

ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

PERGESERAN KERAPATAN WAKTU SECARA *REAL TIME* PADA SINYAL *ADUIO* DENGAN *TIME SCALE MODIFICATION* (TSM) YANG DITERAPKAN PADA *SENAM AEROBIC*

Afif Nur Azizah
NRP 0721154000024

Dosen Pembimbing
Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc.
Dr. Eko Mulyanto Yuniarno, ST., MT.

Departemen Teknik Komputer
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC184801

***REAL-TIME DENSITY SHIFT ON AUDIO SIGNALS WITH
TIME SCALE MODIFICATION (TSM) METHOD APLLIED
TO AEROBIC EXERCISE***

Afif Nur Azizah
NRP 0721154000024

Advisor
Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc.
Dr. Eko Mulyanto Yuniarno, ST., MT.

Departement of Computer Engineering
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pergeseran Kerapatan Waktu secara *Real Time* pada Sinyal *Audio* dengan *Time Scale Modification (TSM)* yang Diterapkan pada *Senam Aerobic***" adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

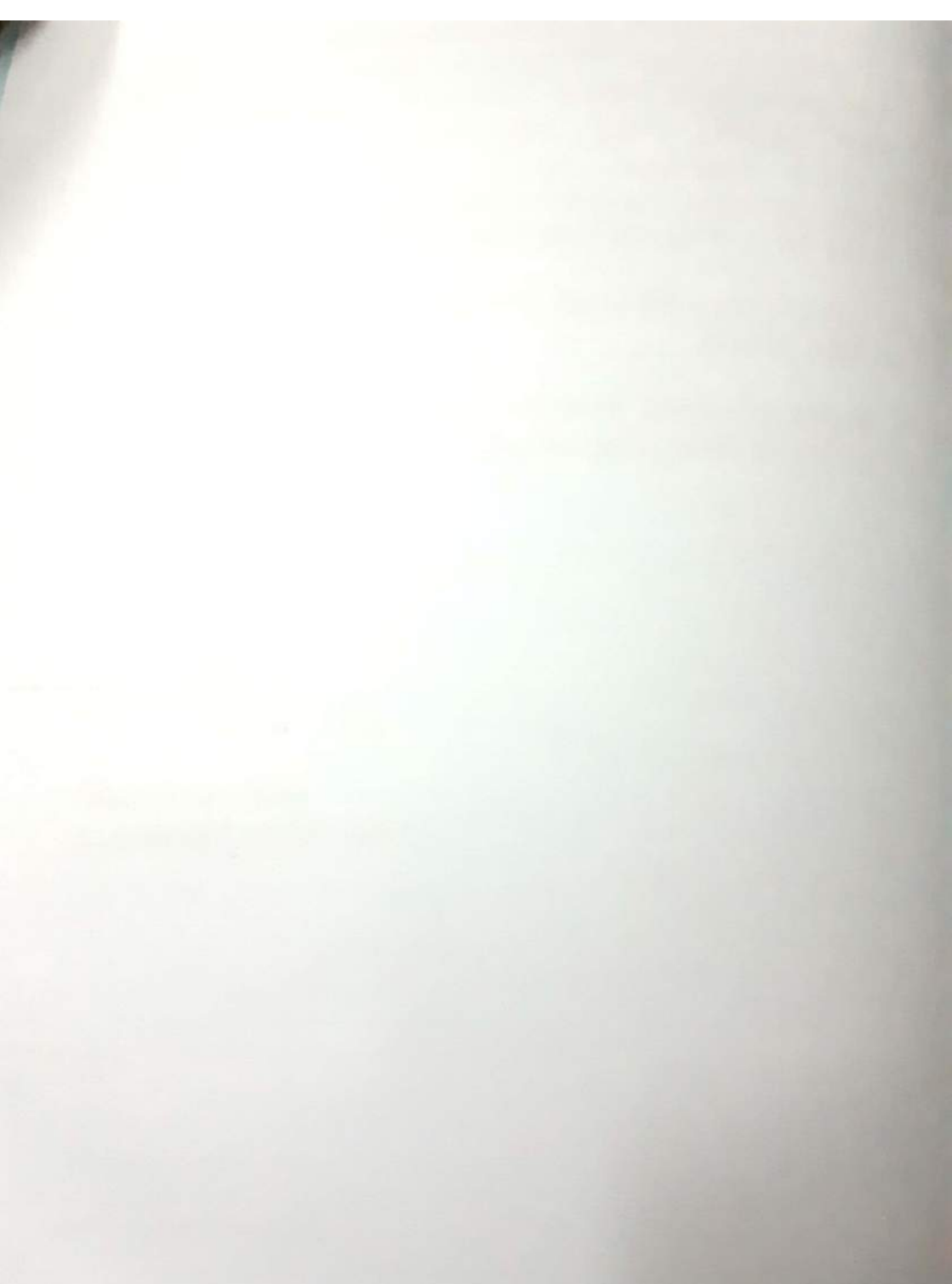
Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2019



Afif Nur Azizah
NRP. 0721154000024



LEMBAR PENGESAHAN

Pergeseran Kerapatan Waktu secara *Real Time* pada Sinyal *Audio* dengan *Time Scale Modification (TSM)* yang Diterapkan pada *Senam Aerobic*

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Afif Nur Azizah (NRP. 0721154000024)

Tanggal Ujian : 18 Juni 2019

Periode Wisuda : September 2019

Disetujui oleh :

Prof. Dr. Ir. Yovon K. Suprpto, M.Sc.
NIP. 195409251978031001

(Pembimbing I)

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP. 196806011995121009

(Pembimbing II)

Arief Kurniawan, ST., MT.
NIP. 197409072002121001

(Penguji I)

Dr. Diah Puspito Wulandari, ST., M.Sc.
NIP. 198012192005012001

(Penguji II)

Susi Juniastuti, ST., M.Eng.
NIP. 196506181999032001

(Penguji III)



Mengetahui
Kepala Departemen Teknik Komputer

Dr. I Ketut Eddy Purnama, S.T., M.T
NIP. 196907301995121001

RESEARCH REPORT

The purpose of this study was to investigate the effects of the proposed intervention on the target population. The study was conducted over a period of six months, during which time data was collected and analyzed. The results of the study indicate that the intervention had a significant positive impact on the target population, with improvements observed in the areas of interest.

The study was conducted in a controlled environment, and the results were consistent across all groups. The findings suggest that the intervention is a viable option for addressing the issues faced by the target population. Further research is needed to explore the long-term effects of the intervention and to identify the most effective components of the program.

The study was limited by the relatively small sample size and the short duration of the intervention. Despite these limitations, the results provide valuable insights into the effectiveness of the proposed intervention. The findings have important implications for the development of future interventions and for the implementation of existing programs.

The study was conducted in a controlled environment, and the results were consistent across all groups. The findings suggest that the intervention is a viable option for addressing the issues faced by the target population. Further research is needed to explore the long-term effects of the intervention and to identify the most effective components of the program.

The study was limited by the relatively small sample size and the short duration of the intervention. Despite these limitations, the results provide valuable insights into the effectiveness of the proposed intervention. The findings have important implications for the development of future interventions and for the implementation of existing programs.

Conclusion
The study has shown that the proposed intervention is effective in addressing the issues faced by the target population. The findings have important implications for the development of future interventions and for the implementation of existing programs.

ABSTRAK

- Nama Mahasiswa : Afif Nur Azizah
Judul Tugas Akhir : Pergeseran Kerapatan Waktu Secara *Real Time* pada Sinyal *Audio* dengan *Time Scale Modification* (TSM) yang Diterapkan pada Senam *Aerobic*
Pembimbing : 1. Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc.
2. Dr. Eko Mulyanto Yuniarno, ST., MT.

Modifikasi Skala Waktu (TSM) adalah suatu metode untuk mempercepat atau memperlambat sinyal suara tanpa mempengaruhi konten frekuensi, seperti nada yang didengar dari setiap komponen nada yang ada. Sebagai contoh, TSM pidato harus dilakukan dengan berbicara pada kecepatan yang lebih lambat atau lebih cepat sesuai dengan kebutuhan, tanpa distorsi vokal yang diucapkan. Dalam hal ini, teknologi TSM digunakan untuk melakukan percepatan atau perlambatan hasil rekaman *audio*, tidak hanya mengubah durasi daripada *audio*, tetapi dapat memengaruhi frekuensi sinyal. Hal ini dapat mendistorsi konten daripada *audio* tersebut sehingga dapat mengubah kecepatan *audio* atau dapat dikatakan terdengar lebih cepat, lebih lambat, ataupun normal. Pada penelitian ini akan dilakukan proses *time scalling* atau *time stretching*, yaitu pergeseran kerapatan waktu pada sinyal audio yang dituju tanpa mengubah frekuensi awal dengan menggunakan metode *Waveform Similarity and Overlap Add* (WSOLA). Modifikasi ini menghasilkan efek kecepatan pemutaran rekaman yang dipercepat atau diperlambat tanpa mempengaruhi *pitch* atau *timbre* dari sinyal asli. Penelitian ini dilakukan secara *real time* tanpa menunggu audio selesai terlebih dahulu atau dikatakan dalam satu durasi *audio*. Dalam percobaan ini terdapat beberapa macam rekayasa kecepatan pemutaran *audio*. Dalam hal ini percepatan *beat audio* atau perlambatan ritme *audio* dapat dilakukan diawal *audio*, ditengah-tengah *audio*, dan diakhir *audio*. Dimana percepatan *beat audio* ini menyesuaikan dengan ritme detak jantung pengguna yang dapat dideteksi dan diukur oleh perangkat *smartwatch Mi Band 2* dalam satuan detak jantung

yaitu bpm (*bit per minute*). Saat detak jantung pengguna cepat atau diatas 90 bpm maka *beat* kecepatan pada *audio* akan diubah menjadi diperlambat, ketika detak jantung lambat atau dibawah 70 bpm maka *beat* pada *audio* akan dipercepat, dan ketika detak jantung pengguna normal atau sama dengan 70-90 bpm maka *beat* pada *audio* menjadi normal. Hal ini dilakukan agar ritme detak jantung pengguna perangkat tersebut berada dalam keadaan *rate* normal.

Kata Kunci : WSOLA, *time scalling*, *beat audio*, detak jantung

ABSTRACT

Name : Afif Nur Azizah
Title : Real-Time Density Shift On Audio Signals With Time Scale Modification (TSM) Method Applied To Aerobic Exercise
Advisors : 1. Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc.
2. Dr. Eko Mulyanto Yuniarno, ST., MT.

Accelerating or decelerating audio recordings, not only change the duration of the audio, but also frequency signal. It can dis or decelerating audio recordings, not only change the duration of the audio, but also frequency signal. It can distort the content rather than the audio so the audio can change its speed or it can sound faster, slower or normal. In this test, time scaling or time stretching process will be carried out, namely, a shift in the time density of the intended audio signal without changing the initial frequency using the Waveform Similarity and Overlap Add method (WSOLA). This modification makes the speed of recording playback speed that into accelerated or slowed without affecting the pitch or timbre of the original signal. It is done in real time without waiting for the audio to run out first or can be compressed in one audio duration. It makes the acceleration of the audio beat or slowdown in audio rhythm can be done at the beginning of the audio, in the middle of the audio, and at the end of the audio. Where the beat adjusts to the user's heart rate detected by the Mi Band 2 smartwatch device with a heart rate measurement unit in bpm (bit per minute) If the user's heart rate is fast or above 90 bpm the beat on the audio speed will be changed to slow down. If the beat the heart is slow or below 70 bpm, the beat on the audio will be accelerated, and If the user's heart rate is normal or equal to 70-90 bpm, the beat on the audio becomes normal. This is done so that can make the user's heart rate condition at its a normal rate.

Keywords : WSOLA, Time scalling, Beat of music, Heart rate

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas segala karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Pergeseran Kerapatan Waktu Secara *Real Time* pada Sinyal *Audio* dengan *Time Scale Modification* (TSM) yang Diterapkan pada Senam *Aerobic*.**

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. I Ketut Eddy Purnama, ST., MT. selaku Kepala Departemen Teknik Komputer, Fakultas Teknik Elektro, Institut Teknologi Sepuluh Nopember.
3. Bapak Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc. selaku dosen pembimbing I dan Bapak Dr. Eko Mulyanto, ST., MT. selaku dosen pembimbing II yang selalu memberikan arahan selama mengerjakan penelitian tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Penulis juga berterimakasih kepada Bambang Aria Yudhistira untuk *support* yang telah diberikan.
6. Seluruh teman-teman dari Teknik Komputer dan Laboratorium B201 Teknik Komputer ITS.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2019

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak

Abstract

KATA PENGANTAR

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

NOMENKLATUR

| | | |
|----------|--|----------|
| 1 | PENDAHULUAN | 1 |
| 1.1 | Latar belakang | 1 |
| 1.2 | Permasalahan | 3 |
| 1.3 | Tujuan | 3 |
| 1.4 | Metodologi | 4 |
| 1.5 | Batasan masalah | 5 |
| 1.6 | Sistematika Penulisan | 6 |
| 1.7 | Relevansi | 7 |
| 2 | TINJAUAN PUSTAKA | 9 |
| 2.1 | <i>Time Scale Modification</i> (TSM) | 9 |
| 2.2 | Pengaruh Musik Terhadap Jantung | 9 |
| 2.3 | Manfaat <i>Aerobic</i> Bagi Kesehatan | 10 |
| 2.4 | Audio | 11 |
| 2.5 | <i>Time Stretching</i> | 13 |
| | 2.5.1 <i>Waveform Similarity Overlap-Add</i> (WSOLA) | 16 |
| | 2.5.2 Proses <i>Sampling</i> | 17 |
| 2.6 | Format File .WAV | 17 |
| 2.7 | <i>Photoplethysmography</i> (PPG) | 18 |
| | 2.7.1 Proses <i>Wearable Device</i> | 19 |
| 2.8 | notasi Musik | 19 |
| | 2.8.1 <i>Beat</i> atau Ketukan | 20 |

| | | |
|----------|---|-----------|
| 2.8.2 | Tempo | 21 |
| 2.9 | <i>Bluetooth</i> | 21 |
| 2.9.1 | Penggunaan Frekuensi pada <i>Bluetooth</i> | 22 |
| 2.9.2 | Komunikasi Data Menggunakan <i>Bluetooth</i> | 22 |
| 2.9.2.1 | Transmisi Asinkron (<i>Asynchronous Connection Less (ACL)</i>) | 22 |
| 2.9.2.2 | Transmisi Sinkron (<i>Synchronous Connection Oriented (SCO)</i>) | 23 |
| 3 | DESAIN DAN IMPLEMENTASI SISTEM | 25 |
| 3.1 | Desain Sistem | 25 |
| 3.1.1 | Pengolahan <i>Audio</i> | 29 |
| 3.1.2 | <i>Heart Rate</i> | 30 |
| 3.1.3 | Cara Kerja <i>User Interface (UI)</i> | 31 |
| 3.1.3.1 | <i>Folder</i> Daftar Lagu | 32 |
| 3.1.3.2 | <i>List</i> Lagu | 33 |
| 3.1.3.3 | <i>Pop Up</i> Pengolahan Musik | 33 |
| 3.1.3.4 | <i>Play List</i> User Interface | 35 |
| 3.2 | Implementasi Sistem | 35 |
| 3.2.1 | Pembagian <i>Audio</i> Menjadi Tiga Bagian | 36 |
| 3.2.2 | Pendektesian <i>Heart Rate</i> dengan <i>Mi Band 2</i> dan dikoneksikan dengan <i>audio</i> | 37 |
| 3.2.3 | <i>User Interface Screen</i> | 38 |
| 4 | PENGUJIAN DAN ANALISA | 41 |
| 4.1 | Pengujian Sistem dalam Penggunaan Memori | 41 |
| 4.1.1 | Memori yang Digunakan dalam Proses <i>-init</i> | 41 |
| 4.1.2 | Memori yang Digunakan dalam Proses <i>-play</i> | 44 |
| 4.2 | Pengujian Perangkat | 45 |
| 4.2.1 | Pengujian Pada <i>Audio</i> | 45 |
| 4.2.2 | <i>Heart Rate</i> pada <i>Mi Band 2</i> | 47 |
| 4.3 | Pengujian <i>Delay Mi Band 2</i> | 49 |
| 4.4 | Pengujian <i>User Interface</i> | 51 |
| 4.4.1 | <i>Buttton add</i> | 52 |
| 4.4.2 | Men-load Lagu | 53 |
| 4.4.3 | <i>Buttton Mi Band 2</i> | 54 |
| 4.4.4 | <i>Buttton play and stop</i> | 54 |
| 4.4.5 | Fungsi <i>Buttton Next</i> | 56 |

| | | |
|----------|--|-----------|
| 4.4.6 | Fungsi <i>Button Previous</i> | 57 |
| 4.4.7 | Fungsi <i>Button Delete</i> | 58 |
| 4.5 | <i>Eksperimen Desain</i> | 58 |
| 4.6 | Pengujian Kecepatan <i>Beat Audio</i> dan <i>Tempo Audio</i> | 60 |
| 4.6.1 | <i>Audio</i> Jenis Pop | 60 |
| 4.6.2 | <i>Audio</i> Jenis Rock | 60 |
| 4.6.3 | <i>Audio</i> Jenis Klasik | 61 |
| 4.6.4 | <i>Audio</i> Jenis <i>Disco</i> | 61 |
| 4.7 | Membandingkan Nilai <i>Audio</i> yang Terdahulu Sebelum Dieksekusi | 61 |
| 4.8 | Pengujian pada <i>Audio</i> | 63 |
| 4.8.1 | Kecepatan <i>Audio</i> dari Normal ke Cepat . . . | 64 |
| 4.8.2 | Kecepatan <i>Audio</i> dari Normal ke Lambat . . | 65 |
| 4.8.3 | Kecepatan <i>Audio</i> dari Cepat ke Normal . . . | 66 |
| 4.8.4 | Kecepatan <i>Audio</i> dari Lambat ke Normal . . | 67 |
| 5 | PENUTUP | 69 |
| 5.1 | Kesimpulan | 69 |
| 5.2 | Saran | 69 |
| | DAFTAR PUSTAKA | 71 |
| | Biografi Penulis | 73 |

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

| | | |
|------|--|----|
| 1.1 | Gambaran umum metode penelitian. | 4 |
| 2.1 | Gelombang suara | 12 |
| 2.2 | <i>Time stretching : overlap and add</i> | 14 |
| 2.3 | <i>Time stretching : synchronous overlap and add</i> | 14 |
| 2.4 | <i>PSOLA : pitch analysis and block windows</i> | 15 |
| 2.5 | <i>Depiction of the PSOLA algorithm</i> | 15 |
| 2.6 | Prinsip kesamaan bentuk gelombang <i>overlap-add</i> (WSOLA) | 16 |
| 2.7 | Mi Band | 19 |
| 2.8 | Simbol not | 20 |
| 2.9 | Nilai ketukan | 21 |
| 3.1 | Metodologi sistem secara umum. | 25 |
| 3.2 | Gambaran umum kerja sistem. | 27 |
| 3.3 | Skema alur kerja pengolahan musik. | 29 |
| 3.4 | Skema alur kerja <i>heart rate</i> | 30 |
| 3.5 | Tampilan desain UI pada aplikasi. | 31 |
| 3.6 | Folder daftar lagu. | 33 |
| 3.7 | <i>List</i> lagu. | 34 |
| 3.8 | <i>Pop up</i> pengolahan musik. | 34 |
| 3.9 | <i>Folder play list user interface</i> | 35 |
| 3.10 | Pembagian Kecepatan <i>Audio</i> | 36 |
| 3.11 | Mengkoneksi <i>heart rate Mi Band 2</i> dengan <i>audio</i> | 38 |
| 3.12 | Mengkoneksi <i>heart rate Mi Band 2</i> dengan <i>audio</i> | 39 |
| 4.1 | <i>-init audio</i> menggunakan Lenovo E330. | 42 |
| 4.2 | <i>-init audio</i> menggunakan ASUS. | 42 |
| 4.3 | <i>Folder</i> potongan <i>audio</i> menggunakan Lenovo E330. | 43 |
| 4.4 | <i>Folder</i> potongan <i>audio</i> menggunakan ASUS. | 43 |
| 4.5 | <i>-play audio</i> menggunakan Lenovo E330. | 44 |
| 4.6 | <i>-play audio</i> menggunakan komputer ASUS. | 44 |
| 4.7 | <i>Delay audio</i> menggunakan komputer Lenovo E330. | 46 |
| 4.8 | <i>Delay audio</i> menggunakan komputer ASUS. | 46 |

| | | |
|------|--|----|
| 4.9 | <i>Heart rate Mi Band 2</i> menggunakan komputer Lenovo E330. | 47 |
| 4.10 | Rata-rata <i>Heart rate Mi Band 2</i> menggunakan komputer ASUS. | 48 |
| 4.11 | <i>Delay Mi Band 2</i> menggunakan komputer Lenovo E330. | 49 |
| 4.12 | <i>Delay Mi Band 2</i> menggunakan komputer ASUS. | 50 |
| 4.13 | Rata-rata <i>delay Mi Band 2</i> menggunakan komputer Lenovo E330. | 51 |
| 4.14 | Rata-rata <i>delay Mi Band 2</i> menggunakan komputer ASUS. | 51 |
| 4.15 | Folder daftar lagu. | 52 |
| 4.16 | Memilih lagu. | 52 |
| 4.17 | Memproses kecepatan <i>audio</i> | 53 |
| 4.18 | Hasil penambahan lagu | 53 |
| 4.19 | Mengaktifkan <i>Mi Band 2</i> | 54 |
| 4.20 | Memainkan lagu. | 55 |
| 4.21 | Menghentikan lagu. | 55 |
| 4.22 | Mengklik <i>button next</i> | 56 |
| 4.23 | Lagu berganti. | 56 |
| 4.24 | Memilih lagu sebelumnya. | 57 |
| 4.25 | Lagu berubah. | 57 |
| 4.26 | Lagu berubah. | 58 |
| 4.27 | Eksperimen desain | 59 |
| 4.28 | Melakukan perbandingan nilai sebelum <i>audio</i> dieksekusi | 62 |

DAFTAR TABEL

| | | |
|-----|---|----|
| 3.1 | Parameter kecepatan <i>audio</i> berdasarkan detak jantung. | 32 |
| 4.1 | Pengujian <i>audio</i> dari normal ke cepat | 64 |
| 4.2 | Pengujian <i>audio</i> dari normal ke lambat | 65 |
| 4.3 | Pengujian <i>audio</i> dari cepat ke normal | 66 |
| 4.4 | Pengujian <i>audio</i> dari lambat ke normal | 67 |

Halaman ini sengaja dikosongkan

NOMENKLATUR

- α : *Skala waktu (s)*
 F : *Frekuensi (Hz)*
 T : *Periode sinyal*
 BPM : *beat per minute*

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Kesehatan merupakan hal yang sangat penting agar manusia dapat bertahan hidup dan melakukan aktivitas, terutama pada kesehatan jantung. Pentingnya menjaga kesehatan jantung dimana mengingat jantung sebagai organ utama dalam saluran peredaran darah dan menjadi sentral organ yang menentukan reaksi kinerja organ tubuh yang lain.

Jantung berperan dalam sistem sirkulasi dan berfungsi sebagai alat pemompa darah. Kontraksi dan relaksasi yang teratur dari otot-otot jantung memungkinkan darah yang mengandung banyak oksigen dari paru-paru dipompakan ke seluruh tubuh dan darah yang berasal dari seluruh tubuh dipompakan ke dalam paru-paru pada saat yang bersamaan.

Mekanisme ini berlangsung terus-menerus dan memungkinkan jaringan tubuh kita mendapatkan asupan oksigen dan nutrisi yang dibawa oleh darah untuk proses metabolisme tubuh. Apabila dalam kerjanya jantung mendapatkan suatu masalah tertentu akan mengganggu fungsi tubuh yang lain juga. Penyakit jantung memiliki faktor resiko yang dapat dicegah dengan pola hidup sehat salah satunya dengan cara rajin berolahraga.

Sebelum berolahraga sangat dianjurkan terlebih dahulu untuk melakukan perhitungan detak jantung menggunakan *Mi Band 2* untuk mengetahui kondisi dari pengguna. *Mi Band 2* merupakan *smartband* untuk mengamati tingkat aktivitas sepanjang hari, juga menghitung jarak jalan yang telah ditempuh, kalori yang terbakar, pencapaian olahraga dan detak jantung. Dengan adanya *Mi Band 2* yang nantinya akan menghubungkan detak jantung pengguna dengan irama *beat audio* untuk menemani pengguna dalam

berolahraga. Sistem pelacakan beat *audio* pada aplikasi multimedia merupakan salah satu hal penting. Dimana pada sistem ini mengenali posisi temporal dari not seperempat, maupun perpotongan not lainnya pada *audio*.

Seperti contohnya video editing, di mana trek visual dapat secara otomatis disinkronisasi dengan track audio menggunakan beat tracking, dimana untuk memfasilitasi pengeditan video untuk promosi *audio* seperti pada MTV. Sistem pengeditan pada audio atau sistem perekaman hard disk, dimana pelacakan *beat* dilakukan untuk peng-indeks-an *audio* secara otomatis.

Penggunaan sistem ini dapat menangani sinyal akustik sebagai kumpulan ketukan, bukan data gelombang akustik mentah [1]. Solusi sederhana untuk menyelesaikan dua kasus di atas, yaitu dengan memperlambat atau mempercepat pemutaran rekaman *audio*. Namun hal ini dapat menimbulkan suatu permasalahan dimana terjadinya perubahan pada frekuensi sinyal yang ada pada *audio*.

Pada saat seorang mendengarkan musik favorit, tubuh melepaskan zat biokimia alami yang disebut dengan endorphin. Endorphin dapat meminimalisir keluhan rasa sakit serta dapat memicu peningkatan kesehatan jantung. Musik instrumentasi yang tanpa memiliki lirik justru dinilai lebih memberikan efek baik dibandingkan dengan musik yang memiliki lirik. Hal tersebut dikarenakan beberapa musik memiliki lirik yang dapat memicu rasa kesal dan berakibat peningkatan aliran darah, dan hal ini dapat menyebabkan resiko serangan jantung.

Seperti yang diketahui bahwa saat jantung berdetak akan menghasilkan sinyal listrik yang berbeda baik saat relaksasi ataupun saat jantung memompa darah. Sinyal listrik inilah yang digunakan sebagai dasar pengukuran denyut jantung pada ECG yang terdapat pada fasilitas-fasilitas kesehatan.

Penelitian ini menggunakan beberapa jenis musik dengan berbagai variasi tempo musik. Sehingga dapat disimpulkan bahwa variasi tempo musik akan mempengaruhi sinyal listrik yang dihasilkan oleh jantung [2].

Time-scale modification (TSM) adalah proses mempercepat

atau memperlambat sinyal audio tanpa mempengaruhi konten frekuensi [3]. Hasilnya, *audio* pada rekaman audio terdengar dengan kecepatan yang lebih lambat atau lebih cepat tanpa ada distorsi pada pemutarannya.

Demikian pula pada *audio*, yang berubah adalah *timing* bukan *tuning*. Diantara sejumlah teknik TSM, metode *Waveform Similarity and Overlap Add* (WSOLA) telah digunakan secara luas karena kemudahan komputasinya sehingga memungkinkan penerapan secara *real time*.

Ketika pengguna melakukan aktivitas olahraga salah satunya yaitu dengan melakukan senam *aerobic* dimana kecepatan gerakan dapat berpatokan terhadap kecepatan musik yang sudah dihubungkan dengan detak jantung penggunanya. Sehingga dapat terus menjaga kestabilan atau mengembalikan detak jantung penggunanya dalam *range* normal [4].

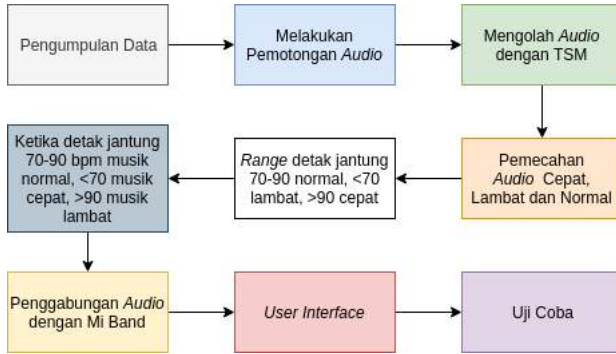
1.2 Permasalahan

1. Cara paling sederhana dalam mengubah durasi audio agar terdengar lebih cepat atau lebih lambat yaitu dengan mengkonversi *sample rate*. Namun hal ini menghasilkan suara yang tidak diinginkan pendengar karena perubahan frekuensi suara pada audio.
2. Cara menyesuaikan *heart rate* dengan audio yang dimainkan ketika detak jantung cepat, lambat dan normal.
3. Cara pemutaran audio dilakukan secara *real time*.

1.3 Tujuan

Penelitian ini bertujuan untuk mempertahankan informasi frekuensi pada rekaman suara yang bisa disesuaikan kecepatan pada pemutarannya sehingga dapat menjadi solusi bagi pendengar yang membutuhkan kecepatan yang lebih cepat atau yang lebih lambat untuk memudahkan pendengar mengikuti ritme *audio* dalam berolahraga dalam satu *audio*.

1.4 Metodologi



Gambar 1.1: Gambaran umum metode penelitian.

1. Pengumpulan Data

Pengumpulan data bertujuan untuk mendapatkan informasi terkait sinyal *audio* dan detak jantung yang nantinya akan digabungkan dan *audio* diolah sesuai dengan kecepatan detak jantung penggunaannya.

2. Melakukan Pemotongan *audio*

Pemotongan *audio* dilakukan untuk membagi satu *audio* menjadi tiga bagian yaitu cepat, lambat dan normal. hal ini dilakukan untuk mempermudah pengolahan *audio* agar delay tidak ada ketika *audio* dinyalakan.

3. Mengolah *audio* dengan *Time Scale Modification* (TSM)

Setelah *audio* dipotong-potong kemudian dilakukan pengolahan dengan TSM pada metode *Waveform Similarity and Overlap Add* (WSOLA), dimana *audio* yang dipotong-potong tersebut diolah kecepatannya menjadi cepat, lambat dan normal. Untuk kecepatan *audio* yang dipercepat menjadi dua kalinya dari *audio* normal dan untuk perlambatan kecepatan *audio* yang diperlambat menjadi setengah kalinya kecepatan *audio* normal. Kemudian melakukan pemrosesan *audio* dengan *load* dan lagu di-play. Proses yang dilakukan dari Pemotong-

an *audio* dan pengolahan *audio* dengan TSM dimasukkan ke dalam memori sehingga cepat diolah dan saat melakukan *play audio* yang nantinya akan digabungkan dengan *Mi Band 2*.

4. Penggabungan *audio* dengan *Mi Band 2*

Setelah *audio* selesai diolah dan berjalan sesuai dengan kecepatan yang telah ditentukan kemudian menggabungkan dengan kecepatan detak jantung pengguna. Ketika detak jantung pengguna kurang dari 60 bpm (*beats per minute*) maka kecepatan *audio* berubah menjadi cepat, jika detak jantung pengguna lebih dari 100 bpm maka kecepatan *audio* berubah menjadi lambat, dan jika detak jantung pengguna normal maka kecepatan *audio* menjadi normal.

5. *User Interface*

Pembuatan *user interface* untuk mempermudah pengguna menggunakan penelitian ini, sehingga aplikasi dapat dipasang pada *mobile phone* pengguna dan dapat digunakan ketika berolahraga.

6. Uji Coba

Uji coba dilakukan untuk mengetahui seberapa jauh hasil yang didapatkan dari penelitian ini, dan apakah sudah sesuai dengan target yang dituju.

1.5 Batasan masalah

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah:

1. Data rekaman *audio* dalam format .WAV
2. Percepatan dan perlambatan *beat rate* dilakukan pada satu *audio* tanpa harus menunggu *audio* selesai terlebih dahulu
3. Perputaran *audio* secara *real time*
4. Penggunaan *user interface* hanya dapat dilakukan di komputer bukan berupa aplikasi *mobile phone*
5. Olahraga yang dilakukan hanya yang berupa bergerak ditempat seperti senam *aerobic*

1.6 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Dasar Teori
Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu terkait dengan *Time Scale Modification*, *heart rate*, dan *real time* perputaran audio.
3. BAB III Perancangan Sistem dan Implementasi
Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah hingga menghasilkan visualisasi. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.
4. BAB IV Pengujian dan Analisa
Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji
5. BAB V Penutup
Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

1.7 Relevansi

Penelitian mengenai *signal processing* merupakan yang sangat dibutuhkan dan dapat dipakai dalam pemenuhan kebutuhan informasi dan komputasi yang semakin beragam saat ini. Dalam tugas akhir ini, dilakukan penelitian mengenai *Time Scale Modification* (TSM) yang disesuaikan dengan *heart rate*.

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 *Time Scale Modification* (TSM)

Algoritma *Time-Scale* (TSM) merupakan metode dimana tingkat kompleksitasnya rendah. Oleh karena itu, metode ini dapat berguna pada aplikasi *real-time*. Semua algoritma *Time scale Modification* (TSM) terdiri dari dua bagian, yaitu langkah analisis dan sintesis. Pada langkah analisis, sinyal suara dibagi menjadi *frame* yang tumpang tindih ini disebut pergeseran waktu analisis (*analysis time shift*).

Tujuan sintesis adalah untuk menyambung kembali *frame* yang telah bergeser secara waktu sintesis bergantung pada faktor skala waktu sesuai dengan pergeseran waktu sintesis (*synthesis time shift*). Perpindahan waktu sintesis bergantung pada faktor skala waktu α (*alpha*).

2.2 Pengaruh Musik Terhadap Jantung

Musik dapat memberikan pengaruh terhadap jantung pendengarnya, dimana musik dapat memicu rasa relaks, mempercepat penyembuhan, meningkatkan fungsi mental dan menciptakan rasa sejahtera. Musik sendiri dapat mempengaruhi fungsi-fungsi fisiologis, seperti respirasi, denyut nadi, dan tekanan darah. Musik dan suara alam dapat meminimalkan persepsi pengguna terhadap suara-suara disekitarnya atau pikiran-pikiran yang membuat cemas dan meningkatkan rasa nyeri, hal tersebut dikarenakan adanya konvergensi yang terjadi antara input sensorik seperti terapi relaksasi suara alam serta kombinasi keduanya dan *output* saraf yang mengatur rasa sakit serta respon stress [5].

Pada saat seorang mendengarkan musik favorit, tubuh melepaskan zat biokimia alami yang disebut dengan endorphin. Endorphin dapat meminimalisir keluhan rasa sakit serta dapat memicu peningkatan kesehatan jantung. Musik instrumentasi yang tanpa

memiliki lirik justru dinilai lebih memberikan efek baik dibandingkan dengan musik yang memiliki lirik. Hal tersebut dikarenakan beberapa musik memiliki lirik yang dapat memicu rasa kesal dan berakibat peningkatan aliran darah, dan hal ini dapat menyebabkan resiko serangan jantung.

Seperti yang diketahui bahwa saat jantung berdetak akan menghasilkan sinyal listrik yang berbeda baik saat relaksasi ataupun saat jantung memompa darah. Sinyal listrik inilah yang digunakan sebagai dasar pengukuran denyut jantung pada ECG yang terdapat pada fasilitas-fasilitas kesehatan. Penelitian ini menggunakan beberapa jenis musik dengan berbagai variasi tempo musik. Sehingga dapat disimpulkan bahwa variasi tempo musik akan mempengaruhi sinyal listrik yang dihasilkan oleh jantung [2].

2.3 Manfaat *Aerobic* Bagi Kesehatan

Selain masalah tulang, semakin bertambahnya usia juga semakin membuat kita bermasalah dengan sendi. Latihan *aerobic* dengan rutin akan menjaga fleksibilitas sendi bagi penggunanya. Jika fleksibilitas sendi kita meningkat maka kita juga bisa terhindar dari berbagai masalah seperti kelainan tulang dan juga radang sendi. Masalah fleksibilitas sendi yang menurun mungkin tidak bisa dilawan sepenuhnya seiring dengan penambahan usia. Tapi paling tidak pengguna bisa mencegah masalah tersebut untuk datang lebih cepat.

Aerobic merupakan salah satu aktivitas olahraga yang dilakukan secara sistematis dengan meningkatkan beban secara bertahap dan terus-menerus dengan menggunakan energi yang berasal dari pembakaran dengan menggunakan okosigen, dan membutuhkan oksigen tanpa menimbulkan kelelahan. Pada latihan olahraga ini-pun memiliki beberapa kriteria didalamnya sebagai takaran latihan, yaitu: frekuensi latihan tiga sampai lima kali setiap minggunya, intensitas latihan 60-80 % dari denyut jantung maksimal, dan durasi latihan 20-60 menit. Latihan olahraga *arobic* sendiri bergantung terhadap ketersediaan oksigen untuk membantu proses pembakaran sumber energi, sehingga jantung, paru-paru, dan pembuluh darah untuk mengangkut oksigen agar proses pembakaran sumber energi dapat berjalan sengan sempurna.

Metabolisme energi pada latihan ini berjalan dengan melakukan pembakaran simpanan lemak, karbohidrat, dan sebagian kecil (kurang dari 5%) dari pemecahan simpanan protein yang terdapat pada tubuh untuk menghasilkan adenosine trifosfat. Proses metabolisme ketiga sumber energi tersebut juga berjalan dengan adanya oksigen yang diperoleh melalui proses pernapasan [5].

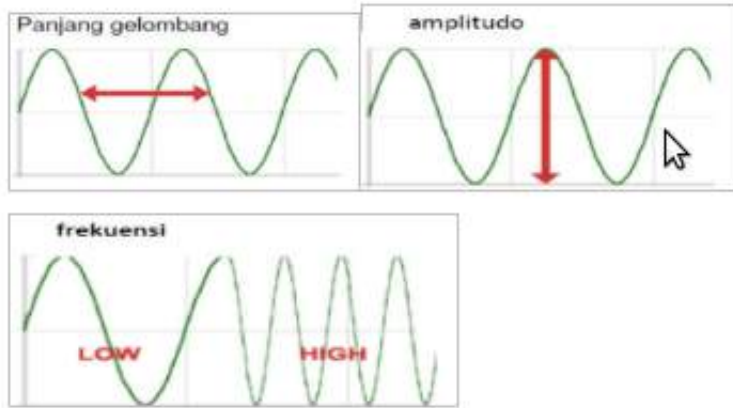
2.4 Audio

Gelombang suara memiliki tiga sifat penting dalam pengolahan *audio*, dimana meliputi panjang gelombang yang dihasilkan dari sebuah benda yang bergetar. Salah satu contohnya yaitu ketika seseorang memainkan alat musik gitar, dimana senar pada gitar yang dipetik, maka akan menimbulkan suatu getaran. Pada getaran ini dapat merambat di udara, atau air, atau material lainnya. Namun, ada tempat dimana satu-satunya tidak dapat merambatkan suara yaitu pada suatu ruangan hampa udara.

Gelombang suara inipun memiliki lembah dan bukit, dimana satu buah lembah dan bukit akan menghasilkan satu siklus atau periode didalamnya. Siklus ini berlangsung berulang-ulang, hingga membawa pada sebuah atau suatu konsep frekuensi. Gelombang suara dapat juga ditunjukkan dalam suatu grafik standar *x versus y* seperti di tunjukan pada gambar 2.1.

Pada gambar tersebut memungkinkan dalam memvisualisasi gelombang dari sudut pandang matematis dimana pada gambar tersebut menghasilkan kurva yang dikenal sebagai bentuk gelombang. Periode gelombang adalah suatu jarak antar titik gelombang dan titik ekuivalensi pada fasa berikutnya. Sedangkan amplitudo itu sendiri merupakan suatu kekuatan atau daya dari gelombang sinyal dan atau daya ekuivalen pada fasa berikutnya. Kemudian adanya tinggi gelombang yang bisa dilihat sebagai suatu grafik dalam menggambarkan proses yang terjadi.

Gelombang yang lebih tinggi mempresentasikan sebagai volume yang lebih tinggi, sehingga dapat disebut sebagai amplifier untuk perangkat yang menambah amplitude. Frekuensi sendiri adalah jumlah getaran yang terjadi dalam waktu satu detik. Frekuensi diukur dalam satuan *hertz* (Hz) atau siklus per detik. Ketika suatu getaran gelombang suara semakin cepat, maka menimbulkan freku-



Gambar 2.1: Gelombang suara

ensi yang semakin tinggi. Dimana suara lebih tinggi diinterpretasikan sebagai jalur yang lebih tinggi. Seperti pada persamaan [1].

$$F = \frac{1}{T}$$

Dimana dari persamaan diatas apabila periode suatu sinyal suara (T) diperbesar ataupun diperlambat maka frekuensi dari suara (F) yang dihasilkan akan berubah menjadi kecil. Dan begitu juga sebaliknya, apabila nilai dari periode sinyal suara (T) diperkecil atau dipercepat maka frekuensi dari suara (F) yang dihasilkan berubah menjadi besar.

Telinga manusia dapat mendengar atau menerima bunyi antara 20 Hz hingga kHz (20.000 Hz) sesuai batasan sinyal audio. Nilai 20 Hz sebagai frekuensi suara terendah yang dapat di dengar oleh manusia, sedangkan 20 kHz merupakan frekuensi tertinggi yang dapat didengar oleh manusia. Gelombang suara juga memiliki variasi didalamnya, sebagaimana variasi tekanan media perantara seperti udara. Suara diciptakan oleh getaran dari suatu obyek, dengan adanya getaran tersebut yang menyebabkan gendang telinga manusia mampu menangkap getaran, yang kemudian oleh otak dianggap sebagai suara [3].

2.5 Time Stretching

Time Scale Modification (TSM) atau *time stretching* adalah proses memperlambat atau mempercepat sinyal *audio* tanpa mengubah *pitch*. beberapa contoh dari pengaplikasian dari skala waktu ini yaitu dalam mempelajari bahasa asing dengan metode *listening*, dimana *audio*-nya dapat dirubah kecepatannya untuk mempermudah pengguna dalam belajar. Skala waktu dilakukan dengan membagi sinyal menjadi *frame* tetap menjadi tumpang tindih. *Frame* tumpang tindih ini kemudian bergeser sesuai dengan tujuan keseluruhan (mempercepat atau memperlambat) dan dikombinasikan untuk memberikan output direkonstruksi.

Penerapan *Time Scale Modification* (TSM) misalnya pada *audio* dimana *remixing* untuk mengatur kecepatan pemutaran rekaman yang ada sehingga mereka dapat dimainkan secara bersamaan pada tempo yang sama. Bidang lain dari pengaplikasian ini adalah penyesuaian aliran *audio* dalam video klip. Misalnya, saat membuat video gerakan lambat, TSM juga dapat digunakan untuk menyinkronkan materi *audio* dengan konten visual.

Time Scale Modification (TSM) sendiri memiliki beberapa metode didalamnya, yaitu :

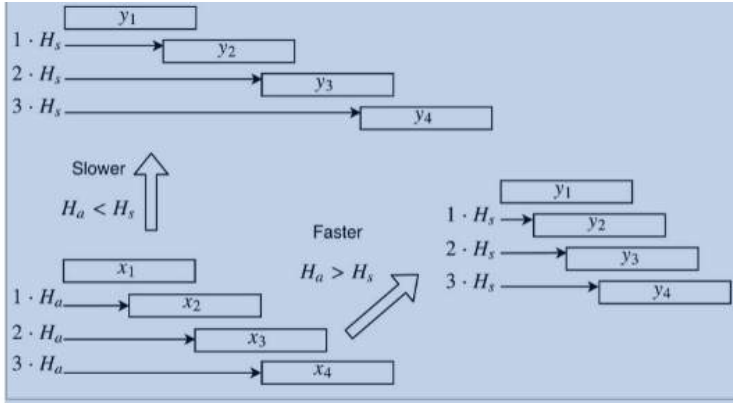
1. *Overlap and Add* (OLA)

Merupakan teknik yang banyak digunakan dalam penyaringan digital. Dalam implementasinya, sistem dipaksa untuk menjadi invarian waktu dan teknik sintesis filter konvensional digunakan untuk merancang blok filter [6].

Seperti pada gambar 2.2, OLA merupakan contoh prosedur TSM domain waktu dimana modifikasi pada kerangka analisis diterapkan secara murni dalam domain waktu. Secara umum, prosedur dari TSM domain waktu tidak hanya efisien tetapi juga menjaga timbre dari sinyal input ke tingkat yang tinggi.

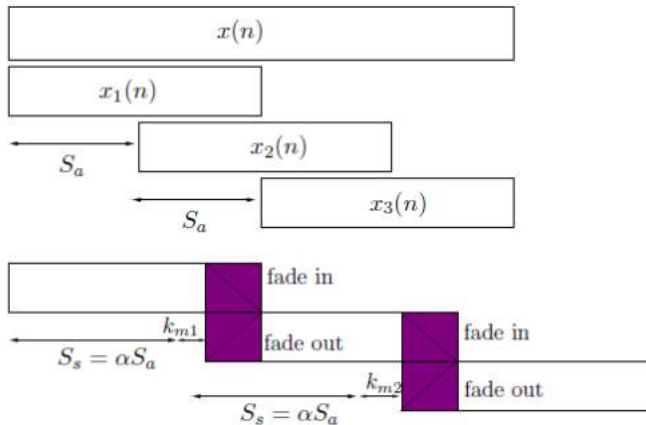
2. *Synchronous Overlap and Add* (SOLA)

Cara kerja dari metode ini yaitu dengan memotong data *audio* menjadi urutan pendek antara puluhan hingga ratusan *miliseconds* dari masing-masing data dan kemudian menggabungkan



Gambar 2.2: *Time stretching : overlap and add*

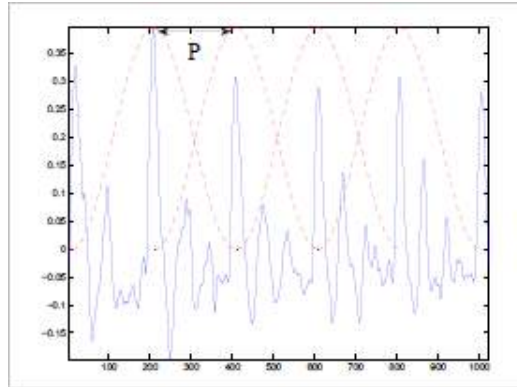
kembali secara bersama-sama dengan jumlah data *audio* yang cocok baik dilewati atau sebagian diulang di antara urutan ini, untuk mencapai waktu pemutaran yang lebih pendek atau lebih lama dari aslinya [7].



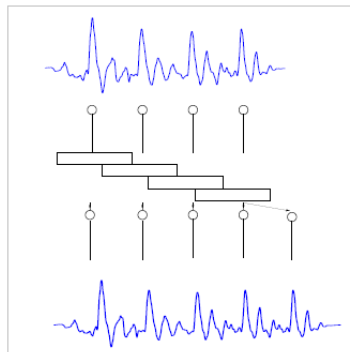
Gambar 2.3: *Time stretching : synchronous overlap and add*

3. Pitch Synchronous Overlap and Add (PSOLA)

Teknik yang banyak digunakan untuk mengubah *audio* dan / atau skala waktu dari sinyal *audio* itu sendiri. Metode ini dapat diimplementasikan dengan cukup efisien untuk dijalankan pada sistem yang tertanam. Namun tergantung pada jenis sinyalnya [8].



Gambar 2.4: *PSOLA : pitch analysis and block windows*

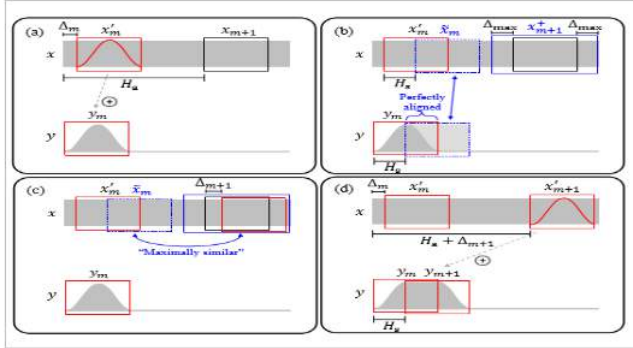


Gambar 2.5: *Depiction of the PSOLA algorithm*

4. *Waveform Similarity Overlap-Add (WSOLA)*

Dimana algoritmanya terjadi tumpang tindih yang disinkron-

an tumpang tindih berbasis kesamaan-kesamaan menghasilkan output suara berkualitas tinggi, secara algoritmik dan efisien secara komputasi dan kuat, serta memungkinkan pemrosesan *online* (secara *real time*) dengan faktor penskalaan waktu yang bebas namun dapat ditentukan dalam waktu yang bervariasi [9].



Gambar 2.6: Prinsip kesamaan bentuk gelombang *overlap-add* (WSOLA)

2.5.1 *Waveform Similarity Overlap-Add* (WSOLA)

WSOLA adalah teknik pemrosesan sinyal digital yang digunakan untuk pengolahan sintesis suara. WSOLA dapat digunakan untuk memodifikasi durasi dari sinyal suara. Algoritma WSOLA bekerja dengan membagi bentuk gelombang suara pada bagian *overlap*.

Untuk mengubah frekuensi sinyal, segmen dipindahkan ke segmen yang lebih dekat. Untuk mengubah *time scale*, segmen diulang beberapa kali dengan cara dieliminasi. Segmen tersebut kemudian digabungkan dengan menggunakan teknik *time scale*.

WSOLA adalah sebuah konsep *waveform similarity* diusulkan dengan tujuan untuk memecahkan masalah modifikasi skala suara. WSOLA bekerja dalam konteks *Short Time Fourier Transform* (STFT). Algoritma WSOLA menghasilkan *output* kualitas suara

bagus dengan komputasi yang efisien secara algoritma sehingga memungkinkan pengolahan suara secara *real time* dengan faktor skala waktu yang telah ditentukan.

Beberapa variabel yang terlibat pada penelitian ini adalah:

1. *Current Factor* adalah besaran faktor *pitch* sekarang
2. *Sample rate* adalah banyaknya jumlah *sample* (bentuk frekuensi) yang di ambil dalam satuan waktu (*milisecond*) dari sinyal yang diterima dalam bentuk terus-menerus (*Continuous Signal*) menjadi sinyal yang terpisah (*Discrate Signal*) atau dalam bahasa sederhana adalah batas frekuensi yang dapat dikirim per-*milisecondnya*.
3. Panjang *window* atau *windowing* berguna untuk menganalisa suatu sinyal yang panjang dengan cara mengambil suatu bagian yang mewakilinya (daftar pustaka). Sehingga dengan *windowing* dapat menemukan posisi *overlapping* terbaik.

2.5.2 Proses *Sampling*

Sinyal suara merupakan sinyal yang tidak terbatas dalam domain waktu (*infinite time interval*). Suara alat musik dan manusia menghasilkan sinyal berupa sinyal kontinyu. Dalam pemrosesan dengan metode WSOLA dimana sinyal suara harus dibentuk dalam potongan-potongan waktu yang terbatas *finite time interval*. Hal itu dikarenakan sinyal yang ada dipotong-potong dalam slot-slot interval dalam waktu tertentu.

2.6 Format File .WAV

Penelitian ini menggunakan format file .wav. *Waveform audio format* (WAV) adalah standar format berkas audio yang dikembangkan oleh Microsoft dan IBM. Hal ini telah menjadi salah satu format digital yang paling banyak didukung oleh file audio pada PC karena popularitas Windows dan hampir setiap program modern yang dapat membuka dan atau menyimpan audio digital mendukung format file tersebut.

Format file wav terdiri dari 3 bagian, yaitu *main chunk*, format *chunk*, dan data *chunk*. Sinyal suara yang dipresentasikan file wav dalam bentuk diskrit, berupa deret bilangan yang mempresen-

tasikan amplitudo dalam domain waktu. Pada bagian file *header* terdapat informasi tentang file wav tersebut, diantaranya menyatakan nilai *sample rate*, jumlah *channel*, dan bit per *sample*. Dari keterangan pada file header tersebut dapat diketahui berapa *sample* yang dicuplik dari sinyal analog tiap detik [10].

1. Bagian *Main Chunk*

ChunkID : berisi kata "RIFF" dalam format ASCII.

ChunkSize : berisi informasi ukuran *chunk*.

Format : berisi kata "WAVE"

2. Bagian Format *Chunk*

SubChunk1ID : berisi kata "fmt"

SubChunk1Size : berisi informasi ukuran *subchunk* 18

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

4. Bagian Data *Chunk*

SubChunk2ID : berisi kata "data".

SubChunk2Size : berisi informasi ukuran *subchunk* 2. $SubChunk2Size = NumSamples \times NumChannels \times BitsPerSample / 8$. $NumSamples = (DataByte / NumChannels) / BitsPerSample$. Data : Data suara aktual dalam byte, mempresentasikan *amplitudo* tiap sampel dari sinyal.

2.7 *Photoplethysmography (PPG)*

PPG merupakan sebuah metode yang digunakan untuk mengetahui kondisi sistem dari kardiovaskular dengan mengukur perubahan volume darah pada jaringan kulit. Dimana dalam penerapannya yaitu metode ini menggunakan sensor optik untuk menangkap sinyal elektrik yang berasal dari sumber cahaya yang terpantul karena perubahan aliran darah selama aktivitas dari jantung. Dimana sinyal yang dihasilkan dari metode ini telah banyak dikembangkan dalam *smart watch (Wearable Device)* untuk pengukuran detak jantung [11].

2.7.1 Proses *Wearable Device*

Pada saat ini banyak sekali *Wearable Device* yang bisa digunakan untuk kebutuhan sehari-hari seperti notifikasi pengingat dan juga untuk berolahraga, dimana fitur yang ditawarkan salah satunya yaitu untuk menghitung detak jantung, langkah, alarm, kalori yang terbakar dalam satu hari dan lain-lain. Salah satu contoh dari *Wearable Device* adalah *Smart Watch Xiaomi Mi Band 2*.

Gambar 2.3 merupakan *Wearable Device Xiaomi Mi Band 2* yang dilengkapi dengan fitur pendeteksi detak jantung menggunakan PPG, pendeteksi langkah layar OLED (*Organic Light-Emitting-Diode*) dan tombol sentuh. Mi Band 2 menggunakan koneksi *Bluetooth* dan tombol sentuh. Mi Band 2 menggunakan koneksi *Bluetooth LE* untuk berkomunikasi dengan host.



Gambar 2.7: Mi Band

2.8 notasi Musik

Musik adalah suatu susunan dimana tinggi-rendah nada yang dimainkan dalam waktu. Hal ini dapat dilihat dari notasi musik yang menggambarkan besarnya waktu dalam arah horizontal dan tinggi-rendah nada dalam arah vertikal.

Ritme sendiri ditentukan oleh panjang atau lamanya waktu dari suatu bunyi. Sedangkan panjang ataupun pendeknya suatu bunyi digambarkan dengan simbol-simbol yang disebut dengan not

(*pitch*). Panjang pendeknya diam juga digambarkan dengan simbol-simbol yang disebut dengan tanda istirahat (*rest*) [12].

| NOT | HARGA | TANDA ISTIRAHAT |
|---|----------------|---|
|  | Penuh |  |
|  | Setengah |  |
|  | Seperempat |  |
|  | Seperdelapan |  |
|  | Seperenambelas |  |

Gambar 2.8: Simbol not

2.8.1 *Beat* atau Ketukan

Beat atau irama pada *audio* merupakan bagian dari struktur ritme, seperti 1-2-3-4 yang biasanya dihitung ketika memainkan ataupun mendengar sebuah lagu. *Beat* membagi sebuah frasa menjadi bagian-bagian yang seragam, dimana satu *beat* yang nantinya dapat dibagi lagi menjadi beberapa bagian kecil bila dibutuhkan. Contohnya yaitu saat adanya irama 4/4, satu *beat* sama dengan satu ketukan, sehingga satu bar berirama berisi 4 *beat*.

| Note | Name | Beats |
|---|----------------|--------------------|
|  | Whole note | 4 beats |
|  | Half note | 2 beats |
|  | Quarter note | 1 beat |
|  | Eighth note | $\frac{1}{2}$ beat |
|  | Sixteenth note | $\frac{1}{4}$ beat |

Gambar 2.9: Nilai ketukan

2.8.2 Tempo

Tempo adalah penandaan pada seberapa cepat lagu dimainkan. Tempo merupakan hitungan konstan yang memberi irama pada sebuah musik sehingga dapat menentukan cepat lambatnya musik dimainkan. Dimana satuan atau nilai dari tempo sendiri yaitu BPM (*Beat per Minute*). Sehingga ketika tempo dari suatu musik 100 dalam satu menit maka terdapat 100 ketukan dalam musik tersebut.

2.9 Bluetooth

Bluetooth adalah sebuah perangkat komunikasi tanpa menggunakan (*wireless*) yang beroperasi pada pita frekuensi (2400-2483,5) MHz yang dikenal sebagai pita ISM (*Industrial Scientific and Medical*) *multipoint*. *Bluetooth* mengkonsumsi daya rendah dengan tiga kelas output power yang berbeda, diantaranya yaitu : kelas 100 mW, dapat menambah radius jangkauan dari 10 meter ke 100 meter. Fasilitas layanan masing 1 MHz, menggunakan salah satu metode *spread spectrum* yaitu FHSS (*Frequency Hopping Spread Spectrum*). Saat ini, *bluetooth* didukung oleh banyak perusahaan terkenal yang bergerak dibidang elektronika dan telekomunikasi dan tergabung dalam *Special Interest Group* (SIG) [13].

Komunikasi antar perangkat tanpa menggunakan kabel yang dikenal dengan teknologi *wireless*, dimana tujuan dari perangkat ini untuk menggantikan peran kabel, khususnya untuk jarak yang dekat. *Bluetooth* yang dimaksudkan bukan hanya menggantikan peran kabel dalam melakukan pertukaran informasi tetapi juga mampu menawarkan fitur yang baik dengan biaya yang relatif rendah, *interoperability* yang menjanjikan dan mudah dalam pengoperasiannya. Perangkat keras *bluetooth* dapat berupa card atau usb adapter yang dipasang pada (ke) komputer, printer, perangkat mobile, headset, kamera dan perangkat lainnya.

Bluetooth mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara host-host *bluetooth* namun jarak jangkauannya terbatas dan tergantung pada power *output*-nya yaitu sekitar 10 meter untuk power *output* 1 mW.

2.9.1 Penggunaan Frekuensi pada *Bluetooth*

Bluetooth merupakan *wireless* standar dengan jangkauan terbatas menggunakan gelombang radio dengan pita frekuensi yaitu 2,4 GHz (2400-2483,5 MHz). Sistem radio *bluetooth* mengatur penggunaan pita frekuensi ini, yang digunakan oleh *transmitter* perangkat *bluetooth*.

2.9.2 Komunikasi Data Menggunakan *Bluetooth*

Kanal komunikasi *bluetooth* dapat digunakan untuk komunikasi data dan suara, dimana dengan dua jenis layanan yaitu sinkron dan asinkron.

2.9.2.1 Transmisi Asinkron (*Asynchronous Connection Less (ACL)*)

Pada transmisi ini dimana setiap kali pengiriman data dilakukan perkaraktanya. Antara satu karakter dengan karakter lainnya tidak ada waktu antara yang tetap. Karakter dapat dikirimkan sekaligus ataupun beberapa karakter kemudian berhenti untuk waktu tidak tertentu, kemudian dikirimkan sisanya. Akibatnya, penerima harus melakukan sinkronisasi setiap kali pengiriman bit data, apakah data yang dikirimkan diterima dengan benar. Sehingga penerima mengetahui bit pertama yang dimulai dari sinyal data, dengan

cara memberikan suatu pulsa yang disebut *start pulse* pada awal tiap karakter.

2.9.2.2 Transmisi Sinkron (*Synchronous Connection Oriented (SCO)*)

Pada transmisi ini, dimana blok atau frame data dikirimkan setiap elemen 8 bit secara kontinu tanpa delay. Panjang tiap blok sama. Waktu antara akhir dari bit terakhir dari suatu karakter dan awal bit pertama karakter berikutnya harus nol atau kelipatan dari waktu satu karakter. Untuk mencapai sinkronisasi ini pengirim harus mengirim karakter khusus dan penerima harus mengenalinya.

Transmisi sinkron digunakan untuk pengiriman audio atau suara, dengan transmisi kecepatan tinggi, yang mentransmisikan satu blok data. Dalam sistem ini baik pengirim dan penerima bekerja secara bersamaan dan sinkronisasi dilakukan setiap sekian ribu bit data.

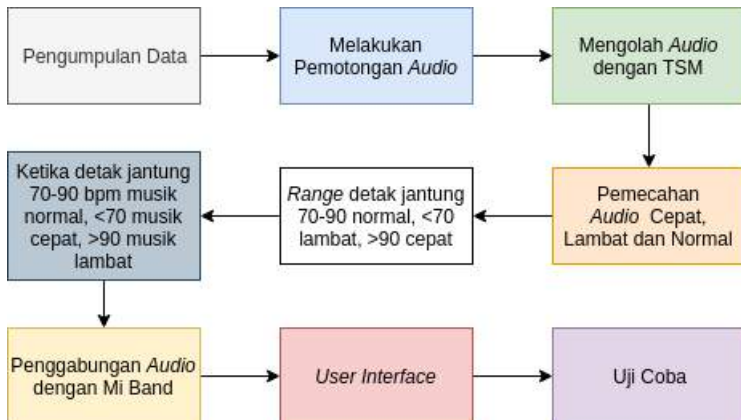
Halaman ini sengaja dikosongkan

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan.

3.1 Desain Sistem



Gambar 3.1: Metodologi sistem secara umum.

Dalam mengimplementasikan penelitian ini, diperlukannya metodologi yang mendukung dalam skema pengerjaannya seperti pada gambar 3.1. Perlunya pengumpulan data pada penelitian ini yaitu menitik beratkan pada data detak jantung penggunanya dikarenakan detak jantung yang akan mengatur *beat* daripada *audio* yang diproses. Data yang dikumpulkanpun bukan hanya sekedar data berupa detak jantung pengguna namun ada beberapa tambahan pengumpulan data yaitu *delay* dari *audio* yang diolah dan *delay Mi Band 2*.

Perlunya pemotongan *audio* pada proses ini untuk membagi *audio* menjadi tiga bagian. Hal ini dilakukan agar mudah diolah ketika dimainkan, pembagian *audio* ini menjadi normal, dipercepat, dan diperlambat. Pemotongan *audio* dalam satu durasi lagu dibagi menjadi satu per-empat bagian-bagian.

Kemudian dilakukannya pengolahan *audio* dengan menggunakan *Time Scale Modification* (TSM) dengan menggunakan salah satu metode dari TSM tersendiri yaitu *Waveform Similitary and Overlap Add*(WSOLA), dimana dengan pengolahan *audio* menggunakan *Time Scale Modification* yang mengatur kecepatan daripada *audio* yang dibagi menjadi tiga bagian tersebut.

Pembagian *range* pada jantung yaitu dengan nilai 70-90 BPM dikatakan normal, kurang dari 70 BPM dikatakan lambat, dan lebih dari 90 BPM dikatakan cepat. *Range* dari detak jantung ini yang nantinya akan mengontrol daripada kecepatan *audio*.

Ketika detak jantung pengguna memiliki nilai atau *range* 70-90 BPM maka *audio* yang diputar kecepatannya adalah normal. Namun, ketika detak jantung pengguna memiliki nilai atau *range* kurang dari 70 BPM maka kecepatan *audio* berubah menjadi lambat. Ketika detak jantung pengguna memiliki nilai atau *range* diatas 90 BPM maka *audio* akan berubah menjadi lebih cepat.

Setelah terjadinya pengolahan *audio* menggunakan *Time Scale Modification* *audio* yang tadinya hanya dengan kecepatan normal dalam satu durasi lagu, maka akan terpecah-pecah menjadi beberapa kecepatan *audio* (dibagi menjadi seperempat bagian atau potongan-potongan lagu) baik kecepatan normal, dipercepat dan diperlambat. Pengguna juga perlu dalam mengaktifkan *Mi Band 2* agar bisa mengatur kecepatan *audio*-nya sesuai dengan kecepatan detak jantungnya. *Audio* yang sebelumnya terpecah dan dibagi-bagi disatukan kembali dan diolah menjadi satu lagu penuh dengan kecepatan yang berbeda sesuai dengan detak jantung.

Untuk mempermudah pengguna dalam penggunaan penelitian ini diperlukannya *User Interface*. Dimana didalamnya terdapat beberapa *button* yang membantu pengguna dalam mengoperasikan *Mi Band 2* dengan *audio*. Setelah semua dapat dijalankan maka dapat melakukan pengujian dan pengujian dilakukan dengan mengguna-

kan beberapa komputer dengan spesifikasi komputer yang berbeda.

Pada sistem yang dibangun, digunakan sebuah *Wearable Device Xiaomi Mi Band 2* guna mendapatkan nilai dari *heart rate*. Data tersebut kemudian diolah untuk menentukan kontrol kecepatan dari musik yang diputar oleh peengguna. Guna merealisasikan penelitian tugas akhir ini, dibuat sebuah sistem yang menggambarkan konsep dan kerja sistem.



Gambar 3.2: Gambaran umum kerja sistem.

Pada Gambar 3.2 diuraikan secara singkat yaitu terdiri dari

dua proses terdiri dari Generasi Suara dan Pembangunan Beat Suara. Dimana pada Generasi Suara, original dari sinyal *audio* diolah terlebih dahulu untuk dipotong-potong menjadi tiga bagian yaitu cepat, lambat dan normal. Kemudian oleh *Time Scale Modification* (TSM) dengan menggunakan metode *Waveform Similitary and Overlap Add Method* (WSOLA) mengolah *audio* cepat dan lambat, dari hasil pemrosesan pertama kali ini terjadi *delay* dikarenakan TSM sedang mengolah *audio*. Kemudian dari proses pemotongan tersebut TSM menghasilkan potongan-potongan *audio* yang terdiri dari *audio* normal, cepat, dan lambat.

Proses dan hasil olahan *audio* dimasukan ke dalam memori. Hal ini bertujuan untuk mempermudah pengolahan *audio* ketika *audio* akan di *load* sehingga tidak adanya *delay* atau mengurangi *delay* atau dapat dikatakan *delay* mendekati angka nol. Sehingga ketika akan digabungkan dengan *Mi Band 2 audio* akan terdengar berjalan dengan normal seperti *audio* pada umumnya, hanya bedanya dihubungkan dengan kecepatan dari detak jantung pengguna.

Pada proses kedua yaitu Pembangunan *Beat* Suara, dimana pengguna terlebih dahulu mengecek *heart rate* sebelum melakukan olahraga seperti lari dengan menggunakan *Mi Band 2*. Hal ini dilakukan guna untuk mengetahui kondisi jantung pengguna sebelum menjalani olahraga. Kemudian alat mendeteksi *heart rate* selama satu menit. Namun pada pengambilan data *heart rate* pengguna dilakukan setiap detik untuk terus mengetahui kondisi dari jantung penggunanya, hal ini bertujuan agar detak jantung selalu stabil atau normal.

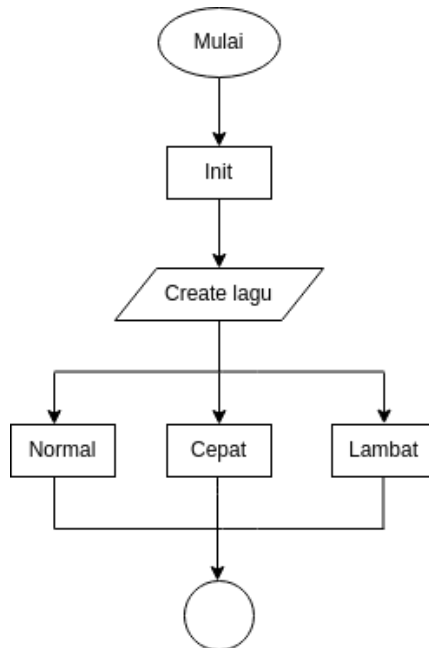
Ketika pengguna mulai berolahraga dan terjadi kenaikan atau penurunan *heart rate* akan muncul notifikasi dimana *audio* mulai untuk dipercepat atau diperlambat sesuai dengan kecepatan *heart rate* sehingga *heart rate* kembali dalam kecepatan normal atau sekitar 70-90 BPM (*beat per minute*) dan *audio* juga akan kembali normal. Ketika *heart rate* bernilai kurang dari 70 BPM (*beat per minute*) maka akan mengambil file *audio* dengan kecepatan yang dipercepat, sedangkan ketika *heart rate* bernilai diatas 90 BPM (*beat per minute*) maka akan mengambil file *audio* dengan kecepatan diperlambat, dan ketika *heart rate* normal maka akan mengambil

file *audio* dengan kecepatan normal, dan dimainkan *audionya*.

Hal ini terus dilakukan sampai satu *audio* selesai (satu lagu selesai). Pemutaran *audio* dipercepat, diperlambat, maupun normal dilakukan dalam satu pemutaran *audio* tanpa harus menunggu *audio* selesai terlebih dahulu untuk mengganti *beat rate audio* dan dilakukan secara *real time*.

3.1.1 Pengolahan *Audio*

Pengolahan *audio* dilakukan untuk mempermudah pemrosesan dalam pemanggilan lagu baik dipercepat, diperlambat dan normal. Sehingga meminimalisir adanya *delay* ketika *audio* dimainkan.

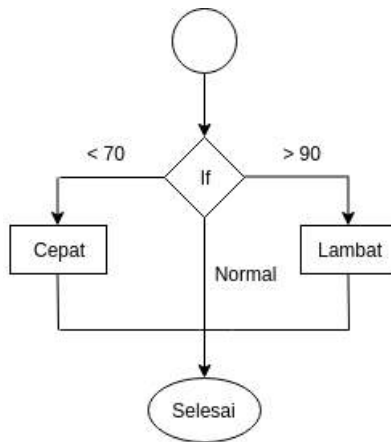


Gambar 3.3: Skema alur kerja pengolahan musik.

Pada Gambar 3.3 sebelum pengolahan *audio* dilakukan terlebih dahulu yaitu meng-init *audio* untuk membagi *audio* menjadi tiga bagian yaitu cepat, lambat dan normal. Hal ini dilakukan untuk

menghilangkan *delay audio* ketika dimainkan. *delay* pada *audio* hanya terjadi pada saat meng-init *audio* pada proses ini, hal ini terjadi dikarenakan TSM melakukan pembacaan *audio* yang sudah dibagi menjadi tiga bagian dan diolah sesuai kecepatan yang diperlukan sampai pemrosesan *audio* selesai.

3.1.2 Heart Rate



Gambar 3.4: Skema alur kerja *heart rate*.

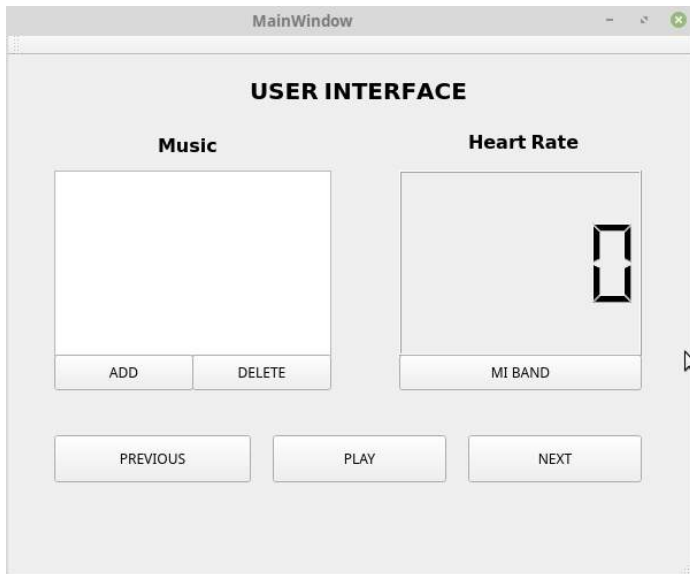
Sebelum memainkan *audio* terlebih dahulu *Mi Band 2* dihubungkan dengan perangkat *Bluetooth* pada komputer. Seperti pada gambar 3.4, *Mi Band 2* akan mendeteksi detak jantung pengguna selama satu menit dan akan menghasilkan *output* detak jantung berupa angka. Pengambilan data detak jantung ini dilakukan setiap detik setelah pengecekan terlebih dahulu selama satu menit.

Hal ini dilakukan agar tidak terjadinya kesalahan dalam perhitungan detak jantung ketika disambungkan dengan *audio*. Ketika *audio* dinyalakan dan disambungkan dengan detak jantung pengguna, kecepatan *audio* akan disesuaikan dengan detak jantung pengguna. Ketika detak jantung dibawah 70 BPM maka *audio* akan diganti kecepatannya menjadi cepat sampai detak jantung kembali normal dan kecepatan *audio* menjadi normal kembali. Dan ketika detak jantung terlalu cepat dapat dikatakan diatas 90 BPM maka

audio akan berubah menjadi lambat sampai detak jantung kembali normal dan *audio* kembali dengan kecepatan normal. Menurut *American Heart Association*, detak jantung pada orang dewasa yang lebih tua dan manula yaitu 60-100 BPM. Namun dikarenakan untuk mendapatkan detak jantung pengguna dibawah 60 BPM dalam aktivitas sehari-hari sangatlah sulit, karena pada normalnya detak jantung dibawah 60 BPM adalah saat seseorang sedang beristirahat (tidur) maka range detak jantung pengguna dirubah nilainya menjadi 70-100 BPM.

3.1.3 Cara Kerja *User Interface*(UI)

Pada *User interface* (UI) terdiri dari beberapa *button* untuk mengontrol serta mengolah *audio* dan *Mi Band 2*. Pengguna juga dapat memilih beberapa *audio* yang disediakan pada aplikasi. Pada aplikasi ini juga menampilkan detak jantung pengguna dan musik yang dimainkan seperti tertera pada Gambar 3.5.



Gambar 3.5: Tampilan desain UI pada aplikasi.

Pengguna dapat melihat kondisi bpm detak jantungnya pada

smart watch Mi Band 2 atau pada aplikasi yang ada di komputer. Dengan adanya tampilan *Heart Rate* ini untuk mempermudah pengguna dalam memonitoring detak jantungnya setiap detiknya. Tampilan *audio* untuk memberitahu pengguna bahwa *audio* yang sedang diputar sesuai dengan *audio* yang dipilihnya dan sudah tersedia di aplikasi. Pada bagian *audio* terdapat dua *button* yang terdiri dari *button* "ADD" dan "DELETE". *Button* "ADD" untuk menambahkan daftar *audio* yang akan didengar oleh pengguna selama berolahraga dan *Button* "DELETE" untuk menghapus bagian *audio* yang tidak diinginkan pengguna. *Audio* yang digunakan atau yang sudah di-Add akan tampil sebagai menu musik pada *screen Music*. Pada bagian detak jantung terdapat *button* "MI BAND" dimana untuk mengkoneksikan atau menyambungkan *user interface* dengan *Mi Band 2* sehingga nantinya detak jantung akan ditampilkan pada *screen "Heart Rate"*. Kemudian terdapat *button* "PREVIOUS" yang digunakan untuk kembali ke *audio* sebelumnya, *buton* "PLAY" untuk memutar *audio* dan pada *button* ini ketika di klik akan otomatis berubah menjadi *button* "STOP", dan *button* "NEXT" untuk ke menu *audio* selanjutnya.

Tabel 3.1: Parameter kecepatan *audio* berdasarkan detak jantung.

| <i>Heart Rate</i> (bpm) | <i>Audio</i> |
|-------------------------|--------------|
| $\geq 60-100$ (bpm) | Normal |
| >100 (bpm) | Lambat |
| <60 (bpm) | Cepat |

Guna informasi notifikasi untuk batas kecepatan *audio* sesuai dengan detak jantung pengguna yaitu dinyatakan detak jantung normal pada kecepatan 60-100 (bpm) dan kecepatan *audio* menjadi normal, perintah pada kecepatan *audio* menjadi dipercepat ketika detak jantung mencapai kecepatan diatas 60 (bpm) dan perintah pada kecepatan *audio* menjadi diperlambat ketika detak jantung mencapai kecepatan diatas 100 (bpm) seperti pada gambar 3.6.

3.1.3.1 *Folder* Daftar Lagu

Pemilihan *audio* telah disediakan dalam *folder* yang ada pada komputer, sehingga pengguna dapat memilih *audio* yang di ingink-

an. Pengguna juga dapat menambahkan atau menghapus *audio* yang ada pada *folder*, namun hal ini hanya berpengaruh dalam *folder* ini saja. Tidak berpengaruh pada *user interface*-nya dikarenakan ada *folder* khusus dalam mengolah *audio*-nya sebelum dimainkan.



Gambar 3.6: Folder daftar lagu.

3.1.3.2 List Lagu

Terdapat beberapa lagu di dalam *folder* daftar lagu. Pengguna dapat merubah isi daripada *folder* tersebut sesuai dengan keinginan penggunanya. Pada *folder* inilah pengguna memilih lagu yang akan diolah atau dimasukkan ke *user interface*. Nantinya setelah dipilih lagu pada *folder* ini, kemudian lagu diolah pada *user interface* dan dimasukkan ke *folder play list* milik *user interface*.

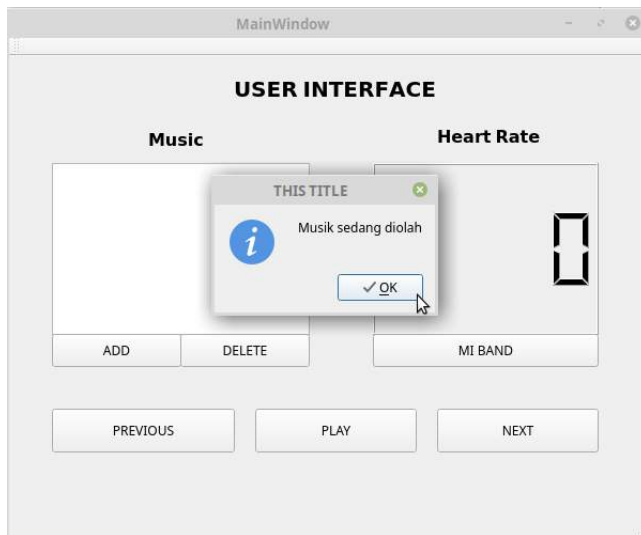
3.1.3.3 Pop Up Pengolahan Musik

Ketika pengguna sudah memilih lagu yang ingin dimasukkan kedalam *user interface*, kemudian lagu akan diolah kecepatannya terlebih dahulu dan masuk ke dalam *folder play list user interface*. Sehingga pengolahan ini akan memakan cukup waktu tergantung panjang lagu, sehingga perlu adanya *pop up* untuk memberitahu pengguna bahwa sedang terjadi proses dalam pengolahan lagu. Akan tetapi ketika lagu yang sebelumnya sudah pernah diolah atau di-ADD kembali maka tidak diperlakukan lagi untuk mengolah musiknya seperti pada awal di-ADD namun *pop up* akan tetap muncul hanya sebagai pemberitahuan bahwa musik sedang diolah untuk di-

| Name | Size | Modified |
|----------------------|---------|----------|
| A Thousand Miles.wav | 50,5 MB | Wed |
| Forever yours.wav | 36,9 MB | Wed |
| I'm not sorry.wav | 37,7 MB | Wed |
| Lily.wav | 37,6 MB | Wed |
| Lucky.wav | 35,6 MB | Wed |
| Mama.wav | 33,5 MB | Wed |
| On my way.wav | 38,3 MB | Wed |
| Q&A.wav | 38,9 MB | 16 Mar |
| So am i.wav | 37,9 MB | Wed |
| Tolong.wav | 39,6 MB | Wed |

Gambar 3.7: *List lagu.*

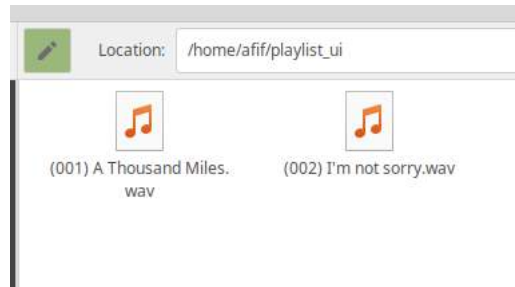
masukkan ke dalam *screen play list user interface.*



Gambar 3.8: *Pop up pengolahan musik.*

3.1.3.4 *Play List* User Interface

Ketika lagu telah diolah oleh *user interface* maka lagu tersebut akan masuk ke dalam *folder play list user interface* hal ini dilakukan untuk memisahkan lagu yang telah diolah oleh *user interface* dengan yang belum diolah yaitu di *folder* musik biasa. Kemudian *list* lagu yang sudah siap dimainkan akan tampil pada *Screen Music*.



Gambar 3.9: *Folder play list user interface.*

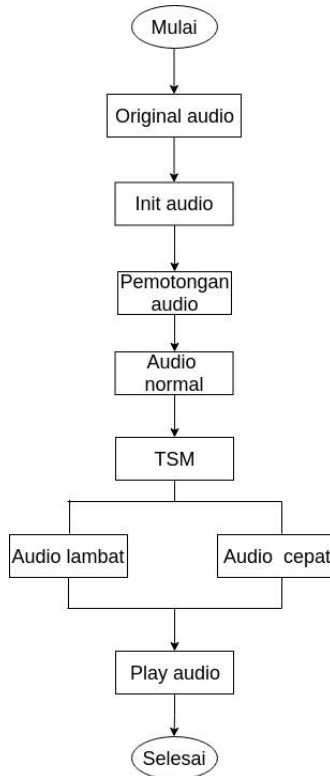
Ketika *play list* pada *screen* sudah muncul maka lagu tersebut *audio*-nya sudah diolah dan dapat diputar oleh pengguna. *Pop up* ini akan terus muncul hanya pada saat pengguna ingin membuat *play list* lagu atau menambahkan *play list* pada *screen music*. Setiap *audio* yang sudah diolah dan masuk pada *screen music* maka masuk ke *folder playlist ui*. Terpisahnya pengolahan *audio* ini untuk meminimalisir kesalahan yang terjadi ketika akan menghapus *audio* yang tidak diinginkan dari dalam *folder playlist ui* bukan pada *user interface*.

3.2 Implementasi Sistem

Secara keseluruhan, alur implementasi sistem ini terbagi menjadi tiga. Pertama yaitu membagi *audio* menjadi tiga bagian kemudian diolah kecepatannya menjadi cepat, lambat, dan normal. Kedua mendeteksi detak jantung pengguna untuk dikoneksikan dengan *audio* dalam mengatur kecepatan *audio*-nya. Ketiga pembuatan *user interface* (UI) untuk mempermudah pengguna dalam menggunakan penelitian ini sehingga perlu adanya aplikasi.

3.2.1 Pembagian *Audio* Menjadi Tiga Bagian

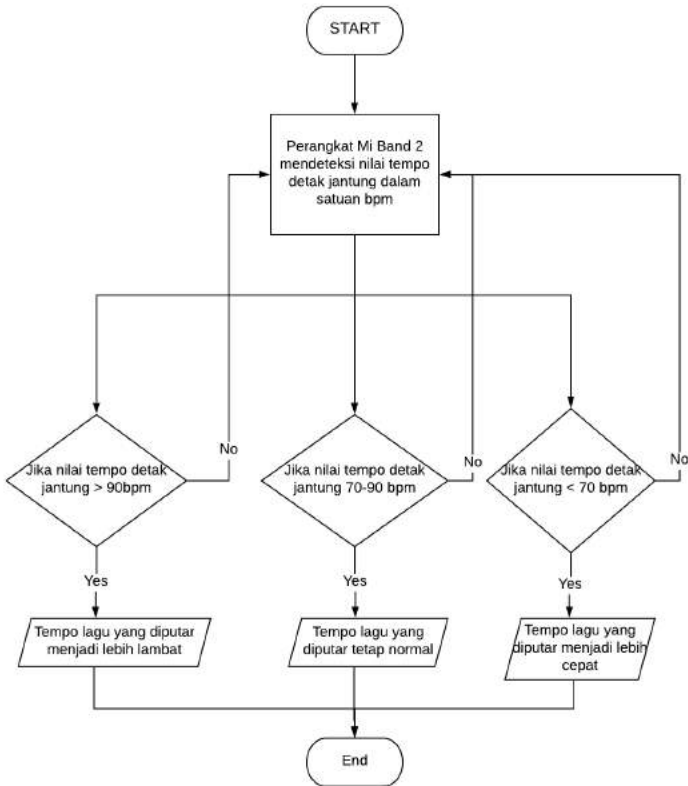
Pada pembagian *audio* dimana terbagi menjadi tiga bagian yaitu dipercepat, diperlambat, dan normal. *Audio* terlebih dahulu di-*init*, dimana proses ini dilakukan untuk membagi kecepatan *audio* menjadi tiga bagian agar mudah diolah dan meminimalisir adanya *delay* saat *audio* dimainkan. Setelah proses peng-*init*-an selesai maka pengguna dapat memainkan *audio* dan dapat menggunakannya saat berolahraga, dimana terlebih dahulu dihubungkan pada *Mi Band 2* oagar bisa memonitoring detak jantung serta *audio* dimainkan sesuai dengan detak jantung penggunanya.



Gambar 3.10: Pembagian Kecepatan *Audio*.

3.2.2 Pendektesian *Heart Rate* dengan *Mi Band 2* dan dikoneksikan dengan *audio*

Pada pembagian *audio* dimana terbagi menjadi tiga bagian yaitu dipercepat, diperlambat dan normal. Percepatan *audio* bergantung dengan *heart rate* yang telah dideteksi oleh *Mi Band 2*. Ketika *Mi Band 2* mendeteksi detak jantung pengguna berada pada nilai antara 60-100 bpm maka *audio* memiliki kecepatan normal atau berjalan sesuai dengan kecepatan semulanya. Namun, ketika kecepatan detak jantung pengguna dideteksi kurang dari 60 bpm maka *audio* akan berubah menjadi dipercepat. Adanya perubahan kecepatan *audio* ini agar detak jantung pengguna kembali ke nilai detak jantung normal atau bernilai di antara nilai 60-100 bpm. Kemudian ketika detak jantung pengguna di atas 100 bpm maka *audio* akan berubah kecepatannya menjadi diperlambat sampai dengan detak jantung pengguna bernilai 60-100 bpm atau kembali menjadi normal. Untuk menyesuaikan dalam kondisi aktivitas dalam kehidupan sehari-hari maka *range* detak jantung dirubah agar bisa mendapatkan nilai yang diinginkan dan berdasarkan kenyataan pada lapangan.

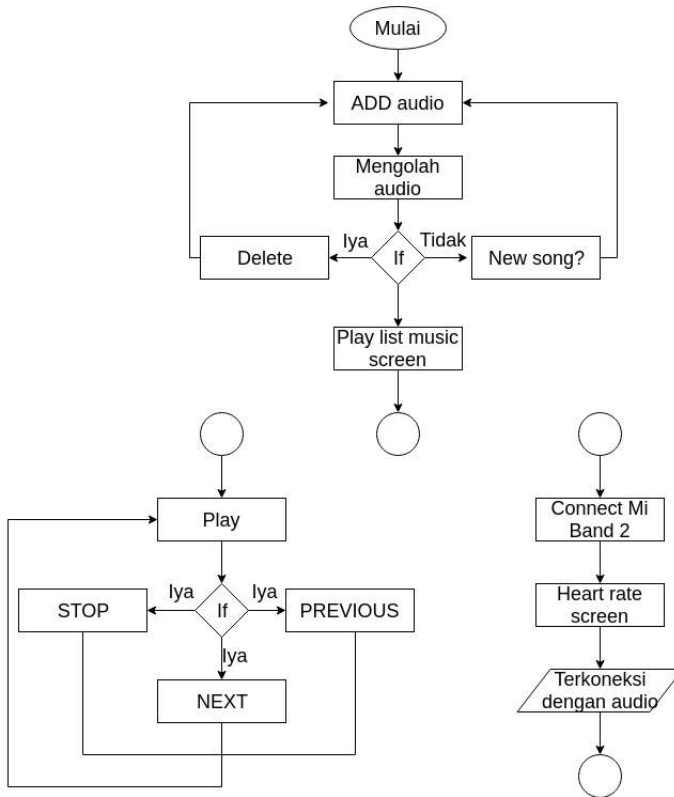


Gambar 3.11: Mengkoneksi heart rate Mi Band 2 dengan audio.

3.2.3 User Interface Screen

Pada *user interface* terdapat beberapa *button* dan *screen*, dimana *screen* pada bagian "*Music*" untuk menampilkan bagian lagu yang sudah diproses atau diolah oleh *user interface*, sedangkan *screen* pada bagian "*Heart Rate*" untuk menampilkan detak jantung pengguna yang sudah terkoneksi atau sudah tersambung dengan *Mi Band 2*. Dengan adanya *user interface* ini untuk mempermudah pengguna dalam mengaplikasikan *Mi Band 2* dengan *audio* yang sudah disediakan maupun yang ingin diolah oleh penggunanya.

Gambar 3.13 akan menjelaskan alur penggunaan *user interface*.



Gambar 3.12: Mengkoneksi *heart rate Mi Band 2* dengan *audio*.

Dalam penggunaan *user interface* pengguna pada gambar 3.11 dapat terlebih dahulu meng-add *audio* yang ingin dimainkan ketika berolahraga, kemudian musik akan diproses kecepatannya oleh WSOLA (meng-load *audio*) dan proses ini membutuhkan waktu bergantung pada panjangnya *audio* yang akan diproses. Pengguna juga dapat menghapus *list audio* yang ada pada *screen music* dan menambahkannya sesuai dengan kebutuhan. Setiap ada *audio* baru yang ingin di-add maka akan ada proses pengolahannya yang

memerlukan waktu namun ketika *audio* tersebut dihapus namun di-*add* kembali dan sebelumnya sudah pernah diproses maka tidak memerlukan pemrosesan WSOLA-nya kembali.

Pengguna juga perlu menghubungkan *Mi Band 2* dengan *user interface* agar mudah memantau detak jantung pengguna itu sendiri, nilai dari detak jantung akan ditampilkan pada *screen hear rate*. Ketika musik dan *Mi Band 2* sudah dikoneksikan dan diproses oleh *user interface* maka pengguna sudah bisa menggunakannya. Pada *button play* dapat berubah otomatis menjadi *button stop* hal ini untuk mempermudah pengguna dan meminimalisir jumlah *button* yang terdapat pada *user interface*. Seperti pada aplikasi pemutar *audio* lainnya, pengguna juga dapat memngembalikan ke *list audio* sebelumnya (*previous audio*) serta dapat melanjutkan ke *list audio* setelahnya (*next audio*).

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi dengan menggunakan dua komputer dari jenis yang berbeda. Komputer pertama yaitu Lenovo E330 yang memiliki spesifikasi thinkPad-Edge-E330, *Intel(R) Celeron(TM) i5-3230M CPU 1000M @2.60GHz 2.60 GHz*, RAM 8.00 GB dengan *system type* 64-bit dan komputer kedua yaitu Asus yang memiliki spesifikasi *processor Intel(R) Celeron(R) CPU 1000M @1.80GHz 1.80 GHz*, RAM 4.00 GB dengan *system type* 64-bit. Pengujian dibagi menjadi empat bagian antara lain:

1. Pengujian sistem dalam memori - meliputi pengujian memori yang dihabiskan dalam *me-load audio* dan memainkan *audio*.
2. Pengujian perangkat yang digunakan - meliputi pengujian akurasi nilai indeks *delay* saat *audio* dimainkan berdasarkan nilai indeks dari *Mi Band 2* (detak jantung pengguna).
3. Pengujian validasi *delay Mi Band 2* - meliputi pengujian validasi nilai indeks dari hasil pengukuran *delay Mi Band 2*.
4. Pengujian *user interface* - meliputi pengujian *user interface* pada kinerja musik dengan *Mi Band 2*.

Dengan dilaksanakannya beberapa pengujian tersebut, sehingga dapat ditarik kesimpulan dari pelaksanaan tugas akhir ini.

4.1 Pengujian Sistem dalam Penggunaan Memori

Pada bagian ini, dilakukan pengujian untuk mengetahui waktu rata-rata yang dihabiskan dalam *men-load* satu buah *audio* dan memori yang dihabiskan dalam memproses *audio* tersebut. Percobaan ini dilakukan pada beberapa macam *audio*.

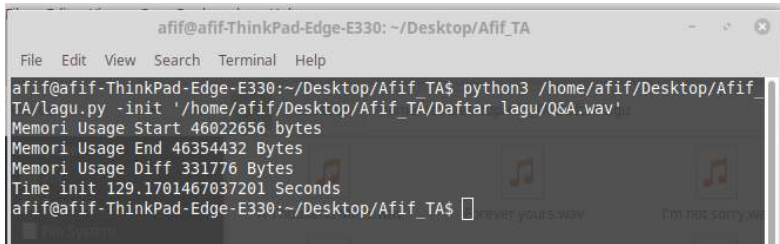
4.1.1 Memori yang Digunakan dalam Proses *-init*

Sebelum *audio* diproses bersamaan dengan *Mi Band 2*, terlebih dahulu diolah yaitu dengan meng-*init audio* tersebut. Hal ini bertujuan untuk memotong satu per-empat bagian pada *audio* dengan membaginya menjadi tiga jenis kecepatan yaitu dipercepat, diperlambat dan normal. Hal ini dilakukan agar mempermudah dalam pemrosesannya dan mengurangi adanya *delay* yang terjadi saat

dijalankan.

1. Menggunakan komputer Lenovo E330

Hasil yang didapatkan saat meng-*init audio* menggunakan komputer ini yaitu:

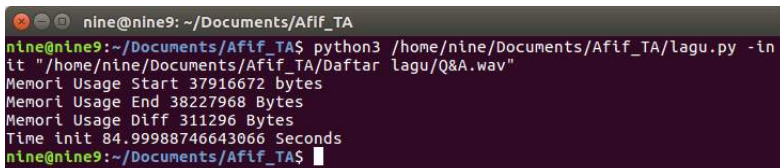


```
afif@afif-ThinkPad-Edge-E330: ~/Desktop/Afif_TA
File Edit View Search Terminal Help
afif@afif-ThinkPad-Edge-E330:~/Desktop/Afif_TA$ python3 /home/afif/Desktop/Afif
TA/lagu.py -init '/home/afif/Desktop/Afif_TA/Daftar lagu/Q&A.wav'
Memori Usage Start 46022656 bytes
Memori Usage End 46354432 Bytes
Memori Usage Diff 331776 Bytes
Time init 129.1701467037201 Seconds
afif@afif-ThinkPad-Edge-E330:~/Desktop/Afif_TA$
```

Gambar 4.1: -*init audio* menggunakan Lenovo E330.

2. Menggunakan komputer Asus

Hasil yang didapatkan ketika meng-*init audio* menggunakan komputer ini yaitu:



```
nine@nine9: ~/Documents/Afif_TA
nine@nine9:~/Documents/Afif_TA$ python3 /home/nine/Documents/Afif_TA/lagu.py -in
it "/home/nine/Documents/Afif_TA/Daftar lagu/Q&A.wav"
Memori Usage Start 37916672 bytes
Memori Usage End 38227968 Bytes
Memori Usage Diff 311296 Bytes
Time init 84.99988746643066 Seconds
nine@nine9:~/Documents/Afif_TA$
```

Gambar 4.2: -*init audio* menggunakan ASUS.

Pada gambar 4.1 memori yang digunakan dalam satu kali proses meng-*init audio* yaitu memori yang dihabiskan dalam satu kali proses meng-*init audio* yaitu memori *usage start* dengan nilai 46022656 bytes, sedangkan memori *usage end* dengan nilai 46354432 bytes, dan memori *usage diff* dengan nilai 331776 bytes. Sedangkan untuk *time init* yang dihabiskan dalam proses ini yaitu 129.1701 seconds. Sedangkan pada gambar 4.2 dimana memori yang dihabiskan dalam satu kali proses meng-*init audio* yaitu memori *usage start* dengan nilai 37916672 bytes, sedangkan memori *usage end* dengan

nilai 38227968 *bytes*, dan memori *usage diff* dengan nilai 3111296 *bytes*. *Time init* yang dihabiskan dalam proses ini yaitu 84.9998 *seconds* dengan *audio* berjudul "Q and A" dan panjang *audio* yaitu 3:40 detik.

1. Menggunakan komputer Lenovo E330

Hasil yang didapatkan saat meng-*init audio* menggunakan komputer ini yaitu:



Gambar 4.3: Folder potongan *audio* menggunakan Lenovo E330.

2. Menggunakan komputer Asus

Hasil yang didapatkan ketika meng-*init audio* menggunakan komputer ini yaitu:



Gambar 4.4: Folder potongan *audio* menggunakan ASUS.

Kemudian hasil dari *-init audio* disimpan pada memori dan dalam folder lagu dengan total potongan-potongan *audio* yaitu 44 buah seperti pada gambar 4.3 dan gambar 4.4.

4.1.2 Memori yang Digunakan dalam Proses *-play*

Setelah *audio* selesai melewati proses *-init* maka *audio* sudah dapat dijalankan sesuai dengan kecepatan pada jantung penggunaannya. Dari potongan-potongan *audio* yang tadinya dibagi-bagi menjadi satu perempat bagian, akan dipanggil dan diolah bergantung pada detak jantung penggunanya.

1. Menggunakan komputer Lenovo E330

Hasil yang didapatkan saat mem-*play audio* menggunakan komputer ini yaitu:

```
speed 2 | file-44 : 242.13361740112305 | delay : 0.2243518829345703
speed 2 | file-44 : 242.17891693115234 | delay : 0.2243518829345703
speed 2 | file-44 : 242.9516315460205 | delay : 0.2243518829345703
speed 2 | file-44 : 246.11902236938477 | delay : 0.2243518829345703
speed 2 | file-44 : 250.16283988952637 | delay : 0.2243518829345703
speed 2 | file-44 : 255.6297779083252 | delay : 0.2243518829345703
speed 2 | file-44 : 255.74922561645508 | delay : 0.2243518829345703
speed 2 | file-44 : 266.1299705505371 | delay : 0.2243518829345703
speed 2 | file-44 : 275.8316993713379 | delay : 0.2243518829345703
Memori Usage Start 45809664 bytes
Memori Usage End 52436992 Bytes
Memori Usage Diff 6627328 Bytes
Time init 215.7497627735138 Seconds
afif@afif-ThinkPad-Edge-E330:~/Desktop/Afif_TA$
```

Gambar 4.5: *-play audio* menggunakan Lenovo E330.

2. Menggunakan komputer Asus

Hasil yang didapatkan ketika mem-*play audio* menggunakan komputer ini yaitu:

```
speed 1 | file-44 : 447.1578598022461 | delay : 24.53899383544922
speed 1 | file-44 : 447.1738338470459 | delay : 24.53899383544922
speed 1 | file-44 : 447.1980963184248 | delay : 24.53899383544922
speed 1 | file-44 : 447.2062587738037 | delay : 24.53899383544922
speed 1 | file-44 : 447.2222328186035 | delay : 24.53899383544922
speed 1 | file-44 : 447.2384452819824 | delay : 24.53899383544922
speed 1 | file-44 : 447.2544193267822 | delay : 24.53899383544922
speed 1 | file-44 : 447.27863179816113 | delay : 24.53899383544922
speed 1 | file-44 : 447.28684425354004 | delay : 24.53899383544922
speed 1 | file-44 : 447.30305671691895 | delay : 24.53899383544922
speed 1 | file-44 : 447.31950759887695 | delay : 24.53899383544922
speed 1 | file-44 : 447.33548164367676 | delay : 24.53899383544922
speed 1 | file-44 : 454.809990409851074 | delay : 24.53899383544922
Memori Usage Start 37531648 bytes
Memori Usage End 43716608 Bytes
Memori Usage Diff 6184960 Bytes
Time init 205.22122430801392 Seconds
nline@ntne9:~/Documents/Afif_TA$
```

Gambar 4.6: *-play audio* menggunakan komputer ASUS.

Pada gambar 4.5 dan gambar 4.6 dimana memiliki nilai yang

berbeda dalam memori yang digunakan saat menajalankan *audio* bersamaan dengan *Mi Band 2* dimana pada saat menggunakan komputer Lenovo E330 tertera nilai dari memori yang dihabiskan dalam satu proses pengolahan *audio* bersamaan dengan *Mi Band 2* yaitu memori *usage start* bernilai 45809664 *bytes*, memori *usage end* bernilai 52436992 *bytes*, memori *usage diff* bernilai 6627328 *bytes* dan *time init* bernilai 215.7497 *seconds*.

Sedangkan pada saat dijalankan pada komputer kedua yaitu ASUS tertera nilai dari memori yang dihabiskan dalam satu proses pengolahan *audio* bersamaan dengan *Mi Band 2* yaitu memori *usage start* bernilai 37531648 *bytes*, memori *usage end* bernilai 43716608 *bytes*, memori *usage diff* bernilai 6184960 *bytes* dan *time init* bernilai 205.2212 *seconds*. *Time init* pada proses saat *play* berbeda dengan saat *init* pada awal pemrosesan *audio*, yaitu pada saat meng-*init audio* saat pertama kali yaitu untuk memotong *audio* menjadi satu perempat bagian menjadi dipercepat, diperlambat dan normal, sedangkan dalam proses *-play* yaitu untuk mengolah *audio* dalam pemrosesan kecepatannya yang tadi sudah dipotong-potong menjadi beberapa bagian.

4.2 Pengujian Perangkat

Pada penelitian ini, melakukan pengujian terhadap perangkat yang digunakan dalam mendeteksi detak jantung pengguna yaitu *Mi Band 2* yang kemudian digabungkan dengan *audio* untuk merubah kecepatannya. Ada dua macam pengujian yang dilakukan yaitu pada kecepatan *audio* dengan nilai *delay*-nya dan nilai *heart rate* yang dideteksi oleh *Mi Band 2* saat dijalankan secara bersamaan.

4.2.1 Pengujian Pada *Audio*

Pada saat *audio* dijalankan bersamaan dengan *Mi Band 2*, kecepatan pada *audio* yaitu *beat audio* bergantung pada detak jantung pengguna, oleh karena itu kecepatan *audio* sangatlah bergantung pada nilai BPM (*bit per minute*) yang dideteksi oleh *Mi Band 2* dari detak jantung pengguna.

1. Menggunakan komputer Lenovo E330

Delay yang terjadi saat menggunakan komputer ini yaitu:

```

aifff@afif-ThinkPad-Edge-E330: ~/Desktop/Afif_TA
File Edit View Search Terminal Help
speed 2 | file-44 : 195.56665420532227 | delay : 0.2243518829345703
speed 2 | file-44 : 202.1162509918213 | delay : 0.2243518829345703
speed 2 | file-44 : 202.16035842895508 | delay : 0.2243518829345703
speed 2 | file-44 : 214.11895751953125 | delay : 0.2243518829345703
speed 2 | file-44 : 216.17579460144043 | delay : 0.2243518829345703
speed 2 | file-44 : 216.81857109069824 | delay : 0.2243518829345703
speed 2 | file-44 : 223.81043434143066 | delay : 0.2243518829345703
speed 2 | file-44 : 224.07293319702148 | delay : 0.2243518829345703
speed 2 | file-44 : 224.13015365600586 | delay : 0.2243518829345703
speed 2 | file-44 : 224.6851921081543 | delay : 0.2243518829345703
speed 2 | file-44 : 242.13361740112305 | delay : 0.2243518829345703
speed 2 | file-44 : 242.17891693115234 | delay : 0.2243518829345703
speed 2 | file-44 : 242.9516315460205 | delay : 0.2243518829345703
speed 2 | file-44 : 246.11902236938477 | delay : 0.2243518829345703
speed 2 | file-44 : 250.16283980952637 | delay : 0.2243518829345703
speed 2 | file-44 : 255.6297779083252 | delay : 0.2243518829345703
speed 2 | file-44 : 255.74922561645508 | delay : 0.2243518829345703
speed 2 | file-44 : 266.1299705505371 | delay : 0.2243518829345703
speed 2 | file-44 : 275.8316993713379 | delay : 0.2243518829345703
Memori Usage Start 45809664 bytes
Memori Usage End 52436992 Bytes
Memori Usage Diff 6627328 Bytes
Time init 215.7497627735138 Seconds
afifff@afif-ThinkPad-Edge-E330:~/Desktop/Afif_TA$

```

Gambar 4.7: Delay audio menggunakan komputer Lenovo E330.

2. Menggunakan komputer Asus

Delay yang terjadi saat menggunakan komputer ini yaitu:

```

nline@nline9: ~/Documents/Afif_TA
speed 1 | file-44 : 447.06034660339355 | delay : 24.53899383544922
speed 1 | file-44 : 447.07655906677246 | delay : 24.53899383544922
speed 1 | file-44 : 447.09277153015137 | delay : 24.53899383544922
speed 1 | file-44 : 447.1094608306885 | delay : 24.53899383544922
speed 1 | file-44 : 447.1259117126465 | delay : 24.53899383544922
speed 1 | file-44 : 447.1418857574463 | delay : 24.53899383544922
speed 1 | file-44 : 447.1578598022461 | delay : 24.53899383544922
speed 1 | file-44 : 447.1738338470459 | delay : 24.53899383544922
speed 1 | file-44 : 447.1900463104248 | delay : 24.53899383544922
speed 1 | file-44 : 447.2062587738037 | delay : 24.53899383544922
speed 1 | file-44 : 447.2222328186035 | delay : 24.53899383544922
speed 1 | file-44 : 447.2384452819824 | delay : 24.53899383544922
speed 1 | file-44 : 447.2544193267822 | delay : 24.53899383544922
speed 1 | file-44 : 447.27063179016113 | delay : 24.53899383544922
speed 1 | file-44 : 447.28684425354004 | delay : 24.53899383544922
speed 1 | file-44 : 447.30305671691895 | delay : 24.53899383544922
speed 1 | file-44 : 447.31950759887695 | delay : 24.53899383544922
speed 1 | file-44 : 447.33548164367676 | delay : 24.53899383544922
speed 1 | file-44 : 454.80990409851074 | delay : 24.53899383544922
Memori Usage Start 37531648 bytes
Memori Usage End 43716608 Bytes
Memori Usage Diff 6184960 Bytes
Time init 205.22122430801392 Seconds
nline@nline9:~/Documents/Afif_TA$

```

Gambar 4.8: Delay audio menggunakan komputer ASUS.

Namun pada saat pemrosesan audio terdapat delay didalam-

nya. Sehingga baik pada *audio* dan *Mi Band 2* yang dijalankan secara *real time* tetap terdapat *delay* didalamnya. Namun hal ini dapat diminimalisir dengan mengolah terlebih dahulu *audio* dengan meng-*init audio* pada proses awal sebelum dimainkan.

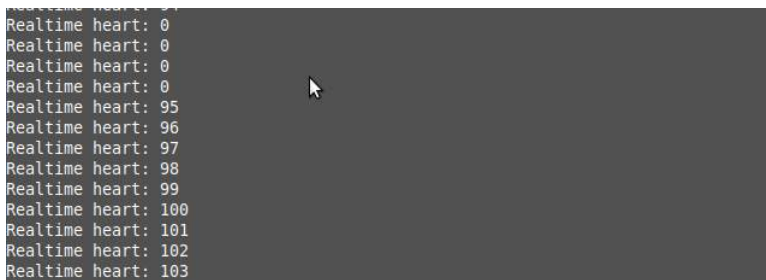
Seperti pada gambar 4.7 dan gambar 4.8 dimana *delay* dari *audio* yang dijalankan, *delay* daripada *audio* tersebut hampir mendekati nilai nol sehingga tidak begitu terdengar dengan jelas patahan dari *audio* yang terpotong-potong kecepataannya yang kemudian digabungkan menjadi satu buah lagu kembali. *Delay* pada setiap *audio* tidaklah sama tergantung pada panjangnya sebuah *audio* tersebut dan tergantung pada komputer yang digunakan.

4.2.2 Heart Rate pada *Mi Band 2*

Nilai dari detak jantung pengguna sangatlah berpengaruh pada kecepatan *audio*, nilai dari detak jantung pengguna didapatkan dari *Mi Band 2* yang dihubungkan dengan *bluetooth* ke perangkat komputer. Namun pada komputer ASUS tidak memiliki akses *bluetooth* pada komputernya sehingga ditulis secara manual. Sedangkan pada komputer dengan jenis Lenovo E330 memiliki akses *bluetooth* sehingga terhubung langsung dan nilai murni didapatkan dari *Mi Band 2*.

1. Menggunakan komputer Lenovo E330

Delay yang terjadi saat menggunakan komputer ini yaitu:

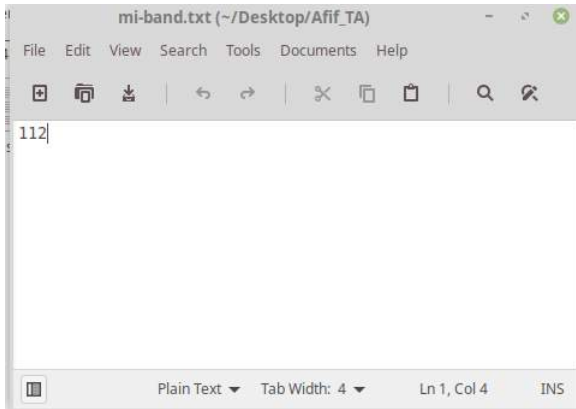


```
Realtime heart: 0
Realtime heart: 0
Realtime heart: 0
Realtime heart: 0
Realtime heart: 0
Realtime heart: 95
Realtime heart: 96
Realtime heart: 97
Realtime heart: 98
Realtime heart: 99
Realtime heart: 100
Realtime heart: 101
Realtime heart: 102
Realtime heart: 103
```

Gambar 4.9: Heart rate *Mi Band 2* menggunakan komputer Lenovo E330.

2. Menggunakan komputer Asus

Delay yang terjadi saat menggunakan komputer ini yaitu:



Gambar 4.10: Rata-rata *Heart rate Mi Band 2* menggunakan komputer ASUS.

Pada gambar 4.9 menggunakan komputer Lenovo E330 dimana tersambung langsung dengan perangkat *bluetooth* pada komputer sehingga nilai secara *real time* dari *Mi Band 2*. Sedangkan pada gambar 4.10 menggunakan komputer Asus, dimana merupakan pemrosesan mendapatkan nilai detak jantung dengan cara *manual* yaitu menulis pada *script mi-band.txt*. Dengan menggunakan dua jenis komputer yang berbeda terlihat perbedaan yang sangat signifikan.

Nilai dari detak jantung pengguna sangatlah berpengaruh pada kecepatan *audio*, nilai dari detak jantung pengguna didapatkan dari *Mi Band 2* yang dihubungkan dengan *bluetooth* ke perangkat komputer. Namun pada komputer ASUS tidak memiliki akses *bluetooth* pada komputernya sehingga ditulis secara manual. Sedangkan pada komputer dengan jenis Lenovo E330 memiliki akses *bluetooth* sehingga terhubung langsung dan nilai murni didapatkan dari *Mi Band 2*.

4.3 Pengujian *Delay Mi Band 2*

Pada saat *audio* dijalankan bersamaan dengan *Mi Band 2*, kecepatan pada *audio* yaitu *beat audio* bergantung pada detak jantung penggunaanya, oleh karena itu kecepatan *audio* sangatlah bergantung pada nilai BPM yang dideteksi oleh *Mi Band 2*.

1. Menggunakan komputer Lenovo E330

Delay yang terjadi saat menggunakan komputer ini yaitu:

```
670.4316139221191
1822.948932647705
543.1702136993408
347.9182720184326
234.5118522644043
1448.7721920013428
1471.482276916504
1337.7704620361328
1141.8423652648926
1145.8513736724854
2400.5038738250732
3625.1449584960938
879.1565895080566
880.805253982544
795.7887649536133
3312.5011920928955
6628.5765171051025
9944.462537765503
13273.737668991089
```

Gambar 4.11: *Delay Mi Band 2* menggunakan komputer Lenovo E330.

2. Menggunakan komputer ASUS

Delay yang terjadi saat menggunakan komputer ini yaitu:

```
mi-band.txt x delay_mi_band.txt x
1 |1557850977683.5212
2 |1557850982764.9468
3 |1557850987800.03
4 |1557850992858.3772
5 |1557850997875.2195
6 |1557851002997.7222
7 |1557851008050.4329
8 |1557851011386.386
9 |1557851014726.3335
10 |1557851018321.7605
11 |1557851021637.646
12 |1557851024966.4983
13 |1557851028325.3728
14 |1557851031729.3887
15 |1557851035081.3076
16 |1557851038411.8125
17 |1557851043457.6138
18 |1557851048490.4238
19 |1557851053499.7366
20 |1557851063505.3342
21 |1557851073454.8074
22 |1557851083397.15
23 |1557851093368.907
24 |1557851098416.3762
25 |1557851103448.2705
26 |1557851108465.0
27 |1557851113512.99
28 |1557851116852.065
```

Gambar 4.12: *Delay Mi Band 2* menggunakan komputer ASUS.

Pada gambar 4.11 dan gambar 4.12, dimana ketika detak jantung pengguna yang dideteksi oleh *Mi Band 2* mencapai nilai diatas 90 BPM maupun dibawah nilai 70 BPM, terjadi *delay* dalam pemrosesan *audio*-nya. Sehingga dapat dikatakan tidak langsung mengganti kecepatan *audio*-nya ketika angka mencapai 91 BPM atau 69 bpm, *audio* diolah setelah nilai detak jantung di beberapa detik selanjutnya. Seperti saat 91, 92, 93, 94, dan 95 (nilai bisa berubah sesuai dengan detak jantung pengguna) baru dieksekusi ketika pada nilai 95 BPM, ketika nilai dari detak jantung penggunanya dibawah 70 BPM seperti 69, 68, 67, 66, 65 bisa saja terjadi *delay* saat akan mengeksekusi kecepatan *audio*-nya, begitu juga saat detak jantung kembali ke normal atau sekitar 70-90 BPM tetap terjadi *delay* didalamnya. Sehingga perlunya menghitung rata-rata dari *delay* yang ada.

1. Menggunakan komputer Lenovo E330

Delay yang terjadi saat menggunakan komputer ini yaitu:

| | A | B | C |
|---|------------------|---|-----------------|
| 1 | 3974.7588634491 | | 2301.9714734771 |
| 2 | 3819.420337677 | | |
| 3 | 7140.43807983398 | | |
| 4 | 10455.0890922546 | | |
| 5 | 1278.87344360352 | | |
| 6 | 1232.63597488403 | | |
| 7 | 1239.05920982361 | | |
| 8 | 7.71713256835938 | | |

Gambar 4.13: Rata-rata *delay Mi Band 2* menggunakan komputer Lenovo E330.

2. Menggunakan komputer ASUS

Delay yang terjadi saat menggunakan komputer ini yaitu:

| | A | B | C | D |
|----|------------------|---|------------------|---|
| 1 | 1557850977683.52 | | 1557851082205.62 | |
| 2 | 1557850982764.95 | | | |
| 3 | 1557850987800.03 | | | |
| 4 | 1557850992858.38 | | | |
| 5 | 1557850997875.22 | | | |
| 6 | 1557851002997.72 | | | |
| 7 | 1557851008050.43 | | | |
| 8 | 1557851011386.39 | | | |
| 9 | 1557851014726.33 | | | |
| 10 | 1557851018321.76 | | | |

Gambar 4.14: Rata-rata *delay Mi Band 2* menggunakan komputer ASUS.

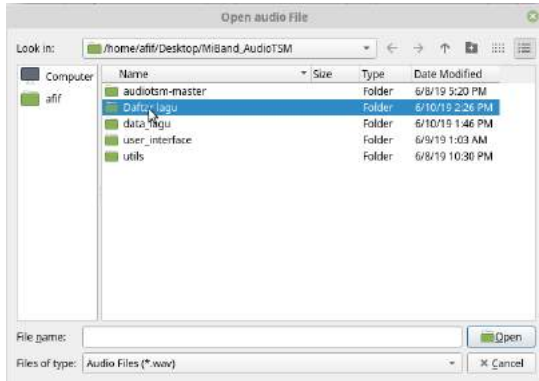
Pada gambar 4.7 merupakan rata-rata *delay Mi Band 2* saat kecepatan *audio* dirubah sesuai dengan perintah atau nilai dari *heart rate Mi Band 2*. Sehingga nilai dari rata-rata *delay*-nya yaitu 1557859792066.95 *milisecond* atau setara dengan 1.5 detik.

4.4 Pengujian *User Interface*

Pada pengujian *user interface*, dimana pada pengujian ini dapat menampilkan *audio* dan menggabungkannya dengan *Mi Band 2*.

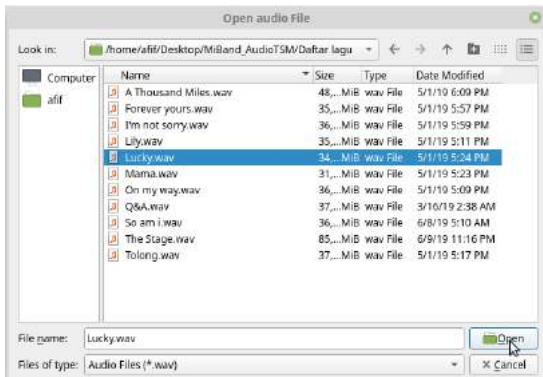
4.4.1 *Buttton add*

Sebelum memainkan lagu, terlebih dahulu memilih lagu yang sudah disediakan pada folder Daftar lagu seperti pada gambar 4.15.



Gambar 4.15: Folder daftar lagu.

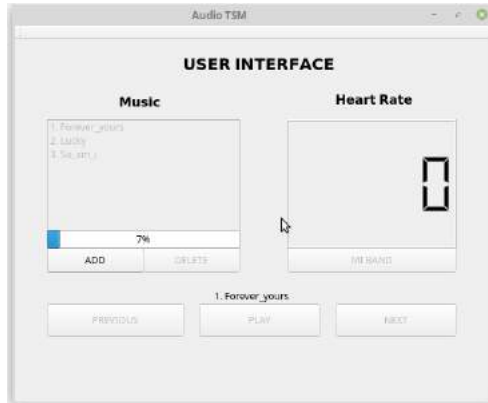
Kemudian memilih lagu yang ingin diproses dan dimainkan pada *user interface* seperti pada gambar 4.16.



Gambar 4.16: Memilih lagu.

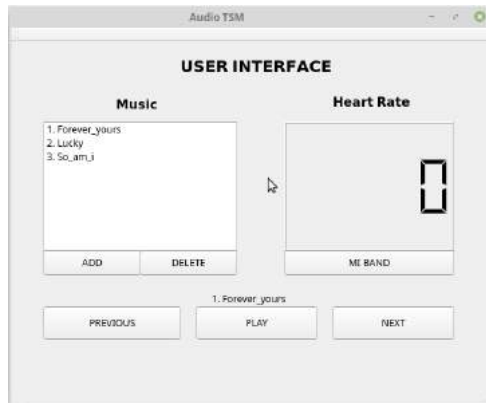
4.4.2 Men-load Lagu

Setelah lagu dipilih maka lagu akan diolah kecepatannya seperti pada gambar 4.17.



Gambar 4.17: Memproses kecepatan *audio*

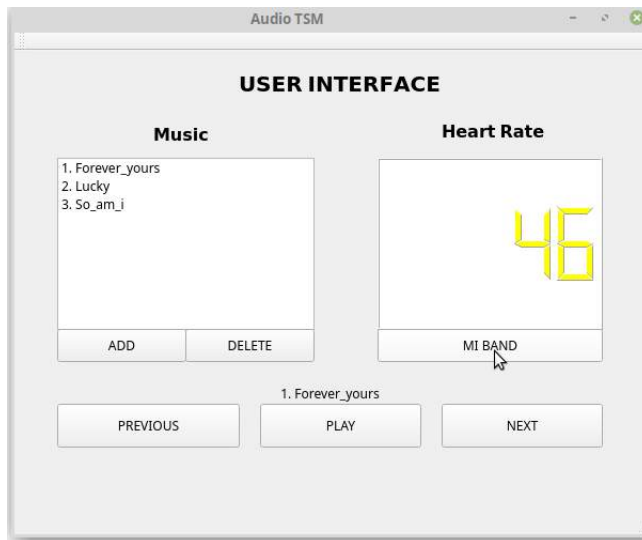
Kemudian pada *screen music* akan tampil lagu yang sudah di *add* dan di *load* seperti pada gambar 4.18.



Gambar 4.18: Hasil penambahan lagu

4.4.3 *Button Mi Band 2*

Setelah pemilihan lagu selesai, pengguna terlebih dahulu mengaktifkan *Mi Band 2* dengan *button Mi Band* sebelum memulai memainkan lagunya.

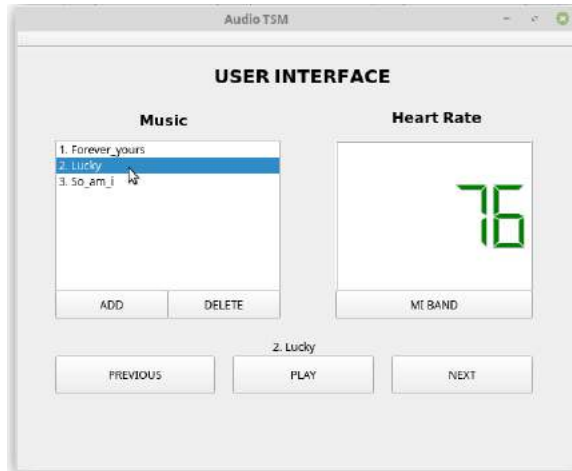


Gambar 4.19: Mengaktifkan *Mi Band 2*

Seperti pada gamabar 4.19, setelah *Mi Band 2* terhubung maka akan keluar nilai dari detak jantung penggunanya.

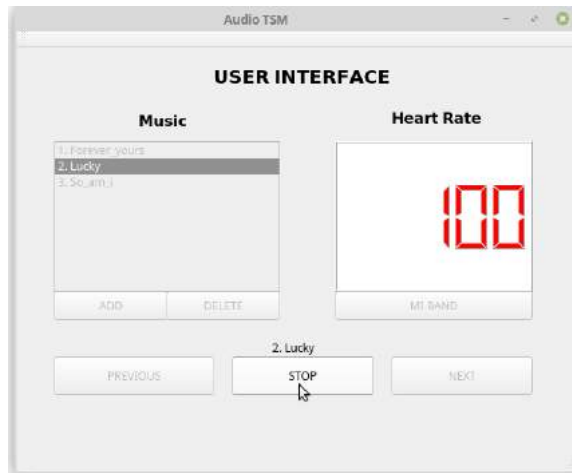
4.4.4 *Button play and stop*

Pengguna sudah dapat menggunakan *user interface* dan memainkan lagu untuk berolahraga seperti senam *aerobic* dan memainkan lagu dengan menekan *button play* seperti pada gambar 4.20.



Gambar 4.20: Memainkan lagu.

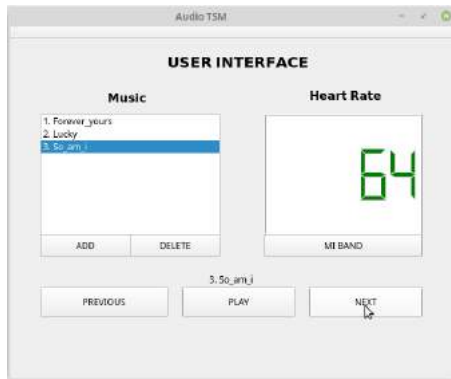
Dan ketika ingin menghentikan lagu, dapat menggunakan *buton stop* seperti pada gambar 4.21.



Gambar 4.21: Menghentikan lagu.

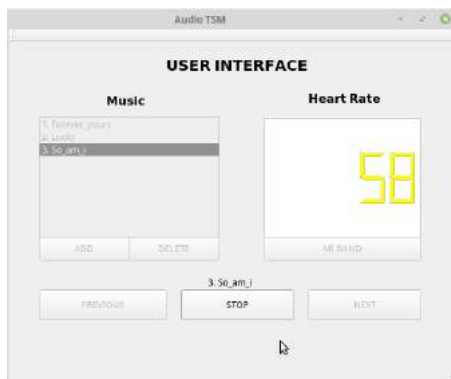
4.4.5 Fungsi *Button Next*

Ketika pengguna ingin mengganti lagu pada *list* selanjutnya. Pengguna dapat menggunakan *button next* yang berfungsi untuk memainkan lagu di *list* berikutnya. Seperti contohnya, lagu yang sebelumnya dimainkan yang berjudul "*Lucky*" berubah menjadi lagu yang berjudul "*So am i*".



Gambar 4.22: Mengklik *button next*.

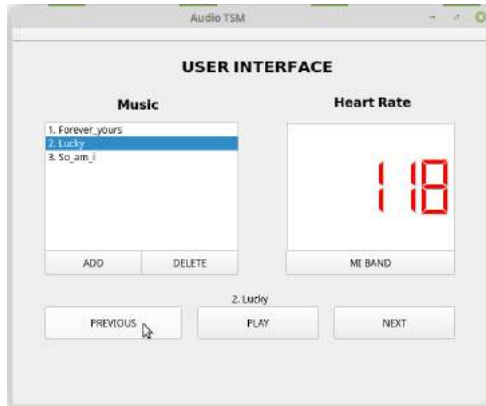
Seperti pada gambar 4.22 dan gambar 4.23 dimana lagu yang dimainkan berubah.



Gambar 4.23: Lagu berganti.

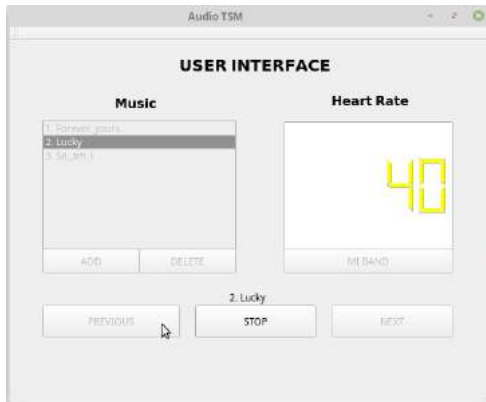
4.4.6 Fungsi *Button Previous*

Pada *button* ini berfungsi untuk mengganti lagu ke lagu sebelumnya seperti pada gambar 4.24.



Gambar 4.24: Memilih lagu sebelumnya.

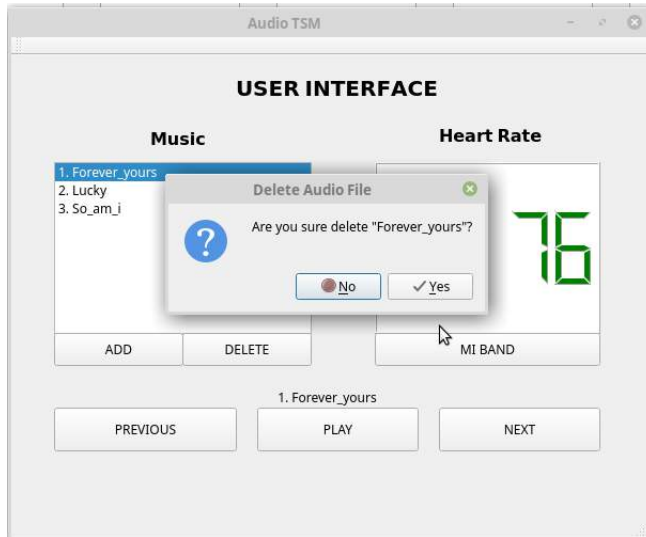
Setelah pengguna mengganti lagu maka yang lagu yang dimainkan adalah lagu yang sebelumnya berjudul "Lucky".



Gambar 4.25: Lagu berubah.

4.4.7 Fungsi *Button Delete*

Fungsi pada *button* ini yaitu untuk menghapus lagu yang sudah tidak ingin didengar oleh penggunanya seperti pada gambar 4.26.

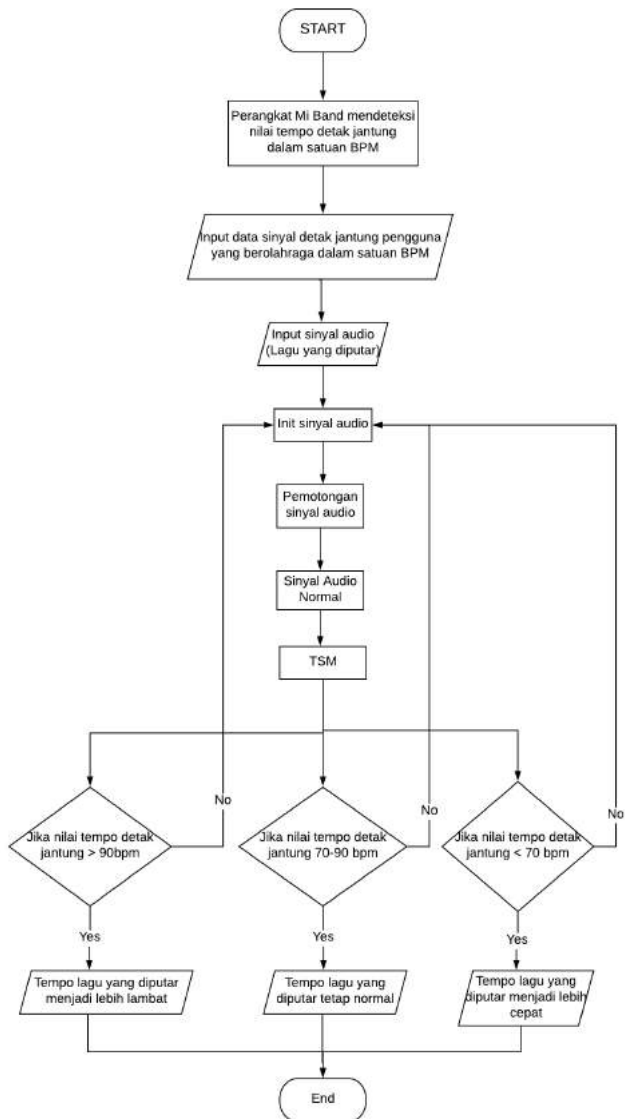


Gambar 4.26: Lagu berubah.

Pengguna dapat menghapus lagu yang diinginkan dengan terlebih dahulu memilihnya. Dan dapat kembali menambahkan lagu tersebut dengan *button add* yang sudah tersedia.

4.5 *Eksperimen Desain*

Dimana pada gambar 4.16 menunjukkan *eksperimen desain* yang dilakukan dalam percobaan penelitian ini. Pengguna menggunakan *Mi Band 2* dengan *audio* dijalankan secara bersamaan. Yang nantinya output dari lagu bergantung dari *heart rate* penggunanya. Dari sanalah perubahan *audio* pada lagu berbeda kecepatannya.



Gambar 4.27: Eksperimen desain

4.6 Pengujian Kecepatan *Beat Audio* dan Tempo *Audio*

4.6.1 *Audio* Jenis Pop

Pengujian kecepatan *audio* pada jenis musik ini dengan cara mengeksekusi pada terminal, didapatkan hasil dimana lagu dengan judul lagu " *Q and A*" yang dinyanyikan oleh Seventeen feat. Ailee, dimana kecepatan *beat audio* yang diproses dalam satu lagu dimana dipercepat empat kali, diperlambat tiga kali dan normal lima kali. Namun terjadi *error* pada potongan *audio* ke-33 dengan total potongan *audio* sebanyak 44 buah. Pada potongan tersebut terjadi *error* dimana *audio* mati namun proses pada terminal tetap berjalan sampai potongan ke-44.

Tempo pada *audio* lagu ini dimana durasi 3:40 detik, bernilai 120 BPM (*beat per minute*) dengan size *audio* yaitu 37976 kb (*kilobyte*). Nilai tersebut merupakan nilai sebelum terjadinya pengolahan *audio*. Sedangkan ketika sudah diolah, durasi dari *audio* berubah menjadi 3:32 detik, sedangkan tempo dari *audio* tetaplah sama yaitu 120 BPM (*beat per minute*) namun size berubah menjadi 36684 kb (*kilobyte*).

4.6.2 *Audio* Jenis Rock

Pengujian kecepatan *audio* pada jenis musik ini dengan cara mengeksekusi pada terminal, didapatkan hasil dimana lagu dengan judul " *The Stage*" yang dinyanyikan oleh Avenged Sevenfold, dimana kecepatan *beat audio* yang diproses dalam satu lagu dimana dipercepat empat kali, diperlambat tiga kali dan normal lima kali. Namun terjadi *error* pada potongan *audio* ke-33 dengan total potongan *audio* sebanyak 44 buah. Pada potongan tersebut terjadi *error* dimana *audio* mati namun proses pada terminal tetap berjalan sampai potongan ke-44.

Tempo pada *audio* lagu ini dimana durasi 8:30 detik, bernilai 120 BPM (*beat per minute*) dengan size *audio* yaitu 87961 kb (*kilobyte*). Nilai tersebut merupakan nilai sebelum terjadinya pengolahan *audio*. Sedangkan ketika sudah diolah, durasi dari *audio* belum diketahui karena terjadinya *bug* saat dijalankan pada terminal.

4.6.3 *Audio* Jenis Klasik

Pengujian kecepatan *audio* pada jenis musik ini dimana menggunakan *user interface*, didapatkan hasil yaitu lagu dengan judul "Amazing Grace" yang dinyanyikan oleh Celtic Woman, dimana kecepatan *beat audio* yang diproses dalam satu lagu dimana dipercepat empat kali, diperlambat tiga kali dan normal lima kali. Dengan jumlah potongan *audio* sebanyak 54 buah.

Tempo pada *audio* lagu ini dimana durasi 4:30 detik, bernilai 120 BPM (*beat per minute*) dengan size *audio* yaitu 46512 kb (*kilobyte*), nilai tersebut merupakan nilai sebelum terjadinya pengolahan *audio*. Sedangkan ketika sudah diolah, durasi dari *audio* berubah menjadi 7:22 detik, sedangkan tempo dari *audio* tetaplah sama yaitu 120 BPM (*beat per minute*) namun size berubah menjadi 76165 kb (*kilobyte*).

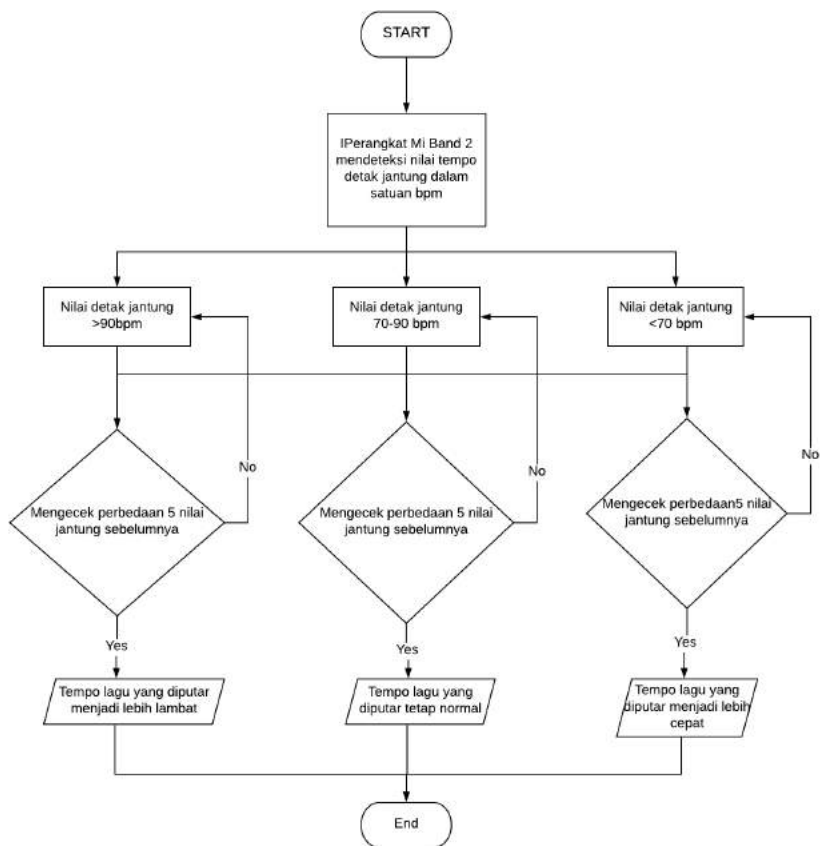
4.6.4 *Audio* Jenis Disco

Pengujian kecepatan *audio* pada jenis musik ini dimana menggunakan *user interface*, didapatkan hasil yaitu lagu dengan judul "Disco Lazy Time" yang dinyanyikan oleh Nidji Band, dimana kecepatan *beat audio* yang diproses dalam satu lagu dimana dipercepat sebanyak tujuh kali, diperlambat sebanyak enam kali dan normal sebanyak enam kali. Dengan jumlah potongan *audio* sebanyak 52 buah.

Tempo pada *audio* lagu ini dimana durasi 4:2 detik, bernilai 120 BPM (*beat per minute*) dengan size *audio* yaitu 49129 kb (*kilobyte*), nilai tersebut merupakan nilai sebelum terjadinya pengolahan *audio*. Sedangkan ketika sudah diolah, durasi dari *audio* berubah menjadi 7:06 detik, sedangkan tempo dari *audio* tetaplah sama yaitu 120 BPM (*beat per minute*) namun size berubah menjadi 79947 kb (*kilobyte*).

4.7 Membandingkan Nilai *Audio* yang Terdahulu Sebelum Dieksekusi

Pada pengujian ini dilakukan untuk mengetahui nilai *heart rate* sebelum mengolah kecepatan *audio*. Didapatkan bahwa, *audio* diolah setelah memiliki nilai yang berbeda dari lima nilai sebelumnya baik untuk kecepatan normal, lambat, dan cepat.



Gambar 4.28: Melakukan perbandingan nilai sebelum *audio* dieksekusi

Pada gambar 4.28, *Mi Band 2* mendeteksi detak jantung penggunaanya dan untuk mengatur kecepatan dari *audio* yang dimainkan. Ketika *range* detak jantung pengguna baik kurang dari 70 BPM, lebih dari 90 BPM dan sekitar 70-90 BPM maka sebelum memainkan *audio* terlebih dahulu mengecek dan membandingkan dengan nilai-nilai sebelumnya sebanyak 5 kali nilai sebelumnya. Jika terjadi perbedaan sampai dengan 5 kalinya nilai yang terdahulu maka

audio akan dieksekusi perubahan kecepatannya.

Hal ini dilakukan untuk memastikan apakah data dari detak jantung yang didapatkan sesuai dengan kondisi yang ada dan tidak dieksekusi begitu saja. Namun, jika nilai dari detak jantung yang dideteksi nilainya naik turun atau tidak sampai lima nilai perbedaan dari nilai sebelumnya maka tidak dapat dieksekusi dan kembali atau terus, mendeteksi detak jantung dengan menggunakan *Mi Band 2*. Sehingga *audio* tersebut dirubah berdasarkan atau sesuai dengan kondisi dan kebutuhan dari detak jantung penggunaannya tersebut.

Sebagai contohnya yaitu, ketika detak jantung pengguna yang dideteksi oleh *Mi Band 2* mencapai nilai diatas 90 BPM maupun dibawah nilai 70 BPM, terjadi *delay* dalam pemrosesan *audio*-nya. Sehingga dapat dikatakan tidak langsung mengganti kecepatan *audio*-nya ketika angka mencapai 91 BPM atau 69 bpm, *audio* diolah setelah nilai detak jantung di beberapa detik selanjutnya untuk membandingkan nilai-nilai sebelumnya. Seperti saat 91, 92, 93, 94, dan 95 (nilai bisa berubah sesuai dengan detak jantung pengguna) baru dieksekusi ketika pada nilai 95 BPM. Begitu juga ketika nilai dari detak jantung penggunaannya dibawah 70 BPM seperti 69, 68, 67, 66, 65 dimana eksekusi terjadi ketika mencapai nilai 65 BPM karena membandingkan nilai-nilai sebelumnya untuk memastikan apakah sudah tepat untuk merubah kecepatan dari *audio*. Saat detak jantung kembali ke normal atau sekitar 70-90 BPM tetap melakukan pengecekan nilai-nilai sebelumnya baru setelah itu *audio* kembali menjadi kecepatan normal.

4.8 Pengujian pada *Audio*

Pada pengujian ini dilakukan untuk mengetahui perpindahan kecepatan *audio* sehingga dapat mengetahui keakuratan atau validasi nilai dari kinerja sistem dari penelitian ini.

4.8.1 Kecepatan *Audio* dari Normal ke Cepat

Tabel 4.1: Pengujian *audio* dari normal ke cepat

| <i>Heart Rate</i> (bpm) | Perubahan Tempo <i>Audio</i> |
|-------------------------|------------------------------|
| 64 (bpm) | Berhasil |
| 54 (bpm) | Berhasil |
| 55 (bpm) | Berhasil |
| 67 (bpm) | Berhasil |
| 44 (bpm) | Berhasil |
| 50 (bpm) | Tidak Berhasil |
| 62 (bpm) | Berhasil |
| 60 (bpm) | Berhasil |
| 50 (bpm) | Berhasil |
| 61 (bpm) | Berhasil |
| 68 (bpm) | Berhasil |
| 53 (bpm) | Berhasil |
| 40 (bpm) | Berhasil |
| 56 (bpm) | Berhasil |
| 59 (bpm) | Berhasil |
| 63 (bpm) | Berhasil |
| 62 (bpm) | Berhasil |
| 51 (bpm) | Tidak Berhasil |
| 58 (bpm) | Berhasil |
| 57 (bpm) | Berhasil |

Seperti pada tabel 4.1, dimana didapatkan nilai-nilai *heart rate Mi Band 2* untuk mengubah kecepatan *audio* dari normal ke cepat. Pada tabel tersebut didapatkan hasil bahwa dari 20 data yang tertera diantaranya 18 berhasil dalam memproses kecepatan *audio* normal ke cepat sedangkan 2 diantaranya tidak berhasil dikarenakan terjadinya *bug* dalam proses dan tidak mengeluarkan suara dari *audio* yang dimainkan walaupun proses terus berjalan pada terminal.

4.8.2 Kecepatan *Audio* dari Normal ke Lambat

Tabel 4.2: Pengujian *audio* dari normal ke lambat

| <i>Heart Rate</i> (bpm) | Perubahan Tempo <i>Audio</i> |
|-------------------------|------------------------------|
| 96 (bpm) | Berhasil |
| 95 (bpm) | Berhasil |
| 101 (bpm) | Berhasil |
| 98 (bpm) | Berhasil |
| 95 (bpm) | Berhasil |
| 94 (bpm) | Tidak Berhasil |
| 96 (bpm) | Berhasil |
| 102 (bpm) | Berhasil |
| 97 (bpm) | Berhasil |
| 92 (bpm) | Tidak Berhasil |
| 98 (bpm) | Berhasil |
| 96 (bpm) | Berhasil |
| 99 (bpm) | Berhasil |
| 104 (bpm) | Berhasil |
| 105 (bpm) | Berhasil |
| 111 (bpm) | Berhasil |
| 120 (bpm) | Berhasil |
| 106 (bpm) | Berhasil |
| 100 (bpm) | Berhasil |
| 121 (bpm) | Berhasil |

Seperti pada tabel 4.2, dimana didapatkan nilai-nilai *heart rate Mi Band 2* untuk mengubah kecepatan *audio* dari normal ke lambat. Pada tabel tersebut didapatkan hasil bahwa dari 20 data yang tertera diantaranya 18 berhasil dalam memproses kecepatan *audio* normal ke cepat sedangkan 2 diantaranya tidak berhasil dikarenakan terjadinya *bug* dalam proses dan tidak mengeluarkan suara dari *audio* yang dimainkan walaupun proses terus berjalan pada terminal.

4.8.3 Kecepatan *Audio* dari Cepat ke Normal

Tabel 4.3: Pengujian *audio* dari cepat ke normal

| <i>Heart Rate</i> (bpm) | Perubahan Tempo <i>Audio</i> |
|-------------------------|------------------------------|
| 80 (bpm) | Berhasil |
| 70 (bpm) | Berhasil |
| 78 (bpm) | Berhasil |
| 76 (bpm) | Berhasil |
| 74 (bpm) | Berhasil |
| 90 (bpm) | Berhasil |
| 88 (bpm) | Berhasil |
| 85 (bpm) | Berhasil |
| 77 (bpm) | Berhasil |
| 82 (bpm) | Tidak Berhasil |
| 71 (bpm) | Berhasil |
| 87 (bpm) | Berhasil |
| 75 (bpm) | Berhasil |
| 73 (bpm) | Berhasil |
| 89 (bpm) | Berhasil |
| 83 (bpm) | Berhasil |
| 72 (bpm) | Berhasil |
| 84 (bpm) | Tidak Berhasil |
| 85 (bpm) | Berhasil |
| 79 (bpm) | Berhasil |

Seperti pada tabel 4.3, dimana didapatkan nilai-nilai *heart rate Mi Band 2* untuk mengubah kecepatan *audio* dari cepat ke normal. Pada tabel tersebut didapatkan hasil bahwa dari 20 data yang tertera diantaranya 18 berhasil dalam memproses kecepatan *audio* normal ke cepat sedangkan 2 diantaranya tidak berhasil dikarenakan terjadinya *bug* dalam proses dan tidak mengeluarkan suara dari *audio* yang dimainkan walaupun proses terus berjalan pada terminal.

4.8.4 Kecepatan *Audio* dari Lambat ke Normal

Tabel 4.4: Pengujian *audio* dari lambat ke normal

| <i>Heart Rate</i> (bpm) | Perubahan Tempo <i>Audio</i> |
|-------------------------|------------------------------|
| 70 (bpm) | Berhasil |
| 75 (bpm) | Berhasil |
| 77 (bpm) | Berhasil |
| 80 (bpm) | Berhasil |
| 82 (bpm) | Berhasil |
| 88 (bpm) | Berhasil |
| 90 (bpm) | Berhasil |
| 83 (bpm) | Tidak Berhasil |
| 74 (bpm) | Berhasil |
| 85 (bpm) | Berhasil |
| 75 (bpm) | Berhasil |
| 89 (bpm) | Berhasil |
| 71 (bpm) | Berhasil |
| 87 (bpm) | Berhasil |
| 73 (bpm) | Berhasil |
| 76 (bpm) | Berhasil |
| 86 (bpm) | Tidak Berhasil |
| 84 (bpm) | Berhasil |
| 88 (bpm) | Berhasil |
| 78 (bpm) | Berhasil |

Seperti pada tabel 4.4, dimana didapatkan nilai-nilai *heart rate Mi Band 2* untuk mengubah kecepatan *audio* dari lambat ke normal. Pada tabel tersebut didapatkan hasil bahwa dari 20 data yang tertera diantaranya 18 berhasil dalam memproses kecepatan *audio* normal ke cepat sedangkan 2 diantaranya tidak berhasil dikarenakan terjadinya *bug* dalam proses dan tidak mengeluarkan suara dari *audio* yang dimainkan walaupun proses terus berjalan pada terminal.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

1. Adanya *Bug* pada saat *audio* dan *Mi Band 2* dieksekusi secara bersamaan, namun dalam *Bug* ini tidak adanya keterangan *error* yang terjadi pada bagian mana, dan *audio* tetap berjalan namun tanpa suara.
2. Ketika *audio* akan diproses, adanya *delay* pada *audio* karena terlebih dahulu *audio* diolah kecepatannya sehingga ketika dimainkan *delay* hampir tidak terdengar.
3. Ketika *audio* dengan *Mi Band 2* dihubungkan adanya *delay* saat mengeksekusi kecepatan audio yang disesuaikan dengan detak jantung penggunaannya.
4. Guncangan terhadap *Mi Band 2* dapat mempengaruhi nilai *heart rate*.
5. Musik dapat memberikan pengaruh terhadap detak jantung penggunaannya dengan mengikuti kecepatan musik agar kembali normal.
6. Sistem protokol *bluetooth* memudahkan dalam mengatur dan mengontrol *audio* yang digabungkan dengan detak jantung penggunaannya sehingga dapat memberikan *output* dimana dapat mengatur *speed audio* yang disesuaikan dengan detak jantung agar detak jantung tetap stabil.
7. *Bluetooth* mampu menawarkan solusi yang cukup efektif dan efisien dimana pengguna dapat melakukan mentransfer data detak jantungnya, walaupun dengan jangkaun yang relatif terbatas, tergantung pada *output poweur* yang dimiliki.

5.2 Saran

Untuk pengembangan penelitian selanjutnya terdapat beberapa saran sebagai berikut :

1. Agar nilai pada *Mi Band 2* sesuai dengan *heart rate* maka menjaga kestabilan dari *Mi Band 2* dengan kata lain tidak menggoncangkan *Mi Band 2*.
2. Perlu adanya penelitian lebih lanjut mengenai "Bug" yang terjadi pada saat *audio* dan *Mi Band 2* dimainkan secara bersamaan.
3. Pengguna maupun masyarakat diharapkan untuk tetap berolahraga dan menjaga kesehatannya, karena faktor serta gaya hidup seseorang mampu mempengaruhi kebugaran jasmani. Dengan melakukan senam *aerobic* dapat meningkatkan kekuatan otot pada jantung dan paru-paru agar sistem kardiovaskuler tetap terjaga karena dengan seiring usia bertambah maka kebugaran pada tubuhpun akan berkurang atau dapat dikatakan berpengaruh pada kesehatan tubuh.
4. Penelitian ini masih dapat dikembangkan lagi agar lebih sempurna oleh penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] M. Goto and Y. Muraoka, "A beat tracking system for acoustic signals of music," in Proceedings of the second ACM international conference on Multimedia, pp. 365–372, ACM, 1994. (Dikutip pada halaman 2).
- [2] "Stimulasi denyut jantung dengan pemutar musik pada android." <https://docplayer.info/32000655-Stimulasi-denyut-jantung-dengan-pemutar-musik-pada-android.html>. Terakhir diakses pada tanggal 26 Juni. (Dikutip pada halaman 2, 10).
- [3] N. Saputri, "Modifikasi skala waktu rekaman suara menggunakan metode wsola," graduate's thesis, Institut Teknologi Sepuluh Nopember, Januari 2018. (Dikutip pada halaman 3, 12).
- [4] "Manfaat latihan olahraga aerobik terhadap kebugaran fisik manusia." <https://media.neliti.com/media/publications/59907-ID-none.pdf>. Terakhir diakses pada tanggal 26 Juni. (Dikutip pada halaman 3).
- [5] "Pengaruh terapi musik alam terhadap frekuensi denyut jantung pada pasien selama operasi dengan anestesi spinal di rsud pandan arang boyolali." [file:///home/afif/Downloads/240-Article Terakhir](file:///home/afif/Downloads/240-Article%20Terakhir) diakses pada tanggal 11 Juni. (Dikutip pada halaman 9, 11).
- [6] "Overlap-save and overlap-add filters: Optimal design and comparison." <https://ieeexplore.ieee.org/document/5419990>. Terakhir diakses pada tanggal 23 Mei. (Dikutip pada halaman 13).
- [7] "Time and pitch scaling in audio processing." <https://www.surina.net/article/time-and-pitch-scaling.html>. Terakhir diakses pada tanggal 23 Mei. (Dikutip pada halaman 14).

- [8] “Understanding pitch synchronous overlap-add (psola).” <http://dafx.org/understanding-pitch-synchronous-overlap-add-psola/>. Terakhir diakses pada tanggal 23 Mei. (Dikutip pada halaman 15).
- [9] “An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech.” <https://ieeexplore.ieee.org/abstract/document/319366>. Terakhir diakses pada tanggal 23 Mei. (Dikutip pada halaman 16).
- [10] “Konversi suara digital dengan menggunakan algoritma *Waveform Similarity Overlap-Add* (wsola).” <http://etheses.uin-malang.ac.id/8324/1/08650049.pdf>. Terakhir diakses pada tanggal 11 Juni. (Dikutip pada halaman 18).
- [11] D. A. Habibi, “Kontrol kecepatan treadmill otomatis berdasarkan data sensor ppg (photoplethysmogram) pada smartwatch menggunakan single board computer,” tugas akhir, Institut Teknologi Sepuluh Nopember, Juli 2018. (Dikutip pada halaman 18).
- [12] “Teori musik 1.” <http://staffnew.uny.ac.id/upload/131782842/-pendidikan/diktat-teori-musik-1.pdf>. Terakhir diakses pada tanggal 10 Juni. (Dikutip pada halaman 20).
- [13] “Aplikasi dan tinjauan teknis *Bluetooth* untuk komunikasi tanpa kabel.” <https://media.neliti.com/media/publications/221574-aplikasi-dan-tinjauan-teknis-bluetooth-u.pdf>. Terakhir diakses pada tanggal 11 Juni. (Dikutip pada halaman 21).

BIOGRAFI PENULIS



Aff Nur Azizah yang disapa Affif, lahir di Malang Jawa Timur pada tanggal 2 Oktober 1996. Merupakan anak kedua dari empat bersaudara. Penulis lulus dari SMP Negeri 1 Batukliang dan melanjutkan ke SMA Negeri 1 Batukliang. Penulis melanjutkan ke jenjang strata satu di Jurusan Teknik Multimedia dan Jaringan ITS yang sekarang telah berganti nama menjadi Departemen Teknik Komputer ITS. Dalam masa kuliah, penulis tertarik dengan pengembangan *Hardware* dan *Internet of Things (IoT)*. Selain itu penulis aktif berorganisasi, yaitu dengan menjadi staff Himpunan Mahasiswa Teknik Elektro pada bagian Riset dan Teknologi 2016-2017 dan menjabat sebagai asisten Lab B201 Telematik Departemen Teknik Komputer ITS 2017-2019.

Halaman ini sengaja dikosongkan