KONVERSI SUARA DIGITAL DENGAN MENGGUNAKAN *ALGORITMA WAVEFORM SIMILARITY OVERLAP-ADD* (*WSOLA*)

S K R I P S I

Oleh :

RASUNA FAHRULY S.
NIM. 08650049

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015

KONVERSI SUARA DIGITAL DENGAN MENGGUNAKAN *ALGORITMA WAVEFORM SIMILARITY OVERLAP-ADD* (*WSOLA*)

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh

RASUNA FAHRULY S.
NIM. 08650049

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015

HALAMAN PERSETUJUAN

KONVERSI SUARA DIGITAL DENGAN MENGGUNAKAN *ALGORITMA WAVEFORM SIMILARITY OVERLAP-ADD* (*WSOLA*)

SKRIPSI

Oleh

RASUNA FAHRULY S.
NIM: 08650049

Telah Disetujui,

Malang, 10 Maret 2015

Dosen Pembimbing I	Dosen Pembimbing II
(TOTOK CHAMIDY, M.KOM)	**(Dr. M. FAISAL, M.T)**
NIP. 19691222 200604 1 001	NIP. **19740510 200501 1 007**

Mengetahui:

Ketua Jurusan Teknik Informatika

DR. Cahyo Crysdian
NIP.197404242009011008

HALAMAN PENGESAHAN

KONVERSI SUARA DIGITAL DENGAN MENGGUNAKAN *ALGORITMA WAVEFORM SIMILARITY OVERLAP-ADD* (*WSOLA*)

S K R I P S I

Oleh :

Rasuna Fahruly S.
NIM. 08650049

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer Strata Satu (S. Kom)

Tanggal

Susunan Dewan Penguji :			**Tanda Tangan**
1. Penguji Utama	**:**	**Syahiduz Zaman, M.Kom** **NIP. 19700502 200501 1 005**	**( )**
2. Ketua Penguji	**:**	**Dr. M. Amin Hariyadi, M.T** **NIP. 19670118 200501 1 001**	**( )**
3. Sekretaris	**:**	**Totok Chamidy, M.Kom** **NIP. 19691222 200604 1 001**	**( )**
4. Anggota Penguji	**:**	**Dr. Muhammad Faisal, M.T** **NIP. 19740510 200501 1 007**	**( )**

Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

S U R A T P E R N Y A T A A N

Saya yang bertandatangan di bawah ini saya:

Nama	:	Rasuna Fahruly S.
NIM	:	08650049
Fakultas/Jurusan	:	Sains dan Teknologi / Teknik Informatika
Judul Penelitian	:	Konversi Suara Dengan Menggunakan *Algoritma Waveform Similarity Overlap - Add* (*WSOLA*)

Menyatakan dengan sebenar-benarnya bahwa skripsi yang saya buat tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan tanpa paksaan dari siapapun.

Malang, ……. 2015

Yang Membuat Pernyataan,

Rasuna Fahruly S.

NIM: 08650049

MOTTO

بَيْضَةُ اليَوْمِ خَيْرٌ مِنْ دَجَاجَةِ الغَدِ

"Telur hari ini lebih baik daripada ayam esok hari."

HALAMAN PERSEMBAHAN

Segala Puji bagi Allah SWT,
Kupersembahkan sebuah karya teruntuk orang-orang
yang ada di sekelilingku yang sangat kusayangi dan aku banggakan

Karya ini saya Persembahkan kepada :

Ayah (Ahmad Baihaqi), Ibunda tercinta (Sulikah)

Terima Kasih atas Do'a, Dukungan, Bimbingan yang tak pernah berhenti mengiringi di setiap langkahku. Terima kasih atas segala pengorbanan, nasehat dan perjuangan serta limpahan kasih sayang yang telah ayah & ibu curahkan.

Adik - adikku (Gusti Rana & Ghoniyah), dan besarku yang telah memberi dukungan dalam segala hal.

Untuk Mas Rozaq yang selalu member motivasi dan dukungannya dalam menyelesaikan penelitian ini

Sahabat-sahabatku Deeraja 08, semoga Allah SWT senantiasa melimpahkan rahmat dan barokah ilmu yang kelak mengantarkan kita semua untuk menjalani kehidupan mendatang. Amin...

Serta rekan-rekan dan semua pihak yang tidak dapat disebutkan satu persatu, yang telah membantu dari awal kuliah hingga tersusunnya skripsi ini. Terimakasih...

K A T A P E N G A N T A R

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Syukur Alhamdulillah, segala puji dan syukur dengan tulus kami persembahkan ke hadirat Allah SWT, karena hanya dengan rahmat, petunjuk dan hidayah-Nya penulis mampu menyelesaikan tugas akhir yang berjudul *KONVERSI SUARA DIGITAL DENGAN MENGGUNAKAN ALGORITMA WAVEFORM SIMILARITY OVERLAP-ADD (WSOLA).*

Shalawat serta salam penulis haturkan pada junjungan Nabi Muhammad SAW yang memberikan motivasi bagi umat Islam, khususnya bagi penulis untuk selalu berproses menuju insan yang memiliki intelektual tinggi & berakhlak mulia.

Penyelesaian skripsi ini merupakan suatu pekerjaan rumit & panjang yang wajib penulis selesaikan. Namun berkat ma'unnah Allah SWT dan bantuan dari berbagai pihak baik berupa moril maupun materiil , akhirnya tugas akhir ini dapat terselesaikan dengan baik. Oleh karena itu penulis menyampaikan rasa hormat, ungkapan terima kasih kepada:

1. Bapak Totok Chamidy, M.Kom, selaku pembimbing I yang dengan sabar memberikan arahan, saran dan motivasi pada peneliti sehingga skripsi ini dapat terselesaikan dengan baik.
2. Bapak Dr. M. Faisal, MT, selaku Dosen Pembimbing II yang bersedia meluangkan waktu untuk memberikan bimbingan dan arahan dalam menyelesaikan skripsi ini.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang.

4. Bapak Fatchurrohman, M.Kom, selaku Dosen Wali yang bersedia meluangkan waktu untuk memberikan masukan dalam proses penyelesaian skripsi ini.
5. Seluruh Dosen & civitas akademik Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah mengajarkan banyak hal dan selalu memberikan semangat untuk terus berproses hingga akhir perkuliahan peneliti.
6. Kedua orang tuaku, Ayah dan Mama yang senantiasa memberikan doa dan kasih sayang yang tiada henti. Serta adik Gusti, adik Niyah, mas Rozaq yang selalu memotivasi penulis untuk menyelesaikan penelitian ini.
7. Seluruh teman – teman seperjuangan Teknik Informatika, yang selalu memberikan motivasi untuk penyelesaian skripsi ini.
8. Teman – teman Deeraja 08 terima kasih atas semua dukungan yan telah diberikan.

Sebagaimana pepatah "tiada gading yang tak retak", maka skripsi ini pun masih banyak terdapat kekurangan dan kelemahan. Oleh karena itu peneliti mengaharapkan kritik dan saran untuk perbaikan di masa mendatang.

Penulis berharap semoga skripsi ini bisa memberikan manfaat bagi pembaca. Selain itu peneliti berharap semoga skripsi ini dapat memberikan nilai guna baik bagi peneliti maupun bagi pembaca. *Amin Ya Robbal'Alamin*

Malang, 10 Maret 2015

Peneliti,

Rasuna Fahruly S.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

S, Fahruly, Rasuna. 2015, Konversi Suara Digital Dengan Menggunakan Algoritma Waveform Similarity Overlap – Add (WSOLA). Skripsi, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing : Totok Chamidy, M. Kom
Kata Kunci : Konversi, Suara, Waveform Similarity Based Overlap-Add (WSOLA), WAV.

Dalam sebuah kasus persidangan dibutuhkan saksi anonim. Hal ini bertujuan untuk mengolah suara dari saksi supaya tidak dikenali. Meskipun hanya bersaksi dengan suaranya saja. Oleh karena itu, dibutuhkan sebuah teknologi yang mampu melakukan perubahan (konversi) suara saksi tersebut.

Algoritma Waveform Similarity Based Overlap-Add (WSOLA) adalah salah satu algoritma yang memungkinkan untuk melakukan proses konversi suara dengan cara memodifikasi durasi sinyal suara. Dan bekerja dengan cara membagi bentuk gelombang suara pada bagian *overlape*. Untuk merubah frekuensi sinyal, segmen dipindahkan ke segmen yang lebih dekat dan mengubah t*ime scale* segmen diulang beberapa kali untuk dilakukan eleminasi. Algoritma Waveform Similarity Based Overlap-Add (WSOLA) mengolah input buffer dari byte sample file audio. Format file audio yang didukung adalah format .wav. Sebelum dilakukan pengolahan aplikasi akan melakukan pengecekan apakah ada efek konversi suara atau tidak. Jika tidak ada efek konversi suara maka file audio langsung dimainkan oleh aplikasi. Namun jika ada efek konversi suara aplikasi akan mengolah byte sample terlebih dahulu dengan menggunakan Waveform Similarity Based Overlap-Add (WSOLA).

Dari hasil penelitian dan uji coba yang telah dilakukan algoritma Waveform Similarity Based Overlap-Add (WSOLA) mampu melakukan konversi suara dengan tingkat keberhasilan 90% dengan menggunakan parameter 88 hingga parameter 320 dan menggunakan parameter -88 hingga parameters (-200).

ABSTRACT

S, Fahruly, Rasuna. 2015 Digital Voice Conversion Algorithm Using Waveform Similarity Overlap - Add (WSOLA). Thesis, Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Supervisor: Totok Chamidy, M. Kom

Keywords: Conversion, Voice, Waveform Similarity Based Overlap-Add (WSOLA) , WAV.

In a court case sometimes required an anonymous witness. Anonymous witnesses testified only by his voice alone, without showing identity and face. It aims to prevent witnesses from harm on his testimony. Although only testified with his voice alone, anonymous witnesses can still be identified. Therefore, it takes a technology that can make changes (conversion) voice of the witness.

Algorithm Based Waveform Similarity Overlap-Add (WSOLA) is one of an algorithm that allows to perform voice conversion process by modifying the duration of the sound signal. And works by dividing the form of sound waves at the *overlape.* To change the signal frequency, the segment moved closer to a segment and change the t *ime scale* segment repeated several times to do elimination.

Algorithm Based Waveform Similarity Overlap-Add (WSOLA) processing the input buffer of bytes sample audio file. Audio file formats supported are .wav format. Prior to processing the application will check whether there is sound or not convertible securities. If there is no effect of the conversion of audio files direct sound played by the application. But if there is a conversion effect sound byte application will process the sample in advance using the Waveform Similarity Based Overlap-Add (WSOLA) .

From the results of research and trials that have been conducted algorithm Based Waveform Similarity Overlap-Add (WSOLA) is able to perform voice conversion process with a success rate of 90% by using parameters 88 to parameters 320 and use the parameter -88 to parameter (-200).

الملخص

فحرولي راسونا. 2015. تغير الصوت الإصبعي باستخدام ألغورتما وفروم سيميلاريتي أوفورلاب-أدد (WSOLA). بحث جامعي، كلية العلوم الطبيعية والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: توتوك حميدي الماجستير.

الكلمات الأساسية: التحويل المكاني، الصوت، وفروم سيميلاريتي أوفورلاب-أدد (WSOLA)، (WAV).

في بعض الأحيان، تحتاج مشكلات المحكمة إلى الشاهد المغفل. لا يشهد الشاهد المغفل إلاّ بالصوت، ولا يسلم البطاقة الهوية ووجهه. يهدف هذا الحال لصرف الشاهد عن الخطيرات التي يشهده. رغم أن يشهد بالصوت فقط، لكن الشاهد المغفل لا يزال بمعروف ذاتيته. لذلك يحتاج إلى التكنولوجي لتغير صوت الشاهد.

ألغورتما وفروم سيميلاريتي أوفورلاب-أدد (WSOLA) هو أحد ألغورتما الممكن لعملية تغير الصوت بتحول مدة الصوت وتقسيم أمواج الصوت في جزء أوفورلاب. لتغير تواتر الإيماء، تحول بعض القطعة إلى قطعة أخرى الأقرب وتغير معيار القطعة وتكرره مرات لإوالته.

ألغورتما وفروم سيميلاريتي أوفورلاب-أدد (WSOLA) يغير القوة الداخلي من القوة السمعي. وصيغة الملف السمعي الموافق هو صيغة واف (.WAV). قبل أن يغير في التطبيق، أن يفتش وجود تأثير تغير الصوت. وجود تأثير تغير الصوت يدل على التغير باستجدام ألغورتما وفروم سيميلاريتي أوفورلاب-أدد (WSOLA) في أوله، والعكس، غير وجود تأثير تغير الصوت يدل على أن الملف السمعي يطبق بالتطبيق.

من نتيجة البحث والتجربة التي حصلت الباحثة، أن استخدامألغورتما وفروم سيميلاريتي أوفورلاب-أدد (WSOLA) فعال في تغير الصوت بالنسبة 90%، مااستطاع التطبيق بتغير الصوت إلى الملف السمعي على استعجال 320 أو أكثر.

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Dalam sebuah kasus persidangan terkadang diperlukan seorang saksi anonym, yakni saksi yang tidak boleh diketahui identitasnya tetapi hanya suaranya. Dalam sistem peradilan di negara Indonesia, saksi seperti itu diperbolehkan jika memang kesaksianya akan mengancam dan menimbulkan bahaya bagi yang bersaksi. Di dalam ajaran agama islam kita juga tidak diperbolehkan untuk membahayakan orang lain.

عَنْ أَبِي سَعِيْدٍ سَعَدْ بْنِ سِنَانِ الْخُدْرِي رَضِيَ اللهُ عَنْهُ أَنَّ رَسُوْلَ اللهِ صَلَّى الله عليه وسلَّمَ قَالَ : لاَ ضَرَرَ وَلاَ ضِرَارَ [حَدِيْثٌ حَسَنٌ رَوَاهُ ابْنُ مَاجَه وَالدَّارُقُطْنِي وَغَيْرُهُمَا مُسْنَداً، وَرَوَاهُ مَالِك فِي الْمُوَطَّأ مُرْسَلاً عَنْ عَمْرو بْنِ يَحْيَى عَنْ أَبِيْهِ عَنِ النَّبِيِّ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ فَأَسْقَطَ أَبَا سَعِيْدٍ وَلَهُ طُرُقٌ يُقَوِّي بَعْضُهَا بَعْضاً]

Dari Sa'id Sa'd bin Malik bin Sinan Al-Khudri rodhiallohu 'anhu, bahwa Rosulullah sholallahu 'alaihi wa sallam beliau bersabda, "Tidak boleh membahayakan orang lain, dan tidak boleh membalas bahaya orang lain melebihi bahaya yang diberikannya." (Hadits hasan diriwayatkan oleh Ibnu Majah, Daruquthni, dan yang lainnya dengan disanadkan dan diriwayatkan oleh Malik dalam Al-Muwatha' secara mursal, dari Amr bin Yahya, dari bapaknya, dari Nabi sholallahu 'alaihi wa sallam dengan meniadakan Abu Sa'id. Hadits ini menguatkan satu dengan yang lainnya).

Imam Nawawi menjelaskan pada kata لاَ ضِرَارَ (tidak boleh membahayakan), yakni tidak boleh salah seoang dari kalian membahayakan yang lainnya dengan tampa hak dan tidak boleh pula memulai kejahatan kepadanya, sedangkan kata وَلاَ ضِرَارَ yaitu jangan membalas bahaya siapa yang membahayakanmu, jika seseorang mencaci kamu janganlah kamu mencacinya, jika memukulmu, janganlah kamu memukulnya. Tetapi tuntutlah hakmu darinya kepada Hakim dengan tampa membalas terlebih dahulu.

Saksi anonym bersaksi hanya dengan suaranya saja. Dengan maksud supaya tidak dikenali, akan tetapi, suara seseorang bersifat unik, seseorang masih bisa dikenali dari suaranya. Oleh karena itu, harus ada sebuah alat yang memungkinkan untuk merubah suara saksi tersebut agar benar-benar tidak dikenali. Teknologi multimedia telah memungkinkan proses pengkonversian suara. Sehingga memungkinkan untuk mengubah pola suara digital menjadi pola suara lain dengan karakteristik yang berbeda dan memberikan identitas baru, dengan tetap menjaga konten aslinya. Prosesnya dengan cara mengubah frekuensi file (tekanan/nada) audio tersebut tanpa merubah konten suara file audio.

Untuk melakukan konversi suara digital dibutuhkan metode tertentu untuk dapat mengimplementasikanya. Sudah banyak metode yang dilakukan oleh penelitian-penelitian lain untuk mengembangkan teknologi ini, salah satunya adalah algoritma Fast Fourier Transform (FFT). Penelitian yang dilakukan oleh Stephen M Bernsee menunjukkan hasil yang cukup maksimal ketika aplikasi pitchshifting dengan menggunakan algoritma Fast Fourier Transform dijalankan pada computer dengan spesifikasi yang tinggi akan tetapi sebaliknya. Penelitian lain dengan menggunakan Algoritma constant-q transform dilakukan oleh

Christian Schörkhube dan Anssi Klapuri,dari University of Music and Performing Arts. Hasil penelitian ini tidak dilakukan uji coba secara objektif, akan tetap algoritma constant-q transform mampu melakukan pitch shifting pada file audio monophonic music dan dense polyphonic.

Metode yang akan dibahas pada tugas akhir adalah algoritma pitch shifting dengan Waveform Similarity Based Overlap-Add (WSOLA) dan akan dilakukan pengujian dan analisis efek dari pengimplementasian WSOLA pada sistem konversi suara.

1.2 RUMUSAN MASALAH

Berdasarkan penjelasan pada latar belakang, maka perumusan masalah dalam penelitian ini yaitu Bagaimana cara merancang dan membangun sebuah aplikasi yang mengimplementasikan algoritma Waveform Similarity Based Overlap-Add (WSOLA) untuk melakukan proses konversi suara digital?

1.3 TUJUAN PENELITIAN

Berdasarkan rumusan masalah, maka tujuan dalam penelitian ini adalah merancang dan membangun sebuah aplikasi yang mengimplementasikan algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*)* untuk melakukan proses konversi suara digital.

1.4 MANFAAT PENELITIAN

Manfaat yang dapat diambil dari penelitian ini adalah

1. Aplikasi yang dihasilkan bisa digunakan untuk proses penyamaran saksi anonym dalam kasus persidangan (saksi yang tidak ingin diketahui identitasnya).
2. Sebagai referensi tingkat keberhasilan algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*)* untuk melakukan proses konversi nada.

1.5 BATASAN PENELITIAN

Batasan masalah pada penelitian ini adalah :

1. Suara yang dikonversi adalah suara digital dengan format .WAV.
2. Konversi suara dilakukan secara realtime, tidak merubah hasil konversi suara ke dalam bentuk file atau format lain.
3. Aplikasi dibangun menggunakan bahasa pemrogaman java dan menggunaka teknologi java sound yang telah di*bundled* pada Java SE Development Kid (JDK).

1.6 SISTEMATIKA PENULISAN

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pendahuluan, membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penyusunan tugas akhir, metodelogi, dan sistematika penyusunan tugas akhir.

BAB II LANDASAN TEORI

Landasan teori berisikan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah dasar teori yang berkaitan dengan teori suara, pengolahan suara digital, format file audio, konversi suara dan algoritma*Waveform Similarity Based Overlap-Add (*WSOLA*)*.

BAB III METODE PENELITIAN

Bab III menjelaskan metode penelitian yang digunakan. Meliputi penentuan variable, sumber data, analisis kebutuhan fungsional serta desain dan analisis perancangan system, meliputi perancangan desain system, rancangan alur sitem dan rancangan penerapan algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*)* di aplikasi.

BAB IV HASIL DAN PEMBAHASAN

Membahas hasil penerapan perancangan sistem, uji coba sistem serta analisis hasil uji coba sistem. Khususnya analisis keberhasilan algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*)* untuk melakukan konversi nada.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari hasil uji coba dan analisis pada tahap sebelumnya, serta saran yang harus dilakukan untuk perbaikan penelitian untuk topik konversi suara digital.

1.7 METODE PENELITIAN

Untuk mencapai tujuan yang telah dirumuskan sebelumnya, maka metodologi pengumpulan data yang dilakukan dalam penulisan skripsi ini adalah *library research* yaitu suatu cara penelitian dan pengumpulan data teoritis dari buku-buku, artikel, jurnal dan berbagai literatur yang mendukung penyusunan skripsi. Pengumpulan data juga dilakukan dengan melakukan uji coba aplikasi pada pengguna yang disertai dengan angket.

BAB II

STUDI LITERATUR

2.1 PENELITIAN TERKAIT

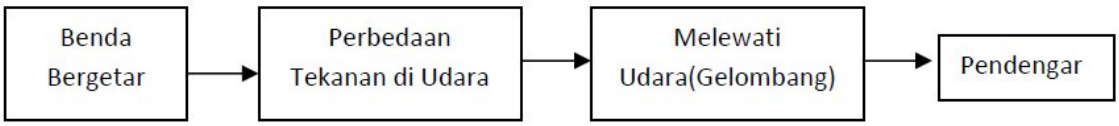
Penelitian Terdahulu dilakukan oleh Adriyakti Subiyakto dengan judul *Speech Synthesizer Berbasis Diphone Menggunakan Algoritma Waveform Similarity Overlap-Add (WSOLA)*. Pada penelitian ini algoritma WSOLA digunakan untuk menggabungkan unit ucapan diphone, sehingga perangkaian antar diphone yang mengandung transisi antar dua bunyi yang berdekatan (*adjacent phones*), menjadi halus tanpa bunyi yang bersifat eksplosif. Penelitian ini berkesimpulan dengan menggunakan *diphone concatenation* dan penerapan algoritma WSOLA maka sintesis ucapan yang dihasilkan ternyata dapat dimengerti dengan jelas, lancar dalam pengucapan dan datar tanpa intonasi dan kemampuan system dalam mensintesis suara ucapan manusia termasuk dalam kategori cukup.

Penelitian lainya dilakukan oleh Werner Verhelst and Marc Roelands dari Vrije University Brussel Faculty of Applied Science, dept. ETRO/DSSP Pleinlaan 2, B-1050 Brussels Belgia yang berjudul *An Overlap-Add Technique Based On Waveform Similarity (WSOLA) For High Quality Time-Scale Modification Of Speech* dalam penelitianya memberikan kesimpulan bahwa algoritma WSOLA berhasil menghasilkan output suara dengan kualitas tinggi secara algoritma dan komputasi yang efisien yang berkerja dalam konteks STFT.

Penelitian selanjutnya dilakukan oleh Ibnu Daqiqil, Aryanto, Uchi Dayana Ibnu Daqiqil, Aryanto, Uchi Dayana Fakultas Ilmu Komputer Universitas Muhammadiyah Riau dalam penelitian ini melakukan pengenalan gelombang suara menjadi sebuah transkip *chord*, dimana nadanya berasal dari masukan audio file berformat wave. Chord yang dikenali dalam aplikasi ini yaitu chord mayor dan chord minor. Proses kerja aplikasi diawali dengan proses sampling audio file masukan. File dalam format wave yang merupakan sinyal dalam domain waktu dibagi ke dalam frame-frame (frame blocking). Dari frame- frame tersebut kemudian ditransformasikan dengan Fast Fourier Transform (FFT) menjadi sinyal dalam domain frekuensi. Hasil FFT dipetakan menurut nada yang bersesuaian menjadi PCP yang kemudian diskala dalam rentang 0–1. Tiap data hasil dari ransformasi sinyal dalam domain frekuensi dipetakan ke satu nada (yang memiliki frekuensi terdekat) dari 12 nada pada PCP.

2.2 SUARA

Suara adalah fenomena fisik yang dihasilkan oleh getaran benda atau getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu (Binanto, 2010). Suara berhubungan erat dengan rasa "mendengar". Suara atau bunyi biasanya merambat melalui udara. Suara atau bunyi tidak bias merambat melalui ruang hampa.



Gambar 2.1 Proses Terjadinya Suara

(Sumber dari https://pti08.wordpress.com/tag/audio diakses pada tanggal 26 Agustus 2014 23.36)

Suara atau *sound* diproduksi oleh sebuah obyek yang bergetar, contohnya *loudspeaker, musical instrument,* ataupun pita suara manusia. Getaran mekanik dari sebuah *loudspeaker* membuat pergerakan udara terdorong dan tertarik dari kondisi stabil, adanya gerakan mendorong dan menarik yang terus menerus dari sebuah speaker membuat tekanan udara berubah yang pada akhirnya menyebabkan terjadinya sebuah gelombang suara. Sebuah gelombang suara dapat dideskripsikan oleh frekuensi dan amplitudo. Frekuensi 1 Hz berarti 1 *cycle* gelombang lengkap setiap satu detik. Satuan sebuah frekuensi adalah Hertz (Hz).

Frekuensi *audible* (human *hearing rang)* adalah 20 Hz sampai 20000 Hz. Dalam kenyataan praktis sebuah sumber suara selalu diproduksi pada banyak frekuensi secara simultan. Amplitudo sebuah gelombang mengacu pada besarnya perubahan tekanan dan tingkat kerasnya (*loudness*) gelombang suara. Sebuah sinyal suara diproduksi dan ditransmisikan melalui udara, akhirnya diterima pada telinga manusia. Telinga manusia memiliki gendang pendengaran (*eardrum*) yang dapat bergetar pada saat menerima gerakan gelombang udara (*push and pull).*

Suara dihasilkan oleh getaran suatu benda. Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya. Pola osilasi yang terjadi dinamakan sebagai "Gelombang". Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai **"**Periode**"**. Contoh suara periodik: instrument musik, nyanyian burung, dll. Sedangkan contoh suara nonperiodik: batuk, percikanombak, dll. Suara berkaitan erat dengan.

a. Frekuensi

Frekuensi adalah banyaknya getaran dalam 1 detik yang dihitung dengan menggunakan Satuan : *Hertz (Hz)* atau *cycles per second (cps)*.

Panjang gelombang suara (*wavelength*) dirumuskan = c/f

Dimana c = kecepatan rambat bunyi

Dimana f = frekuensi

Panjang gelombang suara bias dihitung juga dengan rumus : λ= c/f.

Berdasarkan frekuensinya, suara dibagi menjadi 4, yaitu :

1. Infrasound = 0Hz – 20 Hz
2. Pendengaran manusia = 20Hz – 20 KHz
3. Ultrasound = 20KHz – 1 GHz
4. Hypersound = 1GHz – 10 THz

Manusia membuat suara dengan frekuensi: 50Hz – 10KHz. Sinyal suara musik memiliki frekuensi: 20Hz – 20Khz. Maka Sistem multimedia menggunakan suara yang berada dalam range pendengaran manusia. Suara yang berada pada range pendengaran manusia sebagai "*Audio*" dan gelombangnya sebagai "Accoustic Sinyal". Suara diluar range pendengaran manusia dapat dikatakan sebagai "*Noise*" (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat di dengar manusia).

Frekuensi adalah jumlah getaran yang terjadi dalam waktu satu detik atau banyaknya gelombang/getaran listrik yang dihasilkan tiap detik. Frekuensi dilambangkan dalam huruf f. Periode adalah selang waktu yang diperlukan

untuk melakukan satu getaran sempurna. Periode dilambangkan dengan huruf T. Hubungan antara frekuensi dan periode adalah berbanding terbalik,

b. Amplitudo

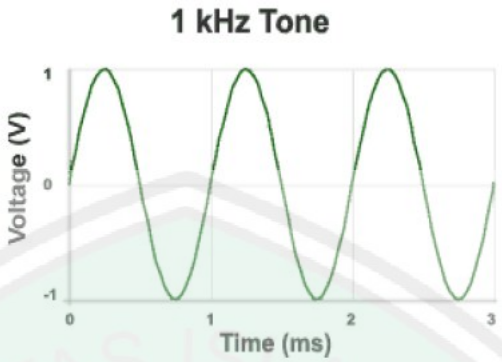
Amplitudo adalah Keras lemahnya bunyi atau tinggi rendahnya gelombang yang dihitung dengan satuan amplitudo adalah decibel (db. Bunyi mulai dapat merusak telinga jika tingkat volumenya lebih besar dari 85 dB dan pada ukuran 130 dB akan mampu membuat hancur gendang telinga.

c. Velocity

Velocity adalah Kecepatan perambatan gelombang bunyi sampai ke telinga pendengar yang dihitung dengan menggunaka Satuan m/s. Pada udara kering dengan suhu 20 °C (68 °F)m kecepatan rambat suara sekitar 343 m/s.

Gelombang suara bervariasi sebagaimana variasi tekanan media perantara seperti udara. Suara diciptakan oleh getaran dari suatu obyek, yang menyebabkan udara disekitarnya bergetar. Getaran udara ini kemudian menyebabkan kendang telinga manusia bergetar, yang kemudian oleh otak di interpretasikan sebagai suara. Diilustrasikan pada gambar speaker menciptakan gelombang suara.

Gelombang suara berjalan melalui udara kebanyakan dengan cara yang sama seperti perjalanan gelombang air melalui air. Dalam kenyataannya, karena gelombang air mudah untuk dilihat dan dipahami, ini sering digunakan sebagai analogi untuk mengilustrasikan bagaimana perambatan gelombang suara.



Gambar 2.2 Gelombang Suara

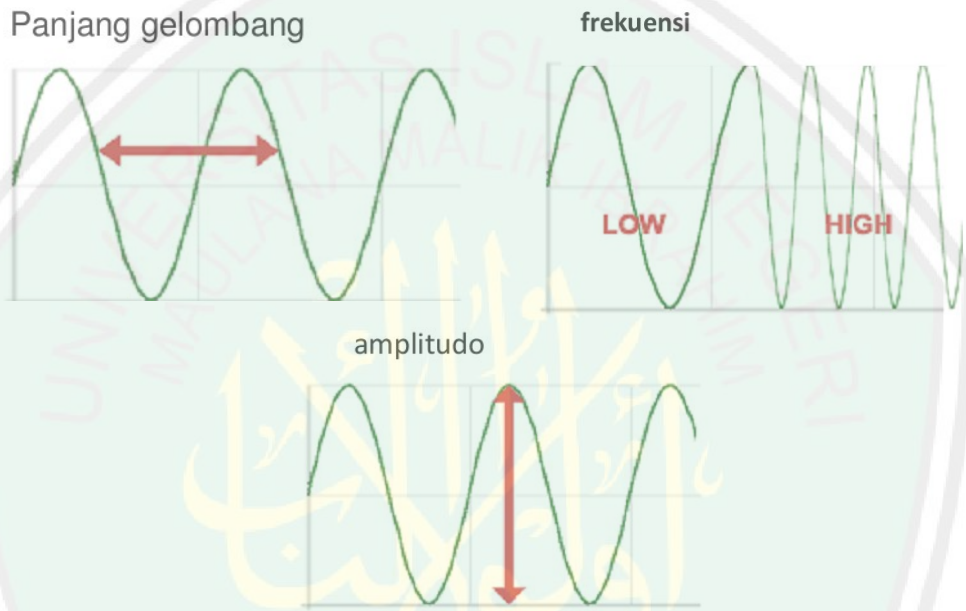
(Sumber dari http://elektronika-dasar.web.id/teori-elektronika/sinyal-audio-gelombang-suara diakses pada tanggal 26 Agustus 2014 23:45)

Gelombang suara dapat juga ditunjukkan dalam suatu grafik standar x versus y seperti ditunjukkan gambar 1-5. Ini memungkinkan untuk memvisualisasi gelombang dengan sudut pandang matematis, menghasilkan kurva yang dikenal sebagai bentuk gelombang. Gelombang ditunjukkan pada nada konstan frekuensi tertentu. Nois dapat didengar dan digunakan sebagai uji atau identifikasi sinyal. Tes nada dibuat dalam bentuk gelombang yang baik ideal untuk tujuan teknis.

Bentuk grafis gelombang dua dimensi namun gelombang sebenarnya dalam bentuk tiga dimensi. Grafik menunjukkan perjalanan gelombang sepanjang jalur dari kiri ke kanan, namun perjalanan gelombang sebenarnya mengembang berlapis dari sumber. Oleh karena itu model kerja dua dimensi menjelaskan dengan baik bila berpikir tentang bagaimana suara berjalan dari satu tempat ke tempat lain.

Semua gelombang pasti memiliki tiga sifat penting untuk kerja audio yaitu panjang gelombang, amplitudo dan frekuensi. Panjang gelombang adalah jarak antar titik gelombang dan titik ekuivalen pada fase berikutnya. Amplitudo adalah

Kekuatan atau daya gelombang sinyal. Tinggi gelombang yang bisa dilihat sebagai grafik. Frekuensi adalah Jumlah getaran yang terjadi dalam waktu satu detik. Diukur dalam hertz atau siklus perdetik. Getaran gelombang suara semakin cepat, frekuensi semakin tinggi.



Gambar 2.3 Panjang Gelombang, Frekuensi, Amplitudo
(Sumber dari http://www.academia.edu/5873076/TEKNIK_AUDIO_VIDEO diakses pada tanggal 27 Agustus 2014 06:00)

Frekuensi, Periode, Fase

Frekuensi adalah jumlah getaran yang terjadi dalam waktu satu detik atau banyaknya gelombang/getaran listrik yang dihasilkan tiap detik. Frekuensi dilambangkan dalam huruf f. Periode adalah selang waktu yang diperlukan untuk melakukan satu getaran sempurna. Periode dilambangkan dengan huruf T. Hubungan antara frekuensi dan periode adalah berbanding terbalik, berarti semakin besar frekuensinya periodenya akan semakin kecil. Secara matematis dapat dituliskan :

$f = 1/T \longrightarrow T = 1/f$

Dengan pengertian:

f : frekuensi, dalam siklus per detik atau Herz

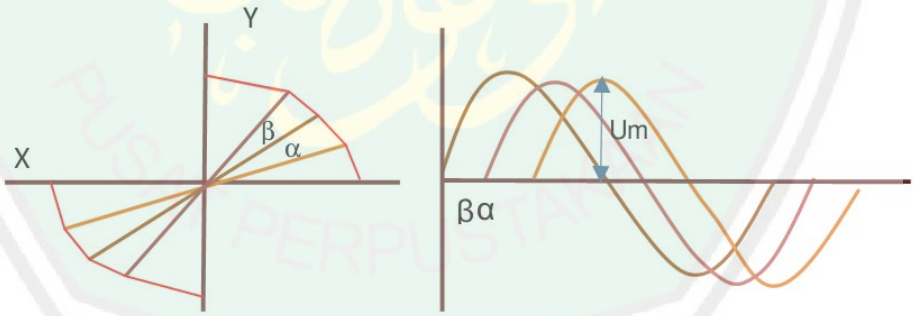
T : periode, dalam detik.

Jika kecepatan perputaran sudut dinyatakan dengan ω, maka frekuensinya sama dengan kecepatan sudut dibagi dengan besarnya sudut satu putaran penuh (2π) atau dapat ditulis :

$f = \omega/2\pi \longrightarrow \omega = 2\pi f$

Dengan pengertian ω adalah kecepatan sudut dalam Radial/detik.

Fase dari listrik arus bolak-balik artinya pergeseran periode waktu arus bolak-balik dari posisi baris nol. Gambar 1 – 10 menggambarkan tiga kumparan yang serupa ditempatkan sejauh sudut α dan β, bergerak pada medan yang sama dengan kecepatan sudut yang sama pula.



Gambar 2.4 Tiga kumparan ditempatkan sejauh sudut α dan β
(Sumber dari http://www.academia.edu/5873076/TEKNIK_AUDIO_VIDEO diakses pada tanggal 27 Agustus 2014 06:00)

Pada gambar terlihat bahwa besarnya tegangan induksi dari ketiga kumparan sama, tetapi harga nol dan maksimumnya tidak bersamaan dalam mencapainya. Hal tersebut berarti beda fase antara 1 dan 2 adalah βo; beda fase antara 2 dan 3 α o, dan beda fase antara 1 dan 3 adalah (α + β)o. Jika besarnya tegangan sesaat:

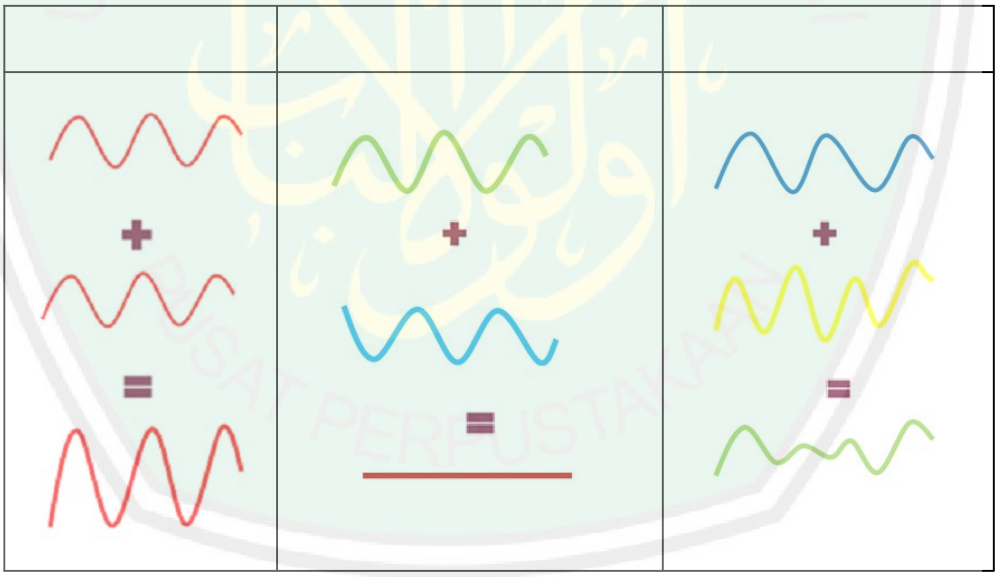
u1 = Umsin ωt, maka

u2 = Umsin (ωt – β), dan

u3 = Um sin {ωt – (α + β)}

Beda fase dalam rangkaian listrik dikenal istilah "lag" atau "lead". Lag artinya harga maksimum atau nol yang dicapai satu siklus lebih lambat atau ketinggalan dari siklus lainnya. Sedangkan lead artinya harga maksimum atau nol yamg dicapai satu siklus mendahului siklus lainnya. Tabel 1-1 berikut mengilustrasikan bagaimana gelombang suara (gelombang lain) bercampur satu sama lain, hasilnya tergantung pada hubungan fase nya.

Tabel 1.1 Ilustrasi gelombang suara bercampur satu sama lain



(Sumber dari http://www.academia.edu/5873076/TEKNIK_AUDIO_VIDEO diakses pada tanggal 27 Agustus 2014 06:00)

Penjumlahan sefase Penjumlahan beda fase 180° Perbedaan Gelombang

- Gelombang suara dalam fase yang sama dijumlahkan menghasilkan gelombang yang lebih kuat.

- Gelombang suara dengan fase berlawanan, tertinggal 180° masing-masing dijumlahkan menghasilkan nol. Ini banyak dijumpai pada kerja piranti penundaan nois.
- Gelombang suara yang mempunyai hubungan fase bervariasi menghasilkan pengaruh suara yang berbeda.

2.3 SUARA DIGITAL

Suara yang kita dengar sehari-hari adalah gelombang analog. Gelombang ini berasal dari tekanan udara yang ada disekeliling kita dengan bantuan gendang telinga. Gendang telinga ini bergetar dan getaran ini dikirim dan diterjemahkan menjadi informasi suara dan dikirim ke otak, sehingga bisa didengar oleh kita.

Komputer hanya mampu mengenal sinyal dalam bentuk digital yang merupakan tegangan yang diterjemahkan dalam angka 0 dan 1 --> bit. Tegangan ini berkisar mendekati angka 5 volt bagi angka 1 dan mendekati 0 volt bagi angka 0. Komputer mampu melihat angka-angka 0 dan 1 menjadi kumpulan bit-bit dan menerjemahkan menjadi sebuah informasi.

a. Proses Konversi analog ke digital

Dalam memasukkan suara analog ini sehingga dapat dimanipulasi oleh peralatan elektronik yang ada. Alat yang diperlukan untuk melakukan ini adalah transducer. Dalam hal ini, transducer adalah istilah untuk menyebut sebuah peralatan yang dapat mengubah tekanan udara (yang kita dengar sebagai suara) ke dalam tegangan elektrik yang dapat dimengerti oleh perangkat elektronik, serta sebaliknya. Contoh transducer adalah mikrofon dan

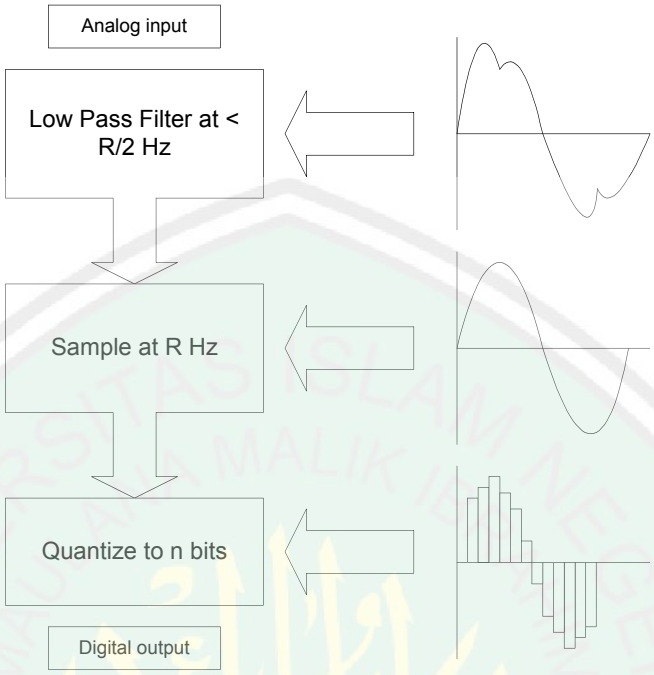
speaker. Mikrofon dapat mengubah tekanan udara menjadi tegangan elektrik, sementara speaker melakukan pekerjaan sebaliknya.

Tegangan elektrik diproses menjadi sinyal digital oleh sound card. Ketika Anda merekam suara atau musik ke dalam komputer, sound card akan mengubah gelombang suara (bisa dari mikrofon atau stereo set) menjadi data digital, dan ketika suara itu dimainkan kembali, sound card akan mengubah data digital menjadi suara yang kita dengar (melalui speaker), dalam hal ini gelombang analog. Proses pengubahan gelombang suara menjadi data digital ini dinamakan Analog-to-Digital Conversion (ADC), dan kebalikannya, pengubahan data digital menjadi gelombang suara dinamakan Digital-to-Analog Conversion (DAC).

Membatasi frekuensi sinyal yang akan diproses dengan ***Low Pass Filter.***

Proses pengubahan dari tegangan analog ke data digital ini terdiri atas beberapa tahap yang ditunjukkan pada gambar 2.2. yaitu:

- Membatasi frekuensi sinyal yang akan diproses dengan *Low Pass Filter*.
- Mencuplik sinyal analog ini (melakukan sampling) menjadi beberapa potongan waktu.
- Cuplikan-cuplikan ini diberi nilai eksak, dan nilai ini diberikan dalam bentuk data digital.

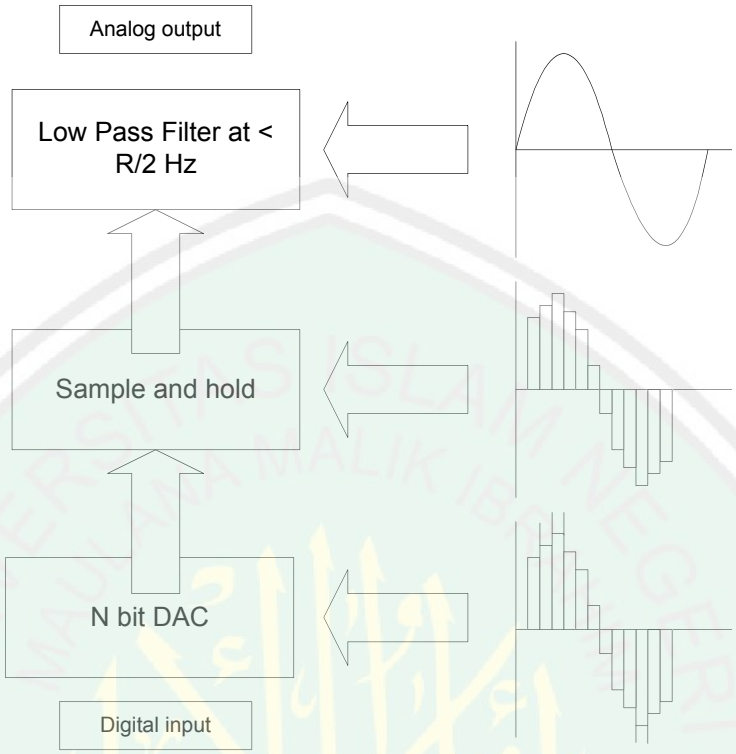


Gambar 2.5 Konversi sinyal analog ke digital
(Sumber dari https://ribkaginting.wordpress.com/2013/03/20/perhitungan-sound-audio-digital/ diakses pada tanggal 3 September 2014 19.00)

b. Proses Konversi digital ke analog

Proses sebaliknya, yaitu pengubahan dari data digital menjadi tegangan analog juga terdiri atas beberapa tahap, yang ditunjukkan pada gambar 2.3, yaitu:

- Menghitung data digital menjadi amplitudo-amplitudo analog.
- Menyambung amplitudo analog menjadi sinyal analog.
- Memfilter keluaran dengan Low Pass Filter sehingga bentuk gelombang keluaran menjadi lebih mulus.



Gambar 2.3Konversi sinyal digital ke analog.
(Sumber dari https://ribkaginting.wordpress.com/2013/03/20/perhitungan-sound-audio-digital/ diakses pada tanggal 3 September 2014 19.00)

Proses pengubahan sinyal analog menjadi digital tadi harus memenuhi sebuah kriteria, yaitu kriteria *Nyquist*. Kriteria ini mengatakan bahwa untuk mencuplik sebuah sinyal yang memiliki frekuensi X Hertz, maka kita harus mencupliknya minimal dua kali lebih rapat, atau 2 Hertz. Jika tidak, sinyal tidak akan dapat dikembalikan ke dalam bentuk semula.

Kelebihan audio digital adalah kualitas reproduksi yang sempurna. Kualitas reproduksi yang sempurna yang dimaksud adalah kemampuannya untuk menggandakan sinyal ausio secara berulang-ulang tanpa mengalami penurunan kualitas suara.

2.3.1 Sample rate

Sample rate adalah banyaknya jumlah *sample* (bentuk Frekuensi) yang di ambil dalam satuan waktu (detik) dari signal yang diterima dalam bentuk terus-menerus *(Continuous Signal)* menjadi signal yang terpisah *(Discrete Signal)*., atau dalam bahasa sederhana adalah batas frekuensi yang dapat dikirim perdetiknya.

Tingkat sampling yang umum digunakan pada Audio Digital adalah 44,1 kHz, 48 kHz, 88,2 kHz, 96 kHz, dan 192 kHz. Untuk standar Audio CD, menggunakan tingkatan Sampling 44,1 kHz (44100 Hz). Alasan kenapa Audio CD menggunakan sampling rate 44,1 kHz karena batas kemampuan telinga manusia untuk menangkap frekuensi suara adalah dari 20 Hz sampai 20 kHz, sehingga sample rate yang paling cocok/Efisien untuk digunakan adalah 44,1 kHz (*Nyquist-Shannon Sampling Theory*).

2.3.2 Bit Rate

Bit adalah satuan data yang di dasari oleh penghitungan bilangan Biner. Bilangan Biner adalah bilangan yang digunakan untuk proses data di peralatan eletronik, atau dalam bahasa sederhananya disebut bahasa mesin. *Bit Rate* adalah kecepatan transfer data yang terkompresi di dalam audio data pada setiap detiknya. *Bit depth* adalah besarnya data yang terdapat di setiap detiknya. *Bit depth* adalah ukuran data digital yang anda terima/proses dari proses konversi dari analog ke digital pada audio converter anda.

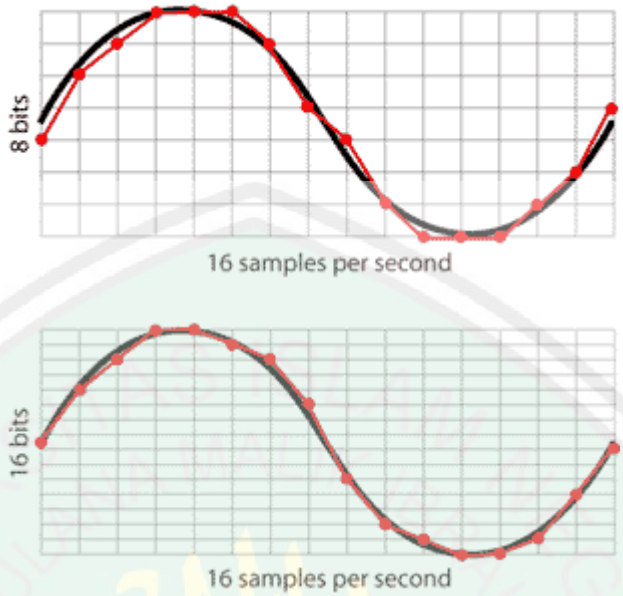
Di dalam metode *PCM sampling* (*Pulse-Code Modulation* yaitu metode yang digunakan untuk mewakili sample sinyal analog ke dalam sample digital) , *bit depth* akan mempengaruhi *signal-to-noise ratio* (S / N, yaitu ratio dari besarnya signal dengan

besarnya noise). Ukuran *bit depth* tidak akan membatasi *frequency range* yang sudah di atur melalui *Sample rate*.

Semakin ditingkatkan *bit depth*, kesalahan kuantisasi (gelombang sinyal audio) akan berkurang dan S/N akan meningkat, begitu pula sebaliknya. Hubungan antara *bit depth* dengan S / N yaitu, setiap *bit depth* meningkat 1-bit maka S / N akan meningkat 6 dB.

Secara teoritis, 24-bit ukuran audio digital memiliki S / N maksimum 144 dB, dan 16-bit ukuran audio digital memiliki S / N maksimum 96dB. Namun dalam dunia nyata, 24-bit ukuran audio digital S / N yang batasi menjadi 124 dB (21-bit). *bit depth* digunakan untuk menentukan *Dinamic Range (*yaitu signal tertinggi dan signal terendah yang mampu di rekam atau pun diolah). Untuk 16-bit, signal tertinggi yang dapat di capai adalah 96dB, dan terendah hingga -90dB.dan dan *dynamic range* untuk 24-bit secara teoritis mulai dari -120 dB hingga 144 dB.

Untuk standar Audio CD, *bit depth* yang ddigunakan adalah 16-bit, diukur dari efisiensi *dynamic range* (didasari batas kemampuan telinga manusia mendengarkan minimal 0 dB SPL dan paling maksimal 135 dB SPL mampu mendengar beserta rasa sakit di telinga). untuk 24-bit, 144 dB merupakan ukuran yang terlalu kuat, dan dapat memungkinkan telinga manusia terluka dan pada -120 dB sudah jauh sekali dari batas minimum pendengaran manusia. Namun 24-bit sangat cocok apa bila amplitudo yang digunakan mampu di gunakan hingga 124 dB (pembulatan dari 144 dB)



Gambar 2.6 Bit Rate Persecond
(Sumber dari http://hendra-dwi-irvanto.blogspot.com/ diakses pada tanggal 4 September 2014 7:30)

2.4 KONVERSI SUARA

Dalam seni musik terdapat istilah konversi. Untuk mengubah data audio analog menjadi digital, proses konversi data yang diperlukan. Tranformasi ini dilakukan melalui beberapa cara. Cara pertama adalah dengan encoding. Dalam encoding, sinyal analog diubah menjadi kode biner. Cara kedua adalah dengan metode sampling. Pada proses ini, akan diambil contoh data audio dan data tersebut kemudian dikuantisasi ke dalam level bit tertentu. Semakin besar level bit yang dipakai, kualitas audio yang dipakai akan menjadi bagus dan mendekati kualitas master audionya.

Setiap manusia akan mempunyai kisaran *pitch* tersendiri, tergantung dari bentuk pangkal tenggorokan yang dimilikinya. Untuk pria memiliki kisaran *pitch* sebesar 50-250 Hz dan untuk wanita memiliki kisaran *pitch* sebesar 120-500 Hz. Tinggi rendahnya nilai *pitch* tergantung pada intonasi suara dan tingkat emosi dari manusia

2.5 Perhitungan Konversi Suara Digital

System audio digital adalah terbentuk atas konversi analog ke digital. Audio digital dibuat saat mengkonversikan sebuah gelombang suara ke dalam angka, prosesnya disebut dihitizing. Bila suara yang dibutuhkan untuk digunakan dalam aplikasi computer, perlu mengubah gtaran udara suara menjadi sinyal listrik, yang disebut digital sinyal-sinyal aliran 0 dan 1 tersebut.

Suara digital merupakan sampel suara. Sampel suara diambil dan disimpan sebagai informasi digital dalam bit dan byte. Kualitas dari recoding digital tergantung pada seberapa sering sampel diambil (angka sampling atau frekuensi dalam kilohertz, 1000 sampel per detik) dan berapa banyak angka yang digunakan untuk mempresentasikan nilai dari tiap sample.

Ukuran bit sampelnya :

1. 8 bit
2. 16 bit

Semakin besar ukuran sampel, semakin baik data yang mendeskripsikan suara. Ukuran sample 8 bit menyediakan 256 unit untuk mendeskripsikan range dinamis atau amplitude level suara dalam satuan waktu dari potingan suara dinamis. Sementara itu, ukuran file 16 bit menyediakan 65.536 unit untuk mendeskripsikan range dinamis.

1. Bit resolusi

Bit rate adalah suatu ukuran kecepatan transfer suatu data dari satu tempat ke tempat lain yang diukur dengan waktu seperti Kbps (Kilobit per second), Mbps (Megabit per second) dan seterusnya.

2. Mono

   File mono adalah suara datar (satu channel). Perhitungan monofik dan stereo (dalam byte). Angka sampling *durasi recording dalam detik* (bit/resolusi/8) *1

3. Stereo

   File stereo adalah suara bervariasi (2 telingan manusia), ukuran file stereo 2x lebih besar dibandingkan file mono. Angka sampling * durasi recording dalam detik * durasi recording dalam detik * (bit resolusi/8) *2

   Contoh :

   Formula untuk recording 100 detik pada 44.1 kHz, resolusi 16 bit. Recording stereo 100 detik pada 44.1 kHz, resolusi 16 bit menjadi

   $$44100 \times 100 \times \frac{16}{8} \times 2 = 17.640.000\ byte$$

   Recording mono 100 detik pada 44.1 kHz, resolusi 16 bit menjadi

   $$44100 \times 100 \times \frac{16}{8} \times 1 = 8.820000\ byte$$

   Dari Byte ke Kbyte dan Mbyte

   Dari contoh sebelumnya : Stereo =17.640.000 byte

   = 17.640.000 / 1024 byte = 17226,5625 Kbyte

   =17226,5625 / 1024 = 16,8 Mbyte

2.6 Format File Audio Secara Umum

Mendengarkan musik/audio di komputer adalah aktivitas yang sudah biasa dilakukan. Walaupun tampaknya sederhana, namun file-file audio di komputer

terdiri dari berbagai macam variasi. Masing-masing file audio mempunyai ciri khas yang berbeda, dan seperti halnya software, format file audio pun mengenal free dan open format, serta propietary format.

Suara manusia (atau suara yang dihasilkan alat musik) merupakan fenomena fisik yang dihasilkan oleh suatu getaran. Getaran ini menghasilkan tekanan yang berbeda-beda di udara sekitarnya. Pola osilasi yang terjadi di udara tersebut diistilahkan dengan gelombang.

Bentuk gelombangnya adalah gelombang analog atau kontinu yang membawa informasi. Dua parameter/ karakteristik terpenting yang dimiliki oleh gelombang analog adalah amplitudo dan frekuensi. Amplitudo merupakan ukuran tinggi rendahnya tegangan dari gelombang analog, sedangkan frekuensi adalah jumlha gelombang analog dalam satu detik.

Gelombang suara ini memiliki lembah dan bukit, satu buah lembah dan bukit akan menghasilkan satu siklus. Siklus ini berlangsung berulang-ulang dan perulangan siklus tiap detiknya disebut frekensi. Satu unit frekuensi dinamakan sebagai Hartz atau Hz. Telinga manusia dapat mendengar bunyi antara 20 Hz hingga 20 kHz (20000). Artinya, bila sebuah benda dapat bergetar dan menghasilkan siklus tiap detiknya sebesar 20 kali, maka telinga dapat menangkap suara dari getaran benda tersebut.

Banyaknya cycle dalam 1 detik inilah yang menentukan “pitch” atau nada dari suatu suara. Contohnya, nada A adalah 440 cycle per detik. Sedangkan keras/pelannya suatu suara diwakili oleh amplitudo.

2.7 Dari Analog Menuju Digital

Gelombang suara analog ini tidak dapat langsung direpresentasikan atau direkam pada komputer. Komputer perlu untuk mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Komputer melakukan penyimpanan angka tersebut ke dalam sebuah file sebagai sebagai data yang nantinya digunakan saat file tersebut diakses (di-decode menjadi suara). Tiap satuan pengukuran ini dinamakan "sample".

Sebagai contoh, suatu CD Audio memiliki sampling rate sebesar 44,1 kHz atau 44100 Hz. Artinya dalam satu detik, sample yang diambil sebanyak 44.100. CD Audio ini merupakan format digital pertama yang dikembangkan oleh Sony pada tahun 1979. Pada tahun-tahun berikutnya, muncul berbagai format dengan media fisik penyimpanan yang berbeda-beda.

Dari format tersebut, bagaimana ukuran file ditentukan? Pada setiap sample diperlukan 2 byte (atau 16-bit data). Pada kualitas musik yang stereo untuk membedakan jalur kanan dan jalur kiri, maka diperlukan tambahan 2 x 2 byte = 4 byte, sehingga untuk dalam 1 detik yang terdiri dari 44.100 sample, besar file hasil penyimpanan adalah 4x44.100 atau 176.400 byte (172 KB). Jika durasi music adalah 4 menit, maka ukuran file sebesar 172 KB*4*60 detik = 41.280 KB (40 MB lebih). Karena begitu besarnya ukuran dari sebuah file audio, maka mulailah dikembangkan teknik kompresi agar ukurannya dapat menjadi lebih kecil. Salah satu teknik komresi adalah dengan mengurangi jumlah sample tiap detiknya. Kompresi ini akan berakibat menurunnya kualitas suara. Sekali kualitas suara diturunkan, maka tidak mungkin untuk dikembalikan ke kualitas suara

aslinya, dikarenakan adanya beberapa informasi (sampling rate) yang dihilangkan. Jenis kompresi semacam ini diiistilahkan sebagai lossy compression. Cara kompresi lain yang dikenal adalah lossless compression.

2.8 Jenis-Jenis Format File Audio

Secara umum, ada 3 kelompok utama format file audio, yaitu:

- Format file audio tanpa kompresi, seperti file WAV, AIFF, AU dan raw header-less PCM.
- Format file audio dengan kompresi lossy, seperti MP3, Vorbis, Mousepack, AAC, TRIAC, dan lossy Windows Media Audio (WMA).
- Format audio dengan kompresi lossless, seperti FLAC, Monkey's Audio (filename extension APE), WavPack (filename extension WV), Shorten, Tom's lossless audio compressor (TAK), TTA, ATRAC Advanced Lossless, Apple Lossless, MPEG-4 SLS, MPEG-4 ALS, MPEG-4 DST, Windows Media Audio Lossless (WMA Lossless).

Dari format-format tersebut, terbagi menjadi 3 bagian, yaitu format yang free dan open (seperti wav, ogg, mpc, flac, aiff, raw, au, dan midi), free (gsm, dct, vox, aac, mp4, dan mmf), serta propeietary (mp3, wma, atrac, ra, ram, dss, msv, dvf, m4p, 3gp, amr, dan awb).

A. Format CD

Ekstensi : cda

File dengan ekstensi .cda merupakan representasi dari track CD-audio. File dengan format .cda dapat langsung dijalankan melalui CD-ROM, sementara filenya sendiri tidak mempunyai informasi kode modulasi

apapun sehingga jika dikopi ke dalam harddisk, file tersebut menjadi tidak dapat di-play. Pada November 1984, dua tahun setelah CD diproduksi secara missal, Sony mengeluarkan Discman sebagai media pemutar portable. Agar dapat mengambil/mengkopi file audio dari CD-Audio, dibutuhkan software khusus atau ripping untuk mengubah dari format .cda menjadi format lain yang dapat disimpan di computer.

B. Format Advanced Audio Coding (AAC)

Ekstensi : .m4a, .m4b, .m4p, .m4v, .m4r, .3gp, .mp4, .aac

AAC merupaka format audio menggunakan lossy compression (data hasil kompresi tidak bisa dikembalikan lagi ke data sebelum dikompres secara sempurna, karena ada data yang hilang).

Cara kerja AAC :

- Bagian-bagian sinyal yang tidak relevan dibuang
- Menghilangkan bagian-bagian sinyal yang redundan
- Dilakukan proses MDCT (Modified Discret Cosine Transform) berdasarkan tingkat kompleksitas sinyal
- Adanya penambahan Internal Error Connection
- Kemudian sinyal disimpan atau dipancarkan

Saat ini, AAC merupakan standar format untuk telepon selular seperti Apple's iPhone, Sony Ericsson, N-series, dan model S40 dari Nokia, serta telepon sel berbasis Android. Juga perangkat portable seperti iPod, iTunes, Sony Playstation Portable, generasi terbaru dari Walkman Sony, semua jenis telepon Nintendo's Wii, Nintendo DSi, mendukung format AAC.

Kepopuleran format ini dikarenakan audio codec-nya yang menyempurnakan MP3, seperti pada jangkauan sample rate yang lebih banyak (8 Hz-96 kHz), memiliki 48 channerl, dan suara yang lebih bagus untuk bit yang lebih rendah (di bawah 16 Hz).

Portable player untuk format file AAC adalah Archos, Creative Zen Portable, Microsoft Zune, SanDisk Sansa, Sony Playstation Portable (PSP), Sony Walkman, Nintendo DSi, dan Cowon.

C. Format Waveform Audio (WAV)

Ekstensi : .wav atau .wv

WAV merupakan format file audio yang dikembangkan oleh Microsoft dan IBM sebagai standar untuk menyimpan file audio pada PC, dengan menggunakan coding PCM (Pulse Code Modulation). Tidak seperti AAV, file WAV adalah file audio yang tidak terkompres sehingga seluruh sampel audio disimpan semuanya di media penyimpanan dalam bentuk digital. Karena ukurannya yang besar, file WAV jarang digunakan sebagai file audio di Internet.

D. Format Audio Interchange File Format (AIFF)

Ekstensi : .aiff, .aif, .aifc

File AIFF merupakan format file audio standar yang digunakan untuk menyimpan data suara untuk PC dan perangkat audio elektronik lainnya, yang dikembangkan oleh Apple pada tahun 1988. Standar dari file AIFF adalah uncomressed code pulse-modulation (PCM), namun juga ada varian

terkompresi yang dikenal sebagai AIFF AIFF-C atau aifc, dengan berbagai kompresi codec.

E. Format MPEG Audio Layer 3 (MP3)

Ekstensi : .mp3

Pada awalnya, format MP3 ini dikembangkan oleh seorang Jerman bernama Karlheinz Brandenburg, memakai pengodean Pulse Code Modulation (PCM). Prinsip yang dipergunakan oleh MP3 adalah mengurangi jumlah bit yang diperlukan dengan menggunakan model psychoacoustic untuk menghilangkan komponen-komponen suara yang tidak terdengar oleh manusia – sehingga adapat digolongkan file audio dengan kompresi lossy.

Pada tahun 1991, file MP3 distandarisasi dan tahun 1994 hingga akhir tahun 2000, popularitas dari MP3 semakin meningkat dengan semakin mudahnya akses Internet. Munculnya software untuk menjalankan file MP3 seperti Winamp di tahun 1997 yang dikembangkan oleh Nullsoft, dan player console untuk Linux, mp123, juga membuat file MP3 semakin digemari.

Beberapa batasan dari file MP3 ini adalah :

- Bit rate terbatas, maksimum 320 kbit/s (beberapa encoder dapat menghasilkan bit rate yang lebih tinggi, tetapi sangat sedikit dukungan untuk mp3-mp3 tersebut yang memiliki bit rate tinggi).

- Resolusi waktu yang digunakan mp3 dapat menjadi terlalu rendah untuk sinyal-sinyal suara yang sangat transient, sehingga dapat menyebabkan noise.
- Resolusi frekuensi terbatasi oleh ukuran window yang panjang kecil, mengurangi efisiensi coding.
- Tidak ada scale factor band untuk frekuensi di atas 15,5 atau 15,8 kHz.
- Mode jointstereo dilakukan pada basis per frame.
- Delay bagi encoder/decoder tidak didefinisikan, sehingga tidak ada dorongan untuk gapless playback (pemutaran audio tanpa gap). Tetapi, beberapa encoder seperti LAME, dapat menambahkan metadata tambahan yang memberikan informasi kepada MP3 player untuk mengatasi hal ini.

F. Format MIDI

Ekstensi : .mid

Merupakan standar yang dibuat oleh perusahaan alat-alat music elektronik berupa serangkaian spesifikasi agar berbagai instrument dapat berkomunikasi.

Dengan menggunakan format MIDI, perangkat elektronik seperti keyboard dan computer dapat melakukan sinkronisasi satu sama lain.

Interface MIDI terdiri dari 2 komponen yaitu :

- Perangkat keras, merupakan hardware yang terhubung dengan peralatan (keyboar/computer)

- Data format yang mengandung pengkodean informasi (spesifikasi instrument, awal/akhir nada, frekuensi dan volume suara).

G. Format Monkey's Audio

Ekstensi : .ape, .api

Merupakan format file audio dengan kompresi lossless sehingga tidak mengurangi kualitas suara. Umumnya, sebuah file audio dengan format Monkey's Audio mempunyai ukuran lebih besar 3-5 kali dibandingkan dengan format MP3 (pada bitrate 192 Kb/s). Secara resmi, Monkey Audio hanya mendukung platform Windows, seperti yang ditulis di website resminya. Pada masa-masa mendatang, Monkey Audio akan mendukung untuk platform Linux dan Mac OS. Player yang dapat digunakan untuk menjalankan file format ini adalah Monkey's Audio.

2.9 FORMAT FILE .WAV

Penelitian ini menggunakan format file.wav. WAV adalah singkatan dari istilah dalam bahasa Inggris *waveform audio format* merupakan standar format berkas audio yang dikembangkan oleh Microsoft dan IBM. Hal ini telah menjadi slah satu format digital yang paling banyak didukung oleh file audio pada PC karena popularitas Windows dan sejumlah besar program yang ditulis untuk platform. Hamper setiap program modern yang dapat membuka dan atau menyimpan audio digital mendukung format file, sehingga keduanya sangat berguna dan persyaratan virtual untuk pengembang penrangkat lunak untuk memahami.

Format file wav merupakan bagian dari spesifikasi RIFF milik M icrosoft yang digunakan untuk penyimpanan file-file multimedia. File wav dimulai dengan bagian header dan diikuti oleh rentetan data chunk. File wav terdiri dari 3 bagian, yaitu main chunk, format chunk, dan data chunk. Sinyal suara yang direpresentasikan file wav dalam bentuk discrete, berupa deret bilangan yang merepresentasikan amplitudo dalam domain waktu. Pada bagian file header terdapat informasi tentang file wav tersebut, diantaranya menyatakan nilai sample rate, jumlah channel, dan bit per sample. Dari keterangan pada file header7 tersebut dapat diketahui berapa sampel yang dicuplik dari sinyal analog tiap detik. Struktur WAV dapat dilihat pada Gambar di bawah ini :

Gambar 2.7 Struktur WAV

(Sumber dari http://duniainformatikaindonesia.blogspot.com/2013/03/struktur-file-wave_18.html diakses pada tanggal 7 September 2014 15:55)

1. Bagian Main Chunk

   ChunkID : berisi kata "RIFF" dalam format ASCII.

   ChunkSize : berisi informasi ukuran chunk.

   Format : berisi kata "WAVE"

2. Bagian Format Chunk

   SubChunk1ID : berisi kata "fmt".

   SubChunk1Size : berisi informasi ukuran subchunk 18

3. AudioFormat : informasi jenis kompresi data chunk. Misalnya bernilai 1

untuk kompresi PCM.

NumChannels : banyaknya channel. M isal: M ono=1, Stereo=2.

SampleRate : sample rate dari file wav, misal 8000 untuk 8000Hz, 44100 untuk 44100 Hz. ByteRate : banyaknya byte tiap detik. ByteRate = SampleRate xNumChannels x BitsPerSample / 8.

BitsPerSample : ukuran bits untuk tiap sampel.M isal: 8 bit = 8

4. Bagian Data Chunk

SubChunk2ID : berisi kata "data".

SubChunk2Size : berisi informasi ukuran subchunk2. SubChunk2Size = NumSamples x NumChannels x BitsPerSample / 8. NumSamples = (DataByte / NumChannels) / BitsPerSample.

Data: Data suara aktual dalam byte, merepresentasi-kan amplitudo tiap sampel dari sinyal.

2.10 ALGORITMA Waveform-Similarity-Based Synchronized Overlap-Add (WSOLA).

WSOLA adalah teknik pemrosesan sinyal digital yang digunakan untuk pengolahan sintesis suara. WSOLA dapat digunakan untuk memodifikasi durasi dari sinyal suara. Algoritma WSOLA bekerja dengan membagi bentuk gelombang suara pada bagian overlape. Untuk mengubah frekuensi sinyal, segmen dipindahkan ke segmen yang lebih dekat. Untuk mengubah time scale, segmen diulang beberapa kali dengan cara dieliminasi. Segmen tersebut kemudian digabungkan dengan menggunakan teknik *time scale*.

WSOLA adalah Sebuah konsep waveform similarity diusulkan dengan tujuan untuk memecahkan masalah modifikasi skala suara. Wsola bekerja dalam konteks short-time Fourier transform. Algoritma WSOLA menghasilkan output kualitas suara yang bagus dengan komputasi yang efisien secara algoritma sehingga memungkinkan pengolahan suara secara online dengan faktor skala waktu yang telah ditentukan.

Beberapa variabel yang terlibat pada penelitian ini adalah :

1. Current Factor adalah besaran faktor pitch sekarang
2. *Sample rate* adalah banyaknya jumlah *sample* (bentuk Frekuensi) yang di ambil dalam satuan waktu (detik) dari signal yang diterima dalam bentuk terus-menerus *(Continuous Signal)* menjadi signal yang terpisah *(Discrete Signal)*. atau dalam bahasa sederhana adalah batas frekuensi yang dapat dikirim perdetiknya. Tingkat sampling yang umum digunakan pada Audio Digital adalah 44,1 kHz, 48 kHz, 88,2 kHz, 96 kHz, dan 192 kHz. Untuk standar Audio CD, menggunakan tingkatan Sampling 44,1 kHz (44100 Hz). Alasan kenapa Audio CD menggunakan sampling rate 44,1 kHz karena batas kemampuan telinga manusia untuk menangkap frekuensi suara adalah dari 20 Hz sampai 20 kHz, sehingga sample rate yang paling cocok/Efisien untuk digunakan adalah 44,1 kHz (*Nyquist-Shannon Sampling Theory*).
3. Sequence permili second adalah panjang 1 proses sequence dalam milisecond. Yang membatasi seberapa panjang suara asli diolah dengan menggunakan algoritma time-strecth. Semakin besar nilai yang digunakan, maka semakin sedikit sequence yang digunakan di dalam proses. Nilai yang lebih besar

terdengar lebih baik ketika memperlambat tempo dan lebih buruk ketika menaikkan tempo. Menaikkan nilai sequence akan mengurangi beban komputasi dan sebaliknya. Dalam penelitian ini menggunakan standart music yakni 82 permilisecond.

4. Panjang Windows m/s berguna untuk menemukan posisi overlopping terbaik. Rguna untuk membatasi seberapa lebar window algorotma yang memungkinkan untuk melakukan pencampuran lokasi yang optimal saat pencampuran urutan suara kembali secara bersama-sama. Semakin besar pengaturan jendela ini, semakin tinggi kemungkinan untuk menemukan posisi yang lebih baik saat pencampuran terjadi. Akan tetapi, pada saat yang bersamaan pengaturan nilai jendela yang besar akan menyebabkan "drifting" artifact yang disebabkan urutan konsekuen akan diambil pada interval yang lebih merata. Jika suara terdengar seperti mengambang hal ini berarti panjang window terlalu panjang. Panjang window yang besar akan mengakibatkan beban komputasi begitu juga sebaliknya.
5. OverlapMs = 12;Overlap length in milliseconds. Ketika terjadi pencampuran sequence suara kembali secara bersama-sama ke sebuah bentuk continues stream. Parameter ini mendefinisikan seberapa panjang periode 2 sequence yang dibiarkan overlap satu sama lain. Pengaturan yang terlalu besar akan mengakibatkan proses komputasi yang besar begitu juga sebaliknya.
6. Tempo : 1.0 berarti tidak berubah, 2.0 berarti berubah + 100% dan 0.5 adalah setengah dari kecepatan.

2.11 Kajian islam

Seni baca al-Qur'an itu sudah ada sejak zaman Rasulullah SAW. Rasulullah SAW adalah seorang qari yang mampu mendengungkan suaranya tatkala membaca al-Qur'an. Rasulullah SAW adalah orang yang menyukai seni baca al-Qur'an, beliau sangat senang ketika membaca al-Qur'an dengan memakai lagu dan irama. Meskipun tidak selalu memakai lagu ketika Rasulullah SAW membaca al-Qur'an.

Tujuan dari Rasulullah membaca al-Qur'an dengan memakai lagu adalah untuk mencontohkan kepada umat Islam agar mau belajar dan tertarik untuk membaca al-Qur'an. Dengan demikian melagukan bacaan ayat suci al-Qur'an adalah seni baca yang tinggi nilainya dalam ajaran agama Islam. Di kalangan sahabat sendiri dan juga qari kenamaan yang disayangi Nabi SAW seperti: Abdullah bin Mas'ud dan Abu Musa Al-Asy'ari ketika membaca al-Qur'an juga sering dilagukan. Dengan demikian menunjukkan bahwa sejak zaman Nabi dan sahabat, membaca al-Qur'an dengan lagu yang merdu sudah ada. Seiring dengan perkembangan zaman dan teknologi yang semakin maju sebenarnya masyarakat masih bisa belajar tilawah melalui media elektronik, (mp3, vcd, dan lain-lain), tetapi kenyataannya masih ada mahasiswa belajar tilawah al-Qur'an, padahal belajar tilawah al-Qur'an itu tidak wajib hukumnya.

Hal ini sesuai dengan beberapa pendapat para ulama tentang hukum tilawah yaitu:

1. Pendapat dari Abu Abdillah Muhammad bin Idris As-Syafi'i Al-Muttalibi Al-Qurashi dalam kitab Mukhtashar menegaskan bolehnya membaca al-Qur'an dengan lagu (al-han).
2. Pendapat Syaikh Mahmud Khalil al-Hushari, sebagai tokoh qurra kenamaan berpendapat bahwa tilawatil Qur'an adalah boleh selama tidak keluar dari kaedah-kaedah tajwid yang ditetapkan oleh para ulama. Adapun sebaliknya, yakni membaca dengan lagu tapi keluar dari kaedah-kaedah yang ditentukan adalah haram hukumnya menurut ijma'(pendapat) ulama.
3. Pendapat Abu Hasan Ali bin Muhammad Habibal Mawardi al-Bashri, bahwa melagukan al-Qur'an prinsipnya adalah boleh selama tidak keluar dari kaedah-kaedah tajwid, maksudnya adalah bisa menyesuaikan antara lagu dan tajwid, sehingga lagu sendiri tidak merusak bacaan.

Dari beberapa pendapat para ulama yang telah disebutkan, bahwasanya membaca al-Qur'an dengan lagu adalah dibolehkan asalkan tidak keluar dari kaedah-kaedah tajwid yang telah ditentukan oleh para ulama. Di dalam belajar tilawah al-Qur'an, suara adalah faktor yang paling menentukan, di samping tajwid dan makharijul huruf. Memang di antara tajwid dan makharijul huruf tidak dapat dipisahkan, walaupun mempunyai sifat-sifat yang tidak sama.

Dalam hal ini suara yang bersih, merdu dan menggema adalah pembawaan seseorang yang tidak dapat diusahakan sedangkan lagu adalah sesuatu usaha yang dapat dipelajari dan dicapai oleh seseorang. Pembawaan suara yang indah dan bagus sangat memerlukan adanya pemeliharaan terutama pengaturan pernapasan. Setiap orang yang berniat ingin mempelajari tilawah al-Qur'an dengan baik, maka

ia harus memulai dari tingkat pemeliharaan tubuh, khususnya alat yang berhubungan dengan pernafasen. Tilawah al-Qur'an akan lebih banyak membutuhkan nafas dan suara. Organ pernafasan yang perlu diperhatikan adalah berpusat pada bagian perut, dada, leher, dan bagian kepala.

Untuk memiliki pernafasen yang baik dalam tilawah al-Qur'an, ada beberapa hal yang harus diperbuat, antara lain berolahraga, melakukan pergerakan pada seluruh tubuh sampai terasa panas dan berkeringat. Suara yang bagus dalam melagukan al-Qur'an adalah suara bening, suara merdu, suara asli dan mampu menggunakan tinggi rendahnya nada. Tidak sedikit orang yang mempunyai suara baik, menjadi hilang dengan sia-sia karena tidak ada pelatihan yang dilakukan secara rutin. Sebaliknya ada orang yang mempunyai suara yang sederhana tetapi berkat latihan yang bersungguh-sunguh akhirnya menjadi suara yang bagus, atau setidaknya ia akan mengetahui cara-cara melagukan al-Qur'an dengan baik.

Al-Suyuti mengatakan di sunnahkan untuk memperindah suara dalam membaca al-Qur'an dan menghiasinya dengan alasan hadits Ibnu Hibban yang artinya: "Perindahlah al-Qur'an dengan suara kalian."

BAB III

METODE PENELITIAN

3.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan penelitian positivis kuantitatif, penelitian yang menganalisis data yang berbentuk angka. Menurut Arikunto (2010), penelitian kuantitatif merupakan penelitian yang dalam prosesnya banyak menggunakan angka-angka mulai dari pengumpulan data, penafsiran terhadap data, serta penampilan dari hasilnya.

Penelitian ini merupakan sebuah studi kasus analisis keberhasilan implementasi algoritma Waveform Similarity Based Overlap-Add (WSOLA). Jenis format file audio yang dikonversi adalah format file .WAF dengan alasan file format audi .WAV adalah format asli yang mudah untuk dilakukan analisis.

3.2 Objek Penelitian

Menurut Sugiyono (2009) pengertian objek penelitian adalah suatu atribut atau sifat atau nilai dari orang, objek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya. Objek penelitian dalam penelitian ini adalah sebuah aplikasi audio player, yang telah dibuat oleh peneliti menggunakan bahasa pemrogaman java dan teknologi java sound. Aplikasi tersebut mampu melakukan konversi suara dengan menggunakan yang telah dibangun dengan menggunakan algoritma Waveform Similarity Based Overlap-Add (WSOLA).

3.3 Pendefinisian Variable

Pada penelitian ini terdapat beberapa variabel yang digunakan dalam menerapkan metode Waveform Similarity Based Overlap-Add (WSOLA) yaitu:

3.3.1 Variabel bebas

Variabel bebas merupakan variabel yang bisa mempengaruhi variabel penghubung dan terikat. Dalam penelitian ini, variabel bebas yang digunakan yaitu inputan berupa file audio dalam format WAV. Fortmat WAV yang digunakan untuk penelitian adalah format .WAV original dan format .WAV yang telah dilakukan kompresi.

3.3.2 Variabel Penghubung

Variabel penghubung adalah variabel yang bisa mempengaruhi variabel terikat. Variabel-variabel penghubung dalam penelitian ini bisa mempengaruhi variabel terikat (kualitas output suara). Variabel-variabel tersebut adalah:

a. Factor Pitchshift adalah besaran pitch yang diberikan
b. Tempo adalah ukuran kecepatan suara.
c. Gain adalah rasio antara input dan output.
d. Sample Rate adalah banyaknya jumlah sample yang diambil dalam satuan detik.
e. sequenceMs adalah adalah panjang 1 proses sequence dalam milisecond.
f. seekWindowMs adalah panjang window permilisecond
g. overlapMs adalah jumlah overlape permilisecond.

3.3.3 Varuiabel Terikat

Variabel terikat merupakan variabel yang dipengaruhi oleh variabel bebas dan terikat. Dalam penelitian ini, variabel terikat yang digunakan yaitu kualitas output suara yang dihasilkan dari proses pengolahan suara yang diinputkan.

3.4 Sumber Data

Sumber data seluruh yang tertulis dalam penelitian ini didapat melalui internet, buku baik buku berupa file maupun buku yang dalam bentuk cetakan serta data dari jurnal penelitian terdahulu. Sumber data lain yang digunakan dalam penelitian ini adalah gambar-gambar yang terkait dengan penelitian ini.

3.5 Prosedur Penelitian

Prosedur penelitian adalah tahapan penelitian yang dilakukan oleh penulis, untuk melakukan penelitian. Tahapan-tahapan penelitian tersebut sebagaimana pada flowchart pada gambar 3.1.

3.5.1 Study Literature

Study literature (kajian pustaka) merupakan keegiatan yang dilakukan penulis untuk melakukan penelusuran literatur yang bersumber dari buku, media, pakar ataupun dari hasil penelitian orang lain yang bertujuan untuk menyusun dasar teori proses konversi suara. Di dalam proses ini dilakukan proses observasi progamming audio pada java dan library yang mendukung untuk pembuatan aplikasi menggunakan java.

1. Perancangan dan Anaslisis Kebutuhan Sistem.

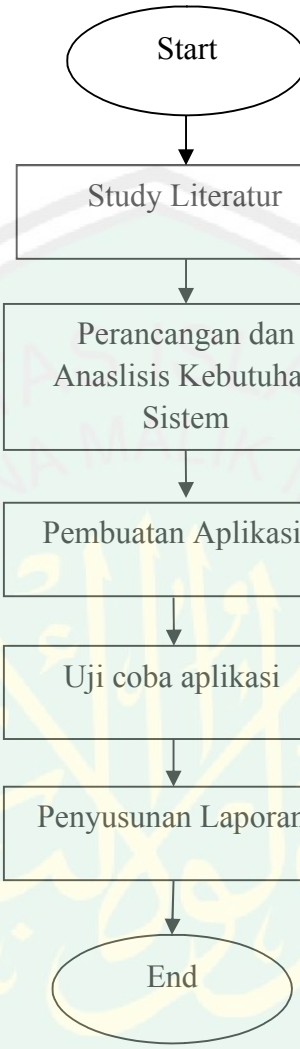
   Perancanaan dan pembuatan aplikasi ini dibagi menjadi 2 tahapan, yaitu:

a. Analisis

Mengidentifikasi permasalahan-permasalahan yang ada pada aplikasi yang akan di bangun, yang meliputi perangkat keras (hardware), perangkat lunak (software) dan pengguna. Membuat analisa terhadap kesimpulan-kesimpulan dari hasil observasi.

b. Perancangan

Memahami perancangan sistem sesuai dengan data yang ada dan mengimplementasikan model yang diinginkan sesuai kebutuhan penelitian. Pemodelan sistem ini berupa pembuatan use case diagram, flowchart dan desain interface guna mempermudah dalam proses-proses selanjutnya.



Gambar 3.1 Prosedur penelitian

2. Pembuatan Aplikasi

   Membuat dan menyelesaikan program serta keseluruhan, yaitu menggabungkan perancangan aplikasi yang berdasarkan sintaks dan struktur.

3. Uji coba dan Evaluasi

   Uji coba dan evaluasi dilakukan untuk melakukan evaluasi keberhasilan aplikasi.

4. Penyusunan Laporan

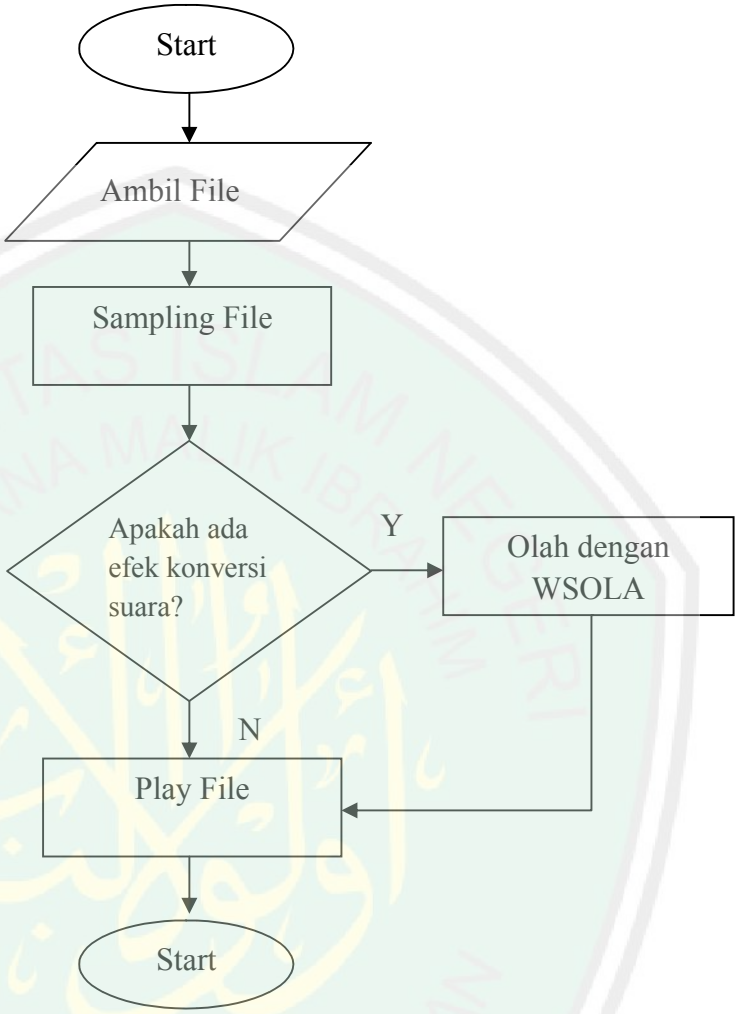
Penyusunan laporan bertujuan untuk mendokumentasikan hasil uji coba yang telah dilakukan pada proses sebelumnya.

3.6 Perancangan dan Analisis System

Desain system menggambarkan rancangan serta alur yang akan dibangun. Desain system tersebut terdiri dari :

3.6.1 Desain Alur Sistem

Alur system menjelaskan alur pengolahan sinyal dari proses pengambilan sinyal dari file audio, proses konversi, dan pemutaran file yang telah di transpose. Alur system secara umum sebagaimana tergambarkan pada flowchart 3.2.



Gambar 3.2 ***Flowchat aplikasi secara umum***

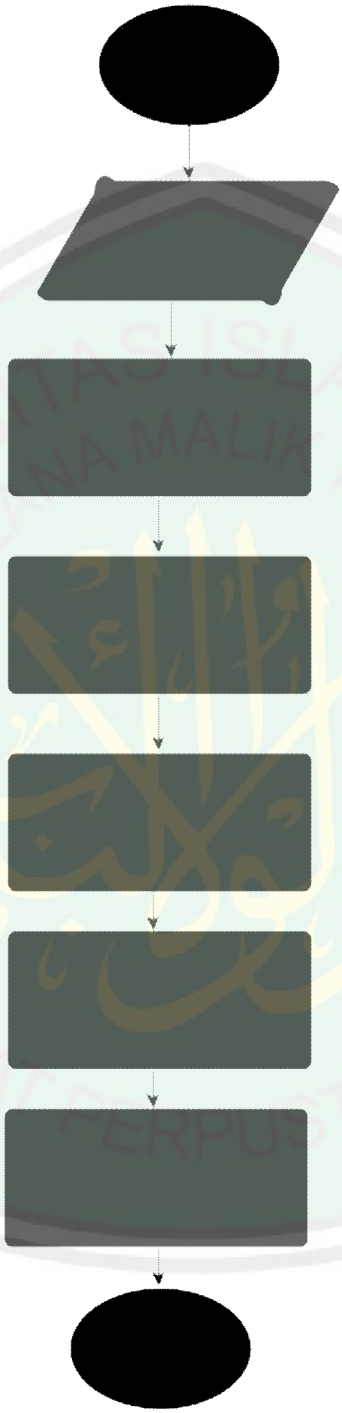
Pada gambar 3.2 Menjelaskan alur system Flowchat aplikasi secara umum.

1. Sistem dimulai dengan penginputan file audio, File yang diinputkan ke dalam system adalah file audio dengan format .WAV. Jika file audio sesuai maka file audio siap untuk diproses, sedangkan kalau tidak sesuai user harus menginputkan ulang file audio dengan format .WAV.
2. Proses sampling, yakni mengkonversi suatu sinyal waktu-kontinu mejadi sinyal diskrit yang diperoleh dengan mengambil cuplikan (*sample*) sinyal waktu-kontinu pada saat waktu-diskrit.

3. Pengecekan efek *pitchshift*, Aplikasi akan melakukan pengecekan apakah user memberikan efek pitcshift atau tidak. Jika terdapat efek *pitchshift* maka sinyal tersebut diolah dengan menggunakan algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*)*. Jika user tidak memberikan efek *pitchshift* maka sinyal original tersebut akan di play menggunkan *engine* java sound.

3.6.2 Desain Alur Sistem Penerapan Algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*).*

Jika terdapat efek *pitchshift* maka sinyal tersebut diolah dengan menggunakan algoritma *Waveform Similarity Based Overlap-Add (*WSOLA*)* dengan langkah-langkah sebagaimana gambar 3.3.



Gambar 3.3 Proses Konversi suara dengan Wafeform Similarity Based Overlap-Add *(*WSOLA*).*

Berdasarkan flowchart pada gambar 3.3, maka proses pengkonverian suara dengan metode Wafeform Similarity Based Overlap Add (WSOLA) adalah sebagai berikut:

1. Input Buffer, Buffer adalah sinyal yang telah dilakukan proses penyamplingan sebagaimana pada gambar 3.2.
2. Mencari Overlape Position Terbaik

   langkah-langkahnya adalah :

   a. Menurunkan amplitudo pada samples midBuffer.

      Temp = n x (panjang overlape – overlape n);

      pRefMidBuffer[n] = pMidBuffer ke n x temp;

   b. Melakukan proses scanning untuk nilai korelasi terbaik dengan menguji setiap posisi yang memungkinkan pada rentang yang diizinkan.

   c. Menghitung nilai korelasi untuk posisi percampuran berdasarkan tempOffset.

   d. aturan heuristik untuk mencari nilai yang mendekati range nilai tengah.

   e. Mengecek correlation value tertinggi
3. Overlape

   output n + offset = input [n+offset] x posisi

   output[i + outputOffset] = (input [ke n+ inputOffset] x n + pMidBuffer ke n x itemp) / overlapLength
4. Mengcopy sequence sample dari input ke output.
5. Mengcopy sequence terakhir dari input buffer ke mid buffer untuk dicampur awalan proses sequence proses.

3.6.3 Desain Interface

Desain interface adalah rancangan tampilan sistem yang akan dibuat pada penelitian ini. Pembuatan desain interface berdasarkan pada kemampuan user pada aplikasi sebagaimana yang telah dijelaskan pada Use case sistem konversi suara dengan Wafeform Similarity Based Overlap-Add *(*WSOLA*)*

Gambar 3.7 Desain Interface

Sesuai rancangan antarmuka pada gambar 3.7 masing-masing komponen mempunyai fungsi sebagai berikut:

a. Datagrid, Untuk menampilkan file yang telah diinputkan ke aplikasi.
b. Browse File, Untuk melakukan Pengambilan File.
c. Button Play, Untuk memutar file yang ada pada list data grid.
d. Slider Volume, Untuk menambah dan mengurangi volume.
e. Slider Pitchshift, untuk menambah dan mengurangi efek pitcshift.
f. Spinner, Untuk mengetahui prosentase efek pitchshift yang telah diberikan.
g. Slider Gain, mengatur gain.

h. Checkbox tempo, untuk melakukan opsi perubahan kecepatan suara atau tidak.

3.7 Analisis Kebutuhan Sistem

3.7.1 Kebutuhan Non Fungsional

Analisis kebutuhan sistem dilakukan untuk mencari kebutuhan apa saja yang diperlukan untuk membuat sistem yang akan dibangun. Kebutuhan sistem yang diperlukan antara lain:

1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan unutk implementsi sistem yang akan dibuat. Adapun perangkat keras dibutuhkan anatara lain :

a. Processor yang digunakan intel Pentium core 2 duo 2.00 Ghz
b. Memory yang digunakan 1 GB 44.1 KHz
c. Harddisk, sebagai media storage yang digunakan 160 GB
d. Microphone
e. Speaker
f. Keyboard

2. Kebutuhan Perangkat Lunak

Pada bagian ini dijelaskan mengenai kebutuhan perangkat lunak dari aplikasi sistem identifikasi pengenalan suara pembicara. Adapun perangkat lunak yang dibtuhkan antara lain :

a. Sistem Operasi

Sistem Operasi yang digunakan adalah Windows 7 . Dipilih karena sistem operasi ini user friendly dengan aplikasi apapun, semua aplikasi compatible dengan sistem Operasi ini

b. Library dan Native Library

Library yakni sekumpulan class yang mempunyai tugas spesifik untuk mempermudah proses pengerjaan aplikasi. Library yang digunakan adalah Java sound yang telah terbundled pada JDK.

c. Netbeans

Netbeans digunakan sebagai editor pembuatan aplikasi.

3.8 Contoh Perhitungan Manual Algoritma WSOLA

Perhitungan manual memberikan gambaran penerapan algoritma Wafeform Similarity Based Overlap-Add (WSOLA) yang dilakukan secara matematis, dengan study kasus pemberian parameter sebagai berikut :

Sample Rate	: 44100	overlape/ms	: 12
Sequence /ms	: 82	tempo / pitscift factor	: 2
Seek Window/ms	: 28	Best Correlation	: -10
Best Offset	: 0		

Langkah Perhitungan Manualnya adalah sebagai berikut

a. Mencari Panjang Overlape

Panjang Overlape = (Sample Rate x Overlape /ms) : 1000

= 22050.0 x 3.0: 1000 = 66

b. Mencari Panjang Window

Panjang Window = (Sample Rate x Sequence /ms) : 1000

= 22050.0 x 82.0: 1000 = 1808

Panjang yang dicari = (Sample Rate x Panjang Window) : 1000

= 22050.0 x 28.0: 1000 = 617

c. Mencari Jumlah sample bit yang di skip

Sample Bit yang di skip =

tempo x (Panjang Seek Window - Panjang Overlape)

1.2x (1808 - 66=2090.4

Bit yang diskip =

nominal skip + 0.5

2090.4 + 0.5) =2090

d. Mencari sample req

sample Req =Mengambil nilai yang tertinngi penjumlahan antara skip + panjang over lape dan panjang Window panjang window, nilai tertinggi ditambah Panjang window yang dicari.

Output Buffer = Panjang Seek Window - Panjang Overlape

e. Mencari Overlape Position Terbaik

langkah-langkahnya adalah :

1. **Menurunkan amplitudo pada sample bit yang berada di tengah tengah Buffer dengan rumus.**

   Temp ke n = n x (panjang overlape – overlape ke n);

   pRefMidBuffer[n] = pMidBuffer ke n x temp;

   temp =0x (66-0)=0.0

pRefMidBuffer ke 0=0.0x 0.0=0.0

temp =1x (66-1)=65.0

pRefMidBuffer ke 1=0.0x 65.0=0.0

temp =2x (66-2)=128.0

pRefMidBuffer ke 2=0.0x 128.0=0.0

temp =3x (66-3)=189.0

2. **Melakukan proses scanning untuk nilai korelasi terbaik dengan menguji setiap posisi yang memungkinkan pada rentang yang diizinkan.**

   Compare Position = 0+ 0

   Corr =0.0x -0.0050050355=0.0

   Norm = 0.0x0.0=0.0

   Corr =0.0x -0.003418073=0.0

   Norm = 0.0x0.0=0.0

   Corr =0.0x -0.002929777=0.0

   Norm = 0.0x0.0=0.0

3. **Menghitung nilai korelasi untuk posisi percampuran berdasarkan tempOffset dengan rumus.**

   calcCrossCorr = 0.0/1.0=0.0

4. **Aturan heuristik untuk mencari nilai yang mendekati range nilai tengah.**

   tmp = (double) (2 x tempOffset - seekLength) / seekLength;

   currentCorrelation = ((currentCorrelation + 0.1) x (1.0 - 0.25 x tmp x tmp));

tmp= (2 x4-617) : 617=-0.9870340356564019

currentCorrelation = (0.07564409531139593+ 0.1) x (1 - 0.25 x -0.9870340356564019x-0.9870340356564019)=0.07564409531139593

5. **Mengecek correlation value tertinggi**

tmp= (2 x1-617) : 617=-0.9967585089141004

currentCorrelation = (0.07516181187268349+ 0.1) x (1 - 0.25 x -0.9967585089141004x-0.9967585089141004)=0.07516181187268349

tmp= (2 x2-617) : 617=-0.993517017828201

currentCorrelation = (0.07532309838214396+ 0.1) x (1 - 0.25 x -0.993517017828201x-0.993517017828201)=0.07532309838214396

Hasil tersebut dilakukan proses perhitungan dengan rumus sebagai berikut :

output n + offset = input [n+offset] x posisi

output[i + outputOffset] = (input [ke n+ inputOffset] x n + pMidBuffer ke n x itemp) / overlapLength

output = (-0.0054933317x0 x0.0 ke n x66) /66 =0.0

-0.0068527926

BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan membahas proses implementasi dari analisis dan perancangan yang telah dijelaskan pada Bab III. Implementasi merupakan sebuah proses untuk merubah semua rancangan ke dalam bahasa pemrogaman komputer. Fokus utama pembahasan pada bab IV ini adalah hasil dan pembahasan dari pembuatan dan langkah-langkah untuk membangun aplikasi konversi suara yang telah dilakukan pemrosesan dengan menggunakan algoritma *Waveform Simlirarity Base Overlape Add.*

4.1 Alat dan Bahan yang digunakan

Untuk melakukan implementasi dari analisis dan perancangan pada bab III diperlukan bahan-bahan sebagai berikut:

1. Kebutuhan *hardware* (Kebutuhan Perangkat Keras)

   Untuk mengembangkan aplikasi ini pada sebuah notebook dengan spesifikasi sebagai berikut:

   Processor : AMD A6-4455M APU with Radeon HD Graphics 2.10 GHz

   Memory : 2 GB

   Hardisk : 500 GB

   Operating System : Windows 8

2. Kebutuhan *Software* (Kebutuhan Perangkat Lunak)

   Untuk menembangkan aplikasi dengan menggunakan bahasa pemrogaman java dan untuk memudahkan proses pembuatan aplikasi menggunakan IDE Netbeans 8.0 dan JDK 1.7. untuk memperindah layout menggunakan editor gambar photoshop dan corel draw. Sedangkan untuk melakukan dokumentasi dan pembuatan laporan menggunakan software Microsoft office 2013.

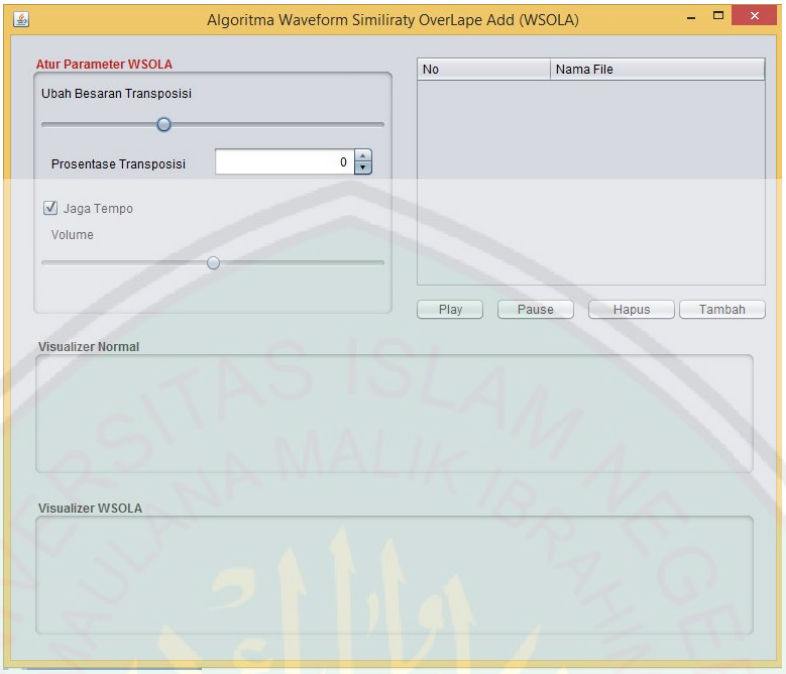
3. Kebutuhan Data

   Data yang digunakan untuk penelitian ini adalah data file audio yang mempunyai format .wav, data tersebut didapatkan dari internet dan beberapa data membuat sendiri dengan melakukan proses perekaman suara dan menyimpanya ke format .wav.

4.2 Hasil Implementasi Desain Sistem

Gambar 4.1 adalah hasil dari implementasi desain sistem yang telah dibuat pada BAB III. Masing-masing fungsi menu berfungsi sebagai berikut:

a. Ubah Besaran Konversi, slider tersebut berfungsi untuk melakukan besaran perubahan prosentase konversi. Jika pada prosentase menunjukkan angka 1, maka aplikasi tidak akan melakukan konversi suara. Jika aplikasi bernilai <1 atau bernilai negatif, maka aplikasi akan melakukan konversi suara ke nada yang rendah. Sebaliknya jika bernilai > 1, atau bernilai positif maka aplikasi akan melakukan konversi suara ke nada yang tinggi. Semakin tinggi prosentase tranposisi maka akan semakin dirubah ke nada yang lebih tinggi.



Gambar 4.1 Aplikasi Konversi Suara

b. Ubah Besaran Konversi, slider tersebut berfungsi untuk melakukan besaran perubahan prosentase konversi. Jika pada prosesntase menunjukkan angka 1, maka aplikasi tidak akan melakukan konversi suara. Jika aplikasi bernilai <1 atau bernilai negatif, maka aplikasi akan melakukan konversi suara ke nada yang rendah. Sebaliknya jika bernilai > 1, atau bernilai positif maka aplikasi akan melakukan konversi suara ke nada yang tinggi. Semakin tinggi prosentase tranposisi maka akan semakin dirubah ke nada yang lebih tinggi.

c. Jaga Tempo

Konversi suara dengan menggunakan *Waveform Simlirarity Base Overlape Add*, akan merubah tempo suara. Hal tersebut dikarenakan adanya proses *time sretching* atau pemampatan waktu sehingga ketika besaran konversi bernilai positif maka suara akan berjalan lebih cepat daripada suara aslinya. *Checkbox*tersebut untuk memberikan pilihan kepada user untuk menjaga

tempo suara atau membiarkan suara berjalan lebih cepat atau lebih lambat sesuai dengan besaran prosentase konversi.

d. Play

Untuk memainkan file suara yang telah ada di *grid* suara. Secara default button play akan memainkan suara yang ada pada *grid* paling atas.

e. Hapus

Berfungsi untuk menghapus file suara yang telah ditambahkan ke *grid.* Untuk menghapus file, terlebih dahulu harus mengklik nama filenya pada *grid*, kemudian megklik *button* hapus.

f. Tambah

Untuk menambahkan file suara ke data *grid,* file yang bisa ditambahkan hanyalah file yang mempunyai extensi .waf.

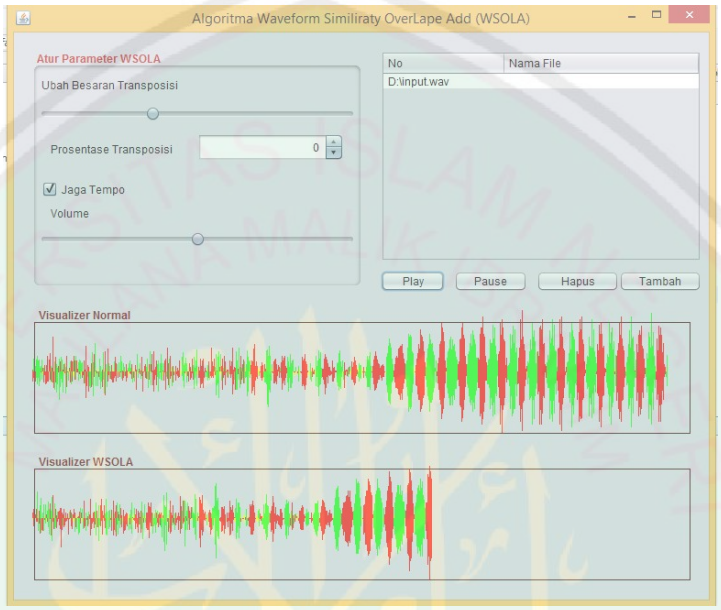
g. Visualizer Normal

Visualizer Normal memberikan gambaran waveform suara ketika belum dilakukan proses konversi suara dengan *Waveform Simlirarity Based Overlape Add.*

h. Visualizer WSOLA

Visualizer WSOLA memberikan gambaran waveform suara saat sudah dilakukan proses konversi suara dengan *Waveform Simlirarity Based Overlape Add.*

Pada gambar 4.2 bisa membandingkan waveform visualizer saat belum dilakukan proses WSOLA dan waveform saat sudah dilakukan proses konversi suara dengan *Waveform Simlirarity Based Overlape Add.*



Gambar 4.2 Perbandingan Visualizer Waveform

Pada gambar 4.2 terlihat bahwa visualizer normal mempunyai waveform yang lebih panjang jika dibandingkan dengan waveform suara yang sudah terproses dengan *Waveform Similarity Based On Overlape Add.*

4.3 Hasil Impelementasi Algoritma*Waveform Similarity Based Overlape Add.*

Proses transpsosisi suara dimulai dengan menginputkan file audio ke grid data, sebagaimana pada gambar 4.1. proses penginputan file audio ke grid data diimplementasikan dengan menggunakan *source code*pada gambar 4.3.

```
JFileChooser fc = new JFileChooser();
// File chooser untuk menginputkan file audio
int returnVal = fc.showOpenDialog(null);
if (returnVal == JFileChooser.APPROVE_OPTION) {
            File songFile = fc.getSelectedFile();
Object[] data = {songFile.toString()};
defaultTableModel.addRow(data);
        }
```

Gambar 4.*3 Source Code* Input File Audio ke Grid Data

Untuk melakukan penginputan suara menggunakan class JFileChooser untuk *membrowse* direktori lokal, dan melakukan pemilihan file. File yang bisa ditambahkan adalah file audio dengan ekstensi file .wav. kemudian menambahkan url file tersebut ke table data grid agar bisa dimainkan dan diberi efek konversi.File audio yang telah ditambahkan ke grid data, diolah dan dimainkan dengan method startfile(), Metho startfile() sebagaimana pada gambar 4.4.

```
private void startFile(File inputFile, Mixer mixer) {
//methode untuk memainkan file audio
if (dispatcher != null) {
dispatcher.stop();
        } try {
      if (inputFile != null) { format =
      AudioSystem.getAudioFileFormat(inputFile).getFormat();
            } else {
format = new AudioFormat(44100, 16, 1, true, true);
            }
rateTransposer = new RateTransposer(currentFactor);
gain = new GainProcessor(1.0);
audioPlayer = new AudioPlayer(format);
sampleRate = format.getSampleRate();            if
(originalTempoCheckBox.getModel().isSelected() && inputFile
!= null) {  wsola = new
WaveformSimilarityBasedOverlapAdd(WaveformSimilarityBasedOve
rlapAdd.Parameters.musicDefaults(currentFactor,
sampleRate));
wsola.addAudioComponentEventListener(new
AudioComponentEventListener() {
            } else {wsola = new
WaveformSimilarityBasedOverlapAdd(WaveformSimilarityBasedOve
rlapAdd.Parameters.musicDefaults(1, sampleRate));
```

```
wsola.addAudioComponentEventListener(new
AudioComponentEventListener() {
if (inputFile == null) {
DataLine.Info dataLineInfo = new
DataLine.Info(TargetDataLine.class, format);
TargetDataLine line = (TargetDataLine)
mixer.getLine(dataLineInfo);
line.open(format, wsola.getInputBufferSize());
line.start();
final AudioInputStream stream = new AudioInputStream(line);
dispatcher = new AudioDispatcher(stream,
wsola.getInputBufferSize(), wsola.getOverlap());
            } else {
if (format.getChannels() != 1) {
dispatcher = AudioDispatcher.fromFile(inputFile,
wsola.getInputBufferSize() * format.getChannels(),
wsola.getOverlap() * format.getChannels());
dispatcher.addAudioProcessor(new
MultichannelToMono(format.getChannels(), true));
                } else {
dispatcher = AudioDispatcher.fromFile(inputFile,
wsola.getInputBufferSize(), wsola.getOverlap());
                }
            }
     wsola.setDispatcher(dispatcher);
dispatcher.addAudioProcessor(wsola);
dispatcher.addAudioProcessor(rateTransposer);
dispatcher.addAudioProcessor(gain);
dispatcher.addAudioProcessor(audioPlayer);
            Thread t = new Thread(dispatcher);
t.start();
        } catch (UnsupportedAudioFileException | IOException |
LineUnavailableException e) {
        }
    }
```

Gambar 4.4 *Sourcecode* Penerapan Class WSOLA

Pada method starfile() tersebut juga terdapat *source code* penerapan algoritma *Waveform Similarity Based Overlape Add,* penerapan fungsi untuk jaga tempo, fungsi untuk visualizer normal dan fungsi untuk visualizer WSOLA.

4.3 Hasil dan Pembahasan Penerapan Algoritma

Pada pemrogamanya *Waveform Similarity Based Overlape Add* diimplemetasikan di sebuah class bernama Waveform Similarity Based Overlap Add .java class tersbut dipanggil dan difungsikan di sebuah class utama yang diberi nama Mainform.java. pemanggilan class Waveform Similarity Based Overlap Add .java.

Penerapan algoritma *Waveform Similarity Based Overlape Add* juga diterapkan pada method startfile(),Untuk penerapan algoritma *Waveform Similarity Based Overlape Add* diperlukan sebuah variabel yang didefinisikan dengan *currentFactor*, *currentFactor* adalah besaran tranpsosisi suara. Secara default *currentFactor* konversi suara bernilai 1, yang berarti tidak ada efek konversi suara.

Untuk penerapan algoritma *Waveform similarity overlape add* akan dibahas *step by step* sebagaimana pada bab III pada subbab perancangan penerapan algoritma.

```
if (originalTempoCheckBox.getModel().isSelected() &&
inputFile != null) {

wsola = new
WaveformSimilarityBasedOverlapAdd(WaveformSimilarityBasedOve
rlapAdd.Parameters.musicDefaults(currentFactor,
sampleRate));wsola.addAudioComponentEventListener(new

else {

wsola = new
WaveformSimilarityBasedOverlapAdd(WaveformSimilarityBasedOve
rlapAdd.Parameters.musicDefaults(1, sampleRate));

}

//Penerapan algoritma Waveform Similarity Overlape Add pada
aplikasi
```

Gambar 4.5 *source code* penerapa algoritma *waveform similarity overlape add*

4.3.1 Pencarian Panjang Overlape

```
overlapLength = (int) ((params.getSampleRate() *
params.getOverlapMs())/1000);
```

Gambar 4.6 *Sourcecode*Pencarian Panjang Overlape

Panjang overlape dihasilkan dari perkalian sample rate dan overlape permilisecond. Sample rate adalah jumlah sample yang bisa diambil dalam satuan detik, pada penelitian ini menggunakan sample rate standart file audio yakni 44,1 kHz. Sedangkan overlape adalah seberapa panjang periode yang diperbolehkan dan terjadi overlape dalam satuan detik, overlape awal diinisialisai dan diberi value 3.

4.3.2 Pencarian Panjang Window

```
seekWindowLength = (int) ((params.getSampleRate() *
params.getSequenceMs())/1000);
```

Gambar 4.7 *Sourcecode* Pencarian Panjang Window

Panjang window merupakan hasil perkalian dari sample rate dan sequence permili seconds.Sequence merupakan panjang antrian dari pemrosesan tunggal. Panjang sequence juga mewakili byte suara normal yang diambil. Sedangkan panjang window dicari dalam satuan miliseconds dan merupakan pembatas algoritma waveform similarity based overlape add untuk menemukan posisi terbaik yang memungkinkan untuk overlape.

4.3.3 Pencarian Jumlah sample bit yang di skip

```
double nominalSkip = tempo * (seekWindowLength – o
verlapLength);
```

Gambar 4.8*Sourcecode* Pencarian Jumlah Sample Bit yang Diskip

Jumlah sample yang diskip merupakan hasil perkalian tempo dan panjang window dikurangi dengan panjang overlape.Sedangkan nilai variable tempo sesuai dengan nilai prosentase konversi suara. Nilai default variable tempo adalah 1, yang berarti tidak ada jumlah sample bit yang di skip.

4.3.4 Pencarian Sample Request

```
sampleReq = Math.max(intskip + overlapLength,
seekWindowLength) + seekLength;
```

Gambar 4.9 *Sourcecode* Pencarian Sample Req

Variable samplereq digunakan untuk menampung nilai sample yang di request, sample req dicari dengan menggunakan nilai tertinggi antara penjumlahan antara sample bit yang di skip dan panjang overlape dengan panjang window. Nilai tertinggi tersebut dijumlahkan dengan panjang window.

4.3.5 Pencarian Posisi Overlape Terbaik

Posisi overlape terbaik merupakan posisi dimana terjadi overlape sample byte antrian yang mempunyai nilai yang hanpir sama. Untuk melakukan pencarian posisi overlape terbaik harus melakukan proses *cross correction* terlebih dahulu.

```
double calcCrossCorr(float[] mixingPos, float[] compare, int
offset){
// method untuk menghitung cross correction
      double corr = 0;
      double norm = 0;
      for (int i = 1; i < overlapLength; i ++){
      corr += mixingPos[i] * compare[i + offset];
      norm += mixingPos[i] * mixingPos[i];
          }
      if (norm < 1e-8){
            norm = 1.0;
          }
      return corr / Math.pow(norm,0.5);
      }
```

Gambar 4.10 *Sourcecode* Perhitungan *Cross Correction*

Perhitungan cross correction dilakukan dengan cara membagi nilai hasil penjumlahan correction sebanyak nilai panjang overlape dengan pemangkatan dari hasil perkalian normalization.

```
private int seekBestOverlapPosition(float[] inputBuffer, int
postion) {

//method untuk mencari posisi overlape terbaik

            int bestOffset;double bestCorrelation,
      currentCorrelation;

            int tempOffset;int comparePosition;

            precalcCorrReferenceMono();

//untuk mempercepat proses perhitungan nilai cross correction

            bestCorrelation = -10;bestOffset = 0;

for (tempOffset = 0; tempOffset < seekLength; tempOffset++) {

                  comparePosition = postion + tempOffset;

                  currentCorrelation = (double)
calcCrossCorr(pRefMidBuffer, inputBuffer,comparePosition);

//perhitungan cross correction

      double tmp = (double) (2 * tempOffset - seekLength) /
seekLength;

currentCorrelation = ((currentCorrelation + 0.1) * (1.0 - 0.25 *
tmp * tmp));

                  if (currentCorrelation > bestCorrelation) {

                        bestCorrelation = currentCorrelation;

                        bestOffset = tempOffset;

                  }

            }

            return bestOffset;

      }
```

Gambar 4.11 *Sourcecode* Pencarian Best Overlape

Function pada gambar 4.11 digunakan untuk melakukan penacarian posisi overlape terbaik. Untuk mencari posisi overlape terbaik harus melakukan

perhitungan cross correction. Overlape terbaik dihasilkan dari nilai yang paling tinggi.

```
private void overlap(final float[] output, int
outputOffset, float[] input,int inputOffset){

// method untuk mencari overlape

int i=0;

for(i = 0 ; i < overlapLength ; i++){

     int itemp = overlapLength - i;

     output[i + outputOffset] = (input[i + inputOffset] * i
+ pMidBuffer[i] * itemp ) / overlapLength;

          }

      }
```

Gambar 4.12 *Sourcecode* Overlape

Function overlape digunakan untuk mencari ouput dari pengolahan waveform dengan algoritma Waveform Similarity Based On Overlape Add. Output dihasilkan dari penjumlahan input, buffer pada posisi tengah, temporary waveform dibagi dengan panjang overlape.

4.3.7 Pemrosesan Audio bit sample

Semua function yang sudah dijelaskan, dipanggil dalam satu function, yakni function `process()`. Function `process` merupakan function utama untuk mengolah input dari file audio byte dengan menggunakan algoritma Waveform Similarity Based On Overlape Add. Function ini merupakan implement dari class interface AudioProcessor.java. Difunction tersebut sample byte audio dirubah kedalam bentuk type data float dan diolah dengan WSOLA. Kemudian hasil dari

pengolahan algoritma WSOLA yang berupa type data float dikembalikan lagi ke type data byte agar bisa dimainkan dengan menggunakan java sound.

```
public boolean process(AudioEvent audioEvent) {

//method untuk pemrosesan audio suara.

audioFloatBuffer = audioEvent.getFloatBuffer();

assert audioFloatBuffer.length == getInputBufferSize();

int offset =  seekBestOverlapPosition(audioFloatBuffer,0);

overlap(outputFloatBuffer,0,audioFloatBuffer,offset);

int sequenceLength = seekWindowLength - 2 * overlapLength;

System.arraycopy(audioFloatBuffer, offset + overlapLength,
outputFloatBuffer, overlapLength, sequenceLength);

System.arraycopy(audioFloatBuffer, offset + sequenceLength +
overlapLength, pMidBuffer, 0, overlapLength);

assert outputFloatBuffer.length == getOutputBufferSize();

audioEvent.setFloatBuffer(outputFloatBuffer);

audioEvent.setOverlap(0);

for (AudioComponentEventListener listener :
audioComponentEventListeners.getListeners(AudioComponentEventListe
ner.class)) {

listener.sampleChanged(audioFloatBuffer);
listener.wsola_sampleChanged(outputFloatBuffer);

}if(newParameters!=null){

applyNewParameters();dispatcher.setStepSizeAndOverlap(getInputBuff
erSize(),getOverlap());

            }

            return true;

}
```

Gambar 4.13 Function Utama untuk Melakukan Proses Pengolahan Audio Byte Sample

4.4 Uji Coba

Uji coba aplikasi dilakukan untuk mengetahui seberapa besar keberhasilan algoritma *Waveform similarity Based Overlape Add*. Uji coba dilakukan dengan menggunakan dengan menggunakan spesifikasi komputer sebagai berikut:

Processor : AMD A6-4455M APU with Radeon HD Graphics 2.10 GHz

Memory : 2 GB

Hardisk : 500 GB

Operating System : Windows 8

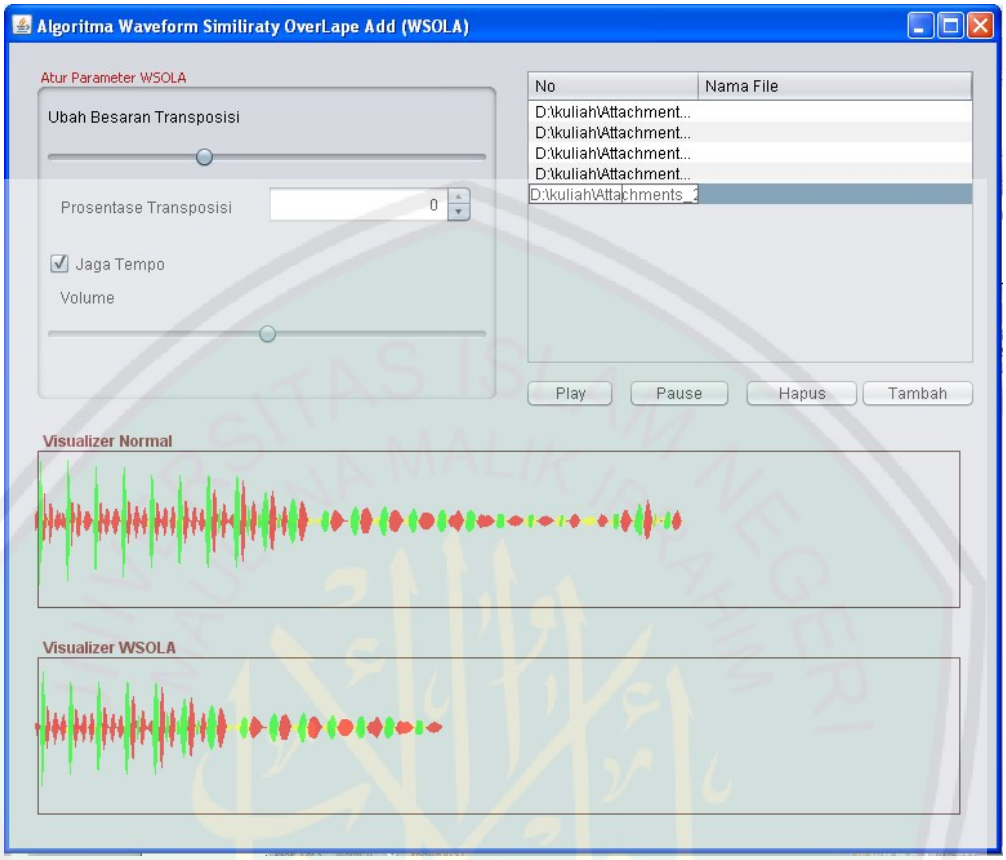
Uji coba meliputi uji coba keberhasilan implementasi algoritma *Waveform Similarity Based Overlape Add* untuk melakukan proses tranposisi suara. Ada beberapa file .wav yang digunakan. Sumber data untuk melakukan uji coba (file audio .wav) beberapa diunduh dari internet dan beberapa data dibuat sendiri dengan melakukan rekaman suara kemudian menyimpanya kedalam hardisk dengan format .wav.

Skenario uji coba ini dengan memilih file yang mempunyai ekstensi .wav untuk dimainkan diaplikasi, kemudian diberi efek konversi suara, jika terjadi perubahan suara maka aplikasi dianggap berhasil melakukan konversi suara akan tetapi jika dilakukan efek konversi dan tidak terjadi perubahan suara maka aplikasi dianggap tidak berhasil melakukan konversi suara.

Keberhasilan uji coba dihitung dengan menggunakan rumus sebagai berikut :

$$\text{Prosentase Keberhasilan} = \frac{\qquad}{\qquad} \quad 100\%$$

$$\text{Prosentase Kegagalan} = \frac{\qquad}{\qquad} \quad 100\%$$



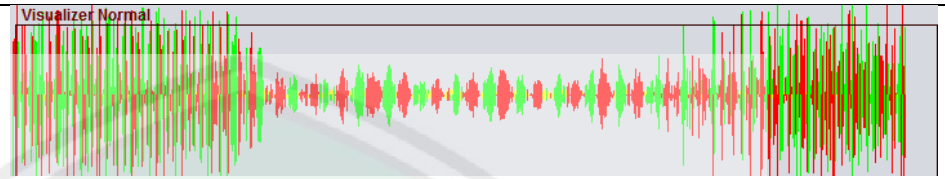
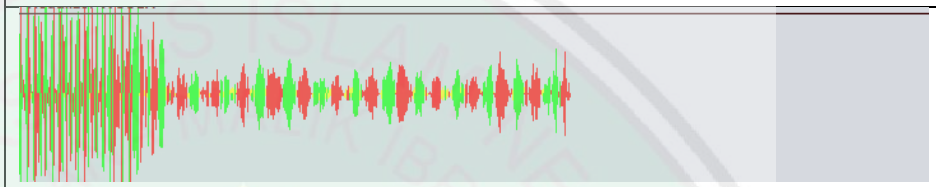
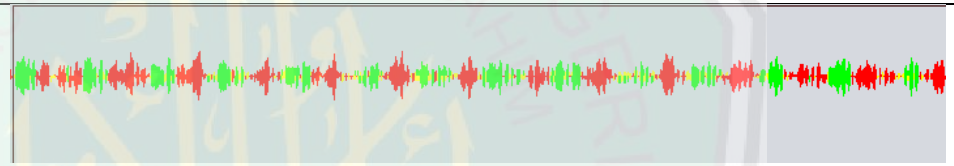
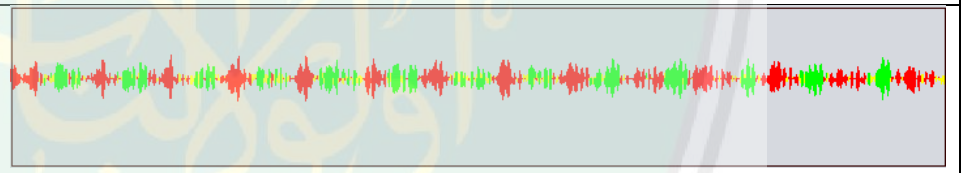
Gambar 4.14 Tampilan Aplikasi Ketika Uji Coba

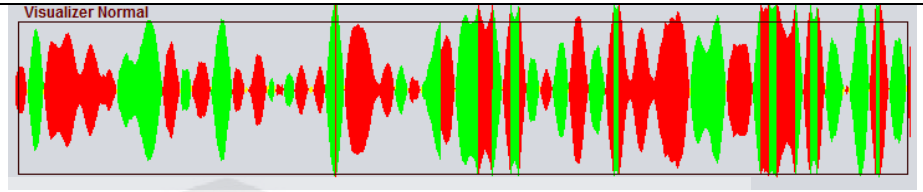
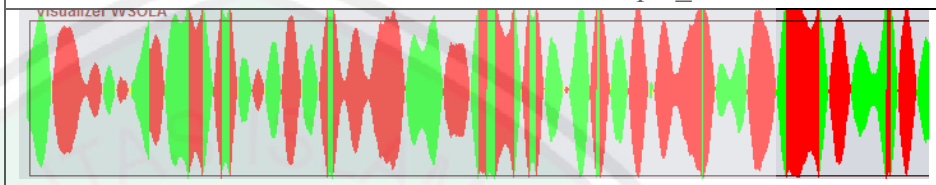
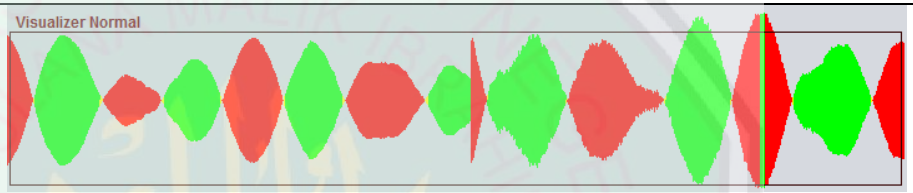
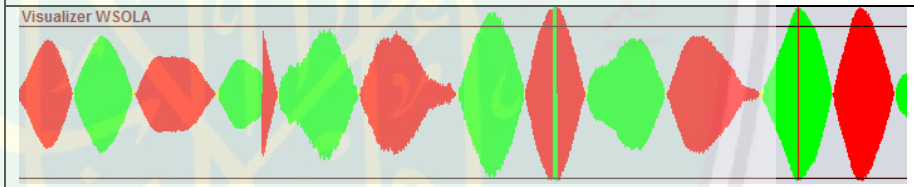
Uji coba dilakukan pada hari senin, tanggal 01 maret 2015.Gambar 4.14 adalah contoh tampilan aplikasi ketika dilakukan uji coba. Hasil dari uji sebagaimana pada table 4.1. Hasil uji coba berdasarkan 10 kali uji coba dengan berbagai format file audio adalah sebagai berikut:

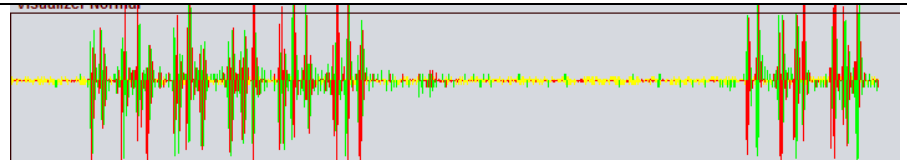
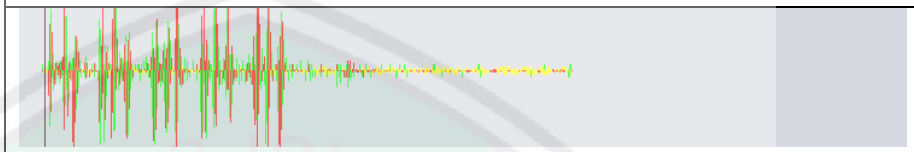
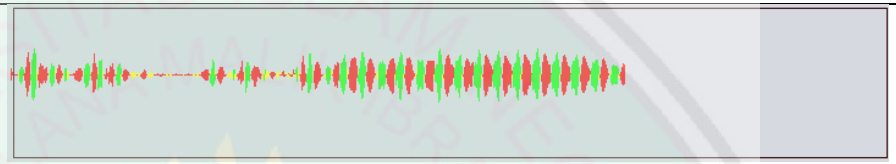
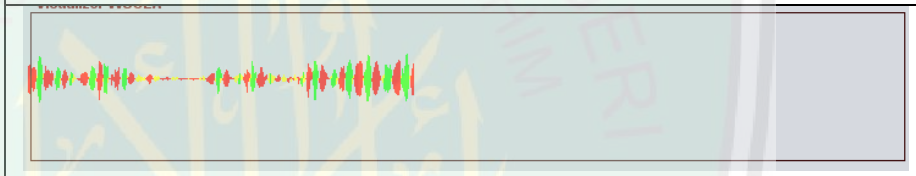
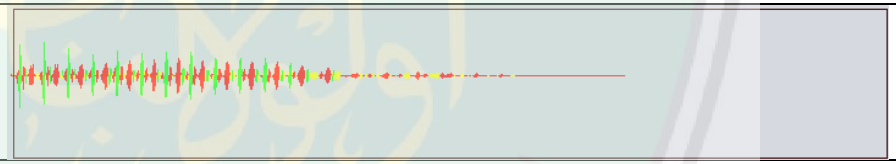
Prosentase Keberhasilan = — 100= 90%

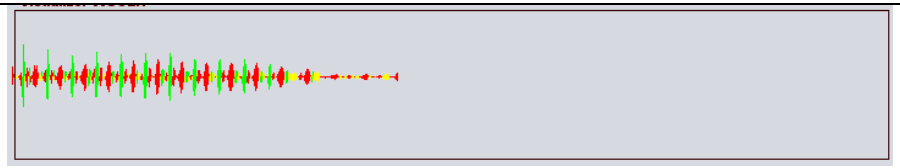
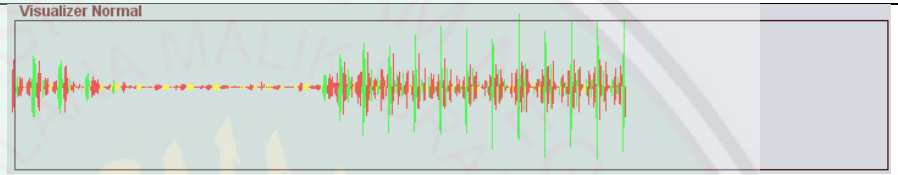
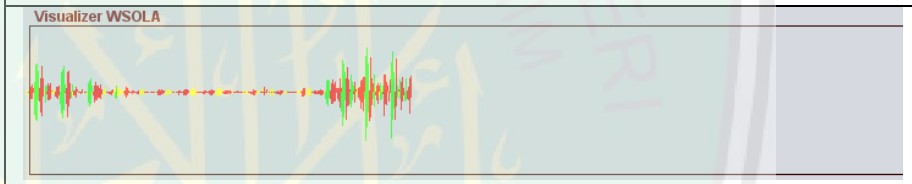
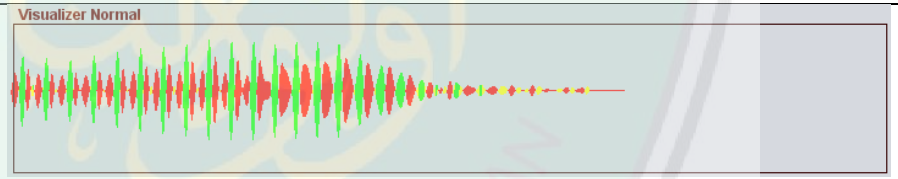
Prosentase Keberhasilan = — 100= 10%

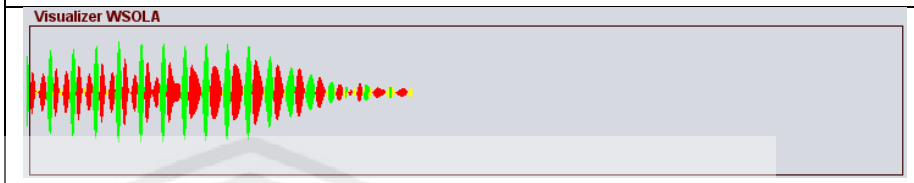
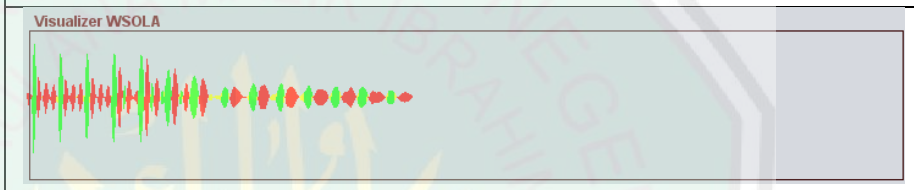
Tabel 4.1 Tabel Hasil Uji Coba 1

No	Nama File	Visualizer	Hasil Uji Coba
1	Sample_01.Wav	 Visualizer Normal File Audio Sample_01.wav  Visualizer WSOLA File Audio Sample_01.wav	Aplikasi mampu melakukan konversi suara. Ketika melakukan konversi suara up dan konversi suara down menghasilkan suara yang jernih. Terhadap perbedaan visualizer normal dan visualizer WSOLA.
2	Sample_02.Wav	 Visualizer Normal File Audio Sample _02.wav  Visualizer WSOLA File Audio Sample_02.wav	Aplikasi mampu melakukan konversi suara, suara yang dihasilkan setelah konversi mempunyai suara yang jernih. Akan tetapi, ketika dilakukan konversi dengan parameter <100 file audio macet.

3	Sample_03.wav	 Visualizer Normal File Audio Sample_03.wav  Visualizer WSOLA File Audio Sample_03.wav	Aplikasi mampu melakukan konversi dengan parameter < 100 dengan hasil suara yang kurang jernih.
4	Sample_04.wav	 Visualizer Normal File Audio Sample_04.wav  Visualizer WSOLA File Audio Sample_04.wav	Aplikasi mampu memutar audio file, suara yang dihasilkan tidak lancar, ketika dilakukan efek konversi dengan parameter > 100 aplikasi macet dan suara audio berhenti.

5	Sample_05.wav	 Visualizer Normal File Audio Sample_05.wav  VisualizerWSOLA File Audio Sample_05.wav	Aplikasi mampu melakukan konversi suara dengan parameter >100 dengan hasil suara yang jernih.
6	Sample_06.wav	 Visualizer Normal File Audio Sample_06.wav  Visualizer WSOLA File Audio Sample_06.wav	Aplikasi mampu melakukan konversi suara dengan parameter > 100 dengan hasil suara yang jernih.
7	Sample_07.wav		Aplikasi mampu memutar audio file, suara yang dihasilkan tidak lancar, ketika dilakukan efek konversi suara dengan

		 Visualizer Normal File Audio Sample_07.wav  Visualizer WSOLA File Audio Sample_07.wav	parameter < 100 aplikasi macet dan suara audio berhenti.
8	Sample_08.WAV	 Visualizer Normal File Sample_08.wav  Visualizer WSOLA File Sample_08.wav	Aplikasi mampu melakukan konversi suara dengan parameter < 100 dengan hasil suara yang jernih.
9	Sample_09.wav		Aplikasi mampu melakukan konversi suara dengan parameter < 100 dengan hasil suara yang jernih.

		Visualizer Normal File Sample_09.wav  Visualizer WSOLA File Audio Sample_09.wav	
10	Sample_10.wav	 Visualizer Normal File Audio Sample_10.wav  Visualizer WSOLA File Audio Sample_10.wav	Aplikasi mampu memutar audio file, suara yang dihasilkan tidak lancar, ketika dilakukan efek konversi suara dengan parameter < 100 aplikasi macet dan suara audio berhenti.

Tabel 4.2 Tabel Uji Coba 2

No	Nama File	Durasi	Parameter 80	Parameter 120	Parameter 160	Parameter 200	Parameter 240	Parameter 280	Parameter 320
1	Sample_01.wav	00:01:43	Dikenali	Dikenali	Dikenali	Tidak dikenali	Tidak Dikenali	Tidak dikenali	Tidak dikenali
2	Sample_02.wav	00:02:05	Dikenali	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
3	Sample_03.wav	00:01:34	Dikenali	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
4	Sample_04.wav	00:01:39	Dikenali	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
5	Sample_05.wav	00:01:43	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali
6	Sample_06.wav	00:01:48	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali
7	Sample_07.wav	00:01:22	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
8	Sample_08.wav	00:01:26	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
9	Sample_09.wav	00:02:01	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali
10	Sample_10.wav	00:01:40	Dikenali	Dikenali	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
Prosentase Keberhasilan			100%	80%	60%	40%	20%	10%	0%

Tabel 4.3 Tabel Uji Coba 3

No.	Nama File	Durasi	Parameter -80	Parameter -120	Parameter -160	Parameter -200
1	Sample_01.wav	00:01:43	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
2	Sample_02.wav	00:02:05	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
3	Sample_03.wav	00:01:34	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
4	Sample_04.wav	00:01:39	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
5	Sample_05.wav	00:01:43	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
6	Sample_06.wav	00:01:48	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
7	Sample_07.wav	00:01:22	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
8	Sample_08.wav	00:01:26	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
9	Sample_09.wav	00:02:01	Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
10	Sample_10.wav	00:01:40	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali	Tidak Dikenali
Prosentasi Keberhasilan			20%	15%	10%	0%

Dari prosentase keberhasilan tersebut algoritma *Waveform Similarity Based Overlape Add* mampu melakukan konversi suara digital, dapat dilihat pada table 4.2 dapat diketahui bahwa file audio yang mengunakan parameter 320 pendengar tidak dapat mengenali suara asli dengan presentase 0%. Dan pada table 4.3 dapat diketahui bahwa file audio yang menggunakan parameter -200 pendengar pun juga tidak dapat mendengar suara asli dengan parameter 0 % dan masih mampu melakukan proses konversi suara dengan suara yang jernih dan jelas didengar oleh telinga.

4.5Kajian Islam

Inti kegunaan aplikasi dalam penelitian ini adalah melakukan perubahan frekuensi suara agar tidak bisa dikenali.Salah satu kegunaanya adalah untuk melakukan perubahan suara pada sumber berita atau saksi dalam sebuah kasus persidangan. Baik, narasumber atau saksi mempunyai kewajiban yang sama yakni menyampaikan sesuatu kebenaran. Akan tetapi tidak semua kebenaran itu bisa diterima oleh orang lain.

Menurut etimologi (bahasa) kata saksi dalam bahasa arab dikenal dengan Asy-syahadah (الشهاةد) adalah bentuk isim masdar dari kata يشهد - شهد (syahida-yasyhadu) yang artinya menghadiri, menyaksikan (dengan mata kepala sendiri) dan mengetahui. Kata syahadah juga bermakna al-bayinan (bukti), yamin (sumpah) dan iqrar (pengakuan). (Moenawir, 2002: 746-747).

Secara terminologi (istilah).Al-Jauhari menyatakan bahwa “kesaksian berarti berita pasti. Musyahadah artinya sesuatu yang nyata, karena saksi adalah orang yang menyaksikan sesuatu yang orang lain tidak mengetahuinya. Dikatakan

juga bahwa kesaksian berarti seseorang yang memberitahukan secara benar atas apa yang dilihat dan didengarnya (Ihsanudin dkk : 2005: 94)

Dalam kamus Istilah fiqih, ”Saksi adalah orang atau orang-orang yang mengemukakan keterangan untuk menetapkan hak atas orang lain. Dalam pengadilan, pembuktian dengan saksi adalah penting sekali, apalagi ada kebiasaan di dalam masyarakat bahwa perbuatan-perbuatan hukum yang dilakukan itu tidak dicatat. (Mujib dkk, 1994: 306)

Adapun hukum kesaksian itu adalah fardhu ain bagi orang yang memikul nya bila dia dipanggil untuk itu dan dikhawatirkan kebenaran akan hilang; bahkan wajib apabila dikhawatirkan lenyapnya kebenaran meskipun dia tidak dipanggil untuk itu. (Sayyid, 1887 : 86) karena Allah Ta’ala berfirman:

وَلَا تَكْتُمُوا۟ ٱلشَّهَـٰدَةَ ۚ وَمَن يَكْتُمْهَا فَإِنَّهُۥٓ ءَاثِمٌ قَلْبُهُۥ ۗ وَٱللَّهُ بِمَا تَعْمَلُونَ عَلِيمٌ

﴿٢٨٣﴾

Artinya:
“….Janganlah kamu (para saksi) menyembunyikan persaksian; dan barang siapa menyembunyikannya, maka ia adalah orang yang berdosa hatinya, Dan Allah maha mengetahui (Q.S Al-Baqoroh 283). (Depag RI, 2007 : 71).

Karena demikian pentingnya proses persaksian atau sumber berita, maka saksi anonim atau narasumber anonim diperbolehkan dengan syarat-syarat tertentu. Saksi anonim diperbolehkan jika keselamatan sumber tersebut terancam bila identitasnya dibuka. Unsur "keselamatan" tersebut tentunya yang bisa diterima oleh akal sehat, nyawanya yang benar-benar terancam atau nyawa anggota keluarga langsungnya yang terancam (anak, istri, suami, orang tua, saudara kandung).

Dalam dunia reportase jurnalistik, saksi anonim sudal dilindungi oleh Pasal 4 ayat (4) UU Pers serta pasal 170 KUHAP Khususnya pasal 4 ayat 4 UU Pers yang mengatur soal hak tolak wartawan. Hak tolak wartawan, dalam penjelasan pasal tersebut dimaksudkan sebagai hak tolak untuk tidak mengungkapkan identitas narasumber dalam pemberitaan. Narasumber disini artinya narasumber anonim, narasumber yang identitasnya sengaja disembunyikan. Artinya, wartawan hanya mempunyai hak tolak untuk menyebutkan identitas narasumber anonim dalam proses peradilan. Entah sebagai saksi di tingkat penyelidikan, penyidikan hingga penuntutan.Hak tolak hanya gugur lewat putusan pengadilan, dengan alasan kepentingan dan ketertiban umum.Seperti biasa, kepentingan dan ketertiban umum ini tidak pernah mempunyai definisi yang jelas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil uji coba dan pembuatan aplikasi konversi suara dengan menggunakan algoritma algoritma *Waveform Similarity Based Overlape Add* maka dapat disimpulkan bahwa:

1. Algoritma *Waveform Similarity Based Overlape Add* mampu melakukan konversi suara dengan tingkat keberhasilan 90% dengan menggunakan parameter 88 – 320 dan -88 – (-200). Dan terjadi perbedaan panjang frame audio antara frame audio yang sudah dilakukan konversi suara dengan frame audio yang belum dilakukan proses konversi suara. Panjang frame yang sudah dilakukan proses konversi hanya setengah dari panjang frame asli.

1.2 Saran

Untuk perbaikan penelitian terkait konversi suara, maka penulis memberikan saran sebagai berikut:

1. Menggunakan teknologi pemrogaman yang mendukung untuk memainkan semua jenis format audio suara. Karena format .wav jarang sekali dipakai dalam kehidupan sehari-hari.
2. Melakukan perbandingan keberhasilan konversi suara dengan dengan algoritma lain, seperti algoritma Fast Fourier Transform dan Wavelet.
3. Membuat aplikasi yang mampu melakukan peoses konversi secara *realtime,* bukan memainkan audio file yang tersimpan di komputer.

DAFTAR PUSTAKA

Akbar, dkk (2011). "Konversi Nada-Nada Akustik Menjadi Chord Menggunakan Pitch Class Profile". ITS. Surabaya.

Alfiansyah, Yusri (2008). "Pengenalan Chord Otomatis Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization". Universitas Brawijaya. Malang.

Arikunto, S. 2010. Prosedur penelitian : Suatu Pendekatan Praktik. (Edisi

Bachtiar, Irfan Syafur. 2007. Aplikasi Pengenalan Wi-cara HMM untuk Kendali Robot PDA. Surabaya: Po-liteknik Elektronika Negeri Surabaya.

Bhaskoro, Susetyo Bagas dan W. D, Altedzar Riedho.(2012).Aplikasi Pengenalan Gender Menggunakan Suara.Seminar Nasional Aplikasi Teknologi Informasi 2012.

Binanto, Iwan (2010). Multimedia Digital – Dasar Teori dan Pengembangannya. Yogyakarta : Andi.

Demol, Mike*, Verhelst, Werner*, Struyve, Kris**, Verhoeve, Piet**. ***Efficient Non-Uniform Time-Scaling of Speech with WSOLA.*** *Laboratory for Speech and Audio Processing, dept. ETRO-DSSP, Interdisciplinary Institute for Broadband Technology, Vrije Universiteit Brussel, Belgium **Central R&D Department, TELEVIC nv, Belgium.

Driedger, Jonathan, MeinardM¨ uller, Member, IEEE, dan Sebastian, Ewert. ***Improving Time – Scale Modification of Music Signals using Harmonic – Percussive Separatio***. JOURNAL OF L AT EXCLASS FILES, VOL. 11, NO. 4, DECEMBER 2012

Driedger, Jonathan. 3, November 2011. **Time-ScaleModificationAlgorithms forMusicAudioSignals**

Grofit, Shahaf, Student Member, IEEE, dan Lavner, Yizhar, Member, IEEE. ***Time – Scale Modification of Audio Signals Using Enhanced WSOLA With Management of Transients***. IEEE TRANSAXTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING

Ibnu Daqiqil, Aryanto, Uchi Dayana. Konversi Nada Menjadi Chord Menggunakan Metode Pitch Class Profile PadaInstrumen Tunggal. Fakultas Ilmu Komputer Universitas Muhammadiyah Riau.

ICASSP-85, pp. 236-239, 1985.

Ihsanudin, Muhammad Najih, Sri Hidayati (*eds*), *Op. Cit*.

Kuc, R., Introduction to Digital Signal Processing, McGraw-Hill Book Company, Singapore, 1998

M. Abdul Mujieb, Mabruru Tholhah dan Syafi'ah (*eds*), *Kamus Istilah Fiqih,* Jakarta : PT. Pustaka Firdaus, 1994.

Nawawi, Imam dan Imam Ibnu Rajab. 2005. Arba'un Nawawiyah wa ziyadatu Ibni Rajab 'alaiha. Sukoharjo: Maktabah Al-Ghuroba.

Pamungkas, Adi Jarot. (2008). Dasar & Aplikasi Musik Digital. Yogyakarta: Andi Offset.

Purcell et al. 2010. *Kalkulus Edisi Kesembilan Jilid 1*. Jakarta : Erlangga

Revisi). Jakarta : Rineka Cipta

S. Roucos, A. Wilgus, '***High quality Time-Scale Modification of Speech***',

S.Christian, Klapuri. Ansi. 2010. Constant-Q Transform Toolbox For Music Processing. University of Music and Performing Arts.

Sa'diyah, Halimatus. 2008. ***Penerapan Fungsi Transposisi Akord Pada Perpindahan Tangga.*** Jurusan Matematika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

Sayyid Sabiq, Fikih Sunnah-14, Bandung : Alma'arif, 1987, Cet. ke-1.

Sinyal.Yogyakarta:Andi Publisher.

Stewart, James. 2011. *Kalkulus Edisi 5 Buku 3. Jakarta* : Salemba Teknika.

Subiyakto, Adriyakti. 2011. Melakukan penelitian dengan judul ***Speech Synthesizer Berbasis Diphone Menggunakan Algoritma Waveform Similarity Overlap-Add (WSOLA).***

Sugiyono (2010). *Metode Penelitian Kuantitatif Kualitatif & RND*. Bandung : Alfabeta.

Syarah Arbain An-Nawawi plus 8 Hadits Ibnu Rajab; Abdul Muhsin Al-Badr, Jakarta; Darul Ilmi (2005).

Syarah Arbain An-Nawawi; Imam Nawawi, et al; Jakarta; Darul Haq (2006) ISBN 979-3407-76-X

Systems', ICASSP-91, pp. 501-504, 1991.

Tanudjaja, Harlianto.(2008).Pengolahan Sinyal Digital & Sistem Pemrosesan

VERHELST , Wernerand dan ROELANDS, Marc. ***AN OVERLAP-ADD TECHNIQUE BASED ON WAVEFORM SIMILARITY (WSOLA) FOR HIGH QUALITY TIME-SCALE MODIFICATION OF SPEECH***

W. Verhelst, '***On the Quality of Speech Produced by Impulse Driven Linear***

Warson Moenawir, Al-Munawir, Kamus Arab - Indonesia. Surabaya : Pustaka Progresif, 2002, Cet. Ke-25.