

# Perancangan dan Pembuatan Aplikasi Pembelajaran Gitar Berbasis Android Menggunakan Metode Fast Fourier Transform

Rizky Basatha<sup>1\*</sup>, Eddy Triswanto Setyoadi<sup>2</sup>, Arnold Hosea Dermawan<sup>3</sup>

<sup>1</sup>Program Studi Manajemen Informatika, Institut Informatika Indonesia, Surabaya, Jawa Timur

<sup>2</sup>Program Studi Sistem Informasi, Institut Informatika Indonesia, Surabaya, Jawa Timur

<sup>3</sup>Program Studi Teknik Informatika, Institut Informatika Indonesia, Surabaya, Jawa Timur

Email: <sup>1\*</sup>rizky@ikado.ac.id, <sup>2</sup>eddy@ikado.ac.id, <sup>3</sup>arnoldhosea2809@gmail.com

(Naskah masuk: 22 Sep 2021, direvisi: 21 Okt 2021, diterima: 28 Okt 2021)

## Abstrak

*Smartphone* adalah perangkat yang dapat membantu pekerjaan dalam berbagai hal. Seiring dengan berkembangnya teknologi, penggunaan *smartphone* semakin sering kita jumpai, salah satu contohnya adalah penggunaan *smartphone* sebagai media pembelajaran mandiri. Dengan belajar secara mandiri tentunya dapat menekan biaya pengeluaran untuk mengikuti kursus. Pada kasus kali ini kursus yang dimaksud adalah kursus belajar alat musik gitar. Dalam penelitian ini dirancang dan dibuat aplikasi berjudul "*Learn Guitar Chords*" yang bertujuan untuk dapat membantu para pemusik maupun non-pemusik untuk dapat memahami dan memainkan alat musik gitar. Aplikasi ini dikembangkan dengan memanfaatkan sistem operasi *Android* serta menggunakan *Android Studio* dalam pengembangannya. Berdasarkan hasil pengujian *black box testing*, fitur *tuner* pada aplikasi berhasil mendapatkan frekuensi dari suara inputan *user* menggunakan metode *Fast Fourier Transform* dengan baik serta berhasil menjalankan fitur-fitur tambahan seperti *metronome*, *chord library*, dan *mini game* yang dapat membantu pengguna untuk meningkatkan keahlian bermusiknya. Selanjutnya dilakukan pengujian dengan menggunakan metode kuesioner terhadap 30 responden. Aplikasi ini tergolong mudah untuk digunakan sehingga dapat membantu untuk meningkatkan keahlian bermusik pengguna. Hal ini dapat dibuktikan oleh 30 responden yang sudah mencoba aplikasi *Learn Guitar Chords*.

**Kata Kunci:** *Android*, Aplikasi Pembelajaran Gitar, *Tuner*, *Fast Fourier Transform*.

## *The Design and Development of Android-Based Guitar Learning Application Using Fast Fourier Transform Method*

### Abstract

*Smartphones are devices that can help work in various ways. Along with the development of technology, the use of smartphones is increasing in any sectors, one example is the use of smartphones as independent learning tools. By studying guitar independently, of course, it can reduce the cost of attending the guitar lesson. In this research, an application entitled "Learn Guitar Chords" was designed and created which aims to help both musicians and non-musicians understand how to play the guitar better. This application was developed by utilizing the Android operating system and using Android Studio Application in its development phase. Based on the results of the black box testing, the tuner feature in the Learn Guitar Chords application managed to get the frequency from the user input voice using the fast fourier transform method successfully and ran additional features such as a metronome, chord library, and mini game that could help users to improve their guitar skills. Furthermore, testing using a questionnaire method on 30 respondents. The result is application is relatively easy to use so that it can help to improve the user's musical skills. This can be proven by 30 respondents who have tried the Learn Guitar Chords application.*

**Keywords:** *Android*, *Guitar Learning Application*, *Tuner*, *Fast Fourier Transform*..

## I. PENDAHULUAN

Dengan adanya perkembangan teknologi dibidang musik tentunya membangun antusias sebagian orang untuk dapat berkembang dan mengasah kemampuan masing-masing individu. Akan tetapi waktu belajar yang diperlukan untuk menjadi mahir pada satu alat musik adalah sangat lama. Tidak hanya terkendala oleh waktu, kita juga diharuskan untuk mengeluarkan sebagian dana untuk mengikuti kursus musik tertentu dengan biaya yang tidak murah setiap bulannya.

Dalam kasus ini penulis memilih salah satu contoh alat musik, yaitu gitar. Setiap orang telah mengetahui dan mendengar alat musik ini. Tentunya orang yang ingin mempelajari atau bahkan ingin mahir dalam memainkannya membutuhkan waktu yang lama untuk dapat menghafal kunci-kunci yang ada, maupun kunci-kunci yang diperlukan dalam sebuah lagu serta dana yang tidak sedikit untuk dapat mengikuti kursus. Tidak hanya itu kita juga harus bertemu guru untuk melatih secara berkala.

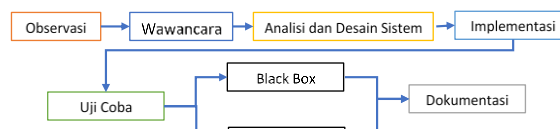
Maka dari itu penulis memikirkan bagaimana cara untuk dapat belajar alat musik gitar secara mandiri, tanpa memerlukan dana yang banyak dan harus bertemu guru secara berkala untuk berlatih. Pemilihan telepon seluler platform berbasis Android untuk salah satu pengembangan aplikasi selain bersifat open source dan lebih mudah dalam pengoperasiannya, sifat dari telepon seluler yang fleksibel menjadi juga salah satu alasannya [1].

Dengan aplikasi ini (*Learn Guitar Chords*) tentunya dapat memudahkan pengguna yang ingin mempelajari alat musik gitar. Pada aplikasi ini terdapat kumpulan *chord* gitar pada setiap nada. Pengguna juga dapat memastikan suara setiap senar gitarnya tidak salah dengan menggunakan *tuner* yang tersedia pada aplikasi. Pengguna juga dapat berlatih dengan memainkan *mini game* yang ada untuk melatih keahlian bermusiknya, serta pengguna dapat berlatih dengan lagu yang dapat diubah *pitch/transpose* dan *beat* sesuai keinginan. Dengan bantuan *Fast Fourier Transform* maka aplikasi dapat mendeteksi dan melihat spektrum getaran dari suara yang masuk. FFT adalah algoritma yang efisien untuk menghitung *Discrete Fourier*.

FFT adalah algoritma yang efisien untuk menghitung *Discrete Fourier Transform* (DFT) dan merupakan perhitungan yang sangat mendasar dalam sistem DSP. Arsitektur untuk FFT telah dipelajari secara ekstensif untuk aplikasi yang berbeda [2].

## II. METODOLOGI PENELITIAN

Metodologi penelitian pada penelitian ini menggunakan 3 bagian inti dari penelitian, yaitu analisis dan desain, implementasi, dan uji coba. Dalam tahap analisis dan desain, lebih fokus pada tampilan dan analisis suara yang akan diubah ke dalam gelombang frekuensi. Sedangkan pada tahap uji coba digunakan uji coba *BlackBox* dan kuisisioner. Diagram Blok Penelitian lebih detail dapat dilihat pada Gambar 1 di bawah.



Gambar 1. Diagram Blok Penelitian

Pada gambar di atas dapat dilihat bahwa ada beberapa tahapan sebelum tahap analisis dan desain sistem. Pada tahap observasi adalah tahapan dalam membandingkan dengan aplikasi yang pernah ada sebagai dasar proses pembuatan aplikasi. Sedangkan pada tahap wawancara, yaitu tahapan dimana melakukan percakapan dengan narasumber. Dimana narasumber tersebut merupakan pengguna yang dalam tingkat pemula yang ingin belajar alat musik gitar.

### A. Nada dan Frekuensi

Nada merupakan bunyi yang dihasilkan dari sumber bunyi yang memiliki frekuensi yang teratur [3]. Frekuensi adalah kecepatan perubahan amplitudo terhadap waktu. Frekuensi biasanya dinyatakan dalam satuan *Hertz* (Hz). Frekuensi juga dapat dihitung dengan menghitung jumlah getaran per satu detik atau dengan jarak relatif nada terhadap frekuensi. Terdapat juga tangga nada kromatik dengan interval 1, interval sendiri adalah jarak antara dua nada. Tangga nada kromatik terdiri dari C, C#, D, D#, E, F, F#, G, G#, A, A#, B, C.

Kres atau *sharp* (#) sendiri adalah penandaan terhadap nada yang memiliki  $\frac{1}{2}$  nada lebih tinggi. Sedangkan mol atau *flat* (b) adalah penandaan terhadap nada yang memiliki  $\frac{1}{2}$  nada lebih rendah. Sebagai contoh C#/D<sup>b</sup> yang berarti, nada C dinaikkan  $\frac{1}{2}$  nada atau nada D yang diturunkan  $\frac{1}{2}$  nada.

### B. Chord

*Chord* merupakan hasil bunyi dari 3 nada yang dibunyikan secara bersamaan [4]. *Chord* mayor terdiri dari tiga nada do, mi, sol, dan do sebagai *root* atau nada dasar. Sedangkan *chord* minor terdiri dari tiga nada do, ri, dan sol, dengan do juga sebagai *root*. *Chord* mayor biasa dituliskan CM atau C yang dibaca *chord* C mayor, sedangkan untuk *chord* minor dituliskan Cm yang dibaca *chord* C minor.

### C. Fast Fourier Transform

*Fast Fourier Transform* (FFT) merupakan algoritma untuk melakukan transformasi suara yang cepat dan efisien yang biasanya menggunakan DFT [5]. DFT sendiri merupakan proses transformasi sinyal dari *domain* waktu ke *domain* frekuensi.

Misalkan “ $x_0, \dots, x(N-1)$ ” merupakan bilangan kompleks. DFT didefinisikan dengan rumus:

$$x(nT) = \frac{1}{n} \sum_{k=0}^{N-1} x(k) e^{jk\Omega nT}, n = 0, 1, 2, \dots, N-1 \quad (1)$$

Menghitung deret ini diperlukan operasi aritmatika sebanyak  $O(N^2)$ . Sedangkan algoritma FFT hanya memerlukan operasi sebanyak  $O(N \log N)$  untuk menghitung deret yang sama.

D. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu model untuk merancang pengembangan perangkat lunak berbasis *object oriented* [6]. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software*.

III. HASIL DAN PEMBAHASAN

A. Analisis Permasalahan

Berdasarkan latar belakang masalah pada penelitian, dapat diidentifikasi beberapa permasalahan dalam proses pembelajaran alat musik gitar. Permasalahan yang timbul dari sisi peserta yang ingin belajar alat musik tersebut adalah:

- Harga untuk mengikuti kursus belajar gitar yang relatif mahal, semakin banyak pertemuan/durasi semakin mahal.
- Proses belajar mengajar yang diharuskan datang ke tempat/tatap muka pada waktu yang ditentukan, tidak dapat sewaktu-waktu.

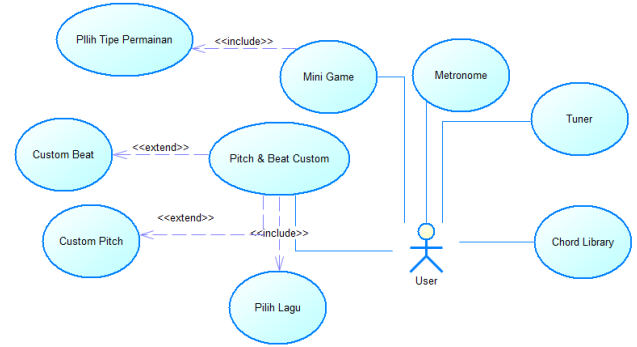
Dari permasalahan di atas dapat diambil kesimpulan bahwa menggunakan jasa kursus pelatihan alat musik gitar memerlukan biaya yang besar, waktu yang terikat pada jadwal, serta pelaksanaan belajar mengajar yang diharuskan bertempat pada tempat kursus. Dengan menggunakan aplikasi "Learn Guitar Chords", peserta kursus tidak perlu mengeluarkan biaya yang besar, menyiapkan jadwal yang harus ditepati, serta datang ke tempat kursus. Melainkan peserta dapat menentukan waktu dan tempat kapanpun dan dimanapun untuk belajar gitar serta menyiapkan kuota internet untuk mengunduh aplikasi.

B. Desain Sistem

Pada pembuatan sistem, diperlukan pemahaman mengenai rancangan dan alur kerja sistem tersebut. Dengan pemahaman tersebut dapat mempermudah pembuatan sistem untuk setiap bagian yang akan dibuat. Pada subbab ini akan dijelaskan mengenai *Unified Modeling Language* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, Dan *Sequence Diagram* untuk mempermudah memahami alur kerja sistem.

1. Use Case Diagram

*Use Case Diagram* merupakan salah satu pemodelan sistem dengan konsep berorientasi objek. Diagram ini dapat mendeskripsikan fungsionalitas sistem yang dilakukan sesuai yang diinginkan aktor yang berinteraksi dengan sistem. Setiap *use case* terdapat skenario atau alur dari proses *use case* yang dilakukan aktor dan sistem. Skenario yang menjelaskan *use case* terbagi menjadi dua, yaitu skenario normal untuk menggambarkan alur sistem tanpa terjadi kesalahan dan alternatif untuk menggambarkan alur sistem jika terjadi kesalahan sehingga harus melakukan proses lainnya. Berikut merupakan *Use Case Diagram* "Learn Guitar Chords"

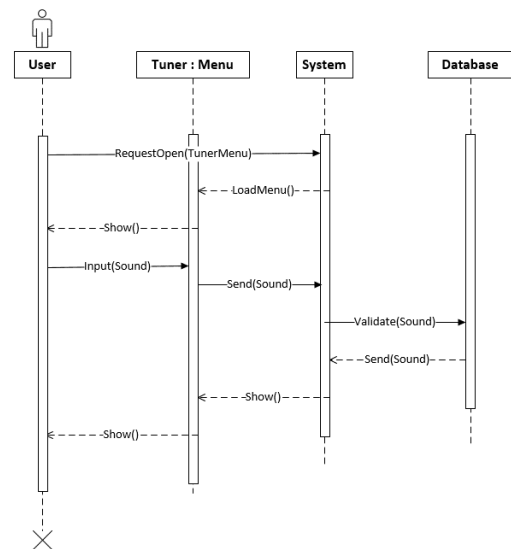


Gambar 2. Use Case Diagram Aplikasi Learn Guitar Chords

Pada Gambar 2 di atas dijelaskan terdapat satu aktor pengguna yang dapat memilih menu *metronome*, *tuner*, *chord library*, *mini game*, atau *pitch & beat custom*. Pengguna dapat memilih menu *tuner* untuk menyesuaikan nada setiap senar pada gitar. Pada menu ini pengguna akan memetik senar gitar secara bergantian dan menyesuaikan tinggi/rendahnya nada tersebut. Pada menu *metronome*, pengguna dapat berlatih agar permainan gitarnya *on beat* dengan mengatur *beat per minute* (*bpm*) dan *beat* sesuai keinginan. Pada menu *chord library*, sistem akan menampilkan kamus *chord* yang ada beserta bunyi dari *chord* tersebut. Pada menu *mini game* pengguna dapat memilih untuk melatih *chord diagram*, *chord sound*, atau *practice with guitar* dengan membunyikan *chord* tertentu. Pada menu *pitch & beat custom*, pengguna akan memilih lagu dari perangkat lalu pengguna juga dapat mengatur *beat* lagu dan tinggi rendahnya nada yang digunakan pada lagu.

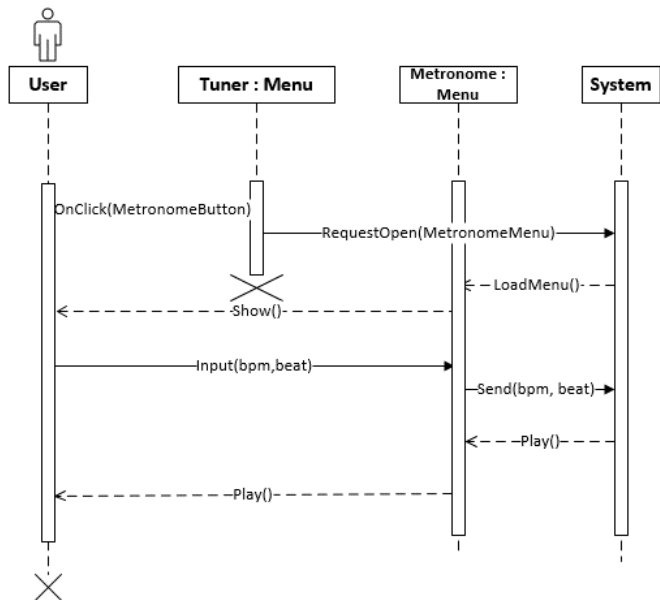
2. Sequence Diagram

*Sequence Diagram* merupakan diagram dengan gambaran detail rangkaian pesan antar objek dalam desain sistem. Diagram ini berfokus pada urutan interaksi visual dengan sumbu vertikal yang mewakili waktu serta sumbu horizontal yang mewakili objek. Satu *sequence diagram* dapat mewakili satu atau lebih *use case*.



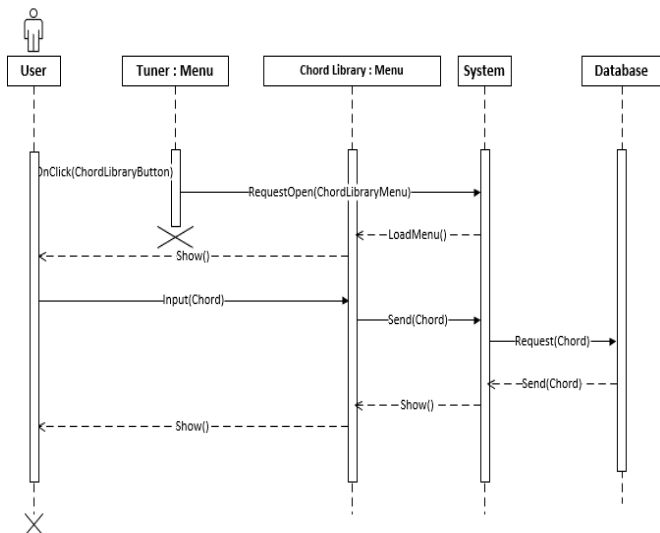
Gambar 3. Sequence Diagram Tuning

Seperti terlihat pada Gambar 3, *sequence diagram tuning* mendeskripsikan interaksi pengguna dengan sistem ketika berada pada menu *tuner*. Kondisi awal adalah saat pengguna membuka aplikasi, sistem akan membuka halaman utama dan menunggu *input user* berupa suara. Saat pengguna mengirimkan *input* suara, sistem akan memvalidasi suara tersebut dengan data yang ada lalu sistem akan menampilkan hasil kecocokkan suara.



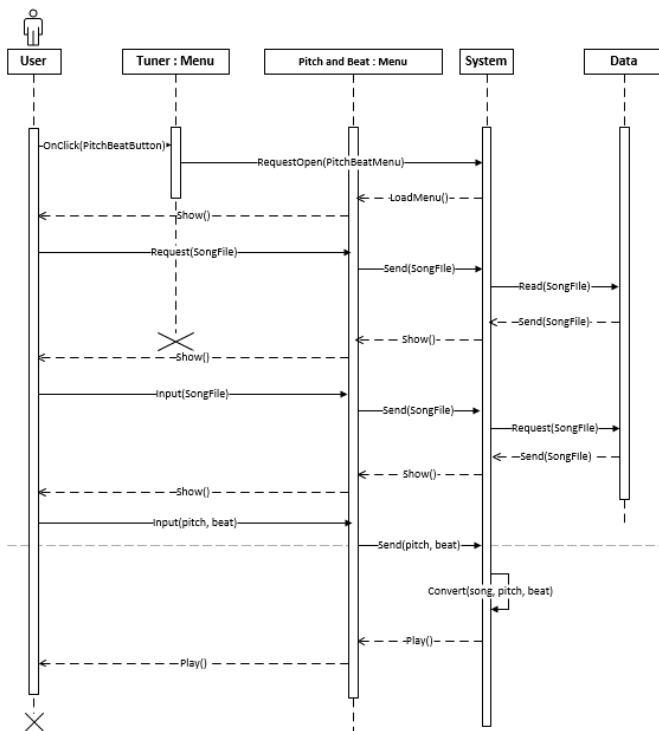
Gambar 4. *Sequence Diagram Metronome*

Terlihat pada Gambar 4, *sequence diagram metronome* mendeskripsikan interaksi pengguna dengan sistem ketika berada pada menu *metronome*. Pada halaman awal, pengguna menekan tombol *metronome* supaya dapat mengirimkan pesan untuk membuka *metronome* ke sistem, lalu sistem akan menampilkan *metronome* ke pengguna. Pada menu ini pengguna dapat memberi *input bpm* dan *beat*, yang kemudian sistem akan memainkan *beat* tersebut sesuai *input user*.



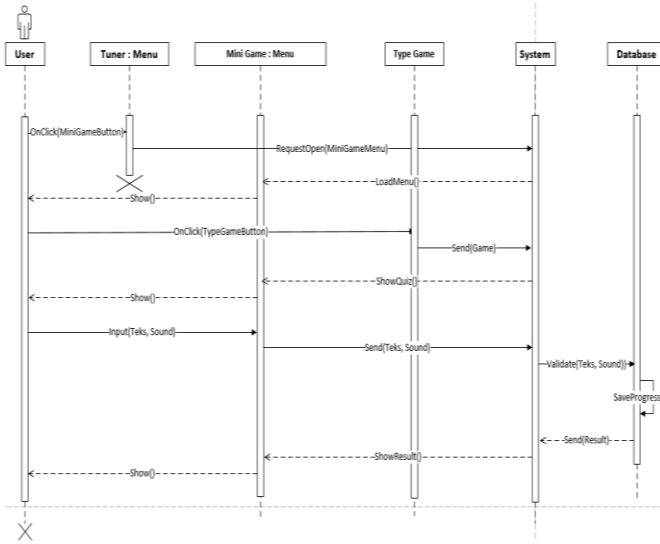
Gambar 5. *Sequence Diagram Chord Library*

Ditampilkan pada Gambar 5, *sequence diagram chord library* mendeskripsikan interaksi pengguna dengan sistem ketika berada pada menu *chord library*, pada halaman awal, pengguna menekan tombol *chord library* untuk dapat mengirimkan pesan untuk membuka *chord library* ke sistem, lalu sistem akan menampilkan *chord library* ke pengguna.



Gambar 6. *Sequence Diagram Pitch and Beat*

Pada Gambar 6 di atas, *sequence diagram pitch and beat* mendeskripsikan interaksi pengguna dengan sistem ketika berada pada menu *pitch and beat*. Pengguna menekan tombol *pitch and beat* untuk dapat mengirimkan pesan untuk membuka *pitch and beat* ke sistem, lalu sistem akan menampilkan *pitch and beat* ke pengguna. Pada menu ini pengguna dapat memilih lagu yang ada pada perangkat dengan menekan tombol pilih lagu, lalu sistem akan memindai berkas lagu pada perangkat, setelah pengguna memilih lagu, pengguna dapat mengatur *pitch* dan *beat* pada lagu sesuai keinginan, yang kemudian sistem akan mengkonversi dan memainkan lagu yang sudah diubah *pitch* dan *beat* sesuai *input user*.



Gambar 7. Sequence Diagram Mini Game

Pada Gambar 7 di atas, *sequence diagram mini game* mendeskripsikan interaksi pengguna dengan sistem ketika berada pada menu *mini game*. Pengguna menekan tombol *mini game* untuk dapat mengirimkan pesan untuk membuka *mini game* ke sistem, lalu sistem akan menampilkan *mini game* ke pengguna. Pada menu ini pengguna dapat memilih *game* yang ada seperti *game* dengan jawaban teks atau suara. Pada saat memulai *game*, sistem akan menampilkan soal yang kemudian pengguna memberikan jawaban berupa teks maupun suara. Setelah pengguna memberikan jawaban, sistem akan memvalidasi jawaban pengguna dengan jawaban yang telah tersimpan. Setelah itu sistem akan menyimpan progress pengguna dan menampilkan hasil jawaban serta *score*.

Pada sistem ini, dilakukan proses filtering menggunakan metode *windowing* untuk mengurangi adanya kebocoran frekuensi [7]. Di bawah ini merupakan kode dari proses *windowing* pada fitur *tuner*. Proses yang sama digunakan juga dalam *mini game*.

```

1. w = new Windowing();
2. w.initialize();
3. noteConversion = new
   NoteConversion(w);
4. startRecording();
5. protected Float
   doInBackground(Float.. params){
6. FastYin fastYin = new
   FastYin(SAMPLE_RATE,
   NUMBER_OF_SAMPLES); N =
   AudioRecord.getMinBufferSize(SAMPLE_R
   ATE, AudioFormat.CHANNEL_IN_MONO,
   AudioFormat.ENCODING_PCM_16BIT);
7. samples = new
   short[NUMBER_OF_SAMPLES];
8. AudioRecord recorder = new
   AudioRecord(MediaRecorder.AudioSource
   .VOICE_RECOGNITION, SAMPLE_RATE,

```

```

   AudioFormat.CHANNEL_IN_MONO,
   AudioFormat.ENCODING_PCM_16BIT, N *
   10);
9. recorder.startRecording();
10. while(tuner.isListening()) {
11. recorder.read(samples, 0,
   samples.length);
12. float[] inFSamples =
   shortToFloat(samples);
13. PitchDetectionResult pResult =
   fastYin.getPitch(inFSamples);
14. if(pResult.isPitched()) {
15. res = pResult.getPitch();
16. }
17. publishProgress(res);
18. }
19. recorder.stop();
20. recorder.release();
21. }
22. protected void
   onProgressUpdate(Float... values) {
23. tuner.turnLightsOff();
24. String st =
   noteConversion.getNote(values[0],
   tuner);
25. tuner.updateTvNote(st);
26. }

```

Dimana segmen program di atas berjalan saat pengguna memulai proses *tuning*. Pada baris ke 1 hingga baris ke 3 digunakan untuk inialisasi frekuensi yang akan digunakan untuk mencocokkan inputan dari pengguna. Baris ke 4 digunakan untuk memulai perekaman saat pengguna memetik senar gitar.

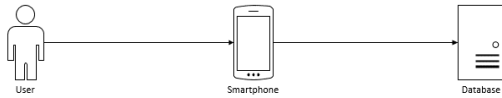
Pada baris ke 6 digunakan untuk inialisasi *library* yang akan digunakan. Pada baris ke 7, N berfungsi untuk menampung inputan suara dari pengguna. Pada baris ke 8 digunakan untuk menampung hasil pengurutan dari data *sample*. Pada baris ke 9 dan ke 10 digunakan untuk memulai perekaman dari *audio recorder*. Pada baris ke 12 hingga ke 18 dilakukan pengecekan apakah *recorder* masih berjalan. Pada baris ke 12 dan 13 digunakan untuk membaca masukkan dan mengurutkannya. Pada baris ke 14 digunakan untuk mendapatkan *pitch* dari *sample* data tersebut.

Pada baris ke 15 dilakukan pengecekan jika sudah sesuai dengan *pitch* yang diinginkan. Jika sudah pada baris ke 16 dilakukan penampungan data tersebut. Pada baris ke 26 dan 27 dilakukan pemberhentian dari *recorder*.

Pada prosedur baris ke 30 digunakan setelah prosedur *doInBackground* berhenti. Pada baris ke 31 digunakan untuk me-reset indikator warna pada *imageview* jika frekuensi terdeteksi. Pada baris ke 32 digunakan untuk mendapatkan *note* yang diinginkan. Lalu pada baris ke 33 digunakan untuk mengubah teks hasil pada tampilan.

C. Desain Arsitektural

Desain arsitektural digunakan untuk menjelaskan alur dari aplikasi yang akan dibuat. Alur yang dimaksud berisi semua interaksi pengguna dengan aplikasi. Berikut gambaran desain arsitektural pengguna aplikasi:

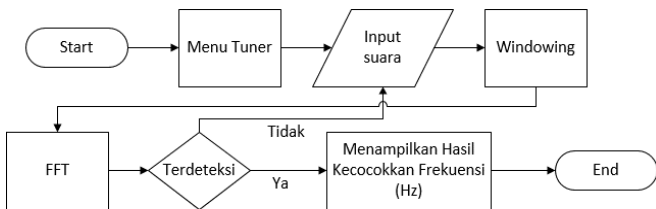


Gambar 8. Desain Arsitektural Aplikasi *Learn Guitar Chords*

Dari Gambar 8 di atas dapat diketahui bahwa pengguna dapat menggunakan aplikasi melalui *smartphone* yang terhubung dengan *database* yang berfungsi untuk menyimpan *progress mini game* dan *chord library* yang berupa gambar dan suara.

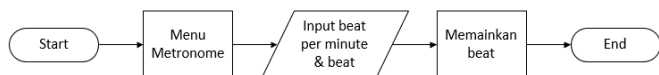
D. Desain Prosedural

Desain prosedural perlu dibuat untuk menetapkan detail algoritma yang akan dinyatakan dalam aplikasi ini. Desain prosedural dibuat setelah menyelesaikan desain sistem, desain arsitektural, dan desain rancangan antarmuka.



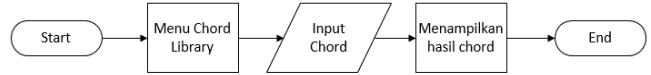
Gambar 9. *Flowchart Tuner*

Seperti terlihat pada Gambar 9, pada menu *tuner* pengguna akan memberikan *input* berupa suara dengan memetik senar gitar. Setelah itu sistem akan melakukan *windowing* untuk mengurangi kebocoran spektral yang dapat timbul akibat rendahnya *sample rate* (Fs). Setelah itu dilakukan FFT yang bertujuan untuk mengubah sinyal dari domain waktu ke dalam domain frekuensi, deretan angka dalam *domain* waktu  $f(x)$  diubah ke dalam domain frekuensi  $f(u)$ . Setelah itu jika terdeteksi sistem akan menampilkan hasil kecocokan frekuensi standar *tuning* gitar. Jika tidak sistem akan memproses ulang.



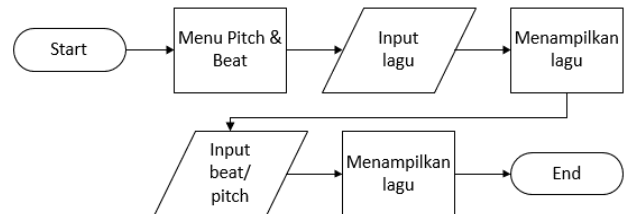
Gambar 10. *Flowchart Metronome*

Seperti terlihat pada Gambar 10, pada menu ini pengguna memberikan masukkan *beat* dan *beat per minute* ke dalam aplikasi yang kemudian sistem akan mulai memainkan *beat* yang sesuai dengan masukkan pengguna.



Gambar 11. *Flowchart Chord library*

Seperti terlihat pada Gambar 11, pada menu ini aplikasi akan menampilkan gambar dari *chord* masukkan dari pengguna. Pada menu ini pengguna juga dapat memainkan suara dari *chord* tersebut dengan menekan tombol.



Gambar 12. *Flowchart Pitch and Beat*

Seperti terlihat pada Gambar 12, pada menu ini untuk dapat melakukan perubahan *beat/pitch* pada lagu, pengguna harus memberikan *input* berupa data lagu yang terdapat pada perangkat *smartphone*. Setelah lagu terbaca oleh sistem pengguna dapat mengubah *pitch/beat* yang sesuai.

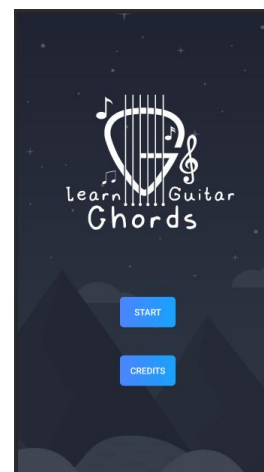


Gambar 13. *Flowchart Mini Game*

Seperti terlihat pada Gambar 13, pada menu ini pengguna disajikan beberapa soal untuk mengasah kemampuannya lalu pengguna akan memberikan *input* jawaban dapat berupa teks maupun suara *chord*. Sistem akan mencocokkan dengan data dan menampilkan hasilnya.

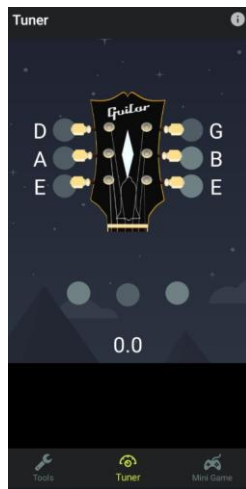
E. Desain Antar Muka

Berikut ini akan dijelaskan tampilan antar muka dari aplikasi *Learn Guitar Chords* yaitu halaman utama, *tuner*, *tools*, *metronome*, *beat*, *chord library*, *pitch and beat*, dan *mini game*.



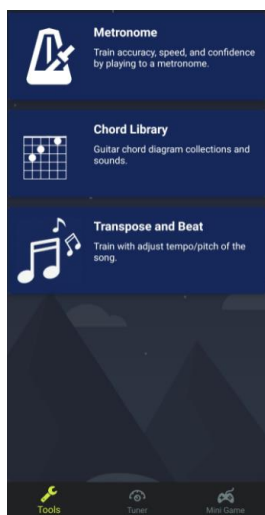
Gambar 14. Halaman Utama

Gambar 14 di atas adalah tampilan *utama* yang ditampilkan pada saat pengguna menjalankan aplikasi. Pada halaman ini terdapat tombol *start* dan tombol *credits*. Pada saat pengguna menekan tombol *start*, maka sistem akan menampilkan halaman *tuner*. Pada saat pengguna menekan tombol *credit*, sistem akan menampilkan halaman *credit*.



Gambar 15. Halaman *Tuner*

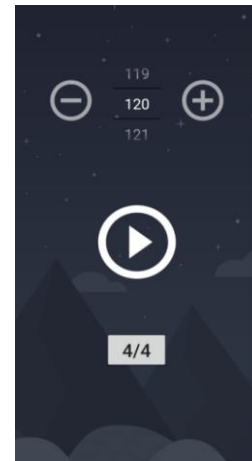
Gambar 15 di atas adalah tampilan ketika pengguna masuk ke halaman *tuner*. Pada halaman ini pengguna dapat menjalankan fitur *tuner* dengan memetik senar gitar secara bergantian yang nantinya sistem akan menampilkan frekuensi suara inputan dan menampilkan kecocokkan frekuensi untuk *tuning* gitar dengan aturan standar.



Gambar 16. Halaman *Tools*

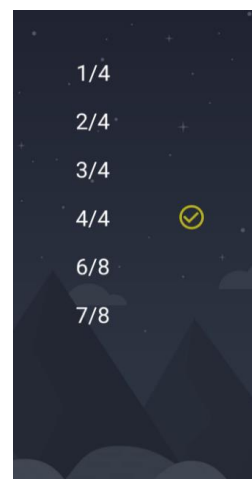
Gambar 16 di atas adalah tampilan halaman *tools* ketika pengguna menekan *navigation button tools* pada bagian bawah. Pada halaman ini terdapat tombol *metronome* yang berfungsi untuk membuka halaman *metronome*, tombol *chord library* yang berfungsi untuk membuka halaman *chord library*, dan tombol *transpose and beat* yang berfungsi untuk membuka halaman *transpose and beat*. Pada halaman *tools*

juga terdapat *navigation button* pada bagian bawah halaman *tools*.



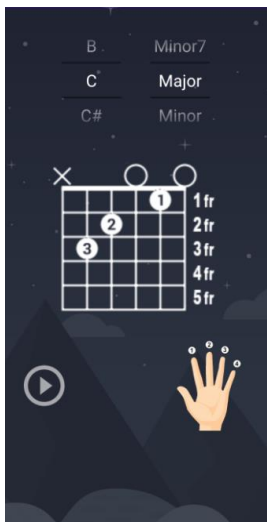
Gambar 17. Halaman *Metronome*

Gambar 17 di atas adalah tampilan halaman *metronome*. Pengguna dapat menekan tombol + untuk menambah 1 poin bpm dan menekan tombol - untuk mengurangi 1 poin bpm. Pengguna juga dapat melakukan *scroll* pada bpm *picker*. Pengguna dapat memainkan *beat* sesuai bpm yang terpilih dengan menekan tombol *play*. Pengguna juga dapat mengganti *beat* dengan menekan tombol *beat* pada bagian bawah.



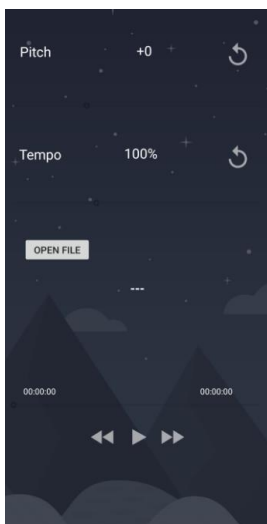
Gambar 18. Halaman *Beat*

Gambar 18 di atas adalah tampilan halaman *beat*, pengguna dapat mengganti *beat* dengan memilih dan menekan salah satu *beat* yang tersedia. Pada halaman ini pengguna hanya memilih salah satu *beat* dari pilihan yang disediakan pada bagian kiri. Pada bagian kanan terdapat *icon* sebagai *status* terpilihnya *beat*. Setelah memilih salah satu *beat* sistem akan menutup halaman *beat*, menyimpan *beat* baru, dan kembali ke halaman *metronome*.



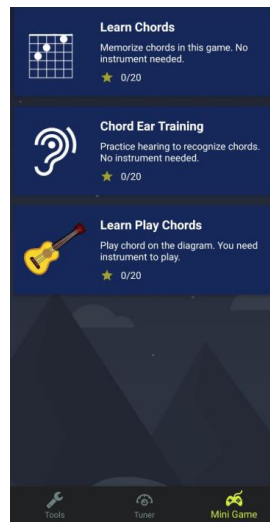
Gambar 19. Halaman Chord library

Gambar 19 di atas adalah tampilan halaman *chord library*. Pengguna dapat melihat *chord diagram*, melihat urutan jari untuk menekan *chord*, memainkan suara dari *chord* dengan menekan tombol *play*, dan mengganti *chord* dengan melakukan *scroll* pada pilihan *chord* yang disediakan.



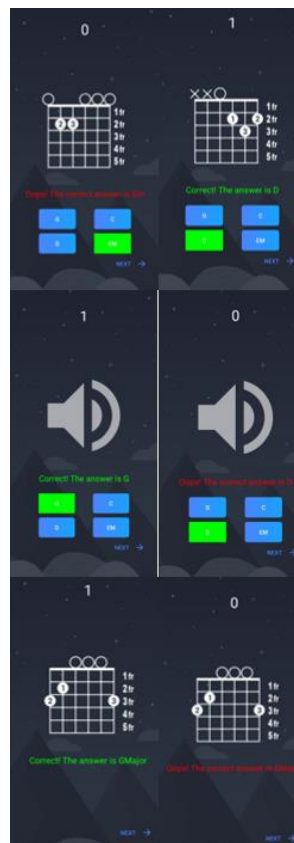
Gambar 20. Halaman Transpose and Beat

Gambar 20 di atas adalah tampilan halaman *transpose and beat*. Pengguna dapat memutar *file audio* dari perangkat. Setelah itu pengguna dapat mengubah *pitch/tempo* dengan menggeser *seekbar* dan mengembalikan *pitch/tempo* seperti semula dengan menekan tombol *reset*. Pengguna juga dapat melakukan *play/pause* dengan menekan tombol *play/pause*. Pengguna juga dapat lompat ke tiga detik sesudah dan tiga detik sebelum dengan menekan tombol *skip forward/backward*. Pada halaman ini pengguna diharuskan memilih *file audio* terlebih dahulu untuk dapat menggunakan fitur lainnya.



Gambar 21. Halaman Mini Game

Gambar 21 di atas adalah tampilan halaman *mini game*. Pengguna dapat memilih tiga jenis permainan. Pertama pengguna diminta untuk menebak *chord* dari gambar tersebut, yang kedua pengguna diminta untuk menebak *chord* dari suara yang dikeluarkan sistem, dan yang ketiga pengguna diminta untuk memberikan *input* suara dari *chord* dengan menggunakan gitar.



Gambar 22. Halaman Quiz

Gambar 22 di atas adalah tampilan halaman *quiz*. Pada halaman ini sistem akan memberikan soal dan pengguna



diminta untuk menjawab. Jika pengguna menjawab benar sistem akan menampilkan pesan “correct...”. Sebaliknya jika salah sistem akan menampilkan pesan “oops...” lalu pengguna dapat menekan tombol *next* untuk lanjut ke soal berikutnya. Setelah itu sistem akan menampilkan soal selanjutnya atau halaman *finish* jika soal sudah habis.

F. Pengujian

1. Uji Coba *BlackBox*

Uji coba yang dilakukan pada aplikasi *Learn Guitar Chords* ini menggunakan metode *black box testing* dan kuesioner. *Black box testing* adalah pengujian yang dilakukan dengan mengamati hasil eksekusi melalui data uji dan cek fungsional perangkat lunak. Dari hasil uji coba dengan metode *black box testing*, aplikasi dapat mendeteksi suara inputan pengguna pada fitur *tuner* seperti terlihat pada Tabel 1. Aplikasi juga dapat menjalankan fitur *chord library*, *metronome*, *pitch and beat transpose*, dan *mini game* dengan baik. Hal ini menunjukkan bahwa secara fungsionalitas aplikasi dapat berjalan seperti yang diharapkan.

Tabel 1. Hasil Uji Coba *BlackBox Tuner*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Akhir
1	Penggunaan tombol <i>tools</i>	Sistem menampilkan halaman <i>tools</i>	Berhasil
2	Penggunaan tombol <i>tuner</i>	Sistem menampilkan halaman <i>tuner</i>	Berhasil
3	Penggunaan tombol <i>mini game</i>	Sistem menampilkan halaman <i>mini game</i>	Berhasil
4	Memindai <i>input</i> suara	Sistem dapat mendeteksi suara <i>input</i> dan menampilkan hasil frekuensi suara	Berhasil

Hasil uji coba dengan metode *black box testing* aplikasi dapat mendeteksi suara inputan pengguna pada fitur *mini game* seperti terlihat pada Tabel 2.

Tabel 2. Hasil Uji Coba *BlackBox Mini Game*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Akhir
1	Penggunaan tombol <i>learn chords</i>	Sistem menampilkan halaman <i>quiz learn chords</i>	Berhasil
2	Penggunaan tombol <i>chord ear training</i>	Sistem menampilkan halaman <i>quiz chord ear training</i>	Berhasil
3	Penggunaan tombol <i>learn play chords</i>	Sistem menampilkan halaman <i>quiz learn play chord</i>	Berhasil

4	Penggunaan tombol <i>tools</i>	Sistem menampilkan halaman <i>tools</i>	Berhasil
5	Penggunaan tombol <i>tuner</i>	Sistem menampilkan halaman <i>tuner</i>	Berhasil
6	Penggunaan tombol <i>mini game</i>	Sistem menampilkan halaman <i>mini game</i>	Berhasil

2. Uji Coba Kuisisioner

Kuesioner merupakan metode yang menggunakan pengumpulan data dimana akan diberikan sejumlah pertanyaan atau pernyataan kepada pengguna aplikasi atau responden. Hasil uji coba dengan metode kuesioner oleh 30 responden dapat membuktikan bahwa aplikasi sangat bermanfaat dan berguna bagi pengguna. Menurut responden aplikasi juga sangat mudah untuk digunakan sehingga dapat membantu untuk meningkatkan keahlian bermusik pengguna.

Pada Tabel 3 dapat dilihat ada 10 pertanyaan di luar pertanyaan jenis kelamin. Dari seluruh jawaban dianalisis menggunakan metode *Descriptive Statistics Frequencies* pada aplikasi SPSS. Prosedur deskriptif dalam SPSS memberi cara mudah untuk mendapatkan gambaran komprehensif dari semua bidang numerik dalam kumpulan data [8]. Berikut adalah daftar dari 10 pertanyaan pada kuisisioner tersebut:

- Pertanyaan pertama berkaitan dengan minat pengguna mempelajari alat musik gitar.
- Pertanyaan kedua mengenai apakah pengguna pernah belajar alat musik gitar.
- Pertanyaan ketiga berkaitan apakah pengguna pernah mencoba aplikasi serupa.
- Pertanyaan keempat mengenai pendapat pengguna terhadap desain antar muka aplikasi.
- Pertanyaan kelima merupakan pertanyaan yang berkaitan dengan akurasi *tuner* pada aplikasi.
- Pertanyaan keenam merupakan pertanyaan tentang pendapat pengguna terhadap fitur yang disediakan.
- Pertanyaan ketujuh merupakan pertanyaan mengenai apakah seluruh informasi yang diberikan atau ditampilkan pada aplikasi sudah jelas dan dapat dipahami.
- Pertanyaan kedelapan berkaitan dengan kemudahan aplikasi untuk digunakan.
- Pertanyaan kesembilan merupakan pertanyaan yang berkaitan dengan peningkatan kemampuan gitar pengguna setelah menggunakan aplikasi.
- Pertanyaan kesepuluh berkaitan dengan apakah pengguna akan merekomendasikan kepada kerabat.

Tabel 3. Hasil Uji Coba Kuisisioner

No	JK	Soal									
		1	2	3	4	5	6	7	8	9	10
1	2	1	2	2	1	1	2	1	1	1	1
2	2	1	1	1	1	2	1	1	1	1	1
3	2	1	1	1	1	2	2	1	1	1	1
4	2	1	1	1	1	2	1	1	1	1	1

5	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	2	2	2	2	1	2	1
7	1	1	1	1	1	1	1	1	1	1	1
8	2	1	1	1	2	2	2	1	1	2	1
9	1	1	1	1	1	2	2	2	1	1	1
10	1	1	1	2	1	2	2	1	2	2	1
11	1	2	2	2	2	2	2	1	2	2	1
12	1	1	1	1	1	2	1	2	1	2	1
13	1	1	1	1	2	1	2	1	1	2	1
14	1	1	1	1	1	1	1	1	1	1	1
15	2	1	1	2	1	2	2	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1
17	2	1	1	1	1	1	1	1	1	2	1
18	1	1	2	2	2	2	2	2	2	2	1
19	1	1	1	1	1	1	2	1	1	1	1
20	1	1	1	2	1	1	2	1	1	1	1
21	1	1	1	1	1	1	2	1	2	1	1
22	1	1	1	2	1	1	2	1	2	1	1
23	1	1	1	2	1	1	2	1	2	1	1
24	2	1	1	1	1	1	2	1	1	1	1
25	1	1	1	1	1	1	1	1	2	1	1
26	1	1	1	1	1	1	1	1	2	1	1
27	1	1	1	2	1	1	1	1	2	1	1
28	2	1	1	2	1	1	2	1	2	1	1
29	1	1	1	1	1	1	1	2	1	1	1
30	1	1	1	2	1	1	2	1	2	1	1

#### IV. KESIMPULAN

Berdasarkan hasil uji coba yang telah dilakukan diperoleh beberapa kesimpulan. Aplikasi *Learn Guitar Chords* dikembangkan dengan menggunakan *Android Studio*, dengan menerapkan metode *Fast Fourier Transform* dapat memperoleh frekuensi dari inputan *user* dengan baik. Berdasarkan hasil pengujian yang dilakukan, aplikasi *Learn Guitar Chords* sangat bermanfaat dan berguna bagi pengguna. Dimana lebih dari 70% responden mendukung bahwa tampilan desain antarmuka pada aplikasi menarik. Lalu lebih dari 70% responden mendukung bahwa aplikasi informatif sehingga dapat memberi informasi yang berguna bagi pengguna. Lebih dari 70% responden mendukung bahwa aplikasi sangat membantu untuk meningkatkan keahlian

bermusik khususnya gitar. Semua responden (100%) mendukung bahwa responden akan merekomendasikan aplikasi *Learn Guitar Chords* kepada rekan pengguna. Lebih dari 80% responden dengan kategori pemusik pemula berpendapat bahwa aplikasi *Learn Guitar Chords* sangat membantu untuk meningkatkan keahlian bermusik mereka.

Berdasarkan hasil uji coba yang telah dilakukan diperoleh beberapa saran yang bermanfaat dan membangun yang dapat ditambahkan di pengembangan selanjutnya. Menambahkan fitur untuk mendeteksi *chord/note* pada file *audio* yang sedang diputar. Menambahkan lebih banyak pilihan dan gambar dari *chord* agar pengguna dapat melihat dan mendengar lebih banyak *chord*. Memberikan fitur *portrait lock orientation* agar orientasi layar aplikasi tidak berubah ketika perangkat diubah ke posisi miring 90°.

#### REFERENSI

- [1] Dewi, N.K.C., Anandita, I.B.G., Atmaja, K.J. & Aditama, P.W. (2018). Rancang Bangun Aplikasi Mobile Siska Berbasis Android. *Sintech*, Vol. 1(2), pp. 100-107.
- [2] Liu, W., Liao Q., Qiao, F., Xia, W., Wang, C. & Lombardi, F. (2017). Approximate Designs for Fast fourier transform (FFT) With Application to Speech Recognition. *IEEE Transactions on Circuits and System 1: Regular Papers*, Vol. 66(12), pp. 4727-4739
- [3] Aprilinda, Y. & Ariani, F. (2016). Aplikasi Tuner Gitar Berbasis Android Untuk Membantu Dalam Bermain dan Menyetel Gitar. *Explore: Jurnal Sistem Informasi & Telematika*, Vol. 6(2), pp. 48-63. Diakses dari <http://jurnal.ubl.ac.id/index.php/explore/article/viewFile/642/611> pada tanggal 20 Februari 2021.
- [4] Hidayat, R. (2016). *Beberapa Istilah Musik*. Diakses dari <http://staff.unila.ac.id/riyanhidayat/2016/12/08/istilah-musik/> pada tanggal 10 Maret 2021.
- [5] Oktarian, N., Sharippudin & Irawan. (2020). Perancangan Aplikasi Pembelajaran Musik Gitar Berbasis Android. *Jurnal Ilmiah Mahasiswa Teknik Informatika*, Vol. 2(3), pp. 186-198.
- [6] Sonata, F. & Sari, V.W. (2019). Pemanfaatan UML Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer To Customer. *Komunika*, Vol. 8(1), pp. 22-31. DOI: 10.31504/komunika.v8i1.1832
- [7] Yulio, I.B., Wijayanto, I. & Susatio, E. (2016). Perancangan dan Implementasi Tuner Gitar Berbasis Fast Fourier Transform dan Harmonic Product Spectrum Pada Platform Android. *e-Proceeding of Engineering*, Vol. 3(2), pp. 1536-1543.
- [8] Barry, K.S. & Babinec, A.J. (2017). *Data Analysis with IBM SPSS Statistics*. Birmingham: Packt.