

RANCANG BANGUN APLIKASI SINKRONISASI NADA PADA TUNING GITAR DENGAN MENGGUNAKAN ALGORITMA FAST FOURIER TRANSFORM BERBASIS ANDROID

Aah Sumiah¹, Dede Try Lesmana²

^{1,2}Universitas Kuningan

Jl. Cut Nyak Dhien no.36A Kuningan

aah.sumiah@uniku.ac.id¹, dedetrylesmana@gmail.com²

Abstrak

Gitar merupakan salah satu alat musik yang sangat populer dan banyak digunakan di seluruh dunia. Untuk menghasilkan nada yang harmonis dan tepat, senar pada gitar memerlukan penalaan (tuning) yang baik. Sayangnya, ini merupakan salah satu masalah yang cukup rumit, terutama bagi para pemula yang belum bisa mengidentifikasi nada gitar. Karena dibutuhkan pendengaran yang sangat akurat dan teliti untuk menyatakan apakah nada-nada pada senar tersebut sudah akurat atau masih belum akurat. Pada skripsi ini, penulis merancang sebuah aplikasi berbasis android yang berfungsi sebagai alat deteksi dari nada gitar. Untuk proses tuning tersebut digunakan algoritma FFT yang berfungsi untuk menghitung transformasi Fourier diskrit dengan cepat dan efisien. Suara dari senar gitar digunakan sebagai input berupa pengirim sinyal masuk dalam bentuk frekuensi dan mengolah frekuensi tersebut sehingga menghasilkan output berupa parameter jarum indikator dalam bentuk antarmuka. Dari hasil penelitian ini didapatkan aplikasi gitar ini menghasilkan tuner gitar yang mampu berjalan secara realtime dengan akurasi yang cukup bagus.

Kata Kunci : Aplikasi, Android, Tuner Gitar, FFT, Frekuensi

Abstract

Guitar is a musical instrument that is very popular and widely used in the world. To produce harmonic and precise tones, tuning is required to make guitar's sound working well. Unfortunately, this is one problem that is quite complicated, especially for beginners who have not been able to identify the tone of the guitar. Because it takes a very accurate and precise hearing to state whether the tones in the strings still are accurate or not accurate. In this paper, the author designs an android-based application that serves as a means of tone detection of the guitar. The FFT algorithm is used for tuning process which serves to calculate the discrete Fourier transform quickly and efficiently. The sound of a guitar string is used as the input of the incoming signal in the form of frequency, then the frequency of processing to generate output parameter needle indicator in the form of interface. From the results of this study, the application of this guitar produces a guitar tuner that is capable of running in real time with pretty good accuracy

Keywords : Application, Android, Guitar Tuner, FFT, Frequency

1. Pendahuluan

Alat musik merupakan instrumen yang dapat memproduksi suara yang teratur. Dalam fungsinya alat musik mengandung nilai-nilai keindahan dan kehadiran alat musik akan tampak dari bunyi-bunyi yang khas dimana tempat alat musik itu berasal. Banyak jenis alat musik seperti gitar, piano, seruling, drum, klarinet, biola, trompet dan masih banyak lagi. Sebagian alat musik terlihat sederhana tetapi dapat menghasilkan nada yang indah dan disukai oleh pendengarnya.

Tuner atau Penyelaras Nada Gitar adalah rutinitas wajib yang harus dijalani seorang pemain gitar sebelum mulai memainkan gitarnya. Gitar yang tidak selaras tentu tidak bisa menghasilkan nada yang enak untuk didengar.

Bagi seorang pengguna gitar terutama pemula terkadang masih mengalami kesulitan dalam mengenali nada. Terutama dalam melakukan setem yang tepat pada gitar merupakan suatu hal yang sulit untuk dilakukan dengan cara pendengaran nada tanpa bantuan aplikasi untuk mencocokkan setiap nadanya. Biasanya hal tersebut dikarenakan seorang pemain gitar pemula belum dapat mengingat nada pada masing-masing senar gitar tersebut. Kesulitan terjadi pada saat melakukan setem nada yang tepat pada masing-masing senar gitar

Dengan semakin pesatnya perkembangan teknologi saat ini, khususnya ditekhnologi perangkat *mobile* berbasis *Android* maka penulis memanfaatkan fenomena ini dengan membuat suatu aplikasi *smartphone* berbasis *mobile Android* sebagai alat bantu penyelaras nada.

2. Rumusan dan Batasan Masalah

Setelah penulis mengidentifikasi permasalahan yang ada, penulis memberikan batasan pada masalah penelitian ini adalah :

1. Input sistem berupa suara yang dihasilkan oleh masing-masing senar gitar yang diambil menggunakan *microphone*.
2. Output sistem berupa nilai frekuensi dan kecocokan nada gitar.
3. Objek Penelitian adalah gitar akustik yang memiliki enam senar.
4. Uji coba sistem dilakukan pada lingkungan hening dan apabila dipakai di keramaian dapat

menggunakan kabel dengan jek melalui *microphone*.

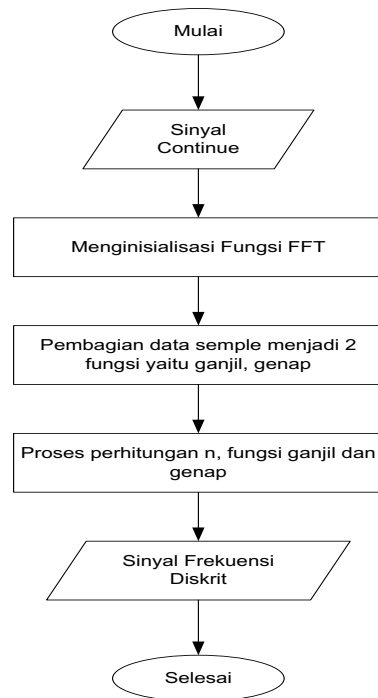
5. Pengujian dilakukan dengan kondisi awal semua senar gitar bernada tidak beraturan.
6. Aplikasi ini dibuat dengan menggunakan Bahasa pemrograman java berbasis *Android*.
7. Untuk mengatur tangga nada gitar menggunakan Algoritma *Fast Fourier Transform*.

3. Landasan Teori

3.1 Algoritma Blowfish

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

Berikut ini merupakan diagram alir Fast Fourier Transform :



Gambar 1. Diagram Alir Fast Fourier Transform

DFT merupakan metode yang berguna dalam menentukan amplitudo dan komponen-komponen frekuensi harmonik ke-m dari suatu sinyal periodik atau merupakan koefisien-koefisien deret fourier.

Berawal dari DFT-N data,

$$X[m] = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nm/N} \quad (1)$$

X(n) dipilih menjadi genap dan ganjil sehingga

persamaan (1) menjadi :

$$X[m] = \sum_{n=0}^{(\frac{N}{2})-1} x(2n)e^{-\frac{j2\pi(2n)m}{N}} + \sum_{n=0}^{(\frac{N}{2})-1} x(2n)e^{-j2\pi(2n)m/N} = \sum_{n=0}^{(\frac{N}{2})-1} x(2n)e^{-j2\pi(2n)m/N} + e^{-j2\pi m/N} \sum_{n=0}^{(\frac{N}{2})-1} x(2n+1)e^{-j2\pi(2n)m/N} \quad (2)$$

Dengan mendefinisikan $W_N = e^{-j2\pi m/N}$, persamaan

(2) menjadi :

$$X[m] = \sum_{n=0}^{(\frac{N}{2})-1} x(2n)W_N^{2nm} + W_N^m \sum_{n=0}^{(\frac{N}{2})-1} x(2n+1)W_N^{2nm} \quad (3)$$

Karena $W_N^2 = e^{-j(2\pi)^2} = e^{-j\frac{2\pi}{(\frac{N}{2})}}$, maka $W_N^2 =$

$W_{\frac{N}{2}}$. Jadi persamaan (3) menjadi :

$$X[m] = \sum_{n=0}^{(\frac{N}{2})-1} x(2n)W_{\frac{N}{2}}^{2nm} + W_{\frac{N}{2}}^m \sum_{n=0}^{(\frac{N}{2})-1} x(2n+1)W_{\frac{N}{2}}^{2nm} \quad (4)$$

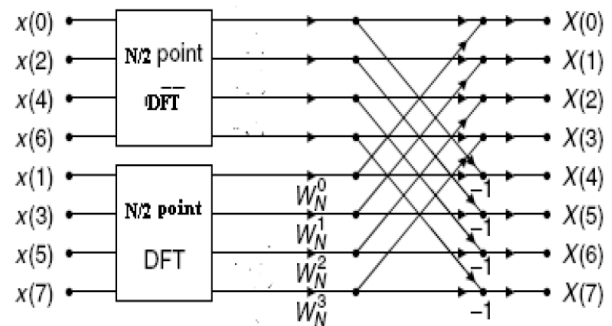
Setelah domain waktu dibagi dua, domain

frekuensi juga dibagi menjadi dua yaitu :

$$X(m + \frac{N}{2}) = \sum_{n=0}^{(\frac{N}{2})-1} x(2n)W_{\frac{N}{2}}^{n(m + \frac{N}{2})} + W_{\frac{N}{2}}^{m + \frac{N}{2}} \sum_{n=0}^{(\frac{N}{2})-1} x(2n+1)W_{\frac{N}{2}}^{n(m + \frac{N}{2})} = \sum_{n=0}^{(\frac{N}{2})-1} x(2n)W_{\frac{N}{2}}^{nm} - W_{\frac{N}{2}}^m \sum_{n=0}^{(\frac{N}{2})-1} x(2n+1)W_{\frac{N}{2}}^{nm} \quad (5)$$

Persamaan (4) dan (5) merupakan *FFT radix-2 Decimation in Time* (DIT) yang mana *sequence* data dipilah menjadi dua bagian menjadi genap dan ganjil dan menggambarkan gabungan dua DFT $N/2$ data. Penggunaan sifat periodik dari fungsi kernel membuat perhitungan menjadi lebih efisien karena cukup mengganti tanda operasi menjadi minus.

Secara sederhana persamaan (4) dan (5) digambarkan menggunakan diagram kupu-kupu (*butterfly diagram*) yaitu:



Gambar 2. Diagram Kupu-kupu (*butterfly diagram*) FFT Radix-2 DIT (Decimation in Time). Li Tan, Digital Signal Processing. (2008 : 129).

Cara Perhitungan FFT :

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

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

$$\begin{aligned} x(1) &= \frac{1}{3} \left[141,12 \left(\cos \frac{2\pi * 1 * 1}{3} \right) \right] - j \sin \frac{2\pi * 1 * 1}{3} + \\ &\quad \left[706,4 \left(\cos \frac{2\pi * 1 * 2}{3} \right) \right] - j \sin \frac{2\pi * 1 * 2}{3} + \\ &\quad \left[1130,88 \left(\cos \frac{2\pi * 1 * 3}{3} \right) \right] - j \sin \frac{2\pi * 1 * 3}{3} \\ &= 656,49 - 0,073j \end{aligned}$$

$$\begin{aligned} x(2) &= \frac{1}{3} \left[141,12 \left(\cos \frac{2\pi * 2 * 1}{3} \right) \right] - j \sin \frac{2\pi * 2 * 1}{3} + \\ &\quad \left[706,4 \left(\cos \frac{2\pi * 2 * 2}{3} \right) \right] - j \sin \frac{2\pi * 2 * 2}{3} + \\ &\quad \left[1130,88 \left(\cos \frac{2\pi * 2 * 3}{3} \right) \right] - j \sin \frac{2\pi * 2 * 3}{3} \\ &= 647,77 - 0,1454j \end{aligned}$$

$$\begin{aligned} x(3) &= \frac{1}{3} \left[141,12 \left(\cos \frac{2\pi * 3 * 1}{3} \right) \right] - j \sin \frac{2\pi * 3 * 1}{3} + \\ &\quad \left[706,4 \left(\cos \frac{2\pi * 3 * 2}{3} \right) \right] - j \sin \frac{2\pi * 3 * 2}{3} + \\ &\quad \left[1130,88 \left(\cos \frac{2\pi * 3 * 3}{3} \right) \right] - j \sin \frac{2\pi * 3 * 3}{3} \\ &= 633,21 - 0,2167j \end{aligned}$$

Gunakan rumus $[R^2 + I^2]^{1/2}$

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

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

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

Pada tahap ini, suara inputan yang telah digitalisasi tersebut akan dikenai proses framing dan windowing.

Framing diperlukan karena suara gitar merupakan sinyal yang berubah-ubah seiring dengan berjalannya waktu, tetapi ketika dilihat dalam waktu yang singkat, karakteristiknya cukup stasioner. Oleh karena itu dilakukanlah analisis spektral dalam jangka waktu yang singkat tersebut. Sinyal nada disegmentasi menjadi blok-blok dengan durasi kecil kisaran 20-30 ms yang juga dikenal dengan frame blocking. Dengan kata lain frame blocking adalah pembagian sinyal audio menjadi beberapa frame yang nantinya dapat memudahkan dalam perhitungan dan analisa sinyal, dimana sebuah frame terdiri dari beberapa sampel tergantung dari berapa detik suara akan disampel dan berapa besar frekuensi sampling.

Contoh Perhitungan *frame blocking* :

Signal : [1, 5, 8]

Misal dengan waktu tiap 800ms dan sampel rate 441100Hz

Maka didapatkan frame siza (N)

$N=44100*0,08=3528$ sampel

Overlapping frame (M)=1764

$x(n)=y(M+n)$

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

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

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

Sedangkan fungsi windowing adalah suatu cara melewati 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 dapat dinyatakan sebagai berikut :

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

Berikut ini beberapa jenis fungsi windows yang telah didefinisikan :

1. Hann Windows

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

2. Hamming Windows

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

3. Triangular Windows

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

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

4. Rectangular Windows

$$W(n)=1 \text{ untuk } n=0,1,2,\dots,N-1 \text{ (Trio Safaat 2016)}$$

Fungsi window (juga dikenal sebagai fungsi apodisasi atau fungsi tapering) adalah fungsi matematika yang memiliki nilai nol di luar dari beberapa interval yang dipilih. Berikut rumus untuk metode hanning window pada sample ke-N.

Proses Windowing menghasilkan spectrum suara dalam domain waktu, untuk tidak terjadi kesalahan dalam proses pencocokan nada maka spectrum domain waktu diubah menjadi sinyal frekuensi dengan menggunakan proses Fast Fourier Transform merupakan salah satu metode untuk transformasi sinyal suara menjadi sinyal frekuensi. Artinya, proses perekaman suara disimpan dalam bentuk digital berupa gelombang spectrum suara berbasis frekuensi. Hasil dari proses fast fourier transform menghasilkan pendeteksian gelombang frekuensi domain dalam bentuk diskrit

4. Metodologi Penelitian

4.1 Metode Pengembangan Perangkat Lunak

Metodologi penelitian pengembangan perangkat lunak yang digunakan oleh penulis dalam penelitian ini adalah pendekatan *Rational Unified Proses (RUP)*. Ciri khas dari RUP dalam proses pengembangan perangkat

lunak berbasis *The Unified Modeling Language (UML)* adalah adanya *use case* yang digunakan

4.2 Teknik Pengumpulan Data

Adapun metode pengumpulan data penulis lakukan untuk kelancaran dan kemudahan penulisan penelitian yaitu Metode Studi Pustaka dan Metode Observasi.

Metode Studi Pustaka yang dilakukan yaitu memperoleh data dan informasi dari berbagai jurnal, karya ilmiah dan buku-buku terkait yang berhubungan dengan permasalahan yang ada untuk mengungkapkan berbagai macam teori yang diperlukan dalam penulisan laporan ini.

Metode observasi yang dilakukan yaitu Penulis melakukan observasi atau pendekatan untuk mendapatkan data primer dengan cara mengamati terhadap kegiatan yang berjalan akan memberikan gambaran yang jelas terhadap permasalahan yang terjadi

5. Analisis dan Perancangan Sistem

5.1 Analisis Sistem

Tahapan analisis sistem ini yaitu membahas mengenai algoritma yang digunakan pada proses sinkronisasi nada pada tuning gitar dengan menggunakan algoritma *fast fourier transform*.

Analisis sistem adalah penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan. Tahap analisis sistem digunakan untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Frekuensi pada masing-masing senar gitar

Tabel 1 frekuensi nada gitar (Daryl Achilles, 2000)

String	Frequency	Scientific pitch notation
1 (E)	329.63 Hz	E4
2 (B)	246.94 Hz	B3
3 (G)	196.00 Hz	G3
4 (D)	146.83 Hz	D3
5 (A)	110.00 Hz	A2
6 (E)	82.41 Hz	E2



→ Pre-Processing → Gelombang Digital

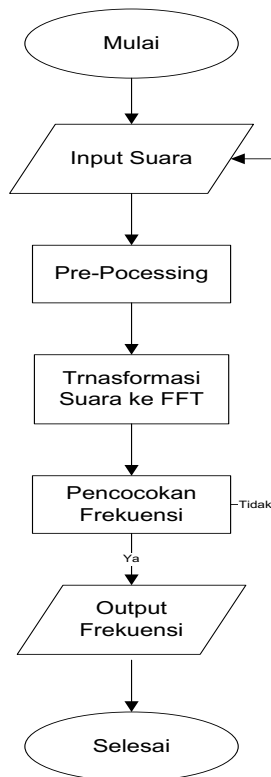
Pre-Processing

Pada tahap ini gelombang suara gitar akustik yang diinputkan sebelumnya akan diubah menjadi gelombang digital untuk dibuatkan spektrum domain frekuensinya dengan melakukan proses konversi analog ke digital, yaitu dengan mengubahnya menjadi bilangan biner secara terbalik atau dikenal dengan istilah bit-reversal.

Tabel 2. contoh bit reversal

Desimal	Biner	Bit-Reversal	Desimal
1	0001	1000	8
2	0010	0100	4
3	0011	1100	12
4	0100	0010	2
5	0101	1010	10
6	0110	0110	6
7	0111	1110	14

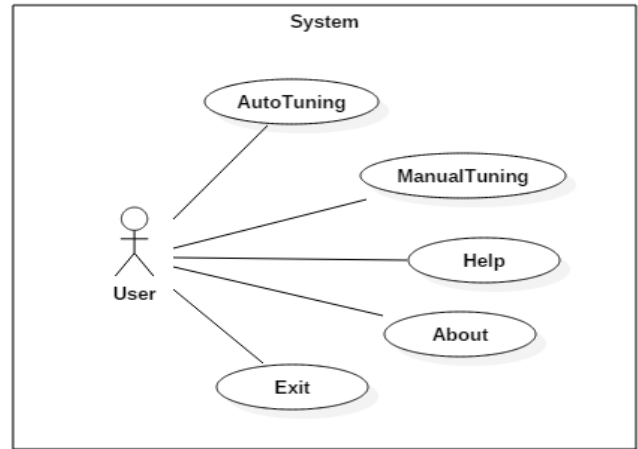
Flowchart Aplikasi Tuning Gitar



Gambar 3. Flowchart Aplikasi Tuning Gitar

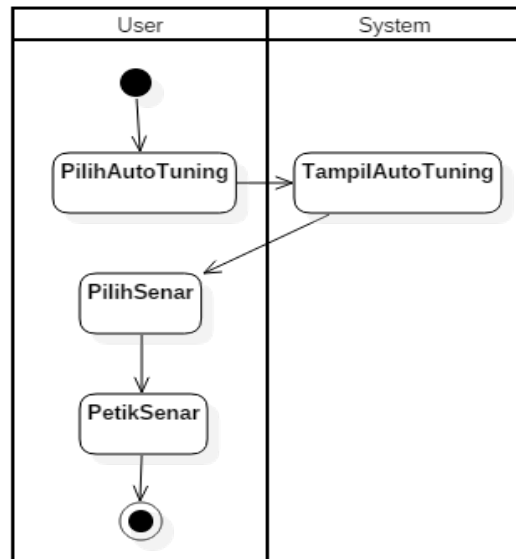
5.2 Perancangan Sistem
Use Case Diagram

Use case Diagram merupakan gambaran skenario dari interaksi antara user dengan sistem. Sebuah diagram use case menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi. Berikut adalah gambaran use case diagram dari aplikasi tuning gitar

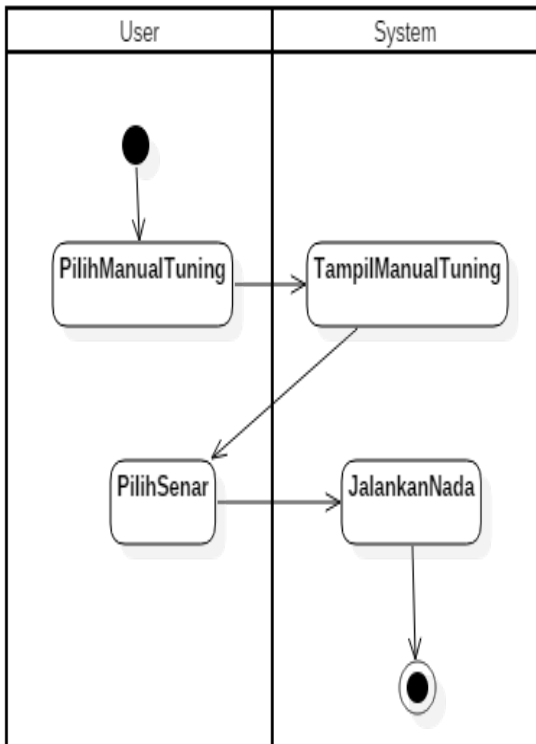


Gambar 4. Use Case Aplikasi Tuning Gitar

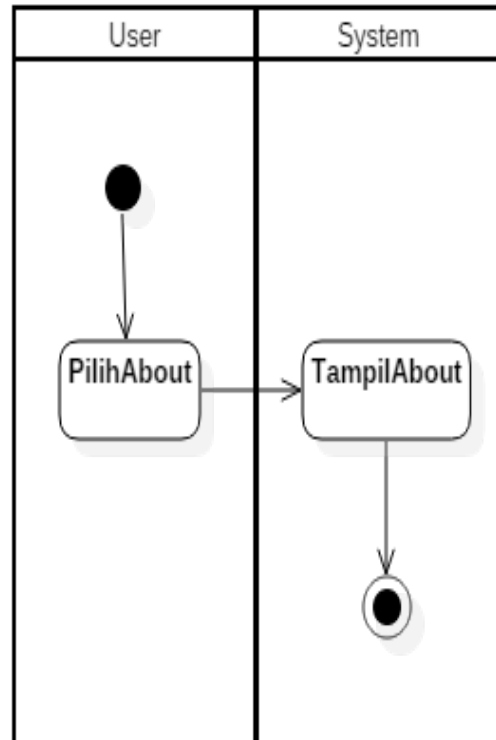
Activity Diagram



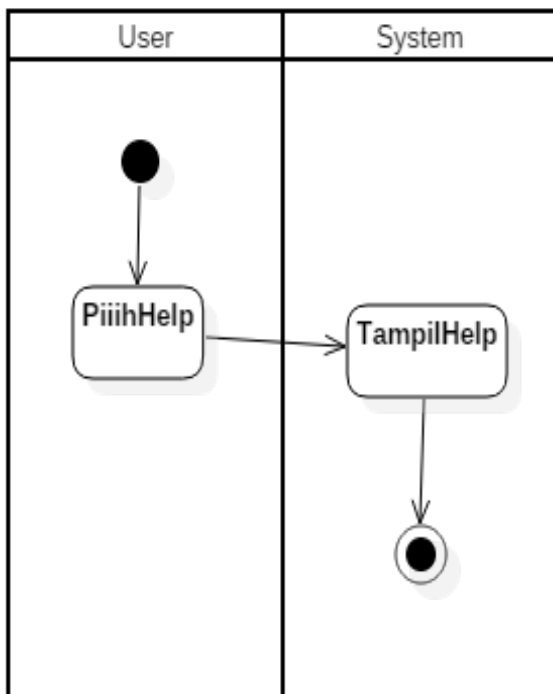
Gambar 5. Activity Diagram Auto Tuning



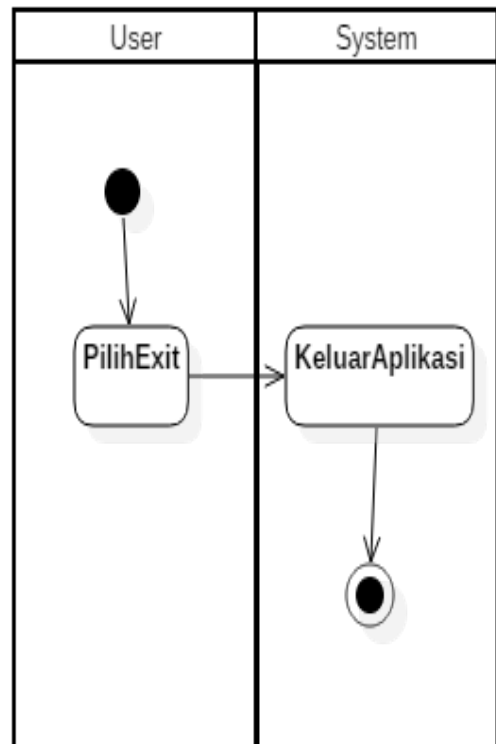
Gambar 6. Activity Diagram Manual Tuning



Gambar 8.. Activity Diagram About

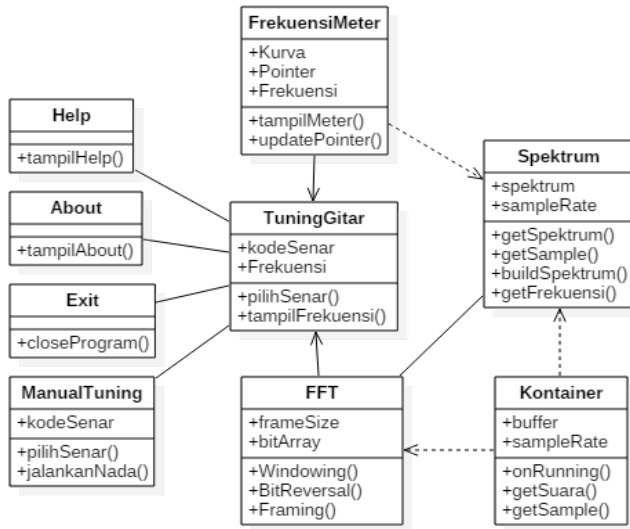


Gambar 7. Activity Diagram Help



Gambar 9.. Activity Diagram Exit

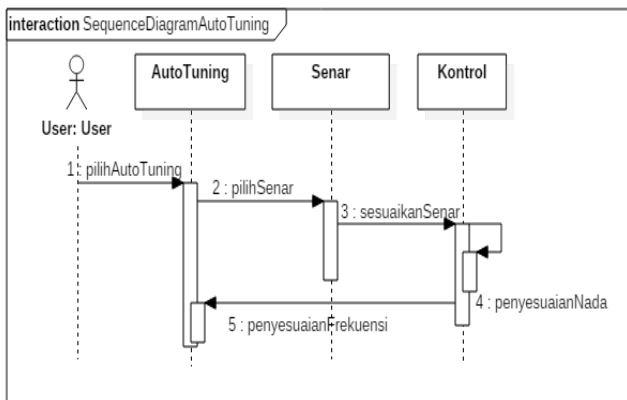
Class Diagram



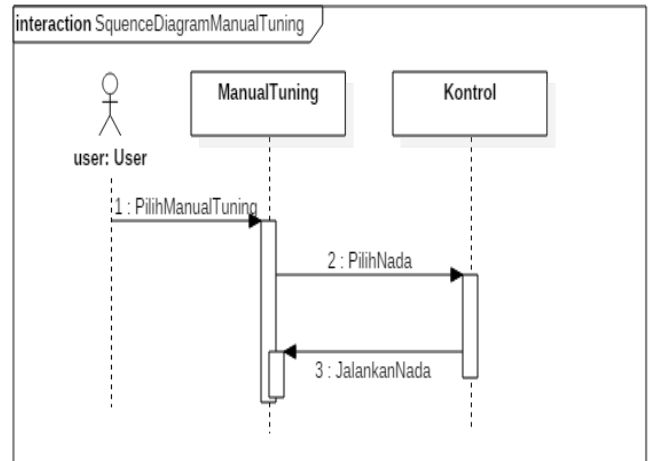
Gambar 10. Class Diagram

Berdasarkan class diagram diatas class utama yaitu TuningGitar yang memiliki atribut kode senar dan nilai frekuensi serta memiliki method tampil frekuensi dan pilih senar. Class TuningGitar tersebut memiliki asosiasi dengan class help, about dan exit dimana ketiga class tersebut tidak memiliki fungsi algoritmik. Class TuningGitar tersebut memiliki direct asosiasi dengan class FFT dan class FrekuensiMeter, kedua class tersebut akan terlibat secara langsung ketika proses tuning gitar. Sedangkan class FFT memiliki asosiasi dengan class Spektrum, dimana class ini memiliki ketergantungan dengan class FrekuensiMeter dan Kontainer.

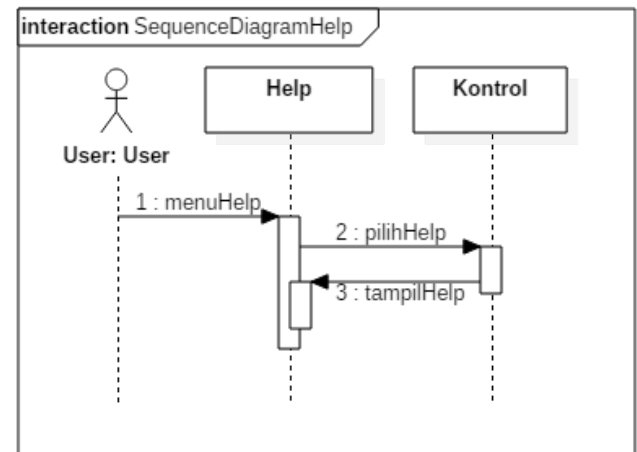
Sequence Diagram



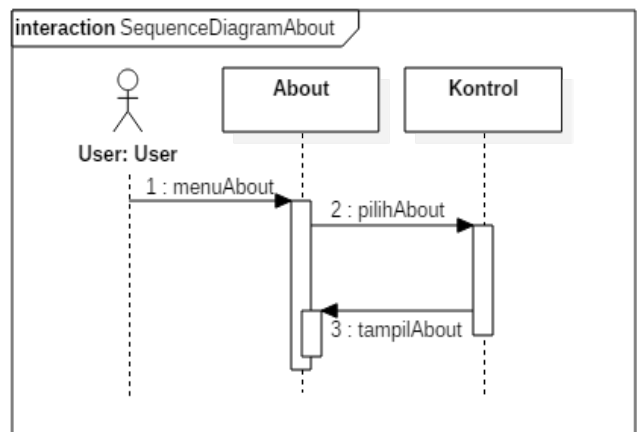
Gambar 11. Sequence Diagram Auto Tuning



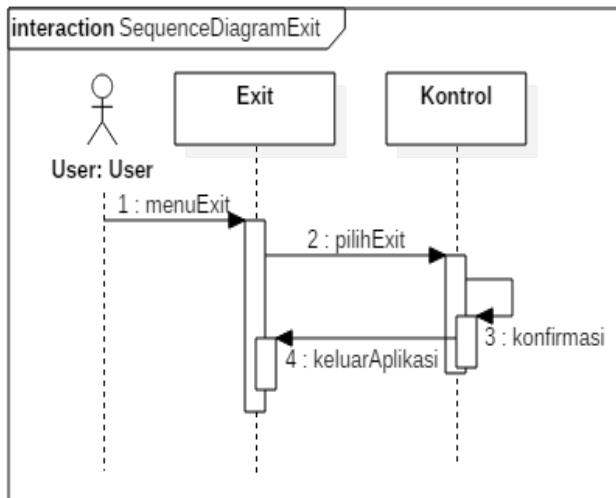
Gambar 12. Sequence Diagram Manual Tuning



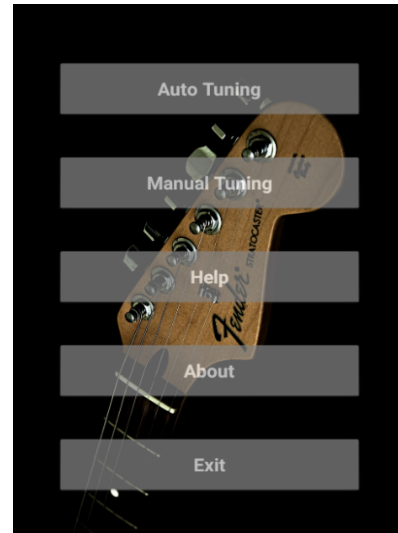
Gambar 13. Sequence Diagram Help



Gambar 14. Sequence Diagram About



Gambar 15. Sequence Diagram Exit



Gambar 16. Tampilan Menu Utama

6. Implementasi Sistem

6.1 Implementasi Sistem

Implementasi merupakan tahap pengembangan rancangan menjadi kode program, pada awal bagian ini di jabarkan spesifikasi perangkat keras dan perangkat lunak di mana program diimplementasikan. Bagian utama implementasi adalah penjabaran rancangan kelas yang ditulis dalam sintaks bahasa pemrograman java, disamping itu disajikan juga tampilan Aplikasi Tuning Gitar setelah di implentasikan pada telepon selular android.

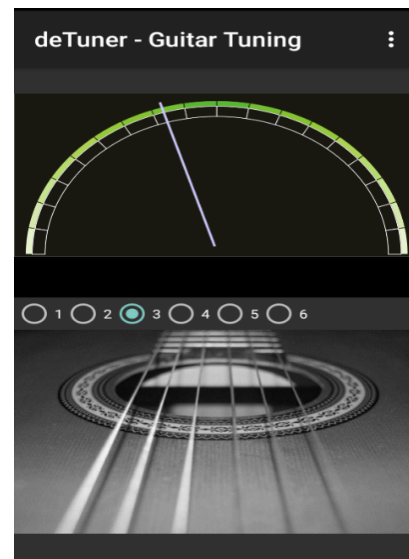
6.2 Desain Tampilan

1. Tampilan Menu utama Antar Muka

Tampilan menu utama dari aplikasi Tuning Gitar terdiri dari lima buah button. Kelima button tersebut memiliki fungsi masing-masing sesuai dengan nama button tersebut.

a. Tampilan Auto Tuning

Tampilan antarmuka dari activity Auto Tuning berisi beberapa obyek untuk melakukan Tuning Gitar. Ketika melakukan tuning, suara senar gitar yang dipetik harus sesuai dengan nada default secara umum.



Gambar 17. Tampilan Auto Tuning

Pendeteksian senar gitar yang sesuai dengan nada standar tersebut dapat dilihat pada layar berdasarkan jarum indikator pada frekuensi meter. Jika jarum indikator berada di area hijau tengah maka dapat disimpulkan bahwa suara senar gitar tersebut telah sesuai, atau tingkat kekencangan/kekendoran senar sudah benar.

b. Tampilan Login

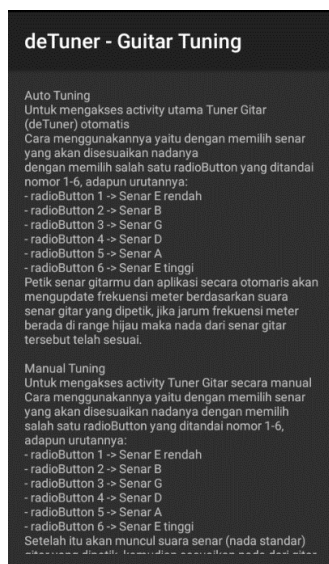
Tampilan menu dari activity manual tuner ini terdiri dari enam buah senar gitar yang direpresentasikan dalam radioButton dimana masing-masing radioButton tersebut memiliki fungsi untuk memutar suara standar dari senar gitar yang sudah pas.



Gambar 18. Tampilan Manual Tuning

c. Tampilan Help

Tampilan dari activity help ini memuat informasi seputar cara penggunaan aplikasi yang ditampilkan dalam bentuk textView serta objek focusable yang disetting false sehingga keyboard tidak akan muncul.

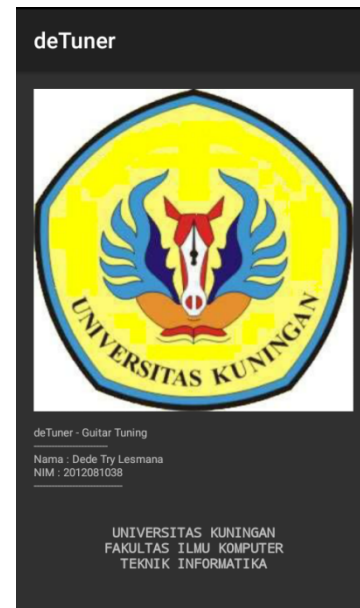


Gambar 19. Tampilan Help

Berdasarkan activity help tersebut maka diperoleh tampilan antarmuka dari file xml yang disimpan pada folder layout. File xml tersebut digunakan untuk mengatur penempatan serta beberapa atribut dari objek-objek yang ditampilkan.

d. Tampilan About

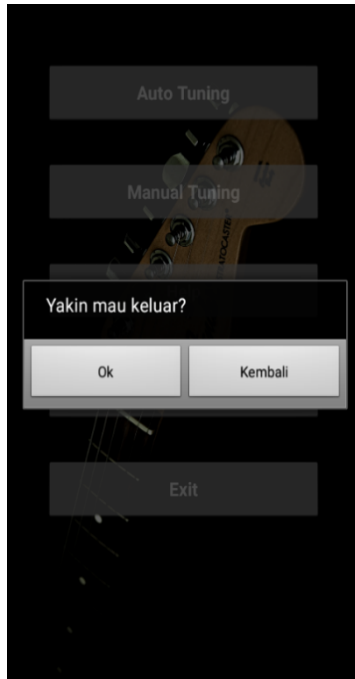
Tampilan dari activity about ini memuat informasi seputar pengembang yang ditampilkan dalam bentuk imageView dan textView serta objek focusable yang disetting false sehingga keyboard tidak akan muncul.



Gambar 20. Tampilan Form User

e. Tampilan Exit

Tampilan antarmuka dari fungsi exit ini yaitu pesan konfirmasi ketika pengguna memilih atau menekan button exit. Pesan konfirmasi tersebut ditampilkan dengan menggunakan class bawaan dari java yaitu class AlertDialog dengan pilihan positif untuk keluar dari aplikasi dan pilihan negatif untuk membatalkan keluar dari aplikasi.



Gambar 21. Tampilan Exit

Di *form* status user terdapat status yang dipublikasikan oleh kita dan user lain.

Pengujian Sistem

Pengujian dilakukan dalam dua tahap yaitu black box testing dan white box testing. Black box testing menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. White box testing menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan, pengujian dilakukan dengan memeriksa logik dari kode program.

1. Black Box Testing

Pengujian Black Box merupakan suatu metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Proses pengujian ini dilakukan untuk mengetahui apakah aplikasi perangkat lunak dapat berjalan dan berfungsi dengan benar.

Tabel 3. Pengujian Black Box Testing

No	Fungsi yang diuji	Cara Menguji	Hasil yang diharapkan	Hasil uji
1.	Mengaktifkan Aplikasi	Calon pengguna membuka aplikasi.	Sistem akan menampilkan form pendaftaran.	Sistem akan menampilkan peringatan untuk mengisi form pendaftaran.
2.	Daftar	Calon pengguna mengisi data yang disediakan.	Sistem akan menampilkan pemberitahuan bahwa data berhasil disimpan.	Sistem akan menampilkan pemberitahuan bahwa data berhasil disimpan.
3.	Kirim Pesan	Pengguna mengirimkan pesan kepada pengguna lainnya.	Sistem dapat mengirimkan pesan yang telah di enkripsi dan menampilkan dekripsi.	Sistem akan menampilkan pemberitahuan sukses.
4.	Status User	Pengguna memasukkan status yang dipublikasikan	Sistem menampilkan status yang telah dibuat oleh pengguna.	Sistem akan menampilkan pemberitahuan sukses.

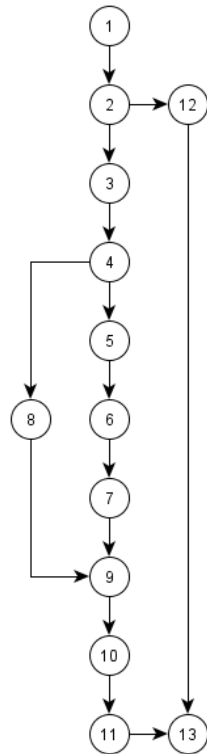
2. White Box Testing

Pengujian White Box berfokus pada structural kontrol program. Adapun metode yang digunakan dalam pengujian white box ini adalah metode basis Path untuk memastikan bahwa semua statement pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa kondisi logis yang telah diuji.

Langkah-langkah pengujian *White Box* adalah sebagai berikut :

1. $V(G) = E - N + 2$ hasilnya sama dengan $V(G) = P + 1$.
2. Flowgraph mempunyai region yang sama dengan jumlah $V(G)$ maka aplikasi dikatakan sudah terbukti efektif dan efisien.

Flow Graph Notation



Gambar 22. Flow Graph Notation Blowfish

Keterangan Gambar :

Node (N) = 13

Edge (E) = 14

Predicate Node (P) = 0

a. $V(G) = E - N + 2$
 $V(G) = 13 - 14 + 2$

= 1

b. $V(G) = P + 1$
 $V(G) = 0 + 1$

= 1

Deskripsi Tiap Node :

1. Baris kode program diatas merupakan fungsi bawaan android sdk yang digunakan untuk menampilkan layout dari file xml pada folder layout.

2. Baris kode program diatas digunakan untuk mendeklarasikan variable handler dimana variable tersebut memuat nilai frekuensi bertipe float dan menampilkannya terhadap fungsi updateDisplay.
3. Baris kode program diatas digunakan untuk mendeklarasikan variable mCapture yang berisi nilai dari variable mHandler sebagai parameter pada fungsi CaptureThread tersebut.
4. Selanjutnya fungsi CaptureThread akan bekerja dengan membawa parameter mHandler yang memuat suara dari senar gitar yang dipetik dan mengubahnya menjadi sinyal digital dengan memanggil fungsi spektrum.
5. Pada fungsi spektrum tersebut inputan merupakan suara gitar yang diterima oleh gadget kemudian akan diproses dengan cara melakukan bit traversal sehingga bisa diproses/ditransformasi lebih lanjut dengan memanggil class FFT.
6. Pada tahap ini proses dilakukan perubahan tipe data dari byte menjadi double terhadap suara yang masuk yang digunakan untuk mendapatkan nilai suara yang masuk.
7. Proses hanningWindow dilakukan untuk menghitung spektrum rata-rata dengan cara menganalisa suatu sinyal yang panjang dan mengambil satu bagian yang cukup mewakili.
8. Karena suara gitar yang masuk selalu berubah beriringan dengan waktu tertentu maka dilakukan proses pengambilan data berupa frekuensi dalam satuan waktu tersebut.
9. Agar pendeteksian suara gitar sesuai maka dideklarasikan beberapa variable untuk menampung suara dari senar gitar tersebut dengan cara menampung nilai standar dari suara senar gitar tersebut.
10. Setelah aplikasi menerima suara yang masuk maka akan dipanggil fungsi updateFrekuensiTarget(); untuk menampung nilai standar dari suara senar gitar tersebut.
11. Proses ini merupakan proses utama yang menampung nilai frekuensi sebagai acuan dasar nada standar dari suara senar gitar yang dipetik.
12. Proses ini dilakukan untuk menampilkan pergerakan jarum indikator berdasarkan suara yang diterima gadget dalam satuan waktu tertentu.

13. Proses ini digunakan untuk memilih ulang tuning senar gitar dengan cara memilih salah satu radioButton yang mewakili senar gitar tersebut.

Berdasarkan hasil perhitungan cyclomatic complexity terdapat empat independent path (jalur) yaitu:

Path 1 : 1 – 2 – 12 – 13

Path 2 : 1 – 2 – 3 – 4 – 8 – 9 – 10 – 11 – 13

Path 3 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 13

7. Kesimpulan dan Saran

7.1 Kesimpulan

Berdasarkan hasil penelitian pada bab-bab sebelumnya, maka dapat disimpulkan bahwa :

Penelitian ini bertujuan untuk membangun aplikasi tuning gitar absensi dengan menggunakan algoritma *Fast-Fourier Transform*. Berdasarkan hasil yang diperoleh dari perencanaan, pembuatan dan pengujian dapat disimpulkan beberapa hal, diantaranya :

1. Aplikasi dapat menerima input berupa suara dengan menggunakan microphone dengan hasil keluaran dari aplikasi ini yaitu nilai frekuensi periodik yang direpresentasikan dalam bentuk grafis.
2. Objek Penelitian adalah gitar akustik yang memiliki enam senar dengan uji coba sistem dilakukan pada lingkungan hening dan apabila dipakai di keramaian dapat menggunakan kabel dengan jek melalui *microphone*.

7.2 Saran

Mengingat masih adanya ketidaksempurnaan yang masih ada dalam aplikasi ini serta demi kepentingan pengembangan aplikasi ini sendiri, maka dapat diberikan beberapa saran – saran yang mungkin dapat dipertimbangkan :

1. Diharapkan agar pada proses tuning gitar lebih baik dimana aplikasi menyediakan banyak fitur seperti penyertaan nada lainnya.
2. Diharapkan penelitian ini dapat dilanjutkan agar didapat hasil yang lebih optimal khususnya dalam konteks pengolahan suara.

8. Daftar Pustaka

- [1] Daryl Achilles (2000). *Tensions of Guitar Strings*. Physics 398 EMI.
- [2] Kadir, Abdul, dan Terra CH. Triwahyuni (2003).. *Pengenalan Teknologi Informasi*. Andi. Yogyakarta.
- [3] Muhammad Fadhlulah (2015). Identifikasi Nada Gitar dengan Menggunakan Metode Fast Fourier Transform (FFT). Universitas Sumatra Utara
- [4] Robby Dianputra¹, Diyah Puspitaningrum², Ernawati³ (2014). Implementasi Algoritma *Fast Fourier Transform* Untuk Pengolahan Sinyal Digital Pada *Tuning* Gitar dengan *Open String*. Bengkulu: Informatika
- [5] Tan, Li. (2008). *Digital Signal Processing: Fundamentals and Applications*. Singapore: Elsevier dan Academic Press. Hlm. 129.
- [6] Trio Safaat (2006), Implementasi *Fast Fourier Transform* Pada Pengenalan Nada Piano Berbasis Android. Jurusan Teknik Informatika . Fakultas Sains dan Teknologi. Universitas Islam Negri Maulana Malik Ibrahim Malang