IMPLEMENTASI *FAST FOURIER TRANSFORM* PADA PENGENALAN NADA PIANO BERBASIS ANDROID

SKRIPSI

Oleh :
TRIO SAFAAT
10650059

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016

HALAMAN JUDUL
IMPLEMENTASI *FAST FOURIER TRANSFORM* PADA PENGENALAN NADA PIANO BERBASIS ANDROID

SKRIPSI

Oleh :
TRIO SAFAAT
10650059

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016

IMPLEMENTASI *FAST FOURIER TRANSFORM* PADA PENGENALAN NADA PIANO BERBASIS ANDROID

SKRIPSI

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

Oleh:
TRIO SAFAAT
10650059

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2016

IMPLEMENTASI *FAST FOURIER TRANSFORM* PADA PENGENALAN NADA PIANO BERBASIS ANDROID

SKRIPSI

Oleh :

Nama : Trio Safaat
NIM : 10650059
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji :

Tanggal :……

Dosen Pembimbing I	**Dosen Pembimbing II**
Totok Chamidy, M.Kom	**Ririen Kusumawati, M.Kom**
NIP. 19830616 201101 1 004	**NIP. 197203092005012002**

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

IMPLEMENTASI *FAST FOURIER TRANSFORM* PADA PENGENALAN NADA PIANO BERBASIS ANDROID

SKRIPSI

Oleh :

TRIO SAFAAT

NIM. 10650059

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

Tanggal, April 2016

Susunan Dewan Penguji :		**Tanda Tangan**
1. Penguji Utama	**: Dr. Muhammad Faisal, M.T** **NIP. 19740510 200501 1 007**	**( )**
2. Ketua Penguji	**: Dr. Suhartono, M. Kom** **NIP. 19680519 200312 1 001**	**( )**
3. Sekretaris	**: Totok Chamidy, M.Kom** **NIP. 19830616 201101 1 004**	**( )**
4. Anggota Penguji	**: Ririen Kusumawati, M.Kom** **NIP. 19720309 200501 2 002**	**( )**

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama	:	Trio Safaat
NIM	:	10650059
Fakultas/Jurusan	:	Sains dan Teknologi / Teknik Informatika
Judul Penelitian	:	Implementasi *Fast Fourier Transform* pada Pengenalan Nada Piano Berbasis Android

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 05 Mei 2016
Yang membuat pernyataan,

Trio Safaat
NIM. 10650059

MOTTO

"As long as we still have determination preserved in spirit, then there is no too late to start a new beginning."

فَإِنَّ مَعَ ٱلْعُسْرِ يُسْرًا ۝٥ إِنَّ مَعَ ٱلْعُسْرِ يُسْرًا ۝٦

"Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan." (QS. Asy Syarh: 5-6).

PERSEMBAHAN

Dengan mengucap syukur ALHAMDULILLAH kepada Allah SWT atas terselesaikannya tugas akhir / skripsi ini. Saya ucapkan terimakasih sebanyak – banyaknya kepada :

- Bapak dan Ibu, orang tua yang selalu memberikan doa yang tiada henti – hentinya. Motivasi, bimbingan dan kesabarannyayang selalu membuat saya untuk terus melangkah. THEY ARE GREAT PARENTS.
- Dosen-dosen di jurusan Teknik Informatika yang telah memberikan bimbingan, pendidikan dan pengalaman kepada saya selama menjalani studi di jurusan Teknik Informatika.
- Kakak dan adik-adikku yang selalu memberikan dukungan doa. Tiada yang paling mengharukan saat berkumpul bersama kalian. Maaf belum bisa menjadi panutan seutuhnya, tapi aku akan selalu menjadi yang terbaik untuk kalian.
- Teman seperjuangan selama kuliah : Ubik, Agung, Ahmad Saifullah dan teman seperjuangan yang lain, yang telah semangat berjuang bersama – sama selama proses perkuliahan.Tetap semangat sampai kapanpun.
- Teman seperjuangan skripsi : Putra, Syahid dan teman seperjuangan skripsi yang lain,telah semangat berjuang bersama – sama untuk menyelesaikan tugas akhir ini. Semoga kita selalu diberikan semangat dalam menyelesaikan perjuangan yang lainnya.
- Teman – teman INFINITY, terima kasih atas bantuan, dukungan dan doa kalian selama ini. Semoga kita bisa sukses bersama – sama.
- Sahabat-sahabat Mangsar: Ari, Musleh, Yayang, Feri, Osby, Gigih dan sahabat Mangsar yang lainnya. Kalian telah menghibur dan memberikan pengalaman selama proses skripsi ini berlangsung. STAY HAPPINESS TOGETHER.
- Teman-Teman Kontrakan : Rifa'i, Gilang, Aji, Danang, Andi, Fery serta lainnya sebagai pengisi hari – hariku sekaligus sebagai pengganggu dalam penyelesaian tugas akhir ini, tetapi kalian adalah keluarga baruku selama di malang.
- Terima kasih kepada pihak – pihak atau orang – orang yang belum saya sebutkan. Terima kasih telah mendukung, membantu, memberikan pengalaman dan mendoakan saya selama proses penyelesaian tugas akhir skripsi ini. Kalian adalah guru terhebat yang pernah saya temui.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji syukur penulis panjatkan atas kehadirat Allah SWT, Pemelihara seluruh alam raya, yang atas limpahan rahmat, taufik dan hidayah-Nya, penulis mampu menyelesaikan Tugas Akhir yang berjudul **"Implementasi Fast Fourier Transform Pada Pengenalan Nada Piano Berbasis Android"**. Tugas akhir ini dikerjakan demi memenuhi salah satu syarat untuk menyelesaikan program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri (UIN) Malang.

Penulis menyadari bahwa tugas akhir ini bukanlah tujuan akhir dari belajar, karena belajar adalah sesuatu yang tidak terbatas. Penulis juga menyadari sepenuhnya bahwa tugas akhir ini masih jauh dari kesempurnaan baik isi, bahasa serta penulisannya. Hal ini disebabkan keterbatasan penulis dari segi ilmu pengetahuan, bahan atau data, dana dan waktu sehingga kritik dan saran yang membangun sangat penulis harapkan demi kesempurnaan tugas akhir ini.

Terselesaikannya tugas akhir ini tentunya tak lepas dari dorongan dan uluran tangan berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih dan apresiasi kepada:

1. Prof. Dr. H. Mudjia Rahardjo, M.Si selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Bayyinatul Muchtaromah,M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian selaku ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Totok Chamidy, M.Kom dan Ririen Kusumawati, M.Komselaku dosen pembimbing skripsi, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.

5. Segenap civitas akademika jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
6. AyahandaSugiono dan IbundaSri Rahayu tercinta yang senantiasa memberikan doa, spirit, tenaga, biaya, dan restunya kepada penulis dalam menuntut ilmu.
7. Semua pihak yang ikut membantu dalam menyelesaikan skripsi ini baik berupa materiil, moril maupun spirituil.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan tugas akhir ini dengan melimpahkan rahmat dan karunia-Nya. Semoga karya penelitian tugas akhir ini dapat memberikan manfaat dan kebaikan bagi banyak pihak demi kemaslahatan bersama serta bernilai ibadah dihadapan Allah SWT. *Amin**Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb.

Malang, Mei 2016

Penulis

PUSAT PERPUSTAKAAN

DAFTAR ISI

HALAMAN JUDUL ii

HALAMAN PENGESAHAN v

PERNYATAAN KEASLIAN PENELITIAN vi

MOTTO vii

PERSEMBAHAN viii

KATA PENGANTAR ix

DAFTAR ISI xi

DAFTAR TABEL xv

DAFTAR GAMBAR xvi

ABSTRAK xvii

BAB I PENDAHULUAN 1

1.1 Latar Belakang Masalah 1

1.2 Identifikasi Masalah 4

1.3 Tujuan 4

1.4 Batasan Masalah 5

1.5 Manfaat Penelitian 5

BAB II LANDASAN TEORI 6

BAB III METODE PENELITIAN 10

3.1 Desain Penelitian 10

3.1.1 Objek Penelitian 10

3.1.2 Prosedur Penelitian 10

3.1.3 Jenis Penelitian 11

3.1.4 Sumber Data 11

3.2 Instrumen Penelitian 11

3.3 Kerangka Teori 12
3.3.1 Nada 12
3.3.2 Piano 12
3.3.3 Kunci (Chord) 13
3.3.4 Audio Digital 16
3.3.5 File Wav 18
3.3.6 Struktur File Wav 20
3.3.6.1 Header File Wav 22
3.3.6.2 Chunk File Wav 23
3.3.6.3 Chunk Format 24
3.3.7 Sampling 25
3.3.8 Frame Blocking 27
3.3.9 Windowing 28
3.3.10 Fast Fourier Transform 29
3.3.11 Euclidean Distance 32
3.3.12 Android 33
3.4 Kerangka Konsep 34
3.4.1 Alur Kerja 34
3.4.2 Alur Kerja Proses Program 37
3.4.2.1 Proses Sampling 37
3.4.2.2 Frame Blocking 38
3.4.2.3 Windowing 39
3.4.2.4 Fast Fourier Transform 40
3.4.2.5 Similarity Matching (Pencocokan Nada) 42
3.4.3 Desain Interface 44

3.4.3.1 Tampilan Pengenalan Kunci 44
3.4.3.2 Tampilan Buka File Wav 45
3.4.3.3 Tampilan Proses Frame Blocking 46
3.4.3.4 Tampilan Proses Windowing 46
3.4.3.5 Tampilan Proses FFT 47
3.4.3.6 Tampilan Proses Matching 47
BAB IV 49
HASIL DAN PEMBAHASAN 49
4.1 Implementasi 49
4.1.1 Spesifikasi Perangkat Keras (*Hardware*) 49
4.1.2 Spesifikasi Perangkat Lunak (Software) 50
4.2 Bentuk Pola Kunci Piano 50
4.3 Hasil Data Latih 51
4.3.1 Kunci Mayor 52
4.3.2 Kunci Minor 52
4.3.3 Data Latih Frame Blocking 53
4.3.4 Data Latih Windowing 54
4.3.5 Data Latih FFT 54
4.4 Fitur Database 56
4.5 Hasil Data Uji 57
4.5.1 Kunci Mayor 57
4.5.2 Kunci Minor 58
4.5.3 Hasil Pencocokan Data Uji Dengan Data Latih 58
4.6 Integrasi Dalam Islam 70
BAB V 74

PENUTUP.....74
5.1 Kesimpulan.....74
5.2 Saran.....75
DAFTAR PUSTAKA.....76
LAMPIRAN.....79

UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM
PUSAT PERPUSTAKAAN

DAFTAR TABEL

Tabel 3.1 Tabel Rangkaian Akor 14
Tabel 3.2 Akor Dasar dan Kebalikannya 15
Tabel 3.3 Penjelasan Struktur File Wav 22
Tabel 3.4 Nilai Jenis Chunk RIFF 23
Tabel 3.5 Format Chunk RIFF 24
Tabel 3.6 Nilai – Nilai Chunk Format File Wav 25
Tabel 4.7 Hasil Perhitungan Jarak Data Latih dan Data Uji 66
Tabel 4.8 Hasil Kesesuaian Data Uji 69

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM PUSAT PERPUSTAKAAN

DAFTAR GAMBAR

Gambar 3.1 Sinyal Analog....25
Gambar 3.2 Bentukan Sinyal Sample....26
Gambar 3.3 (a) Bentukan Proses Pencuplikan (b) Bentukan Proses Pencuplikan yang Lebih Besar....27
Gambar 3.4 DFT resolusi 1 Hz....28
Gambar 3.5 Vektor Space 2 Dimensi pada Euclidean Distance....33
Gambar 3.6 Diagram Blok Sistem....35
Gambar 3.7 Alur Kerja Proses Program....37
Gambar 3.8 Flowchart Window Hamming....39
Gambar 3.9 Diagram Alir FFT....41
Gambar 3.10 Desain Interface Pengenalan Kunci Piano....44
Gambar 3.11 Desain Interface Buka File Wav....45
Gambar 3.12 Desain Interface Proses Frame Blocking....46
Gambar 3.13 Desain Interface Proses Windowing....46
Gambar 3.14 Desain Interface Proses FFT....47
Gambar 3.15 Desain Interface Proses Matching....47
Gambar 4.1 Hasil Tampilan Pengenalan Kunci....51
Gambar 4.2 Sampel Kunci Nada Piano Mayor....52
Gambar 4.3 Sampel Kunci Nada Piano Minor....52
Gambar 4.4 Kode program frame blocking....53
Gambar 4.5 Hasil frame blocking....53
Gambar 4.6 Kode program windowing....54
Gambar 4.7 Hasil windowing....54
Gambar 4.8 Kode program FFT....55
Gambar 4.9 Hasil FFT....55
Gambar 4.10 Kode program file pattern....56
Gambar 4.11 Hasil file pattern....57
Gambar 4.12 Sampel Kunci Nada Piano Mayor....58
Gambar 4.13 Sampel Kunci Nada Piano Minor....58
Gambar 4.14 Kode program euclidean distance....59
Gambar 4.15 Tampilan hasil pencocokan nada....59

ABSTRAK

Safaat, Trio. 2016. Implementasi Fast Fourier Transform Pada Pengenalan Nada Piano Berbasis Android. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Totok Chamidy, M.Kom (II) Ririen Kusumawati, M.Kom

Kata Kunci: *Kunci, Frame Blocking, Windowing, Fast Fourier Transform, Euclidean Distance.*

Piano merupakan suatu alat musik yang digemari oleh sebagian orang, karena suara yang dihasilkan begitu harmonis dan dinamis. Hal yang paling mendasar untuk memainkan alat musik ini tentunya terlebih dahulu mengenali tangga nada kemudian dikenali bentuk pola kuncinya. Saat mendengarkan sebuah alunan musik piano terkadang seseorang tidak mengetahui nada dan kunci apa yang dihasilkan dari musik tersebut. Untuk itu, dibutuhkan sebuah media yang dapat mendeteksi kunci dari suara musik yang dihasilkan. Dalam penelitian ini media yang diterapkan adalah pada perangkat handphone berbasis android. Salah satu keunggulan dari perangkat tersebut adalah dapat dibawa kemana-mana sehingga dapat mengenali kunci nada dimana saja dan kapan saja.

Dalam penelitian ini, kunci yang akan dideteksi adalah kunci standar yaitu kunci mayor dan minor. Proses kerja aplikasi diawali dengan memasukkan file nada berformat file wave. File tersebut di sampling dengan sample rate standar 44100Hz. Selanjutnya, membagi data samplenya menjadi beberapa frame (frame blocking). Dari kumpulan beberapa frame tersebut dibutuhkan proses windowing agar informasi sinyal tidak discontinue dan menimbulkan suatu efek sinyal yang berbeda dengan sinyal aslinya. Proses dilanjutkan dengan mentransformasikan sinyal diskrit yang dihasilkan menjadi sinyal frekuensi menggunakan algoritma fast fourier transform (FFT). Setelah diketahui informasi data dari file masukan kemudian dihitung jarak terdekat dengan informasi dari data latih. Perhitungan jarak tersebut menggunakan rumus Euclidean Distance (ED). Jarak terdekat adalah yang mendekati nilai 0. Output aplikasi ini adalah sebuah kunci mayor dan minor dari jarak terdekat dari kedua informasi data.

ABSTRACT

Safaat, Trio. 2016. Implementation of Fast Fourier Transform On Android-Based Introduction to Piano Tones. Essay. Department of Informatics, Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang. Counsellor.: (I) Totok Chamidy, M.Kom (II) Ririen Kusumawati, M.Kom

Keywords : *Chord, Frame Blocking, Windowing, Fast Fourier Transform, Euclidean Distance.*

The piano is a musical instrument that is favored by most people, because of the resulting sound so harmonious and dynamic. The most fundamental thing to play this instrument must first recognize the scales then recognizable pattern shape of the chords. When listening to a piano music is sometimes a person does not know what the tone and the chord generated from the music. For that, we need a media that can detect the chord of the sound of music produced. Dalam penelitian ini media yang diterapkan adalah pada perangkat handphone berbasis android. One of the advantages of the device is that it can be taken anywhere so it can recognize the chords tone anywhere and anytime.

In this study, the key to be detected is a chord standard that major and minor chords. The working process of application begins by entering tone file format wave files. The file is in the sampling with the standard sample rate 44100Hz. Furthermore, the data sample is split into multiple frames (frames blocking). From the collection of some of the frames required windowing process so that information is not discontinue signals and cause an effect signal that is different from the original signal. The process is continued by transforming the discrete signal into a frequency signal is generated using an algorithm Fast Fourier transform (FFT). Having in mind the information data from the input file is then calculated the shortest distance to the information from the training data. The distance calculation using the formula Euclidean Distance (ED). The shortest distance is approaching a value of 0. The output of this application is a major and minor chords from the closest distance from both information data.

مستخلص البحث

شفاعات، تريو .2016. تنفيذ *Fast Fourier Transform* في مقدمة النغمة البيانو التى تستند الروبوت. بحث جامعي قسم المعلوماتية كلية العلوم والتكنولوجيا في جامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج .المشرف: توتوك خامدي, الماجستير و ريرين كوسوماواتي, الماجستيرة

الكلمات البحث: *Euclidean Distance ,Fast Fourier Transform ,Windowing ,Frame Blocking*

البيانو هو الآلة الموسيقية التي تؤيدها معظم الناس، لأن الصوت الناتج ذلك انسجاما وديناميكية . والشيء الأكثر أهمية أن يلعب هذا الصك يجب أن نعترف أولا جداول ثم نمط التعرف مفتاح الشكل .عند الاستماع إلى موسيقى البيانو في بعض الأحيان شخص لا يعرف ما لهجة ومفتاح ولدت من الموسيقى .لذلك، نحن بحاجة إلى وسائل الإعلام التي يمكن الكشف عن مفتاح صوت الموسيقى المنتجة .في هذا البحث هو وسائل الإعلام تطبيقها على الأجهزة النقالة على أساس الروبوت. واحدة من مزايا الجهاز هو أنه يمكن أن تؤخذ في أي مكان حتى أنه يمكن التعرف على لهجة رئيسية في أي مكان وزمان.

في هذه البحث، والمفتاح ليتم الكشف هو المعيار الرئيسي الذي مفاتيح الأكبر والأصغر. عملية العمل من تطبيق تبدأ عن طريق إدخال ملف نغمة شكل ملفات الموجة (*file wave*). وأخذ عينات من الملف مع معدل عينة القياسية (*sample rate*) يعنى *44100Hz* بعد ذلك، تقسيم البيانات النموذجية لها إلى بعض إطارات (إطارات حجب *frame blocking*). من جمع بعض الأطر اللازمة لعملية النوافذ بحيث المعلومات لا يوقف إشارات وتسبب إشارة مفادها أن يختلف من الإشارة الأصلية .وتستمر هذه العملية عن طريق تحويل إشارة منفصلة إلى إشارة التردد يتم إنشاؤها باستخدام فوريه السريع تحويل خوارزمية *fast fourier transform (FFT)*. بعد تعرف في الاعتبار ثم يتم حساب البيانات والمعلومات من ملف الإدخال أقصر مسافة إلى المعلومات من بيانات التدريب . حساب المسافة باستخدام الصيغة الإقليدية المسافة *Euclidean Distance* (ED) أقصر مسافة تقترب قيمة 0. إخراج (*Output*) هذا التطبيق هو المفتاح الأكبر والأصغر من أقرب مسافة من الثانية المعلومات البيانات

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Alat musik piano dapat menghasilkan gelombang bunyi dengan berbagai jenis nada. Nada adalah bunyi yang beraturan. Sederetan nada yang terhimpun jadi satu dengan harmonis akan menjadi sebuah musik. Dalam penulisannya, nada dilambangkan dengan not atau notasi. Tinggi - rendahnya nada ditentukan oleh frekuensi dasar gelombang bunyi. Jika semakin besar frekuensi dasar gelombang bunyi, maka semakin tinggi pula nada yang dihasilkan. Tinggi - rendahnya nada piano dapat ditangkap oleh indera pendengaran manusia, namun tidak dapat mengetahui secara pasti jenis nada piano apa yang dapat didengar. Oleh karena itu hal ini sangatlah penting bagi seseorang yang ingin mengetahui dan mengenal jenis nada dari alat musik piano.

Manfaat seseorang dalam mengenal nada musik adalah sebagai media hiburan di kala seseorang tersebut merasa penat menjalani aktifitas sehari – hari. Hal tersebut sesuai dengan ayat Al-Qur'an yang menceritakan cipataan Allah Swt. yang rapi, seimbang dan penuh keindahan serta sedap dipandang mata. Serasi bentuk, ukuran dan warna. Semua ini menunjukkan bahwa keindahan, baik itu ilmiah atau buatan (seperti musik) adalah termasuk sesuatu yang memang bisa dinikmati sebagai objek tafakur dan menghibur diri semata. Sebagaimana disebutkan dalam ayat Al-Qur'an sebagai berikut :

ٱلَّذِى خَلَقَ سَبۡعَ سَمَٰوَٰتٍ طِبَاقًاۖ مَّا تَرَىٰ فِى خَلۡقِ ٱلرَّحۡمَٰنِ مِن تَفَٰوُتٍۖ فَٱرۡجِعِ ٱلۡبَصَرَ هَلۡ تَرَىٰ

مِن فُطُورٍ ۝٣

Artinya : "*Yang telah menciptakan tujuh langit berlapis-lapis. Kamu sekali-kali tidak melihat pada ciptaan Tuhan Yang Maha Pemurah sesuatu yang tidak seimbang. Maka lihatlah berulang-ulang, adakah kamu lihat sesuatu yang tidak seimbang.*" (QS. Al-Mulk: 3). (RI 1996).

Sesungguhnya keserasian, keseimbangan dan keindahan yang Allah Taala ciptakan pada alam ini, baik pada bentuk, ukuran, warna, suara yang Allah Swt. ciptakan pada gunung, bukit, sawah, gurun, kebun, sungai, lautan, hewan, gemercikan air, kicauan burung di pagi hari adalah hal yang diberikan-Nya untuk manusia. Tak beda antara yang alami atau buatan khususnya musik, sebab prinsipnya sama – sama memiliki keindahan, sekedar mengistirahatkan jiwa. Pada kenyataanya, setelah bertafakur semakin shalih dan bertobat dari kesalahan setelah merenungi kebesaran-Nya melalui keindahan ciptaan-Nya.

Musik adalah kombinasi nada-nada yang memiliki pola tertentu. Pola tersebut dapat dikenali ketika belajar bermain piano. Cara mudahnya adalah dengan mengenali pola pengkombinasian nada - nada untuk membentuk sebuah kunci atau chord. Tetapi untuk merepresentasikan nada yang didengarkan tersebut menjadi sebuah not atau kunci tidak mudah. Melihat kondisi tersebut dibutuhkan suatu sarana yang dapat membaca nada – nada pada sebuah musik piano, kemudian menerjemahkannya menjadi sederetan notasi atau kunci piano secara otomatis. Seperti halnya seorang komposer, sebuah perangkat lunak

dapat mengenali nada dasar atau kunci dan dapat ditampilkan pada personal computer (PC). Pengenalan nada – nada dasar tersebut dapat dilakukan melalui pengenalan frekuensi suara yang berasal dari instrumen musik yang kemudian diolah dengan perangkat lunak.

Dalam mengolah sinyal suara atau nada piano yang masuk ke dalam sistem perangkat lunak, sinyal informasi yang berupa sinyal analog dalam domain waktu tersebut di konversikan ke dalam domain frekuensi menggunakan algoritma Fast Fourier Transform (FFT). FFT digunakan untuk menghitung Descrete Fourier Transform (DFT) dengan cepat. DFT memisahkan suatu nilai sekuensial menjadi komponen yang berbeda frekuensinya. Komputasi DFT N titik secara jelas dengan menggunakan definisinya membutuhkan jumlah operasi aritmatika sebesar N^2, sedangkan FFT, untukmendapatkan hasil yang sama dengan jumlah titik yang sama, hanya membutuhkan operasi aritmatika sebanyak $\log_N N$. Perbedaannya akan menjadi sangat besar untuk jumlah N (titik) yang banyak karena dalam praktiknya N ini bukan sebesar 1 atau 2, tapi bisa mencapai ribuan atau jutaan. Dengan itu, waktu komputasi dapat dikurangi dengan pendekatan persamaan $N / \log N$.

Penelitian ini mengembangkan dari penelitian sebelumnya (Supriansyah 2013), pada penelitian tersebut FFT diterapkan pada perangkat lunak Matlab. Disini, penulis mencoba menerapkan FFT pada platform android menggunakan bahasa pemrograman java untuk mengolah sinyal nada yang masuk. Sebelum mentransformasikan ke dalam domain frekuensi dibutuhkan proses untuk mengubah sinyal kontinyu menjadi sinyal diskrit, proses ini disebut dengan

proses sampling. Transformasi ini agar mempermudah dalam mengolah sinyal yang telah ditransformasikan karena sinyal dalam bentuk diskrit lebih mudah untuk diolah menjadi sebuah frekuensi (Hz). Jadi, Fast Fourier Transform dapat digunakan untuk mencari power dan frekuensi dasar dari nada alat musik piano. Dengan menggunakan algoritma FFT diharapkan mampu memproses data nada piano atau audio yang masuk dengan data audio baru yang lebih jernih dan berukuran kecil, serta dapat mewakili data audio asli.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat di identifikasi masalahnya adalah mengembangkan pada penelitian sebelumnya yang mana untuk mengetahui jenis – jenis nada masih menggunakan PC dan metode FFT diterapkan pada perangkat lunak Matlab. Sehingga pada penelitian ini pengenalan nada dan algoritma FFT diterapkan pada sistem operasi Android menggunakan bahasa pemrograman java.

1.3 Tujuan

Adapun tujuan dari penelitian ini antara lain :

1. Membangun suatu perangkat lunak yang dapat mendeteksi dan mengenali nada – nada dari alat musik piano secara otomatis dan menampilkannya dalam bentuk kunci.
2. Mengimplementasikan algoritma FFT pada pemrosesan sinyal digital dari nada tunggal piano.

3. Mengimplementasikan algoritma FFT pada sistem operasi android.

1.4 Batasan Masalah

Dalam pembahasannya, penelitian dibatasi oleh hal – hal berikut :

1. Jenis suara keyboard yang digunakan adalah Grand Piano.
2. Nada yang diambil sebanyak 24 nada mayor dan 24 nada minor.
3. Format penyimpanan file nada adalah format wave (*.wav)
4. Tampilan yang digunakan pada aplikasi ini berupa chord dan deret tuts keyboard yang ditekan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini dapat menumbuhkan dan menarik minat bermain musik. Aplikasi yang akan dibuat dapat digunakan sebagai prasarana untuk berlatih bagi pemula yang ingin menambah ilmu dalam bidang musik piano.

BAB II

LANDASAN TEORI

Berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan :

Supriyansyah, (2013) melakukan penelitian tentangpengenalan nada tunggal keyboard (orgen) pada PC berbasis matlab, aplikasi ini dibuat dengan menggunakan Fast Fourier Transform untuk mencari power dan frekuensi dasar dari nada musik keyboard (orgen). Pada aplikasi pengenalan nada tersebut dilakukan dengan cara pengenalan frekuensi yang kemudian diolah menggunakan FFT. Dimana pada penelitian ini aplikasi dibuat menggunakan pemrograman berbasis Matlab. Hasil keluaran dari aplikasi tersebut berupa chord dan gambar tuts keyboard yang ditekan. Pembuatan sistem tersebut terdiri dari proses perekaman suara nada dari alat musik keyboard melalui microphone, sinyal suara kemudian diproses dengan FFT dan kemudian dicari ciri dari sinyal suara tersebut sebagai sinyal acuan untuk dikenali sebagai nada yang nantinya akan divisualisasikan dalam bentuk *chord*.

Hadyansyah, (2010) melakukan penelitian tentang ketepatan nada suara manusia menggunakan FFT berbasis Android, yang dibuat oleh Irvan Hadyansyah. Pada aplikasi tersebut terdapat 2 proses, yaitu proses exstrasi ciri dan proses komperisasi. Tujuan dari sistem ini adalah merancang dan merealisasikan suatu sistem yang dapat menerima input suara manusia secara real time dan membandingkan input tersebut dengan frekuensi referensi. Dalam aplikasi tersebut,

dapat mengidentifikasi ketepatan nada yang dilantunkan oleh manusia. Aplikasi tersebut bersifat realtime, dimana menggunakan suara manusia sebagai sinyal input kemudian dilanjutkan dengan proses ekstraksi suara yaitu menggunakan metode di FFT (*Fast Fourier Transform*) kemudian dibandingkan dengan frekuensi nada dari data latih dengan data uji menggunakan Euclidean Distance.

Aris Tjahyanto, (2011) melakukan penelitian tentang pengarsipan musik gamelan dalam bentuk digital. Beberapa parameter yang diperhatikan dalam penelitian ini yaitu : frekuensi dasar, frekuensi harmonisa, fase, tempo dan jenis instrument yang dimainkan. Jenis suara gamelan yang dibangkitkan adalah *full synthetic* dan *semi syntethic*. Pada penelitian yang menggunakan metode *analysis-by synthesis* ini diperoleh bahwa untuk suara yang dihasilkan full synthetic masih belum memuaskan, terutama belum bisa menirukan suara dentingan khas logam yang dipukul. Sedangkan suara gamelan yang dihasilkan secara semi synsynthetic memiliki kualitas yang setara dengan suara gamelan yang dihasilkan oleh para penabuh gamelan.

Lyana, (2012) melakukan penelitian tentang sistem penentu akor pada audio dengan fast fourier transform dan jaringan syaraf tiruan. Pada penelitian ini bertujuan untuk membantu pemula dalam menentukan akor yang tepat dalam sebuah lagu. Sistem identifikasi akor tersebut menerima masukan berupa file suara piano berformat wav menggunakan metode *Fast Fourier Transform* dan Jaringan Syaraf Tiruan ART 2. Model perangkat lunak yang dirancang pada penelitian tersebut dibagi menjadi beberapa blok, yaitu blok akuisi data, blok preprocessing, dan blok ekstraksi ciri, blok jaringan syaraf tiruan ART 2 dan keluara. Cara kerja

sistem adalah dengan memberikan masukan berupa rekaman audio melodi piano berformat wav. Audio yang direkam akan di *filter*, normalisasi dan *cropping* untuk mendapatkan representasi sinyal aslinya sehingga komponen yang dianggap bukan sinyal masukan akan dihilangkan. Kemudian dilakukan penentuan onset dan segmentasi untuk mengetahui berapa banyak nada dari masukan. Pada ekstrasi ciri diambil dua macam ciri unik dari setiap data. Hasil keluaran ekstrasi ciri masuk ke proses klasifikasi menggunakan JST ART 2 untuk menentukan akor apa saja yang tepat pada melodi piano. Keluaran sistem berupa lagu yang terdiri dari melodi dan akornya. Akurasi pengujian sistem tersebut menggunakan parameter rho=0.98 dan iterasi=7 pada jaringan syaraf tiruan ART 2 didapat 100% untuk data latih dan 98,49% untuk data uji serta akurasi ketepatan akor didapat 79,80%.

Peggy, (2012) melakukan penelitian tentang deteksi lagu pada piano berdasarkan ciri *mel frequency cepstral coefficient* dengan metode klasifikasi *learning vector quantization* dan *euclidean distance*. Prinsip kerja dari sistem yang telah dibangun adalah mendeteksi judul lagu pada piano dimana lagu ujinya disesuaikan dengan lagu referensi *database*. MFCC digunakan sebagai teknik ekstrasi ciri. JST *learning vector quantization* dan e*uclidean distance* digunakan sebagai metode klasifikasi yang berfungsi untuk mengenali vector ciri dari data input hasil ekstrasi ciri MFCC dan mengklasifikasikan ciri tersebut berdasarkan target yang telah ditentukan. Dari dua metode klasifikasi tersebut akan dilakukan perbandingan, metode mana yang menghasilkan performasi yang lebih baik. Sistem memiliki performasi terbaik dengan akurasi tertinggi dalam pendeteksian lagu pada tahap *offline* ketika MFCC mengalami *overlap* 25% dengan menggunakan metode

klasifikasi *euclidean distance*, spesifikasi *threshold* = 0.2, dengan tingkat akurasi sebesar 93.33%. Sedangkan untuk 5 kali percobaan deteksi 15 lagu pada tahap *online* dengan pemberian *filter butterworth,* diperoleh akurasi sistem sebesar 84%.

Fajar, (2011) melakukan penelitian tentang pengenalan dan analisis kualitas penalaan nada tunggal piano secara real time menggunakan metode JST-SOM. Pada sistem tersebut sistem mengekstrasi nada tunggal piano yang berbeda – beda menjadi dua macam ciri yaitu Fundamental-FFT dan Harmonic-FFT. Dua macam ciri tersebut dijadikan masukan dalam metode JST-SOM sehingga memberikan keluaran berupa notasi dan juga sistem memberikan keluaran berupa persentase kualitas penalaan dawai piano. Dalam sistem yang menggunakan metode JST-SOM ini diperoleh performansi maksimum 100% untuk akurasi sistem dalam mengenali nada dan 99,48% untuk akurasi dalam mengukur kualitas penalaan.

BAB III

METODE PENELITIAN

Penulis membagi pengerjaan penelitian menjadi beberapa tahap, antara lain :

1.1 Desain Penelitian

1.1.1 Objek Penelitian

Dalam penelitian ini, peneliti mengambil objek penelitian yaitu nada piano pada range C2 sampai dengan C7 untuk dikenali dalam bentuk chord.

1.1.2 Prosedur Penelitian

Penelitian ini akan dilakukan dalam beberapa tahapan. Berikut ini adalah alur kerja dari penelitian yang menunjukkan tahapan atau proses dalam penelitian :

1. Studi Literatur

   Mencari referensi dan sumber – sumber lainnya mengenai pengenalan nada, teori nada, pemrograman java berbasis android, serta algoritma Fast Fourier Transform (FFT)

2. Pendalaman Materi

   Mempelajari konsep tentang pengenalan nada, teori nada, pemrograman java berbasis android, serta algoritma Fast Fourier Transform.

3. Analisis dan Desain

Menganalisis kebutuhan, mengumpulkan data yang diperlukan dan melakukan perancangan untuk tahap implementasi

4. Implementasi

   Mengimplementasikan algoritma Fast Fourier Transform pada sistem pengenalannada piano berbasis android dalam sebuah perangkat lunak.

5. Pengujian dan Analisis Hasil

   Melakukan pengujian terhadap perangkat lunak dan melakukan analisis terhadap keluaran dari perangkat lunak.

6. Dokumentasi dan laporan

   Penyusunan laporan akhir merupakan dokumentasi dari keseluruhan pelaksanaan penelitian dan diharapkan bermanfaat bagi penelitian lebih lanjut.

1.1.3 Jenis Penelitian

Penelitian yang digunakan adalah jenis penelitian kuantitatif.

1.1.4 Sumber Data

Data yang diperoleh berasal dari berbagai sumber, diantaranya : penelitian kepustakaan, buku – buku referensi terkait dan internet.

1.2 Instrumen Penelitian

Instrumen penelitian adalah alat yang digunakan oleh peneliti dalam melakukan penelitiannya untuk mencapai tujuan penelitian. Berikut adalah instrumen penelitiannya :

1. Variable bebas (Input)

   Yang merupakan variable bebas adalah : nada piano.

2. Variable terikat (Output)

   Yang merupakan variable terikat adalah : notasi huruf dan kunci (chord)

3. Variable penghubung

   Yang merupakan variable penghubung adalah : pengenalan nada piano yang ditelusuri menggunakan metode Fast Fourier Transform

1.3 Kerangka Teori

1.3.1 Nada

Nada adalah bunyi yang teratur, artinya mempunyai bilangan getar (frekuensi) yang tertentu. Tinggi rendahnya bunyi (suara) bergantung pada besar/kecilnya frekuensi tersebut. Dalam musik, tinggi rendah dan panjang pendeknya nada dapat ditunjukkan dengan tanda yang disebut titik nada atau not. Jadi, not berfungsi sebagai huruf musik (Kurniawan 2006). Sederetan nada yang dimainkan secara merdu akan menghasilkan sebuah musik. Dalam dunia musik, nada memunyai notasi atau sistem penulisan yang dilambangkan dalam bentuk not.

1.3.2 Piano

Piano adalah sebuah alat musik yangberbentuk persegi seperti meja yang bisa dimainkan dengan cara ditekan tuts-nya. Bunyi dari piano dihasilkan dari palu yang bergerak ketika tuts ditekan, palu tersebut akan memukul dawai sehingga menimbulkan bunyi (Rahmadi

2008). Jadi, Piano adalah salah satu alat musik yang memiliki tuts sebagai penentu nada mana yang akan kita keluarkan. Piano dimainkan dengan jari – jemari tangan. Piano dapat menghasilkan suara berdasarkan ketukan palu terhadap senar. Setiap nada yang dihasilkan memiliki frekuensi yang berbeda sesuai tinggi rendahnya nada. Piano terdiri dari 88 tuts, keseluruhan terdiri dari kurang lebih 8 oktaf. Nama – nama tuts di piano adalah C C# D D# E F F# G G# A A# B dimana kumpulan tuts ini membentuk 1 oktaf.

1.3.3 Kunci (Chord)

Kunci atau chord nada merupakan hidangan dua nada atau lebih secara bersama – sama sehingga menimbulkan kesan indah. Permainan chord digunakan untuk mengiringi jalannya suatu lagu. Mengetahui dan menghafal rumus pembentuk chord bisa dilakukan untuk dapat memainkan chord dengan baik. Untuk menentukan rangkaian akor terlebih dahulu menentukan nada dasar (do, re, mi, fa sol, la, si). Sebagai contoh untuk menentukan rangkaian akor C sebagai nada dasar, maka C=do; D=re; E=mi; F=fa; G=sol; A=la; B=si. Rangkaian akor selengkapnya akan ditunjukkan pada tabel sebagai berikut :

Nada Dasar (Do)	**Rangkaian Akor (Re, Mi, Fa, Sol, La, Si)**					
C	D	E	F	G	A	B
D	E	F#	G	A	B	C#
E	F#	G#	A	B	C#	D#
F	G	A	A#	C	D	E

Nada Dasar (Do)	Rangkaian Akor (Re, Mi, Fa, Sol, La, Si)					
G	A	B	C	D	E	F#
A	B	C#	D	E	F#	G#
B	C#	D#	E	F#	G#	A#

Tabel 3.1 Tabel Rangkaian Akor

Chord mayor adalah chord yang terdiri atas tiga nada yang dibentuk dengan rumus :

a. Nada pertama sebagai nada dasar.

b. Nada kedua berjarak 2 dari nada pertama.

c. Nada ketiga berjarak 1 ½ dari nada kedua.

Berbeda dengan chord mayor, chord minor adalah chord yang terdiri dari tiga nada yang dibentuk dengan rumus :

a. Nada pertama sebagai nada dasar.

b. Nada kedua berjarak 1 ½ dari nada pertama.

c. Nada ketiga berjarak 2 dari nada kedua (Sugeng 1966).

Dari pernyataan diatas terdapat perbedaan antara chord mayor dan chord minor, perbedaan keduanya terletak pada nada kedua, untuk nada pertama dan nada ketiga keduanya adalah sama. Berikut ini merupakan bentuk pola akor dasar dan kebalikannya :

Posisi Dasar (Simbol Huruf)	Paduan Nada	Balikan I	Balikan II
C mayor	C - E - G	E - G - C	G - C - E
C minor	C - D# - G	D# - G - C	G - C - D#
C# mayor	C# - F - G#	F - G# - C#	G# - C# - F
D mayor	D - F# - A	F# - A - D	A - D - F#
D minor	D - F - A	F - A - D	A - D - F
D# mayor	D# - G - A#	G - A# - D#	A# - D# - G
D# minor	D# - F# - A#	F# - A# - D#	A# - D# - F#

Posisi Dasar (Simbol Huruf)	Paduan Nada	Balikan I	Balikan II
E mayor	E - G# - B	G# - B - E	B - E - G#
E minor	E - G - B	G - B - E	B - E - G
F mayor	F - A - C	A - C - F	C - F - A
F minor	F - G# - C	G# - C - F	C - F - G#
F# mayor	F# - A# - C#	A# - C# - F#	C# - F# - A#
F# minor	F# - A - C#	A - C# - F#	C# - F# - A
G mayor	G - B - D	B - D - G	D - G - B
G minor	G - A# - D	A# - D - G	D - G - A#
G# mayor	G# - C - D#	C - D# - G#	D# - G# - C
G# minor	G# - B - D#	B - D# - G#	D# - G# - B
A mayor	A - C# - E	C# - E - A	E - A - C#
A minor	A - C - E	C - E - A	E - A - C
A# mayor	A# - D - F	D - F - A#	F - A# - D
A# minor	A# - C# - F	C# - F - A#	F - A# - C#
B mayor	B - D# - F#	D# - F# - B	F# - B - D#
B minor	B - D - F#	D - F# - B	F# - B - D

Tabel 3.2 Akor Dasar dan Kebalikannya

Chord adalah kumpulan tiga nada atau lebih yang dimainkan secara bersamaan atau terputus – putus, sehingga menjadi nada yang teratur dan suara yang terdengar harmonis. Chord banyak dimainkan untuk mengiringi atau membuat pola iringan untuk suatu lagu. Tetapi tidak jarang juga chord dimainkan sendiri oleh pianis dalam instrumennya agar instrument yang dihasilkan semakin tinggi nilai seni dan keharmonisannya (Aley 2010). Maka dari itu dengan adanya chord yang terdiri dari tiga nada atau lebih akan terdengar harmonis jika dimainkan secara bersamaan.

1.3.4 Audio Digital

Suara atau audio adalah getaran udara pada frekuensi yang dapat didengar oleh telinga manusia sehingga disebut dengan frekuensi suara atau frekuensi audio.Gendang telinga ini bergetar dan getaran ini dikirim dan diterjemahkan menjadi informasi suara yang dikirimkan ke otak, sehingga kita dapat mendengarkan suara. Frekuensi audio berada diantara 20Hz sampai dengan 20KHz. Karakteristik suara ditentukan antara lain oleh frekuensi, amplitude dan durasi.

Ada dua jenis pengolah audio yaitu audio analog dan audio digital. Audio analog adalah pengolahan suara asli (akustik) melalui peralatan elektronik analog, sedangkan audio digital adalah suara yang melalui pengolahan secara digital melalui komputer. Komputer hanya mampu mengenal sinyal dalam bentuk *digital*. Bentuk *digital* yang dimaksud adalah tegangan yang diterjemahkan dalam angka "0" dan "1", yang juga disebut dengan istilah "*bit*". Tegangan ini berkisar 5 volt bagi angka "1" dan mendekati 0 volt bagi angka "0". Dengan kecepatan perhitungan yang dimiliki computer, computer mampu melihat angka "0" dan "1" ini menjadi kumpulan *bit-bit* dan menerjemahkan kumpulan *bit-bit* tersebut menjadi sebuah informasi yang bernlai.

Cara untuk memasukkan suara *analog* ini sehingga dapat dimanipulasi oleh peralatan elektronik adalah *transducer*. Dalam hal ini, *transducer* adalah istilah untuk menyebut sebuah peralatan yang dapat mengubah tekanan udara (yang kita dengar sebagai suara) ke dalam

tegangan elektrik yang dapat dimengerti oleh perangkat elektronik, serta sebaliknya. Contoh *transducer* adalah mikrofon dan *speaker*. Mikrofon dapat mengubah tekanan udara menjadi tegangan elektrik, sementara *speaker* melakukan pekerjaan sebaliknya (Binanto 2010).

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

Proses pengubahan dari tegangan *analog* ke data *digital* ini terdiri atas beberapa tahap sebagai berikut :

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

Proses sebaliknya yaitu pengubahan dari data digital menjadi tegangan analog juga terdiri atas beberapa tahap sebagai berikut :

1. Menghitung data *digital* menjadi amplitudo – amplitudo *analog*.
2. Menyambung amplitude *analog* ini menjadi sinyal *analog*.
3. Memfilter keluaran dengan *Low Pass Filter* sehingga bentuk gelombang keluaran menjadi lebih mulus. (Binanto 2010)

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

1.3.5 File Wav

File WAV adalah file audio standar yang digunakan oleh Windows, suara yang berupa digital audio dalam file WAV disimpan dalam bentuk gelombang karena itulah file ini memiliki ekstensi .WAV (Wave). Data digital audio dalam file WAV bisa memiliki kualitas yang bermacam – macam. Kualitas dari suara yang dihasilkan ditentukan dari *bitrate, samplerate,* dan jumlah *channel*.

Bitrate merupakan ukuran bit tiap sampelnya, yaitu 8 bits, 16 bits, 24 bits atau 32 bits. Dalam 8 bits WAV semua samplenya hanya akan memakan sebanyak 1 byte saja. Sedangkan untuk 16 bits akan memakan 2 bytes. WAV 16 bits tiap samplenya memiliki nilai antara -32768 sampai 32767, dibandingkan untuk 8 bits yang hanya memiliki

nilai antara antara -128 sampai 127 maka 16 bits WAV file menghasilkan suara yang lebih baik karena datanya lebih akurat daripada 8 bits. Namun, pada kenyataannya telinga manusia tidak mampu membedakan suara yang dihasilkan.

Samplerate menyatakan banyaknya jumlah sample yang dimainkan setiap detiknya. Samplerate yang umum dipakai adalah 8000Hz, 11025Hz, 22050Hz dan 44100Hz. Untuk 8000Hz biasanya disebut sebagai *Telephone quality* karena suara yang dihasilkan menyerupai suara dari telepon. Samplerate 11025Hz sering digunakan untuk merekam suara manusia karena dinilai sudah cukup bagus. Tetapi untuk musik agar dapat memperoleh hasil yang baik maka digunakan samplerate 22050Hz. Sedangkan 44100Hz merupakan *CD quality*, samplerate 44100Hz inilah yang digunakan dalam audio CD karena bagus untuk semua jenis suara. Berbeda dengan bitrate, perbedaan samplerate dapat dikenali dengan mudah, karena tentu saja suara dari radio dengan suara CD player sangat jauh berbeda.

Jumlah channel menentuka suara yang dihasilkan apakah mono atau stereo. Mono memiliki hanya 1 channel, sedangkan stereo 2 channel dan memakan tempat 2 kali lebih bnyak daripada mono. Untuk merekam suara manusia, mono sudah cukup memberikan kualitas yang baik, sedangkan audio menggunakan 2 channel (Ibnu Gunawan 2005).

1.3.6 Struktur File Wav

Aplikasi *multimedia* seperti diketahui memerlukan manajemen penyimpanan dari sejumlah jenis data yang bervariasi, termasuk *bitmap*, data *audio*, data *video*, informasi mengenai control *device* peripheral. *Rule Interchange File Format* (RIFF) menyediakan suatu cara untuk menyimpan semua jenis data tersebut. Tipe data pada sebuah *file* RIFF dapat diketahui dari ekstensi *filenya* (Ibnu Gunawan 2005). Sebagai contoh jenis – jenis *file* yang disimpan dalam bentuk format RIFF adalah sebagai berikut :

1. *Audio / visual interleaved data* (.AVI)
2. *Animated cursor* (.ANI)
3. *Wavfrom data* (.WAV)
4. *Multimedia Movie* (.PAL)
5. *Bitmapped data* (.RDI)
6. *Color palette* (.PAL)
7. *MIDI information* (.RMI)

Format *file* WAV seperti yang diketahui, merupakan bagian dari spesifikasi RIFF Microsoft yang digunakan sebagai penyimpanan *data digital audio*. Format *file* ini merupakan salah satu format *file audio* pada PC. Seiring dengan popularitas Windows maka banyak aplikasi yang mendukung format *file* ini. Karena bekerja pada lingkungan Windows yang menggunakan prosesor Intel, maka format data dari *file*

WAV disimpan dalam format urutan *little-endian* (*least significant byte*) dan sebagian dalam urutan *big-endian*.

Chunk pada *file* RIFF merupakan suatu *string* yang harus diatur untuk tiap kata. Ini berarti ukuran total dari *chunk* harus merupakan kelipatan dari 2 *byte* (seperti 2, 4, 6, 8 dan seterusnya). Jika suatu *chunk* terdiri atas jumlah *byte* yang ganjil maka harus dilakukan penambahan *byte* (*extra padding byte*) dengan menambahkan sebuah nilai nol pada *byte data* terakhir. *Extra padding byte* ini tidak ikut dihitung pada ukuran *chunk*. Oleh karena itu sebuah program harus selalu melakukan pengaturan kata untuk menentukan ukuran nilai dari *header* sebuah *chunk* untuk mengkalkulasi *offset* dari *chunk* berikutnya (Ibnu Gunawan 2005). Penjelasan mengenai struktur *file* Wav yang dimulai dengan *header* RIFF dapat dilihat pada tabel 3.3 :

Offset	Size	Nama Field	Deskripsi
0	4	**ChunkID**	Terdiri atas kata "RIFF" dalam bentuk ASCII (0x52494646 dalam bentuk *big-endian*).
4	4	**Chunksize**	36 + SubChunk2Size atau lebih tepatnya: 4 + (8 + SubChunk1Size) + (8 + SubChunk2Size). Ini adalah besar seluruh *file* dalam *byte* dikurangi 8 *byte* untuk 2 *field* yang tidak termasuk dalam hitungan: ChunkID dan ChunkSize
8	4	**Format**	Terdiri atas kata "WAV" (0x57415645 dalam bentuk *big-endian*).
12	4	**SubChunk1ID**	Terdiri atas kata "fmt " (0x666d7420 dalam bentuk *big-endian*).
16	4	**SubChunk1Size**	16 untuk jenis PCM.
20	2	**AudioFormat**	PCM = 1 (*Linear quantization*). Nilai lebih dari 1 mengindikasikan *file Wav* kompresi.
22	2	**NumChannels**	*Mono* = 1, *Stereo* = 2 dan seterusnya
24	4	**SampleRate**	8000, 44100, dan seterusnya dalam satuan Hz
28	4	**ByteRate**	= SampleRate * NumChannels * BitsPerSample / 8
32	2	**BlockAlign**	= NumChannels * BitsPerSample / 8 Jumlah *byte* untuk satu sampel termasuk semua *channel*.
34	2	**BitsPerSample**	8 bits = 8, 16 bits = 16, dan seterusnya.
36	4	**SubChunk2ID**	Terdiri atas kata "data" (0x64617461 dalam bentuk *big-endian*).
40	4	**SubChunk2Size**	= NumSamples * NumChannels * BitsPerSample / 8
44	*	**Data**	*Data Sound* sebenarnya.

Tabel 3.3 Penjelasan Struktur File Wav

1.3.6.1 Header File Wav

Header File Wav mengikuti struktur format *file* RIFF standar. 8 *byte* pertama dalam *file* adalah *header chunk* RIFF standar yang mempunyai *chunk ID* "RIFF" dan ukuran *chunk* didapat dengan mengurangkan ukuran *file* dengan 8 *byte* yang digunakan sebagai *header*. 4 *byte data* yaitu kata "RIFF" menunjukkan bahwa *file* tersebut merupakan *file* RIFF. *File Wav*

selalu menggunakan kata “WAV” untuk membedakannya dengan jenis *file* RIFF lainnya sekaligus diunakan untuk mendefinisikan bahwa *file* tersebut merupakan *file audio wavform* (Ibnu Gunawan 2005). Sebagaimana dapat dilihat pada tabel 3.4 :

Offset	Ukuran	Deskripsi	Nilai
0x00	4	Chunk ID	"RIFF" (0x52494646)
0x04	4	Ukuran Data Chunk	(ukuran *file*) - 8
0x08	4	Jenis RIFF	"WAV" (0x57415645)
0x10	Chunk WAV		

Tabel 3.4 Nilai Jenis Chunk RIFF

1.3.6.2 Chunk File Wav

Ada beberapa jenis chunk untuk menyatakan file Wav. Kebanyakan file Wav hanya terdiri atas 2 buah chunk, yaitu Chunk Format dan Chunk Data. 2 jenis chunk ini diperlukan untuk menggambarkan format dari sampel digital audio. Meskipun tidak diperlukan untuk spesifikasi file Wav yang resmi, lebih baik menempatkan Chunk Format sebelum Chunk Data. Kebanyakan program membaca chunk tersebut dengan urutan di atas dan jauh lebih mudah dilakukan streaming digital audio dari sumber yang membacanya secara lambat dan linear seperti Internet. Jika Chunk Format lebih dulu ditempatkan sebelum Chunk Data maka semua data dan format

harus di-stream terlebih dahulu sebelum dilakukan playback (Ibnu Gunawan 2005). Sebagaimana dapat dilihat pada tabel 3.5 :

Offset	Ukuran*	Deskripsi
0x00	4	Chunk ID
0x04	4	Ukuran Data Chunk
0x08	*Byte Data Chunk*	

Tabel 3.5 Format Chunk RIFF

1.3.6.3 Chunk Format

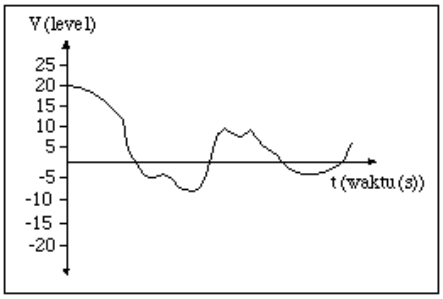
Chunk format terdiri atas informasi tentang bagaiman suatu data wavform disimpan dan cara untuk dimainkan kembali, termasuk jenis kompresi yang digunakan, jumlah kanal, laju pencuplikan (sampling rate), jumlah but tiap sampel dan atriut lainnya. Chunk format ini ditandai dengan chunk ID "fmt" (Ibnu Gunawan 2005). Sebagaiman dapat dilihat pada tabel 3.6 :

Offset	Ukuran	Deskripsi	Nilai
0x00	4	Chunk ID	"fmt " (0x666D7420)
0x04	4	Ukuran Data Chunk	16 + *extra format bytes*
0x08	2	Kode Kompresi	1 - 65,535
0x0a	2	Jumlah Kanal	1 - 65,535
0x0c	4	Laju Pencuplikan	1 - 0xFFFFFFFF
0x10	4	Jumlah rata-rata *byte* tiap detik	1 - 0xFFFFFFFF
0x14	2	*Block align*	1 - 65,535
0x16	2	Bit *Significant* tiap sampel	2 - 65,535
0x18	2	*Extra format bytes*	0 - 65,535
0x1a	*Extra format bytes*		

Tabel 3.6 Nilai – Nilai Chunk Format File Wav

1.3.7 Sampling

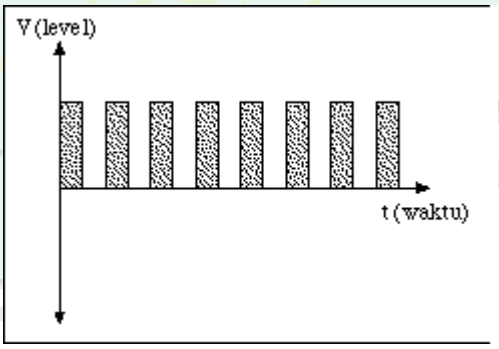
Pada dasarnya semua suara audio, baik vokal maupun nada merupakan bentukan dari suatu getaran. Ini menandakan nada memiliki bentuk gelombangnya masing– masing. Bentukan suatu gelombang umumnya disebut dengan sinyal analog. Gambar dari sinyal analog dapat dilihat pada gambar dibawah ini :



Gambar 3.1 Sinyal Analog

Terdapat sebuah teknik yang memungkinkan sinyal analog tersebut diubah dan diproses menjadi bit – bit digital. Teknik tersebut yaitu teknik sampling. Apabila sinyal analog menjadi sinyal digital maka sinyal ini jauh lebih baik, sedikit noisenya dan juga dapat diproses dengan mudah. Digital Signal Processing merupakan perkembangan dari teknik sampling yang memungkinkan untuk membentuk sample – sample berupa suara seperti yang ada pada piano atau keyboard.

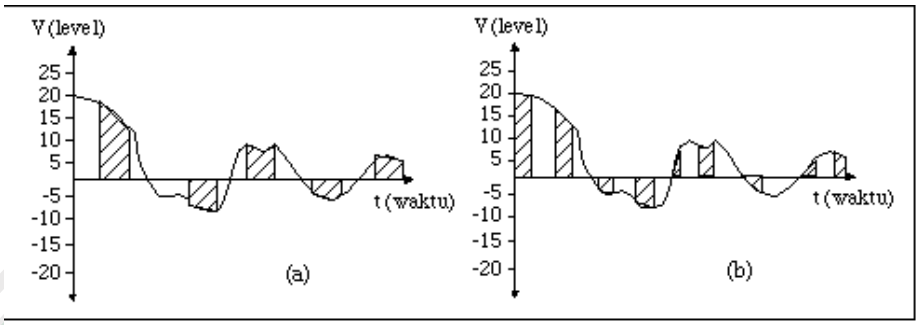
Pada tahap ini terjadi suatu pengambilan sampel dari bentukan sinyal analog. Pengambilan dilakukan pada bagian – bagian sinyal analog dengan pembagian sampel, karena sampel yang di ambil akan lebih menggambarkan sinyal yang asli. Contoh dari bentukan sinyal sample dapat dilihat pada gambar dibawah ini :



Gambar 3.2 Bentukan Sinyal Sample

Teori Shannon menyatakan frekuensi sinyal sampling ini paling sedikit adalah 2 kali frekuensi sinyal yang akan disampling (sinyal analog). Hal tersebut merupakan batas minimum dari frekuensi sample agar nanatinya cuplikan yang diambil menunjukkan bentukan sinyal yang asli (analog). Lebih besar frekuensi sampling tentunya lebih

baik, karena cuplikan akan lebih menggambarkan sinyal yang asli. Seperti terlihat pada gambar dibawah ini :



Gambar 3.3 (a) Bentukan Proses Pencuplikan (b) Bentukan Proses Pencuplikan yang Lebih Besar

Setelah pembagian sampel, maka terbentuklah suatu sinyal analog-diskrit yang bentuknya menyerupai aslinya namun hanya diambil diskrit – diskritnya saja. (Akbar 2011)

1.3.8 Frame Blocking

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 dua syarat yaitu linear dan time invariant (Syahbani 2010). Penentuan jumlah data pada setiap frame dihitung menggunakan persamaan :

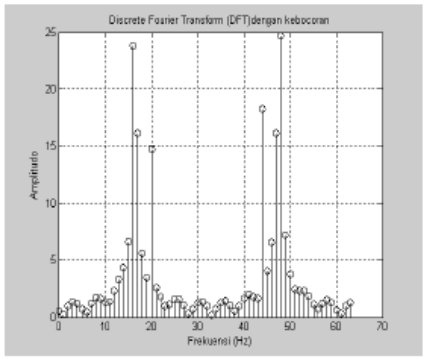
$$N = \frac{fr}{1000} \times fs$$

Dimana *N* adalah jumlah data per *frame*, *fr* adalah lebar waktu *frame* dan *fs* adalah frekuensi suara. (Peggy 2012)

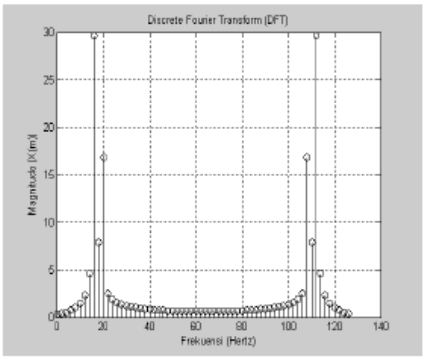
1.3.9 Windowing

Fenomena kebocoran sinyal muncul ketika frekuensi dari sinyal input bukan merupakan bilangan bulat N ≠ fs, seperti tampak pada gambar berikut ini :

Fenomena kebocoran



a) Sinyal ketika frekuensi bukan bilangan bulat dengan *N* (banyaknya data penyamplingan) = 64, f_s (*sampling rate*) = 64 Hz, f_1=16,4 Hz, f_2=20 Hz. (frekuensi masukan).



Sinyal ketika *N* (banyaknya data penyamplingan) tidak sama dengan f_s (*sampling rate*). Dalam hal ini *N* = 64, f_s=128 Hz, f_1=16,4 Hz, f_2=20 Hz. (frekuensi masukan).

Gambar 3.4 DFT resolusi 1 Hz

Pada prinsipnya penggunaan fungsi *windowing* adalah dengan cara melewatkan sinyal yang mempunyai frekuensi sembarang dikonvolusikan dengan fungsi window tertentu sehingga dapat mereduksi sinyal – sinyal yang tergolong bocor sebelum dilakukan proses transformasi. Ada beberapa fungsi *window* yang telah ada, diantaranya *Hann, Hamming, Tringular, Rectangular*. Secara matematis penggunaan windowing pada DFT dapat dinyatakan sebagai berikut :

$$X(m) = \sum_{m=0}^{N-1} w(n) * x(n) W_N^{nm}$$

Berikut ini beberapa jenis fungsi window yang telah didefinisikan :

1. Hann window

   $w(n) = 0{,}5 - 0{,}5 \cos(2\pi n / N - 1)$ untuk $n = 0,1,2,...,N-1$

2. Hamming window

   $w(n) = 0{,}54 - 0{,}46 \cos(2\pi n / N - 1)$ untuk $n = 0,1,2,...,N-1$

3. Tringular window

$$w(n) = \frac{n}{N/2} \text{ untuk } n = 0,1,2,\ldots,N/2$$

$$w(n) = 2 - \frac{n}{N/2} \text{ untuk } n = N/2+1,\ N/2+2,\ \ldots N-1$$

4. Rectangular window

   $w(n) = 1$ untuk $n = 0,1,2,...,N-1$ (Riyanto 2009)

1.3.10 Fast Fourier Transform

Fast Fourier Transform adalah suatu algoritma yang digunakan untuk merepresentasikan sinyal dalam domain waktu diskrit dan domain frekuensi. Membahas mengenai FFT tentunya tidak dapat dilepaskan dari DFT (Discrete Fourier Transform). FFT merupakan turunan dari persamaan DFT dimana jumlah perhitungan digital pada DFT dapat dikurangi secara significant sehingga dengan adanya penemuan FFT maka perhitungan digital terhadap spektrum-spektrum frekuensi dapat diwujudkan secara sederhana dalam implementasinya.

DFT untuk finite-length sequence y(n) yang terdefinisi untuk rentang 0 ≤ n ≤ N- 1 dinyatakan sebagai berikut:

$$y(t) = y(0)\delta(0) + y(T_s)\delta(t - T_s) + y(2T_s)\delta(t - 2T_s) + y\big((N - 1)T_s\big)\delta(t - (N - 1)T_s) \quad \ldots\ldots\ldots\ldots\ldots \quad (1)$$

dimana $\frac{1}{T_s} \geq 2f_{max}$.

Berdasarkan teorema Nyquist, persamaan diatas adalah persamaan penjumlahan fungsi delta yang mengalami pergeseran waktu, dimana masing-masing komponen sinyal memiliki amplitudo sebesar $y(nT_s)$. Transformasi Fourier dari sinyal tersebut sama dengan transformasi Fourier dari sinyal yang tidak mengalami pergeseran waktu seperti contoh:

$$F\{y(nT_s)\delta(0)\} = y(nT_s)F\{\delta(0)\} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

Persamaan di atas dikalikan dengan factor pergeseran waktu masing-masing komponen,

$$Y(\omega) = y(0) + y(T_s)e^{-j\omega T_s} + \cdots + y((N - 1_{T_s})e^{-j\omega(N-1)T_s}$$

$$Y(\omega) = \sum_{n=1}^{N-1} y(nT_s)e^{-j\omega T_s} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Karena $f = \frac{\omega}{2\pi}$ adalah variable diskrit, maka dapat dilakukan penyederhanaan dengan mengambil sinyal sampel yang hanya mengadopsi nilai-nilai diskrit saja seperti:

$$0, \frac{1}{NT_s}, \frac{2}{NT_s}, \ldots, \frac{N-1}{NT_s} = \frac{k}{NT_s} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

dimana k = 0, 1, 2, 3, …, N-1 sehingga persamaan (3) di atas menjadi:

$$Y\left(\frac{2\pi k}{NT_s}\right) = \sum_{n=0}^{N-1} y(nT_s)e^{-j\frac{2\pi k}{NT_s}nT_s} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

karena k = 0, 1, 2, 3, …, N-1 sehingga persamaan (4) dapat disederhanakan:

$$Y(k) = \sum_{n=0}^{N-1} y(n){w_N}^{nk} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Dengan $w_N = e^{-j\frac{2\pi}{N}}$, dimana w_N adalah twiddle factor.

Persamaan (6) di atas, untuk contoh kasus N=4, perhitungan komponen-komponen simyalnya dapat diuraikan sebagai berikut:

$$Y(0) = y_0W^0 + y_1W^0 + y_2W^0 + y_3W^0$$

$$Y(1) = y_0W^0 + y_1W^1 + y_2W^2 + y_3W^3$$

$$Y(2) = y_0W^0 + y_1W^2 + y_2W^4 + y_3W^6$$

$$Y(0) = y_0W^0 + y_1W^3 + y_2W^6 + y_3W^9$$

atau disederhanakan menjadi:

$$Y(k) = y(n){w_N}^{nk} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Berdasarkan persamaan (2.7) di atas, perhitungan DFT memerlukan N2 perkalian kompleks dan N(N-1) penjumlahan kompleks. Namun, ada suatu metode yang dikembangkan untuk mengurangi kompleksitas algoritma menjadi hanya Nlog2N. Teknik tersebut disebut sebagai algoritma Fast Fourier Transform (FFT). Prinsip dasar FFT adalah menguraikan penghitungan N-titik DFT

menjadi penghitungan DFT dengan ukuran yang lebih kecil dan memanfaatkan periodisitas dan simetri dari bilangan kompleks $w_N{}^{nk}$.

FFT dapat dituliskan dalam bentuk sinusoidal sebagai berikut:

$$x(k) = \sum_{n=0}^{N-1} x(n) \sin\left(\frac{2\pi kn}{N}\right) + j \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi kn}{N}\right) \ldots (8)$$

dimana :

x(n) adalah koefisien sinus dan cosinus pada 2π k/N,

k adalah indeks dari frekuensi pada frekuensi ke-N, dan

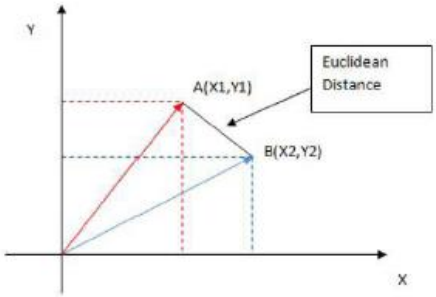
n adalah indeks waktu,

x(k) adalah nilai dari spektrum ke-k (domain frekuensi)

x(n) adalah nilai sinyal pada domain waktu. (Akbar 2011)

1.3.11 Euclidean Distance

Dalam matematika, euclidean distance atau jarak antara dua titik yang dapat diukur dan dihasilkan oleh formula pytagoras. Euclidean distance vector atau sering hanya disebut dengan vector adalah objek geometri yang memiliki panjang (magnitude) dan arah (direction). Sedangkan ruang vektor adalah sebuah struktur matematika yang dibentuk oleh sekumpulan vektor. Vektor – vektor tersebut dapat ditambahkan, dikalikan dengan bilangan real dan lain – lain. Gambar dibawah ini merupakan contoh dari ruang vektor :



Gambar 3.5 Vektor Space 2 Dimensi pada Euclidean Distance

Pada ruang vektor tersebut terdapat dua vektor yaitu vektor A dan vektor B. untuk menghitung jarak antara vektor A dan vektor Bdigunakan persamaan *euclidean distance*.

Berikut dengan penyelesaian dalam menghitung jarak antara vektor A dan vektor B. Panjang vektor A dan B dapat didefinisikan sebagai berikut x :

$||A|| = \sqrt{X1^2 + Y1^2}$ (Persamaan 2.2)

$||B|| = \sqrt{X2^2 + Y2^2}$ (Persamaan 2.3)

Dengan demikian, untuk menghitung jarak antara kedua vektor tersebut menggunakan persamaan sebagai berikut :

$d(\bar{A}, \bar{B}) = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$ (Persamaan 2.4)

(Hadyansyah 2010)

1.3.12 Android

Android adalah platform / system operasi untuk perangkat selular (khususnya smartphone) yang berbasis linux. Seperti halnya produk – produk system operasi seperti Microsoft Windows (95, 98, XP, Vista, 7 dan lain lain). Perbedaannya bahwa Android berjalan di

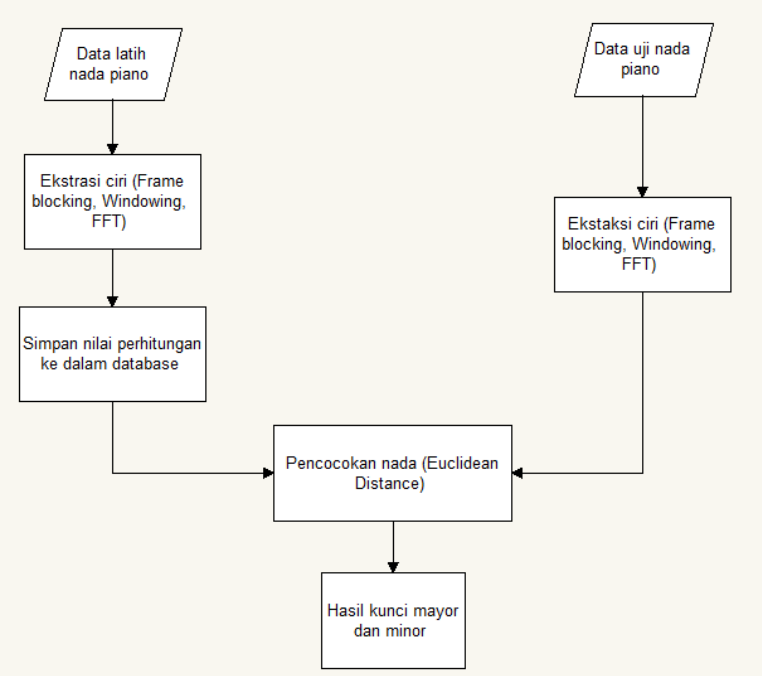
perangkat berbasi mobile / telepon selular. Jika dibandingkan dengan system operasi untuk perangkat mobile lain seperti Symbian, Microsoft Windows mobile, IOS (IPhone OS), Mobile Linux dan lain sebagainya, Android memiliki sedikit kelebihan yaitu bersifat open source, dan telah didukung standard penerbitan API (Application Programming Interface) yang dimanfaatkan secara keseluruhan dengan biaya relatif murah.

Platform android pertama kali dikembangkan olehy perusahaan Android inc yang merupakan sebuah perusahaan baru yang bergerak di bidang perangkat lunak untuk ponsel. Pada tahun 2005 Android inc dibeli oleh Google. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance mendukung pengembangan standar terbuka pada perangkat selular. Di sisi lain, Google merilis kode – kode Android di bawah lisensi Apache, Sebuah lisensi perangkat lunak dan standar terbuka perangkat selular, Sehingga terdapat dua jenis distributor system operasi Android yaitu mendapat dukungan penuh dari Goggle dan yang mendapat dukungan penuh dari Open Handset Distribution (OHD) (Mulyadi 2010).

1.4 Kerangka Konsep

1.4.1 Alur Kerja

Diagram Proses perancangan sistem pengenalan nada piano akan dibahas sesuai blok diagram sistem dibawah ini :



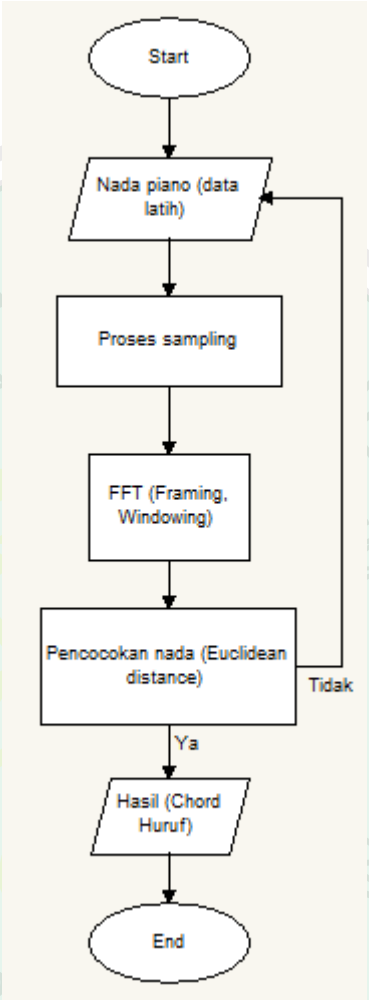
Gambar 3.6 Diagram Blok Sistem

Gambar 3.4.1 diatas merupakan rangkaian secara umum dari diagram blok sistem. Pada diagram tersebut terdapat input dari nada piano yang berasal dari nada piano secara langsung, baik dari alat musik piano yang dimainkan maupun dari suatu alunan musik piano. Adapaun penjelasan dari diagram blok diatas adalah sebagai berikut :

- **Data latih nada piano** adalah masukan nada berupa file .wav. Data tersebut direkam menggunakan software *cool edit pro 2* pada ruang kedap suara untuk menghindari noise yang tidak dibutuhkan.
- **Ekstraksi ciri** merupakan proses sinyal nada piano yang masuk berupa sinyal analog yang kemudian diubah menjadi sinyal diskrit menggunakan proses sampling. Bentukan sinyal diskrit nantinya lebih mudah untuk diproses menjadi deretan nilai matrik dari proses frame blocking, windowing dan FFT.

- **Simpan nilai ke dalam database** merupakan hasil nilai dari perhitungan FFT yang digunakan sebagai nilai referensi atau nilai acuan untuk mencocokkan nilai dari data uji.
- **Data uji nada piano** sama dengan data latih. Data tersebut diperoleh dengan cara merekam kunci nada piano menggunakan software perekam. Data tersebut juga dapat diperoleh dari internet yang berformat .wav.
- **Pencocokan nada** adalah proses mencocokkan nilai FFT dari data uji dengan data latih. Proses tersebut dihitung menggunakan *Euclidean Distance*. Hasil nilai dibandingkan dengan 3 nada yang berdekatan. Kunci nada yang sesuai ditentukan berdasarkan jarak terkecil.

1.4.2 Alur Kerja Proses Program



Gambar 3.7 Alur Kerja Proses Program

Pada alur kerja proses program diatas merupakan tahap – tahap pemrosesan data sinyal nada yang masuk hingga membentuk sebuah nilai frekuensi dan ditampilkan dalam bentuk chord huruf. Berikut ini merupakan penjelasan tiap proses :

1.4.2.1 Proses Sampling

Pada proses sampling ini adalah proses pembacaan dari input file audio wav. Pembacaan akan disimpan ke dalam array, sehingga file

wav dapat diproses melalui penyimpanan ini. Langkah pertama yang harus dilakukan adalah pembuatan tipe data dasar untuk mengambil informasi yang ada pada header file wav.

Pada penelitian ini, dilakukan percobaan mengambil sampel tiap 1 detik dengan frekuensi sampling sebesar 44100Hz yang artinya terdapat 44100 sampel tiap detik.

1.4.2.2 Frame Blocking

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 beberapa detik suara akan disampel dan berapa besar frekuensi sampling. Pada proses ini dilakukan pemotongan sinyal dalam slot – slot tertentu agar memenuhi dua syarat yaitu linear dan time variant.

Contoh perhitungan frame blocking :

Signal : [1, 5, 8]

Misal dengan waktu tiap 800ms dan sampel rate 44100 Hz

Maka didapatkan frame size (N)

N = 44100*0.08 = 3528 sampel

Overlapping frame (M) = 1764

x(n) = y (M+n)

x(1) = 1 (1764+1) = 1765

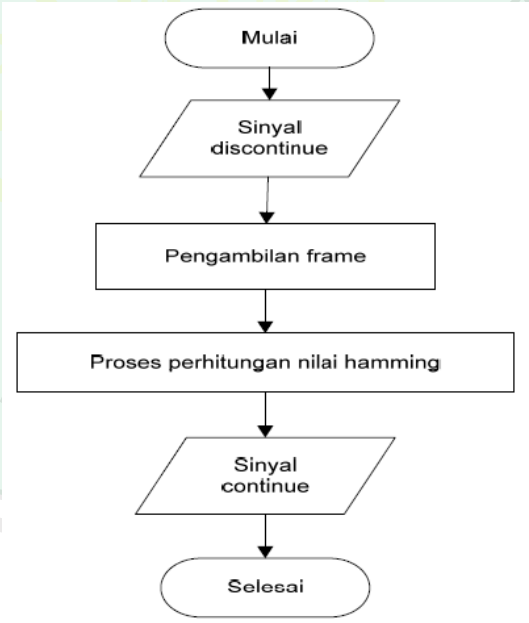
x(2) = 5 (1764+2) = 8830

x(3) = 8 (1764+3) = 14136

1.4.2.3 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.

Pada penelitian ini penulis menggunakan Hamming Window, proses window hamming di gambarkan sebagai berikut :



Gambar 3.8 Flowchart Window Hamming

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

$w[n] = 0.54 – 0.46 * \cos(2\pi n/(N-1))$

Contoh perhitungan Hamming Window :

Diketahui : $\pi = 3.14$

n = 1

N = 3528

Ditanya : w(n)

Jawab : $w(n) = 0.54 – 0.46 \cos(2\pi n/(N-1))$

$w(1) = 0.54 – 0.46 \cos(2*3.14*1/(3528-1))$

$w(1) = 0.54 – 0.46 \cos 1{,}78$

$w(1) = 0.54 – 0.46*1$

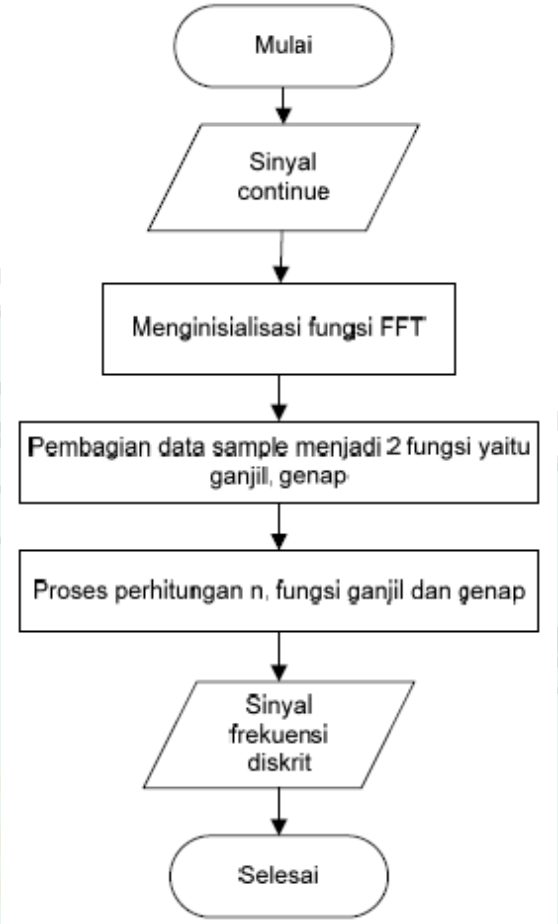
$w(1) = 0.08$

$w(1)*x(1) = 0.08 * 1765 = 141{,}12$ dst

Hasil dari *Hamming Window* adalah [141,12;706,4;1130,88]

1.4.2.4 Fast Fourier Transform

Proses Windowing menghasilkan spektrum suara dalam domain waktu, untuk tidak terjadi kesalahan dalam proses pencocokan nada maka spektrum domain waktu diubah menjadi sinyal frekuensi dengan menggunakan proses Fast Fourier Transform. FFT merupakan salah satu metode untuk transformasi sinyal suara menjadi sinyal frekuensi. Artinya, proses perekaman 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. Berikut ini merupakan diagram alir FFT :



Gambar 3.9 Diagram Alir FFT

Contoh perhitungan FFT :

$$X[k] = \sum_{n=1}^{N=1} x(n) W_N^{kn}$$

$$X[k] = \sum_{n=1}^{N=1} x(n) \cos\left(2 * \pi * k * \frac{n}{N}\right) - jSin\ (2 * \pi * k * \frac{n}{N})$$

$$x(1) = \frac{1}{3}\ [\ 141{,}12(\cos(\frac{2\pi*1*1}{3})] - jsin\frac{2\pi*1*1}{3} +$$

$$[\ 706{,}4(\cos(\frac{2\pi*1*2}{3})] - jsin\frac{2\pi*1*2}{3} +$$

$$[\ 1130{,}88(\cos(\frac{2\pi*1*3}{3})] - jsin\frac{2\pi*1*3}{3} = 656{,}49 - 0{,}073j$$

$$x(2) = \frac{1}{3}\ [\ 141{,}12(\cos(\frac{2\pi*2*1}{3})] - jsin\frac{2\pi*2*1}{3} +$$

$$[\ 706{,}4(\cos(\frac{2\pi*2*2}{3})] - jsin\frac{2\pi*2*2}{3} +$$

$$[\ 1130{,}88(\cos(\frac{2\pi*2*3}{3})] - jsin\frac{2\pi*2*3}{3} = 647{,}77 - 0{,}1454j$$

$$x(3) = \frac{1}{3}\ [\ 141{,}12(\cos(\frac{2\pi*3*1}{3})] - jsin\frac{2\pi*3*1}{3} +$$

$$[\ 706{,}4(\cos(\frac{2\pi*3*2}{3})] - jsin\frac{2\pi*3*2}{3} +$$

$$[\ 1130{,}88(\cos(\frac{2\pi*3*3}{3})] - jsin\frac{2\pi*3*3}{3} = 633{,}21 - 0{,}2167j$$

Gunakan rumus |[R² + I²] ½|

$$x(1) = |\sqrt{656{,}49^2 + 0{,}073^2}| = |656{,}49| = 656{,}49$$

$$x(2) = |\sqrt{647{,}77^2 + 0{,}1454^2}| = |647{,}77| = 647{,}77$$

$$x(3) = |\sqrt{633{,}21^2 + 0{,}2167^2}| = |633{,}21| = 633{,}21$$

1.4.2.5 Similarity Matching (Pencocokan Nada)

Pada proses matching ini menggunakan rumus perhitungan Euclidean Distance (ED) untuk menghitung jarak dari 2 buah frekuensi. Rumus yang digunakan adalah sebagai berikut :

$$\boldsymbol{ED} = \sqrt{(X1 - Y1)^2 + (X2 - Y2)^2 + (X3 - Y3)^2 \ldots (Xn - Yn)^2}$$

Dimana :

X1, X2, X3 = titik pertama (nilai data uji)

Y1, Y2, Y3 = titik kedua (nilai data latih)

Hasil frekuensi data latih yang didapatkan adalah [656,49; 647,77; 633,21]. Dari ketiga nilai tersebut, akan dihitung jarak terdekat dari nilai data uji. Sebagai contoh nilai data uji yang mendekati adalah pada kunci C3 mayor, C3 minor dan C#3 mayor. Ketiga kunci tersebut memiliki nilai *array* masing - masing. Contoh C3 mayor = [656,49; 657,77; 633,21], C3 minor = [655,49; 637,77; 622,21], C#3 mayor = [646,46; 637,99; 630,20]. Proses perhitungannya adalah sebagai berikut:

C3 mayor =

$$\sqrt{(656{,}49-656{,}49)^2+(647{,}77-657{,}77)^2+(633{,}21-633{,}21)^2}$$

$= \sqrt{100}$

= 10

C3 minor =

$$\sqrt{(656{,}49-655{,}49)^2+(647{,}77-637{,}77)^2+(633{,}21-622{,}21)^2}$$

$= \sqrt{222}$

= 14,899

C#3 mayor =

$$\sqrt{(656{,}49-646{,}46)^2+(647{,}77-637{,}99)^2+(633{,}21-630{,}20)^2}$$

$= \sqrt{205{,}3094}$

= 14,328

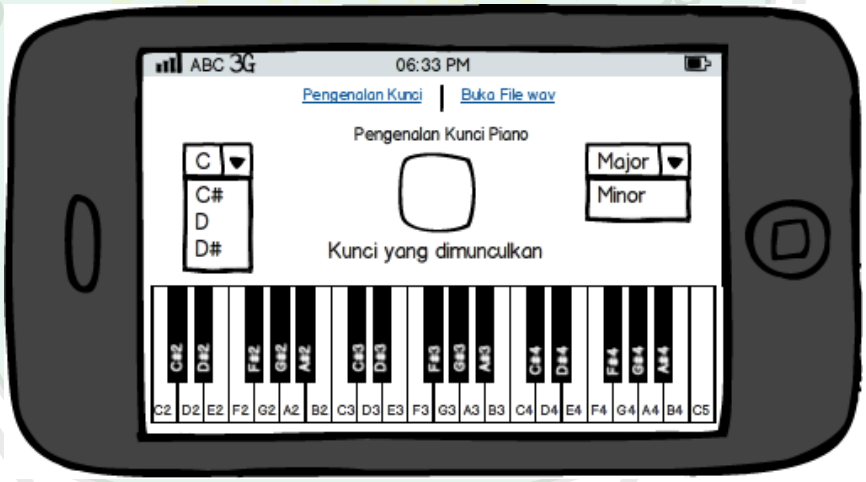
Berdasarkan jarak euclidean distance diatas, jarak frekuensi nada masukan dengan frekuensi referensi dari kunci C3 mayor adalah 10, C3 minor adalah 14,899 dan C#3 mayor 14,328. Jarak terkecil yang

mendekati 0 adalah pada kunci C_3 mayor. Jadi, frekuensi nada masukan sesuai dengan nada C_3 mayor.

1.4.3 Desain Interface

Berikut ini merupakan desain interface yang akan digunakan sebagai tampilan pada aplikasi pengenalan nada piano berbasis android. Desain interface ini menggunakan *software Balsamiq Mockup*. Adapun tampilannya adalah sebagai berikut :

1.4.3.1 Tampilan Pengenalan Kunci

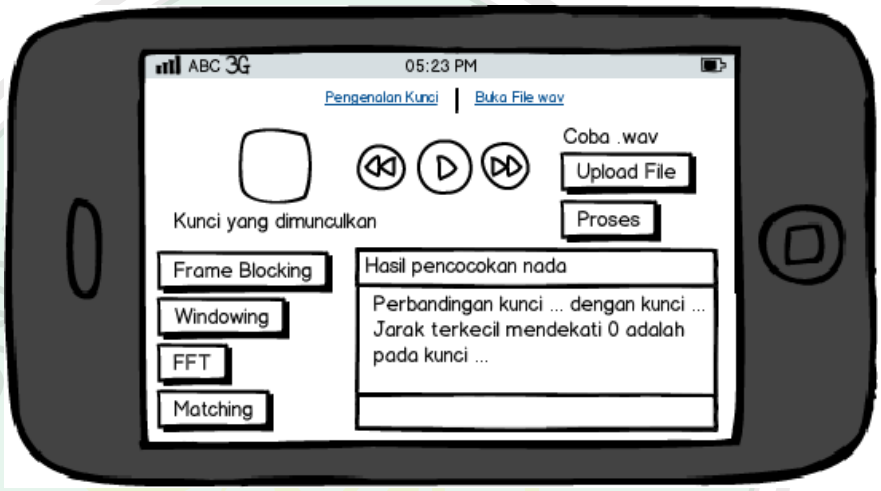


Gambar 3.10 Desain Interface Pengenalan Kunci Piano

Pada gambar diatas terlihat simulasi piano dari tuts nada C2. Pada simulasi tersebut nantinya terdiri dari 61 tuts atau 5 oktaf, dari nada C2 sampai dengan C7. Pada tampilan combo box nantinya terdiri dari kunci C sampai dengan B dan combo box yang kedua terdapat pilihan kunci major dan minor. Jadi, misal ketika user memilih untuk

menampilkan kunci C minor, maka pada simulasi piano akan ditampilkan tuts mana saja yang ditekan saat memainkan kunci C minor.

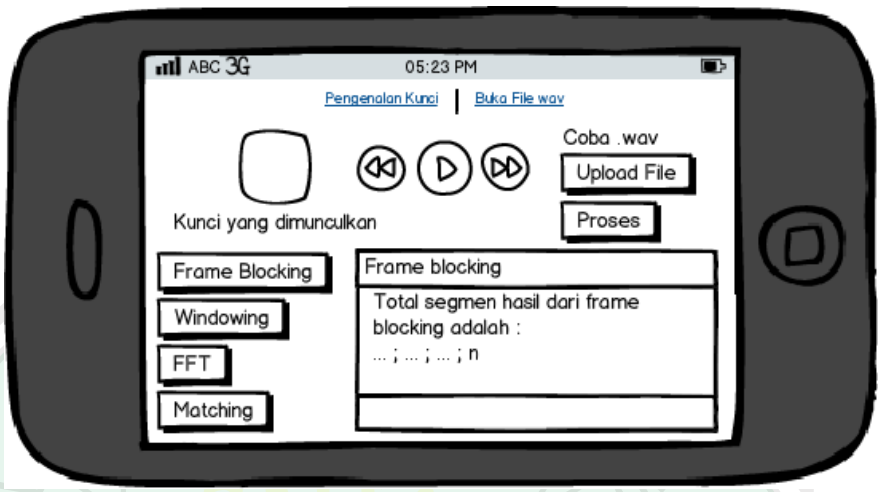
1.4.3.2 Tampilan Buka File Wav



Gambar 3.11 Desain Interface Buka File Wav

Menu Buka File wav, merupakan proses nada dari file wave untuk dijadikan dan ditampilkan dalam bentuk kunci piano. Tombol *Upload File* digunakan untuk membuka file nada *.wav* yang akan diproses. Jadi, saat tombol *Proses* dijalankan membutuhkan waktu beberapa saat untuk mengenali kunci nada yang ada pada file tersebut. Saat proses tersebut selesai maka akan dimunculkan kunci nada piano.

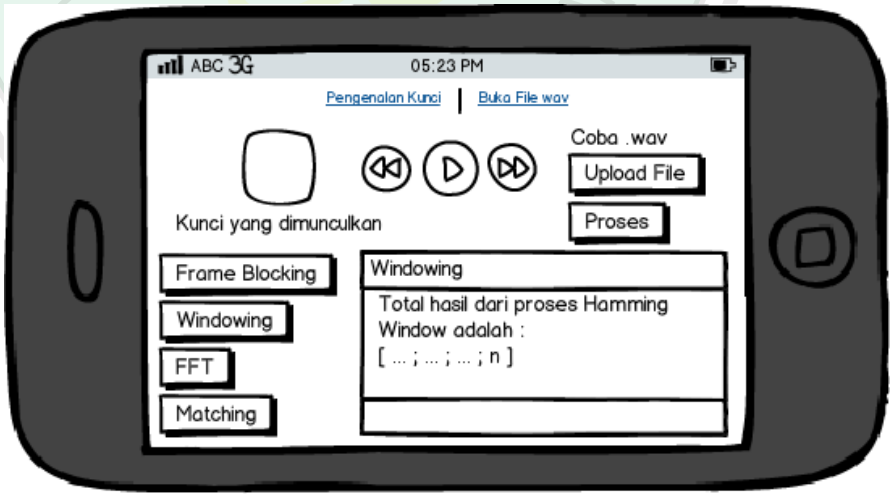
1.4.3.3 Tampilan Proses Frame Blocking



Gambar 3.12 Desain Interface Proses Frame Blocking

Pada proses *frame blocking* akan memunculkan nilaitiap frame berdasarkan nilai frekuensi masukan. Nilaimasukan *x*pada tiap frame akan diketahui sebanyak *n*.

1.4.3.4 Tampilan Proses Windowing

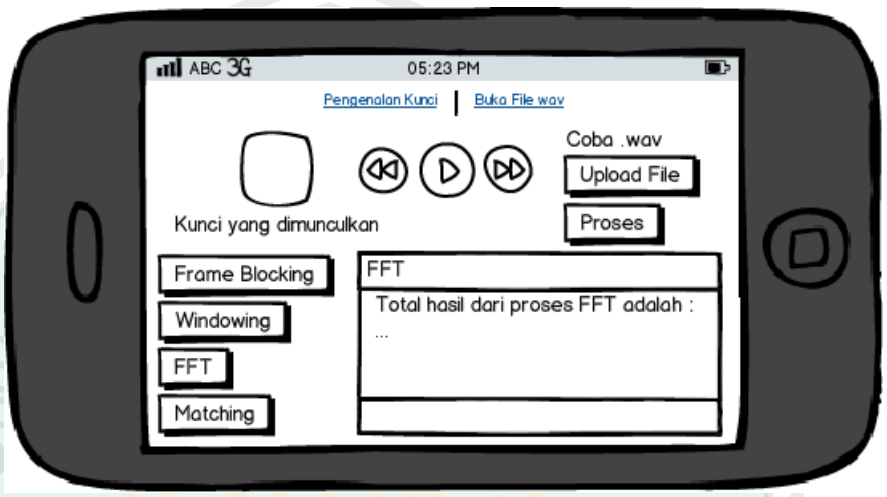


Gambar 3.13 Desain Interface Proses Windowing

Hasil dari proses frame blocking, nilai yang didapat akan diproses lagi menggunakan *Hamming Window* agar sinyal bisa

continue. Hasil nilai dari proses windowing adalah sebanyak nilai sinyal masukan *n*.

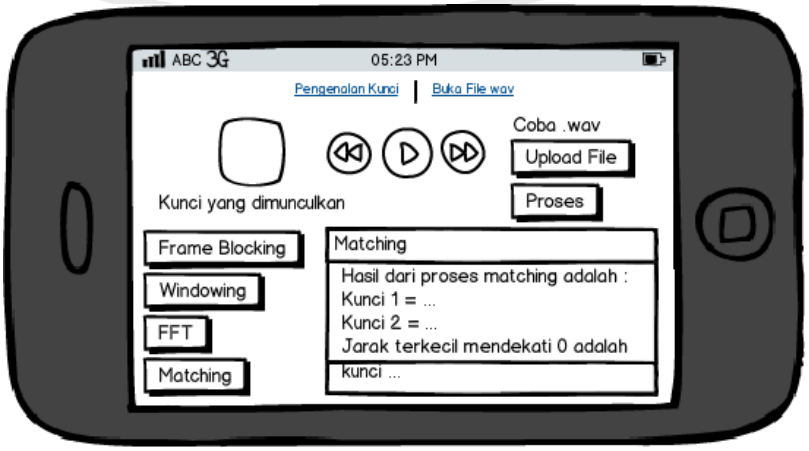
1.4.3.5 Tampilan Proses FFT



Gambar 3.14 Desain Interface Proses FFT

Hasil dari proses *windowing*masih berupa nilai spektrum suara dalam domain waktu, agar dapat dirubah menjadi sebuah nilai frekuensi maka dibutuhkan proses FFT. Hasil nilai dari proses FFT adalah sebanyak nilai sinyal masukan n.

1.4.3.6 Tampilan Proses Matching



Gambar 3.15 Desain Interface Proses Matching

Hasil dari proses FFT berupa frekuensi akan dicocokkan dengan nilai frekuensi standard dari nada piano. Perbandingan antara 2 kunci akan dihitung berdasarkan jarak terkecil dari nilai frekuensi standard piano.

UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM
PUSAT PERPUSTAKAAN

BAB IV

HASIL DAN PEMBAHASAN

1.1 Implementasi

Aplikasi ini dibuat dalam software aplikasi android dengan menggunakan java script dalam pembuatannya. Dalam aplikasi ini terdapat dua pengenalan nada piano, yaitu : pengenalan bentuk kunci mayor dan minor dari piano, yang kedua adalah membuka file kunci nada piano yang berformat .wav kemudian diproses dan dikenali kunci nada piano yang masuk tersebut. Aplikasi ini memerlukan memori yang bersifat sementara karena jika user ingin memulai baru kembali, maka data inputan nada piano yang sebelumnya akan otomatis terhapus. Pembuatan aplikasi android ini menggunakan android tipe 4.2 dan dengan menggunakan software eclipse agar pembuatannya dapat dilakukan di pc atau laptop.

1.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Untuk menjalankan aplikasi ini di butuhkan beberapa komponen *hardware*, antara lain :

1. Laptop Toshiba L740, dengan spesifikasi sebagai berikut :
    - Prosessor Intel Core i3 2,53 Ghz with Graphics HD3000-729MB
    - RAM 2 GB DDR3
    - 14.0” HD LED LCD
    - Hardisk 320 GB
    - Sistem Operasi Windows 7

- Speaker aktif
- Mouse dan keyboard aktif

2. Smartphone berbasis android (Asus Zenfone 5) dengan spesifikasi sebagai berikut :
   - Prosessor Dual Core 2 GHz Intel Atom Z2580
   - RAM 2 GB
   - Sistem Operasi Android v4.4.2 (KitKat)

1.1.2 Spesifikasi Perangkat Lunak (Software)

Untuk membuat aplikasi ini di butuhkan beberapa komponen *software*, antara lain :

1. Windows 7, untuk sistem operasi
2. Eclipse + Android Developer Tools, untuk membangun aplikasi
3. Software Cool Edit Pro, untuk merekam nada piano
4. Photoshop, untuk mendesain tampilan

4.2 Bentuk Pola Kunci Piano

Suatu bentuk kunci nada piano ditampilkan dalam bentuk simulasi *tuts* piano agar pengguna dapat menerapkannya langsung pada piano yang asli. Pada pengenalan kunci piano tersebut pengguna dapat memilih kunci yang akan dikenali. Hasil dari bentuk pengenalan nada tersebut adalah sebagai berikut :

Gambar 4.1 Hasil Tampilan Pengenalan Kunci

Pada tampilan tersebut terdapat simulasi bentuk kunci C mayor menggunakan 6 *finger*. Pada aplikasi terdapat 12 bentuk kunci mayor dan 12 bentuk kunci minor. Dengan adanya simulasi bentuk kunci tersebut diharapkan pengguna dapat mempraktekkan langsung ke dalam piano asli.

4.3 Hasil Data Latih

Sebelum pengujian sistem, dilakukan pemrosesan data latih untuk mengetahui ciri standar dari sebuah kunci nada piano. Data latih ini yang akan dijadikan acuan atau pencocokan dari data uji. Pada dasarnya tahapan yang dilakukan pada proses pengambilan sampel adalah dengan melakukan perekaman sebanyak 24 kunci mayor dan 24 kunci minor. Dalam proses perekaman menggunakan *microphone* yang berkualitas dan menggunakan *software Cool Edit Pro 2* serta dilakukan pada ruangan yang kedap suara.

4.3.1 Kunci Mayor

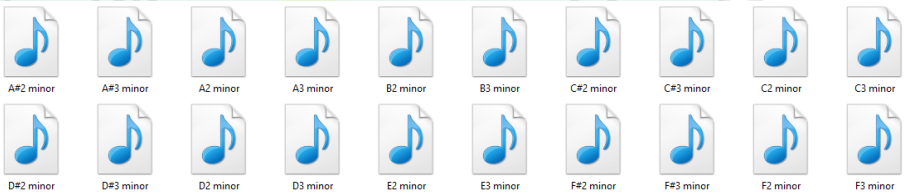
Dari proses perekaman kunci nada piano yang telah dilakukan maka diperoleh hasil sampel kunci nada berupa file audio berformat wav.

Gambar 4.2 Sampel Kunci Nada Piano Mayor

Jumlah kunci nada mayor yang direkam sebanyak 24 sampel dari 3 oktaf nada, dimana C = do.

4.3.2 Kunci Minor

Hasil perekaman kunci nada piano disini merupakan hasil kunci nada minor, adapun hasilnya adalah sebagai berikut :



Gambar 4.3 Sampel Kunci Nada Piano Minor

Jumlah kunci nada minor yang direkam sebanyak 24 sampel dari 3 oktaf nada, dimana C = do.

4.3.3 Data Latih Frame Blocking

Rumus perhitungan frame blocking ditunjukkan pada *source code* berikut ini :

```
short[] amplitudes=wave.getSampleAmplitudes();
int numSamples = amplitudes.length;

int pointer=0;
if (overlapFactor>1){
    int numOverlappedSamples=numSamples*overlapFactor;
    int backSamples=fftSampleSize*(overlapFactor-1)/overlapFactor;
    int fftSampleSize_1=fftSampleSize-1;
    short[] overlapAmp= new short[numOverlappedSamples];
    pointer=0;
    for (int i=0; i<amplitudes.length; i++){
        overlapAmp[pointer++]=amplitudes[i];
        if (pointer%fftSampleSize==fftSampleSize_1){
            i-=backSamples;
        }
    }
    numSamples=numOverlappedSamples;
    amplitudes=overlapAmp;
}

numFrames=numSamples/fftSampleSize;
framesPerSecond=(int)(numFrames/wave.length());
```

Gambar 4.4 Kode program frame blocking

Hasil dari proses perhitungan diatas adalah sebagai berikut. Sebagai contoh hasil dari perhitungan A_3 mayor :

```
Hasil Frame Blocking :
Jml Sample =529200
Jml Data per Frame = 1024
Jml Frame = 516
Hasil Frame Blocking  =
X[0] = 12.0, X[1] = 23.0, X[2] = 22.0, X[3] = 28.0, X[4] = 67.0, X[5] =
45.0, X[6] = 24.0, X[7] = 24.0, X[8] = 39.0, X[9] = 33.0, X[10] = 42.0,
X[11] = 29.0, X[12] = 31.0, X[13] = 39.0, X[14] = 62.0, X[15] = 56.0,
```

Gambar 4.5 Hasil frame blocking

4.3.4 Data Latih Windowing

Rumus perhitungan windowing (*Hamming Window*) ditunjukkan pada *source code* berikut ini :

```
r = pi / m;
for (int n = -m; n < m; n++){
    w[m + n] = 0.54f + 0.46f * Math.cos(n * r);
    System.out.println("w["+m+" + "+n+"] = 0.54f + 0.46f * Math.cos("+n+" * "+r+");");
}
break;
```

Gambar 4.6 Kode program windowing

Hasil dari perhitungan *source code* windowing diatas (A_3 mayor) adalah sebagai berikut :

```
Hasil Windowing - Hamming :
Jml window =529200
Window  =
W[0] = 0.08000001311302185, W[1] = 0.08000867248318244,
W[2] = 0.08003465026764373, W[3] = 0.08007794548835684,
W[4] = 0.08013855651528112, W[5] = 0.08021648106644585,
W[6] = 0.08031171620803551, W[7] = 0.08042425835450101,
W[8] = 0.08055410326869406, W[9] = 0.080701246062027,
```

Gambar 4.7 Hasil windowing

4.3.5 Data Latih FFT

Rumus perhitungan FFT ditunjukkan pada *source code* berikut ini :

```
int imax = (int) (Math.log(fftFrameSize) / Math.log(2.));

double[] warray = new double[(fftFrameSize - 1) * 4];
int w_index = 0;

for (int i = 0,  nstep = 2; i < imax; i++) {
    int jmax = nstep;
    nstep <<= 1;

    double wr = 1.0;
    double wi = 0.0;

    double arg = Math.PI / (jmax >> 1);
    double wfr = Math.cos(arg);
    double wfi = sign * Math.sin(arg);

    for (int j = 0; j < jmax; j += 2) {
        warray[w_index++] = wr;
        warray[w_index++] = wi;

        double tempr = wr;
        wr = tempr * wfr - wi * wfi;
        wi = tempr * wfi + wi * wfr;
    }
}
```

Gambar 4.8 Kode program FFT

Hasil dari perhitungan *source code* FFT diatas (A_3 mayor) adalah sebagai berikut :

```
Hasil FFT :
Y[0] = 172.265625, Y[1] = 86.1328125, Y[2] = 0.0, Y[3] = 0.0, Y[4]
= 172.265625, Y[5] = 258.3984375, Y[6] = 258.3984375, Y[7] =
172.265625, Y[8] = 172.265625, Y[9] = 0.0, Y[10] = 86.1328125,
Y[11] = 172.265625, Y[12] = 172.265625, Y[13] = 86.1328125,
Y[14] = 86.1328125, Y[15] = 172.265625, Y[16] = 172.265625,
Y[17] = 86.1328125, Y[18] = 86.1328125, Y[19] = 172.265625,
Y[20] = 0.0, Y[21] = 86.1328125, Y[22] = 172.265625, Y[23] =
```

Gambar 4.9 Hasil FFT

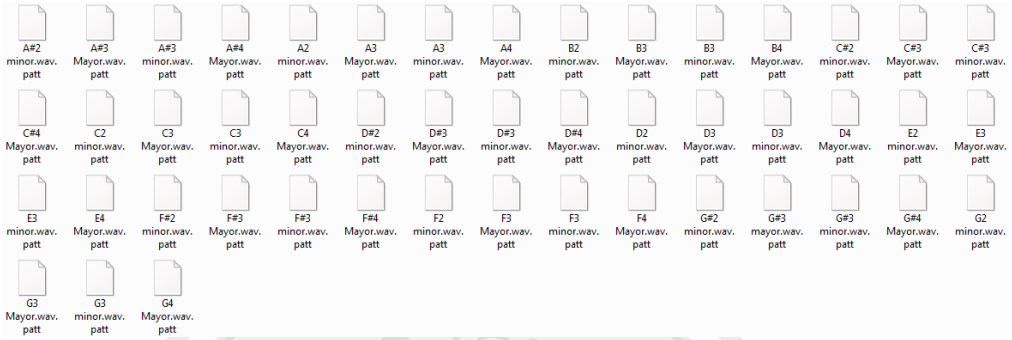
4.4 Fitur Database

Fitur database merupakan acuan untuk proses pencocokan nada. Hasil matriks array dari proses FFT disimpan dalam bentuk file *pattern*. Seperti ditunjukkan dalam *source code* berikut ini :

```
new File(Environment.getExternalStorageDirectory().toString()
        + "/pattern/").mkdir();

AssetManager assetManager = getAssets();
String[] files;
try {
    files = assetManager.list("data");

    List<String> it = Arrays.asList(files);

    for (int i = 0; i < it.size(); i++) {
        InputStream in = null;
        OutputStream out = null;
        try {
            in = assetManager.open("data/" + it.get(i).toString());
            out = new FileOutputStream(Environment
                    .getExternalStorageDirectory().toString()
                    + File.separator
                    + "pattern"
                    + File.separator
                    + it.get(i).toString());
            copyFile(in, out);
            in.close();
            in = null;
            out.flush();
            out.close();
            out = null;
        } catch (Exception e) {
            Log.e("tag", "Failed to copy asset file: "
                    + it.get(i).toString(), e);
        }
```

Gambar 4.10 Kode program file pattern

Data latih dari nada piano disimpan dalam file *pattern* untuk memudahkan pencocokan nada dari data uji. Berikut ini hasil file *pattern* data latih nada piano :



Gambar 4.11 Hasil file pattern

4.5 Hasil Data Uji

Setelah dilakukan proses data latih dan hasilnya disimpan dalam bentuk file pattern, selanjutnya akan dilakukan pengujian data. Data uji nada piano akan dicocokkan cirinya dengan data uji. Pada dasarnya tahapan yang dilakukan pada proses pengambilan sampel data uji sama dengan data latih yaitu dengan melakukan perekaman sebanyak 24 kunci mayor dan 24 kunci minor. Dalam proses perekaman menggunakan *microphone* yang berkualitas dan menggunakan *software Cool Edit Pro 2* serta dilakukan pada ruangan yang kedap suara.

4.5.1 Kunci Mayor

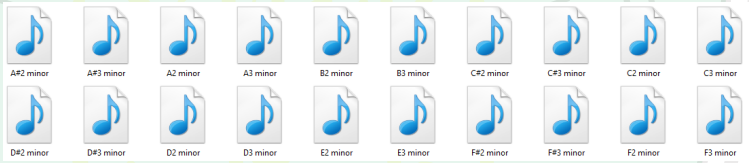
Dari proses perekaman kunci nada piano yang telah dilakukan maka diperoleh hasil sampel kunci nada berupa file audio berformat wav.

Gambar 4.12 Sampel Kunci Nada Piano Mayor

Jumlah kunci nada mayor yang direkam sebanyak 24 sampel dari 3 oktaf nada, dimana C = do.

4.5.2 Kunci Minor

Hasil perekaman kunci nada piano disini merupakan hasil kunci nada minor, adapun hasilnya adalah sebagai berikut :



Gambar 4.13 Sampel Kunci Nada Piano Minor

Jumlah kunci nada minor yang direkam sebanyak 24 sampel dari 3 oktaf nada, dimana C = do.

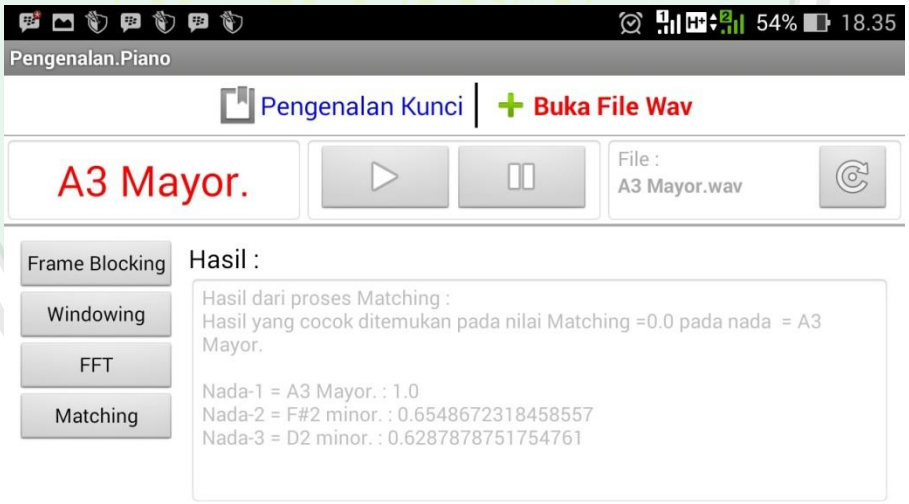
4.5.3 Hasil Pencocokan Data Uji Dengan Data Latih

Pencocokan dua buah nada menggunakan rumus *Euclidean distance*. Adapun rumus dari ED ditunjukkan pada *souce code* berikut ini :

```
    public static double ecludeanDistance(double[] array1, double[] array2)
    {
        double Sum = 0.0;
        for(int i=0;i<array1.length;i++) {
           Sum = Sum + Math.pow((array1[i]-array2[i]),2.0);
        }
        return Math.sqrt(Sum);
    }

}
```

Gambar 4.14 Kode program euclidean distance

Kesesuaian antara dua buah nada dapat diketahui dari jarak yang mendekati nilai 0. Untuk mengetahui jarak data uji dengan data latih maka ditampilkan 3 nada dari data latih yang mendekati 0. Hasil dari pencocokan dua buah nada dari nada A_3 mayor adalah sebagai berikut :



Gambar 4.15 Tampilan hasil pencocokan nada

Hasil *Euclidean Distance* atau perhitungan jarak antara data latih dengan data uji akan ditunjukkan pada tabel berikut ini :

No	Sampel Kunci	Hasil Perhitungan Jarak ED
1	C_2 Minor	G_3 Mayor = 0.0 C_2 Minor = 0.0 E_2 Minor = 0.105
2	$C\#_2$ Minor	$C\#_2$ Minor = 0.0 $G\#_3$ Minor = 0.436 E_2 Minor = 0.699
3	D_2 Minor	D_2 Minor = 0.0 $F\#_2$ Minor = 0.283 A_3 Mayor = 0.371
4	$D\#_2$ Minor	$D\#_2$ Minor = 0.0 $D\#_3$ Mayor = 0.598 $A\#_3$ Mayor= 0.625
5	E_2 Minor	E_2 Minor = 0.0 C_2 Minor = 0.105 G_3 Mayor = 0.162
6	F_2 Minor	F_2 Minor = 0.0 F_3 Mayor = 0.0 C_4 Mayor = 0.025
7	$F\#_2$ Minor	$F\#_2$ Minor = 0.0 D_2 Minor = 0.283 A_3 Mayor = 0.345

No	Sampel Kunci	Hasil Perhitungan Jarak ED
8	G_2 Minor	G_2 Minor = 0.0 G_3 Mayor = 0.258 C_2 Minor = 0.267
9	$G\#_2$ Minor	$D\#_3$ Minor = 0.0 $G\#_2$ Minor = 0.0 E_3 Minor = 0.0
10	A_2 Minor	A_2 Minor = 0.0 F_3 Mayor = 0.405 F_3 Minor = 0.464
11	$A\#_2$ Minor	$A\#_2$ Minor = 0.0 F_4 Mayor = 0.166 $A\#_3$ Minor = 0.166
12	B_2 Minor	B_2 Minor = 0.0 D_4 Mayor = 0.0 D_3 Minor = 0.0
13	C_3 Mayor	C_3 Mayor = 0.0 C_4 Mayor = 0.019 F_2 Minor = 0.197
14	C_3 Minor	$D\#_3$ Minor = 0.0 $G\#_2$ Minor = 0.0 C_3 Minor = 0.0

No	Sampel Kunci	Hasil Perhitungan Jarak ED
15	C#3 Mayor	C#3 Mayor = 0.0 A#3 Minor = 0.313 A#2 Minor = 0.5
16	C#3 Minor	C#3 Minor = 0.0 F3 Minor = 0.167 A#2 Minor = 0.312
17	D3 Mayor	D3 Minor = 0.0 D3 Mayor = 0.0 B3 Minor = 0.12
18	D3 Minor	B2 Minor = 0.0 D3 Minor = 0.0 D3 Mayor = 0.0
19	D#3 Mayor	G#3 Minor = 0.0 D#3 Mayor = 0.0 D#4 Mayor = 0.278
20	D#3 Minor	D#3 Minor = 0.0 G#2 Minor = 0.0 E4 Mayor = 0.0
21	E3 Mayor	G#3 Minor = 0.0 E3 Mayor = 0.0 E3 Minor = 0.092

No	Sampel Kunci	Hasil Perhitungan Jarak ED
22	E_3 Minor	$G\#_2$ Minor = 0.0 E_3 Minor = 0.0 $G\#_3$ Minor = 0.057
23	F_3 Mayor	F_2 Minor = 0.0 F_3 Mayor = 0.0 C_4 Mayor = 0.177
24	F_3 Minor	F_3 Minor = 0.0 $C\#_3$ Minor = 0.167 A_3 Minor = 0.195
25	$F\#_3$ Mayor	$F\#_3$ Mayor = 0.0 $F\#_4$ Mayor = 0.076 $F\#_2$ Minor = 0.416
26	$F\#_3$ Minor	A_4 Mayor = 0.0 $F\#_3$ Minor = 0.0 A_3 Minor = 0.134
27	G_3 Mayor	G_3 Mayor = 0.0 C_2 Minor = 0.0 E_2 Minor = 0.162
28	G_3 Minor	G_3 Minor = 0.0 G_4 Mayor = 0.25 B_3 Minor = 0.427

No	Sampel Kunci	Hasil Perhitungan Jarak ED
29	G#3 Mayor	G#3 Mayor = 0.0 D#4 Mayor = 0.149 G#3 Minor = 0.391
30	G#3 Minor	G#3 Minor = 0.0 B3 Minor = 0.0 D#3 Mayor = 0.0
31	A3 Mayor	A3 Mayor = 0.0 F#2 Minor = 0.345 D2 Minor = 0.371
32	A3 Minor	A4 Mayor = 0.0 A3 Minor = 0.0 F#3 Minor = 0.134
33	A#3 Mayor	A#3 Mayor = 0.0 A#3 Minor = 0.042 F4 Mayor = 0.192
34	A#3 Minor	A#3 Minor = 0.042 A#3 Mayor = 0.042 F4 Mayor = 0.052
35	B3 Mayor	B3 Mayor = 0.0 E3 Mayor = 0.169 B4 Mayor = 0.587

No	Sampel Kunci	Hasil Perhitungan Jarak ED
36	B_3 Minor	$G\#_3$ Minor = 0.0 B_3 Minor = 0.0 E_3 Minor = 0.111
37	C_4 Mayor	C_4 Mayor = 0.0 C_3 Mayor = 0.019 F_2 Minor = 0.026
38	$C\#_4$ Mayor	$C\#_4$ Mayor = 0.0 $A\#_2$ Minor = 0.392 $A\#_3$ Minor = 0.417
39	D_4 Mayor	B_2 Minor = 0.0 D_4 Mayor = 0.0 D_3 Minor = 0.17
40	$D\#_4$ Mayor	$D\#_4$ Mayor = 0.0 $G\#_3$ Mayor = 0.149 $D\#_3$ Minor = 0.162
41	E_4 Mayor	$D\#_3$ Minor = 0.0 E_4 Mayor = 0.0 $G\#_2$ Minor = 0.019
42	F_4 Mayor	F_4 Mayor = 0.0 $A\#_3$ Minor = 0.052 $A\#_2$ Minor = 0.167

No	Sampel Kunci	Hasil Perhitungan Jarak ED
43	F#4 Mayor	F#4 Mayor = 0.0 F#3 Mayor = 0.076 F#3 Minor = 0.432
44	G4 Mayor	G4 Mayor = 0.0 G3 Minor = 0.25 E3 Minor = 0.345
45	G#4 Mayor	G#4 Mayor = 0.0 G#3 Minor = 0.644 F3 Minor = 0.655
46	A4 Mayor	A4 Mayor = 0.0 A3 Minor = 0.0 F#3 Minor = 0.0
47	A#4 Mayor	A#4 Mayor = 0.0 A#3 Mayor = 0.317 D#4 Mayor = 0.46
48	B4 Mayor	B4 Mayor = 0.0 G#2 Minor = 0.583 B3 Mayor = 0.587

Tabel 4.1 Hasil Perhitungan Jarak Data Latih dan Data Uji

Hasil sampel data uji disini terdapat 24 sampel data mayor dan 24 sampel data minor, jadi total sampel adalah 48 sampel data uji kunci

piano. Hasil kecocokan data uji dan data latih akan ditampilkan pada tabel berikut ini :

No.	Sampel Kunci	Hasil Kecocokan	Hasil Kesesuaian
1	C2 Minor	G3 Mayor	Tidak Sesuai
2	C#2 Minor	C#2 Minor	Sesuai
3	D2 Minor	D2 Minor	Sesuai
4	D#2 Minor	D#2 Minor	Sesuai
5	E2 Minor	E2 Minor	Sesuai
6	F2 Mnor	F2 Minor	Sesuai
7	F#2 Minor	F#2 Minor	Sesuai
8	G2 Minor	G2 Minor	Sesuai
9	G#2 Minor	D#3 Minor	Tidak Sesuai
10	A2 Minor	A2 Minor	Sesuai
11	A#2 Minor	A#2 Minor	Sesuai
12	B2 Minor	B2 Minor	Sesuai
13	C3 Mayor	C3 Mayor	Sesuai
14	C3 Minor	D#3 Minor	Tidak Sesuai
15	C#3 Mayor	C#3 Mayor	Sesuai
16	C#3 Minor	C#3 Minor	Sesuai
17	D3 Mayor	D3 Minor	Tidak Sesuai
18	D3 Minor	B2 Minor	Tidak Sesuai
19	D#3 Mayor	G#3 Minor	Tidak Sesuai

No.	Sampel Kunci	Hasil Kecocokan	Hasil Kesesuaian
20	D#3 Minor	D#3 Minor	Sesuai
21	E3 Mayor	G#3 Minor	Tidak Sesuai
22	E3 Minor	G#2 Minor	Tidak Sesuai
23	F3 Mayor	F2 Minor	Tidak Sesuai
24	F3 Minor	F3 Minor	Sesuai
25	F#3 Mayor	F#3 Mayor	Sesuai
26	F#3 Minor	A4 Mayor	Tidak Sesuai
27	G3 Mayor	G3 Mayor	Sesuai
28	G3 Minor	G3 Minor	Sesuai
30	G#3 Minor	G#3 Minor	Sesuai
31	A3 Mayor	A3 Mayor	Sesuai
32	A3 Minor	A4 Mayor	Tidak Sesuai
33	A#3 Mayor	A#3 Mayor	Sesuai
34	A#3 Minor	A#3 Minor	Sesuai
35	B3 Mayor	B3 Mayor	Sesuai
36	B3 Minor	G#3 Minor	Tidak Sesuai
37	C4 Mayor	C4 Mayor	Sesuai
38	C#4 Mayor	C#4 Mayor	Sesuai
39	D4 Mayor	B2 Minor	Tidak Sesuai
40	D#4 Mayor	D#4 Mayor	Sesuai
41	E4 Mayor	D#3 Minor	Tidak Sesuai

No.	Sampel Kunci	Hasil Kecocokan	Hasil Kesesuaian
42	F_4 Mayor	F_4 Mayor	Sesuai
43	$F\#_4$ Mayor	$F\#_4$ Mayor	Sesuai
44	G_4 Mayor	G_4 Mayor	Sesuai
45	$G\#_4$ Mayor	$G\#_4$ Mayor	Sesuai
46	A_4 Mayor	A_4 Mayor	Sesuai
47	$A\#_4$ Mayor	$A\#_4$ Mayor	Sesuai
48	B_4 Mayor	B_4 Mayor	Sesuai

Tabel 4.2 Hasil Kesesuaian Data Uji

Dari tabel diatas, didapatkan hasil data yang sesuai sebanyak 34 kunci nada dan 14 kunci nada yang tidak sesuai. Dari hasil tersebut akan dihitung akurasi dan error dari sistem yang dijalankan. Akurasi adalah ukuran ketepatan sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang sesuai. Secara matematis, rumus yang dapat dituliskan adalah sebagai berikut :

$$Akurasi = \frac{\text{Jumlah data sesuai}}{\text{Jumlah data seluruhnya}} \text{ x } 100\%$$

Dari persamaan diatas, maka didapatkan nilai akurasi sebagai berikut:

$$Akurasi = \frac{34}{48} \text{ x } 100\%$$

$$Akurasi = 70{,}83\ \%$$

Error adalah tingkat kesalahan sistem dalam hal mengenali masukan yang diberikan pada sistem terhadap jumlah data secara

keseluruhan. Secara matematis, rumus yang dapat dituliskan adalah sebagai berikut :

$$Error = \frac{\text{Jumlah data tidak sesuai}}{\text{Jumlah data seluruhnya}} \text{ x } 100\%$$

Dari persamaan diatas, maka didapatkan nilai *error* sebagai berikut:

$$Error = \frac{14}{48} \text{ x } 100\%$$

$$Error = 29{,}16\ \%$$

4.6 Integrasi Dalam Islam

Dalam penelitian tentang pengenalan nada piano terdapat beberapa integrasi dalam konteks Islami. Sumber korelasi yang disebutkan disini adalah dari Al-Qur'an dan Al-Hadits. Sebagai contoh Firman Allah SWT :

أَلَمْ نَجْعَل لَّهُۥ عَيْنَيْنِ ۝٨ وَلِسَانًا وَشَفَتَيْنِ ۝٩

Artinya : "*Bukankah kami telah berikan kepadamu dua buah mata (untuk melihat), lidah dan dua buah bibir (untuk bersuara, mengecap makanan dan minuman).*" (Q.S. Al – Balad : 8-9). (RI 1996).

Apablia dilihat dari ayat diatas maka seluruh perbuatan seperti melihat, mendengar, bersuara dan menggunakan tangan adalah mubah. Ayat diatas juga mengajarkan bahwa manusia boleh melihat dan mendengar apa saja. Manusia boleh melihat pemandangan, mendengarkan suara wanita dan nyanyiannya atau bermain musik kecuali terhadap apa – apa yang dilarang syara' dalam perbuatan itu.

Dalam mendengarkan atau bermain suatu alat musik, penulis sangat menyarankan agar diiringi dengan suara – suara yang mengandung kata – kata yang baik. Sebagai contoh kata – kata yang baik adalah dengan melantunkan shalawat, tartil, mengagungkan nama Allah SWT, bersyukur dengan ciptaan-Nya, dll. Hal tersebut untuk menghindari diri dari kemungkaran dan membuat kita lalai akan keagungan Allah SWT. Sebagaimana firman Allah SWT :

إِنَّمَا يَأۡمُرُكُم بِٱلسُّوٓءِ وَٱلۡفَحۡشَآءِ وَأَن تَقُولُواْ عَلَى ٱللَّهِ مَا لَا تَعۡلَمُونَ ١٦٩

Artinya : "*Sesungguhnya setan akan menyuruh kepada perbuatan jelek dan keji dan setan akan menyuruh kepada kalian untuk mengatakan kepada Allah SWT apa – apa yang kalian tidak ketahui.*" (Q.S. Al – Baqarah : 169). (RI 1996).

Dalam rangka aktualisasi terhadap keindahan itu, Al-Qur'an telah memberikan ruang gerak dan legitimasi faktual yang memungkinkan seni terus hidup dalam kedinamisannya lewat medium apapun keindahan itu diekspresikan. Imam Al – Ghozali dalam kitabnya *Ihya Ullumuddin* yang berkaitan dengan seni suara atau musik diantaranya :

وَٱقۡصِدۡ فِي مَشۡيِكَ وَٱغۡضُضۡ مِن صَوۡتِكَۚ إِنَّ أَنكَرَ ٱلۡأَصۡوَٰتِ لَصَوۡتُ ٱلۡحَمِيرِ ١٩

Artinya : "*Sederhanalah kamu dalam berjalan dan lunakanlah suaramu, sesungguhnya seburuk – buruk suara adalah bunyi keledai.*" (Q.S. Al – Luqman : 19). (RI 1996).

Imam Al – Ghozali mengambil pengertian ayat tersebut dari mahfum mukhalafah, Allah SWT memuji suara yang baik, dengan demikian dibolehkan mendengarkan nyanyian atau musik yang baik. (Ghazali 1100 M).

Dalam hal meliputi musik dan lagu, Imam Al – Ghazali dalam *Ihya Ulumuddin* menulis :

"*Siapa yang tidak berkesan hatinya di musim bunga dengan kembang – kembangnya atau oleh suara musik dan getaran nadanya maka fitrahnya telah mengidap parah yang sulit diobatinya.*" (Ghazali 1100 M).

Dari hadits diatas Imam Al – Ghozali menambahkan pada hukum ini beberapa syarat yang harus diperhatikan dalam mendengarkan musik dan nyanyian :

1. Bahwa tidak setiap nyanyian itu hukumnya mubah (boleh) pokok pembicaraannya harus sesuai dengan sopan santun Islam dan pengajarannya.
2. Cara penyajiannya juga peranan penting, isi syair boleh tidak haram dan tidak tercela tetapi cara biduan atau biduanita menyanyi dengan ungkapan yang tidak enak sengaja menimbulkan rangsangan dan mengalihkan dari wilayah yang dihalalkan ke wilayah yang diharamkan.
3. Nyanyian dan musik tidak boleh dibarengi dengan sesuatu yang diharamkan seperti minuman, bersolek serta bercampur aduk dan berkelakar tanpa batas antara pria dan wanita., karena inilah yang biasa terjadi di tempat – tempat pertunjukan musik sejak dahulu.
4. Nyanyian dan musik sebagaiman semua dibolehkan diisyaratkan tidak berlebih – lebihan dalam segala hal dan yang demikian itu mengindikasikan adanya kekosongan pikiran dan hati dari berbagai kewajiban besar dan cita – cita mulia.

Imam Bukhari dalam *Fath Al-Bari*, diceritakan Ruba'i binti Mu'awwidz bin Afra berkata, Nabi SAW mendatangi pesta perkawinanku, lalu beliau duduk diatas dipan seperti dudukmu denganku, lalu mulailah beberapa orang hamba perempuan kami memukul gendang dan mereka menyanyi dengan memuji orang yang mati syahid pada perang Badar. Tiba – tiba salah seorang di antara mereka berkata, "*Diantara kita ada Nabi SAW yang mengetahui apa yang akan terjadi kemudian.*" Maka Nabi SAW bersabda, "*Tinggalkan omongan itu. Teruskanlah apa yang kamu (nyanyikan) tadi.*" (HR. Bukhari 773 - 852 H).

Dari Aisyah r.a. dia pernah menikahkan seorang wanita kepada pemuda Anshar. Tiba – tiba Rasulullah SAW bersabda, "*Mengapa tidak kalian adakan permainan karena orang Anshar itu suka pada permainan.*" (HR. Bukhari 773 - 852 H).

Dari sumber – sumber Al – Quran maupun Al – Hadits diatas dapat disimpulkan bahwa terdapat adanya hubungan antara suara, nada maupun musik dalam konteks Islami. Hal tersebut masih diperbolehkan jika tidak digunakan dalam hal – hal yang dapat melalaikan diri. Penulis sangat mengharapkan agar dalam mendengarkan maupun mempelajari musik semata – mata untuk mensyukuri ciptaan-Nya.

BAB V

PENUTUP

5.1 Kesimpulan

Aplikasi pengenalan piano dirancang menggunakan metode FFT dengan beberapa tahapan, hinggga akhirnya muncullah nilai jarak kedua nada kunci. Terdapat beberapa tahapan proses pengolahan sinyal untuk menghasilkan sebuah kunci nada. Proses tersebut yaitu *Sampling*, *Frame Blocking*, *Windowing*, Fast Fourier Transform (FFT), *Euclidean Distance* (ED). Untuk memproses sinyal tersebut pertama- tama sinyal analog di rekam, kemudian masuk kedalam pada proses *Sampling* yang berfungsi untuk mengambil diskritnya saja. Pada proses *Frame blocking* berfungsi untuk memudahkan analisa sinyal. Selanjutnya proses *Windowing*, proses ini sinyal akan direduksi puncaknya untuk memperhalus *frame*. Kemudian masuk ke proses FFT, proses ini berfungsi untuk menghasilkan *domain* frekuensi yang hasilnya berupa sekumpulan nilai *array* yang disimpan dalam file *.patterrn* sebagai data acuan.

Pemrosesan sinyal dua data nada adalah dengan proses pencocokan nilai *array* yang terkandung di dalamnya. Sekumpulan kedua nilai *array* dihitung jaraknya menggunakan persamaan *Euclidean Distance*, akan muncul informasi kunci nada dari data masukan, jika jarak nilai mendekati 0 dari data acuan atau data latih. Setelah melakukan pengujian sebanyak 24 sampel didapatkan nilai akurasi 70,83% dan *error* 29,16%.

5.2 Saran

Pada aplikasi pengenalan piano berbasis android ini tentunya masih banyak kekurangan dalam segi tampilan, tahapan – tahapan dalam mengenali nada, tingkat keakuratan, dll. Oleh karena itu perlu dilakukan perbaikan dan pengembangan agar aplikasi pengenalan nada piano ini semakin baik. Adapaun saran – saran dari penulis untuk mengembangkan aplikasi ini adalah sebagai berikut :

1. Tampilan bentuk kunci piano bukan hanya mayor dan minor, tetapi ada bentuk augmented, diminished, mayor7, minor7, dll.
2. Menambahkan fitur pengenalan nada piano secara *real time*.
3. Menggunakan *noise remover* sebelum proses mengenali ciri dari sebuah nada, agar ketahanan sistem terhadap *noise* lebih tinggi.
4. Menambahkan beberapa tahapan – tahapan dalam mengenali ciri nada piano, sehingga menambah nilai akurasi yang diinginkan.

DAFTAR PUSTAKA

Akbar, Fandy. *Konversi Nada - Nada Akustik Menjadi Chord Menggunakan Pitch Class Profile.* Surabaya: Institut Teknologi Sepuluh Nopember, 2011.

Aley, Rie - Rie. *Siapa Pun Bisa Main Keyboard.* Yogyakarta: Bukubiru, 2010.

Aris Tjahyanto, Yoyon K Suprapto, Diah Puspito Wulandari. *Model Analysis-By-Synthesis Aplikasi Pembangkit Suara Gamelan Sintetik.* Surabaya: Institut Teknologi Sepuluh Nopember, 2011.

Binanto, Iwan. *Multimedia Digital - Dasar Teori dan Pengembangannya.* Yogyakarta: Andi, 2010.

Fajar, Galih Ahmad. *Pengenalan dan Analisis Kualitas Penalaan Nada Tunggal Piano Secara Real Time Menggunakan Metode JST-SOM.* Bandung: Fakultas Elektro dan Komunikasi, Institut Teknologi Telkom, 2011.

Imam Al - Ghazali. *Ihya Ulumuddin Juz II.* Kairo: Darul Ihya Al - Kutub Al - Arabiyah, 1100 M.

Hadyansyah, Irvan. *Aplikasi Ketepatan Nada Suara Manusia Menggunakan Metode FFT Berbasis Android.* Bandung: Jurusan Teknik Telekomunikasi, IT Telkom, 2010.

HR. Bukhari, Al -. *Fath Al - Bari Juz III.* Mesir, 773 - 852 H.

Ibnu Gunawan, Kartika Gunadi. *Pembuatan Perangkat Lunak Wave Manipulator Untuk Memanipulasi File Wav.* Surabaya: Jurusan Informatika, Fakultas Teknik Industri, Unversitas Kristen Petra, 2005.

Kurniawan, Arief. *Pembuatan Aplikasi Chord Definer Dengan Fuzzy Inference Rules.* Surabaya: UKP, 2006.

Lyana. *Aplikasi Sistem Penentu Akor Pada Audio Dengan Fast Fourier Transform dan Jaringan Syaraf Tiruan.* Bandung: Fakultas Elektro dan Komunikasi, Institut Teknologi Telkom, 2012.

Mulyadi, S.T. *Membuat Aplikasi Untuk Android.* Yogyakarta: Multimedia Center Publishing, 2010.

Peggy, Anastassja Tania. *Deteksi Lahu Pada Piano Berdasarkan Ekstraksi Ciri Mel Frequency Cepstral Coefficient Dengan Metode Klasifikasi Learning Vector Quantization dan Euclidean Distance.* Bandung: Fakultas Elektro dan Komunikasi, Institut Teknologi Telkom, 2012.

Rahmadi, Andri Nur. *Pengenalan Nada Pada File Audio Berinstrumen Piano Menggunakan Representasi Chroma Dan Hidden Markov Model.* Bandung: Departemen Teknik Informatika Institut Teknologi Telkom, 2008.

RI, Departemen Agama. *Al - Qur'an dan Terjemahan.* Semarang: Toha Putra, 1996.

Riyanto, Sugeng. *Algoritma Fast Fourier Transform (FFT) Decimation In Time (DIT) Dengan Resolusi 1/10 Hertz.* Yogyakarta: Lab. Riset Komputasi, Jurusan Pendidikan Fisika FMIPA UNIY, 2009.

Sugeng, Hr. *Dasar - Dasar Bermain Pianika & Organ.* Semarang: Media Wiyata, 1966.

Supriansyah. *Perancangan Sistem Pengenalan Nada Tunggal Keyboard Menggunakan Matlab Pada PC.* Bandung: Jurnal Teknik Komputer Unikom, 2013.

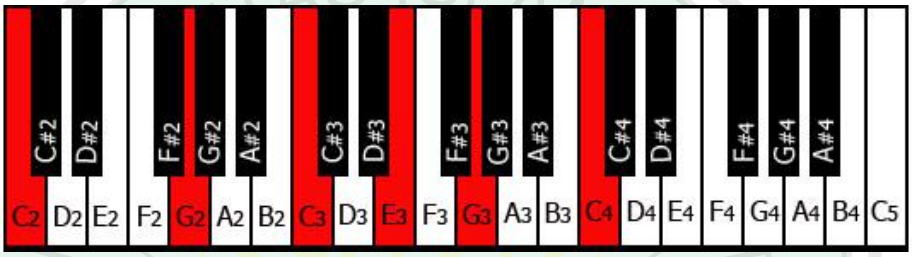
Supriyansyah. *Perancangan Sistem Pengenalan Nada Tunggal Keyboard (Orgen) Pada PC Berbasis Matlab*. Bandung: Jurusan Teknik Komputer, Unikom, 2013.

Syahbani, Nabilla Azzahra. *Pembuatan Transkrip Akord Instrumen Tunggal Menggunakan Metode Enhanced Pitch Class Profile*. Surabaya: Institut Teknologi Sepuluh Nopember (ITS), 2010.

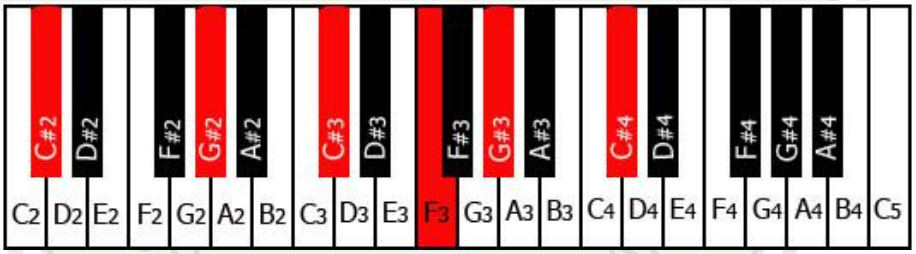
LAMPIRAN

Pada halaman lampiran disini ditampilkan bentuk – bentuk pola kunci mayor dan minor :

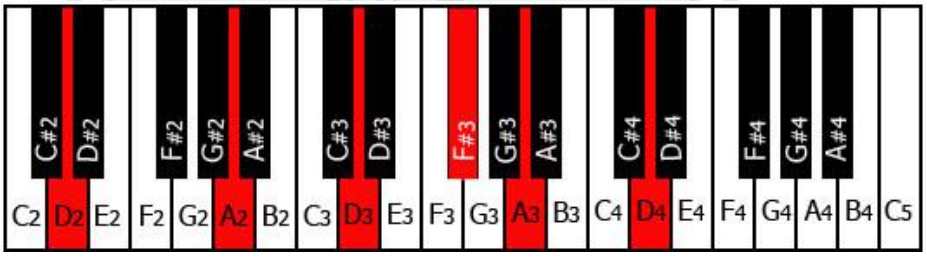
- **Kunci Mayor**
  - **C Mayor**



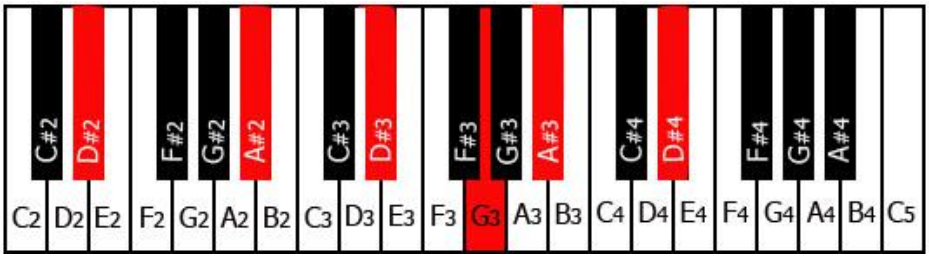
  - **C# Mayor**



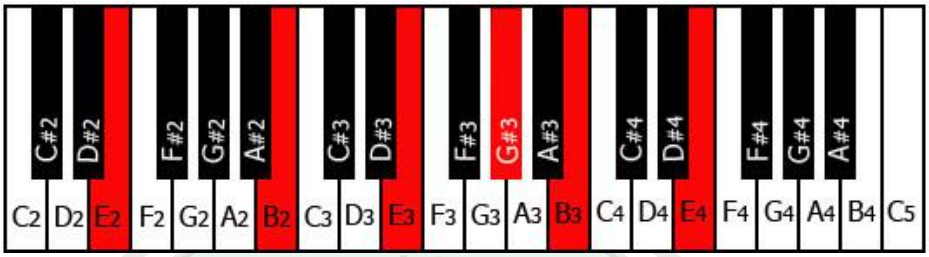
  - **D Mayor**



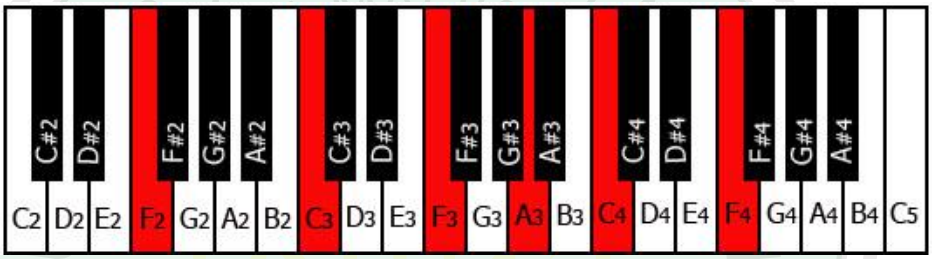
  - **D# Mayor**



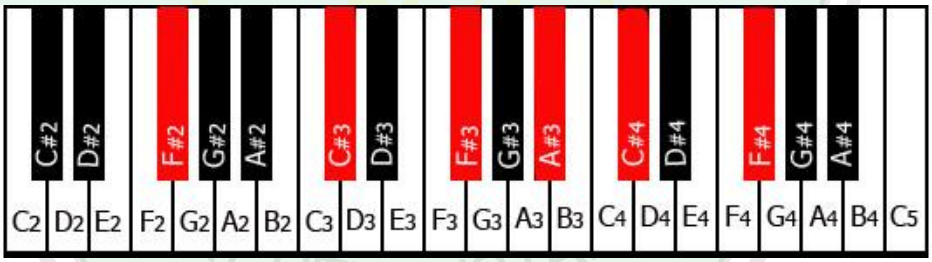
- **E Mayor**



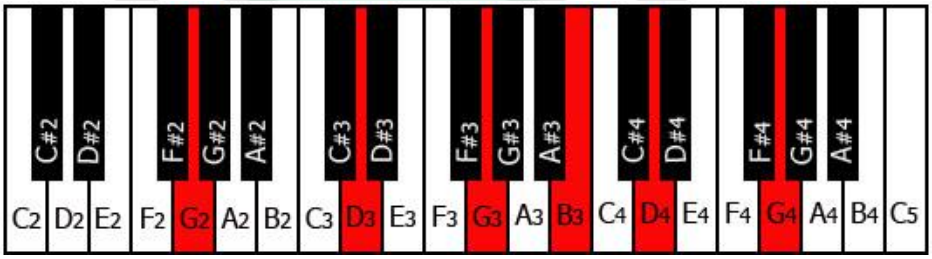
- **F Mayor**



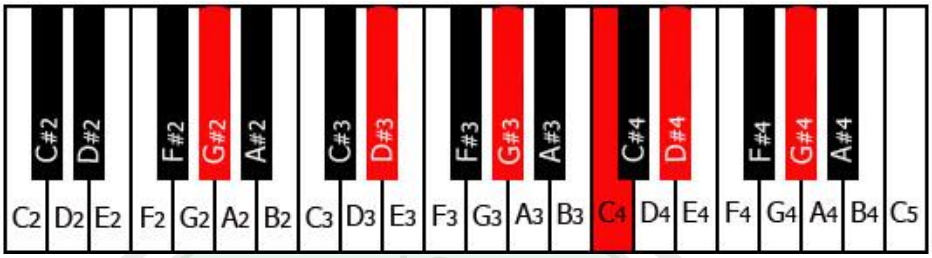
- **F# Mayor**



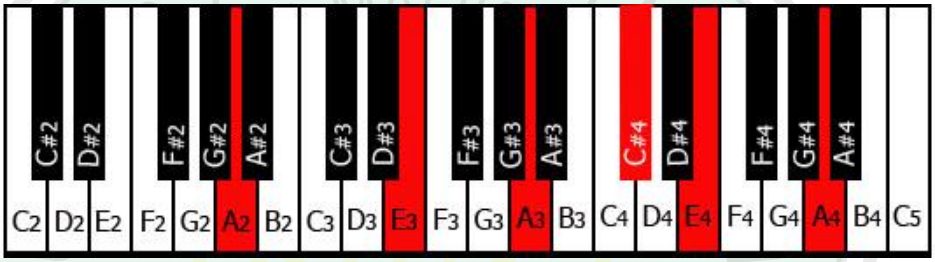
- **G Mayor**



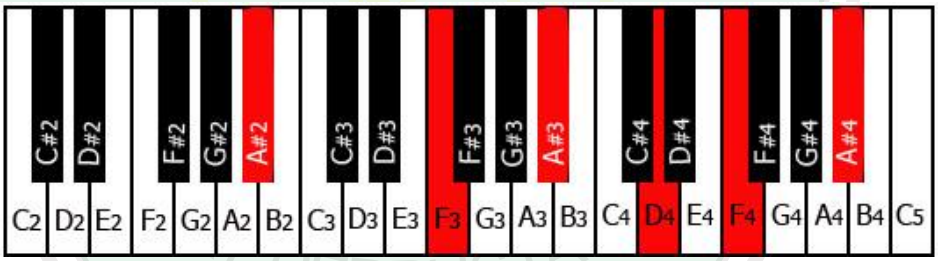
- **G# Mayor**



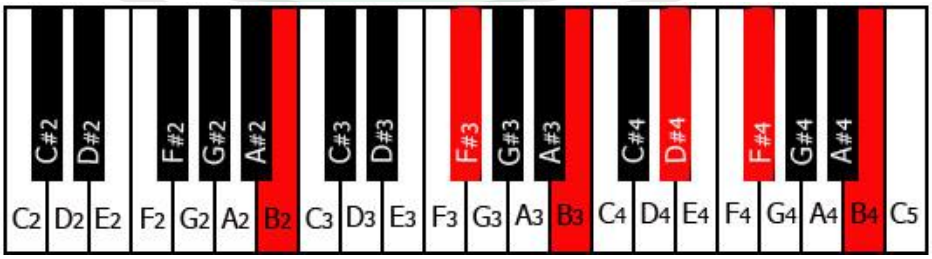
- **A Mayor**



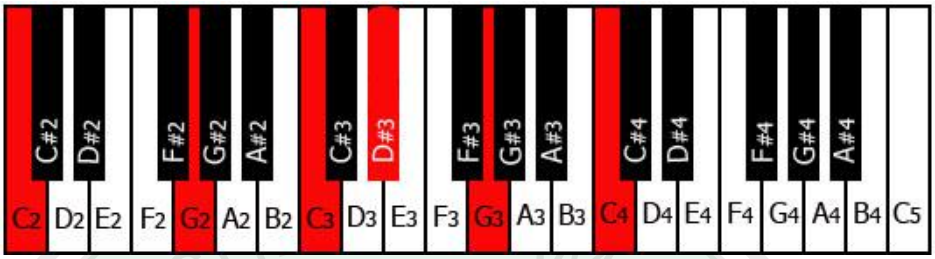
- **A# Mayor**



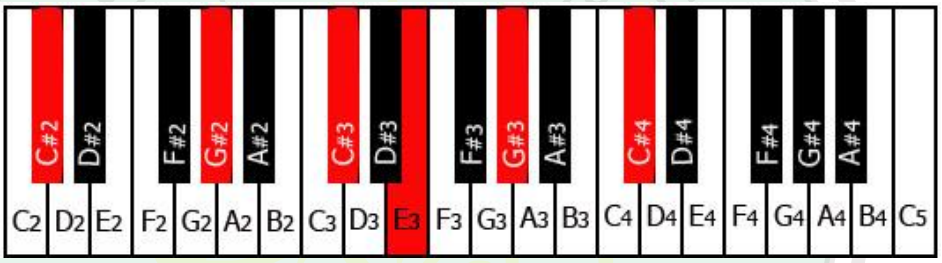
- **B Mayor**



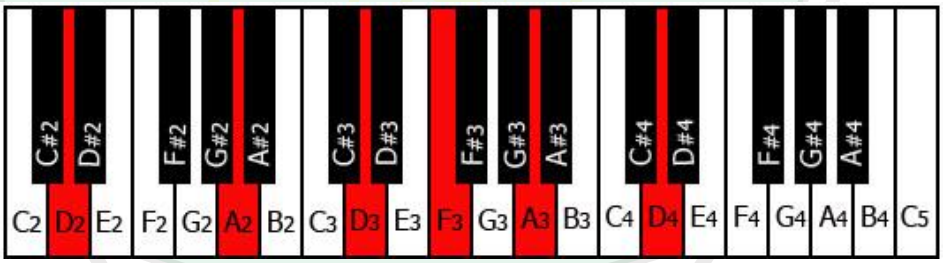
- **Kunci Minor**
  - **C Minor**



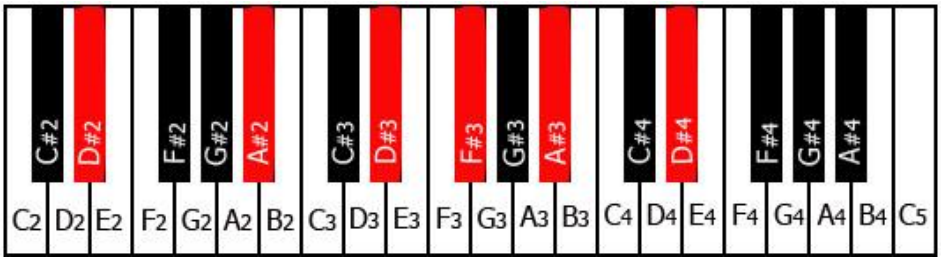
  - **C# Minor**



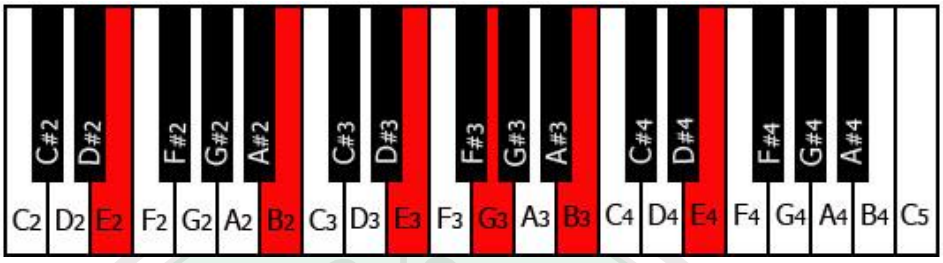
  - **D Minor**



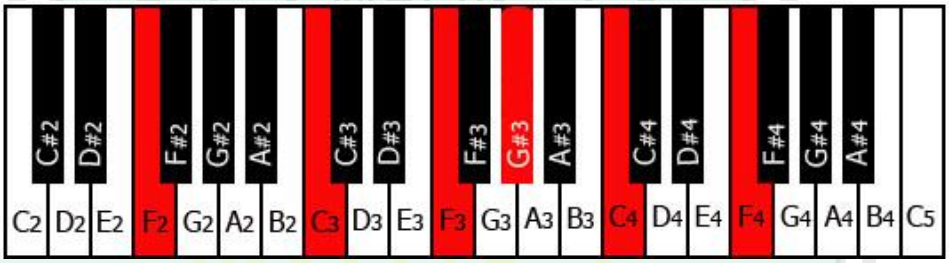
  - **D# Minor**



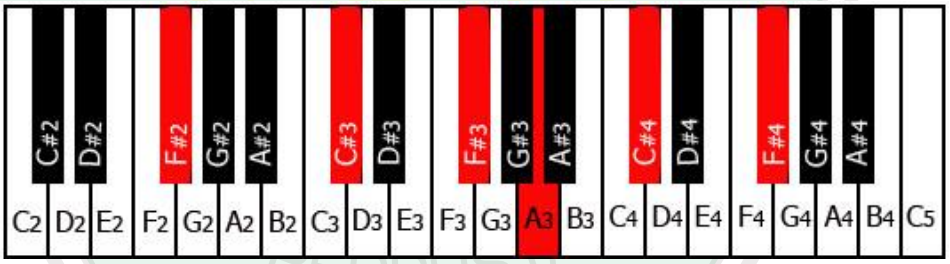
- **E Minor**



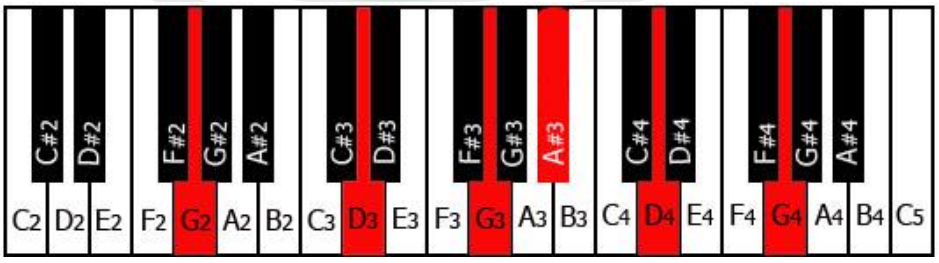
- **F Minor**



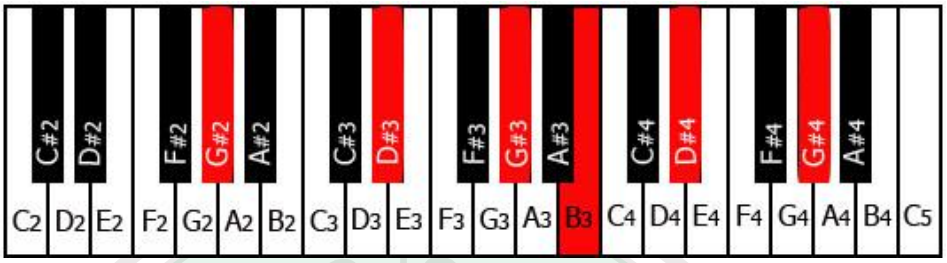
- **F# Minor**



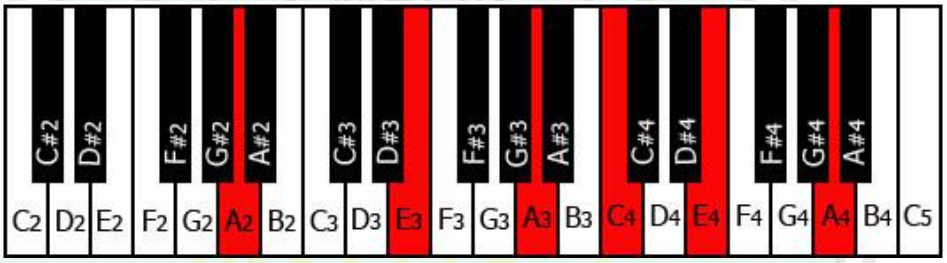
- **G Minor**



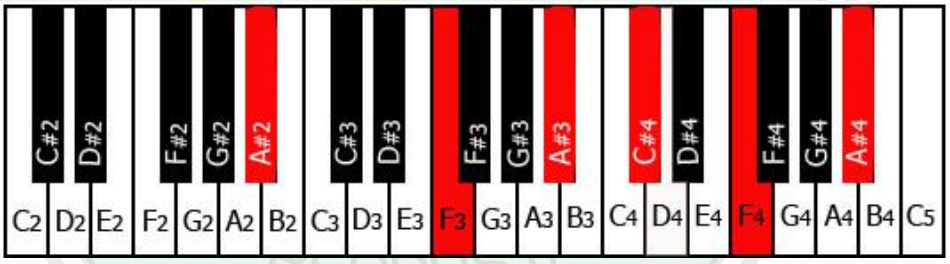
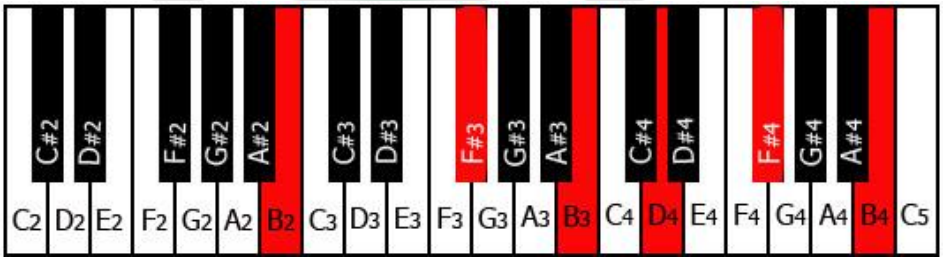
- **G# Minor**



- **A Minor**



- **A# Minor**



- **B Minor**