neural Network* untuk pengenalan suara adalah toleransi kesalahan dan properti non linear [24]. *Neural Network* dapat dilatih untuk melakukan fungsi tertentu dengan cara mengatur nilai-nilai dari koneksi (bobot) antar elemen. Secara umum, *Neural Network* dilatih agar input tertentu menghasilkan output target tertentu.

Beberapa algoritma pelatihan *Neural Network* diantaranya adalah *Back Propagation*, *Learning Vector Quantization*, *Radial*

Base Fuction, Hopfield, dan Kohonen. Algoritma Neural Network yang akan digunakan adalah Back Propagation dimana algoritma ini merupakan algoritma yang paling banyak digunakan untuk supervised Neural Network [8]. Arsitektur jaringannya tidak memiliki koneksi umpan balik tetapi error dipropagasikan selama dilatih. Pada pembelajaran ini digunakan error dengan perhitungan Mean Square Error (MSE). Algoritma pelatihan Back Propagation Neural Network terdiri dari dua tahap, yaitu feed forward dan backward propagation. Penjelasan mengenai tahap tersebut adalah sebagai berikut [25]:

a. Tahap *Feed Forward*

Pada tahap *feed forward*, awalnya dilakukan dengan memberikan input dan targetnya. Kemudian melakukan pengaturan bobot (*weighth*) secara random dan dilakukan perhitungan untuk mendapatkan nilai output.

Masing-masing unit lapisan dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan bias. Perhitungan untuk masing-masing unit ditunjukkan oleh Persamaan (2.8).

$$z = b + \sum_{i=1}^n X_i W_{ij} \quad (2.8)$$

Kemudian dihitung dengan fungsi aktivasi pada Persamaan (2.9).

$$Y = f(z) \quad (2.9)$$

Kemudian dihitung pada fungsi sigmoid pada Persamaan (2.10).

$$f(z) = \frac{1}{1 + \exp(-z)} \quad (2.10)$$

Sinyal keluaran dari fungsi aktifasi tersebut dikirimkan ke semua unit di lapisan luar. Tiap-tiap output layer dikalikan dengan

bobot dan dijumlahkan dengan bias serta dihitung lagi fungsi aktifasinya. Proses yang sama dilakukan hingga layer output.

b. Tahap *Backward*

Tahap *backward* merupakan proses perhitungan error dan mengatur ulang bobot tiap-tiap layer. Setelah diperoleh nilai output pada proses *feed forward*, maka akan dilakukan proses perhitungan *error* untuk kemudian nilai *error* tersebut akan digunakan sebagai variabel untuk meng-*update* bobot tiap-tiap layer. Jika ada selisih antara output yang diharapkan dengan output sebenarnya (*error*), maka akan disebarkan mundur (*backward*) pada tiap layernya. Oleh karena itu, proses tersebut disebut sebagai propagasi balik (*back propagation*) dimana tahap pelatihan dilakukan dengan mengubah bobot yang menghubungkan unit dalam layer *Neural Network*. Masing-masing output layer menerima pola target sesuai dengan pola masukan saat pelatihan. Perhitungan untuk proses ini dapat dilihat pada Persamaan (2.11) dan Persamaan (2.12).

$$\Delta = (Y_{target} - Y)f'(z) \quad (2.11)$$

$$f'(z) = f(z)(1 - f(z)) = Y(1 - Y) \quad (2.12)$$

Delta merupakan selisih target yang ingin dicapai dengan nilai keluaran dikalikan dengan turunan dari fungsi aktivasi yang digunakan. Delta dapat dihitung menggunakan Persamaan (2.11). Sedangkan turunan fungsi aktivasi dapat dicari dengan menggunakan Persamaan (2.12). Masing-masing output layer diperbaiki bias dan bobotnya. Nilai pembaruan bobot dan bias dapat dihitung dengan Persamaan (2.13) dan Persamaan (2.14).

$$W_{1,n}(baru) = W_{1,n}(lama) + \Delta\mu x_i \quad (2.13)$$

$$b_{1,n}(baru) = b_{1,n}(lama) + \Delta\mu \quad (2.14)$$

Pada akhirnya, nilai pembaharuan bobot dan bias ini yang dijadikan sebagai acuan pada proses pengujian. Proses pengujian sendiri merupakan proses penggolongan pola baru berdasarkan sampel pola yang telah dipelajari sebelumnya.

2.10.1 Algoritma *Gradient Descent with Momentum and Adaptive Learning*

Pada standar *backpropagation*, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan, jadi tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (*outlier*). Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola yang serupa (berarti arah gradien sudah benar), maka perubahan bobot dilakukan secara cepat. Namun, apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Dengan Penambahan momentum, bobot baru pada waktu ke $(t+1)$ didasarkan atas bobot pada waktu t dan $(t-1)$. Disini harus ditambahkan dua variabel baru yang mencatat besarnya momentum untuk dua iterasi terakhir. Jika μ adalah konstanta ($0 \leq \mu \leq 1$) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan Persamaan 2.15 dan Persamaan 2.16.

$$w_{kj}(t+1) = w_{kj}(t) + \alpha \delta_k z_j + \mu(w_{kj}(t) - w_{kj}(t-1)) \quad (2.15)$$

$$v_{ji}(t+1) = v_{ji}(t) + \alpha \delta_j x_i + \mu(v_{ji}(t) - v_{ji}(t-1)) \quad (2.16)$$

Algoritma *gradient descent* dengan *adaptive learning rate*, dasarnya sama dengan algoritma *gradient descent* standar dengan beberapa perubahan. Pertama-tama dihitung terlebih dahulu nilai

output jaringan dan error pelatihan. Pada setiap *epoch*, bobot-bobot baru dihitung dengan menggunakan *learning rate* yang ada. Kemudian dihitung kembali *output* jaringan dan *error* pelatihan. Jika perbandingan antara *error* pelatihan yang baru dengan *error* pelatihan lama melebihi maksimum kenaikan kinerja, maka bobot-bobot baru tersebut akan diabaikan, sekaligus nilai *learning rate* akan dikurangi dengan cara mengalikannya dengan rasio untuk menurunkan *learning rate*. Sebaliknya, apabila perbandingan antara *error* pelatihan baru dengan *error* pelatihan lama kurang dari maksimum kenaikan kinerja, maka nilai bobot-bobot akan dipertahankan, sekaligus nilai *learning rate* akan dinaikkan dengan cara mengalikannya dengan rasio untuk menaikkan *learning rate*.

Dengan cara ini, apabila *learning rate* terlalu tinggi dan mengarah ke ketidakstabilan, maka *learning rate* akan diturunkan. Sebaliknya jika *learning rate* terlalu kecil untuk menuju konvergensi, maka *learning rate* akan dinaikkan. Dengan demikian, maka algoritma pembelajaran akan tetap terjaga pada kondisi stabil.

Algoritma *gradient descent with momentum and adaptive learning* (traingdx) merupakan penggabungan antara algoritma *gradient descent with adaptive learning* (traingda) dan algoritma *gradient descent with momentum* (traingdm). Algoritma ini merupakan algoritma *default* yang digunakan oleh MATLAB karena memiliki performa kecepatan pelatihan yang tinggi [26].

2.11 Confusion Matrix

Confusion matrix adalah suatu metode yang biasa digunakan untuk melakukan perhitungan akurasi pada konsep data mining. *Confusion Matrix* memiliki informasi hasil prediksi dan aktual pada data yang telah di klasifikasi. Performa suatu sistem biasanya dievaluasi menggunakan *Confusion Matrix*. Ciri dari *Confusion Matrix* sendiri adalah kolom mempresentasikan kelas prediksi, sedangkan baris mempresentasikan kelas sebenarnya atau sebaliknya. Tabel 2.2 menunjukkan *Confusion Matrix* pada dua kelas.

Tabel 2.2 Confusion Matrix

| | | PREDIKSI | |
|--------|---|----------|----|
| | | 1 | 0 |
| AKTUAL | 1 | TP | FN |
| | 0 | FP | TN |

Nilai yang bisa dihitung menggunakan *Confusion Matrix* yaitu akurasi, sensitivitas dan spesifisitas. Akurasi merupakan hasil bagi dari jumlah prediksi yang terklasifikasi secara benar dibagi total data yang diklasifikasi seperti pada persamaan (2.17).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.17)$$

Sensitivitas adalah perbandingan dari jumlah data TP dengan total data TP dan FN seperti pada persamaan (2.18).

$$Sensitivitas = \frac{TP}{TP + FN} \quad (2.18)$$

Spesifisitas adalah perbandingan dari jumlah data TN dengan total data TN dan FP seperti pada persamaan (2.19).

$$Spesifisitas = \frac{TN}{TN + FP} \quad (2.19)$$

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN SISTEM

Bab ini menjelaskan tentang perancangan dan pembuatan sistem perangkat lunak. Sistem perangkat lunak yang dibuat pada Tugas Akhir ini adalah mengolah data sinyal suara Huruf Hijaiyah dari aplikasi Android yang telah terintegrasi dengan *refreshable barille display* dengan metode *Mel-Frequency Cepstrum Coefficients* (MFCC) yang mendekomposisi sinyal menjadi beberapa fitur sebagai ciri dari data masukan pada proses klasifikasi. *Back Proagation Neural Network* merupakan metode yang digunakan untuk melakukan klasifikasi. Pada bab ini pula akan dijelaskan gambaran umum sistem dalam bentuk *flowchart*.

3.1 Data

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan.

3.1.1 Data Masukan

Data masukan pada bagian ini adalah data yang digunakan sebagai masukan awal dari sistem untuk bagian *software*. Data yang digunakan dalam Aplikasi Pengenalan Ucapan Huruf Hijaiyah Berbasis Android untuk *Refreshable Braille Display* Menggunakan *Mel-Frequency Cepstrum Coefficients* dan *Neural Network* adalah data sinyal suara dari 29 Huruf Hijaiyah yang direkam melalui aplikasi Android. Dataset terdiri dari data rekaman suara sebanyak sepuluh orang yang terdiri dari lima orang laki-laki dan lima orang perempuan. Setiap orang melakukan perekaman sebanyak lima kali terhadap setiap Huruf Hijaiyah yang diucapkan, sehingga keseluruhan data berjumlah 1450 sinyal suara dimana masing-masing terdapat 725 data sinyal suara untuk laki-laki dan perempuan.

Untuk proses pengenalan ucapan dan klasifikasi, data suara laki-laki dipisahkan dengan data suara perempuan sehingga proses berjalan masing-masing. Pada setiap prosesnya, data dibagi menjadi 2 bagian, dimana sebanyak 80% data ditentukan menjadi *data training* dan 20% data menjadi *data testing*. Pada data perempuan, *data training* terdiri atas data suara dari 29 Huruf Hijaiyah sebanyak 4 ulangan dari 5 orang responden perempuan. Sedangkan untuk data testing terdiri dari 29 Huruf Hijaiyah sebanyak 1 ulangan dari 5 orang responden perempuan. Begitu pula untuk data laki-laki, *data training* terdiri atas data suara dari 29 Huruf Hijaiyah sebanyak 4 ulangan dari 5 orang responden laki-laki. Sedangkan untuk data testing terdiri dari 29 Huruf Hijaiyah sebanyak 1 ulangan dari 5 orang responden laki-laki.

3.1.2 Data Proses

Data yang telah dibagi menjadi dua data yaitu data pembelajaran atau *training* dan data pengujian atau *testing* akan diproses menggunakan *Mel-Frequency Cepstrum Coefficients* (MFCC) dan *Neural Network*. Selanjutnya, program yang dikembangkan dapat mengeluarkan hasil dari proses klasifikasi data testing berupa hasil akurasi, sensitivitas, dan spesivitas.

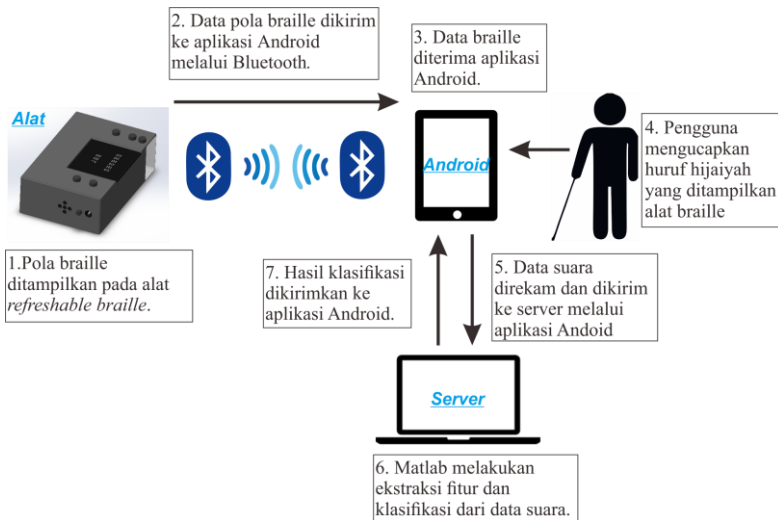
3.1.3 Data Keluaran

Data keluaran adalah data yang dihasilkan dari pemrosesan data masukan. Pada Tugas Akhir ini, akan ada dua jenis data keluaran, yaitu data keluaran masing-masing pemilihan metode dan data keluaran dari aplikasi yang telah dibangun. Data keluaran yang dihasilkan saat pemilihan metode terbaik berupa akurasi tiap kelas dan akurasi rata-rata dalam satu percobaan. Data keluaran pada aplikasi yang telah dibangun berupa hasil klasifikasi jenis suara pengucapan Huruf Hijaiyah berupa teks.

3.2 Desain Umum Keseluruhan Sistem

Tugas Akhir ini terdiri dari dua bagian, yaitu bagian *hardware* dan bagian *software* pada Android dan MATLAB. Bagian *hardware* berupa *refreshable braille display* yang akan menampilkan pola braille pada alat sedangkan bagian *software* bertugas melakukan proses pengenalan suara, dimana pada aplikasi Android akan memproses data yang diterima dari *hardware* serta melakukan perekaman suara pengguna dan MATLAB bertugas melakukan ekstraksi fitur dan klasifikasi. Pada Tugas Akhir ini, terdapat hubungan antara *hardware* dengan aplikasi Android serta aplikasi Android dengan *server* yang akan memanggil MATLAB sehingga dibutuhkan komunikasi antara keduanya. Komunikasi data antara *hardware* dengan aplikasi dilakukan oleh media Bluetooth sedangkan komunikasi antara Android dengan server menggunakan internet.

Alur dari sistem ini yaitu diawali saat *hardware* menampilkan pola braille dari Huruf Hijaiyah tertentu. Di saat yang bersamaan, data Huruf Hijaiyah yang ditampilkan tersebut dikirim ke aplikasi Android melalui media Bluetooth. Aplikasi Android akan menerima data dari pola yang ditampilkan pada *hardware*, kemudian pengguna melakukan pengucapan terhadap huruf yang tampil pada *hardware*. Saat data suara dari pengguna sudah terekam, maka aplikasi akan mengunggah data suara tersebut pada *server* untuk kemudian diproses supaya dapat dicocokkan dengan pola pada data latih untuk dilakukan klasifikasi menggunakan MATLAB. Setelah didapatkan hasil dari proses klasifikasi, maka server akan mengirimkan hasilnya pada aplikasi Android sehingga pengguna dapat mengetahui benar/tidaknya pengucapan Huruf Hijaiyah. Cara kerja dari keseluruhan sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain Umum Alur Sistem

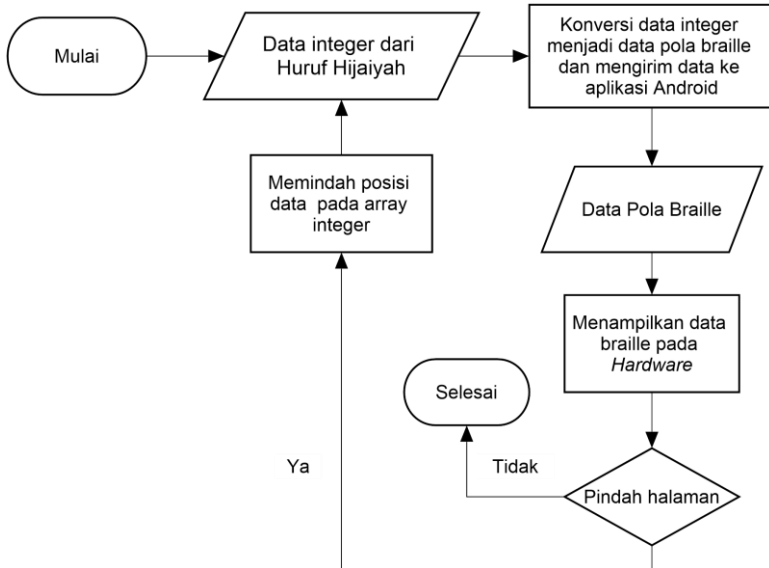
3.2.1 Desain Umum Aplikasi pada *Hardware*

Untuk menjalankan *hardware (refreshable braille display)*, dibutuhkan aplikasi yang tertanam didalamnya. Aplikasi tersebut dibangun dengan menggunakan Arduino IDE yang kemudian disimpan pada komponen Arduino sebagai otak utama *hardware*. Pada aplikasi ini, data masukan didideklarasikan pada program secara langsung berupa representasi Huruf Hijaiyah yang ditulis sesuai dengan urutannya dalam bentuk data *array* dengan tipe integer yang dapat dilihat pada Gambar 3.2.

```
int pattern [ROW] [COLUMN] = {
    {1}, {2}, {3}, {4}, {5},
    {6}, {7}, {8}, {9}, {10},
    {11}, {12}, {13}, {14}, {15},
    {16}, {17}, {18}, {19}, {20},
    {21}, {22}, {23}, {24}, {25},
    {26}, {27}, {28}, {29},
};
```

Gambar 3.2 Data untuk Arduino

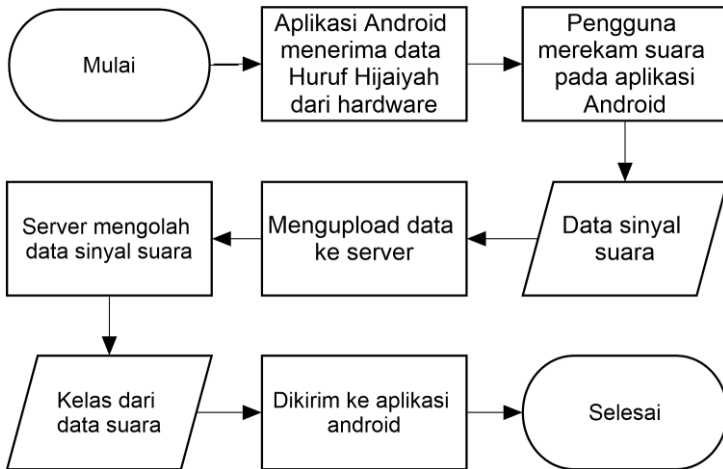
Aplikasi akan mengubah data integer tersebut menjadi pola braille yang ditampilkan pada *hardware*. Selain itu, aplikasi juga akan mengirim data tersebut ke aplikasi di bagian Android. Alur proses tersebut dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Alir Sistem pada *Refreshable Braille Display*

3.2.2 Desain Umum Aplikasi pada Android

Aplikasi Android bertugas untuk menerima data dari *hardware* berupa data integer dari Huruf Hijaiyah yang ditampilkan. Selain itu, pada aplikasi ini, juga dilakukan input terhadap data suara dengan melakukan perekaman. Hasil rekaman suara disimpan dan dikirimkan ke *server* bersamaan dengan data integer Huruf Hijaiyah. Setelah data diproses di bagian *server*, hasil akan dikirim ke aplikasi Android untuk ditampilkan kepada pengguna. Alur proses tersebut dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Alir Sistem pada Android

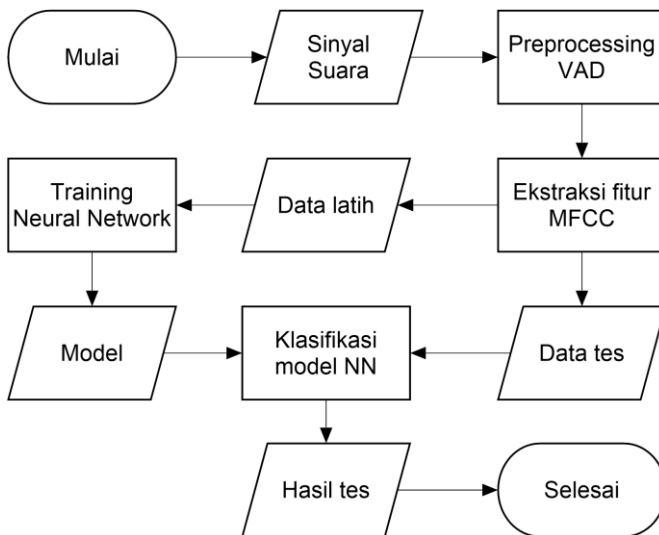
3.2.3 Desain Umum Aplikasi pada MATLAB

Aplikasi pada MATLAB dibangun pada bagian *server*. Aplikasi tersebut berupa Aplikasi Pengenalan Ucapan Huruf Hijaiyah Berbasis Android untuk *Refreshable Braille Display* Menggunakan *Mel-Frequency Cepstrum Coefficients* dan *Neural Network*. Pada aplikasi ini, pengenalan ucapan Huruf Hijaiyah disesuaikan dengan pola yang ditampilkan pada alat *refreshable braille*.

Awalnya, data sinyal suara pada aplikasi Android yang direkam pada *sampling rate* 44100 Hz dan disimpan dalam format *.wav* diterima oleh *server*. Sinyal suara yang berada pada *server* diekstraksi fiturnya menggunakan MFCC. Sebelum dilakukan ekstraksi fitur, suara terlebih dahulu melalui tahap *preprocessing* menggunakan *Voice Activity Detection (VAD)* untuk dipotong bagian sinyal supaya yang diolah kemudian adalah data sinyal bernilai informasi saja karena bagian heningnya sudah dihilangkan melalui proses ini.

Ekstraksi fitur sinyal suara dengan teknik MFCC menghasilkan 13 fitur frekuensi mel untuk setiap *frame* dari sinyal suara. Setelah itu, data tersebut menjadi input untuk proses klasifikasi menggunakan *Back Propagation Neural Network*. Pada klasifikasi ini, digunakan fungsi pelatihan *Variable Learning Rate Gradient Descent* dengan 1 buah *hidden layer*.

Untuk melakukan klasifikasi, maka sebelumnya data dibagi menjadi data latih dan data tes. Data latih akan diproses untuk diekstraksi fiturnya. Selanjutnya, hasil dari proses tersebut akan menjadi model untuk proses klasifikasi. Data tes juga akan diekstraksi fiturnya untuk kemudian akan menjadi input dalam proses klasifikasi. Alur keseluruhan proses tersebut dapat dilihat pada Gambar 3.5.



Gambar 3.5 Diagram Alir Sistem pada MATLAB

3.2.3.1 Tahap *Voice Activity Detection* (VAD)

Sinyal suara mentah diproses menggunakan *Voice Activity Detection* (VAD) untuk memisahkan sinyal suara berisi informasi

dengan sinyal jeda atau *silent*. Masukan dari pemrosesan VAD adalah sinyal suara sedangkan keluarannya adalah nilai 1 atau 0. Pemrosesan VAD dilakukan dengan menggunakan fungsi `vadG729` pada MATLAB.

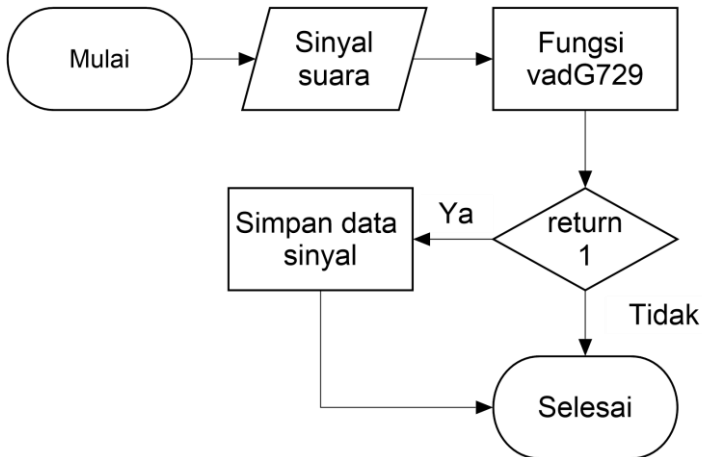
Pada fungsi `vadG729` tersebut, pada awalnya sinyal akan dilakukan preprocessing terlebih dahulu menggunakan *highpass* filter dan *linear predictive analysis*. Algoritma *highpass* filter yang digunakan adalah biquad filter.

Selanjutnya akan diekstraksi perhitungan sinyal suara input menjadi 4 buah fitur, yaitu batas atas dan batas bawah energi dari *frame* sinyal, *Line Spectral Frequencies* (LSF), dan *zero crossing rate*. VAD akan *return* 1 dan jika hasil analisa LPC dari energi *frame* diatas 21 dB. Sebaliknya, jika hasilnya dibawah 21 dB, maka VAD akan *return* 0.

Pada tahap selanjutnya, parameter yang berbeda akan dihitung. Pada perhitungan ini dihasilkan perbedaan antara karakter dari parameter *frame* sinyal dengan *background noise*. Parameter yang dihitung tersebut adalah distorsi spektral, perbedaan energi, perbedaan batas bawah energi, dan perbedaan *zero-crossing*. Apabila terdapat *noise*, maka nilai dari *running average* sinyal akan diperbarui.

Hasil dari VAD akan dibuat menggunakan empat parameter tersebut pada *multi-boundary decision regions*. Keputusan suatu *frame* sinyal dinilai sebagai suatu informasi atau tidak didasarkan pada energi suatu *frame* dan juga keputusan dari *frame* sebelumnya yang berguna untuk memperhalus hasil keputusan dari proses VAD. Pada VAD ini juga diaplikasikan *adaptive threshold*.

Hasil dari proses VAD adalah keputusan bernilai 1 atau 0. Jika bernilai 1, maka sinyal suara berisi informasi. Jika bernilai 0, maka sinyal suara tidak bernilai informasi. Data dengan nilai 1 disimpan dalam *array* sebagai data sinyal suara baru tanpa bagian hening. Alur keseluruhan proses tersebut dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Alir Proses VAD

3.2.3.2 Tahap Ekstraksi Fitur dengan *Mel-Frequency Cepstrum Coefficients* (MFCC)

Ekstraksi fitur digunakan untuk mendapat ciri dari data sinyal suara yang berguna untuk tahap selanjutnya, yaitu klasifikasi. Input dari proses ini merupakan data sinyal suara hasil dari proses VAD, yaitu berupa data sinyal suara bagian informasi saja (yang telah dihilangkan bagian heningnya). Sedangkan output dari proses ini adalah berupa 13 koefisien MFCC yang akan menjadi fitur untuk tahap klasifikasi. Proses MFCC sendiri terdiri dari beberapa tahap yang dapat dilihat pada Gambar 3.7, yaitu:

1. *Pre-emphasize* untuk memperbaiki sinyal dari *noise*. Filter ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara. Perhitungan untuk melakukan filter dapat dilihat pada Persamaan (2.2) dengan nilai koefisien α sebesar 0.95.
2. *Frame blocking* untuk membagi sampel sinyal menjadi beberapa *frame*. Proses ini dilakukan karena sinyal suara terus mengalami perubahan akibat adanya pergeseran artikulasi

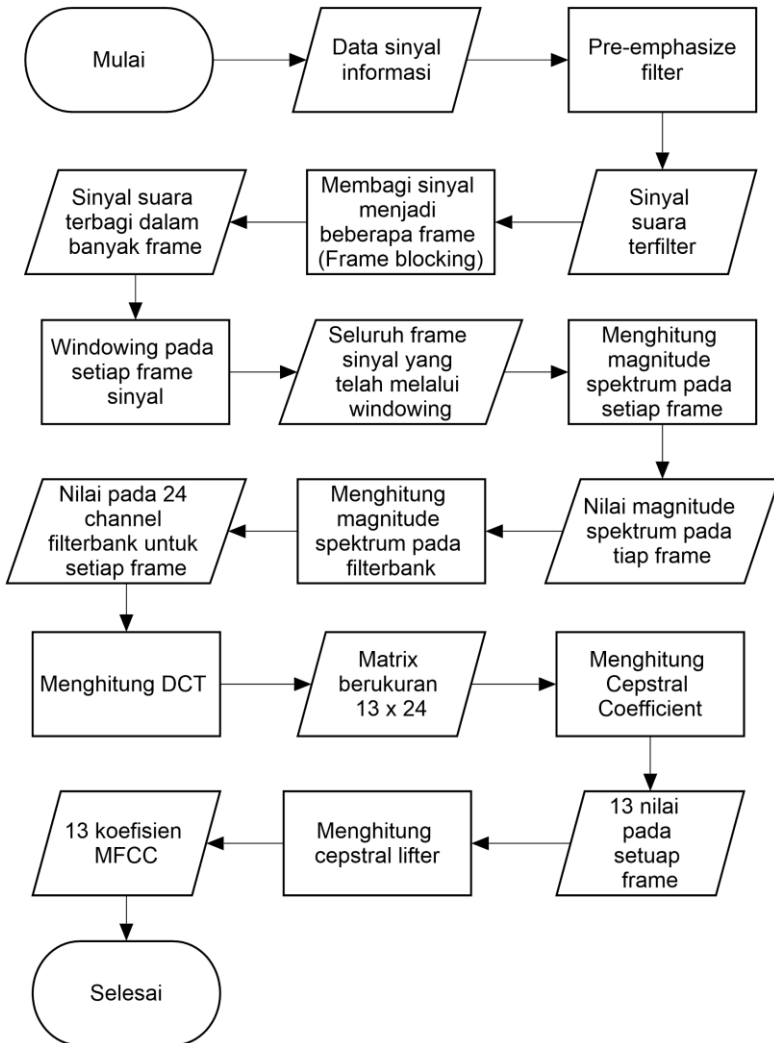
dari organ produksi vokal sehingga sinyal harus diproses pada segmen pendek (*short frame*). Panjang *frame* pemotongan sinyal yang digunakan pada proses ini adalah sebesar 20 ms. Panjang *frame* yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Proses *frame blocking* ini dilakukan pada keseluruhan panjang sinyal. Pada proses ini, sinyal disusun secara *overlapping* pada setiap *frame*. Panjang daerah *overlap* yang digunakan adalah 25% dari panjang *frame*. *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame*.

3. *Windowing* untuk mengurangi efek diskontinuitas pada ujung-ujung *frame*. Proses *framing* dapat menyebabkan terjadinya kebocoran spektral (*spectral leakage*) atau *aliasing*. *Aliasing* adalah *signal* baru dimana memiliki frekuensi yang berbeda dengan *signal* aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate*, ataupun karena proses *frame blocking* dimana menyebabkan *signal* menjadi *discontinue*. Untuk mengurangi kemungkinan terjadinya hal tersebut, maka hasil dari proses *framing* harus melewati proses *window*. Jenis *window* yang digunakan pada Tugas Akhir ini adalah Hamming.
4. *Magnitude* spektrum dari *Fast Fourier Transform* (FFT) untuk mendapatkan sinyal dalam domain frekuensi. Setelah didapatkan sinyal dalam domain frekuensi, maka dihitung *magnitude* spektrum dari data yang merupakan nilai absolut dari hasil perhitungan FFT.
5. *Filterbank* untuk mengetahui ukuran energi dari setiap *frekuensi band*. Pada tahap ini akan dibuat filter yang terdiri atas 24 *channel*. Titik-titik *channel* pada filter ini ditentukan menggunakan frekuensi dalam skala mel yang dihitung menggunakan Persamaan (2.4). Setelah titik-titik tersebut didapatkan, maka dikembalikan pada skala Herzt dan kemudian dibuat *filterbank* menggunakan Persamaan (2.5).

6. *Discrete Cosine Transform* (DCT) untuk mendapatkan *mel cepstrum*. DCT merupakan langkah terakhir dari proses utama MFCC yang dihitung menggunakan Persamaan (2.7). Konsep dasar dari DCT adalah mendekorelasikan *mel spectrum* sehingga menghasilkan representasi yang baik dari fitur spektral lokal. Sesuai dengan nilai yang diperoleh pada tahap *filterbank*, pada bagian ini juga diperoleh 24 nilai untuk setiap *frame* sinyal. Akan tetapi, semakin besar nilai DCT yang diambil, maka dapat menurunkan performa pengenalan suara. Nilai DCT kemudian dikalikan dengan log dari proses *filterbank* sehingga didapatkan nilai fitur dari MFCC. Pada umumnya, nilai yang diambil sebagai hasil fitur dari MFCC adalah sebanyak 2-13. Pada Tugas Akhir ini, diambil fitur sebanyak 13 koefisien MFCC.
7. *Cepstral Liftering* untuk menghaluskan spektrum sinyal. Setelah proses DCT, pada dasarnya sudah dapat diambil fitur dari MFCC. Namun, hasil dari proses utama MFCC tersebut memiliki beberapa kelemahan. *Low order* dari *cepstral coefficients* sangat sensitif terhadap *spectral slope*, sedangkan bagian *high order*nya sangat sensitif terhadap *noise*. Oleh karena itu, *cepstral liftering* menjadi salah satu standar teknik yang diterapkan untuk meminimalisasi sensitifitas tersebut. *Cepstral liftering* dapat dilakukan dengan mengimplementasikan fungsi *window* terhadap *cepstral features*.

$$W[n] = \begin{cases} 1 + \frac{L}{2} \sin\left(\frac{n\pi}{L}\right) & n = 1, 2, \dots, L \\ 0 & \end{cases} \quad (3.1)$$

Dimana, L adalah jumlah *cepstral coefficients* dan N adalah *index* dari *cepstral coefficients*. *Cepstral liftering* menghaluskan spektrum hasil dari *main processor* sehingga dapat digunakan lebih baik untuk *pattern matching*.



Gambar 3.7 Diagram Alir Proses MFCC

Pada Tugas Akhir ini, ekstraksi fitur MFCC dilakukan dengan menggunakan MATLAB. Adapun beberapa parameter yang diperhatikan untuk ekstraksi fitur ini yaitu:

- *Sampling rate* sebesar 44100 Hz.
- *Filterbank channel* sebanyak 24.
- *Cepstral coefficients* sebanyak 13.

3.2.3.3 Tahap Klasifikasi *Neural Network*

Data hasil ekstraksi fitur disimpan pada *file* dengan format .csv yang dibedakan menjadi dua bagian, yaitu data latih dan data tes. Selain menyiapkan data tersebut, untuk melakukan klasifikasi juga disiapkan target yang merupakan kelas dari data latih dan data tes. Semua data tersebut dibaca dan disimpan pada variabel yang bertindak sebagai data input. Setelah proses klasifikasi selesai maka akan menghasilkan output berupa kelas dari data tes. Pada Tugas Akhir ini, klasifikasi dilakukan dengan menggunakan MATLAB. Proses tersebut diawali dengan *me-load* data hasil ekstraksi dan targetnya. Kemudian, menginisiasi nilai random untuk bobot awal. Sebelum dapat digunakan, *network* pada klasifikasi *Neural Network* harus dibangun terlebih dahulu dengan mengatur parameter-parameternya. Pengaturan tersebut diantaranya:

- Jumlah *node input layer* untuk data latih sebanyak 580 data sedangkan untuk data tes sebanyak 145 data.
- Jumlah *hidden layer* sebanyak 1 layer.
- Jumlah *node* pada *hidden layer* sebanyak 10-40 *node*, parameter ini akan masuk pada skenario uji coba.
- Jumlah *node output layer* sebanyak jumlah kelas yang dihasilkan yaitu 29 *node*.
- Fungsi aktivasi menggunakan aktivasi log sigmoid.
- Algoritma pelatihan yang dipakai adalah *Variable Learning Rate Gradient Descent* menggunakan parameter MATLAB `traingdx`.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Dalam mengimplementasikan aplikasi pengenalan suara untuk Huruf Hijaiyah diperlukan beberapa perangkat pendukung sebagai berikut.

4.1.1 Perangkat Keras

Lingkungan implementasi pada tugas akhir ini adalah sebuah Arduino Uno, perangkat Android dengan Sistem Operasi minimum Jelly Bean, dan laptop HP Pavillion Notebook dengan prosesor Intel Core i7 6500U dengan kecepatan 2,6 GHz dan RAM sebesar 12 GB.

4.1.2 Perangkat Lunak

Spesifikasi dari sisi perangkat lunak pada server menggunakan *software* MATLAB R2016b. Penggunaan MATLAB didukung dengan dua *toolbox* utama yaitu *statistical and machine learning toolbox* dan *signal processing toolbox*. Untuk pembangunan aplikasi Android menggunakan Android Studio dan untuk perograman alat menggunakan Arduino IDE. Selain itu, pada tugas akhir ini dalam melakukan pengolahan angka didukung dengan *software* Microsoft Excel.

4.2 Implementasi Arduino

Arduino merupakan *microcontroller* yang digunakan untuk pemrograman alat yang digunakan untuk menampilkan Huruf

Hijaiyah menggunakan pola braille. Selain itu, Arduino juga berfungsi untuk mengirim data pola braille yang ditampilkan pada alat pada aplikasi Android melalui Bluetooth.

4.2.1 Implementasi Pengaturan Komponen Pada Arduino

Supaya alat dapat berfungsi dengan baik, maka Arduino harus diatur supaya komponen-komponen yang terpasang pada Arduino dapat diprogram sesuai dengan fungsinya. Proses pengaturan tersebut dilakukan dengan Kode Sumber 4.1. Baris 3 pada kode berfungsi sebagai inisiasi pada program Arduino. Tombol untuk berpindah pola braille dideklarasikan pada baris 7-9. Sedangkan baris 10-12 berfungsi sebagai inisiasi untuk *shift register*.

```
1. void setup()
2. {
3.   Serial.begin (9600);
4.
5.   pinMode(reset_pin,OUTPUT);
6.
7.   pinMode(next_pin,INPUT);
8.   pinMode(back_pin,INPUT);
9.
10.  pinMode(DS_pin,OUTPUT);
11.  pinMode(STCP_pin,OUTPUT);
12.  pinMode(SHCP_pin,OUTPUT);
13.
14.  writereg();
15. }
```

Kode Sumber 4.1 Pengaturan Komponen Pada Arduino

4.2.2 Implementasi Fungsi Register

Arduino memiliki jumlah pin untuk input dan output yang terbatas. Untuk mengatasi keterbatasan jumlah pin sebagai media memasang komponen-komponen, maka digunakan register supaya

1 pin dapat menjadi beberapa output. Proses pengaturan register dilakukan dengan Kode Sumber 4.2. Setiap register secara bersamaan akan mengaktifkan pin untuk menampilkan braille dengan pola tertentu sesuai dengan banyaknya jumlah sel braille yang ada. Data output yang seharusnya dikonfigurasi pada pin Arduino secara langsung, digantikan oleh register sehingga pemakaian pin bisa lebih hemat. Proses konfigurasi untuk menampilkan pin braille menggunakan *shift register* diperlihatkan pada baris 6-8.

```

1. void writereg()
2. {
3.     digitalWrite(STCP_pin, LOW);
4.     for (int i = 0; i < COLPIN; i++)
5.     {
6.         digitalWrite(SHCP_pin, LOW);
7.         digitalWrite(DS_pin, registers[i]);
8.         digitalWrite(SHCP_pin, HIGH);
9.     }
10.    digitalWrite(STCP_pin, HIGH);
11. }

```

Kode Sumber 4.2 Fungsi Register

4.2.3 Implementasi Konversi Huruf Hijaiyah

Data Huruf Hijaiyah disimpan dalam bentuk integer. Untuk melakukan konversi dari data integer 1-29 sebagai representasi Huruf Hijaiyah ke data boolean 1 dan 0 sebanyak 6 titik, maka dibutuhkan proses konversi sebagaimana Kode Sumber 4.3. Data 1 menunjukkan bahwa pin braille muncul pada alat, sedangkan data 0 menunjukkan pin braille tidak muncul. Setiap sel braille akan menampilkan data yang sesuai, oleh karena itu pada baris 3-18 menunjukkan kondisi untuk konversi data. Pada baris 19-22 menyimpan data braille yang harus ditampilkan dan pada baris 24 memanggil fungsi register untuk menampilkan data pada *refreshable braille display*.

```

1. void changeProcess() {
2.   for (int i = 0; i < COLUMN; i++) {
3.     if (pattern[x][i] == 1)
4.     {
5.       changePattern(i, 1, 0, 0, 0, 0, 0);
6.     }
7.     else if (pattern[x][i] == 2)
8.     {
9.       changePattern(i, 0, 0, 1, 0, 0, 0);
10.    }
11.    else if (pattern[x][i] == 3)
12.    {
13.      changePattern(i, 1, 1, 0, 0, 0, 0);
14.    }
15.    else
16.    {
17.      changePattern(i, 1, 1, 1, 1, 1, 1);
18.    }
19.    for (int y = 0 + (i * 8); y < 8 + (i * 8); y++)
20.    {
21.      registers[y] = convert[x][y];
22.    }
23.  }
24.  writereg();
25. }

```

Kode Sumber 4.3 Konversi Data Braille Arduino

4.2.4 Implementasi Pengiriman Data Braille

Data yang ditampilkan pada alat, selanjutnya akan dikirim ke aplikasi Android dengan melalui Bluetooth. Proses pengiriman data dilakukan dengan Kode Sumber 4.4. Setiap data baru, ditandai oleh simbol '#' sedangkan untuk akhir dari data ditandai oleh simbol '~'. Tujuan penggunaan simbol ini dikarenakan data pada arduino akan dikirim secara kontinu dalam selang waktu tertentu. Oleh karena itu, dibutuhkan tanda supaya data tidak saling bercampur antara satu dengan yang lainnya. Proses tersebut dapat dilihat pada baris 2-4.

```

1. void sendData(){
2.   Serial.print('#');
3.   Serial.print(x+1);
4.   Serial.print('~');
5.   Serial.println();
6.   delay(100);
7. }

```

Kode Sumber 4.4 Pengiriman Data Braille ke Android

4.2.5 Implementasi *Main* Program

Keseluruhan fungsi pada Arduino akan dijalankan secara kontinu dibagian loop(). Proses tersebut dilakukan dengan void loop()

```

1. {
2.   digitalWrite(reset_pin, HIGH);
3.   next_pin = digitalRead(8);
4.   back_pin = digitalRead(9);
5.
6.   sendData();
7.
8.   changeProcess();
9.
10.  if(next_pin == HIGH){
11.    count++;
12.    if(count > 5){
13.      Serial.print("next");
14.      count = 0;
15.      resetRelay();
16.      x++;
17.      if(x >= ROW-1){
18.        x = ROW-1;
19.      }
20.    }
21.  }
22.
23.  if(back_pin == HIGH){
24.    count++;
25.    if(count > 5){
26.      Serial.print("back");

```

```

27.     count = 0;
28.     resetRelay();
29.     x--;
30.     if(x < 0){
31.         x = 0;
32.     }
33. }
34. }
35. }

```

Kode Sumber 4.5. Pada baris 4-5 menunjukkan bagian pin pada Arduino yang dipakai sebagai tombol, yaitu tombol untuk menuju data berikutnya dan tombol untuk menuju data sebelumnya. Baris 7 akan memanggil fungsi untuk mengirim data dari Arduino ke aplikasi Android. Untuk melakukan konversi data menjadi pola braille dilakukan pada baris 9. Baris 11-21 berjalan jika tombol berpindah ke data selanjutnya ditekan. Sedangkan baris 24-36 berjalan jika tombol untuk berpindah ke data sebelumnya ditekan. Keseluruhan program dalam fungsi loop() ini akan dijalankan secara kontinu pada Arduino saat pertama kali dinyalakan sampai alat dimatikan.

```

36. void loop()
37. {
38.     digitalWrite(reset_pin, HIGH);
39.     next_pin = digitalRead(8);
40.     back_pin = digitalRead(9);
41.
42.     sendData();
43.
44.     changeProcess();
45.
46.     if(next_pin == HIGH){
47.         count++;
48.         if(count > 5){
49.             Serial.print("next");
50.             count = 0;
51.             resetRelay();
52.             x++;
53.             if(x >= ROW-1){

```

```

54.         x = ROW-1;
55.     }
56. }
57. }
58.
59. if(back_pin == HIGH){
60.     count++;
61.     if(count > 5){
62.         Serial.print("back");
63.         count = 0;
64.         resetRelay();
65.         x--;
66.         if(x < 0){
67.             x = 0;
68.         }
69.     }
70. }
71. }

```

Kode Sumber 4.5 *Main Program Arduino*

4.3 Implementasi Android

Implementasi pada bagian Android di aplikasi ini meliputi penerimaan data braille melalui Bluetooth, perekaman suara, pengiriman data suara ke server dan penerimaan hasil klasifikasi suara dari server.

4.3.1 Implementasi Penerimaan Data Braille

Data braille yang dikirimkan melalui Arduino, diterima aplikasi Android melalui Bluetooth. Proses penerimaan data dilakukan dengan Kode Sumber 4.6. Data yang dikirimkan dari alat disimpan dalam variabel `readMessage` pada baris 4. Batas akhir dari data dicek menggunakan kode pada baris 6. Data yang diterima dari alat akan diekstrak dan ditampilkan pada aplikasi Android pada baris 8-9. Variabel untuk menampilkan data dari alat direset kembali pada akhir proses di baris 13.

```

1. bluetoothIn = new Handler() {

```

```

2.     public void handleMessage(android.os.Message ms
g) {
3.         if (msg.what == handlerState) {
4.             String readMessage = (String) msg.obj;
5.             recDataString.append(readMessage);
6.             int endOfLineIndex = recDataString.inde
xOf("~");
7.             if (endOfLineIndex > 0) {
8.                 String dataInPrint = recDataString.
substring(0, endOfLineIndex);
9.                 txtString.setText("Data Received =
" + dataInPrint);
10.                int dataLength = dataInPrint.length
();
11.                txtStringLength.setText("String Len
gth = " + String.valueOf(dataLength));
12.                recDataString.delete(0, recDataStri
ng.length());
13.                dataInPrint = " ";
14.            }
15.        }
16.    }
17. };

```

Kode Sumber 4.6 Penerimaan Data Braille ke Android

4.3.2 Implementasi Perekaman Suara

Pengguna melakukan perekaman suara dengan Aplikasi Android menggunakan sampling rate sebesar 44100. Proses perekaman suara dilakukan dengan Kode Sumber 4.7. Perekaman suara menggunakan kelas AudioRecord dari Android seperti pada baris 2. Parameter-parameter perekaman suara dideklarasikan secara global dan dipanggil pada baris 3 kode.

```

1.     private void startRecording(){
2.         recorder = new AudioRecord(MediaRecorder.AudioS
ource.MIC,

```



```

3.         RECORDER_SAMPLERATE, RECORDER_CHANNELS,
   RECORDER_AUDIO_ENCODING, bufferSize);
4.
5.     int i = recorder.getState();
6.     if(i==1)
7.         recorder.startRecording();
8.
9.     isRecording = true;
10.
11.    recordingThread = new Thread(new Runnable() {
12.
13.        @Override
14.        public void run() {
15.            writeAudioDataToFile();
16.        }
17.    }, "AudioRecorder Thread");
18.
19.    recordingThread.start();
20. }

```

Kode Sumber 4.7 Perekaman Suara

4.3.3 Implementasi Pembuatan File .wav

File suara dibuat dengan format .wav dengan *sampling rate* sebesar 44100 menggunakan channel mono. Proses tersebut dilakukan dengan Kode Sumber 4.8. Data perekaman suara dalam bentuk *byte* akan ditulis menjadi sebuah file .wav. Untuk melakukannya digunakan *exception try-catch* pada baris 12-29. Pada baris 20-21 dibuat *header* dari *file* .wav menggunakan fungsi `WriteWaveFileHeader`.

```

1. private void copyWaveFile(String inFilename, String
   outFilename){
2.     FileInputStream in = null;
3.     FileOutputStream out = null;
4.     long totalAudioLen = 0;
5.     long totalDataLen = totalAudioLen + 36;
6.     long longSampleRate = RECORDER_SAMPLERATE;
7.     int channels = 1;

```

```

8.     long byteRate = RECORDER_BPP * RECORDER_SAMPLER
    ATE * channels/8;
9.
10.    byte[] data = new byte[bufferSize];
11.
12.    try {
13.        in = new FileInputStream(inFilename);
14.        out = new FileOutputStream(outFilename);
15.        totalAudioLen = in.getChannel().size();
16.        totalDataLen = totalAudioLen + 36;
17.
18.        AppLog.logString("File size: " + totalDataLen);
19.
20.        WriteWaveFileHeader(out, totalAudioLen, totalDataLen,
21.                               longSampleRate, channels, byteRate)
22.        ;
23.        while(in.read(data) != -1){
24.            out.write(data);
25.        }
26.
27.        in.close();
28.        out.close();
29.    } catch (FileNotFoundException e) {
30.        e.printStackTrace();
31.    } catch (IOException e) {
32.        e.printStackTrace();
33.    }
34. }

```

Kode Sumber 4.8 Pembuatan File .wav

4.3.4 Implementasi Pengiriman Suara

Setelah direkam, suara dikirimkan ke *server* untuk diproses ekstraksi fitur dan dilakukan klasifikasi. Proses tersebut dilakukan dengan Kode Sumber 4.9. Untuk melakukan mengunggah *file* ini digunakan *library* retrofit.

```

1. private void uploadFile() {

```

```

2.     progressDialog.show();
3.
4.     // Map is used to multipart the file using okhttp
tp3.RequestBody
5.     File file = new File(mediaPath);
6.
7.     // Parsing any Media type file
8.     RequestBody requestBody = RequestBody.create(Media
diaType.parse("*/*"), file);
9.     MultipartBody.Part fileToUpload = MultipartBody
.Part.createFormData("file", file.getName(), request
tBody);
10.    RequestBody filename = RequestBody.create(Media
Type.parse("text/plain"), file.getName());
11.
12.    ApiConfig getResponse = AppConfig.getRetrofit()
.create(ApiConfig.class);
13.    Call<ServerResponse> call = getResponse.uploadF
ile(fileToUpload, filename);
14.    call.enqueue(new Callback<ServerResponse>() {
15.        @Override
16.        public void onResponse(Call<ServerResponse>
call, Response<ServerResponse> response) {
17.            ServerResponse serverResponse = respons
e.body();
18.            if (serverResponse != null) {
19.                if (serverResponse.getSuccess()) {
20.                    Toast.makeText(getApplicationContext()
ntext(), serverResponse.getMessage(), Toast.LENGTH_
SHORT).show();
21.                } else {
22.                    Toast.makeText(getApplicationContext()
ntext(), serverResponse.getMessage(), Toast.LENGTH_
SHORT).show();
23.                }
24.            } else {
25.                assert serverResponse != null;
26.                Log.v("Response", serverResponse.to
String());
27.            }
28.            progressDialog.dismiss();
29.        }

```

```

30.
31.     @Override
32.     public void onFailure(Call<ServerResponse>
    call, Throwable t) {
33.
34.     }
35. });
36. }

```

Kode Sumber 4.9 Pengiriman suara

4.3.5 Implementasi Penerimaan Hasil Klasifikasi

Setelah data diproses dari server, data dikirimkan dan diterima aplikasi Android. Proses tersebut dilakukan dengan Kode Sumber 4.10. Data hasil dari proses klasifikasi pada MALTAB diterima oleh aplikasi Android menggunakan kelas `ServerResponse`. Data diterima dari *server* dalam bentuk string. Data string hasil klasifikasi diterima oleh aplikasi Android dari *server* menggunakan kode pada baris 20.

```

1. class ServerResponse {
2.
3.     // variable name should be same as in the json
    response from php
4.     @SerializedName("success")
5.     boolean success;
6.     @SerializedName("message")
7.     String message;
8.     @SerializedName("hasil")
9.     String hasil;
10.
11.     String getMessage() {
12.         return message;
13.     }
14.
15.     boolean getSuccess() {
16.         return success;
17.     }
18.
19.     String getHasil() {

```

```

20.         return hasil;
21.     }
22. }

```

Kode Sumber 4.10 Penerimaan Hasil Klasifikasi Suara ke Aplikasi Android

4.4 Implementasi MATLAB

4.4.1 *Preprocessing* dengan VAD

Sebelum diekstraksi, sinyal dipotong untuk diambil bagian informasinya saja. Proses tersebut dilakukan dengan Kode Sumber 4.11. Proses ini menggunakan *library* bawaan MATLAB yaitu vadG729. Inisialisasi parameter dari VAD dilakukan pada baris 8. Pemanggilan algoritma VAD dilakukan pada baris 22. Untuk mengambil bagian sinyal yang bersuara saja dilakukan menggunakan kode pada baris 26-32. Hasil dari algoritma VAD ini ditampung pada variabel res pada baris 37.

```

1. function[res] = vad(path)
2.
3. audioSource = dsp.AudioFileReader('SamplesPerFrame'
4.     ,80,...
5.     'Filename',path,...
6.     'OutputDataType', 's
7.     ingle');
8. % inisialisasi parameter vad
9. VAD_cst_param = vadInitCstParams;
10. clear vadG729
11. % Run for 10 seconds
12. numTSteps = 1000;
13.
14. i=1;
15. % deklarasi variabel sementara penyimpan sinyal aud
16. io yang bersuara saja
17. val=single(zeros(80000,1));

```

```

18. while(numTSteps)
19. % Retrieve 10 ms of speech data from the audio re
   coder
20. speech = audioSource();
21. % Call the VAD algorithm
22. decision = vadG729(speech, VAD_cst_param);
23. numTSteps = numTSteps - 1;
24.
25. % mengambil bagian yang bersuara saja
26. s = size(speech);
27. if(decision == 1)
28.     for c = 1:s
29.         val(i,1) = speech(c,1);
30.         i = i+1;
31.     end
32. end
33. %scope(decision, speech);
34. end
35.
36. % variabel penampung hasil VAD
37. res=single(zeros(i,1));
38.
39. for k = 1:i
40.     res(k,1) = val(k,1);
41. end
42.
43. end

```

Kode Sumber 4.11 Implementasi VAD

4.4.2 Ekstraksi Fitur MFCC

4.4.2.1 Preemphasis Filter

Preemphasis filter bertujuan untuk mengurangi noise pada sinyal. Proses tersebut dilakukan dengan Kode Sumber 4.12.

```

1. preemphasis = filter( [1 -alpha], 1, cutsig );

```

Kode Sumber 4.12 Proses *Preemphasis Filter*

4.4.2.2 Framing

Framing berfungsi untuk membagi sampel suara menjadi beberapa potongan *frame*. Proses tersebut dilakukan dengan Kode Sumber 4.13. Dalam suatu sinyal, seringkali panjang *frame* tidak pas membagi keseluruhan sinyal sehingga butuh satu *frame* untuk menampung sinyal terakhir yang dilakukan dengan menggunakan kode pada baris 7-8. Selanjutnya *frame* yang terisi *padding* tersebut digabungkan dengan *frame* lainnya menggunakan kode baris 13-15. Keseluruhan data *frame* sinyal kemudian diproses menggunakan *hamming window* pada baris 17-18.

```

1. function[frames] = framing(N, M, preemphasis, hammw
   in)
2.
3. preemphasis = preemphasis(:);
4.
5. nbFrames = floor((length(preemphasis)-N)/M+1);
6.
7. batas = (length(preemphasis)-((nbFrames-1)*M+N));
8. padding = N-batas;
9.
10. preemphasis = [ preemphasis; zeros(padding,1) ];
11. nbFrames = nbFrames+1;
12.
13. indf = M*[ 0:(nbFrames-1) ];
14. inds = [ 1:N ].';
15. indexes = indf(ones(N,1),:) + inds(:,ones(1,nbFrame
   s));
16.
17. frames = preemphasis( indexes );
18. frames = diag( hammw ) * frames;
19.
20. end

```

Kode Sumber 4.13 Proses Framing

4.4.2.3 Windowing

Windowing berfungsi untuk mengurangi efek diskontinuitas pada ujung sinyal hasil proses sebelumnya. Proses

tersebut dilakukan dengan menggunakan *hamming window* pada Kode Sumber 4.14.

```

1. hammwin = zeros(1,nt);
2. for k = 1:nt
3.     hammwin(1,k) = 0.54 - 0.46 * cos(2 * pi * (k-
4.     1) / (nt-1));
5. end
5. Frames = framing(nt, nto, preemphasis, hammwin);

```

Kode Sumber 4.14 Implementasi Windowing

4.4.2.4 *Magnitude Spectrum*

Data diubah dari domain waktu ke domain frekuensi kemudian dihitung energinya menggunakan kode pada baris 2 dengan banyaknya FFT yang dihitung menggunakan kode pada baris 1. Proses tersebut dilakukan dengan Kode Sumber 4.15.

```

1. nfft = 2^nextpow2(nt); % banyak fft
2. mag = abs( fft(Frames,nfft,1) );

```

Kode Sumber 4.15 Implementasi Magnitude Spectrum

4.4.2.5 *Filterbank*

Filterbank bertujuan untuk mengetahui ukuran energi pada *frequency band* tertentu dalam sinyal suara. Proses tersebut dilakukan dengan Kode Sumber 4.16.

```

1. function [ filterbank, f, c ] = trifbank( numchanne
2.     1, nffts, lowfrek, highfrek, fs )
3.     fmin = 0;
4.     fmax = 0.5*fs;
5.     f = linspace( fmin, fmax, nffts );
6.     fw = hz2mel( f );
7.     c = mel2hz( hz2mel(lowfrek)+[0:numchannel+1]*((
8.         hz2mel(highfrek)-
9.         hz2mel(lowfrek))/(numchannel+1)) );

```



```

8.     cw = hz2mel( c );
9.
10.    filterbank = zeros( numchannel, nffts );
11.    for m = 1:numchannel
12.
13.        k = f>=c(m)&f<=c(m+1);
14.        filterbank(m,k) = (f(k)-c(m))/(c(m+1)-
    c(m));
15.        k = f>=c(m+1)&f<=c(m+2);
16.        filterbank(m,k) = (c(m+2)-f(k))/(c(m+2)-
    c(m+1));
17.
18.    end
19. end

```

Kode Sumber 4.16 Implementasi *FilterBank*

4.4.2.6 Konversi ke Satuan Herzt

Pada proses *filterbank* dilakukan konversi dari satuan mel menjadi satuan herzt. Proses tersebut dilakukan dengan Kode Sumber 4.19

```

1. function[res] = mel2hz(mel)
2.     res = 700*exp(mel/1127)-700;
3. end

```

Kode Sumber 4.17 Implementasi Konversi ke Satuan Herzt

4.4.2.7 Konversi ke Satuan Mel

Pada proses *filterbank* dilakukan konversi dari satuan mel menjadi satuan herzt. Proses tersebut dilakukan dengan Kode Sumber 4.18.

```

1. function[res] = hz2mel(hz)
2.     res = 1127*log(1+hz/700);
3. end

```

Kode Sumber 4.18 Implementasi Konversi ke Satuan Mel

4.4.2.8 Discrete Cosine Transform (DCT)

DCT merupakan langkah terakhir dari MFCC. Proses tersebut dilakukan dengan Kode Sumber 4.19.

```

1. function[res] = dctm (numcepstral, numchannel)
2.     res = ( sqrt(2.0/numchannel) * cos( repmat([0:n
umcepstral-1].',1,numchannel) ...
3.         .* repmat(pi*([1:numchannel]-
0.5)/numchannel,numcepstral,1) ) );
4. end

```

Kode Sumber 4.19 Implementasi DCT

4.4.3 Klasifikasi Neural Network

Klasifikasi dilakukan menggunakan *toolbox Neural Network* pada MATLAB. Proses tersebut dilakukan dengan Kode Sumber 4.20. Pemanggilan data latih dan data tes beserta targetnya dilakukan dengan menggunakan kode pada baris 3-6. Sebelumnya dilakukan proses klasifikasi, maka sebelumnya dilakukan inisiasi bobot secara random menggunakan kode pada baris 11. Proses pembangunan jaringan dilakukan pada baris 13 dan proses pelatihan dilakukan pada baris 15. Hasil dari proses klasifikasi dapat dilihat menggunakan kode pada baris 17-18.

```

1. function [c,cm] = nnclassify()
2.
3. x = csvread('traindata.csv');
4. t = csvread('trainclass.csv');
5. testX = csvread('testdata.csv');
6. testT = csvread('testclass.csv');
7.
8. size(x)
9. size(t)
10.
11. setdemorandstream(391418381)
12.

```

```
13. net = patternnet(10, 'traindx');  
14.  
15. [net,tr] = train(net,x,t);  
16.  
17. testY = net(testX);  
18. testIndices = vec2ind(testY);  
19.  
20. [c,cm] = confusion(testT,testY);  
21.  
22. end
```

Kode Sumber 4.20 Implementasi Neural Network

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Lingkungan implementasi pada tugas akhir ini adalah sebuah Arduino Uno, perangkat Android dengan Sistem Operasi minimum Jelly Bean, dan laptop HP Pavillion Notebook dengan prosesor Intel Core i7 6500U dengan kecepatan 2,6 GHz dengan RAM sebesar 12 GB.

5.2 Data Uji Coba

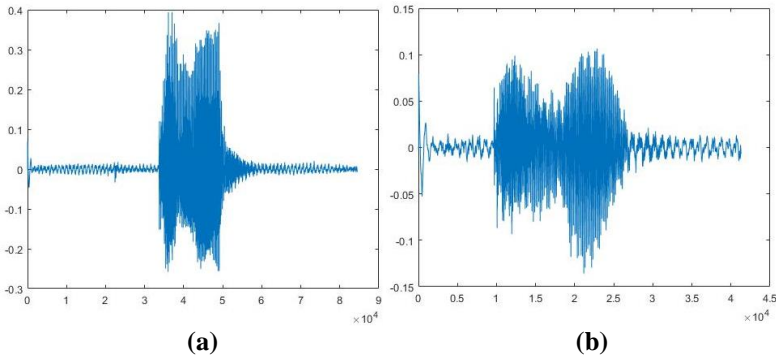
Data uji coba yang digunakan pada Tugas Akhir ini adalah rekaman pengucapan Huruf Hijaiyah melalui aplikasi Android dan dikonversi menjadi *file* audio dengan format “.wav”, *channel* audio *mono*, dan *sample rate* 44100 Hz.

5.3 Hasil Uji Coba

Hasil uji coba untuk Tugas Akhir ini dijelaskan hasil yang didapatkan dari setiap proses yang dilakukan. Perbandingan hasil diambil dari perwakilan dari Huruf Hijaiyah yaitu alif pada satu data laki-laki dan satu data perempuan. Secara keseluruhan, proses yang dilakukan pada Tugas Akhir ini adalah *preprocessing* dengan menggunakan *Voice Activity Detection* (VAD), kemudian dilanjutkan oleh proses ekstraksi fitur MFCC yang terdiri dari proses *preemphasis*, *frame blocking*, spektrum FFT, *filterbank*, *Discrete Cosine Transform* (DCT), dan *cepstral lifter*. Pada proses MFCC tersebut akan dihasilkan fitur untuk menjadi input pada proses klasifikasi menggunakan *Neural Network*.

Untuk tahap awal, data suara didapatkan dengan perekaman terhadap suara laki-laki dan perempuan. Hasil perekaman sinyal suara untuk huruf alif dapat dilihat pada Gambar 5.1 dan untuk

huruf hamzah sampai dengan ya dapat dilihat pada sub bab lampiran.

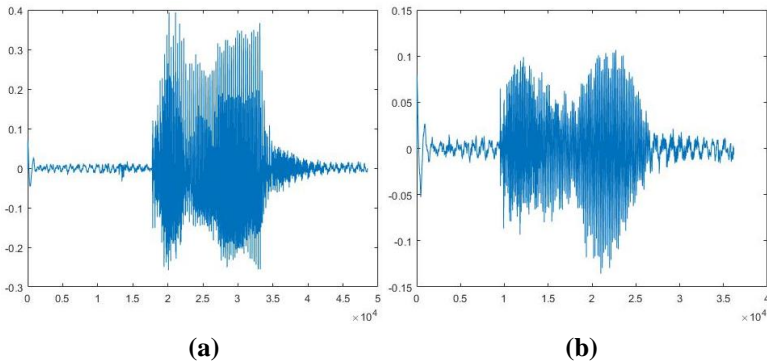


Gambar 5.1 Sinyal Suara Huruf Alif: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

Perekaman suara untuk huruf alif pada suara laki-laki subjek 1, diperoleh data sepanjang 84480 data, sedangkan pada data perempuan subjek 1 didapatkan data sepanjang 41280. Panjang data bervariasi antar subjek dan hurufnya, bergantung pada lamanya durasi perekaman, semakin lama durasi perekaman maka data yang diambil akan semakin panjang.

Proses pertama adalah *preprocessing* dengan menggunakan VAD. Pada tahap ini, panjang data suara akan berkurang dikarenakan data sinyal suara akan dibedakan menjadi sinyal suara informasi dan sinyal suara diam. Bagian diam akan dibuang pada tahap ini. Data suara laki-laki yang awalnya sebanyak 84480 data berkurang menjadi 48481 atau berkurang lebih dari 42% dari data semula. Sedangkan untuk data perempuan, awalnya panjang keseluruhan data adalah 41280 dan pada proses ini menjadi sebanyak 36241 atau berkurang lebih dari 36% dari panjang data semula. Proses *preprocessing* sangat penting dalam Tugas Akhir ini. Data awal hasil perekaman suara jumlahnya sangat banyak, padahal tidak semua bagian data tersebut penting. Jika data yang diproses terlalu panjang, maka menyebabkan waktu untuk melakukan ekstraksi fitur menjadi lama dan juga

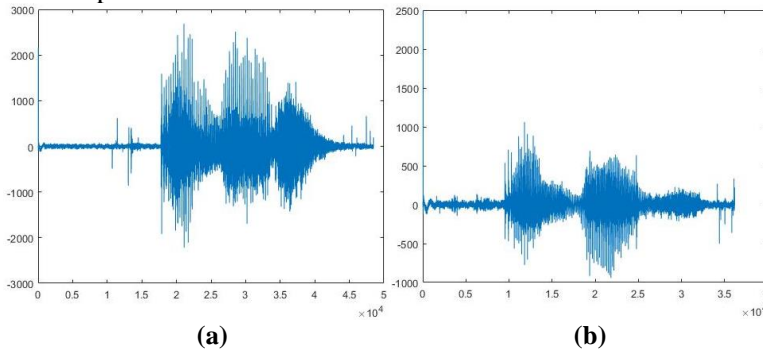
menyebabkan fitur yang dihasilkan menjadi tidak maksimal. Oleh karena itu, pada tahap *preprocessing* ini data diambil bagian informasinya saja sehingga panjang data menjadi berkurang dan proses ekstraksi fitur dapat berlangsung lebih cepat dan fitur yang dihasilkan menjadi lebih optimal. Hasil dari *preprocessing* VAD untuk huruf alif dapat dilihat pada Gambar 5.2 dan untuk huruf hamzah sampai dengan ya dapat dilihat pada sub bab lampiran.



Gambar 5.2 Hasil Pemotongan Sinyal Huruf Alif dengan VAD: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

Setelah *preprocessing*, maka proses ekstraksi fitur MFCC akan dimulai. Guna mengurangi *noise* pada sinyal, pada tahap ekstraksi fitur MFCC ini, akan dilakukan proses *preemphasis* filter yang merupakan tahap pertama MFCC. Filter ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara. Data sinyal suara laki-laki setelah proses VAD memiliki rata-rata nilai amplitudo sebesar -0.00020300128 , dengan adanya *preemphasis* filter ini, rata-rata amplitudo untuk data suara pada laki-laki menjadi -0.3325924 . Sedangkan pada data sinyal suara perempuan setelah proses *preprocessing* amplitudo memiliki rata-rata -0.00021599179 dan setelah proses *preemphasis* filter amplitudo memiliki rata-rata sebesar -0.3538812 . Terdapat peningkatan rata-rata amplitudo setelah proses *preemphasis* filter ini. Hal ini dikarenakan filter

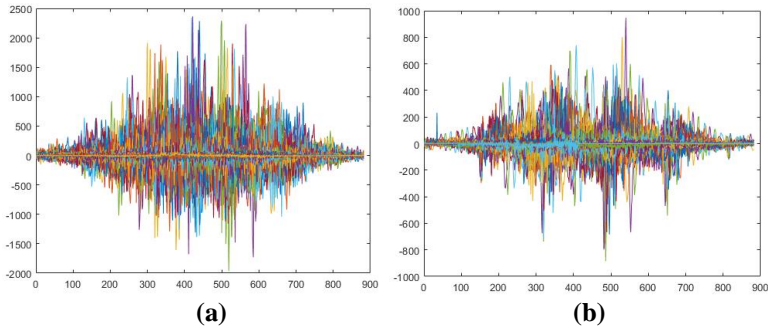
yang digunakan akan meningkatkan nilai amplitudo yang ada. Hasil dari proses untuk huruf alif dapat dilihat pada Gambar 5.3 dan untuk huruf hamzah sampai dengan ya dapat dilihat pada sub bab lampiran.



Gambar 5.3 Hasil *Preemphasis* Filter Pada Sinyal Suara Alif: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

Setiap data sinyal dari proses *filtering* akan dipecah menjadi beberapa bagian sinyal yang disebut dengan *frame* melalui proses *frame blocking*. Pada Tugas Akhir ini, hasil terbaik didapatkan saat sinyal dipecah untuk setiap 20 ms. Setiap *frame* tersebut bertumpuk (*overlap*) terhadap sinyal lainnya sebanyak 25% (5 ms). Selain itu, pada proses ini juga dilakukan proses *windowing* menggunakan *window* Hamming pada setiap *frame* yang dihasilkan. Pada data laki-laki, diperoleh hasil matrix berukuran 882 x 73. Dimana, 882 menyatakan banyaknya data untuk setiap *frame* dan 73 menyatakan banyaknya *frame* yang dihasilkan pada proses ini. Sedangkan pada data perempuan diperoleh hasil matrix sebesar 882 x 55. Dimana, 882 menyatakan menyatakan banyaknya data untuk setiap *frame* dan 73 menyatakan banyaknya *frame* yang dihasilkan. Pada data yang lain juga akan diperoleh jumlah data yang sama untuk setiap ukuran *frame*-nya. Hal ini dikarenakan untuk jumlah data pada setiap *frame* yang dihasilkan tersebut ditentukan oleh parameter *frame duration* pada proses ini. Jumlah data pada setiap *frame* merupakan hasil kali antara parameter *frame*

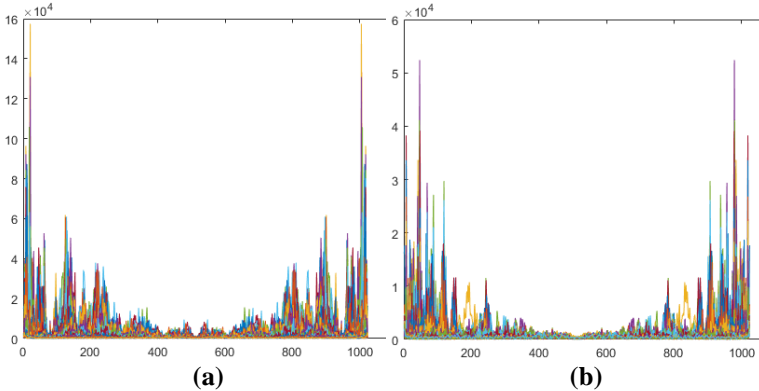
duration dan besarnya *sampling rate*, yaitu sebesar 44100. Seperti yang telah dijelaskan sebelumnya, durasi pemotongan *frame* (parameter *frame duration*) yang dilakukan pada Tugas Akhir ini adalah sebesar 20 ms. Oleh karena itu, banyaknya data pada setiap *frame* yang dihasilkan adalah 882. Untuk banyaknya *frame* yang dihasilkan akan berbeda untuk setiap data yang diproses karena bergantung pada panjang data yang dihasilkan setelah *preprocessing*. Hasil dari proses *frame blocking* untuk huruf alif dapat dilihat pada Gambar 5.4 dan untuk huruf hamzah sampai dengan ya dapat dilihat pada sub bab lampiran.



Gambar 5.4 Hasil *Frame Blocking* Pada Sinyal Suara Alif: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

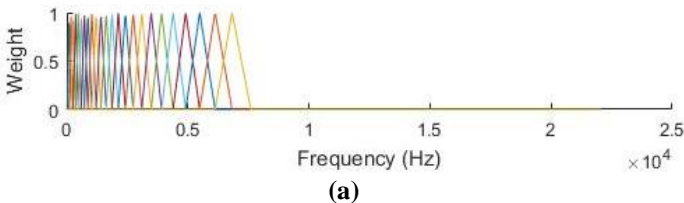
Pada proses berikutnya, *frame* sinyal akan diubah ke domain frekuensi dengan menggunakan FFT untuk kemudian dihitung *magnitude* spektrumnya. Matrix yang dihasilkan dari proses ini berukuran 1024×73 untuk data laki-laki dan berukuran perempuan 1024×55 untuk data perempuan. Dimana, 1024 merupakan nilai yang ditentukan melalui pangkat 2 terdekat dari banyaknya data pada setiap *frame*. Pangkat 2 terdekat tersebut diambil dari nilai paling dekat yang berada di atasnya, dimana nilai pangkat 2 yang berada diantara 882 adalah 512 dan 1024. Oleh karena itu, pada proses ini diambil nilai 1024 sebagai nilai pangkat 2 terdekat dari 882. Sedangkan untuk nilai 73 dan 55 diambil dari banyaknya *frame* dari setiap data suara sesuai dengan hasil yang diperoleh dari

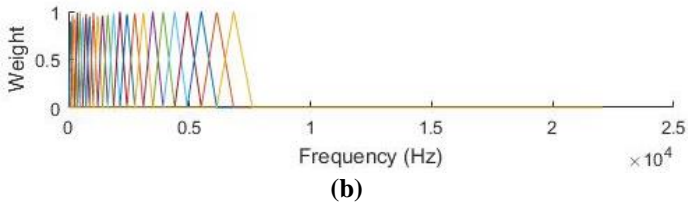
proses *frame blocking*. Dimana dari proses tersebut didapatkan banyaknya *frame* untuk data suara laki-laki sebanyak 75 dan banyaknya *frame* untuk data suara perempuan adalah sebanyak 55. Hasil dari *magnitude* spektrum untuk huruf alif dapat dilihat pada Gambar 5.5 dan untuk huruf hamzah sampai dengan ya dapat dilihat pada sub bab lampiran.



Gambar 5.5 Hasil Magnitude Spectrum Pada Sinyal Suara Alif: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

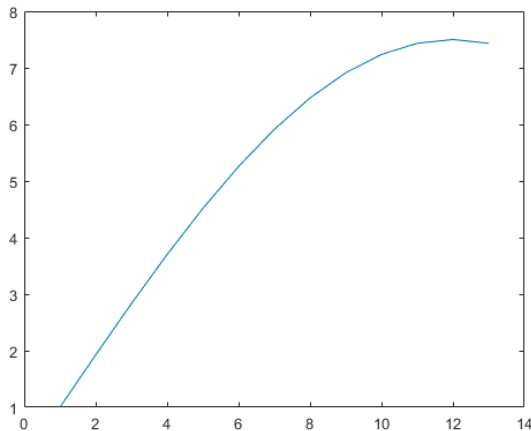
Pada setiap spektrum sinyal akan dicari filterbank yang merupakan representasi dari energi pada *frame* sinyal. Semakin tinggi frekuensi, maka *filterbank* akan semakin lebar. Pada proses ini ditentukan banyaknya *channel* yang ingin dihasilkan yaitu 24. Hasil dari proses *filterbank* adalah matrix berukuran 24 x 513 dapat dilihat pada Gambar 5.6.





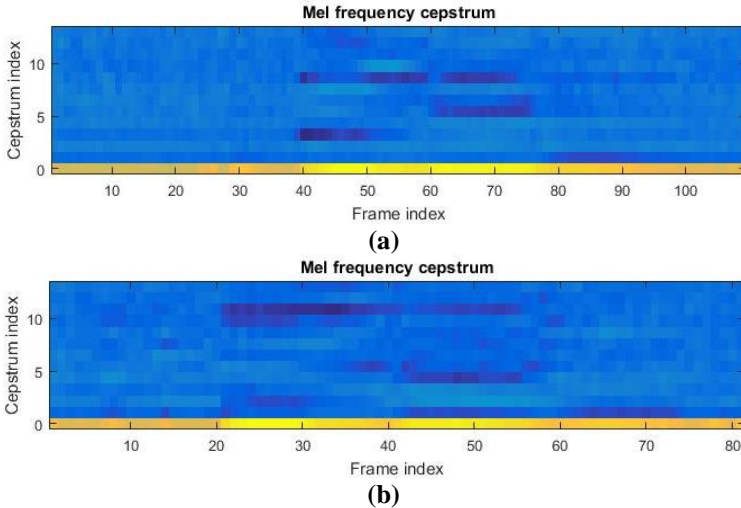
Gambar 5.6 *Filterbank* Sinyal Huruf Alif: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

Proses terakhir dari MFCC adalah DCT dan *cepstral lifter*. Pada proses DCT akan dihitung matrix DCT berukuran 13×24 . Setelah didapatkan matrix DCT maka akan dikalikan dengan log dari nilai *magnitude* spektrum yang telah melalui *filterbank*. Hasil proses tersebut disebut dengan koefisien cepstral. Banyaknya koefisien cepstral adalah 13 nilai untuk setiap *frame*. Sehingga untuk data laki-laki didapatkan 13×73 nilai koefisien cepstral sedangkan untuk data perempuan didapatkan 13×55 nilai koefisien cepstral yang merupakan fitur dari proses MFCC. Fitur tersebut harus melalui *window* pada proses cepstral lifter untuk memperhalus fitur yang didapat. *Window* tersebut memiliki ukuran 1×13 yang dapat dilihat pada Gambar 5.7.



Gambar 5.7 Window Pada Proses Cepstral Lifter

Setelah melalui proses cepstral lifter baru didapatkan nilai dari fitur MFCC sebanyak 13 buah fitur untuk setiap *frame*. Hasil dari proses MFCC untuk huruf alif dapat dilihat pada Gambar 5.8 dan untuk huruf hamzah sampai dengan ya dapat dilihat pada sub bab lampiran.



Gambar 5.8 MFCC Sinyal Huruf Alif: (a) Sinyal Suara Data Laki-Laki, (b) Sinyal Suara Data Perempuan

Adanya perbedaan jumlah *frame* setiap sinyal membuat ukuran matriks fitur tidak seragam serta memiliki ukuran yang sangat besar. Untuk itu, pada Tugas Akhir ini fitur dihitung berdasarkan rata-rata dan standar deviasi dari keseluruhan *frame* setiap fitur. Dengan demikian, maka didapatkan 26 fitur sinyal untuk setiap data suara.

Fitur akan diproses pada tahap klasifikasi menggunakan *Neural Network*. Fungsi *training* yang digunakan adalah *traindx*. Dari proses klasifikasi ini akan didapatkan akurasi dari pengenalan ucapan yang dilakukan.

5.4 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai-nilai parameter yang tepat untuk digunakan pada masing-masing proses. Nilai parameter yang tepat penting untuk diketahui karena penggunaan parameter yang tepat akan memberikan hasil yang terbaik pada keluaran tiap proses.

Skenario pengujian terdiri dari lima macam yaitu:

1. Uji coba penentuan parameter *frame duration*.
2. Uji coba penentuan ukuran *overlap*.
3. Uji coba penentuan jumlah neuron.
4. Uji coba penentuan *learning rate*.
5. Uji coba berdasarkan data *gender*.

5.4.1 Uji Coba Parameter *Frame Duration*

Hasil uji coba parameter *frame duration* pada proses ekstraksi fitur dapat dilihat pada Tabel 5.1. *Frame duration* merupakan parameter yang digunakan untuk menentukan panjang dari potongan sinyal pada proses *frame blocking*. Ukuran untuk potongan durasi *frame* adalah antara rentang 20 ms sampai dengan 40 ms.

Tabel 5.1 Hasil Uji Coba Parameter *Frame Duration*

| <i>Frame Duration</i> | Data Perempuan | Data Laki-Laki | Rata-rata |
|-----------------------|----------------|----------------|---------------|
| 20 | 80.69% | 89.66% | 85.17% |
| 25 | 81.38% | 88.28% | 84.83% |
| 30 | 80.00% | 86.90% | 83.45% |
| 35 | 81.38% | 84.14% | 82.76% |
| 40 | 81.38% | 86.21% | 83.79% |

Berdasarkan hasil uji coba, diperoleh hasil yang berbeda pada jenis data laki-laki dan jenis data perempuan. Pada data perempuan, hasil tertinggi diperoleh pada parameter 25 ms, 35 ms,

dan 40 ms dengan akurasi 81.38%. Namun, perbedaan hasil akurasi pada data perempuan tidak terlalu berbeda untuk setiap variasi parameternya, dimana untuk durasi 20 ms diperoleh hasil 80.69% dan pada durasi 30 ms diperoleh hasil 80.00%. Pada data laki-laki, hasil tertinggi diperoleh pada akurasi sebesar 89.66% dengan durasi *frame* sebesar 20 ms. Berbeda dengan data perempuan, pada data laki-laki akurasi memiliki kecenderungan menurun dengan semakin besarnya durasi *frame*. Rata-rata nilai akurasi tertinggi diperoleh pada data durasi *frame* 20 ms, yaitu sebesar 85.17%.

Data yang digunakan pada proses ekstraksi fitur data pengucapan suara huruf hijaiyah yang telah melalui proses *preprocessing* untuk dihilangkan bagian diamnya. Hal ini menyebabkan panjang data suara rata-rata hanya berkisar antara 1 detik saja. Karena pendeknya data sinyal suara, pemotongan sinyal dengan durasi yang panjang memberikan hasil yang kurang optimal. Sedangkan dengan durasi 20 ms, sinyal suara setiap dapat dianalisa cirinya secara detail di tahap MFCC.

5.4.2 Uji Coba Ukuran *Overlap*

Hasil uji coba ukuran *overlap* pada proses ekstraksi fitur dapat dilihat pada Tabel 5.2. Pada uji coba parameter *frame duration* diperoleh hasil optimal pada durasi *frame* 20 ms. Parameter *frame duration* kemudian ditetapkan untuk mengetahui ukuran *overlap* yang optimal untuk durasi 20 ms tersebut. Ukuran *overlap* sendiri merupakan ukuran tumpang tindih sinyal pada proses *frame blocking*. Proses ini dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame*.

Tabel 5.2 Hasil Uji Coba Parameter Ukuran *Overlap*

| <i>Overlap</i> | Data Perempuan | Data Laki-Laki | Rata-Rata |
|----------------|----------------|----------------|---------------|
| 5 | 81.38% | 90.34% | 85.86% |
| 10 | 80.69% | 89.66% | 85.17% |
| 15 | 79.31% | 84.83% | 82.07% |

Berdasarkan hasil uji coba, nilai akurasi baik dari data laki-laki maupun data perempuan menurun seiring dengan bertambahnya panjang *overlap*. Hasil terbaik pada kedua jenis data berada pada waktu overlap selama 5 ms dimana pada data perempuan didapatkan akurasi sebesar 81.38% dan pada data laki-laki didapatkan akurasi sebesar 90.34%. Rata-rata akurasi pada *overlap* 5 ms adalah 85.86%.

5.4.3 Uji Coba Jumlah Neuron

Hasil uji coba jumlah neuron pada proses klasifikasi dapat dilihat pada Tabel 5.3. Terdapat perbedaan antara hasil dari data laki-laki dan perempuan. Pada data perempuan, hasil akurasi terbaik didapatkan saat jumlah neuron sebanyak 25 dengan persentase 94.48%. Pada data laki-laki, akurasi terbaik didapatkan saat jumlah neuron sebanyak 30 dengan persentase 97.93%. Hasil rata-rata dari data laki-laki dan perempuan tertinggi berada pada jumlah neuron sebanyak 30 dengan persentase sebesar 95.52%.

Tabel 5.3 Hasil Uji Coba Parameter Jumlah Neuron

| Jumlah Neuron | Data Perempuan | Data Laki-Laki | Rata-Rata |
|---------------|----------------|----------------|---------------|
| 10 | 81.38% | 90.34% | 85.86% |
| 15 | 88.28% | 93.10% | 90.69% |
| 20 | 91.72% | 95.86% | 93.79% |
| 25 | 94.48% | 93.10% | 93.79% |
| 30 | 93.10% | 97.93% | 95.52% |
| 35 | 89.66% | 95.86% | 92.76% |
| 40 | 92.41% | 97.24% | 94.83% |

5.4.4 Uji Coba Learning Rate

Hasil uji coba nilai *learning rate* pada proses klasifikasi dapat dilihat pada Tabel 5.4. Dari hasil uji coba menunjukkan

bahwa parameter *learning rate* sama sekali tidak berpengaruh terhadap akurasi data. Nilai *learning rate* sama sekali tidak memiliki dampak pada akurasi data perempuan dan data laki, dimana pada data perempuan akurasi tetap pada 93.10% dan pada data laki-laki akurasi tetap pada 97.93%.

Pada Tugas Akhir ini, Neural Network dibangun dengan menggunakan fungsi pembelajaran *gradient descent with momentum and adaptive learning* (traingdx) yang merupakan penggabungan antara *algoritma gradient descent with adaptive learning* (traingda) dan *algoritma gradient descent with momentum* (traingdm). Oleh karena itu, perubahan parameter *learning rate* sama sekali tidak berpengaruh terhadap hasil akurasi dikarenakan fungsi pembelajaran akan melakukan adaptasi terhadap *learning rate*. Apabila *learning rate* terlalu kecil untuk menuju konvergensi, maka *learning rate* akan dinaikkan dan apabila *learning rate* terlalu tinggi, maka *learning rate* akan diturunkan. Dengan demikian, maka algoritma pembelajaran akan tetap terjaga pada kondisi stabil.

Tabel 5.4 Hasil Uji Coba Parameter *Learning Rate*

| Jumlah Neuron | Data Perempuan | Data Laki-Laki | Rata-Rata |
|---------------|----------------|----------------|-----------|
| 0.001 | 93.10% | 97.93% | 95.52% |
| 0.005 | 93.10% | 97.93% | 95.52% |
| 0.01 | 93.10% | 97.93% | 95.52% |
| 0.025 | 93.10% | 97.93% | 95.52% |
| 0.05 | 93.10% | 97.93% | 95.52% |
| 0.075 | 93.10% | 97.93% | 95.52% |
| 0.1 | 93.10% | 97.93% | 95.52% |
| 0.5 | 93.10% | 97.93% | 95.52% |

5.4.5 Uji Coba Berdasarkan Gender

Hasil uji coba berdasarkan perbedaan jenis kelamin pada proses klasifikasi dapat dilihat pada Tabel 5.5. Data suara laki-laki dan data suara perempuan memiliki hasil akhir akurasi yang berbeda. Hal ini dikarenakan adanya perbedaan warna suara untuk laki-laki dan perempuan. Data suara laki-laki dan perempuan jika digabungkan memiliki akurasi yang menurun dibandingkan jika dilakukan perhitungan masing-masing. Pada data laki-laki akurasi mencapai nilai terbesar yaitu mencapai 97.93%. Pada data perempuan akurasi akhir memiliki nilai 93.34%. Saat data digabungkan, akurasi akhir memiliki nilai 90.34%.

Tabel 5.5 Hasil Uji Coba Parameter Gender

| Gender | Akurasi |
|-----------|---------|
| Laki-Laki | 97.93% |
| Perempuan | 93.10% |
| Gabungan | 90.34% |
| Rata-Rata | 93.79% |

5.5 Hasil dan Evaluasi

Pada parameter uji coba pertama, yaitu panjang durasi *frame* pemotongan sinyal, didapatkan hasil akurasi. Pada parameter uji coba pertama, yaitu panjang durasi *frame* pemotongan sinyal, didapatkan hasil rata-rata akurasi terbaik yaitu sebesar 85.17% pada durasi *frame* 20 ms. Data suara yang menjadi input pada proses ekstraksi fitur memiliki rentang durasi rata-rata kurang lebih selama satu detik. Durasi input data suara yang pendek menyebabkan pemotongan *frame* dengan durasi yang panjang akan membuat ciri yang terekstraksi dari proses MFCC kurang spesifik. Oleh karena itu, data terbaik didapatkan pada durasi 20 ms untuk Tugas Akhir ini.

Pada parameter *overlap*, hasil terbaik diperoleh pada durasi 5 ms dengan rata-rata persentase sebesar 85.86%. Jika *overlap*

sinyal terlalu panjang, maka akan mengakibatkan data *frame* sinyal tumpang tindih pada lebih dari 1 *frame*.

Untuk parameter jumlah neuron pada klasifikasi, didapatkan hasil terbaik pada jumlah neuron sebanyak 30 dengan persentase rata-rata akurasi sebesar 95.52%. Sedangkan pada parameter *learning rate*, tidak memberi perubahan pada akurasi yang dihasilkan. Hal ini dikarenakan fungsi pembelajaran yang digunakan adalah *gradient descent with momentum and adaptive learning* (traingdx) dimana *learning rate* akan disesuaikan apabila terlalu tinggi atau terlalu rendah.

Perbedaan data suara laki-laki dan perempuan menghasilkan hasil akurasi yang berbeda, dimana pada data laki-laki akurasi mampu mencapai 97.93%, sedangkan pada data perempuan mencapai 93.10%. Jika data digabungkan dalam proses klasifikasi, akurasi menjadi menurun pada angka 90.34%. Hal ini dikarenakan adanya perbedaan warna suara antara laki-laki perempuan. Pada data laki-laki amplitudo yang dihasilkan lebih besar daripada data suara perempuan. Sehingga proses klasifikasi akan berlangsung lebih akurat jika data dibedakan berdasar *gender*.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Metode *Mel Frequency Cepstral Coefficient* dan *Back Propagation Neural Network* berhasil digunakan untuk pengenalan ucapan Huruf Hijaiyah dengan rata-rata akurasi untuk data laki-laki dan perempuan masing-masing sebesar 97.93% dan 93.10%.
2. Ekstraksi fitur yang optimal dapat dilakukan dengan panjang *frame* 20 ms dan *overlap* 5 ms.
3. *Back Propagation Neural Network* memberikan hasil yang optimal dengan jumlah neuron sebanyak 30. Perubahan parameter *learning rate* tidak memberikan perubahan pada akurasi.
4. Proses klasifikasi akan berlangsung lebih akurat jika data dibedakan berdasar *gender*. Pada data laki-laki akurasi mencapai 97.93% dan pada data perempuan mencapai 93.10%. Jika data digabungkan, akurasi menjadi menurun pada angka 90.34%.

6.2 Saran

Saran yang dapat diberikan dalam pengujian pengenalan pengucapan Huruf Hijaiyah dengan MFCC adalah sebagai berikut:

1. Menambah jumlah data latihan setiap huruf dan variasi subyek sehingga dapat mempelajari pola yang lebih banyak lagi.
2. Mengembangkan proses untuk menghilangkan *noise* pada

sinyal suara sehingga dapat diperoleh data yang akurat untuk ekstraksi fitur pada berbagai kondisi perekaman suara.

3. Perlu adanya penelitian lanjutan terkait metode ekstraksi fitur yang mampu membedakan pengucapan pada berbagai macam tipe suara.

DAFTAR PUSTAKA

- [1] Anonim, “Situasi Penyandang Disabilitas,” Kementerian Kesehatan RI, Jakarta, 2004.
- [2] Anonim, “Indonesia - Sensus Penduduk 2010,” Badan Pusat Statistik, Jakarta, 2014.
- [3] A. Diono, “Buletin Jendela Data & Informasi Kesehatan,” Kementerian Kesehatan RI, Jakarta, 2014.
- [4] Anonim, “Kebutuhan,” Wikipedia, 14 March 2016. [Online]. Available: <https://id.wikipedia.org/wiki/Kebutuhan>. [Diakses 20 June 2016].
- [5] B. Putra, B. T. Atmaja dan D. Prananto, “Developing Speech Recognition System for Quranic Verse Recitation Learning Software,” *International Journal on Informatics for Development (IJID)*, vol. 1, no. 2, pp. 14-21, 2012.
- [6] A. N. Wahidah, M. S. Suriazalmi, M. L. Niza, H. Rosyati, N. Faradila, A. Hasan, A. K. Rohana dan Z. N. Farizan, “Makhraj Recognition Using Speech Processing,” dalam *Computing and Convergence Technology (ICCCT)*, Seoul, 2012.
- [7] Anonim, “Refreshable braille display,” Wikipedia, 18 April 2016. [Online]. Available: en.wikipedia.org/wiki/Refreshable_braille_display. [Diakses 20 April 2016].
- [8] M. Alsulaiman, G. Muhammad dan Z. Ali, “Comparison of Voice Features for Arabic Speech Recognition,” dalam *International Conference on Digital Information Management (ICDIM)*, Melbourn, 2011.
- [9] Anonim, “Braille,” Wikipedia, 15 Maret 2011. [Online]. Available: <https://id.wikipedia.org/wiki/Braille>. [Diakses 15 Juni 2017].

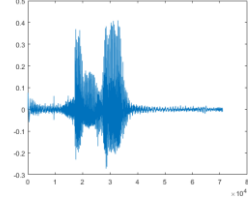
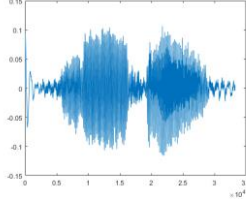
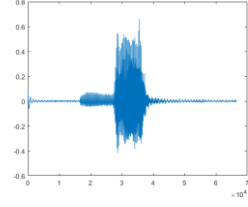
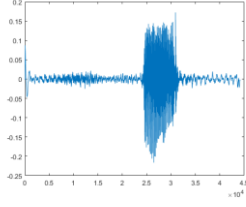
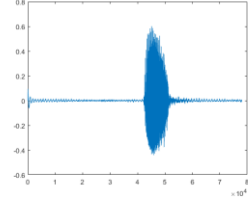
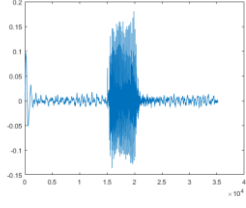
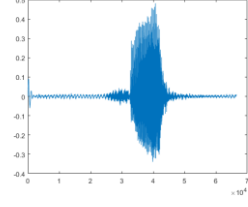
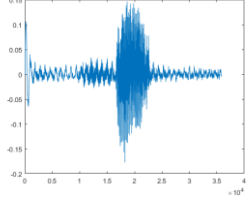
- [10] Anonim, “Arabic Braille,” Wikipedia, 16 Oktober 2016. [Online]. Available: https://en.wikipedia.org/wiki/Arabic_Braille. [Diakses 17 Juni 2017].
- [11] Anonim, “Refreshable braille display,” Wikipedia, 8 Juni 2017. [Online]. Available: https://en.wikipedia.org/wiki/Refreshable_braille_display. [Diakses 17 Juni 2017].
- [12] M. Banzi, *Getting Started with Arduino*, Sebastopol: Make:Books, 2012.
- [13] Ismail, “Advantages and Disadvantages of Using Arduino,” Engineer Experiences, 1 Oktober 2016. [Online]. Available: <http://engineerexperiences.com/advantages-and-disadvantages.html>. [Diakses 18 Juni 2017].
- [14] Anonim, “Arduino Nano,” Arduino, 12 Desember 2014. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardNano>. [Diakses 18 Juni 2017].
- [15] N. Safaat, “Pengenalan Android,” dalam *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Bandung, Informatika, 2015, p. 1.
- [16] S. Primorac dan M. Russo, “Android Application for Sending SMS Messages with Speech Recognition Interface,” dalam *MIPRO*, Opatija, 2012.
- [17] Y. R. B. T. W. Yusuf Bagas Saputro, “Pengembangan Sistem Informasi Kesiswaan Berbasis Android Studi Kasus SMA Negeri 1 Tenganan,” dalam *Seminar Nasional SAINTEK III*, Kupang, 2016.
- [18] A. Setiawan, A. Hidayatno dan R. R. Isnanto, “Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk

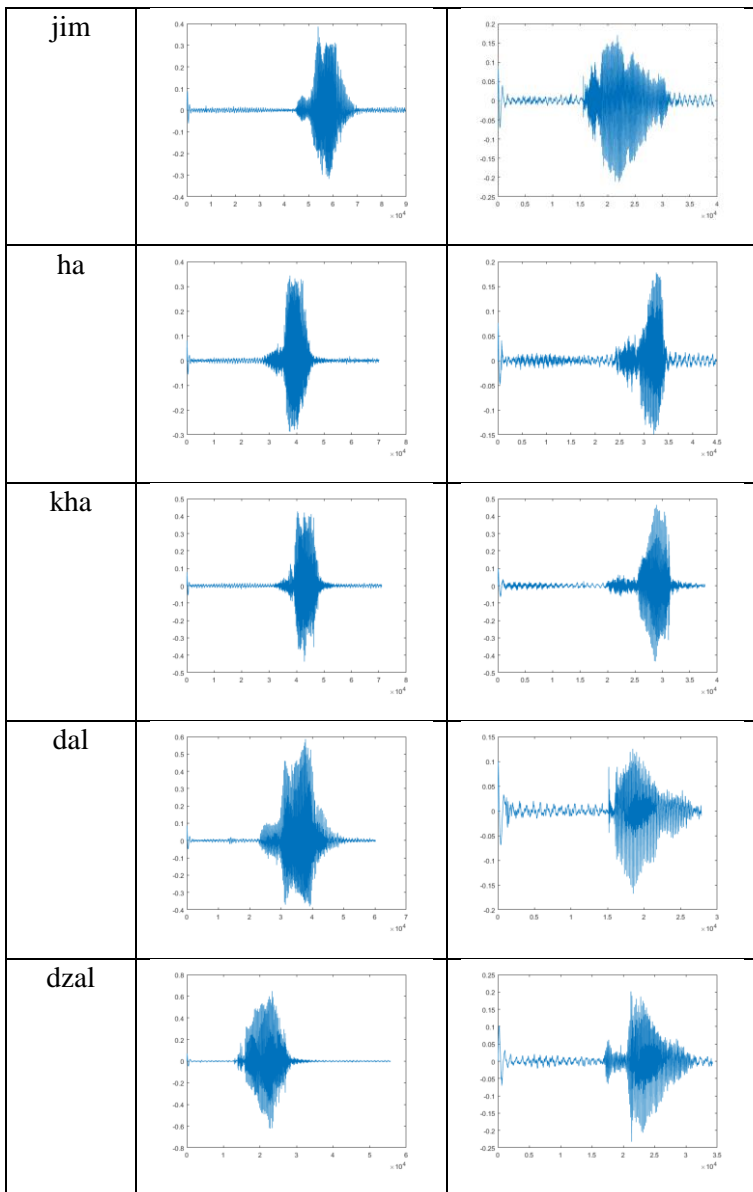
- Mengoperasikan Kursor Komputer,” *TRANSMISI*, vol. 13, no. 3, pp. 82-86, 2011.
- [19] S. K. Saksamudre, P. Shrishrimal dan R. Deshmukh, “A Review on Different Approaches for Speech Recognition System,” *International Journal of Computer Applications*, vol. 115, no. 22, pp. 23-28, 2015.
- [20] K. Sakhnov, Member, IAENG, E. Verteletskaya dan B. Simak, “Dynamical Energy-Based Speech/Silence Detector for Speech Enhancement Applications,” dalam *World Congress on Engineering*, London, 2009.
- [21] A. Farchi, S. Mounir, B. Mounir dan J. Elabbadi, “Performance of Voice Activity Detection Method Based on Zero Crossing Rate and Energy Level In Arabic Language,” *International Research Journal of Mathematics, Engineering and IT*, vol. 2, no. 10, pp. 21-29, 2015.
- [22] A. Buono, W. Jatmiko dan B. Kusumoputro, “Perluasan Metode MFCC 1D ke 2D Sebagai Esktraksi Ciri Pada Sistem Identifikasi Pembicara Menggunakan Hidden Markov Model (HMM),” *MAKARA*, vol. 13, no. 1, pp. 87-93, 2009.
- [23] K. S. Rao dan A. K. Vuppala, “Appendix A MFCC Features,” dalam *Speech Processing in Mobile Environments*, Switzerland, Springer, 2014, pp. 103-105.
- [24] S. M. Azam, Z. A. Mansoor, M. S. Mughal dan S. Mohsin, “Urdu Spoken Digits Recognition Using Classified MFCC and Backproppgation Neural Network,” dalam *Computer Graphics, Imaging, and Visualisation*, Bangkok, 2007.
- [25] W. Mulyadani, “Robot Lengan Pengambil Benda untuk Membantu Pasien dengan Perintah Suara Menggunakan Metode MFCC dan Neural Network,” Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [26] M. Panji, “Muhammad Panji,” 14 Februari 2008. [Online]. Available: <https://panji.web.id/files/skripsi/bab-iii.pdf>. [Diakses 18 Juni 2017].

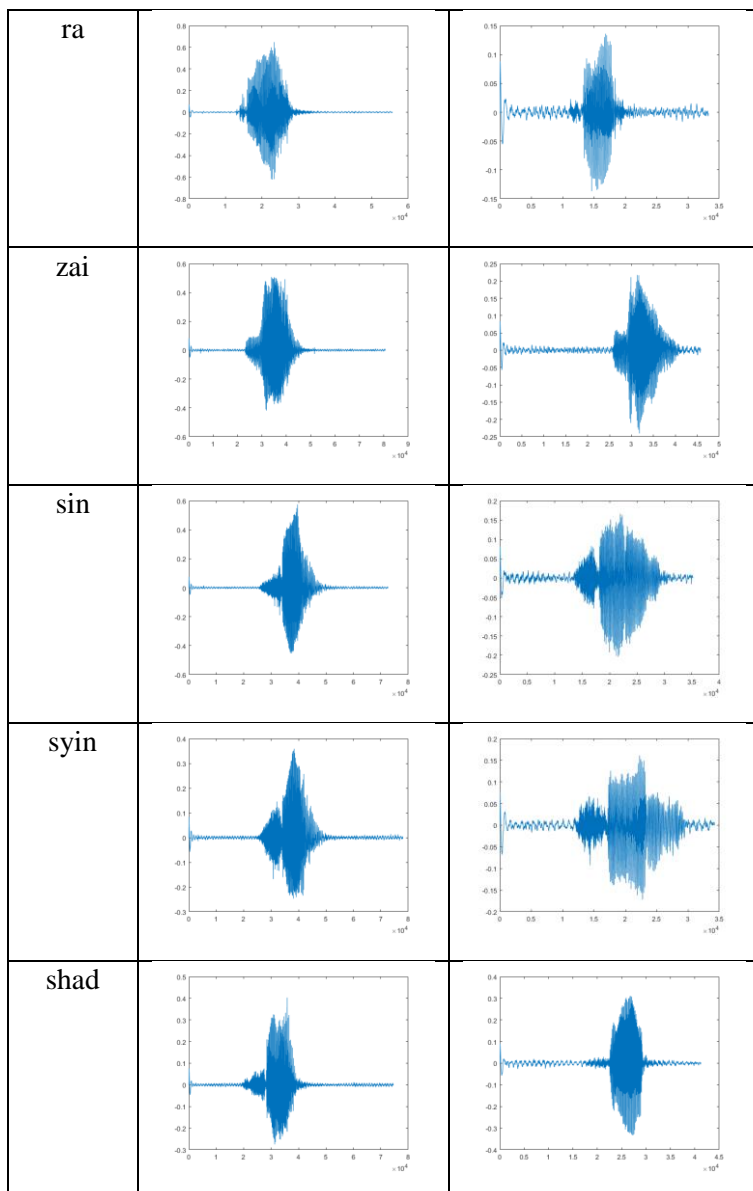
- [27] Anonim, “Abjad Arab,” Wikipedia, 15 Mei 2017. [Online]. Available: https://id.wikipedia.org/wiki/Abjad_Arab. [Diakses 15 Juni 2017].

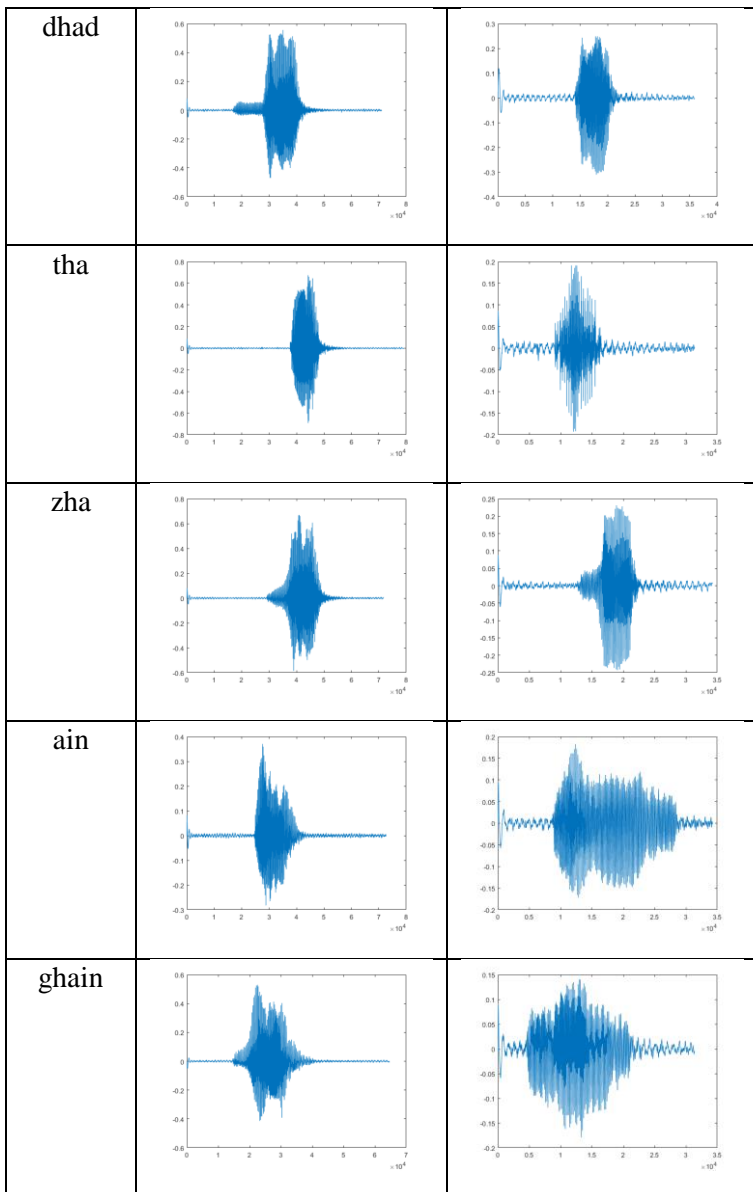
LAMPIRAN

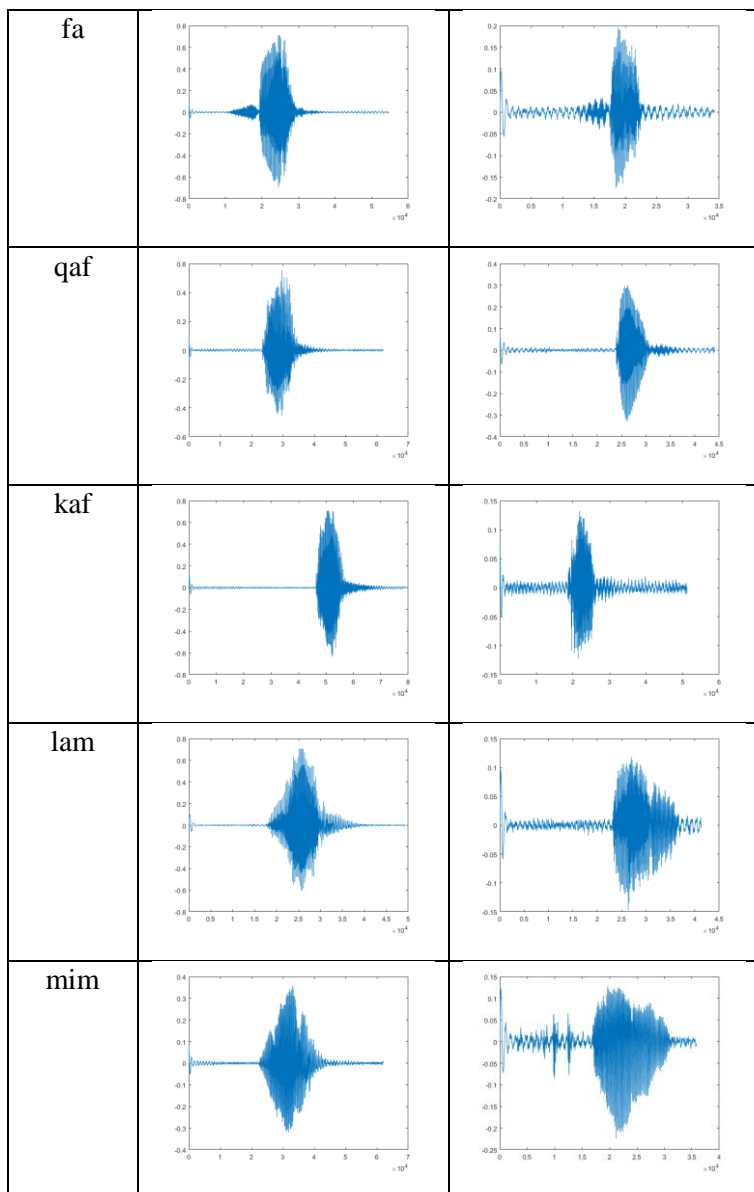
A. Data Sinyal Suara

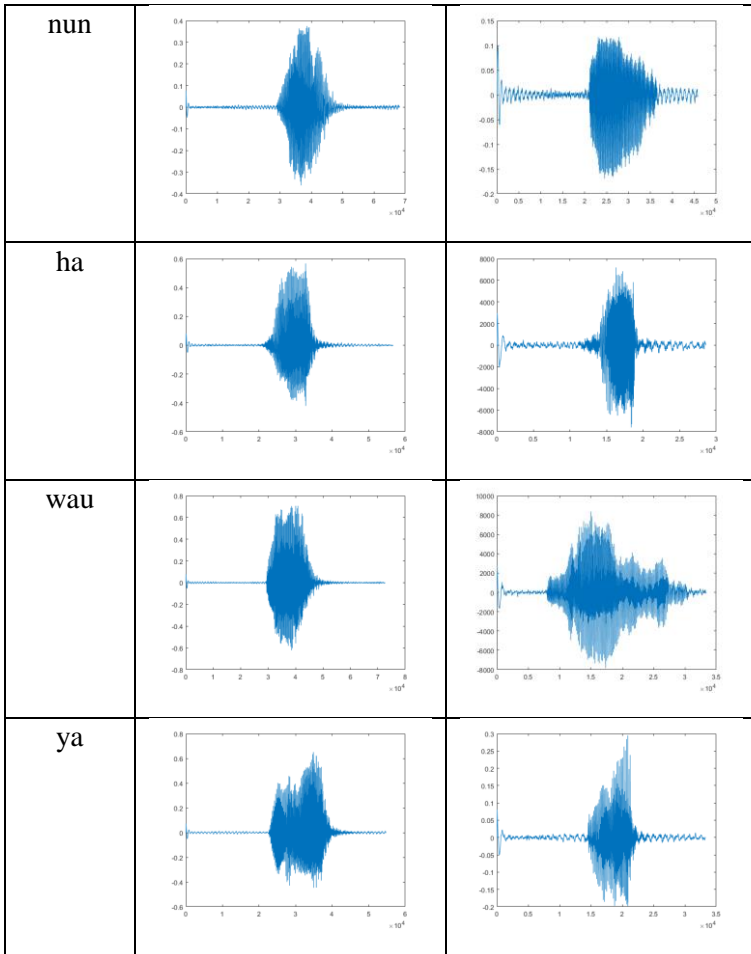
| Huruf | Data Laki-Laki | Data Perempuan |
|--------|---|---|
| hamzah |  A waveform plot for the male voice of the letter 'hamzah'. The x-axis is labeled from 0 to 8 with a multiplier of $\times 10^4$. The y-axis ranges from -0.3 to 0.5. The signal shows a complex burst of energy between 2 and 4 on the x-axis, with a peak amplitude of approximately 0.4. |  A waveform plot for the female voice of the letter 'hamzah'. The x-axis is labeled from 0 to 3.5 with a multiplier of $\times 10^4$. The y-axis ranges from -0.15 to 0.15. The signal shows a complex burst of energy between 1 and 2.5 on the x-axis, with a peak amplitude of approximately 0.1. |
| ba |  A waveform plot for the male voice of the letter 'ba'. The x-axis is labeled from 0 to 7 with a multiplier of $\times 10^4$. The y-axis ranges from -0.8 to 0.8. The signal shows a complex burst of energy between 3 and 4 on the x-axis, with a peak amplitude of approximately 0.6. |  A waveform plot for the female voice of the letter 'ba'. The x-axis is labeled from 0 to 4.5 with a multiplier of $\times 10^4$. The y-axis ranges from -0.25 to 0.2. The signal shows a complex burst of energy between 2.5 and 3.5 on the x-axis, with a peak amplitude of approximately 0.15. |
| ta |  A waveform plot for the male voice of the letter 'ta'. The x-axis is labeled from 0 to 8 with a multiplier of $\times 10^4$. The y-axis ranges from -0.8 to 0.8. The signal shows a complex burst of energy between 4 and 5 on the x-axis, with a peak amplitude of approximately 0.6. |  A waveform plot for the female voice of the letter 'ta'. The x-axis is labeled from 0 to 4 with a multiplier of $\times 10^4$. The y-axis ranges from -0.15 to 0.2. The signal shows a complex burst of energy between 1.5 and 2.5 on the x-axis, with a peak amplitude of approximately 0.15. |
| tsha |  A waveform plot for the male voice of the letter 'tsha'. The x-axis is labeled from 0 to 7 with a multiplier of $\times 10^4$. The y-axis ranges from -0.4 to 0.5. The signal shows a complex burst of energy between 3 and 4 on the x-axis, with a peak amplitude of approximately 0.4. |  A waveform plot for the female voice of the letter 'tsha'. The x-axis is labeled from 0 to 4 with a multiplier of $\times 10^4$. The y-axis ranges from -0.2 to 0.15. The signal shows a complex burst of energy between 1.5 and 2.5 on the x-axis, with a peak amplitude of approximately 0.15. |



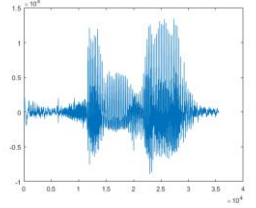
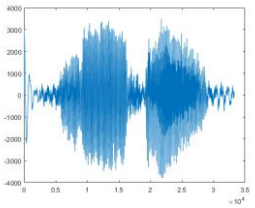
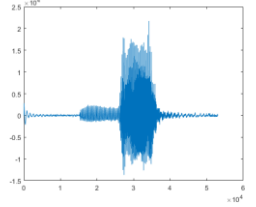
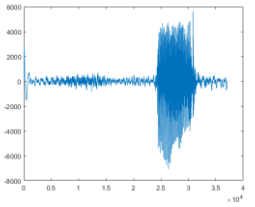
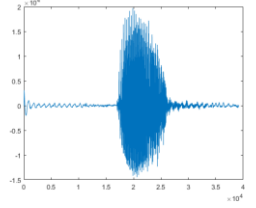
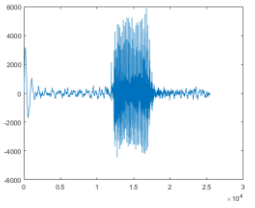
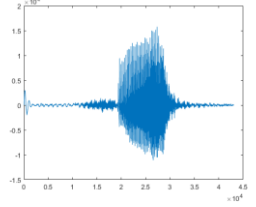
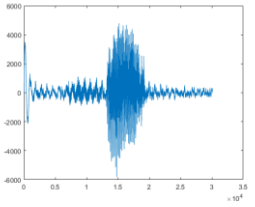


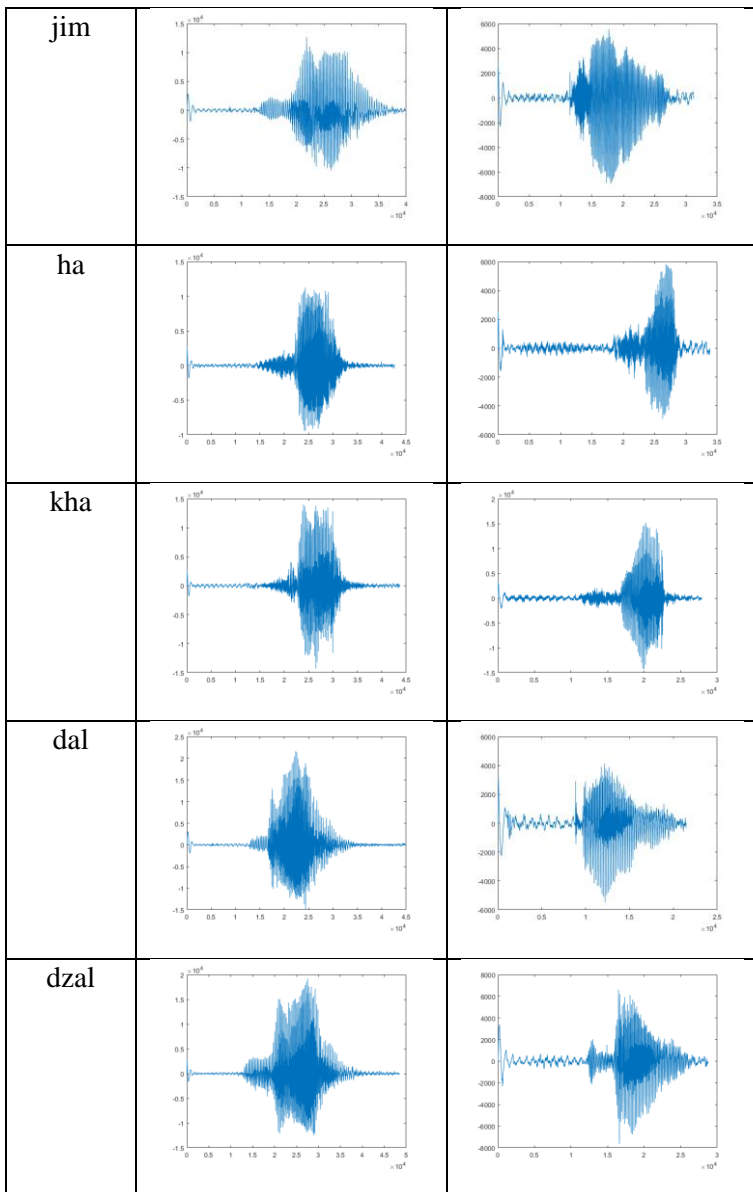


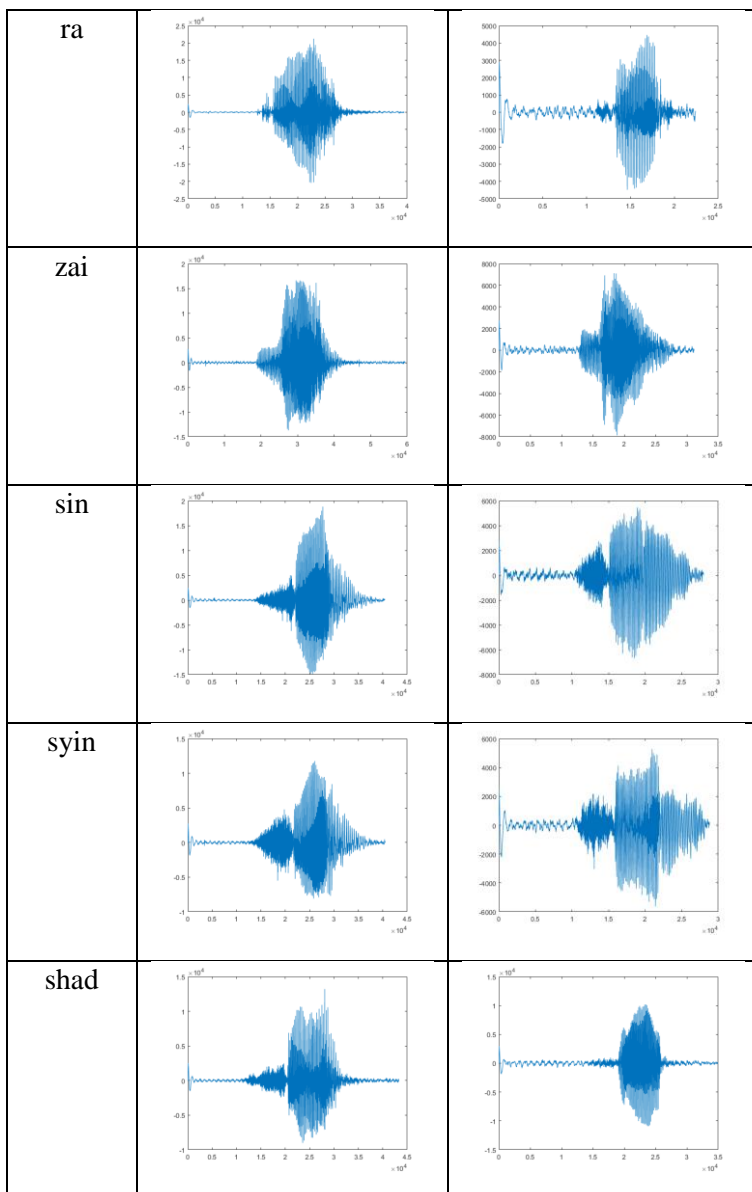


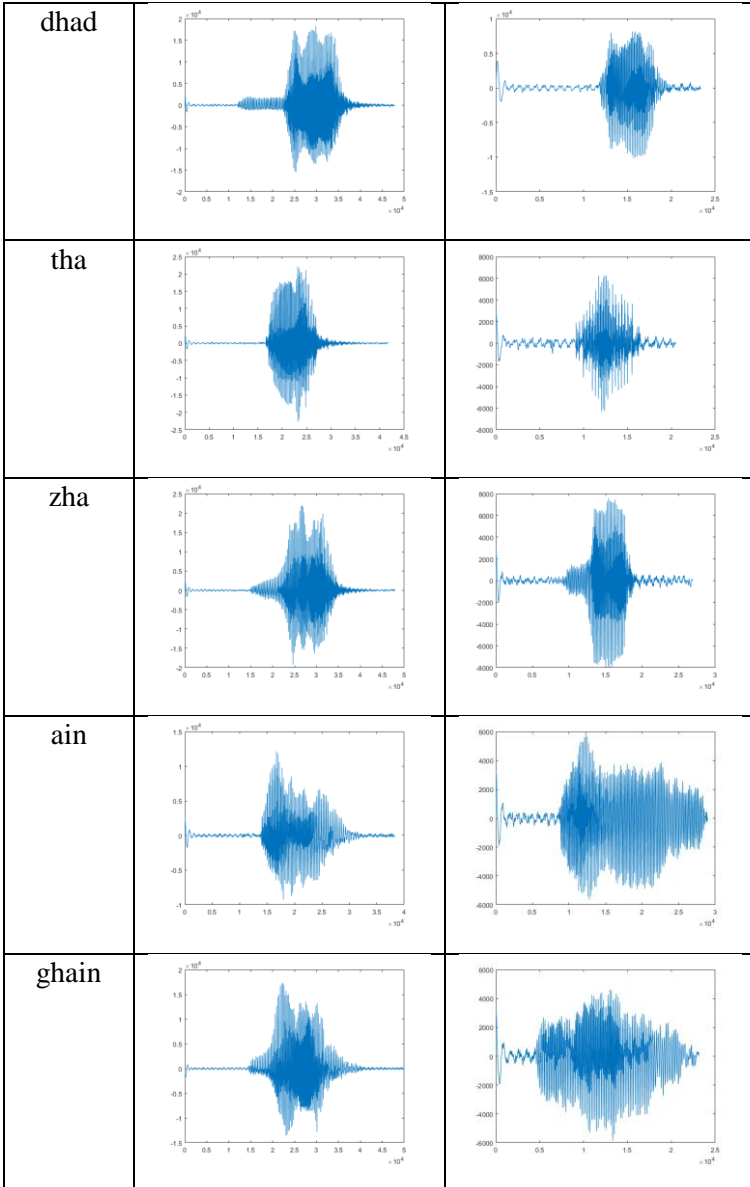


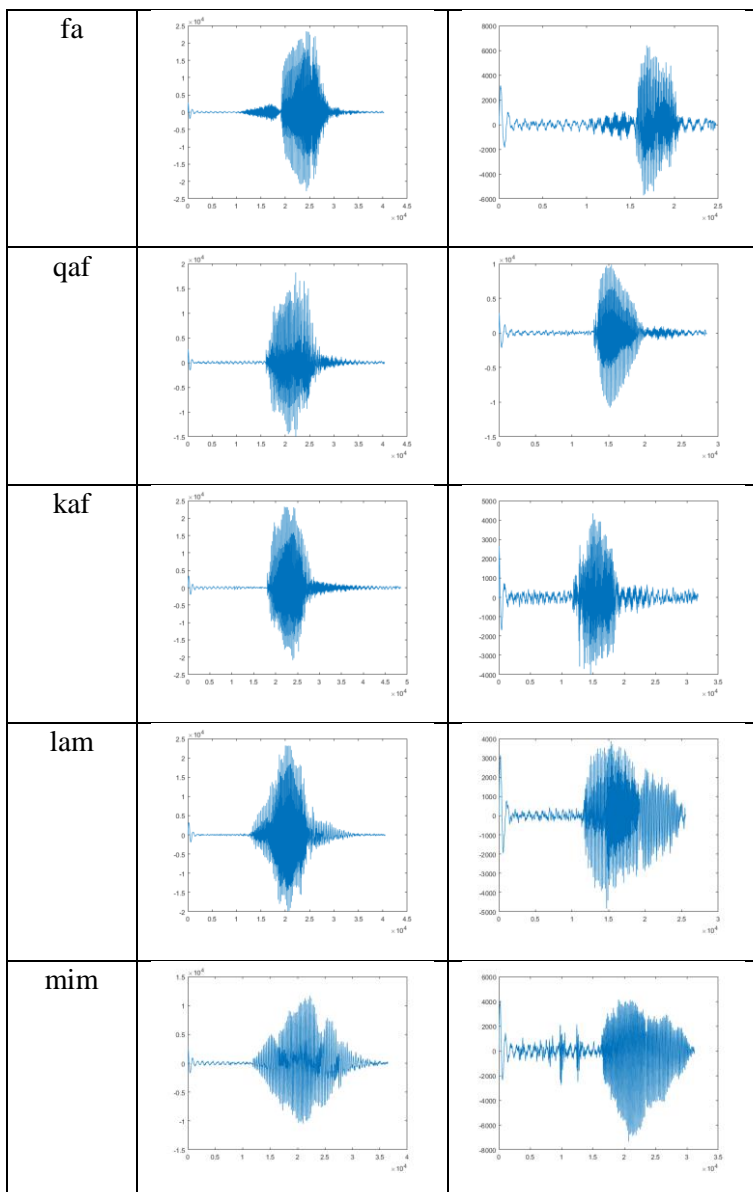
B. Hasil Pemotongan Sinyal

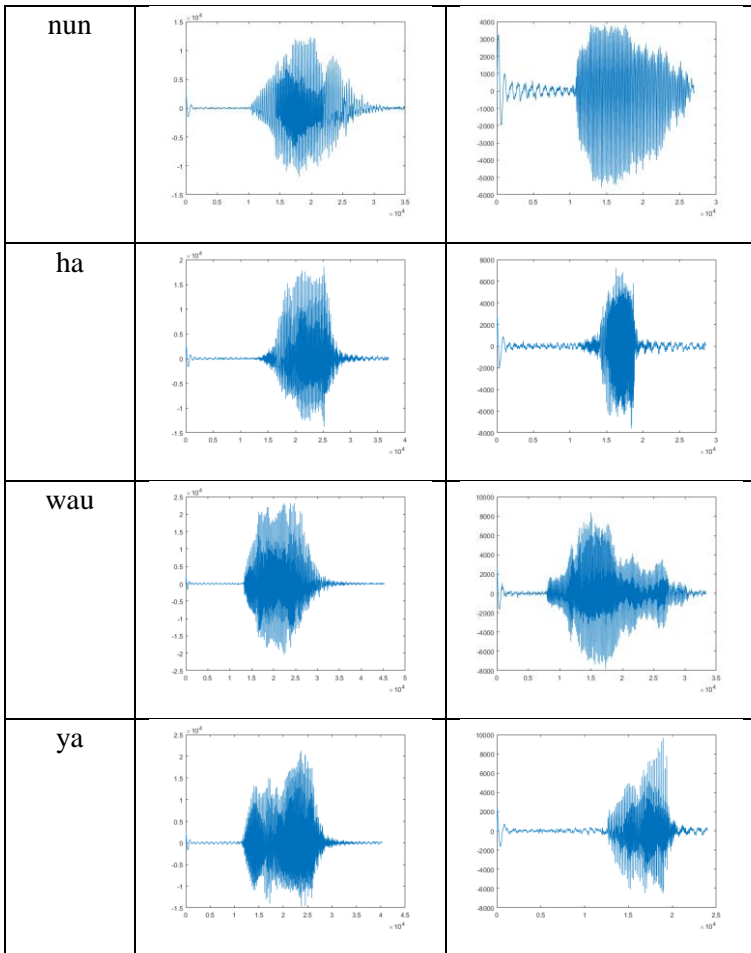
| Huruf | Data Laki-Laki | Data Perempuan |
|--------|---|---|
| hamzah |  |  |
| ba |  |  |
| ta |  |  |
| tsa |  |  |



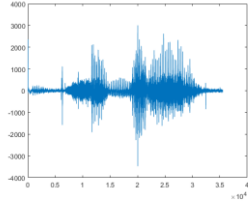
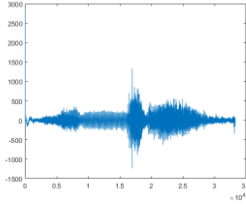
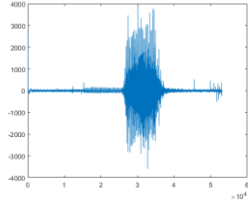
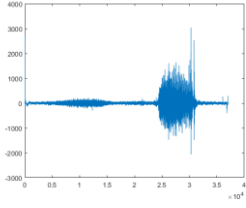
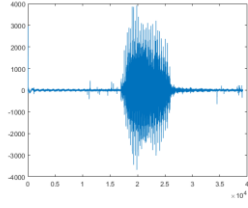
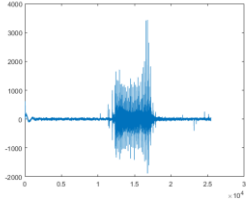
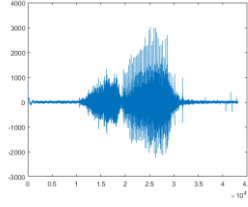
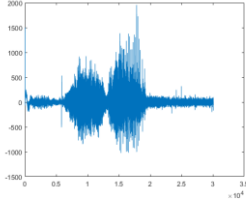


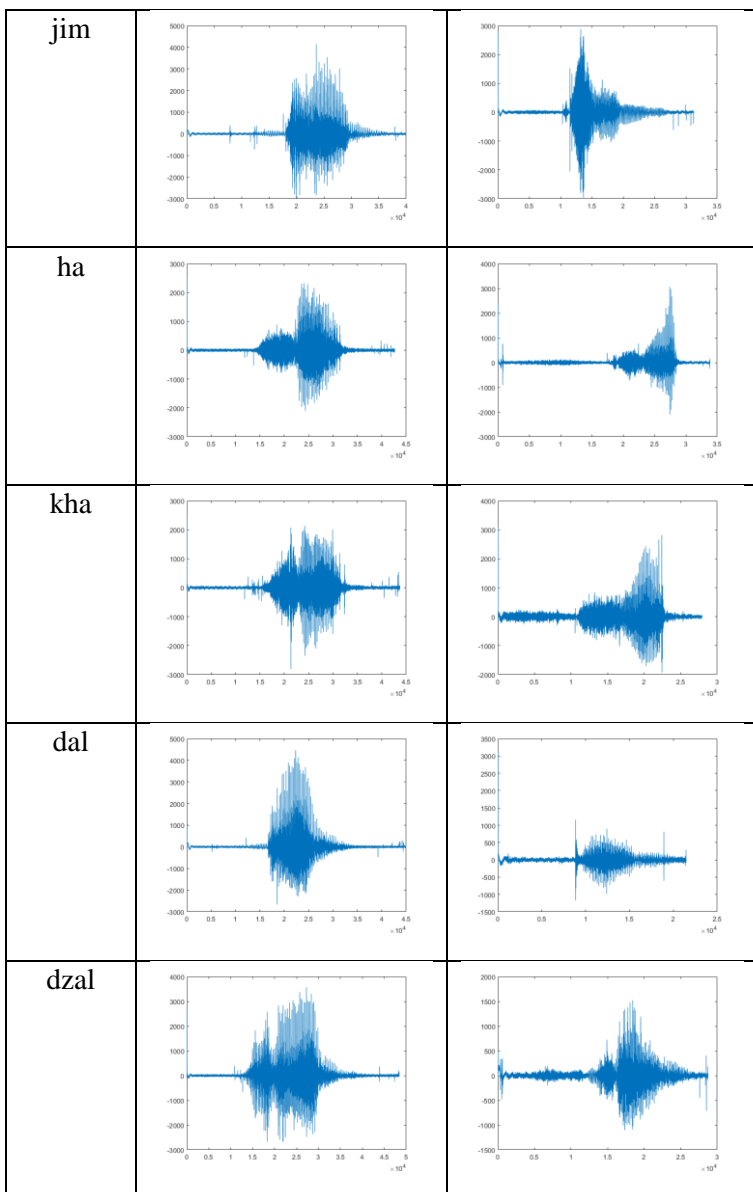


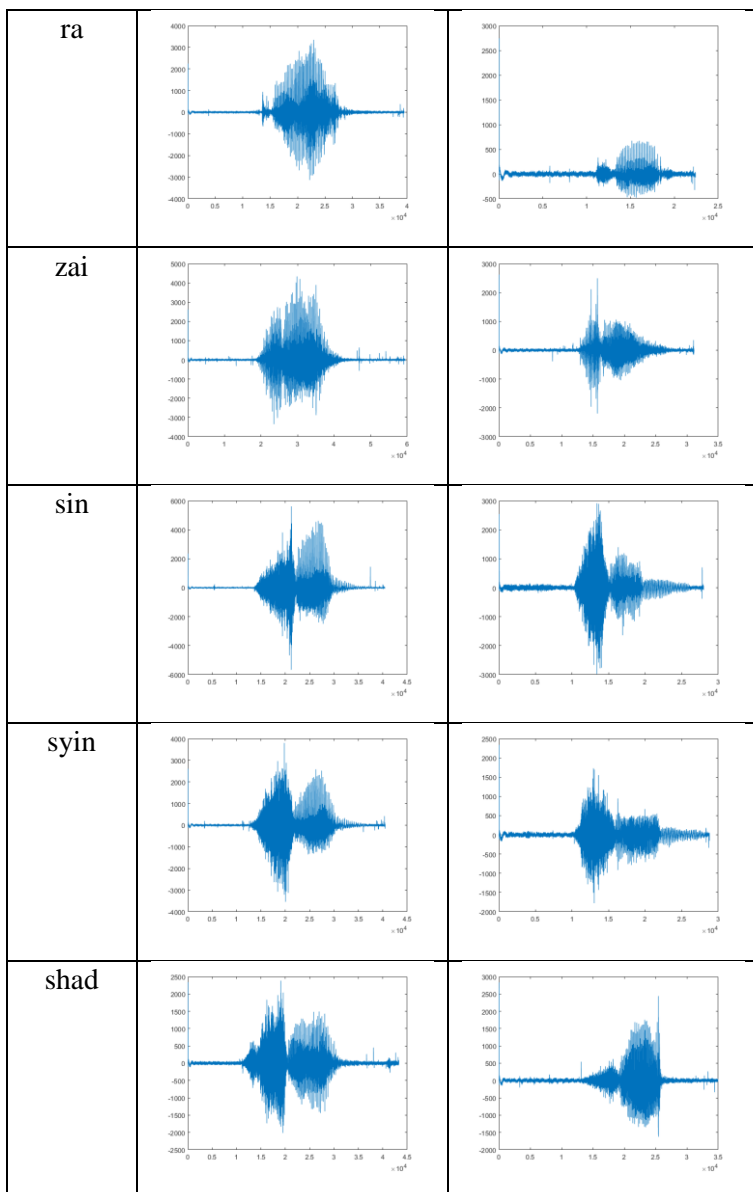


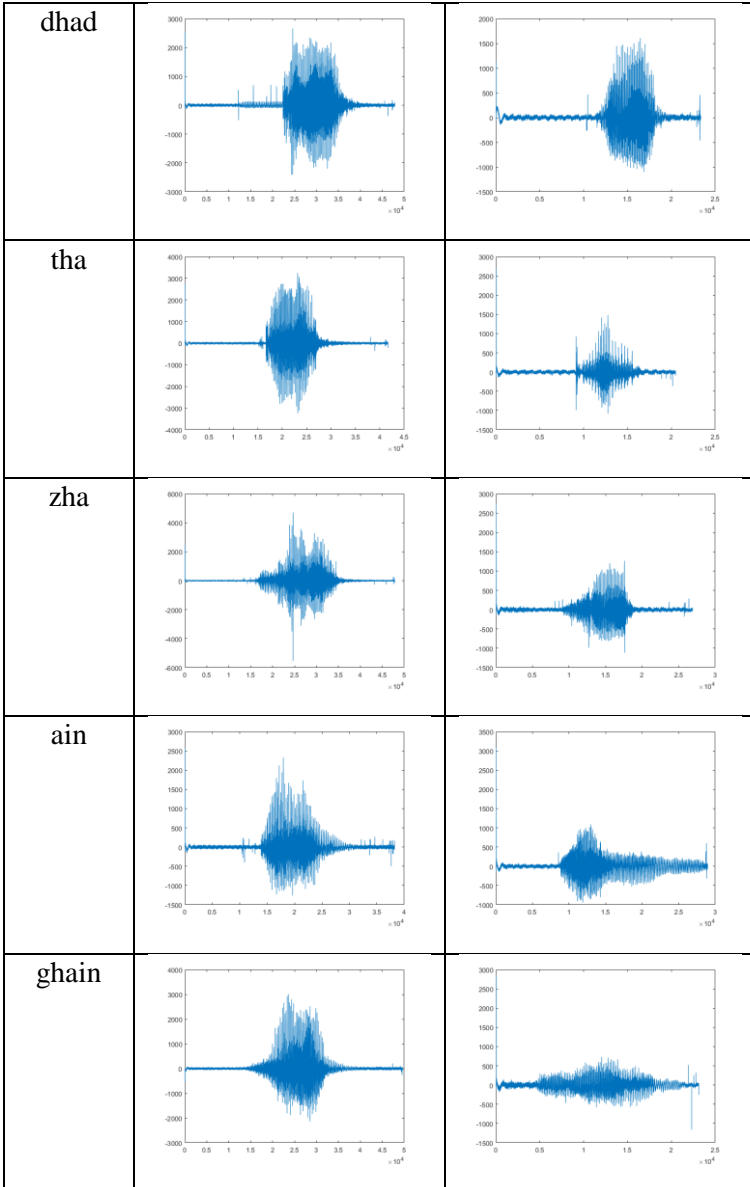


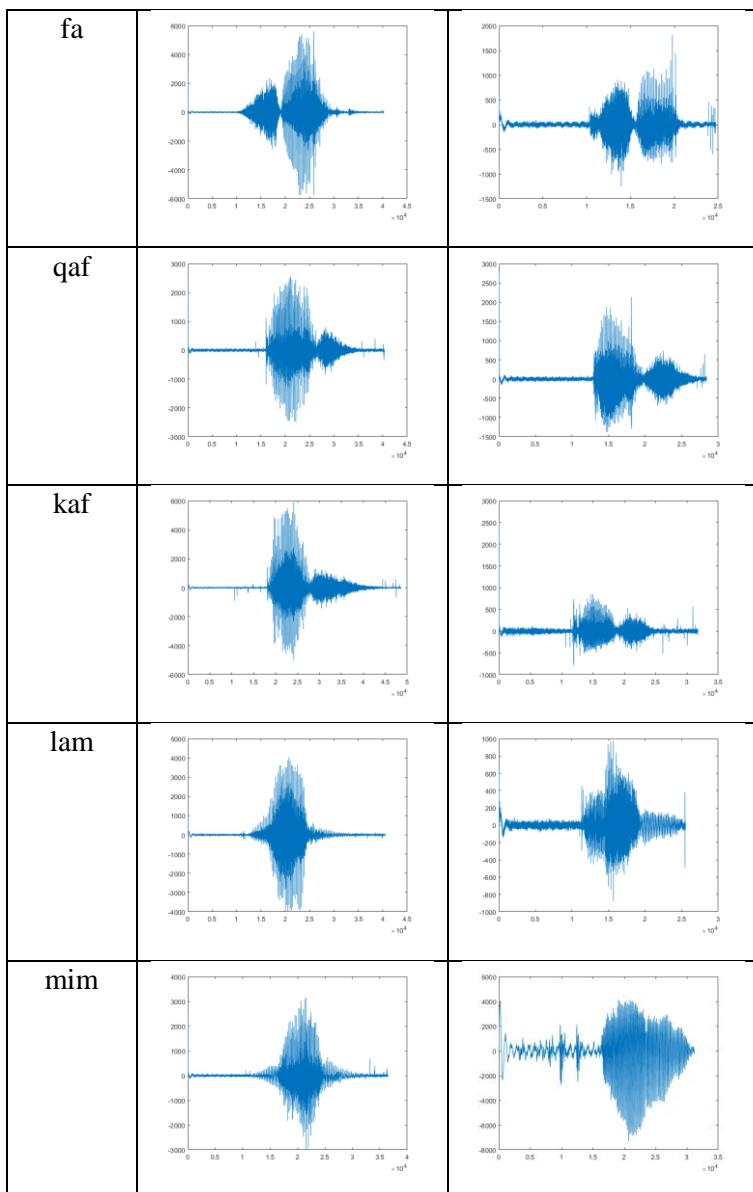
C. Hasil *Preemphasis Filter*

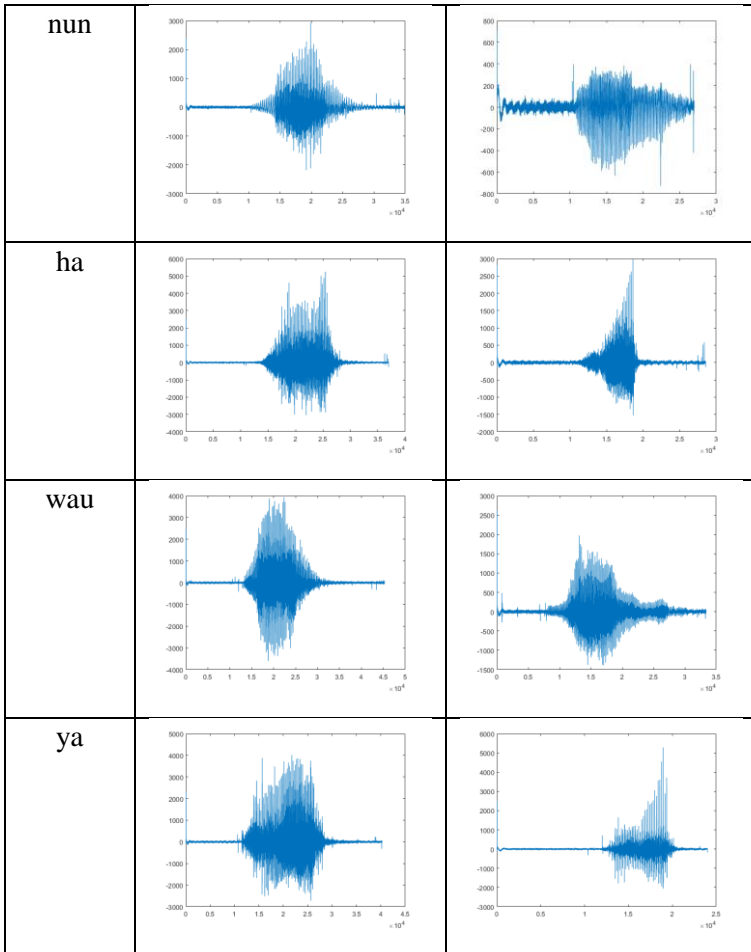
| Huruf | Data Laki-Laki | Data Perempuan |
|--------|---|---|
| hamzah |  |  |
| ba |  |  |
| ta |  |  |
| tsha |  |  |



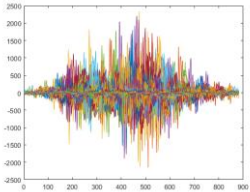
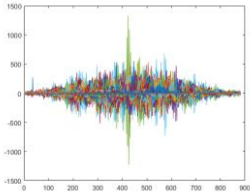
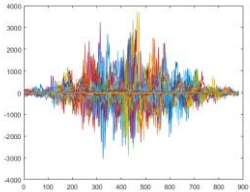
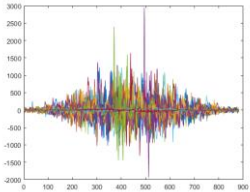
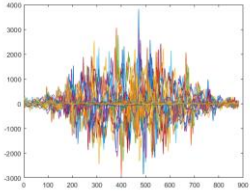
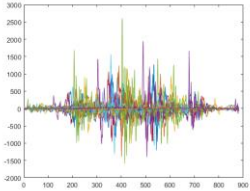
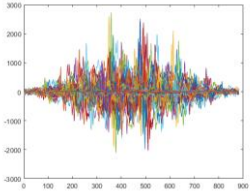
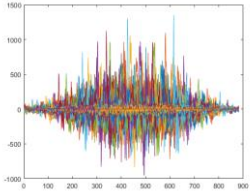


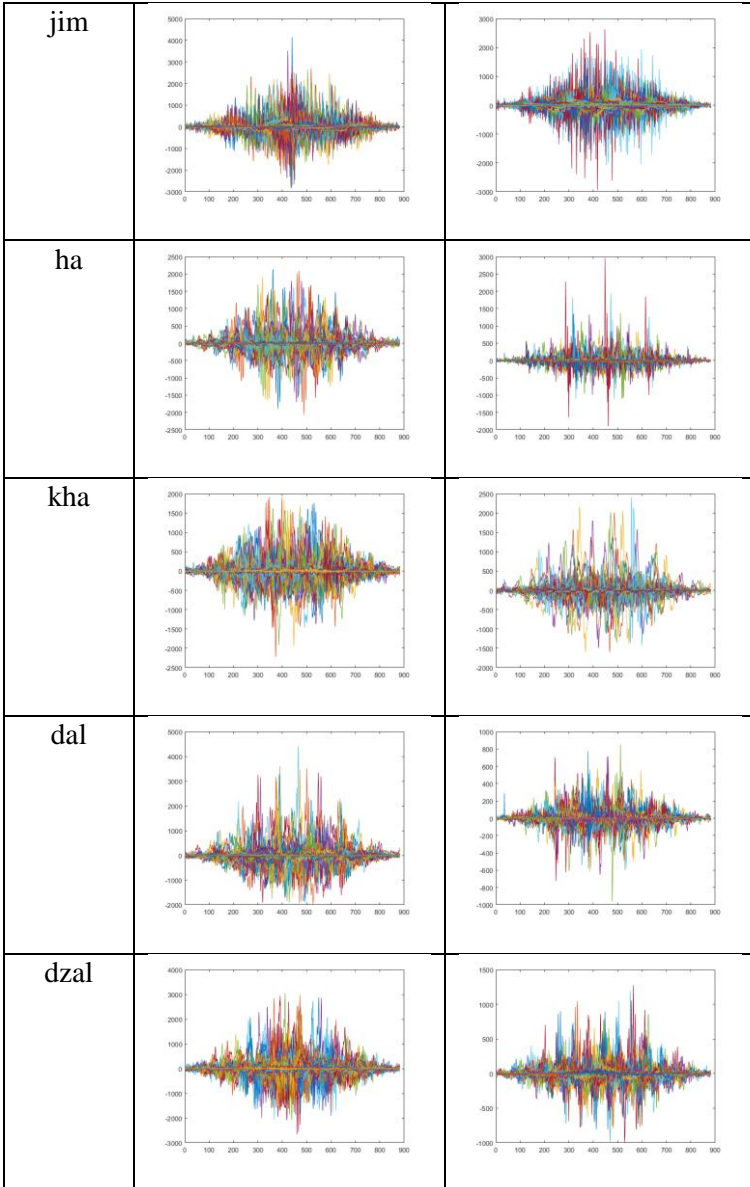


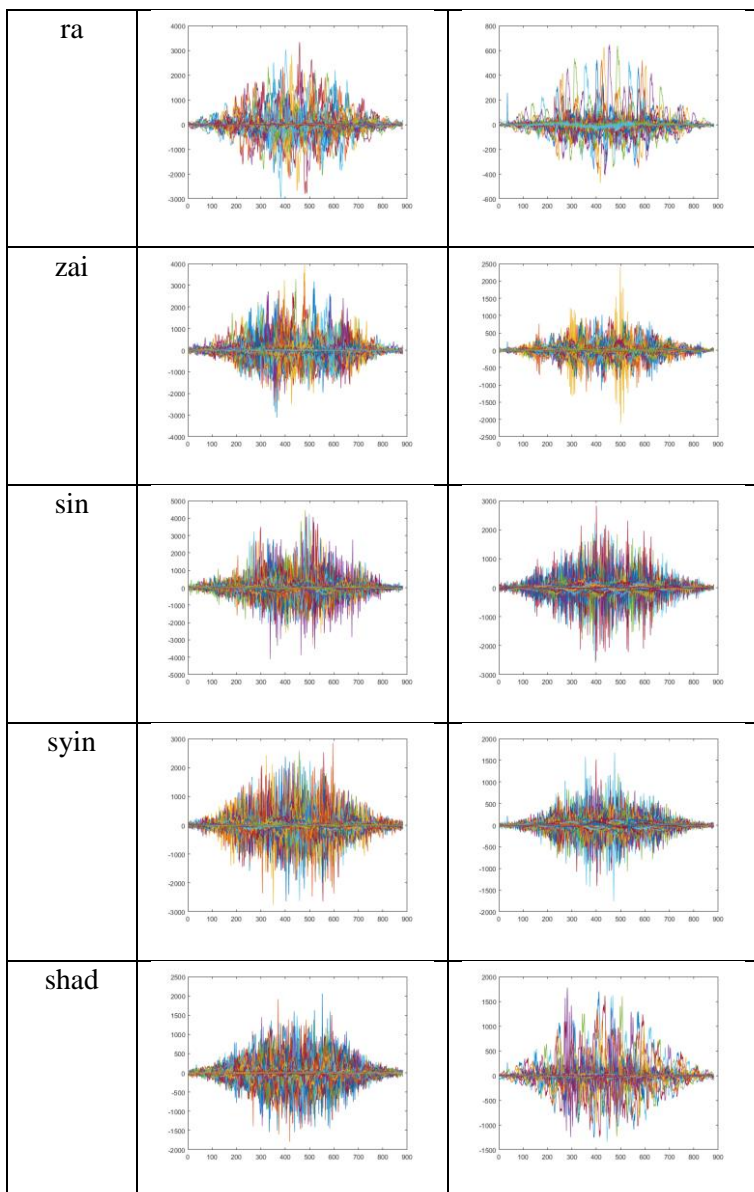


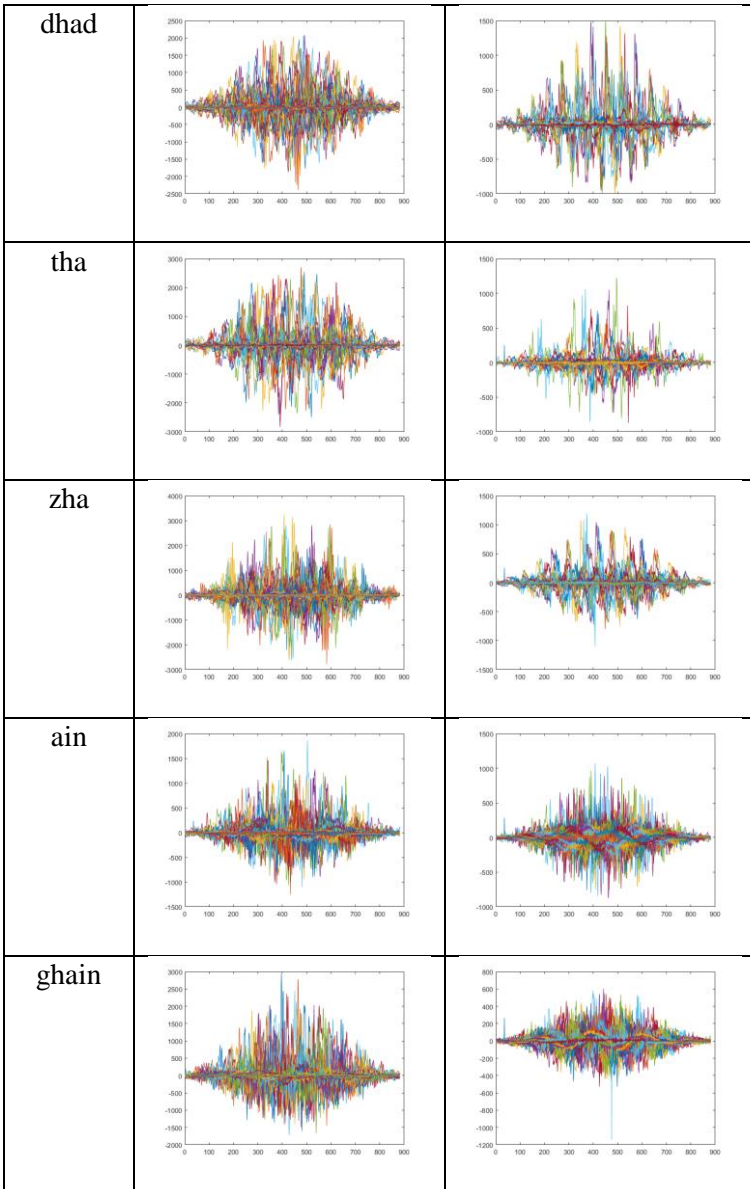


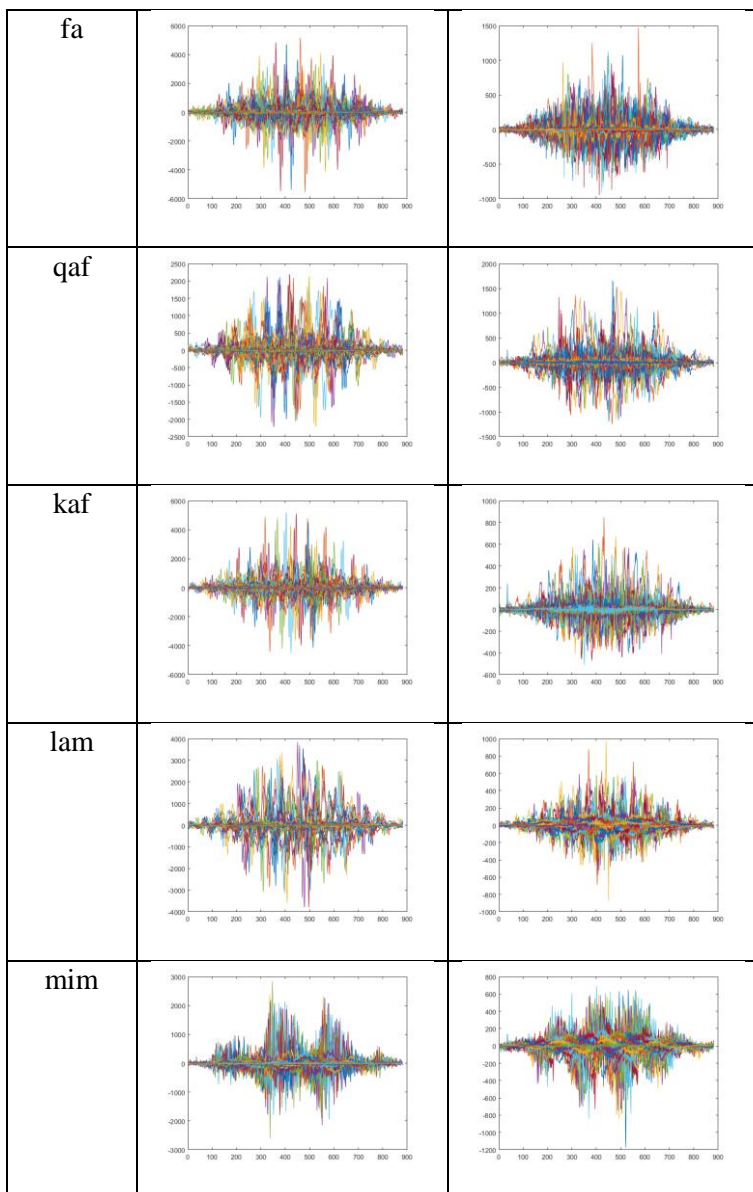
D. Hasil *Frame Blocking* Sinyal

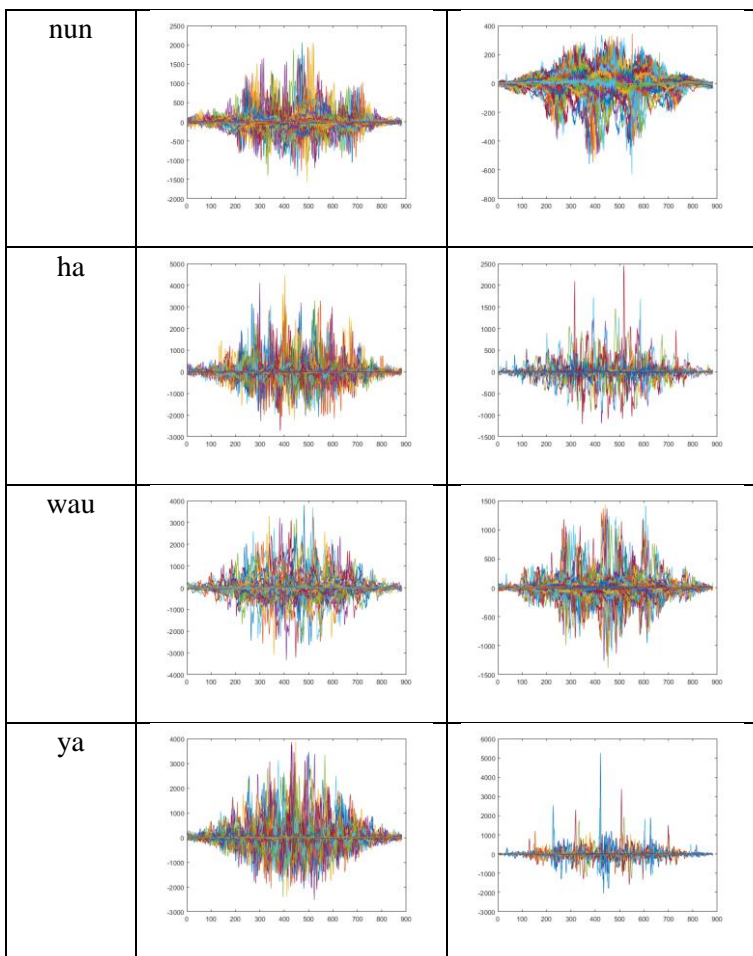
| Huruf | Data Laki-Laki | Data Perempuan |
|--------|---|---|
| hamzah |  |  |
| ba |  |  |
| ta |  |  |
| tsa |  |  |



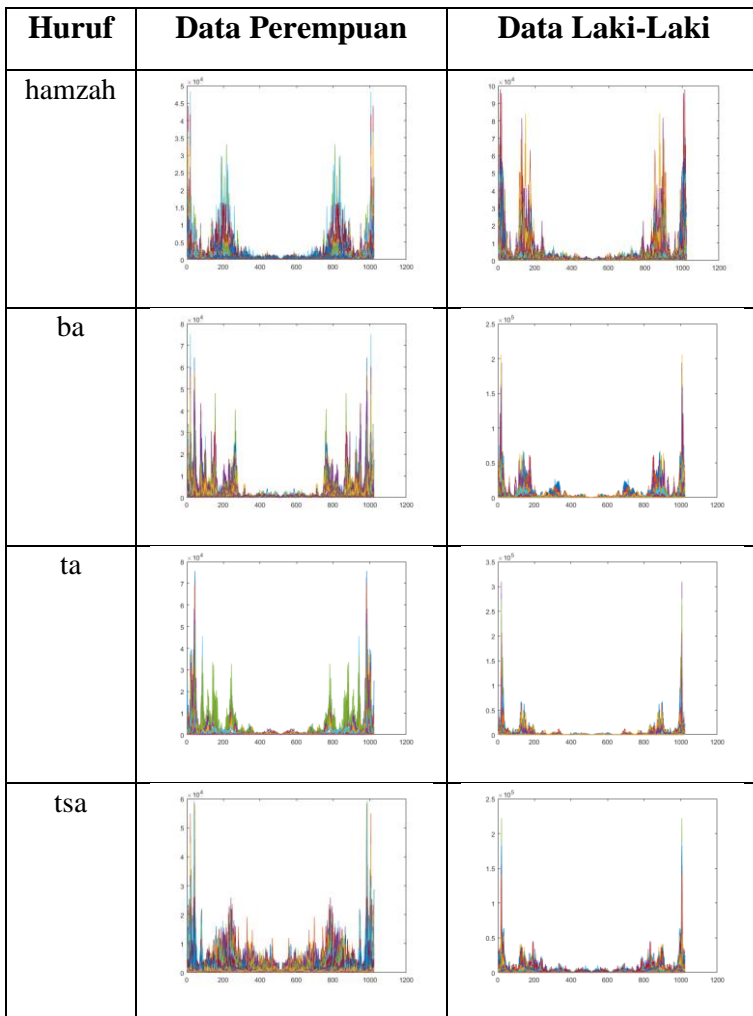


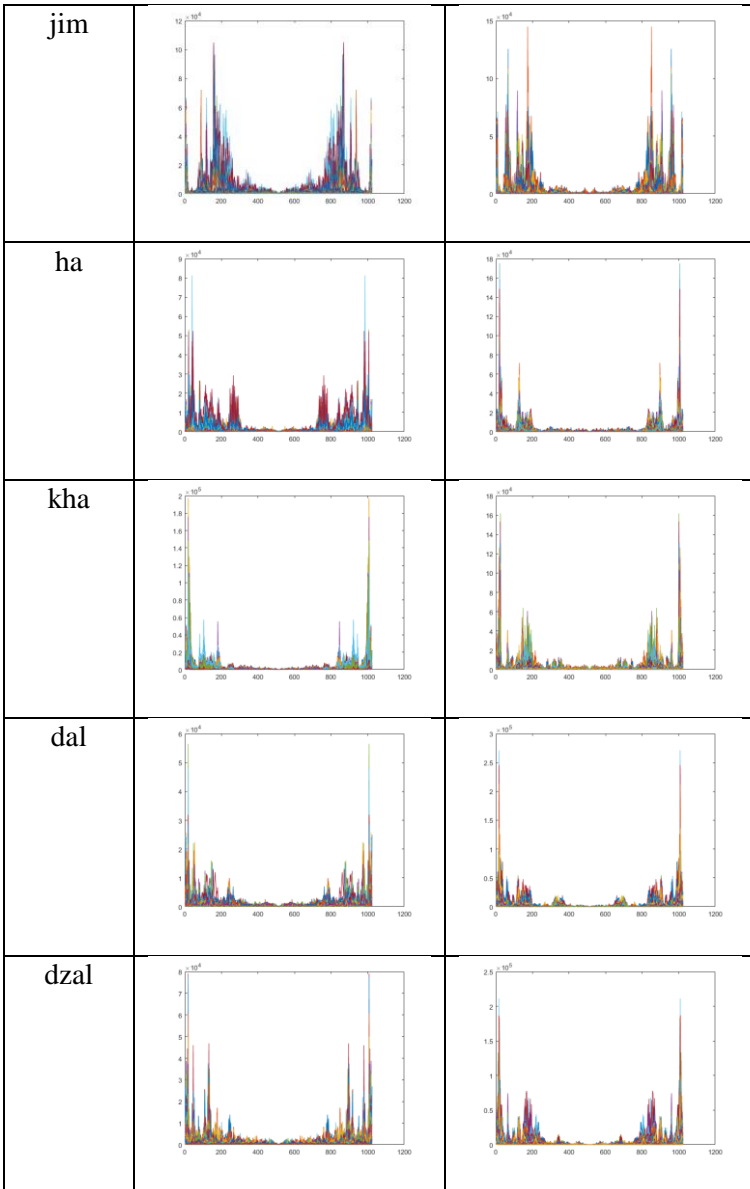


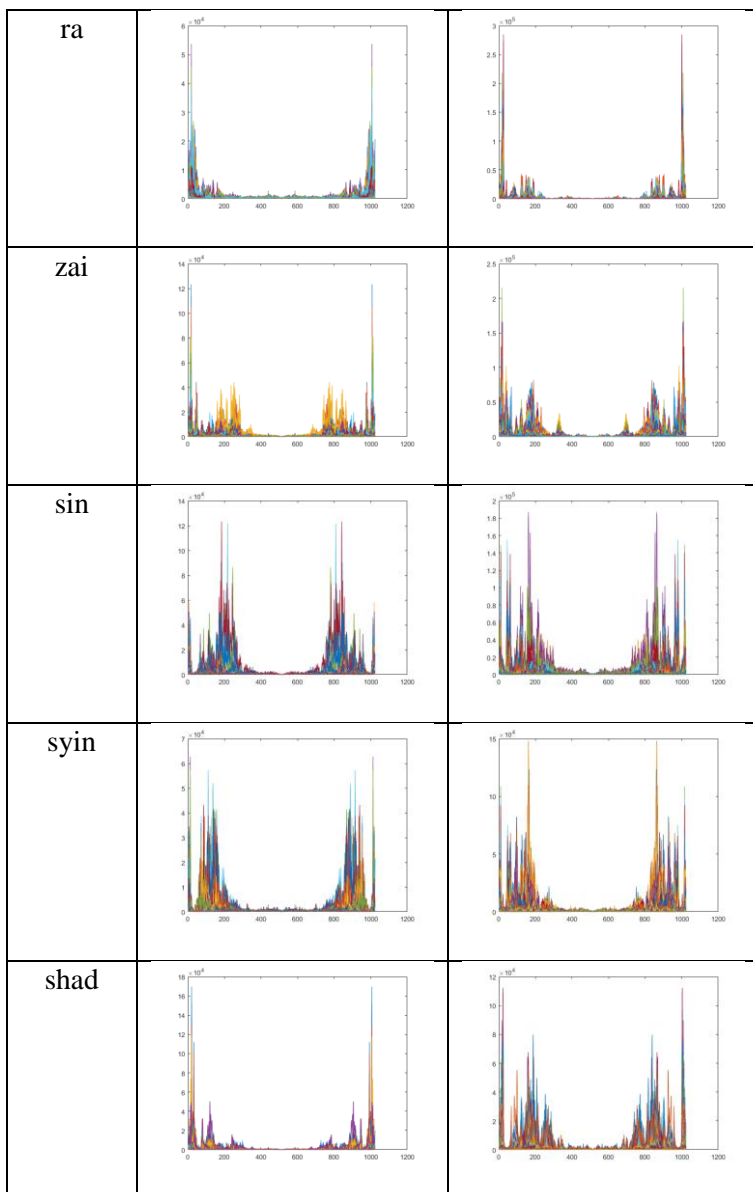


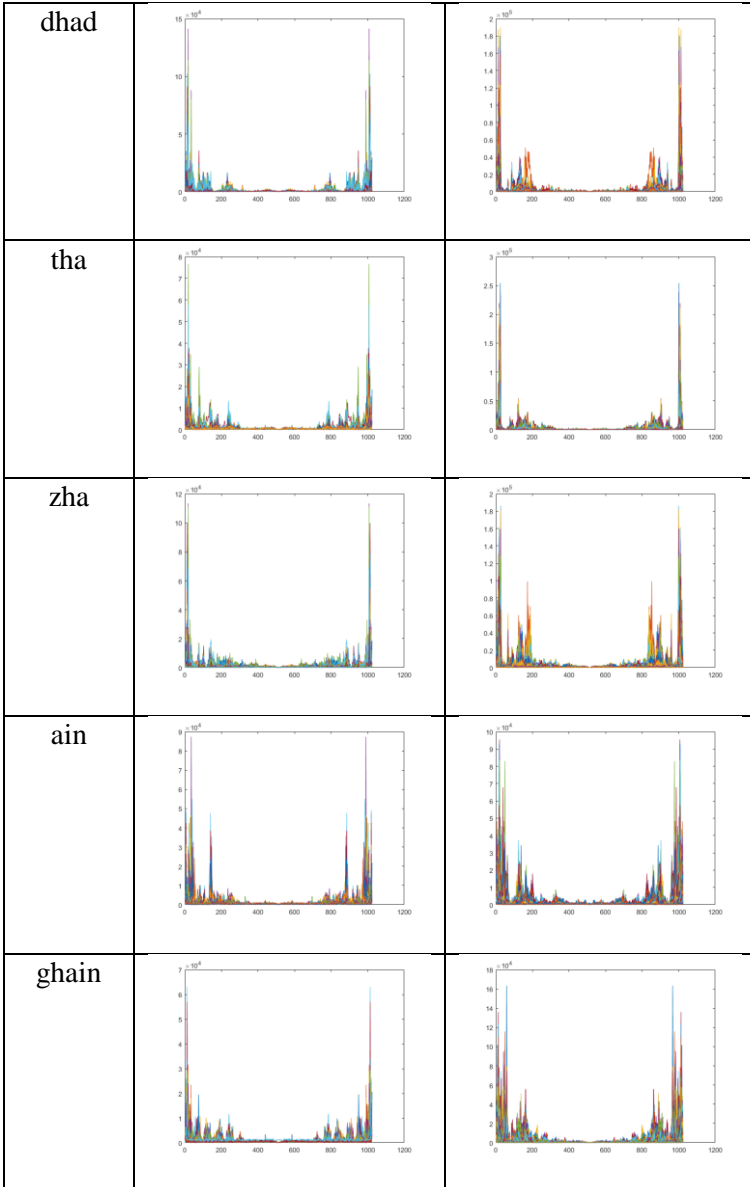


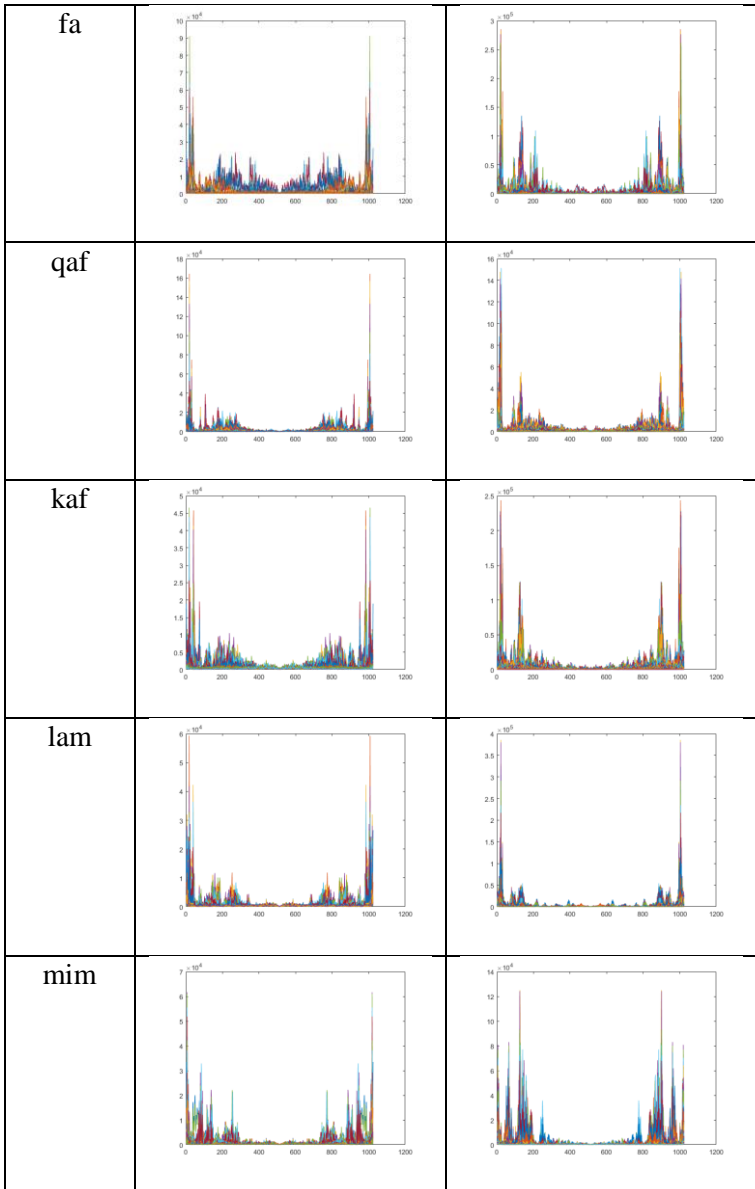
E. Hasil *Magnitude Spectrum*

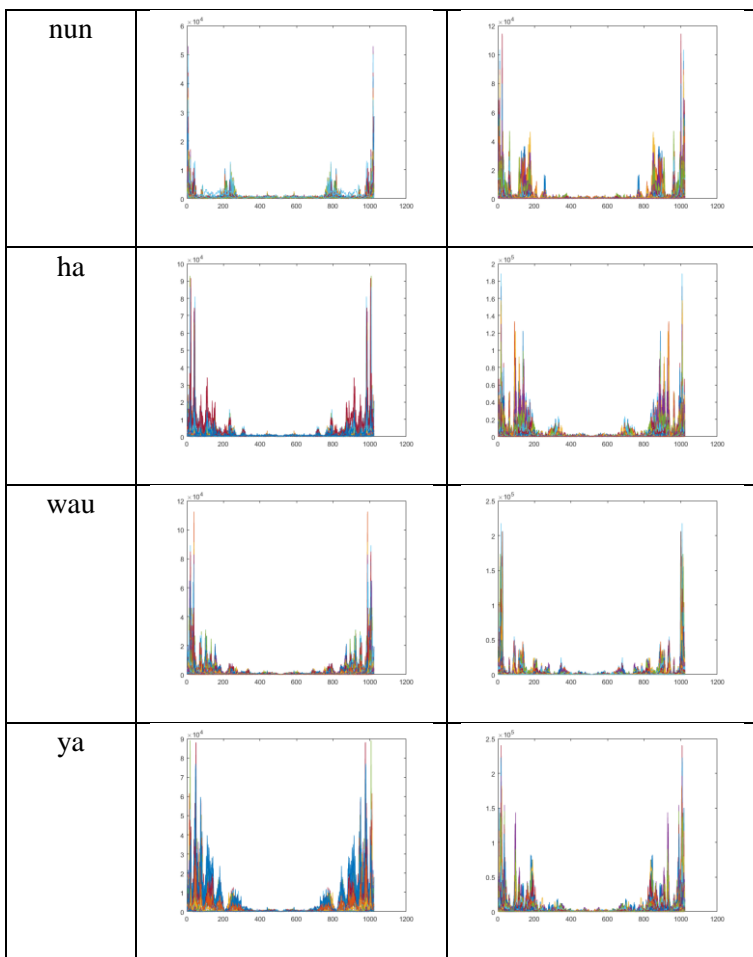




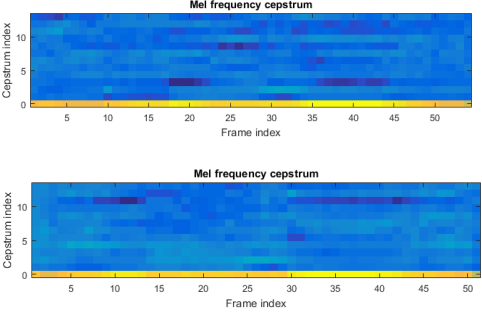
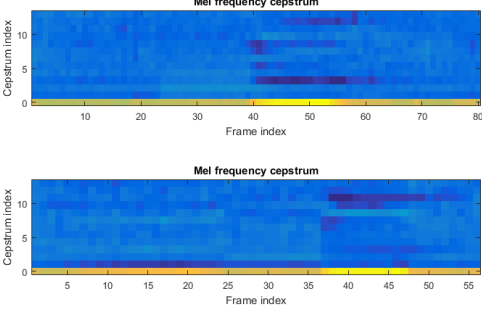
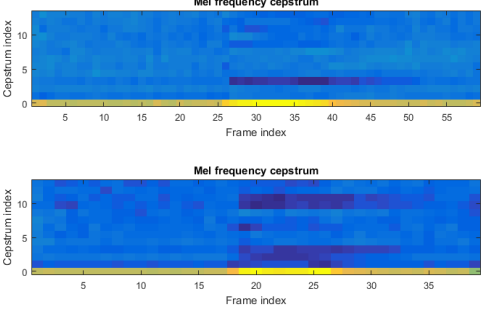


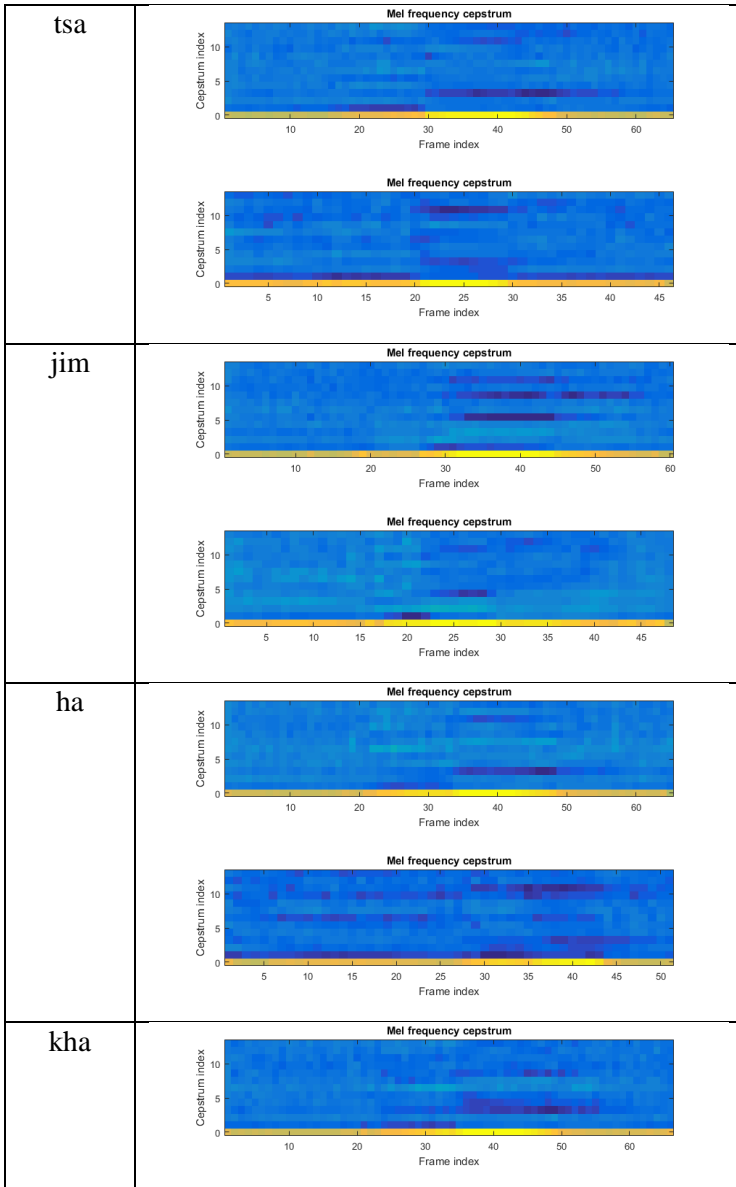


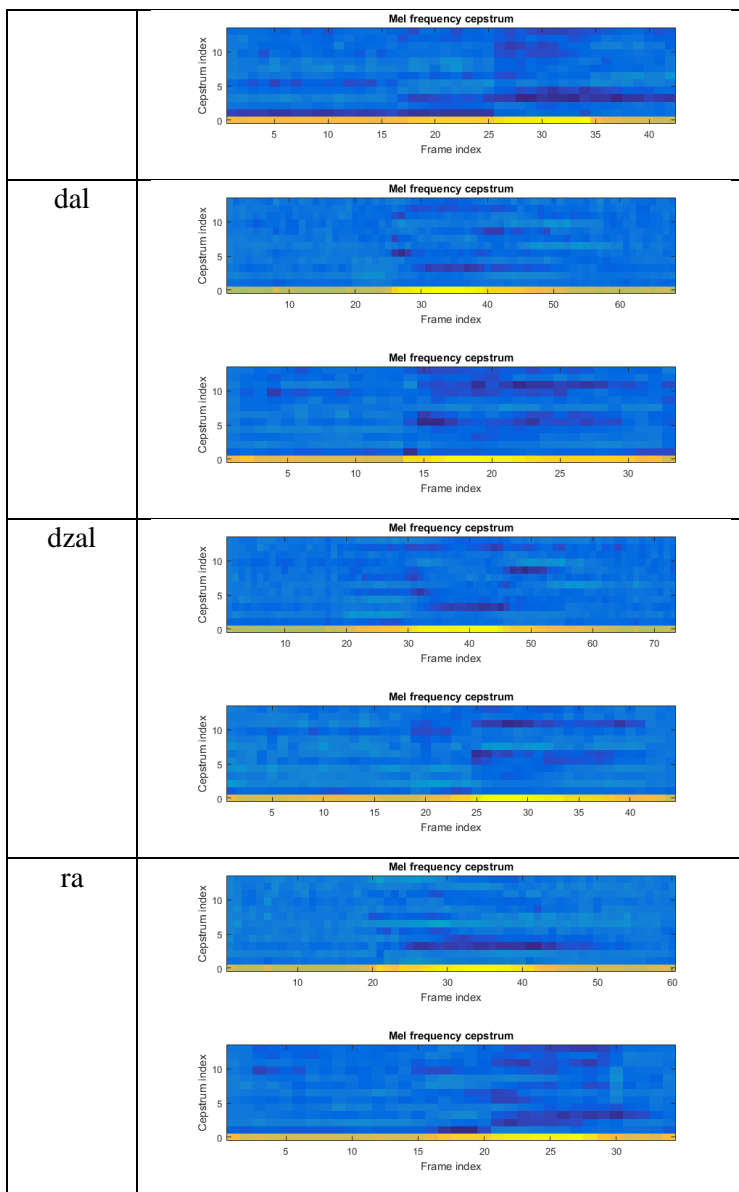


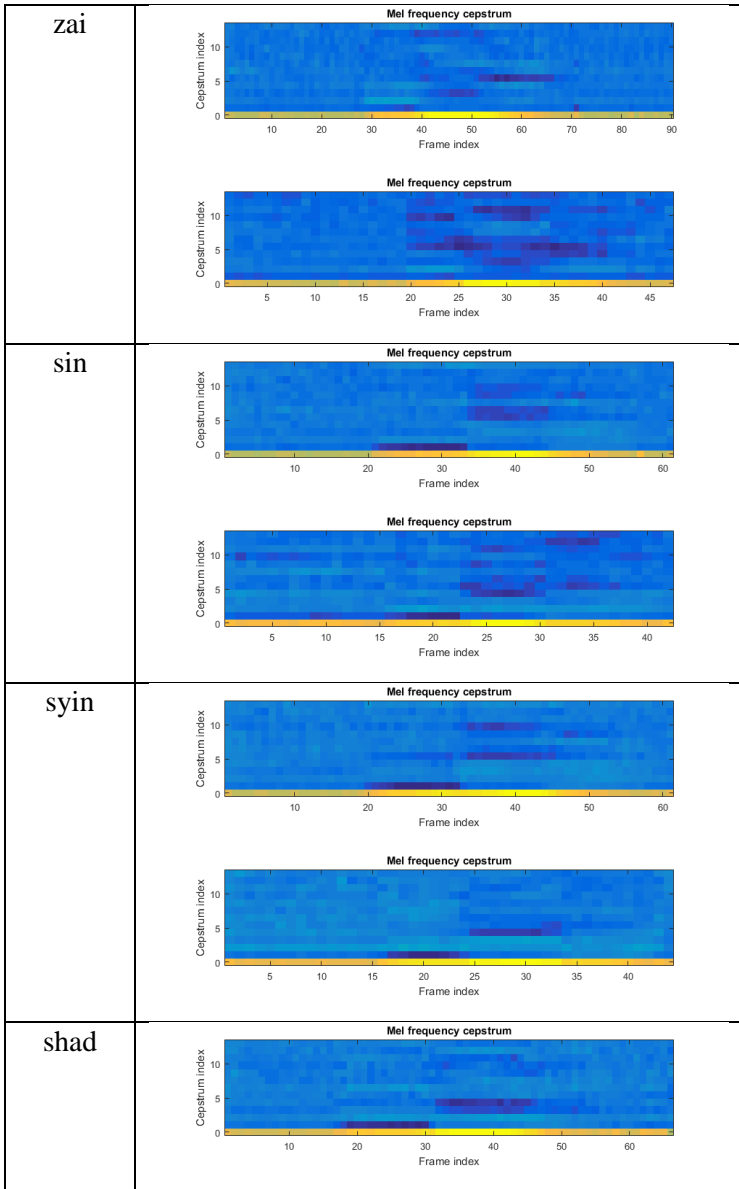


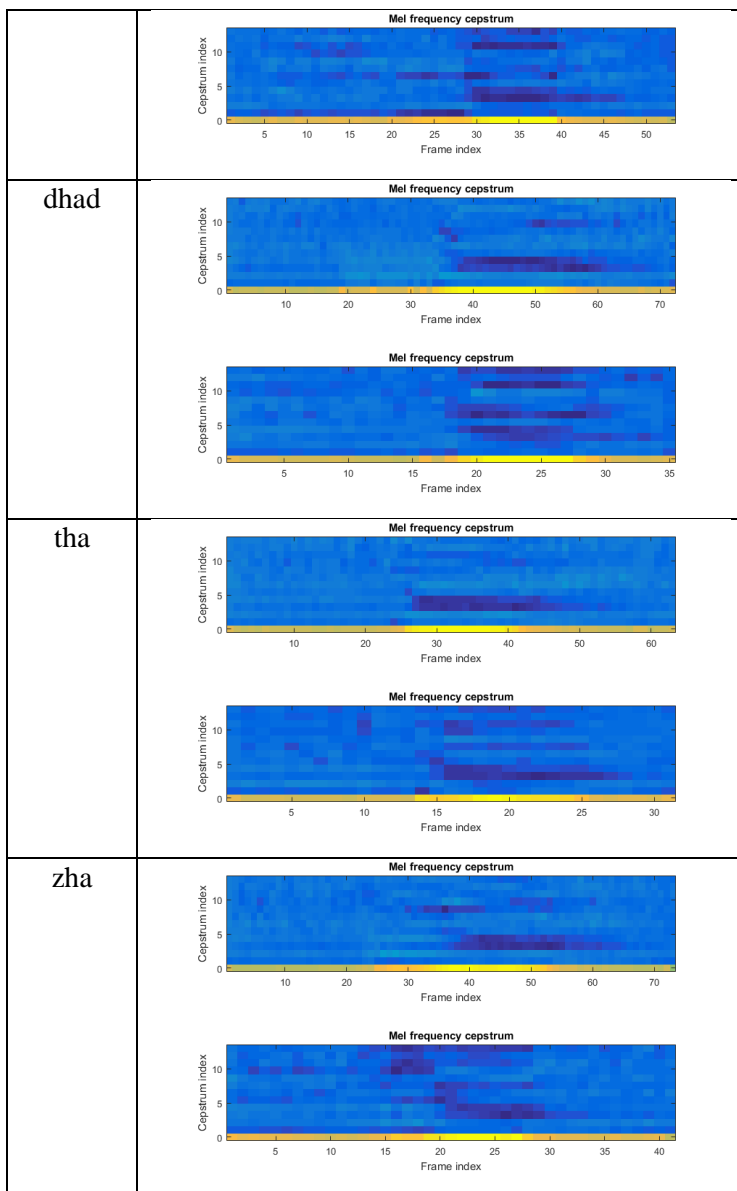
F. Hasil Fitur MFCC

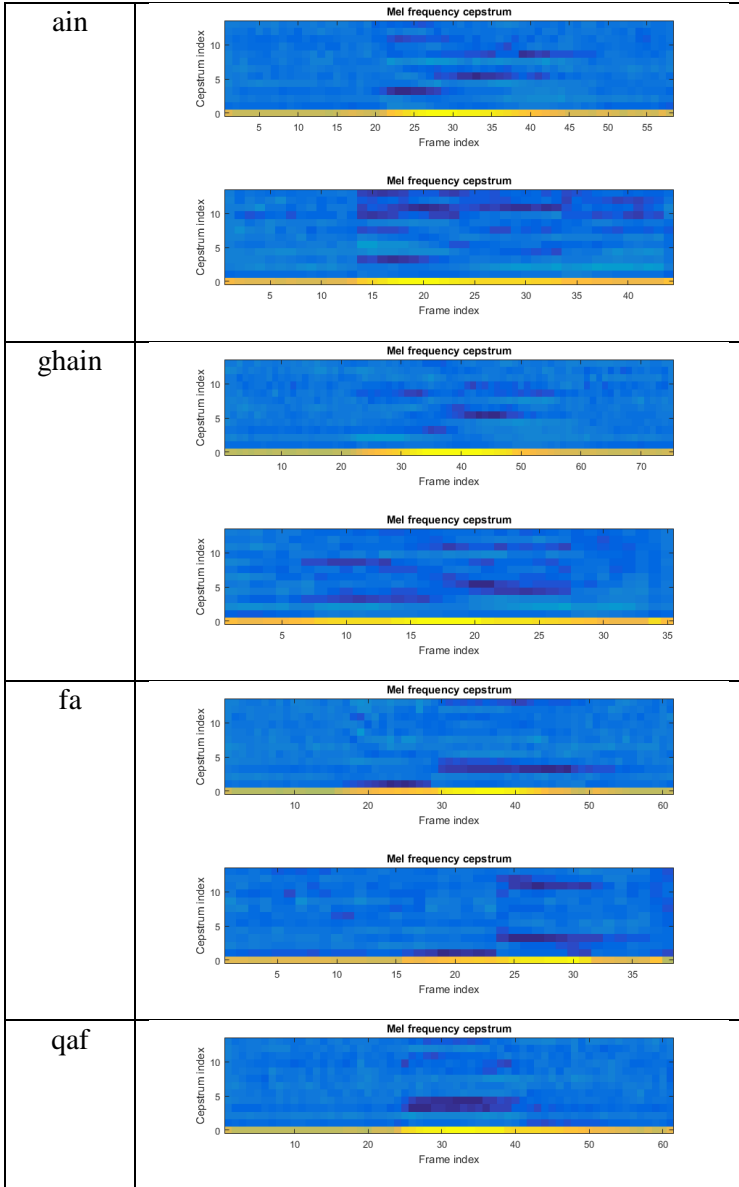
| Huruf | Hasil MFCC untuk Data Laki-Laki (Atas) dan Data Perempuan (Bawah) |
|--------|---|
| hamzah |  <p>The figure displays two Mel frequency cepstrum (MFCC) spectrograms for the letter 'hamzah'. The top plot represents the male data, showing Cepstrum index (0 to 10) on the y-axis and Frame index (0 to 50) on the x-axis. The bottom plot represents the female data, also showing Cepstrum index (0 to 10) on the y-axis and Frame index (0 to 50) on the x-axis. Both plots feature a yellow bar at the bottom, indicating the MFCC coefficients. The spectrograms show varying intensity of cepstral components over time, with the male data showing a more pronounced formant structure compared to the female data.</p> |
| ba |  <p>The figure displays two Mel frequency cepstrum (MFCC) spectrograms for the letter 'ba'. The top plot represents the male data, showing Cepstrum index (0 to 10) on the y-axis and Frame index (0 to 80) on the x-axis. The bottom plot represents the female data, showing Cepstrum index (0 to 10) on the y-axis and Frame index (0 to 55) on the x-axis. Both plots feature a yellow bar at the bottom, indicating the MFCC coefficients. The spectrograms show varying intensity of cepstral components over time, with the male data showing a more pronounced formant structure compared to the female data.</p> |
| ta |  <p>The figure displays two Mel frequency cepstrum (MFCC) spectrograms for the letter 'ta'. The top plot represents the male data, showing Cepstrum index (0 to 10) on the y-axis and Frame index (0 to 55) on the x-axis. The bottom plot represents the female data, showing Cepstrum index (0 to 10) on the y-axis and Frame index (0 to 35) on the x-axis. Both plots feature a yellow bar at the bottom, indicating the MFCC coefficients. The spectrograms show varying intensity of cepstral components over time, with the male data showing a more pronounced formant structure compared to the female data.</p> |

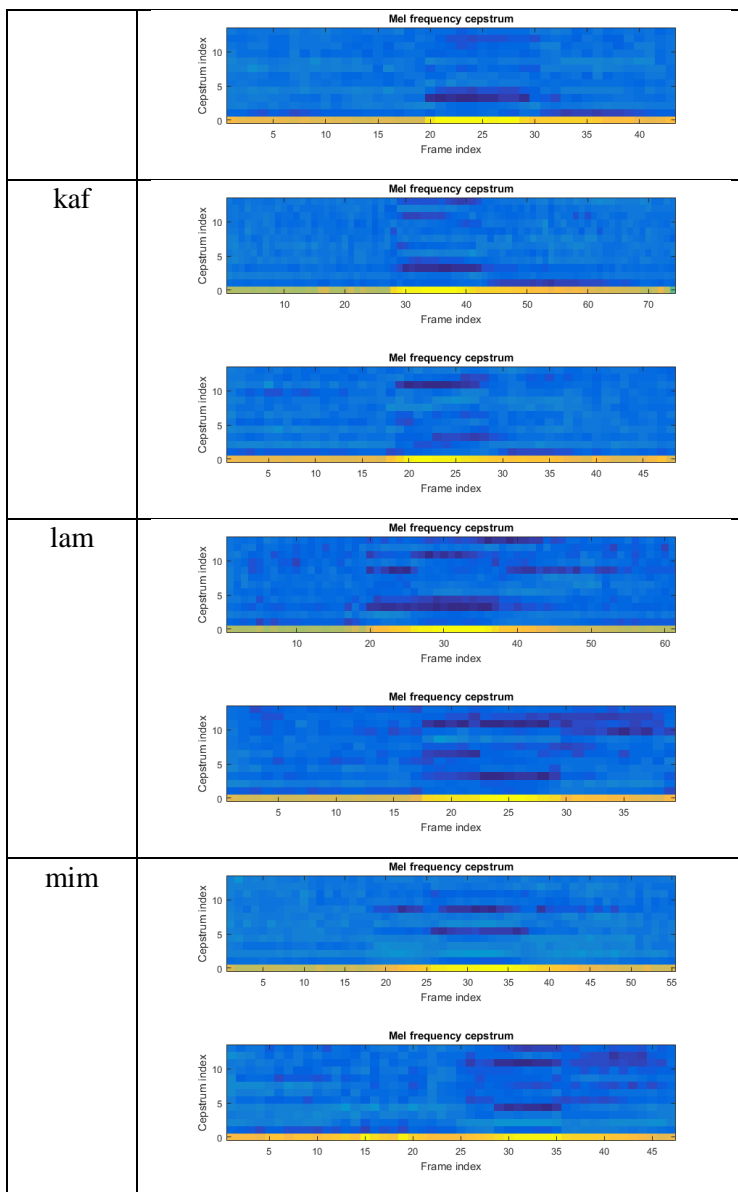


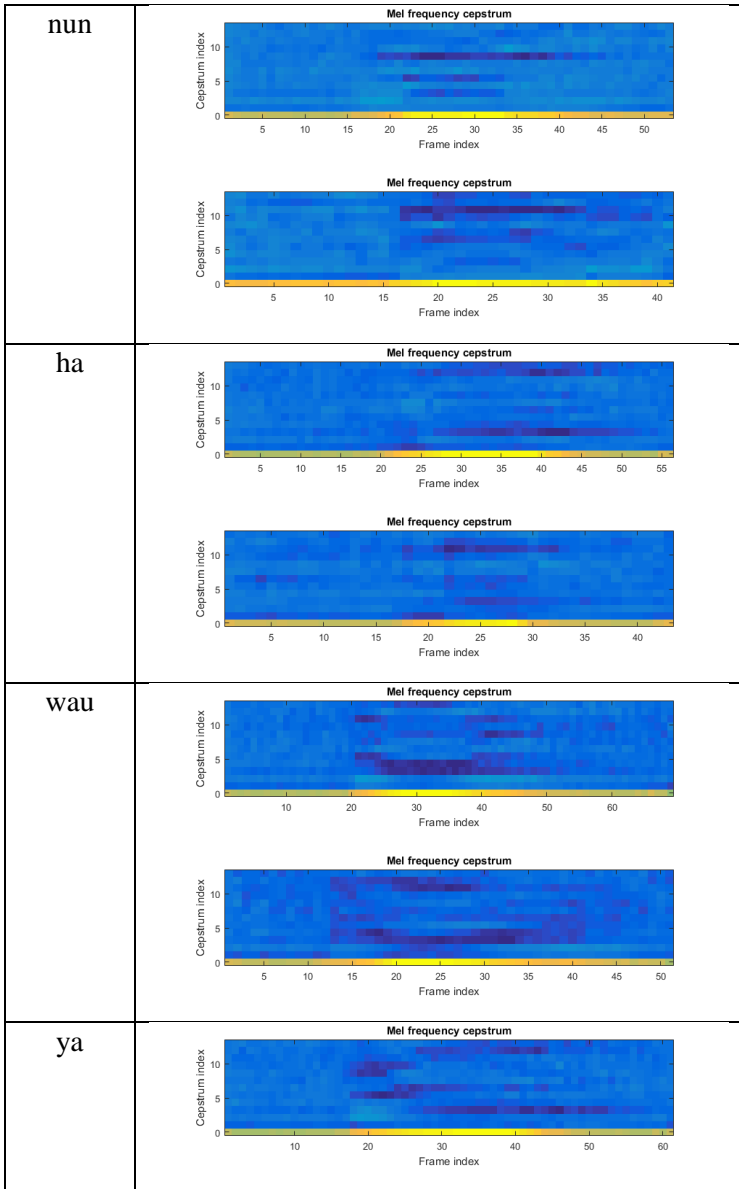


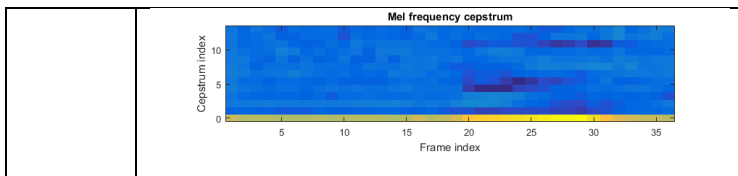












[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Nida Amalia, lahir di Jember, pada tanggal 16 Juli 1994. Penulis menempuh pendidikan mulai dari MIMA KH. Shiddiq Jember (2001-2007), SMPN 1 Jember (2007-2010), SMAN 1 Jember (2010-2013) hingga terakhir Institut Teknologi Sepuluh Nopember Surabaya (2013-2017) di jurusan Teknik Informatika, Fakultas Teknologi Informasi angkatan tahun 2013.

Selain mengikuti kegiatan akademik, penulis mengikuti kegiatan organisasi sekaligus mengembangkan minatnya pada bidang jurnalistik sebagai anggota Departemen Media Informasi HMTTC (2014-2016), dan keilmiahan sebagai staf Kementerian Inovasi Karya BEM ITS (2014-2015). Penulis juga berpartisipasi sebagai anggota aktif UKM Musik ITS (2013-2014) dan klub tari saman TC. Dalam bidang kerohanian, penulis aktif dalam sebagai anggota JMII ITS (2014-2017) dan pengurus harian KMI (2014-2016). Selain itu, penulis juga aktif dalam kegiatan kepanitiaan sebagai staf Acara Pusat ITS Expo 2014, staf Schematics 2014 dan 2015 serta menyelenggarakan kegiatan kepanitiaan FTIf (2014-2015) sebagai pemandu.

Penulis memiliki bidang minat Komputasi Cerdas Visi (KCV) dengan fokus studi pada bidang pengolahan citra digital, *data mining*, komputasi biomedik, dan analisa media sosial. Komunikasi dengan penulis dapat melalui email: **nida.nidaamalia@gmail.com**.

[Halaman ini sengaja dikosongkan]