TRANSPOSISI NADA LAGU MENGGUNAKAN ALGORITMA *FAST FOURIER TRANSFORM*

SKRIPSI

Oleh

LULUK FARKHIAH
NIM. 08650089

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

TRANSPOSISI NADA LAGU MENGGUNAKAN ALGORITMA *FAST FOURIER TRANSFORM*

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)

O l e h

LULUK FARKHIAH
NIM. 08650089

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

HALAMAN PERSETUJUAN

TRANSPOSISI NADA LAGU MENGGUNAKAN ALGORITMA *FAST FOURIER TRANSFORM*

SKRIPSI

Oleh

LULUK FARKHIAH
NIM: 08650089

Telah Disetujui,

Malang, 10 Maret 2015

Dosen Pembimbing I	Dosen Pembimbing II
Totok Chamidy, M.Kom	**Dr. M. Faisal, MT**
NIP.196912222 00604 100 1	**NIP.19740502 00501 100 7**

Mengetahui:

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP.19740424 200901 1 008

HALAMAN PENGESAHAN

TRANSPOSISI NADA LAGU MENGGUNAKAN ALGORITMA *FAST FOURIER TRANSFORM*

S K R I P S I

Oleh :

Luluk Farkhiah
NIM. 08650089

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	:	Luluk Farkhiah
NIM	:	08650089
Fakultas/Jurusan	:	Sains dan Teknologi/Teknik Informatika
Judul Penelitian	:	Transposisi Nada Lagu Menggunakan Algoritma *Fast Fourier Transform*

Menyatakan dengan sebenar-benarnya bahwa skripsi yang saya buat tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiyah 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,

Luluk Farkhiah

NIM: 08650089

MOTO

KESABARAN ITU MENOLONG
SEGALA PEKERJAAN TIADA
KENIKMATAN KECUALI
SETELAH KEPAYAHAN

H A L A M A N P E R S E M B A H A N

Ya allah ya robb,

sepercik yang telah engkau karuniakan kepadaku. Hanya puji syukur yang dapat ku persembahkan kepadaMu. Akhirnya tercapai jua apa yang selama ini menjadi angan dan impianku. Doa serta airmata yang mengiringi, ini semua tidak terlepas dari semua pihak yang telah membantu dalam mewujudkannya. Dan syukur Alhamdulillah karya (skripsi) ini bisa berjalan dengan lancer seiring dengan jalannya air sungai yang mengalir dalam setiap kehidupan.

Karya ini saya persembahkan untuk Aba dan Umi yang dengan penuh cinta dan kasih sayang selalu memberikan yang terbaik buat putra-putrinya. Karya ini merupakan wujud dari doa dan sujud beliau. Serta saya persembahkan untuk kakakku Evi,adikku Bela,ponaanku Nayla serta suami dan anakku Iqbal yang selalu memberiku semangat serta motifasi saya

Seluruh keluarga besarku yang mana selama ini selalu memberikan dukungan dan perhatian.

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

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Alhamdulillah, segala puji dan syukur dengan tulus kami persembahkan ke hadirat Allah SWT, karena hanya dengan petunjuk dan hidayah-Nya peneliti mampu menyelesaikan skripsi yang berjudul *"*TRANSPOSISI NADA LAGU MENGGUNAKAN ALGORITMA *FAST FOURIER TRANSFORM"*

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

Penyelesaian skripsi ini merupakan suatu pekerjaan sangat berat bagi peneliti yang fakir ilmu, namun berkat ma'unnah Allah SWT dan bantuan dari berbagai pihak baik berupa materiil maupun moril, akhirnya skripsi ini dapat terselesaikan dengan baik. Oleh karena itu peneliti menyampaikan rasa hormat, ungkapkan terima kasih serta penghargaan setinggi-tingginya kepada:

1. Totok Chamidy, M.T selaku pembimbing I dan yang dengan sabar memberikan arahan, saran dan motivasi pada peneliti sehingga skripsi ini dapat terselesaikan dengan baik.
2. Mohammad Faisal, M.T selaku pembimbing II serta Dosen wali yang selalu menuntun sehingga skripsi ini dapat terselesaikan dengan baik.
3. Dr. Cahyo Crysdian selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Seluruh Dosen yang telah mengajarkan banyak hal dan selalu memberikan semangat untuk terus berproses hingga akhir perkuliahan peneliti.

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

Penulis berharap semoga skripsi ini bisa dibaca oleh banyak orang, terutama civitas akademika Universtias Islam Negeri Maulana Malik Ibrahim Malang. Selain itu peneliti berharap semoga skripsi ini dapat memberikan nilai guna baik bagi peneliti maupun bagi pembaca. Amin Ya Robbal'Alamin

Malang,……..2015

Peneliti,

Luluk Farkhiah

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Farkhiah, Luluk. 2015, Transposisi Nada Lagu Menggunakan Algoritma Fast Fourier Transform (FFT). Skripsi, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing : Totok Chamidy, M. Kom
Kata Kunci : Transposisi Suara, Fast Fourier Transform, Sinyal.

Di Dunia pengolahan suara digital telah memungkinkan untuk merubah range suara, semisal merubah range suara yang asalanya bass menjadi range suara tenor. Salah satu algoritma yang memungkinkan untuk melakukan proses perubahan range suara digital tersebut adalah Algoritma *Fast Fourier Transform (*FFT*)*. FFT merubah sinyal analog menjadi sinyal diskrit dalam domain waktu, kemudian merubahnya ke domain frekuensi. Proses pengolahan sinyalnya dengan cara membagi sinyal menjadi beberapa bagian yang kemudian masing-maing bagian diselesaikan dengan algoritma yang sama dan hasilnya dikumpulkan kembali.

Sebelum dilakukan proses pengolahan dengan FFT, sinyal dari file audio yang telah disampling dilakukan proses *frame blocking,* yakni membagi sinyal menjadi beberapa frame. Frame blocking menghasikan sinyal discontinue sehingga perlu dilakukan proses windowing dengan teknik *hamming window*.

Dari hasil penelitian dan uji coba yang telah dilakukan dengan berbagai format file audio dapat disimpulkan bahwa algoritma *Fast Fourier Transform* mampu melakukan proses transposisi suara dengan tingkat keberhasilan 60% uji coba parameter yang telah dilakuka kombinasi terbaik adalah oversampling 8 dan FFT Framesize sebesar 1024.

ABSTRACT

Farkhiah, Luluk. 2015, Transposition of Song Tone Using Fast Fourier Transform (FFT) Algorithm. Thesis, Faculty of Science and Technology, Islamic State University, Malang.
Advisor: Totok Chamidy, M. Kom
Keywords: Voice Transposition, Fast Fourier Transform, Signal.

The term of digital voice processing has been possible to change the voice range, for example, changing a digital voice which the origin is bass becomes tenor range voice. One of algorithms which may be possible to do a process of digital voice range changing is *Fast Fourier Transform (*FFT*)* Algorithm. FFT changes an analogue signal to be a discrete signal in a time domain, thus, changing it to be a frequent domain. The process of its signal processing thereby dividing a signal to be some parts, then, each part is finished by the same algorithm and the result is back collected.

Before doing the process of FFT, a signal from audio file which has been sampled is done a process of *frame blocking*—dividing a signal into some frames. Frame blocking results a discontinue signal so that it needs t be done a windowing process thereby *hamming window* technique.

From the research result and trial experiment which have been done by some audio file formats, it can be concluded that *Fast Fourier Transform* algorithm is able to do a voice transposition process with the degree of success 60% of parameter trial experiment which has been done that the best combination is oversampling 8 and FFT Frame size as 1024.

المستخلص

فرخية، لولوء. 2015 التحويل الندى الأغاني باستخدام الخوارزميات (Fast Fourier Transform (FFT). الأطروحة، كلية العلوم والتكنولوجيا الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج.
المشرف، توتوء خامدى, الماجستير
الكلمات البحث: إبدال الصوت، Fast Fourier Transform (FFT)، الإشارة.

فى عالم المعالجة الصوت الرقمي من الممكن لتغيير مجموعة من الأصوات، مثل تغيير نطاق الصوت أصل باس في مجموعة من صوت التينور.
واحد الخوارزمية التي تسمح له بالمشاركة مع التغييرات في نطاق الصوت الرقمي هو خوارزمية Fast Fourier Transform (FFT). تحويل الإشارات التناظرية إلى إشارات منفصلة في المجال الزمني، مجال التردد ومن ثم تغييره إلى. تجهيز إشارة بقسمة إشارة إلى عدة أجزاء والتي هي بعد ذلك يتم الانتهاء من كل قطعة مع نفس الخوارزمية ويتم جمع النتائج مرة أخرى.
قبل تجهيزها باستخدام الخوارزميات ((Fast Fourier Transform (FFT))، إشارة من الملفات الصوتية التي تم أخذ عينات لمعالجة إطار الحجب، أي تقسيم إشارة إلى إطارات متعددة. إطار التوقف عن عرقلة توليد الإشارات التي تحتاج إلى القيام به مع عمليةالنوافذ التقنيات المبالغة النافذة.
من نتائج البحث والتجارب التي أجريت مع مختلف صيغ الملفات الصوتية يمكن أن نخلص إلى أن الخوارزمية Fast Fourier Transform (FFT)) قادر على معالجة تبديل الصوت مع نسبة نجاح 60٪ مجموعات اختبار المعلمة التي بذلت قصارى يتم الإفراط 8 و FFT حجم الإطار 1024

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam ajaran agama Islam dianjurkan membaca kitab suci al-qur'an dengan mengeraskan, memperindah suara, dan khusyuk, sebagaimana hadist yang telah diriwayatkan oleh An-Nasai.

وعن أبي هريرة - رضي الله عنه - قَالَ: سَمِعْتُ رسولَ اللهِ صلى الله عليه وسلم - يقول: «مَا أذِنَ اللهُ لِشَيءٍ مَا أذِنَ لِنَبِيٍّ حَسَنِ الصَّوْتِ يَتَغَنَّى بالقُرْآنِ يَجْهَرُ بِهِ» . متفقٌعَلَيْهِ.

Artinya : Dari Abu Hurairah radhiyallahu 'anhu, beliau berkata: Aku pernah mendengar Rasulullah shallallahu 'alaihi wasallam bersabda: "Tidaklah Allah mendengarkan sesuatu, tidaklah Nabi mendengarkan sesuatu, kecuali suara yang indah ketika membaca al-Qur'an dan menyaringkannya.(Muttafaqun 'alaih).

Dan Ibnu Katsir berpendapat bahwa tujuan memperindah suara dengan maksud membangkitkan semangat untuk merenungi Al-Qur'an dan memahaminya, khusyu' dan penuh dengan ketundukan serta kepatuhan terhadap perintahnya. Sedangkan menyuarakan Al-Qur'an dengan patokan lagu dan irama yang bersifat hiburan dan aturan-aturan seperti musik tidak diperbolehkan karena telah datang sunnah yang menganggap hal itu dosa (Ibnu Katsir: 125-126)

Dibolehkanya memperindah suara ketika membaca Al-Qur'an memunculkan salah satu bidang ilmu yang dikenal luas oleh masyarakat dengan istilah qira'at. Di dalam ilmu qira'at ada lagu-lagu yang dianggap sebagai lagu pokok, yakni bayyati, Shaba, Hijaz, Nahawan, Ros, Jiharkah dan

Syika. Ketujuh jenis lagu pokok dalam seni baca al-qur'an ini biasanya dibawakan beberapa tahap tingkatan nada. Di mulai dari nada yang paling rendah sampai nada yang paling tinggi. Nada-nada tersebut adalah yaitu qarar (rendah), nawa (sedang), jawa b(tinggi), jawabul jawab (sangat tinggi) (Munir, dkk. 1994). Akan tetapi, tidak semua manusia di dunia ini dianugerahi dengan suara yang indah dan mampu membunyikan suara dengan nada yang tinggi dan benar.

Setiap manusia memiliki ciri khas dan range suara yang berbeda-beda, ada yang mempunyai range *soprano*, *alto*, *tenor*, *bass mezzo-soprano*, *baritone*, *soprano professional* dan *bass profesional* (Trioko, 2010). Meminta seseorang dengan suara *bass* menyanyikan lagu *range* tenor tidak akan menghasilkan performa yang baik, melainkan akan terdengar memaksa. Begitu juga ketika meminta seseorang dengan suara *soprano* menyanyikan *part alto*.

Ketika seseorang dengan tipe suara *bass* menyanyikan lagu dengan range *tenor* (nada tinggi) masyarakat menganggap bahwa seseorang tersebut kurang berlatih, padahal tidak ada hubunganya antara berlatih perubahan range suara, akan tetapi ada beberapa orang yang memiliki bakat dengan range vocal yang luar biasa.

Dalam dunia musik profesional, penyanyi akan bertemu dengan istilah transposisi nada (perubahan key nada). misalnya ketika penyanyi laki-laki dengan suara *baritone* menyanyikan lagu "Cinta" Vina Panduwinata, karena range vocal pria dan wanita berbeda, maka harus dilakukan proses

transposisi nada agar suara penyayi dengan melodi yang sedang dinyanyikan seimbang.

Biasanya untuk menyesuaikan nada antara suara penyanyi dengan musik yang mengiringinya, musisi berusaha menyesuaikan nada alat musiknya dengan suara penyanyi. Akan tetapi, proses penyesuaian nada yang dilakukan oleh musisi tidak bisa dilakukan jika musik yang mengiringi penyanyi adalah musik digital.

Pada awalnya musik digital hanyalah tekanan udara music analog yang ditangkap oleh *recorder* dan kemudian menjadi sebuah sinyal listrik analog. Sinyal analog tersebut dirubah menjadi sinyal digital oleh *analog to digital converter* yang kemudian disimpan dalam sebuah file (www.dspguide.com). File itulah yang dapat diputar dan diolah sehingga bisa melakukan konversi suara dengan memanfaatkan algoritma tertentu.

Salah satu formulasi yang ampuh untuk proses pengolahan sinyal adalah menggunakan *Discrete Fourier Transform* (DFT). Prinsip DFT adalah mentransformasikan (alih bentuk) sinyal yang semula analog menjadi diskret dalam domain waktu, dan kemudian diubah ke dalam domain frekuensi. Hal ini dilakukan dengan mengalikan sinyal diskret dengan suatu fungsi kernel. (EEPIS-IT, 2012)

Algoritma lain yang lebih cepat adalah *Fast Fourier Transform* (FFT). Prinsip kerja FFT adalah membagi sinyal hasil penyamplingan menjadi beberapa bagian yang kemudian masing-masing bagian diselesaikan dengan algoritma yang sama dan hasilnya dikumpulkan kembali. Ada tiga

kelas FFT yang umum digunakan di dalam suatu *software* DSP yaitu *Decimation in Time* (DIT), *Decimation in Frequency* (DIF) dan *Split Radix*. Ide ketiga jenis FFT tersebut adalah proses iterasi *sequence data* dilakukan secara berbeda dan memanfaatkan fungsi kernel yang memiliki sifat yang simetris pada suatu nilai tertentu dalam satu periode suatu sinyal. Jenis lain FFT yang sudah digunakan adalah paralel FFT dimana *sequence data* dikerjakan dengan menggunakan *parallel computing* sehingga proses transformasi akan lebih cepat. (Chu, Eleanor & George, 2000)

Penelitian yang akan dilakukan memanfaatkan *Fast Fourier Transform* (FFT) untuk memproses sinyal digital dari sebuah file audio untuk melakukan proses transposisi nada.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, maka perumusan masalah dalam penelitian ini yaitu :

1. Apakah cara melakukan transposisi nada sebuah lagu dapat menggunakan *algoritma fast fourier transform*
2. Seberapa baik kualitas lagu yang dihasilkan ketika nada telah ditransposisi dibandingkan dengan nada asli.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dalam penelitian ini sebagai berikut :

1. Membuktikan bahwa metode *fast fourier transform* dapat digunakan untuk mentransposisi nada.
2. Mengukur kualitas lagu yang dihasilkan ketika nada telah ditransposisi.

1.4 Batasan Penelitian

Batasan masalah pada penelitian ini adalah:

1. Transposisi nada yang dilakukan tanpa mengetahui dan menganalisa jenis tangga nada yang sedang dimainkan, akan tetapi hanya merubah suara (konversi) ke tangga nada yang lebih tinggi atau tangga nada yang lebih rendah.
2. Proses yang dilakukan tidak melakukan pemisahan vocal dan melodi dari file yang sedang dimainkan. Sistem akan melakukan transposisi nada (suara) dari file audio digital yang diinputkan secara langsung.
3. Jenis ekstensi file audio format yang didukung adalah jenis fortmat audio file yang didukung oleh VLC.
4. Aplikasi dibangun menggunakan bahasa pemrogaman java dan menggunakan library VLCJ.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah

1. Memudahkan penyanyi untuk menyesuaikan tinggi suaranya dengan nada lagu yang ingin dinyanyikan.
2. Sebagai referensi tingkat keberhasilan algoritma Fast Fourier Transform (FFT) untuk melakukan proses transposisi nada.

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, metedologi, 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, transposisi suara dan algoritma *Fast Fourier Trasform* (FFT)

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 *Fast Fourier Trasform* (FFT) 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 *Fast Fourier Trasform* (FFT) untuk melakukan transposisi 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 transposisi suara.

DAFTAR PUSTAKA

Seluruh bahan rujukan, sumber literature dan referensi dalam proses pembuatan dan penulisan laporan akan dicantumkan dalam daftar pustaka. Baik daftar pustaka buku elektronik (e-book), jurnal, *hardcover* (cetak) maupun daftar pustaka yang diambil dari *website.*

BAB II

STUDI LITERATUR

2.1 Penelitian Terkait

Eko Aditya Santoso et al (2010) melakukan penelitian tentang Klastering Suara Laki-Laki dan Perempuan Menggunakan Algoritma K-MeansBerdasarkan Hasil Ekstraksi FFT (Fast Fourier Transform) Mekanisme kerja dalam penelitian ini dengan mengumpulkan contoh-contoh suara, kemudian dilakukan ekstraksi ciri menggunakan FFT yang dimana menghasilkan dua fitur utama yaitu nilai maksimum frekuensi dan nilai maksimum simpangan frekuensi yang lalu disimpan dalam file. Pada saat ada input suara yang dimasukan, sistem akan mengolahnya dengan menggabungkan hasil ekstrasi cirinya dengan hasil ekstraksi ciri yang terdapat dalam file, lalu algoritma K-means digunakan untuk menghitung jarak terdekat antar centroid sehingga dapat dikelompokkan menjadi cluster 0 (laki –laki) dan cluster 1 (perempuan). Hasil dari uji coba ini menggunakan 20 data latih dan 10 data uji menghasilkan tingkat akurasi 80% untuk data latih dan 70% untuk data uji.

Nandra Pradipta (2011) melakukan penelitian tentang fast fourier transform digunakan untuk mempercepat proses penghitungan DFT Pengujian dilakukan dengan variasi sinyal berupa sinus, segitiga dan kotak dengan variasi frekuensi dan jumlah cuplikan.Sedangkan frekuensi cuplikan telah ditentukan sebelumnya sebesar 8000Hz.

Ibnu Daqiqil et al (2012) melakukan penelitian tentang pengenalan gelombang suara menjadi sebuah transkip chord, dimana nadanya berasal dari masukan audio file berformat wave. Chord yang akan 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.

Penelitian yang akan dilakukan hampir sama cara kerjanya dengan penelitian-penelitian yang telah disebutkan, akan tetapi setelah mendapatkan sinyal dalam domain frekuensi, sinyal diolah untuk melakukan konversi suara dengan faktor skala 0.5-2.0, hal ini berarti menaikkan / menurunkan frekuensi dari siyal yag sedang diolah.

2.2 Pengertian Bunyi

Bunyi, secara harafiah dapat diartikan sebagai sesuatu yang dapat didengar. Bunyi merupakan hasil getaran dari partikel-partikel yang berada di udara dan energi yang terkandung dalam bunyi dapat meningkat secara cepat dan dapat menempuh jarak yang sangat jauh (Egan, 1972).

Defenisi sejenis juga dikemukakan oleh Bruel & Kjaer (1986) yang menyatakan bahwa bunyi diidentikkan sebagai pergerakan gelombang di udara yang terjadi bila sumber bunyi mengubah partikel terdekat dari posisi diam menjadi partikel yang bergerak.

Secara lebih mendetail, Doelle (1972) menyatakan bahwa bunyi mempunyai dua defenisi, yaitu:

a. Secara fisis, bunyi adalah penyimpangan tekanan, pergeseran partikel dalam medium elastik seperti udara. Definisi ini dikenal sebagai bunyi Obyektif.
b. Secara fisiologis, bunyi adalah sensasi pendengaran yang disebabkan penyimpangan fisis yang digambarkan pada bagian atas. Hal ini disebut sebagai bunyi subyektif.

Secara singkat, Bunyi adalah suatu bentuk gelombang longitudinal yang merambat secara perapatan dan perenggangan terbentuk oleh partikel zat perantara serta ditimbulkan oleh sumber bunyi yang mengalami getaran. Rambatan gelombang bunyi disebabkan oleh lapisan perapatan dan peregangan partikel-partikel udara yang bergerak ke luar, yaitu karena penyimpangan tekanan. Hal serupa juga terjadi pada penyebaran gelombang air pada permukaan suatu kolam dari titik dimana batu dijatuhkan.

Gelombang bunyi adalah gelombang yang dirambatkan sebagai gelombang mekanik longitudinal yang dapat menjalar dalam medium padat,cair dan gas. Medium gelombang bunyi ini adalah molekulyang membentukbahan medium mekanik ini (Sutrisno, 1988). Gelombang bunyi

ini merupakan vibrasi/getaran molekul-molekul zat dan saling beradu satu sama lain namun demikian zat tersebut terkoordinasi menghasilkan gelombang serta mentransmisikan energi bahkan tidak pernah terjadi perpindahan partikel (Resnick dan Halliday, 1992).

2.3 Pengertian Musik

Sejarah perkembangan musik tidak dapat dilepaskan dari perkembangan budaya manusia. Hal ini disebabkan karena musik merupakan salah satu hasil dari budaya manusia di samping ilmu pengetahuan, arsitektur, bahasa dan sastra, dan lain sebagainya. Menurut Banoe (2003: 288), musik yang berasal dari kata muse yaitu salah satu dewa dalam mitologi Yunani kuno bagi cabang seni dan ilmu; dewa seni dan ilmu pengetahuan. Selain itu, beliau juga berpendapat bahwa musik merupakan cabang seni yang membahas dan menetapkan berbagai suara ke dalam pola-pola yang dapat dimengerti dan dipahami oleh manusia. Sementara itu menurut Jamalus (1988: 1), musik adalah suatu hasil karya seni berupa bunyi dalam bentuk lagu atau komposisi yang mengungkapkan pikiran dan perasaan penciptanya melalui unsur-unsur pokok musik yaitu irama, melodi, harmoni, dan bentuk atau struktur lagu serta ekspresi sebagai suatu kesatuan. Lebih lanjut Sylado (1983: 12) mengatakan bahwa musik adalah waktu yang memang untuk didengar. Musik merupakan wujud waktu yang hidup, yang merupakan kumpulan ilusi dan alunan suara. Alunan musik yang berisi rangkaian nada yang berjiwa akan mampu menggerakkan hati para pendengarnya.

Dari pendapat tersebut, dapat dikatakan bahwa musik adalah segala sesuatu yang ada hubungan dengan bunyi dan memiliki unsur-unsur irama, melodi dan harmoni yang mewujudkan sesuatu yang indah dan dapat dinikmati melalui indra pendengar. Dapat ditarik kesimpulan bahwa musik merupakan seni yang timbul dari perasaan atau pikiran manusia sebagai pengungkapan ekspresi diri, yang diolah dalam suatu nada-nada atau suara-suara yang harmonis. Jika musik diartikan sebagai ungkapan sederhana dari suasana hati jiwa atau respon harafiah terhadap peristiwa dari diri pribadi komponis, diperlukan informasi ataupun referensi yang cukup agar kita dapat menarik hubungan langsung antara kehidupan dengan karyanya.

Dalam pembentukkan musik secara utuh, unsur-unsur dan struktur musik mempunyai peranan penting dan keterkaitan yang kuat antara satu dan yang lainnya. Pada dasarnya unsur musik dapat dikelompokkan menjadi unsur-unsur pokok dan unsur-unsur ekspresi.

1. Unsur-unsur pokok meliputi: irama, melodi, harmoni dan bentuk atau stuktur lagu.
2. Unsur-unsur ekspresi meliputi: tempo, dinamik dan warna nada. (Jamalus, 1988: 7).

Teori musik merupakan cabang ilmu yang menjelaskan unsur-unsur musik. Cabang ilmu ini mencakup pengembangan dan penerapan metode untuk menganalisis maupun menggubah musik, dan keterkaitan antara notasi musik dan pembawaan musik. Musik terbentuk dari suara, melodi, notasi, harmoni, ritme, dan nada (Ensiklopedia Indonesia). Teori musik menjelaskan

bagaimana suara dinotasikan atau dituliskandan bagaimana suara tersebut ditangkap dalam benak pendengarnya.Dalam musik, gelombang suara biasanya dibahas tidak dalam panjang gelombangnya maupun periodenya, melainkan dalam frekuensinya. Aspek-aspek dasar suara dalam musik biasanya dijelaskan dalam *pitch*, yaitu tinggi nada, durasi (berapa lama suara ada), intensitas, dan timbre (warna bunyi).

Suara dapat dibagi-bagi ke dalam nada yang memiliki tinggi nada atau tala tertentu menurut frekuensinya ataupun menurut jarak relatif tinggi nada tersebut terhadap tinggi nada patokan.Perbedaan tala antara dua nada disebut sebagai interval.Nada dapat diatur dalam tangga nada yang berbeda-beda.Tangga nada yang paling lazim adalah tangga nada mayor, tangga nada minor, dan tangga nada pentatonik.Nada dasar suatu karya musik menentukan frekuensi tiap nada dalam karya tersebut. Nada memiliki sifat-sifat sebagai berikut:

1. Nada berkaitan dengan frekuensi atau banyaknya getaran tiap detik. Makin besar frekuensi, makin tinggi nadanya.
2. Panjang nada dihitung dengan satuan ketuk yang sifatnya relatif, bisa panjang bisa pendek.
3. Intensitas nada atau keras lembutnya bunyi suatu nada bergantung padalebarnya getaran dan sifatnya relatif.

Melodi adalah serangkaian nada dalam waktu.Rangkaian tersebut dapat dibunyikan sendirian, yaitu tanpa iringan, atau dapat merupakan bagian dari rangkaian akord dalam waktu (biasanya merupakan rangkaian nada

tertinggi dalam akord-akord tersebut). *Notasi musik* merupakan penggambaran tertulis atas musik.Dalam notasi balok, tinggi nada digambarkan secara vertical sedangkan waktu (ritme) digambarkan secara horisontal.Kedua unsur tersebut membentuk paranada, di samping petunjuk-petunjuk nada dasar, tempo, dinamika, dan sebagainya.*Harmoni* secara umum dapat dikatakan sebagai kejadian dua atau lebih nada dengan tinggi berbeda dibunyikan bersamaan, walaupun harmoni juga dapat terjadi bila nada-nada tersebut dibunyikan berurutan (seperti dalam *arpeggio*). Harmoni yang terdiri dari tiga atau lebih nada yang dibunyikan bersamaan biasanya disebut akord. (Pekerti, 1999: 263)

2.4 Transposisi Nada

Dalam seni musik terdapat istilah transposisi. Transposisi adalah pemindahan tangga nada dalam memainkan, menyanyikan, atau menuliskan sebuah lagu dari tangga nada aslinya, tetapi lagunya tetap sama. Setiap tangga nada memiliki kunci nada yang sangat dekat hubungannya dan saling berelasi, yaitu dominan, sub dominannya dan relatif minor maupun relatif mayornya. Transpoisisi ini digunakan antara lain, untuk :

1. Memindahkan lagu dari notasi angka ke notasi balok, atau sebaliknya memindakan suatu lagu dari notasi balok ke notasi angka.
2. Memindahkan suatu lagu dari notasi balok yang berlainan tanda kunci. Misal: dari kunci G ke kunci F, dan sebagainya.
3. Merubah nada dasar dari suatu lagu (Isfanhari dan Nugroho, 2000: 24-25).

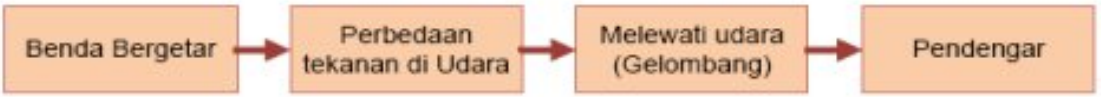
Bagi pemain musik (kecuali perkusi) kadangkala jika suara musik tidak pas dengan yang diinginkan, maka itu menjadi suatu masalah karena tidak tahu cara mengubah akordnya. Sebenarnya mirip dengan cara mengubah tangga nada dari satu nada dasar ke nada yang lain.Deretan nada sebagaimana yang ada di senar gitar ternyata dapat digeser kekiri atau kekanan berpindah-pindah dari kolom satu ke kolom yang lain. Menggeser kekiri berarti menaikkan nada (geser naik), sedangkan menggeser ke kanan berarti menurunkan nada.

Jika naik atau turunnya ke kolom berikutnya (terdekat) dinamakan naik atau turun ½ nada.Jika sampai melampaui satu kolom berarti naik atau turun satu nada.Untuk menaikkan dan menurunkan 1/2 nada tersebut ditulis dengan menggunakan tanda yang disebut “kromatik”. Ada 3 macam tanda kromatik yaitu:

1. Tanda "Kres", meninggikan 1/2 nada dengan simbol #
2. Tanda "Mol", menurunkan 1/2 nada dengan simbol b
3. Tanda "Netral", mengembalikan ke posisi dana semula dengan symbol

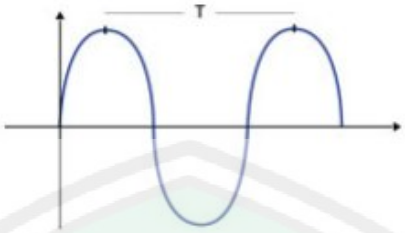
2.5 Teori Audio

Audio (suara) adalah fenomena fisik yang dihasilkan oleh getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu yang disebut frekuensi (Binanto, 2010).



Gambar 2.1 Alur gelombang suara (Binanto, 2010)

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 adalah instrumen musik, nyanyian burung sedangkan contoh suara non periodik adalah batuk, percikan ombak dan lain-lain. Gelombang suara terjadi sebagai variasi tekanan dalam sebuah media, seperti udara. Ia tercipta dari bergetarnya sebuah benda, yang menyebabkan udara sekitarnya ikut bergetar. Udara yang bergetar kemudian diterima oleh telinga, menyebabkan gendang telinga manusia bergetar, kemudian otak menafsirkannya sebagai suara. Gelombang suara berjalan melalui udara, sama seperti gelombang yang terjadi di air. Bahkan, gelombang air lebih mudah untuk dilihat dan dimengerti, hal ini sering digunakan sebagai analogi untuk menggambarkan bagaimana gelombang suara berperilaku. Gelombang suara juga dapat ditampilkan dalam stkitar grafik XY. Hal ini memungkinkan kita untuk membayangkan dan bekerja dengan gelombang dari sudut pkitang matematika.



Gambar 2.2 Gelombang suara (Binanto, 2010)

Perhatikan, bahwa suatu grafik gelombang berbentuk dua dimensi, tetapi di dunia nyata gelombang suara berbentuk tiga-dimensi. Grafik menunjukkan gelombang bergerak sepanjang jalan dari kiri ke kanan, tapi kenyataannya perjalanan gelombang suara bergerak ke segala arah menjauhi sumber. Kira-kira sama seperti riak air yang terjadi ketika kita menjatuhkan sebuah batu ke dalam kolam. Namun model 2-dimensi ini, cukup dapat menjelaskan tentang bagaimana suara bergerak dari satu tempat ke tempat lain. (Binanto, 2010)

2.6 Bagian-bagian Gelombang Suara

Semua gelombang memiliki sifat-sifat tertentu. Ada tiga bagian yang paling penting untuk audio:

a. Panjang gelombang: Jarak antara titik manapun pada gelombang (pada gambar ditunjukkan sebagai titik tertinggi) dan titik setara pada fase berikutnya. Secara harfiah, panjang gelombang adalah jarak yang digambarkan dgn huruf "T".

b. Amplitudo: atau kekuatan sinyal gelombang (intensity). Titik tertinggi dari gelombang bila dilihat pada grafik. Amplitudo tinggi biasa disebut

sebagai volume yang lebih tinggi, diukur dalam dB. Nama perangkat untuk meningkatkan amplitudo disebut amplifier. (Tambunan, 2005)

c. Frequency: Frekuensi waktu yang dibutuhkan oleh gelombang bergerak dari satu fase ke fase berikutnya dalam satu detik. Diukur dalam hertz atau cycles per second. Semakin cepat sumber suara bergetar, semakin tinggi frekuensi. Frekuensi yang lebih tinggi ditafsirkan sebagai pitch yang lebih tinggi. Sebagai contoh, ketika Kita menyanyi dengan suara bernada tinggi Kita memaksa pita suara Kita bergetar lebih cepat. (Wardhana, 2001).

Hal berikutnya yang perlu diperhatikan adalah apa artinya ketika gelombang mencapai titik tertinggi atau titik rendah. Pada sinyal elektronik, nilai tinggi menunjukkan tegangan positif yang tinggi. Ketika sinyal ini dikonversi menjadi gelombang suara, Kita dapat membayangkan nilai-nilai tinggi tersebut sebagai daerah yang mewakili peningkatan tekanan udara. Ketika gelombang menyentuh titik tertinggi, hal ini berhubungan dengan molekul udara yang menyebar bersama-sama secara padat. Ketika gelombang menyentuh titik rendah, molekul udara menyebar lebih tipis (renggang).

Audio diproduksi oleh sebuah objek yang bergetar, contohnya pengeras suara, alat musik, 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 yang dapat didengar manusia 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. (Wardhana, 2001).

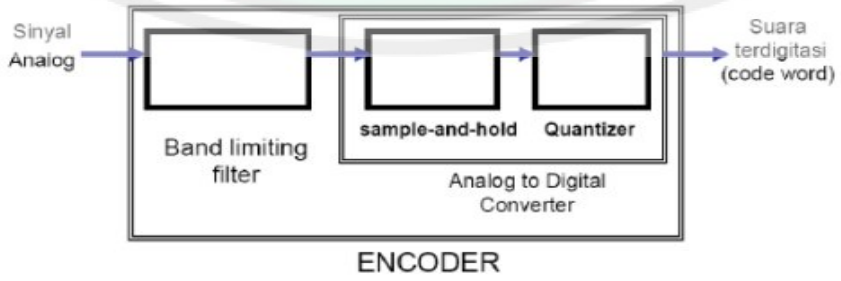
Sebuah sinyal suara diproduksi dan ditransmisikan melalui udara, akhirnya diterima pada telinga manusia. Telinga manusia memiliki gendang pendengaran (eardrum) yang dapat bergetar pada saat menerima gerakan gelombang udara (push and pull). Penggelompokan sound dapat dilakukan berdasarkan acoustic behaviornya. Berdasarkan acoustic behavior-nya sound dibedakan menjadi dua jenis yaitu direct sound dan indirect sound (ambient). Dikatakan direct sound apabila sumber suara berjalan dari sumber suara langsung menuju ke pendengar dalam aliran garis lurus. Indirect sound bila sumber suara dipantulkan terlebih dahulu pada satu atau lebih permukaan bidang sebelum sampai pada pendengar, karena adanya proses pemantulan sinyal suara pada indirect sound maka ditemukan adanya delay time untuk tiba kepada pendengar.

Terdapat beberapa macam indirect sound, tergantung pada room acoustic, sebagai contoh pengaturan pada car audio lebih susah apabila dibandingkan room audio karena bentuk ruang dan material yang sangat memungkinkan terjadinya banyak pantulan sumber suara sebelum sampai ke

pendengar. Hal ini sangat menyulitkan penempatan sumber suara dalam mobil. Echo atau gema terjadi ketika sebuah indirect sound tertunda dalam waktu yang cukup lama untuk dapat didengar pendengar sebagai perulangan sinyal suara sebuah direct sound.

2.7 Representasi Suara

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan "sample". Analog to Digital Conversion (ADC) adalah proses mengubah amplitudo gelombang bunyi ke dalam waktu interval tertentu (sampling), sehingga menghasilkan representasi digital dari suara. Dalam teknik sampling dikenal istilah sampling rate yaitu beberapa gelombang yang diambil dalam satu detik. Sebagai contoh jika kualitas CD Audio dikatakan memiliki frekuensi sebesar 44.100 Hz, berarti jumlah sampel sebesar 44.100 per detik. Proses Sampling Audio Analog ke Digital dapat dilihat seperti pada Gambar 2.3.



Gambar 2.3 Proses *Sampling Analog* ke *Digital* (Binanto, 2010)

Langkah-langkah dalam proses digitasi adalah: 1. Membuang frekuensi tinggi dari source signal. 2. Mengambil sample pada interval waktu tertentu (sampling). 3. Menyimpan amplitudo sampel dan mengubahnya ke dalam bentuk diskrit (kuantisasi) 4. Merubah bentuk menjadi nilai biner. Teknik sampling yang umum pada file audio seperti Nyquist Sampling Rate dimana untuk memperoleh representasi akurat dari suatu sinyal analog secara lossless, amplitudonya harus diambil sample-nya setidaknya pada kecepatan (rate) sama atau lebih besar dari 2 kali lipat komponen frekuensi maksimum yang akan didengar. Misalnya untuk sinyal analog dengan bandwith 15Hz – 10kHz→sampling rate = 2 x 10KHz = 20 kHz (Gunawan, 2005).

2.8 Sinyal Digital

Pada dasarnya semua suara audio, baik vokal maupun bunyi tertentu merupakan suatu bentukan dari getaran. Ini menandakan semua audio memiliki bentuk gelomban gnya masing-masing. Umumnya bentukan gelombangnya disebut dengan sinyal analog. Namun sebuah teknik memungkinkan sinyal ini diubah dan diproses sehingga menjadi lebih baik. Teknik ini memungkinkan perubahan sinyal analog menjadi bit-bit digital. Teknik itu disebut teknik sampling. Jika telah menjadi sinyal digital maka sinyal ini jauh lebih baik, sedikit noisenya dan juga dapat diproses dengan mudah. Digital Signal Prosessing merupakan perkembangan dari teknik ini yang memungkinkan kita membentuk sample-sample yang berupa suara seperti yang ada pada keyboard, syntitizer, Audio Prosessing, dll.

2.8.1 Proses sampling

Pada proses ini terjadi suatu pencuplikan dari bentukan sinyal analog. Pencuplikan dilakukan pada bagian-bagian sinyal analog. Ini dilakukan dengan sinyal-sinyal sample. Ada suatu aturan tertentu dari sinyal ini. Teori Shannon menyatakan frekuensi sinyal ini paling sedikit adalah 2 kali frekuensi sinyal yang akan disampling(sinyal analog). Ini adalah batas minimum dari frekuensi sample agar nantinya cuplikan yang diambil menunjukkan bentukan sinyal yang asli (analog). Lebih besar tentunya lebih baik, karena cuplikan akan lebih menggambarkan sinyal yang asli. Setelah dilakukan proses ini maka terbentuklah suatu sinyal analog-diskrit yang bentuknya menyerupai aslinya namun hanya diambil diskrit-diskrit saja.

2.8.2 Quantisasi (Perhitungan)

Ini adalah proses pembandingan level-level tiap diskrit sinyal hasil sampling dengan tetapan level tertentu. Level-level ini adalah tetapan angka-angka yang dijadikan menjadi bilangan biner. Sinyal-sinyal diskrit yang ada akan disesuaikan levelnya dengan tetapan yang ada. Jika lebih kecil akan dinaikkan dan jika lebih besar akan diturunkan. Prosesnya hampir sama dengan pembulatan angka. Tetapan level yang ada tergantung pada resolusi dari alat, karena tetapan level merupakan kombinasi angka biner, maka jika bitnya lebih besar kombinasinya akan lebih banyak dan tetapan akan lebih banyak. Ini

membuat pembulatan level sinyal diskrit menjadi tidak jauh dengan level aslinya. Dan bentukan sinyal akan lebih bervariasi sehingga akan terbentuk seperti aslinya. Proses ini membuat sinyal lebih baik karena bentukkannya lebih tetap. Proses ini juga mengecilkan error dari suatu sinyal.

2.8.3 Perubahan ke digital

Setelah diquantisasi maka tiap-tiap diskrit yang ada telah memiliki tetapan tertentu. Tetapan ini dapat dijadikan kombinasi bilangan biner, maka terbentuklah bilangan-bilangan biner yang merupakan informasi dari sinyal. Setelah menjadi sinyal digital maka proses-proses perekayasaan dapat dilakukan. Yang harus dilakukan adalah merubah informasi digital tersebut dengan proses digital sehingga menjadi suara-suara yang kita inginkan. Proses dapat dilakukan dengan berbagai macam alat-alat digital (co:komputer). Sample-sample yang ada juga digunakan sebagai informasi untuk menciptakan suara dari berbagai macam alat elektronik (co:keyboard dan syntitizer). Penyimpanan suara juga akan lebih baik karena informasinya adalah digital sehingga berkembanglah CD dan DAT(Digital Tape).

2.8.4 Sistem Input Komputer

Piranti input menyediakan informasi kepada sistem komputer dari dunia luar. Dalam sistem komputer pribadi, piranti input yang paling umum adalah keyboard. Komputer mainframe menggunakan keyboard dan pembaca kartu berlubang sebagai piranti inputnya. Sistem dengan mikrokontroler umumnya menggunakan piranti input yang jauh lebih kecil seperti saklar atau keypad kecil.

Hampir semua input mikrokontroler hanya dapat memproses sinyal input digital dengan tegangan yang sama dengan tegangan logika dari sumber. Level nol disebut dengan VSS dan tegangan positif sumber (VDD) umumnya adalah 5 volt. Padahal dalam dunia nyata terdapat banyak sinyal analog atau sinyal dengan tegangan level yang bervariasi. Karena itu ada piranti input yang mengkonversikan sinyal analog menjadi sinyal digital sehingga komputer bisa mengerti dan menggunakannya. Ada beberapa mikrokontroler yang dilengkapi dengan piranti konversi ini, yang disebut dengan ADC, dalam satu rangkaian terpadu.

2.8.5 Sistem Output Komputer

Piranti output digunakan untuk berkomunikasi informasi maupun aksi dari sistem komputer dengan dunia luar. Dalam sistem komputer pribadi (PC), piranti output yang umum adalah monitor CRT. Sedangkan sistem mikrokontroler mempunyai output yang jauh lebih

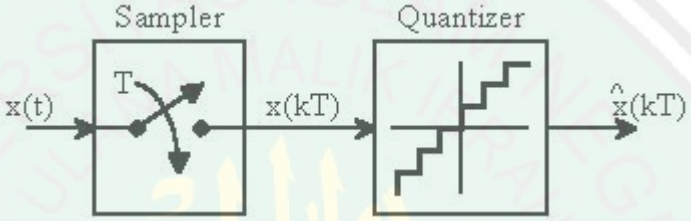
sederhana seperti lampu indikator atau beeper. Frasa kontroler dari kata mikrokontroler memberikan penegasan bahwa alat ini mengontrol sesuatu. Mikrokontroler atau komputer mengolah sinyal secara digital, sehingga untuk dapat memberikan output analog diperlukan proses konversi dari sinyal digital menjadi analog. Piranti yang dapat melakukan konversi ini disebut dengan DAC (Digital to Analog Converter).

2.9 Pengolahan Sinyal Digital

Proses pengolahan sinyal digital, diawali dengan proses pencuplikan sinyal masukan yang berupa sinyal kontinyu. Proses ini mengubah representasi sinyal yang tadinya berupa sinyal kontinyu menjadi sinyal diskrete. Proses ini dilakukan oleh suatu unit ADC (Analog to Digital Converter). Unit ADC ini terdiri dari sebuah bagian Sample/Hold dan sebuah bagian quantiser. Unit sample/hold merupakan bagian yang melakukan pencuplikan orde ke-0, yang berarti nilai masukan selama kurun waktu T dianggap memiliki nilai yang sama. Pencuplikan dilakukan setiap satu satuan waktu yang lazim disebut sebagai waktu cuplik (sampling time). Bagian quantiser akan merubah menjadi beberapa level nilai, pembagian level nilai ini bisa secara uniform ataupun secara non-uniform misal pada Gaussian quantiser.

Unjuk kerja dari suatu ADC bergantung pada beberapa parameter, parameter utama yang menjadi pertimbangan adalah sebagai berikut :

- Kecepatan maksimum dari waktu cuplik.
- Kecepatan ADC melakukan konversi.
- Resolusi dari quantiser, misal 8 bit akan mengubah menjadi 256 tingkatan nilai.
- Metoda kuantisasi akan mempengaruhi terhadap kekebalan noise.



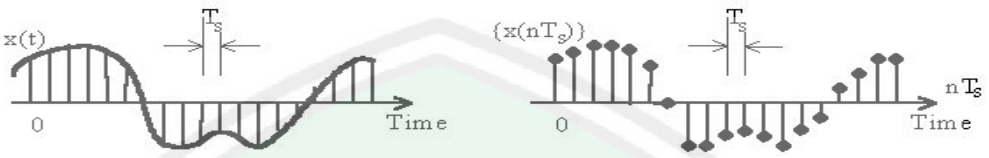
Gambar 2.4 Proses sampling

(Sumber dari *setiawan.blog.uns.ac.id/?p=54* diakses pada tanggal 5 september 2014)

Sinyal input asli yang tadinya berupa sinyal kontinyu, x(T) akan dicuplik dan diquantise sehingga berubah menjadi sinyal diskrete x(kT). Dalam representasi yang baru inilah sinyal diolah.Keuntungan dari metoda ini adalah pengolahan menjadi mudah dan dapat memanfaatkan program sebagai pengolahnya. Dalam proses sampling ini diasumsikan kita menggunakan waktu cuplik yang sama dan konstan, yaitu Ts. Parameter cuplik ini menentukan dari frekuensi harmonis tertinggi dari sinyal yang masih dapat ditangkap oleh proses cuplik ini. Frekuensi sampling minimal adalah 2 kali dari frekuensi harmonis dari sinyal.

Untuk mengurangi kesalahan cuplik maka lazimnya digunakan filter anti-aliasing sebelum dilakukan proses pencuplikan. Filter ini digunakan untuk meyakinkan bahwa komponen sinyal yang dicuplik adalah benar-benar

yang kurang dari batas tersebut. Sebagai ilustrasi, proses pencuplikan suatu sinyal digambarkan pada gambar berikut ini.



Gambar 2.5 Pengubahan dari Sinyal Kontinyu ke Sinyal Diskret

(Sumber dari *setiawan.blog.uns.ac.id/?p=54* diakses pada tanggal 5 september 2014)

Setelah sinyal diubah representasinya menjadi deretan data diskrete, selanjutnya data ini dapat diolah oleh prosesor menggunakan suatu algoritma pemrosesan yang diimplementasikan dalam program. Hasil dari pemrosesan akan dilewatkan ke suatu DAC (Digital to Analog Converter) dan LPF (Low Pass Filter) untuk dapat diubah menjadi sinyal kontinyu kembali. Secara garis besar, blok diagram dari suatu pengolahan sinyal digital adalah sebagai berikut :

Gambar 2.6 Blok Diagram Sistem Pengolahan Sinyal Digital

(Sumber dari *setiawan.blog.uns.ac.id/?p=54* diakses pada tanggal 5 september 2014)

Proses pengolahan sinyal digital dapat dilakukan oleh prosesor general seperti halnya yang lazim digunakan di personal komputer, misal processor

80386, 68030, ataupun oleh prosesor RISC seperti 80860. Untuk kebutuhan pemrosesan real time, dibutuhkan prosesor yang khusus dirancang untuk tujuan tersebut, misal ADSP2100, DSP56001, TMS320C25, atau untuk kebutuhan proses yang cepat dapat digunakan paralel chip TMS320C40. Chip-chip DSP ini memiliki arsitektur khusus yang lazim dikenal dengan arsitektur Harvard, yang memisahkan antara jalur data dan jalur kode.Arsitektur ini memberikan keuntungan yaitu adanya kemampuan untuk mengolah perhitungan matematis dengan cepat, misal dalam satu siklus dapat melakukan suatu perkalian matrix.Untuk chip-chip DSP, instruksi yang digunakan berbeda pula.Lazimnya mereka memiliki suatu instruksi yang sangat membantu dalam perhitungan matrix, yaitu perkalian dan penjumlahan dilakukan dalam siklus (bandingkan dengan 80386, proses penjumlahan saja dilakukan lebih dari 1 siklus mesin).

2.10 Algoritma Fast Fourier Transform (FFT)

Sebelum membahas tentang FFT atau Fast Fourier Transform, akan dibahas tentang Periode dan Frekuensi. Hal ini penting untuk menjelaskan secara rinci tentang fungsi dari FFT dalam pengolahan isyarat.

2.10.1 Periode

Secara umum periode didefinisikan sebagai waktu yang dibutuhkan untuk sebuah isyarat atau gelombang mencapai satu gelombang penuh. Yang perlu dicermati adalah bahwa pengertian ini

berlaku untuk isyarat monokromatis. Isyarat monokromatis yang dimaksud adalah gelombangnya bersifat tunggal, dimana dia pasti memiliki sebuah periode. Dengan demikian isyarat itu dikenal dengan istilah periodis. Secara gamblang kita bisa mengenali nilai-nilai yang terkandung di dalam isyarat termasuk nilai periode nya.

2.10.2 Frekuensi

Secara umum frekuensi diartikan sebagai jumlah gelombang yang terjadi dalam 1 detik. Frekuensi didefinisikan secara sederhana sebagai kebalikan dari waktu. Sehingga waktu yang satuannya adalah detik (second) akan menjadi Hertz (1/second) untuk frekuensi. tentu frekuensi hanya akan memiliki tepat satu nilai spektrum. Yang dikenal dengan spektrum frekuensi. Pengertian frekuensi ini juga berlaku untuk gelombang monokromatis.

2.10.3 Algoritma Fast Fourier Transform (FFT)

FFT merupakan DFT (*Discrete Fourier Transform*) yang memiliki jumlah komputasi lebih sedikit dibanding komputasi DFT biasa. DFT akan menghasilkan jumlah komputasi sebesar N^2 sedangkan FFT akan menghasilkan jumlah komputasi sebesar $(N)\log_2(N)$. Perhitungan FFT menggunakan *butterfly* Radix-2 menghasilkan jumlah komputasi lebih sedikit yakni $(N/2)\log_2(N)$. Jumlah titik dalam ketika menggunakan FFT juga memenuhi syarat

2^N. Untuk alasan tersebut FFT menjadi pilihan ketika dihadapkan dengan alih ragam fourier. Namun ada yang harus dipenuhi sebelum memutuskan untuk menggunakan FFT sebagai solusi. Isyarat yang akan diolah harus bersifat stasioner, aperiodis (tidak periodis). Isyarat periodis kontinu dapat dianalisa menggunakan deret Fourier.

Hasil transformasi ini dipengaruhi oleh beberapa parameter, yaitu sample rate sinyal suara dan FFT size. Sample rate adalah banyaknya sampel input analog yang diambil secara digital dengan satuan Hertz (Hz). Sample rate sinyal suara berpengaruh pada besarnya jangkauan frekuensi dari koefisien hasil FFT. Jangkauan frekuensi hasil FFT adalah setengah dari sample rate sinyal suara yang ditransformasi. Artinya, apabila terdapat sinyal suara dengan sample rate 44100 Hz, maka koefisien-koefisien hasil transformasi dari sinyal suara tersebut berkisar dari 0 Hz sampai 22050 Hz. Jadi, semakin besar sample rate , maka akan semakin detail pula sampel analog yang diambil secara digital.Sedangkan FFT size adalah panjang dari FFT yang digunakan. FFT size berpengaruh terhadap ketelitian tiap koefisien FFT. Semakin besar FFT size, maka tiap koefisien hasil FFT akan mewakili rentang frekuensi yang semakin kecil, sehingga ketelitiannya semakin tinggi. Sebaliknya apabila ukuran sampel FFT semakin kecil, maka tiap koefisien hasil FFT akan mewakili rentang frekuensi yang semakin besar, sehingga ketelitiannya semakin rendah. *Output* dari FFT ditransformasikan ke dalam rentang frekuensi. Nilai

magnitude terhadap frekuensi didapatkan berdasarkan rumus dibawah ini:

$$|M(f) = \sqrt{RealOut\ [j]^2 + ImageOut[j]^2}|$$

Dimana

M(t) = fungsi atau sinyal dalam domain waktu,

$e^{-2\pi ft}$ = fungsi kernel

M(f) = fungsi dalam domain frekuensi

f = frekuensi

2.10.4 Frame Blocking dan Windowing

Frame Blocking adalah pembagian sinyal audio menjadi beberapa *frame* yang nantinya dapat memudahkan dalam perhitungan dan analisa sinyal, satu frame terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi samplingnya. Pada proses ini dilakukan pemotongan sinyal dalam slot-slot tertentu agar memenuhi 2 syarat yaitu linear dan time invariant.

Sinyal terpotong yang discontinue tersebut dikalikan dengan fungsi window agar menjadi sinyal yang continue. Fungsi windowing yang digunakan dalam penelitian ini adalah window Hamming karena fungsi hamming dapat membuat data pada awal frame dan akhir frame mendekati nilai 0 dengan baik. Dengan demikian sinyal menjadi kontinyu.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Desain Penelitian

Penelitian kuantitatif adalah penelitian yang melibatkan pengukuran tingkatan suatu ciri tertentu (Arikunto, 2006). Jenis penelitian yang digunakan dalam penelitian ini yaitu jenis penelitian kuantitatif. Penelitian kuantitatif merupakan jenis penelitian yang didasarkan atas perhitungan presentase, rata-rata, kuadrat, dan perhitungan statistik lainnya. Dengan kata lain ciri penelitian kuantitatif adalah penelitian yang harus melibatkan diri pada perhitungan matematis atau angka-angka.

Sumber data seluruh yang tertulis dalam usulan 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.

3.2 Sumber Data

Pada penelitian ini akanmelakukan transposisi nada menggunakan metode *fast fourier transform.* Algoritma Fast Fourier transform digunakan untuk melakukan proses tranposisi suara yang sumber filenya didapatkan dari internet dengan

berbagai macam format sebagai berikut:

1. MP3 - MPEG Layer 3
2. MP4
3. MPEG
4. MIDI
5. WMV
6. WAV dan lainya sebagaimana file yang didukung VLC Player.

3.3 Instrumen Penelitian

Pada penelitian ini terdapat beberapa variabel yang digunakan dalam menerapkan metode Fast Fourier transform yaitu:

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 suara. Untuk melakukan transposisi suara dengan menggunakan algoritma Fast Fourier transform ada beberapa parameter sebagai berikut:

a. Aspect Ratio

b. Prosentase pitch

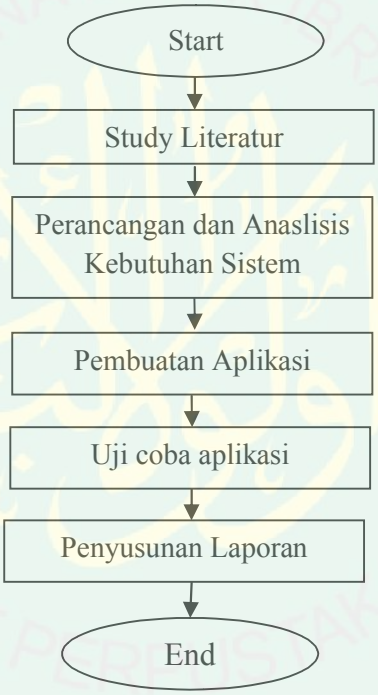
2. Variabel penghubung

Variabel penghubung dari penelitian ini adalah Format file

3. Variabel terikat

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

3.4 Prosedur Penelitian



Gambar 3.1 Prosedur penelitian

Tahapan – tahapan dalam prosedur penelitian yaitu sebagai berikut :

1. Study Literature

Study literature (kajian pustaka) merupakan keegiatan yang dilakukan untuk melakukan penelusuran literatur yang bersumber dari buku, media, pakar ataupun dari hasil penelitian orang lain yang bertujuan untuk

menyusun dasar teori proses transposisi suara. Di dalam proses ini juga melakukan proses observasi progamming audio pada java dan library yang mendukung untuk pembuatan aplikasi menggunakan java.

2. Perancangan dan Anaslisis Kebutuhan Sistem.

   Perencanaan 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.

3. Pembuatan Aplikasi

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

4. Uji coba dan Evaluasi

   Uji coba dan evaluasi dilakukan untuk melakukan evaluasi keberhasilan aplikasi.

5. Penyusunan Laporan

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

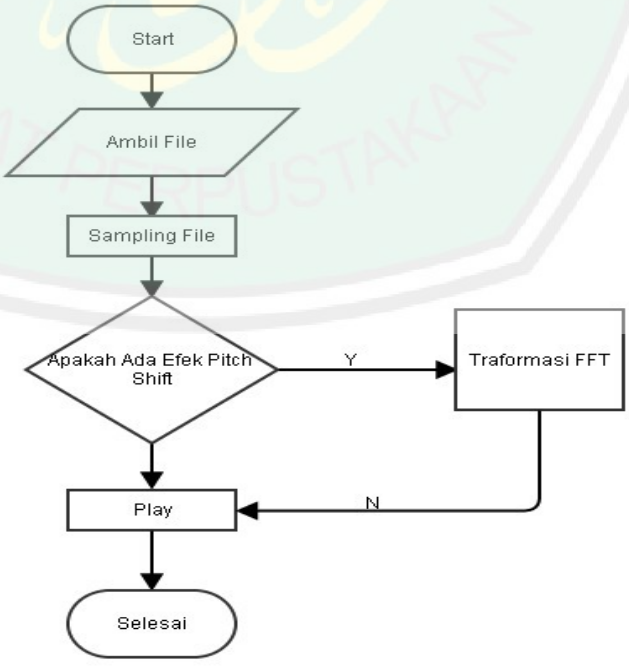
3.5 Desain Sistem

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

1. Alur sistem

Alur system menjelaskan alur pengolahan sinyal dari proses pengambilan sinyal dari file audio, proses transposisi, 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. Sistem dimulai dengan penginputan file audio, dan proses sampling, yaknimengkonversi suatu sinyal waktu-kontinu mejadi sinyal diskrit yang diperoleh dengan mengambil cuplikan (*sample*) sinyal waktu-kontinu pasa saat waktu-diskrit.

Informasi data yang terdapat di dalam file tersebut masih dalam domain waktu, sehingga harus diubah menjadi informasi data dalam domain frekuensi dengan menggunakan algoritma Fast Fourier Transform. Sinyal suara yang diurai menjadi data berbentuk frekuensi memiliki informasi yang lebih mudah.

Perhitungan untuk menentukan nilai frekuensi dan intensitas suara adalah sebagai berikut :

=——…….(persamaan 1)

=$\frac{\Sigma}{\quad}$ ……..(persamaan 2)

I = 20 x log(abs(Y) +)……(persamaan 3)

Keterangan:

Hz : Batasan frekuensi

M : Batas maksimal frekuensi yang ditentukan.

L : Banyak array dalam proses FFT

Fs : Frekuensi sampling

f: Frekuensi (dalam satuan herti : nilai array dalam proses FFT

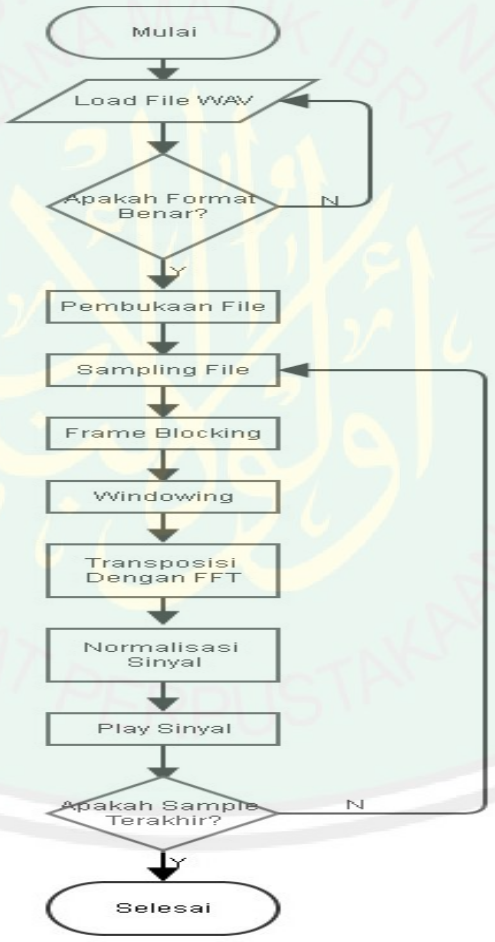
I : Intensitas suara (dalam satuan desibel)

Y : Nilai dari hasil FFT B

ε: Bilangan epsilon (ε = 2.22024 x 10-16)

2. Detail alur sistem dan rancangan penerapan algortima Fast Fourier Transform (FFT)

Langkah-langkah yang akan dilakukan untuk melakukan transposisi suara adalah sebagaimana gambar 3.3.



Gambar 3.3 Proses transposisi suara

1. **Menginputkan file audio kedalam aplikasi.**

   File yang diinputkan ke dalam system adalah file video atau audio dengan format yang didukung oleh library engine yang digunakan, yakni VLCJ.

2. **Validasi FileFormat File**

   Jika file audio sesuai maka file audio siap untuk diproses, sedangkan kalau tidak sesuai user harus menginputkan ulang file audio dengan format yang yang didukung oleh *engine native library* VLCJ.

3. **Sampling file**

   Proses sampling, yaknimengkonversi suatu sinyal waktu-kontinu mejadi sinyal diskrit yang diperoleh dengan mengambil cuplikan (sample) sinyal waktu-kontinu pasa saat waktu-diskrit.

4. ***Frame blocking***

   Frame blocking adalah suatu proses pada tahapan sinyal yang membagi sinyal menjadi beberapa frame yang nantinya dapat memudahkan dalam perhitungan dan analisa suara, setiap potongan-potongan disebut frame. Satu frame terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi samplingnya.

   Contoh perhitungan *frame blocking*:

   Misalnya sinyal disampling dengan waktu tiap 20 ms, dan sampel rate 11025 Hz, Maka akan didapatkan frame size (N)

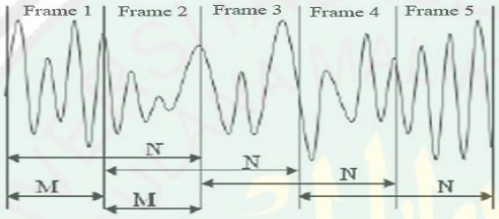
N = 11025 *0.02 = 220.5 sampel

Dan akan didapatkan Overlapping frame (M) = 110.25

X(n) = y(M+n)

X(1) = 100(110.25+1)=11125.

potongan-potongan frame digambarkan dengan gambar dibawah ini:



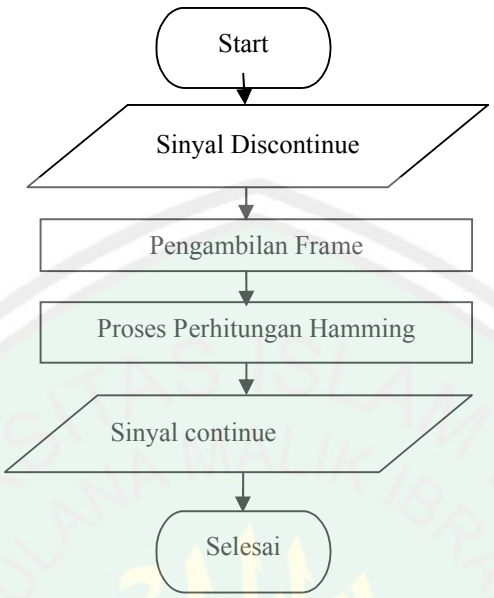
Gambar 3.4 Gambaran proses *frame blocking*

Keterangan:

Frame1, frame2, frame3, frame4, frame5 adalah sinyal suara yang telah dilakukan proses frame blocking, dimana M panjang tiap frame dan N overlapping tiap frame-nya.

5. Windowing

Hasil dari proses frame blocking menghasilkan efek sinyal discontinue, agar tidak terjadi kesalahan data pada proses fourier transform maka sampel suara yang telah dibagi menjadi beberapa frame perlu dijadikan suara continue dengan menggunakan proses windowing. Hal ini dilakukan untuk meminimalkan pada bagian awal dan akhir sinyal.Jika di definisikan sebuah window w(n) dan sinyal tiap bagian adalah i maka sinyal hasil proses windowing. Proses window hamming di gambarkan sebagi berikut



Gambar 3.5 Flowchart proses window hamming

Jenis window yang digunakan adalah *Hamming Window* dengan rumus sebagai berikut:

w[n]= 0.54-0.46*cos(2*p*n/(N-1))

Contoh perhitungan *Hamming Window* adalah sebagai berikut:

Diketahui: p=3.14, n=1,2,3,… N=220,5

Ditanya : w(n)

Jawab :

w(n)= 0.54-0.46 cos (2pn/(N-1))

w(1)= 0.54-0.46 cos(2.3.14.1/(220,5-1))

w(1)= 0.54-0.46 cos 0

w(1)= 0.54-0.46*1

w(1)= 0.08 dst

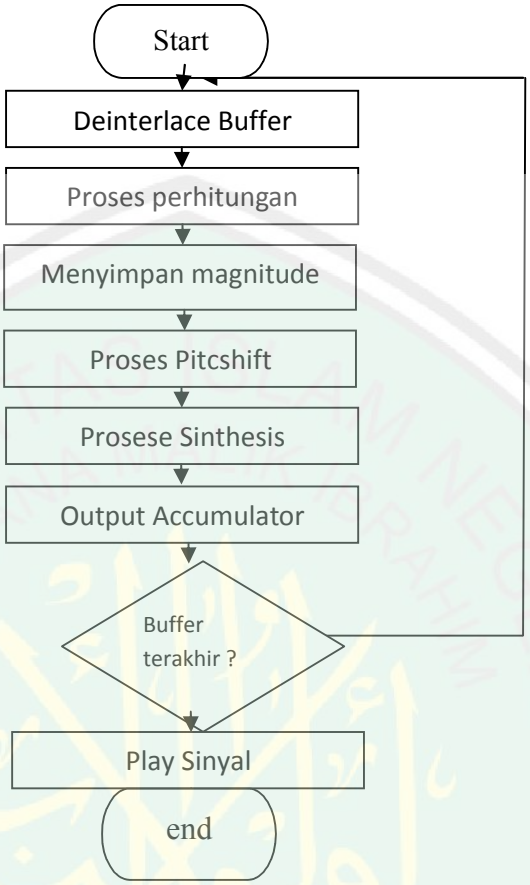
Hasil dari *Hamming Window* adalah [890;924,9;960,36]

6. Transposisi dengan *Fast Fourier Transform* (FFT)

Segala macam transformasi, seperti namanya mempunyai arti pengubahan. Dalam hal ini implementasinya adalah mengubah dari kawasan waktu menjadi kawasan frekuensi. Transformasi tersebut dapat dilakukan pada suatu sistem atau fungsi maupun terhadap suatu sinyal. DFT mengubah sinyal atau sistem dari kawasan waktu ke kawasan frekuensi. Tujuan segala macam transformasi adalah tersebut adalah untuk mempermudah analisisnya.

Proses windowing menghasilkan spektrum suara dalam domain waktu, sedangkan proses FFT merubah spectrum domain waktu menjadi sinyal frekuensi dengan menggunakan proses Fast Fourier Transform. FFT merupakan salah satu metode untuk transformasi sinyal suara menjadi sinyal frekuensi. Artinya file audio suara disimpan dalam bentuk digital berupa gelombang spektrum suara berbasis frekuensi. Hasil dari proses fast fourier transform menghasilkan pendeteksian gelombang frekuensi domain dalam bentuk diskrit.

Untuk Lebih detailnya langkah-langkah FFT dan proses transposisi suara sebagaimana pada flowchart berikut :



Gambar 3.6 Flowchart Proses Fast Fourier Transform

a. Deinterlace buffer

Deinterlace buffer adalah proses konversi buffer ke bentuk non-interlaced dengan cara membagi data sample menjadi 2 fungsi yaitu ganjil dan genap.

b. Proses Perhitungan FFT

Ada beberapa proses perhitungan di dalam proses FFT, untuk mendapatkan variabel-variabel FFT. Perhitungan tersebut adalah

1. Menghitung Magnitude dan Phase

FFT yang dikenakan pada sinyal akan menghasilkan spektrum frekuensi yang terdiri dari spektrum magnitude dan spektrum fasa yang menunjukan hubungan antara magnitude, fasa dengan frekuensi. Artinya kita dapat melihat komponen penyusun sinyal tersebut dari magnitude sinyal pada rentang frekuensi. Begitu juga dengan fasenya. Pada penelitian ini perhitungan magnitude untuk melihat seberapa besar magnitude dan frekuesni,yang menunjukkan dominan frekuensi dari file audio tersebut.

Magnitude adalah variabel ukuran kekuatan sinyanl, yang mengabaikan awah sinyal.sehingga magnitude selalu bernilai positif sehingga biasa disebut sebagai nilai absolute.

Magnitude dihitung dengan rumus:

Magnitude = 2.* ———————— * + *

Sedangkan phasa dihitung dengan menggunakan rumus

Phasa = tan2(imag,real),

2. Menghitung *Phase Difference.*

Phase Difference adalah selisih antara hasil proses perhitungan phase sekarang dengan hasil phase sebelumhya. Dihitung dengan menggunakan rumus:

Phase Difference= phase sekarang – phase sebelumnya;

3. Mapping delta phase ke +/- Pi interval

Mapping Delta Phase adalah menghitung perubahan phase pada interval Pi. Dengan menggunakan rumus:

Mapping Delta Phase = Phase Difference /3.14

4. Deviasi bin frequency dari the +/- Pi interval

Dengan menggunakan rumus:

deviasi bin frequency = Over sampling* Phase Difference /(2x3.14).

5. Menghitung partials' frequency yang benar partials' frequency ke n = n* deviasi bin frequency Phase Difference * deviasi binfrequency;

c. Menyimpan magnitude dan true frequency kedalam analysis arrays

Hasil proses perhitungan tersebut disimpan ke dalam sebuah array untuk dilakukan proses selanjutnya.

d. Proses Pitcshift

Proses pitchshift dilakukan dengan cara melakukan perkalian terhadap magnitude dan true frequency dengan factor pitchift yang telah diatur.

e. Proses Sinthesis

Proses synthesis adalah teknik untuk menghasilkan suara.

Proses synthesis

1. Mendapatkan magnitude dan frekuensi
2. Mengurangi bin mid frekuensi

3. Mendapatkan bin frekuensi dari frekuensi deviasi
4. Mengambil over sampling
5. Mengakumlasi delta phase untuk mendapatkan bin phase
6. Mendapatkan real dan imag part dan re-interleave
7. Proses inverse transform

f. Output Accumulator

Melakukan proses akumulasi frame dan menyusunya sesuai dengan panjangnya frame pada saat proses frame blocking.

g. Play sinyal

Hasil dari Output Accumulator di play dengan menggunakan engine native library agar menghasilkan suara yang telah dilakukan tranposisi dengan FFT.

h. Contoh Perhitungan Manual *Fast Fourier Transform* (FFT)

Dimisalkan Fast Fourier Transform untuk melakukan pengolahan sample digital dari sebuah file music dengan menggunakan parameter sebagai berikut :

FFT Frame Size = 4

Oversampling = 4

Sample Rate =44100

Maksimum Panjang Frame = 8192

Dengan 4 sample dengan nilai berturut-turut sebagai berikut :

-100,50, -55,45, -65. Sebe lum dilakukan perhitungan maka dilakukan proses inisialisasi awal :

a) Mencari Framesize2 = FFT Frame Size /4 = 4/2=2

b) Mencari Step size = FFT Frame Size / Oversampling=4/4=1

c) Mencari Frekuensi perbin =Sample rate/ FFT Frame Size = 44100/4=11025

d) Mencar Expt =(2 x 3.14) x (1/4) =1.5707963267948966

e) Mencari Infifolatency = FFT Frame Size -Step size = 4-1=3

f) Mencari Grover = infifolatency =3

1. Proses Windowing dengan menggunakan rumus.

Windowing = -0.5xCos(2.0x3.14x n : FFT FrameSize+0.5.

Maka Menghasilkan :

Windowing ke 0= -0.5 x cos(2 x 3.141592653589793x0.0 :4.0) + 0.5=0.0

Windowing ke 1 = -0.5 x cos (2 x 3.141592653589793x1.0 : 4.0) + 0.5

= 0.49999999999994

Windowing ke 2= -0.5 x cos(2 x 3.141592653589793x2.0 :4.0) + 0.5= 1.0

Windowing ke 3= -0.5 x cos(2 x 3.141592653589793x3.0 :4.0)

+ 0.5=0.5000000000000001

2. Deinterlace Buffer (Mencari real dan magnitude)

Real Ke n genap = Sample x Windowing ke n (n adalah banyaknya Frame size)

Imag ke n ganjil (diberi nilai 0) = 0.

Maka akan menghasilkan :

Real ke 0=0.0x0.0=0.0

Imag ke 1 =0.0f

Real ke 2=0.0x0.49999999999999994=0.0

Imag ke 3 =0.0f

Real ke 4=0.0x1.0=0.0

Imag ke 5 =0.0f

Real ke 6=-100.0x0.5000000000000001=-50.000000000000014

Imag ke 7 =0.0f

Menghitung ulang real dan imag dengan n adalah Framesize2

Real ke 0 =-50.0

Real ke 1 =-2.185569428547751E-6

Real ke 2 =50.

Image 0 =0.0

image 1 =-50.0

image 2 =0.0

3. Menghitung Magnitude dan Phase

Magnitude = 2.0 x $\sqrt{(\quad)+(\quad)}$

Magnitude ke 0 =2.0 x $\sqrt{50.0 \text{ x} - 50.0 + 0.0 \text{ x } 0.0}$=100.0

Magnitude ke 1 =2.0 x

$\sqrt{2.185569428547751E-6x-2.185569428547751E-6+-50.0x-50.0}$

=100.0000000000001

Magnitude ke 2 =2.0 x $\sqrt{50.0 \text{ x } 50.0 + 0.0 \text{ x } 0.0}$=100

Phase = tan2(imag,real)

Phase 0=tan2(0.0,-50.0)=3.141592653589793

Phase 1=tan2(-50.0,-2.185569428547751E-6)=-1.5707963705062853

Phase 2=tan2(0.0,50.0)=0.0

4. Menghitung Phase Difference

   Phase Difference = Phase – LastPhase

   Karena lastphase belum ada, maka lastphase dinisialisasi dengan angka 0, maka :

   Phase Difference 0 =3.141592653589793-0.0=3.141592653589793

   Phase Difference1=-1.5707963705062853-0.0=-1.5707963705062853

   Phase Difference2=0.0-0.0=0.0

5. Mengurangi perbedaan phase sesuai expct yang sudah didefinisikan

   Phase Difference (Phase Difference - n (framesize2 ke n) x expt)

   Phase Difference 0 =3.141592653589793 – (0x1.5707963267948966) = 3.141592653589793

   Phase Difference 1 =3.141592653589793 – (1x1.5707963267948966) =-3.141592697301182

   Phase Difference 2=-3.141592697301182 – (2x1.5707963267948966) =-3.141592653589793

6. Mapping delta phase ke +/- Pi interval

   Mapping delta phase ke +/- Pi interval : Phase Difference3.141592697301182

   Maka :

   Mapping delta phase 0 =3.141592653589793 : 3.141592653589793=1

   Mapping delta phase 1 =-3.141592697301182 : 3.141592653589793=-1

Mapping delta phase 2 =-3.141592653589793 : 3.141592653589793=-1

Maka aka didapatkan ulang phase difference :

Phase Difference 0=3.141592653589793x2.0=-3.141592653589793

Phase Difference 1=3.141592653589793x-2.0=3.1415926098784044

Phase Difference 2=3.141592653589793x-2.0=3.141592653589793

7. Mendapatkan deviasi dari bin frequensi dari +/- Pi interval

   Dihitung dengan rumus : Over Sampling x Phase Difference / (2.0 x)

   Maka :

   Deviasi 0 =4x-2.0:(2 x3.141592653589793)=-2.0

   Deviasi 1=4x 1.9999999721724657 : (2x3.141592653589793)=

   1.9999999721724657

   Deviasi 2=4x2.0:(2 x3.141592653589793)=2.0

8. Menghitung k-th true frequency

   True Frequency = n x freqPerBin + Deviasi x freqPerBin

   True Frequency 0=0x11025.0+-22050.0x11025.0=-22050.0

   True Frequency 1 =1x11025.0+33074.99969320143x11025.0

   =33074.99969320143

   True Frequency 2=2x11025.0+44100.0x11025.0=44100.0

9. Perkalian pitcshift

   Perkalian Pitchsift = True Frequency x pitchshift

   Pitchshift ke 0 =-22050.0x1.2=-26460.002

   Pitchshift ke 1 =33075.0x1.2=39690.0

   Pitchshift ke 2 =44100.0x1.2=52920.004

10. Mengurangi bin frekuensi tengah

    Bin mid frekuensi = Pitchshift n – (n x freqPerBin)

Bin mid frekuensi ke 0= -26460.002 – (0x11025.0) = -26460.002

Bin mid frekuensi ke 1= -26460.002 – (1x11025.0)=28665.0

Bin mid frekuensi ke 2= 28665.0 - (2 x11025.0) =30870.00390625

11. Mendapatkan Bin deviasi dari frekuensi deviasi

    Bin Deviasi = Bin deviasi : freqPerBinBin Deviasi ke 0

    = -26460.001953125 : 11025.0 = -2.4000001771541952

    Bin Deviasi ke 1 = -2.4000001771541952: 11025.0 = 2.6

    Bin Deviasi ke 2 = 2.6: 11025.0=2.80000035430839

12. Membawa over sampling ke dalam perhitungan

    Bin Deviasi n=2.0x3.141592653589793xBinDeviasi/Over sampling

    Ke 0= 2.0 x3.141592653589793 x-3.769911462580911:11025.0

    =-3.769911462580911

    Ke1=2.0x3.141592653589793x4.084070449666731:11025.0

    =4.084070449666731

    Ke 2= 2.0 x3.141592653589793 x4.398230271572028:11025.0

    =4.398230271572028

13. Menambahkan phase overlape

    phase overlape ke 0=0x1.5707963267948966=-3.769911462580911

    phase overlape ke 1=1x1.5707963267948966=5.654866776461628

    phase overlape ke 2=2x1.5707963267948966=7.539822925161821

14. Mengakumulasi delta phase untuk mendapatkan bin phase

    bin phase e ke 0=-3.7699115+-3.769911462580911=-3.7699115

    bin phase ke 1=5.6548667+5.654866776461628=5.6548667

bin phase 2=7.539823+7.539822925161821=7.539823

15. Mendapatkan real dan imag dan reinterleave

    Untuk yang bagian genap menggunakan rumus = Magnitude x Cos(phase). Sedangkan untuk yang ganjil menggunakan rumus Magnitudex sin(phase).Maka akan menghasilkan :

    Phase ke 0=100.0x cos-3.769911527633667=-80.90168

    Phase ke 1=100.0x sin-3.769911527633667=-80.90168

    Phase ke 2=100.0x cos5.654866695404053=80.901695

    Phase ke 3=100.0x sin5.654866695404053=80.901695

    Phase ke 4=100.0x cos7.539823055267334=30.901634

    Phase ke 5=100.0x sin7.539823055267334=30.901634

16. Melakukan proses windowing ulang

    Windowing : = -0.5 x cos(2.0 x x n / FFT FrameSize)+ 0.5

    Windowing ke 0= 0.5 x cos (2.0 x3.141592653589793x0:4+0.5=0.0

    Windowing ke 1= 0.5 xcos

    (2.0 x3.141592653589793x1:4+0.5=0.49999999999999994

    Windowing ke 2= 0.5 x cos (2.0 x3.141592653589793x2:4+0.5=1.0

    Windowing ke 3= 0.5 x cos

    (2.0 x3.141592653589793x3:4+0.5=0.5000000000000001

17. Output dari proses FFT

    Outputke 0 =0.0 + 2.0 x0.0x30.90165: (2x4)=0.0

    Output ke 1 =-6.6280985 + 2.0 x0.49999999999999994x-53.024788: (2x4)=-6.6280985

Outputke 2 =-32.725433 + 2.0 x1.0x-130.90173: (2x4)=-32.725433

Output ke 3 =-21.322731 + 2.0 x0.5000000000000001x-170.58185: (2x4)=-21.322731

3.6 Desain interface

Desain interface adalah rancangan tampilan sistem yang akan dibuat pada penelitian ini.

Gambar 3.7 Desain Interface

Sesuai rancangan antarmuka pada gambar 3.7 user bisa melakukan penambahan dan penghapusan file yang ditampilkan di *datagrid,* dan memainkanya dengan memberikan efek transposisi nada sesuai dengan pilihan parameter yang mempengaruhi hasil tranposisi suara yaitu :

Oversampling: oversampling pada combobox tersebut adalah 8, 16, dan 32.

Sample Rate : adalah panjangnya rate, pada combobox tersebut berisi 44100, 44800 dan 92.000

Pitch factor : adalah besaran pitch untuk melakukan proses tranposisi, besaran pitch tersebut antara 0,5-2.0.

3.7 Analisis Kebutuhan Sistem

Aplikasi Transposisi Nada ini menggunakan Nada suara apikasi ini terdiri dari beberapa komponen yang dapat digambarkan dalam suatu model .

3.6.1 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 utama yang digunakan untuk mempuat aplikasi ini adalah VLCJ.

c. Netbeans

Netbeans digunakan sebagai editor pembuatan aplikasi.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas implementasi dari proses analisis dan perancangan pada bab III. Proses implementasi adalah proses merubah analisis dan perancangan ke dalam bahasa pemrogaman komputer dengan melakukan proses *development* aplikasi. Untuk melakukan pengembangan aplikasi transposisi nada dengan menggunakan algoritma *Fast Fourier Transform* menggunakan alat dan bahan sebagai berikut.

4.1 Deskripsi Aplikasi

Aplikasi dibangun dengan menggunakan bahasa pemrogaman java, untuk menuliskan kode program menggunakan IDE Netbeans 8.0. Aplikasi yang dihasilkan diharapkan mampu melakukan transposisi suara, yakni melakukan perubahan nada suara asli ke nada suara yang lebih rendah atau ke nada yang lebih tinggi. Untuk melakukan transposisi tersebut perlu melakukan ekstraksi ciri byte file audio yang sedang dimainkan. Kemudian melakukan pengolahan dengan menggunakan algoritma Fast Fourier Transform. Output dari Fast Fourier transform merupakan sample byte yang telah dilakukan proses penurunan atau peninggian nada, dan aplikasi akan memainkan file yang telah diolah tersebut.

4.2 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)

   Pengembangkan 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)

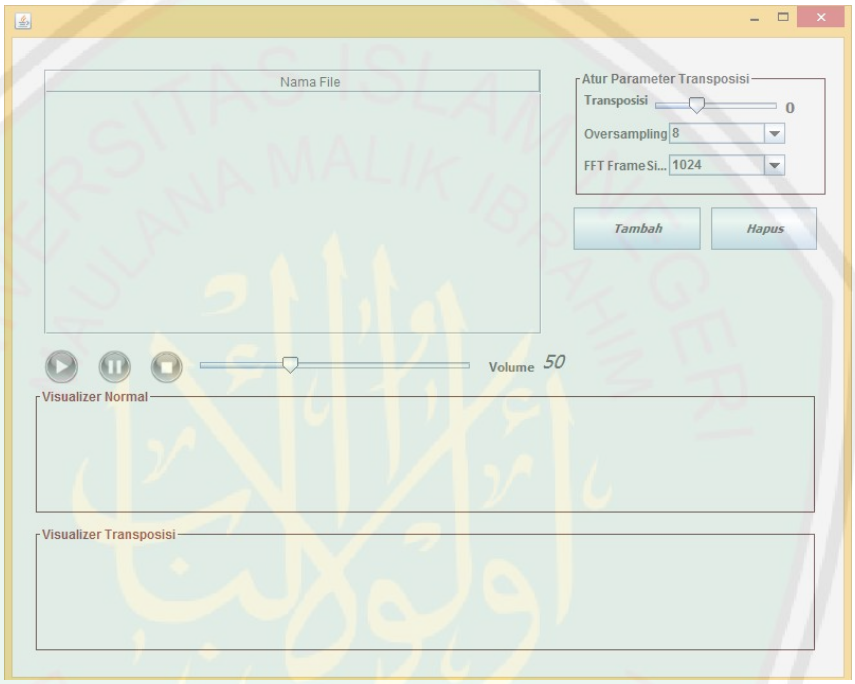
   Mengembangkan aplikasi dengan menggunakan bahasa pemrogaman java dan untuk memudahkan proses pembuatan aplikasi menggunakan IDE Netbeans 8.0 dan JDK 1.7. untuk memperindah layout menggunakan editor gambar photoshop dan corel draw. Sedangkan untuk melakukan dokumentasi dan pembuatan laporan menggunakan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

3. Kebutuhan Data

   Data yang digunakan untuk penelitian ini adalah data file audio yang mempunyai format audio.

4.3 Hasil Implementasi Analisis Desain Sistem

Pada bab III sudah dijelaskan bahwasanya user bisa melakukan pemutaran file audio, memberikan efek pitchshift dan melihat grafik sinyal. Gambar 4.1 adalah implementasi tersebut.



Gambar 4.1 Aplikasi Transposisi Suara

Fungsi-fungsi menu pada gambar 4.1 adalah sebagai berikut :

1. Button Tambah, Untuk menambahkan format audio ke dalam aplikasi. Format audio yang bisa ditambahkan ke aplikasi adalah semua format audio digital.
2. Button Hapus, Untuk menghapus file audio yang telah ditambahkan ke table data audio. Untuk menghapus audio yang telah ditambahkan dengan cara melakukan klik terlebih dahulu pada table kemudia menekan button hapus.

3. Table Nama File, Untuk menampilkan file audio yang telah ditambahkan ke dalam aplikasi.
4. Button Play, Untuk memainkan file audio yang telah ditambahkan ke aplikasi.
5. Slider Volume, untuk mengecilkan atau membesarkan suara file audio yang diputar.
6. Visualizer Transposisi, Untuk menampilkan waveform file audio yang telah dilakukan proses transposisi dengan menggunakan algoritma *Fast Fourier Transform.*
7. Visualizer Normal

   Visualizer Normal, Untuk menampilkan waveform file audio asli, yang belum dilakukan proses transposisi dengan menggunakan algoritma *Fast Fourier Transform.*
8. Atur Parameter Transposisi

   Digunakan untuk melakukan pengaturan parameter guna memperoleh kombinasi parameter yang menghasilkan transposisi suara dengan kualitas yang paling jelas. Ada beberapa parameter yang harus disetting, yaitu:

   a. Parameter Transposisi.

      Parameter transposisi adalah parameter yang digunakan untuk melakukan pengaturan transposisi. Nilai batas bawah adalah 0.5 sedangkan nilai batas atas adalah 2.0. sedangkan defaultnya adalah 1, yang berarti tidak melakukan proses transposisi suara. Jika nilai transposisi dibawah 1-0.5, maka aplikasi melakukan transposisi dengan

ke nada rendah. Sebaliknya jika nilai transposisi diatas 1-2.0 maka aplikasi melakukan transposisi ke nada tinggi.

b. Parameter Oversampling

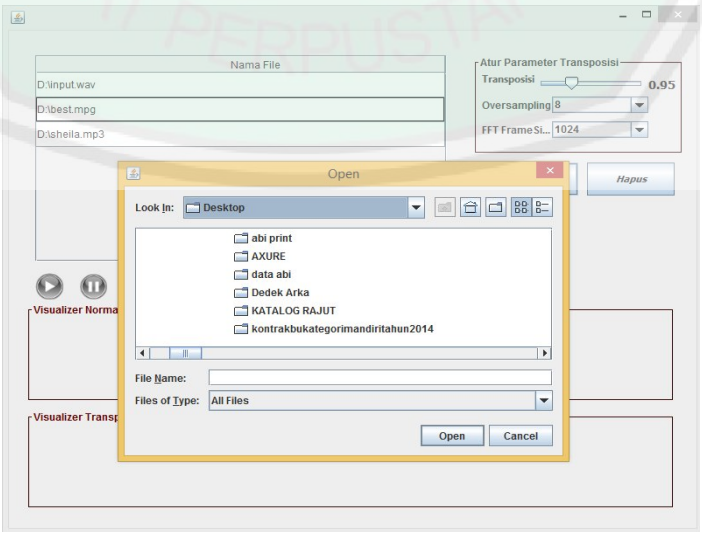
Parameter oversampling digunakan untuk membatasi overlape diantara frame FFT. Secara teori nilai minimalnya adalah 4, sedangkan untuk kualitas terbaik direkomendasikan memberikan nilai 32.

c. Parameter Frame Size

Parameter Frame Size digunakan untuk menentukan ukuran frame yang digunakan fast fourier transform untuk melakukan pemrosesan, nilai yang digunakan biasanya adalah 1024, 2048 dan 4096.

4.4 Hasil Implementasi Penerapan Tranpsosisi Suara

Proses transposisi suara dimulai dengan penginputan file audio ke aplikasi, file yang bisa diinputkan adalah semua jenis file audio. Proses penginputan file ke aplikasi sebagaimana pada gambar 4.2



Gambar 4.2 Input file audio ke aplikasi

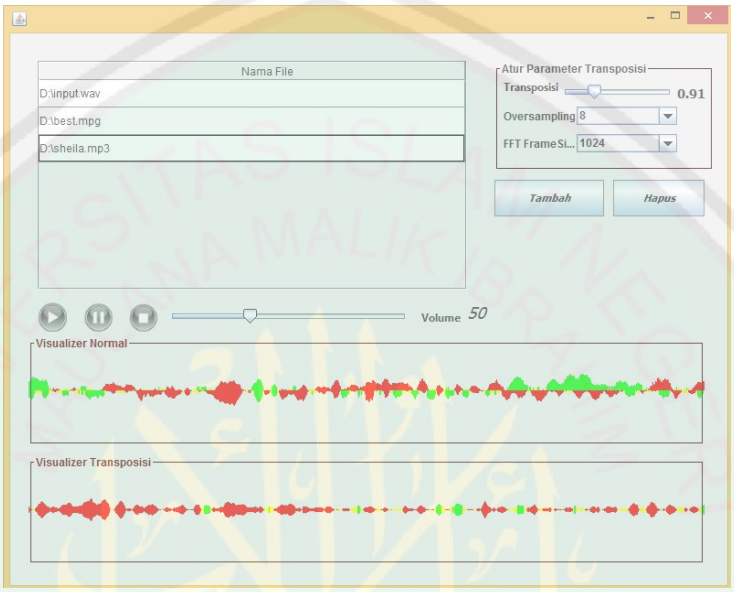
Untuk melakukan penambahan file menggunakan class JFileChooser, kemudian mengambil url tersebut dan menampilkan url tersebut ke sebuah table agar bisa dimainkan. untuk *sourcecode* penambahan file audio sebagaimana pada gambar 4.3.

```
JFileChooser fc = new JFileChooser();
//Melakukan Penambahan file audio ke aplikasi
FileNameExtensionFilter filter = new
FileNameExtensionFilter("MPG audio file", "MPG");
        fc.addChoosableFileFilter(filter);
        int returnVal = fc.showOpenDialog(null);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File songFile = fc.getSelectedFile();
            Object[] data = {songFile.toString()};
            defaultTableModel.addRow(data);
        }
```

Gambar 4.3 sourcecode penambahan file audio

Setelah melakukan penambahan file audio user bisa memainkan file audio dan memberikan efek transposisi dengan cara menggeser *slider* transposisi ke kiri atau kekanan. Penggesaran slider kekanan akan menaikkan suara file audio ke nada yang lebih tinggi, sedangkan penggeseran slider kekiri akan menurunkan suara file audio ke nada yang lebih rendah. Gambar 4.4 adalah contoh tampilan aplikasi ketika dilakukan pemberian efek transposisi nada ke nada yang lebih rendah, dengan memberi nilai transposisi sebesar 0.91. pemberian efek transposisi tersebut selain merubah suara ke

nada yang lebih rendah juga merubah visualizernya, pada gambar 4.4 juga terlihat jelas bahwa terjadi perbedaan tingggi waveform antara visualizer normal dan visualizer transposisi.



Gambar 4.4 Tampilan aplikasi ketika memainkan file audio

4.5 Pembahasan Penerapan Algoritma Fast Fourier Transform

Algoritma fast fourier transform digunakan untuk mengubah inputan suara menjadi frekuensi. frekuensi tersebut kemudian akan disesuaikan dengan sistem sehingga didapatkan nada default. Proses pengolahan sinyal digital dilakukan setelah proses pembacaan file audio. Kemudian melakukan proses *feature extraction*, sebuah proses yang mengkonversi sinyal suara menjadi beberapa parameter sehingga dihasilkan data yang berdimensi lebih kecil.

Setelah dilakukan proses *feature ekstraction*, langkah selanjutnya adalah proses frame blocking. Proses *Frame Blocking* yaitu melakukan blok

terhadap sinyal-sinyal menjadi frame-frame N sampel, dengan frame-frame berdekatan dengan spasi M (M < N). Frame pertama terdiri dari N sampel pertama. Frame kedua dengan M sampel setelah frame pertama, dan overlap dengan N–M sampel. Dengan cara yang sama, frame ketiga dimulai 2M sampel setelah frame pertama (atau M sampel setelah frame kedua) dan overlap dengan N– 2M sampel. Proses ini berlanjut hingga semua sinyal suara dihitung dalam satu atau banyak frame. Nilai untuk N dan M adalah N = 256 dan M =100.

Jadi, proses frame tersebut dilakukan secara terus-menerus hingga semua sinyal dapat terproses. Selain itu, proses ini umumnya dilakukan secara overlapping untuk setiap frame-nya. Panjang daerah overlap yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang frame.

Setelah mendapatkan byte sample dari proses ekstraksi ciri dan frame blocking, maka langkah selanjtunya yang dilakukan adalah melakukan pengolahan byte sample yang telah terbagi menjadi beberapa frame tersebut dengan algoritma fast fourier transform. Dalam penelitian ini, algoritma fast fourier transform diimplementasikan menjadi sebuah fungsi yang melakukan perubahan sinyal (transposisi nada) dalam sebuah class bernama pitchshift.java. fungsi tersebut memerlukan sebuah inputan besaran pitch (0 – (-2)) dan sample sinyal hasil dari *frame blocking* yang akan diolah dengan tipe data short. Langkah awal yang dilakukan adalah menginisiasi semua variabel yang dibutuhkan algoritma Fast Fourier Transform.

```
private float pitchShift = (float) 1.0;
private  long fftFrameSize =2048;
private  long osamp = 16;//fators of 2
private float sampleRate = 44100;
private final int MAX_FRAME_LENGTH = 8192;
private final float[] gInFIFO = new float[MAX_FRAME_LENGTH];
private final float[] gOutFIFO = new float[MAX_FRAME_LENGTH];
private final float[] gFFTworksp = new float[2*MAX_FRAME_LENGTH];
private final float[] gLastPhase = new float[MAX_FRAME_LENGTH/2+1];
private final float[] gSumPhase = new float[MAX_FRAME_LENGTH/2+1];\
private final float[] gOutputAccum = new float[2*MAX_FRAME_LENGTH];
private final float[] gAnaFreq = new float[MAX_FRAME_LENGTH];
private final float[] gAnaMagn = new float[MAX_FRAME_LENGTH];
private float[] gSynFreq = new float[MAX_FRAME_LENGTH];
private float[] gSynMagn = new float[MAX_FRAME_LENGTH];
private int gRover = 0;
private double magn, phase, tmp, window, real, imag;
private final double freqPerBin, expct;
private int i,k, qpd, index, inFifoLatency, stepSize,
fftFrameSize2;
```

Gambar 4.5 Source code inisialisasi variabel

Penjelasan variabel-variable yang digunakan pada source code diatas adalah sebagai berikut :

a. PitchShift: nilai yang digunakan untuk menentukan besaraan pitch yang diinginkan. Valuenya antara 0.5 sampai 2. Dimana 1 untuk nilai normal.

b. fftFrameSize : ukuran FFT Frame size yang digunakan untuk melakukan pemrosesan, biasanya bernilai 1024, 2048 dan 4096.
c. Osamp atau oversampling, factor yang menentukan overlape antara frame STFT yang berdekatan. Nilai minimalnya adalah 4 dan nilai yang direkomendasikan adalah 32 untuk kualitas yang terbaik.
d. sample rate, banyaknya jumlah sample dalam bentuk frekuensi yang diambil dalam satuan waktu. Dan dihitung dalam satuan Hz. Biasanya 4100 Hz.
e. MAX_FRAME_LENGTH, batas maksimal panjang frame.
f. gInFIFO, variabel untuk menampung sample yang diiputkan. Berupa array dengan batas panjang array sama dengan nilai MAX_FRAME_LENGTH.
g. gOutFIFO, variabel yang menampung nilai dari hasil pengolahan proses pitchshift. Berupa array dengan batas panjang array sama dengan nilai MAX_FRAME_LENGTH.
h. gFFTworksp, variabel yang menampung nilai FFT yang sedang diproses, berupa array dengan nilai batas panjang 2 kali nilai MAX_FRAME_LEGTH.
i. GLastPhase, variabel untuk menampung phase terakhir.
j. gSumPhase, variabel untuk menampung jumlah phase.
k. gOutputAccum, variabel untuk menampung nilai dari proses accumulator.
l. gAnaFreq, array variabel untuk menampung frekuensi yang dianalisis.
m. gSynFreq, array variabel untuk menampung frekuensi yang disentisiskan,

n. gSynMagn, array variabel untul menampung magnitude yang disintesiskan,

o. gRover, array variable untuk menyimpan rover.

```
fftFrameSize2     = (int) (fftFrameSize/2);
stepSize          = (int) (fftFrameSize/osamp);
freqPerBin  = sampleRate/(double)fftFrameSize;
expct = 2.*Math.PI*(double)stepSize/(double)fftFrameSize;
inFifoLatency = (int) (fftFrameSize-stepSize);
```

Gambar 4.6 Inisialisasi variable tambahan

Setelah melakukan inisialisasi variabel, dilakukan juga inisialisasi variabel tambahan. Variabel-variabel tersebut adalah:

a. fftFramesize2

b. stepSize, variabel yang digunakan untuk menampung panjang output hasil dari pemrosesan FFT.

c. freqPerBin

d. expct

e. inFifoLatency, input First In First Out Latency

Pada class pitchshift.java, audi byte sample yang telah dihasilkan oleh proses frame blocking dilakukan proses windowing terlebih dahulu. Proses windowing ini bertujuan untuk mengurangi terjadinya kebocoran spectral atau aliasing yang mana merupakan suatu efek dari timbulnya sinyal baru yang memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek tersebut dapat terjadi karena rendahnya jumlah sampling rate atau karena

proses frame blocking yang menyebabkan sinyal menjadi discontinue. Penelitian ini menggunakan teknik Hamming Window yang menghasilkan sidolable level yang tidak terlalu tinggi (kurang lebih -43 dB). Selain itu, noise yang dihasilkan pun tidak terlalu besar (kurang lebih 1.36 BINS).

```
Melakukan proses windowing dan reinterleave
for (k = 0; k < fftFrameSize;k++) {
window = -
.5*Math.cos(2.*Math.PI*(double)k/(double)fftFrameSize
)+.5;
 gFFTworksp[2*k] = (float) (gInFIFO[k] * window);
 gFFTworksp[2*k+1] = 0.0f;
 }
```

Gambar 4.7 sourcecode proses windowing

Untuk tiap frame akan dilakukan proses FFT dengan window function Hamming, sehingga menghasilkan N buah data *RealOut* dan *Imaginer*. Begitu pula dengan ukuran data tiap frame yang ditentukan sendiri melalui setting pada aplikasi dengan nilai default sebesar 1024.

Hasil dari FFT tersebut adalah simetris, sehingga hanya N/2 buah data yang digunakan untuk proses selanjutnya. Dari N/2 buah data tersebut akan dihitung rata-rata nilai magnitude yang menjadi nilai dari frame tersebut. Karena sampling rate yang digunakan adalah sebesar 44100Hz, maka frekuensi tertinggi yang diperoleh adalah 22050Hz dengan pembagian

frekuensi sebesar 22050Hz/512 data = 43.07 Hz/data. Nilai frekuensi tersebut sesuai dengan batas frekuensi tertinggi yang dapat didengar oleh manusia.

```
for (k = 0; k <= fftFrameSize2; k++) {

/* deinterlace FFT buffer */
real = gFFTworksp[2*k];
imag = gFFTworksp[2*k+1];
/* menghtung magnitude and phase */
magn = 2.*Math.sqrt(real*real + imag*imag);
phase=Math.atan2(imag,real);

/* Menghitung perbedaan phase */
tmp = phase - gLastPhase[k];                    /* subtract
expected phase difference */
tmp -= (double)k*expct;
/* pemetaan delta phase into +/- Pi interval */
qpd = (int) (tmp/Math.PI);
                        if (qpd >= 0) qpd += qpd&1;
                        else qpd -= qpd&1;
                        tmp -= Math.PI*(double)qpd;
/* Mendapatkan deviasi dari bin frequency dari the +/- Pi
interval */
tmp = osamp*tmp/(2.*Math.PI);
/* menghitung the k-th  bagian frekuensi yang benar */
tmp = (double)k*freqPerBin + tmp*freqPerBin;
/* menyimpan magnitude dan frequency  yang benar ke dalam array
analisis*/
                        gAnaMagn[k] = (float) magn;
                        gAnaFreq[k] = (float) tmp;

`
```

Gambar 4.8 ***sourcecode Fast Fourier Transform***

```
for (k = 0; k <= fftFrameSize2; k++) {
/* mendapatkan magnitude dan  frekuensi dari array sinthesis*/
magn = gSynMagn[k];
tmp = gSynFreq[k];
/* mengurangi bin frequency tengah */
tmp -= (double)k*freqPerBin;
/* mendapatkan bin frekuesni dari freq deviation */
tmp /= freqPerBin;
tmp = 2.*Math.PI*tmp/osamp;
/* menambahkan overlap phase advance back in */
tmp += (double)k*expct;
/* mengakumulasikan delta phase untuk mendapatkan bin phase */
gSumPhase[k] += tmp;
tmp+"="+gSumPhase[k]);
phase = gSumPhase[k];
/* mendapatkan  real imag dan imag part dan re-interleave */
gFFTworksp[2*k] = (float) (magn*Math.cos(phase));
gFFTworksp[2*k+1] = (float) (magn*Math.sin(phase));
}
for(k=0; k < fftFrameSize; k++) {
window = -
.5*Math.cos(2.*Math.PI*(double)k/(double)fftFrameSize)+.5;
gOutputAccum[k] +=
2.*window*gFFTworksp[2*k]/(fftFrameSize2*osamp);
}
```

Gambar 4.9 *sourcecode Fast Fourier Transform* lanjutan

4.6 Uji Coba Aplikasi Transposisi Nada

Uji coba dilakukan untuk mengetahui apakah aplikasi yang dibuat telah sesuai dengan deskripsi aplikasi yang telah dijelaskan pada subbab 4.1. selain itu, uji coba juga dimaksudkan untuk mengtahui kombinasi parameter terbaik untuk melakukan proses transposisi suara dengan menggunakan algoritma Fast Fourier Transorm.

4.6.1 Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk menguji apakah sistem yang dikembangkan sesuai dengan apa yang deskripsi aplikasi dan tujuan penelitian. Pengujian ini digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

Pengujian dilakukan dengan menginputkan beberapa format file audio ke dalam aplikasi dan memainkanya. Kemudian memberikan efek transposisi nada ke nada rendah dan ke nada tinggi. Efek tranpsosisi nada ke nada tinggi berarti akan ada perubahan suara dengan nada yang lebih tinggi dari pada suara aslinya, sedangkan efek transposisi ke nada rendah berarti akan ada perubahan suara dengan nada yang lebih rendah daripada suara aslinya. Jika terjadi perubahan suara sesuai dengan efek yang

diberikan dan memberikan suara yang jernih maka aplikasi dianggap berhasil melakukan transposisi nada.

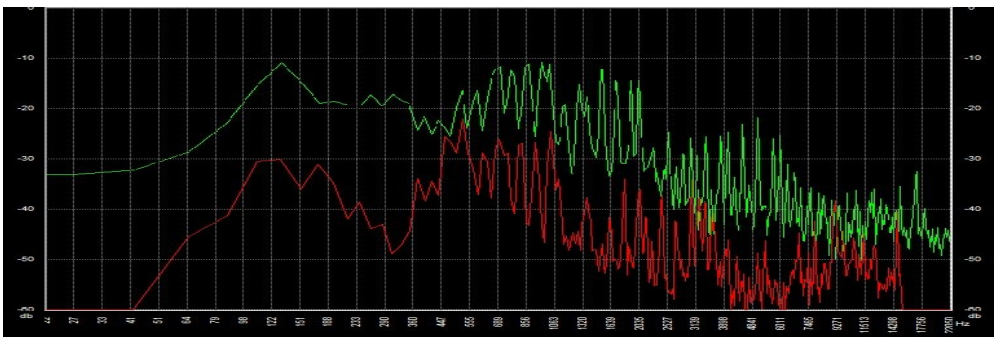
Untuk menghitung keberhasilan dan kegagalan dari algoritma Fast Fourier Transform sebagai berikut :

$$\text{Prosentase keberhasilan} = \frac{\textit{Jumlah Keberhasilan}}{\textit{Banyaknya Uji Coba}} \times 100$$

$$\text{Prosentase kegagalan} = \frac{\textit{Jumlah Kegagalan}}{\textit{Banyaknya Uji Coba}} \times 100$$

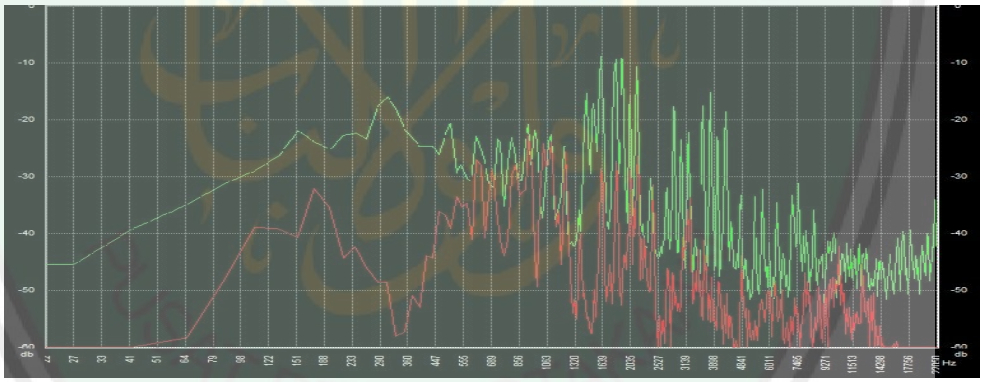
Pengujian dilakukan dengan cara mengambil 5 sample lagu yang berbeda dimana setiap lagu akan diberikan efek tranposisi yang berbeda.

Pengujian pertama dilakukan pada tanggal 8 Mei 2015. Sample lagu yang pertama aplikasi memberikan efek transposisi ke nada yang lebih tinggi dengan memberi nilai transposisi sebesar 1.06 Hasil dari transposisi yang dilakukan menghasilkan nada lagu yang jernih dan jelas, dengan menampilkan spectrum frekuensi yang berbeda, pada gambar 4.10 warna hijau menunjukkan spectrum frekuensi dari nada asli sedangkan warna merah menunjukkan spectrum frekuensi yang sudah di transposisi, dengan mengambil tiga sample nilai frekuensi menunjukkan nada berhasil dinaikkan dengan nilai frekuensi awal 179 dan nilai frekuensi setelah di transposisi 241.



Gambar 4.10 Ketika di transposisi 1.06

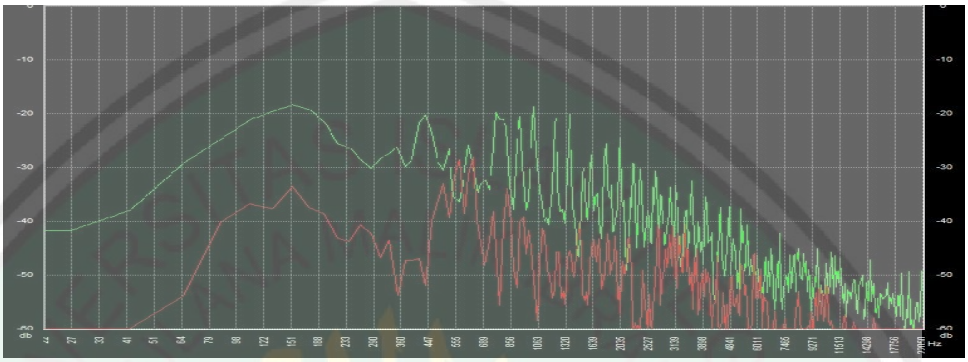
Pengujian kedua dilakukan pada tanggal 8 Mei 2015. Pada sample lagu yang kedua aplikasi memberikan efek transposisi ke nada yang lebih tinggi dengan memberi nilai transposisi sebesar 1.13 Hasil dari transposisi yang dilakukan menghasilkan nada lagu yang jernih dan jelas, dengan menampilkan spectrum frekuensi yang berbeda, pada gambar 4.11 warna hijau menunjukkan spectrum frekuensi dari nada asli sedangkan warna merah menunjukkan spectrum frekuensi nada yang sudah di transposisi, dengan mengambil tiga sample nilai frekuensi menunjukkan nada berhasil dinaikkan dengan nilai frekuensi awal 388 dan nilai frekuensi setelah di transposisi 475.



Gambar 4.11 Ketika di transposisi 1.13

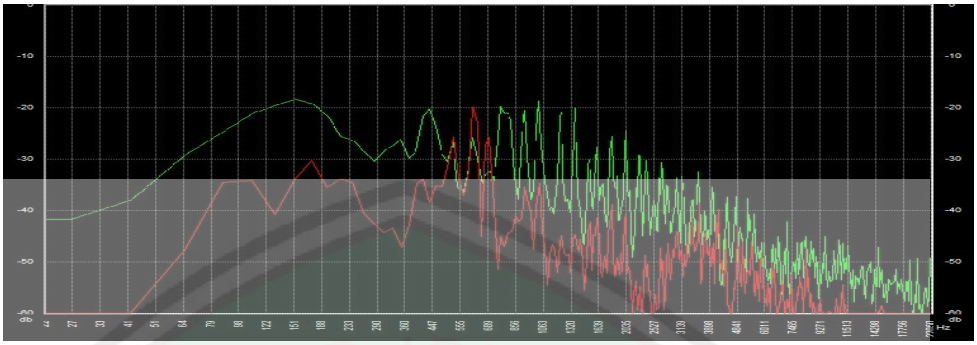
Pengujian ketiga dilakukan pada tanggal 8 Mei 2015. Pada sample lagu yang ketiga aplikasi memberikan efek transposisi ke nada yang lebih tinggi dengan memberi nilai transposisi sebesar 1.16 Hasil dari transposisi yang dilakukan menghasilkan nada lagu yang jernih dan jelas, dengan menampilkan spectrum frekuensi yang berbeda, pada gambar 4.12 warna hijau menunjukkan spectrum frekuensi dari nada asli sedangkan warna merah menunjukkan spectrum frekuensi nada yang sudah di transposisi,

dengan mengambil tiga sample nilai frekuensi menunjukkan nada berhasil dinaikkan dengan nilai frekuensi awal 513 dan nilai frekuensi setelah di transposisi 553.



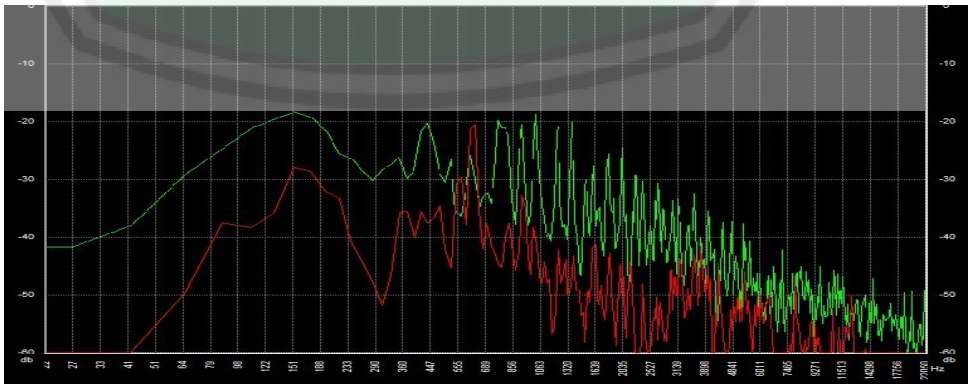
Gambar 4.12 Ketika di transposisi 1.16

Pengujian keempat dilakukan pada tanggal 8 Mei 2015. Pada sample lagu yang keempat aplikasi memberikan efek transposisi ke nada yang lebih rendah dengan memberi nilai transposisi sebesar 0.95 Hasil dari transposisi yang dilakukan menghasilkan nada lagu yang jernih dan jelas, dengan menampilkan spectrum frekuensi yang berbeda pada gambar 4.13 warna hijau menunjukkan spectrum frekuensi dari nada asli sedangkan warna merah spectrum frekuensi nada yang sudah di transposisi, dengan mengambil tiga sample nilai frekuensi menunjukkan nada berhasil diturunkan dengan nilai frekuensi awal sebesar 458 Hz dan nilai frekuensi setelah di transposisi sebesar 361 Hz .



Gambar 4.13 Aplikasi ketika di transposisi 0.95

Pengujian kelima dilakukan pada tanggal 8 Mei 2015. Pada sample lagu yang kelima aplikasi memberikan efek transposisi ke nada yang lebih rendah dengan memberi nilai transposisi sebesar 0.89 Hasil dari transposisi yang dilakukan menghasilkan nada lagu yang tidak jelas dan sedkit bergemuruh, dengan menampilkan spectrum frekuensi yang berbeda pada gambar 4.14 warna hijau menunjukkan spectrum frekuensi dari nada asli sedangkan warna merah menunjukkan spectrum frekuensi yang sudah di transposisi, dengan mengambil tiga sample nilai frekuensi menunjukkan nada berhasil diturunkan dengan nilai frekuensi awal 596 Hz dan nilai frekuensi setelah di transposisi sebesar 508 Hz.



Gambar 4.14. Aplikasi ketika di transposisi 0.89

Hasil dari uji coba yang dilakukan menghasilkan nada lagu yang jelas dan jernih batas maksimum dari besaran parameter transposisi ke nada yang lebih tinggi adalah 1.25 sedangkan batas maksimum dari parameter transposisi ke nada yang lebih rendah adalah 0.93

$$Prosentase\ Keberhasilan = \frac{6}{100} \times 100\% = 60\%$$

$$Prosentase\ Keberhasilan = \frac{4}{100} \times 100\% = 40\%$$

Dari hasil uji coba aplikasi mendapatkan hasil dengan prosentase keberhasilan 60% dan prosesntase kegagalan 40%. Aplikasi mampu memainkan file audio yang diinputkan tetapi dengan transposisi tertentu.

Tabel 4.1 Tabel File Uji Coba

No	Nama File	Format File	Transposisi	Durasi	Ukuran File	Keterangan
1	Wali_aku bukan bang toyyib	Wav	1.06	00:05	326 KB	Sukses
2	CitaCitata_Sakitnya tuh disini.wav	Wav	1.16	00:15	325 KB	Sukses
3	Acnes_Bukan milikmu lagi.wav	Wav	1.13	00:15	362 KB	Sukses
4	Lucky laky_Aku bukan supermen	Wav	1.25	00:25	332 KB	Sukses
5	Juwita bahar_Bukak titik jos	Wav	1.39	00:05	448 KB	Gagal
6	Risma_Cinta satu malam	Wav	1.40	00.15	363 KB	Gagal
7	Ungu_Hampa hatiku	Wav	0.95	00:25	327 KB	Sukses
8	Olga_Hancur hatiku	Wav	0.93	00:25	362 KB	Sukses
9	ST12_ aku terjatuh	Wav	0.89	00:25	336 KB	Gagal
10	Roma_Adu domba	Wav	0.84	00:15	377 KB	Gagal

4.6.2 Uji Coba Parameter Transposisi

Uji coba parameter dimaksudkan untuk mengetahui kombinasi parameter terbaik untuk melakukan proses transposisi nada. Parameter yang dilakukan uji coba adalah besaran pich, oversampling dan FFT Framesize. Uji coba dilakukan dengan memainkan file audio, kemudian memberikan efek transposisi suara dan merubah kombinasi oversampling dan FFT Framesize sebagaimana pada tabel dibawah ini.

4.2 Table Uji coba Parameter

No	OS	Framesize	Keterangan
1	8	1024	File bisa dimainkan dan aplikasi mampu melakukan transposisi suara dengan suara yang cukup jernih.
2	8	2048	File bisa dimainkan dan aplikasi mampu melakukan transposisi suara dengan suara yang kurang jernih.
3	8	4096	File bisa dimainkan , aplikasi tidak mampu melakukan proses transposisi suara.
4	16	1024	File bisa dimainkan dan aplikasi mampu melakukan transposisi suara dengan suara yang cukup jernih.
5	16	2048	File bisa dimainkan dan aplikasi mampu melakukan transposisi suara dengan suara yang kurang jernih.
6	16	4096	File bisa dimainkan , aplikasi tidak mampu melakukan proses transposisi suara.
7	32	1024	File bisa dimainkan dan aplikasi mampu melakukan transposisi suara dengan suara yang cukup jernih.
8	32	2048	File bisa dimainkan dan aplikasi mampu melakukan transposisi suara dengan suara yang kurang jernih.
9	32	4096	File bisa dimainkan , aplikasi tidak mampu melakukan proses transposisi suara.

Keterangan :

OS : Oversampling

Framesize : FFT Framesize

Dari hasil uji coba dapat disimpulkan bahwa parameter oversampling tidak terlalu mempengaruhi kualitas hasil proses transposisi suara. Sebaliknya parameter FFT Framesize sangat mempengaruhi kualitas transposisi suara. Hal ini dapat dibuktikan ketika dilakukan perubahan nilai oversampling aplikasi tetap mampu melakukan proses transposisi suara dengan baik sebaliknya jika dilakukan perubahan nilai parameter FFT Framesize suara hasil transposisi menjadi tidak jelas dan bergemuruh. Dari hasil uji coba yang dilakukan parameter terbaik untuk melakukan tranpsosi suara adalah dengan parameter oversampling 8 dan FFT Framesize 1024, Oversampling 16 dan FFT Framesize 1024 dan oversampling 32.

4.7 Kajian Islam

Kegunaan utama aplikasi yang dibuat oleh peneliti adalah untuk melakukan pengolahan suara agar bisa dilakukan perubahan tinggi nada sebuah audio digital. Dalam agama islam juga mengenal ilmu olah suara, yang sering dikenal dengan seni tilawatil qur'an atau juga dikenal dengan istilah nagham.

Kata nagham secara etimologi paralel dengan kata ghina yang bermakna lagu atau irama. Secara terminologi nagham dimaknai sebagai

membaca Al Quran dengan irama (seni) atau suara yang indah dan merdu atau melagukan Al Quran secara baik dan benar tanpa melanggar aturan-aturan bacaan.

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. (al- Tho'i, 2005:37) 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 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.
2. 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. (al- Tho'i, 2005:21)

   Dari beberapa pendapat para ulama yang telah disebutkan di atas, 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. (al-Qattan,

1973:126) 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 harus memulai dari tingkat pemeliharaan tubuh, khususnya alat yang dengan pernafasan. Tilawah al-Qur'an akan lebih banyak nafas dan suara. Organ pernafasan yang perlu diperhatikan adalah pada bagian perut, dada, leher, dan bagian kepala. (al-Qayum, 2001:17).

Untuk memiliki pernafasan 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:

مْزَيِّنُوا الْقُرْآنَ بِأَصْوَاتِكُزيه انقران

Artinya:"Perindahlah al-Qur'an dengan suara kalian (al-Rahman, 1980:283).

Hadist tersebut juga didukung oleh hadist yang diriwayatkan oleh abu hurairah, bahwasanya nabi selalu membaca al-qur'an dengan memperindah suaranya. Dari Abu Hurairah radhiyallahu'anhu, beliau berkata: Aku pernah mendengan Rasulullah shallallahu 'alaihi wasallam bersabda: "Tidaklah Allah mendengarkan sesuatu, tidaklah Nabi mendengarkan sesuatu, kecuali suara yang indah ketika membaca al-Qur'an dan menyaringkannya. (Muttafaqun 'alaih). Selain itu juga didukung oleh Firman Allah dan Surah Al-Muzammil ayat 4 :

أَوْ زِدْ عَلَيْهِ وَرَتِّلِ ٱلْقُرْءَانَ تَرْتِيلًا ۝٤

Artinya:"Bacalah Al Quran itu secara tartil".

Memperindah suara ketika membaca Al-qur'an selain anjuran Allah dan Rasulnya juga merupakan bagian yang tak terpisahkan dari eksistensi manusia sebagai makhluk yang berbudaya yang memiliki cipta, rasa, dan karsa. Rasa yang melahirkan seni (termasuk nagham) merupakan bagian integral kehidupan manusia yang didorong oleh adanya daya kemauan dalam dirinya. Kemauan rasa itu sendiri timbul karena didorong oleh karsa rohaniah dan pikiran manusia.

Apresiasi terhadap seni Al Quran semakin tenggelam karena membutuhkan modal suara. Modal ini lebih merupakan hak perogratif Allah untuk diberikan kepada yang dikehendaki-Nya. Selain itu, ilmu nagham Al Quran sangat sulit ditransfer ke dalam notasi angka atau nada.

Dan karena sifat eksklusifisme inilah kemudian yang "memaksa" bahwa metode *sima'i*, *talaqqi*, dan *musyahafah* merupakan satu-satunya cara dalam mentransmisikan lagu-lagu Al Quran.

Dalam tatanan seni baca Alquran, tingkatan nada dikenal ada empat tahap, yakni qarar (rendah), nawa (sedang), jawab (tinggi), dan jawabul jawab (sangat tinggi). Dan terbagi menjadi beberapa jenis seni membaca Alquran, seperti *Nahawan*, *Bayati*, *Hijaz*, *Shaba*, *Ras*, *Jiharkah*, *Syika*, dan lainnya. Dan tidak semua tingkatan nada tersebut mampu dilagukan oleh semua manusia.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil uji coba dan pembuatan aplikasi transposisi suara dengan menggunakan algoritma *Fast Fourier Transform* maka dapat diambil kesimpulan bahwa:

1. Algoritma *Fast Fourier Transform* mampu melakukan proses transposisi suara yang diujicobakan dengan hasil suara yang cukup jernih.
2. Parameter FFT Frame Size dangat mempengaruhi tingkat keberhasilan algoritma *Fast Fourier Transform* untuk melakukan transposisi suara,dan Kombinasi parameter terbaik untuk melakukan trosnposisi suara dengan menggunakan algoritma *Fast Fourier Transform* adalah 8 untuk oversampling dan 124 untuk FFT frame size.

2.2 Saran

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

1. Sebaiknya menggunakan bahasa pemrograman selain java untuk melakukan penelitian terkait suara dikarenakan bahasa pemrograman Java sangat minim library untuk mengolah file audio. Bahasa yang cukup bagus adalah visual basic atau matlab.
2. Sebaiknya mencoba mengunakan algoritma selain algoritma *Fast Fourier Transform* dan membandingkannya.

DAFTAR PUSTAKA

al- Tho'i ,Kamaluddin, *Qawaidut Tilawah*, Baghdad : Al-Adhamy

al-Qayum, Abd al-Ghafur al-Sindi (2001). *safahat fi' Ulumul al-Qira'at*, Beirut : Daral-Basya'ir al- Islamiyyah.

al-Rahman, Abu Abd Ahmad bin Syu'aib at-Nasaiy Sunan al-Nasaiy, (1980). Beirut : Dar al-Fikr

Banoe, Pono. (2003*). Kamus Musik*. Jakarta : PT. Kanisius.

Binanto, Iwan. (2010). *Multimedia Digital Dasar Teori dan Pengembanganya*. Yogyakarta : Andi.

Bruel & Kjaer, 1986, *Noise control principles and practices 2nd edition*, Naerum Offset, Denmark.

Chu, Eleanor, Alan George. (2000). *Inside the Fast Fourier Transform Black Box : Serial and parallel FFT Algorithms* . Boca Raton, FL : CRC Press.

Doelle, Leslie L., (1986), "*Akustika Lingkungan*", Erlangga, Jakarta.

EEPIS – IT, (2012). Teknik Elektro dan Informatika, Bandung *Praktikum Pengolahan Sinyal Digital*.

Egan, M.David, (1972), "*Concept in Architectural Acoustics*", McGraw-Hill, United States of America.

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

Isfanhari, Musafir dan Widyo Nugroho. 2000. *Pengetahuan Dasar Musik*. Surabaya : Dinas P dan K Propinsi Daerah Tingkat I Jawa Timur

Jamalus. 1988. *Pengajaran Musik Melalui Pengalaman Musik*. Jakarta: Depdikbud.

Katsir, Ibnu. 2000. *Tafsir Ibnu Katsir*. Bandung. Sinar Baru Algensindo.

Munir, Ahmad dan Sudarsono, 1994, *Ilmu Tajwid Dan Seni Baca Al-Qur'an*, Jakarta, Reneka Cipta.

Pradipta, Nandra. *Implementasi Algoritma FFT (Fast Fourier Transform) Pada Digital Signal Processor (DSP) TMS320C542)*. Jurnal Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro.

Resnick, R. and Halliday, (1992), Fisika, *Penterjemah Pantur Silaban dan Erwin Sucpito*, Erlangga, Jakarta.

Santoso, Eko Aditya, Umar Sagaf, Muhammad Efendi, Reza Fahrur Rasyid, Teguh Adi Gunawan. 2010. *Klastering Suara Laki-Laki dan Perempuan Menggunakan Algoritma K-Means Berdasarkan Hasil Ekstraksi FFT (Fast Fourier Transform)*. Jurnal Universitas Brawijaya.

Sylado, Remy. 1983. *Menuju Apresiasi Musik*. Bandung : Angkasa.