

Nama : Rizky Maulana Ramdhani

NIM : F1D022095

Tanggal : 21 Juni 2025

LAPORAN PROJECT PEMROGRAMAN VISUAL SYMPHORIA MUSIC PLAYER

A. Deskripsi Aplikasi

Symphoria Music Player adalah aplikasi desktop yang dikembangkan menggunakan Python dan framework PyQt5, dirancang untuk memberikan pengalaman pemutaran musik yang lengkap. Aplikasi ini mendukung pemutaran audio untuk format .mp3 dan .wav, pengelolaan daftar putar, visualisasi audio, penyesuaian equalizer, dan tampilan lirik yang tersinkronisasi menggunakan file .lrc. Aplikasi ini memiliki antarmuka modern dengan tema yang dapat diubah antara mode gelap dan terang, serta memanfaatkan basis data SQLite untuk pengelolaan lagu dan daftar putar yang efisien. Fitur utama meliputi impor/ekspor daftar putar, pengurutan lagu berdasarkan judul atau artis, dan pengaitan file lirik dengan lagu.

B. Proses Pengembangan Proyek

Pengembangan Symphoria Music Player mengikuti pendekatan terstruktur dengan desain modular dan prinsip pemrograman berorientasi objek. Berikut adalah langkah-langkah pengembangan:

1. Analisis Kebutuhan dan Desain:

- Mengidentifikasi fitur utama: pemutaran audio, pengelolaan daftar putar, equalizer, visualisasi audio, dan tampilan lirik.
- Merancang arsitektur modular dengan kelas terpisah seperti MusicPlayer, AudioProcessor, DatabaseManager, PlaylistWidget, Equalizer, AudioVisualizer, dan LyricsWidget.
- Merencanakan antarmuka responsif dengan dukungan tema gelap dan terang untuk pengalaman pengguna yang optimal.

2. Pengaturan Lingkungan Pengembangan:

- Menginstal pustaka: PyQt5 untuk antarmuka, mutagen untuk ekstraksi metadata, librosa dan sounddevice untuk pemrosesan audio, numpy dan scipy untuk analisis sinyal, serta sqlite3 untuk basis data.

- Mengonfigurasi basis data SQLite (music_library.db) untuk menyimpan metadata lagu dan daftar putar.

3. Implementasi Komponen Inti:

- **Jendela Utama (MusicPlayer):** Menggunakan QMainWindow untuk membangun jendela utama yang mengintegrasikan semua komponen antarmuka. Kelas ini mengatur tata letak, tombol kontrol (play, pause, next, dll.), dan inisialisasi komponen seperti equalizer dan visualisator.
- **Pengelolaan Basis Data (DatabaseManager):** Mengimplementasikan operasi CRUD untuk mengelola lagu dan daftar putar dalam SQLite. Contohnya, metode add_song mengekstrak metadata menggunakan mutagen dan menyimpannya ke tabel songs.
- **Pemrosesan Audio (AudioProcessor):** Menggunakan librosa.load untuk memuat file audio dan sounddevice.OutputStream untuk pemutaran real-time. Kelas ini juga mengintegrasikan equalizer untuk memodifikasi sinyal audio.
- **Komponen Antarmuka:**
 - **PlaylistWidget:** Mengelola daftar lagu dengan QListWidget, menampilkan metadata, dan mendukung pengurutan serta impor/ekspor.
 - **Equalizer:** Menggunakan slider QSlider untuk menyesuaikan pita frekuensi, dengan preset seperti Rock dan Pop.
 - **AudioVisualizer:** Menggunakan scipy.fft untuk analisis frekuensi dan QPainter untuk menggambar bilah visualisasi.
 - **LyricsWidget:** Mem-parsing file .lrc menggunakan regex dan menampilkan lirik tersinkronisasi dengan QLabel.
- **Tema:** Menggunakan stylesheet Qt untuk mendefinisikan gaya tema gelap dan terang, diterapkan melalui metode apply_dark_theme dan apply_light_theme.

4. Integrasi dan Koneksi Signal-Slot:

- Menggunakan mekanisme signal-slot PyQt5 untuk menghubungkan komponen, seperti song_selected dari PlaylistWidget yang memicu play_song di MusicPlayer.
- Mengintegrasikan AudioProcessor dengan AudioVisualizer dan Equalizer untuk pembaruan real-time.

5. Pengujian dan Debugging:

- Menguji fungsi seperti pemutaran lagu, pengelolaan daftar putar, dan sinkronisasi lirik.
- Memperbaiki masalah seperti buffer overflow pada pemrosesan audio dan kendala unik pada basis data.
- Memastikan kompatibilitas lintas platform dan konsistensi tema.

6. Finalisasi:

- Menambahkan ekspor daftar putar ke CSV dan JSON menggunakan modul csv dan json.
- Mengimplementasikan menu konteks untuk opsi seperti menambahkan lagu ke daftar putar.
- Mengoptimalkan kinerja untuk daftar putar besar dan audio berkualitas tinggi.

C. Penjelasan Kode

Berikut adalah penjelasan singkat beberapa bagian kode penting:

1. **MusicPlayer (MusicPlayer class):**

```
def init_ui(self):
    self.setWindowTitle("Symphoria Music Player")
    self.setGeometry(100, 100, 1200, 900)
    central_widget = QWidget()
    self.setCentralWidget(central_widget)
    main_layout = QHBoxLayout(central_widget)
```

Kode ini menginisialisasi jendela utama dengan ukuran 1200x900 piksel dan mengatur tata letak utama menggunakan QHBoxLayout untuk membagi antarmuka menjadi panel kiri (daftar putar) dan kanan (kontrol, visualisator, dll.).

2. **AudioProcessor (AudioProcessor class):**

```
def set_audio_file(self, file_path):
    self.audio_data_buffer, self.sample_rate =
    librosa.load(file_path, sr=self.sample_rate, mono=True)
    self.equalizer.sample_rate = self.sample_rate
    self.equalizer.init_filters()
    self.current_file = file_path
    self.position = 0
    self.init_audio_output()
```

Metode ini memuat file audio menggunakan librosa, mengatur sample rate, dan menginisialisasi output audio dengan sounddevice. Data audio disimpan sebagai buffer untuk pemutaran.

3. DatabaseManager (DatabaseManager class):

```
def add_song(self, file_path):
    cursor = self.conn.cursor()
    audio_file = File(file_path)
    title = os.path.basename(file_path)
    artist = "Unknown Artist"
    album = "Unknown Album"
    duration = int(audio_file.info.length) if audio_file.info else 0
    cursor.execute('''
        INSERT OR REPLACE INTO songs
        (title, artist, album, duration, file_path, genre, year,
        lyrics_path)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)
    ''', (title, artist, album, duration, file_path, genre, year,
        None))
    self.conn.commit()
```

Metode ini mengekstrak metadata lagu menggunakan mutagen dan menyimpannya ke tabel songs dalam basis data SQLite.

4. AudioVisualizer (AudioVisualizer class):

```
def process_audio_data(self, audio_chunk):
    fft_data = np.abs(fft(audio_chunk))
    fft_data = fft_data[:len(fft_data)//2]
    bar_count = len(self.bars)
    chunk_size = max(1, len(fft_data) // bar_count)
    for i in range(bar_count):
        start_idx = i * chunk_size
        end_idx = min(start_idx + chunk_size, len(fft_data))
        magnitude = np.mean(fft_data[start_idx:end_idx])
        height = min(int(magnitude * 1000), self.height() - 10)
```

Kode ini memproses data audio menggunakan FFT (Fast Fourier Transform) untuk menghitung magnitudo frekuensi, yang kemudian digunakan untuk menggambar bilah visualisasi.

D. Fungsi Utama Aplikasi

Aplikasi ini menyediakan fitur-fitur berikut:

1. Pemutaran Audio (AudioProcessor):

- Memutar file .mp3 dan .wav dengan kontrol putar, jeda, hentikan, lagu berikutnya/sebelumnya, acak, dan ulang.
- Mendukung pengaturan volume dan navigasi posisi lagu.

2. Pengelolaan Daftar Putar (PlaylistWidget):

- Menampilkan lagu dengan sampul, judul, dan artis.
- Mendukung penambahan/penghapusan lagu, pembuatan daftar putar, dan ekspor ke CSV/JSON.

3. Equalizer (Equalizer):

- Menyediakan delapan pita frekuensi (60Hz–12kHz) dengan preset (Rock, Pop, dll.) dan penyesuaian manual.

4. Visualisasi Audio (AudioVisualizer):

- Menampilkan bilah frekuensi real-time dengan warna sesuai tema (merah untuk bass, hijau untuk tengah, biru untuk treble).

5. Tampilan Lirik (LyricsWidget):

- Memuat dan menampilkan lirik .lrc yang tersinkronisasi dengan posisi pemutaran.

6. Pengelolaan Basis Data (DatabaseManager):

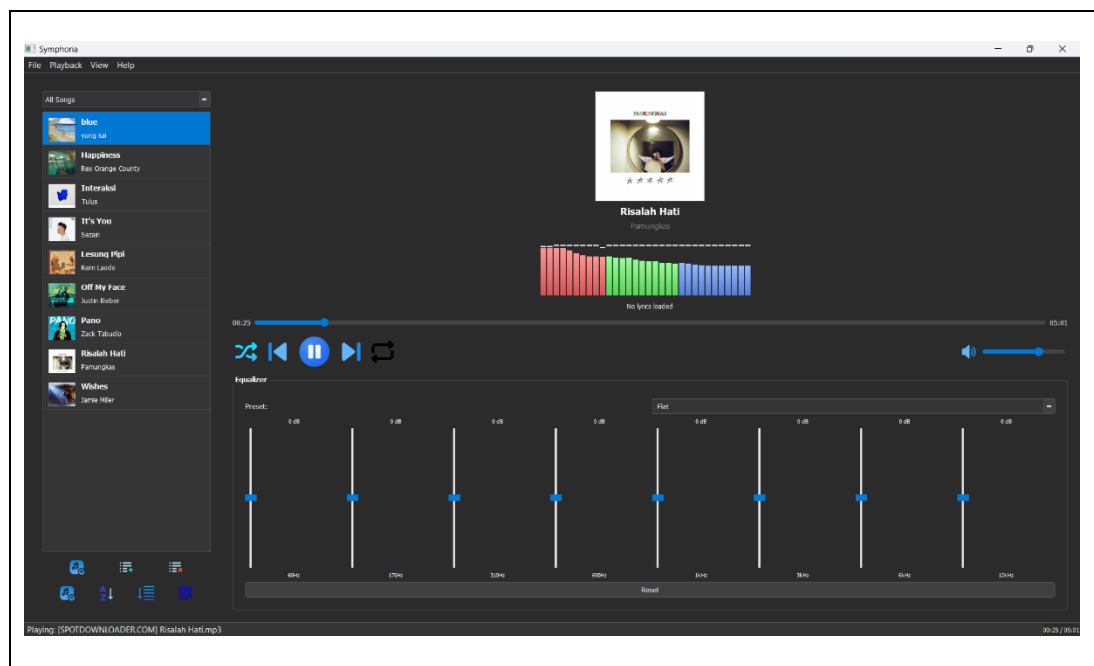
- Mengelola metadata lagu dan daftar putar dengan SQLite.

7. Penggantian Tema:

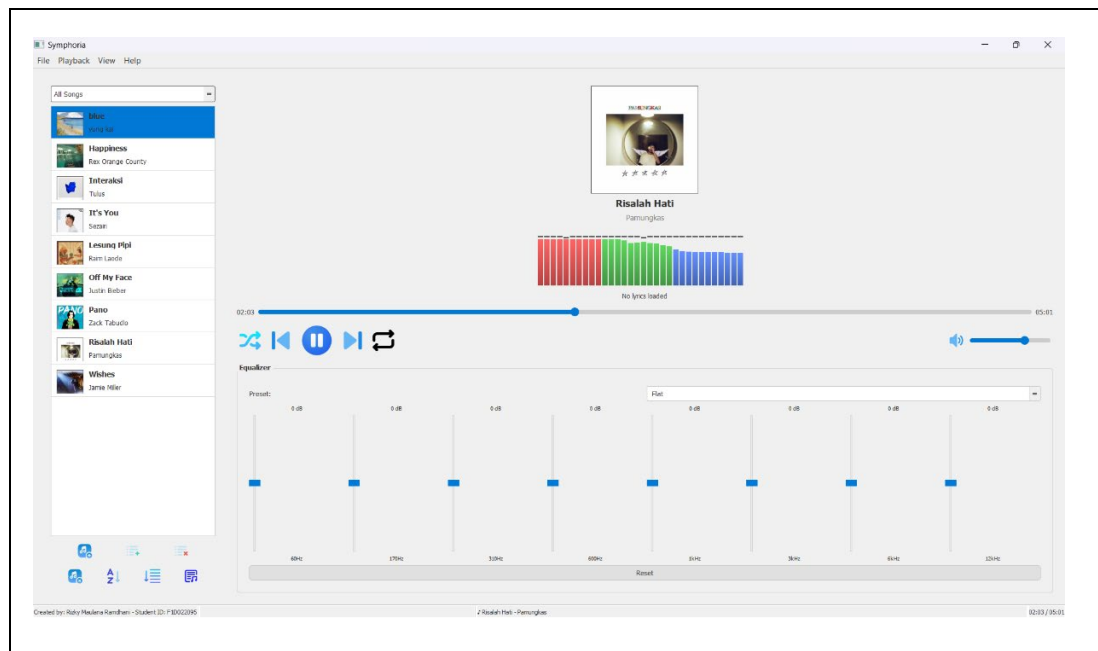
- Mengalihkan antara tema gelap dan terang dengan stylesheet Qt.

E. Screenshot Antarmuka dan Fungsionalitas

1. Antarmuka Utama (Tema Gelap):

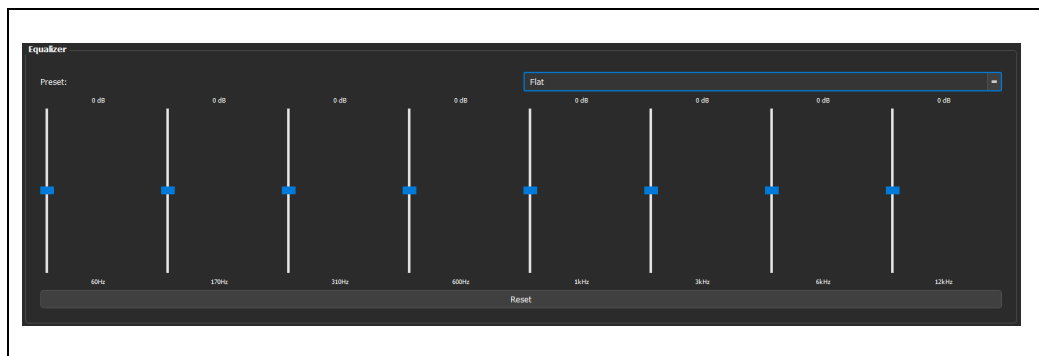


2. Antarmuka Utama (Tema Terang):

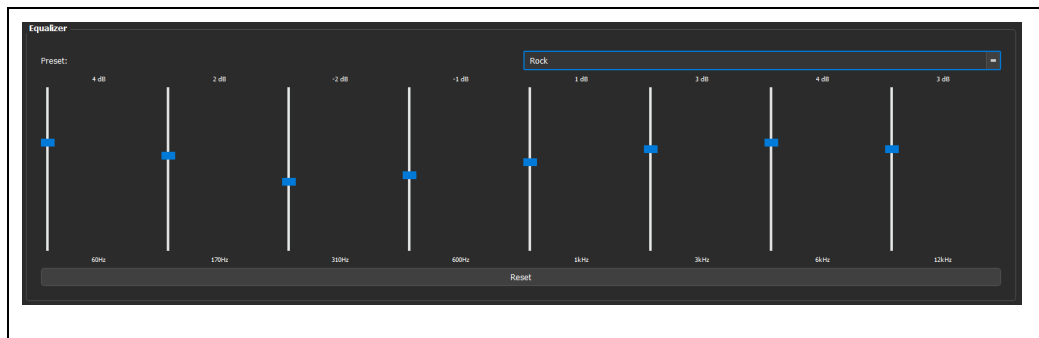


3. Pengaturan Equalizer:

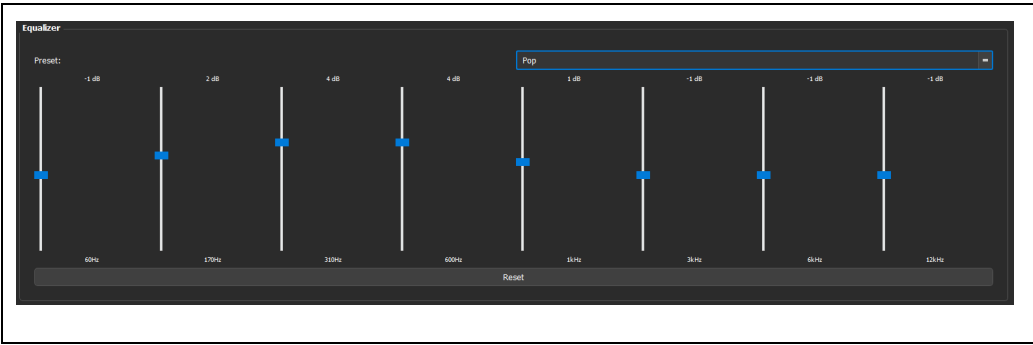
- Preset Flat



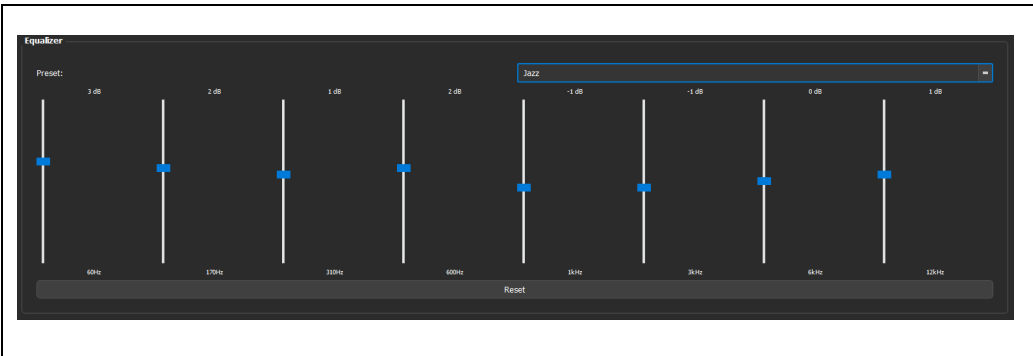
- Preset Rock



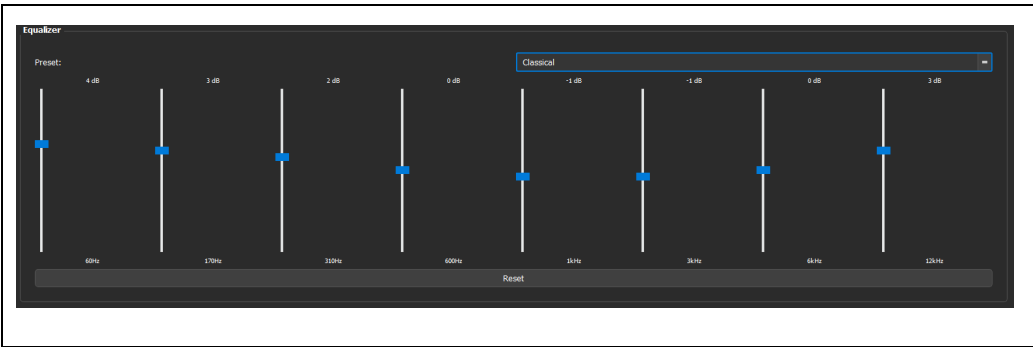
- Preset Pop



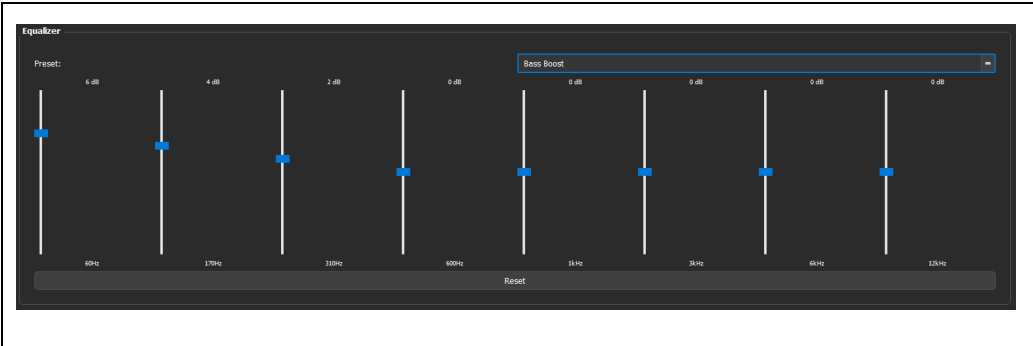
- Preset Jazz



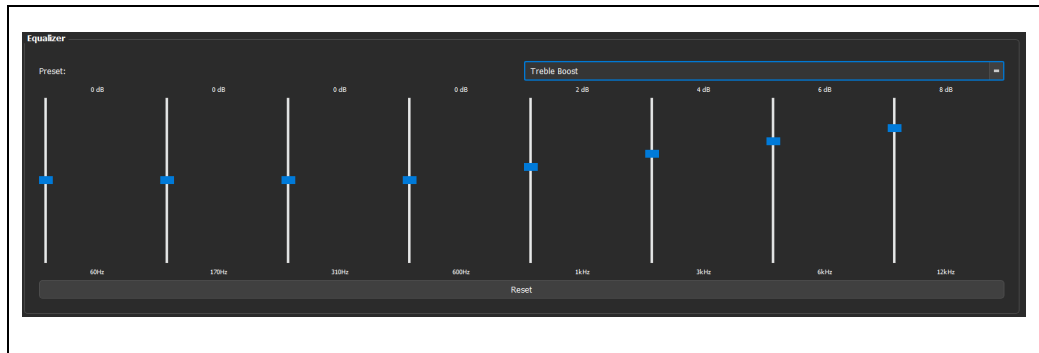
- Preset Classical



- Preset Bass Boost

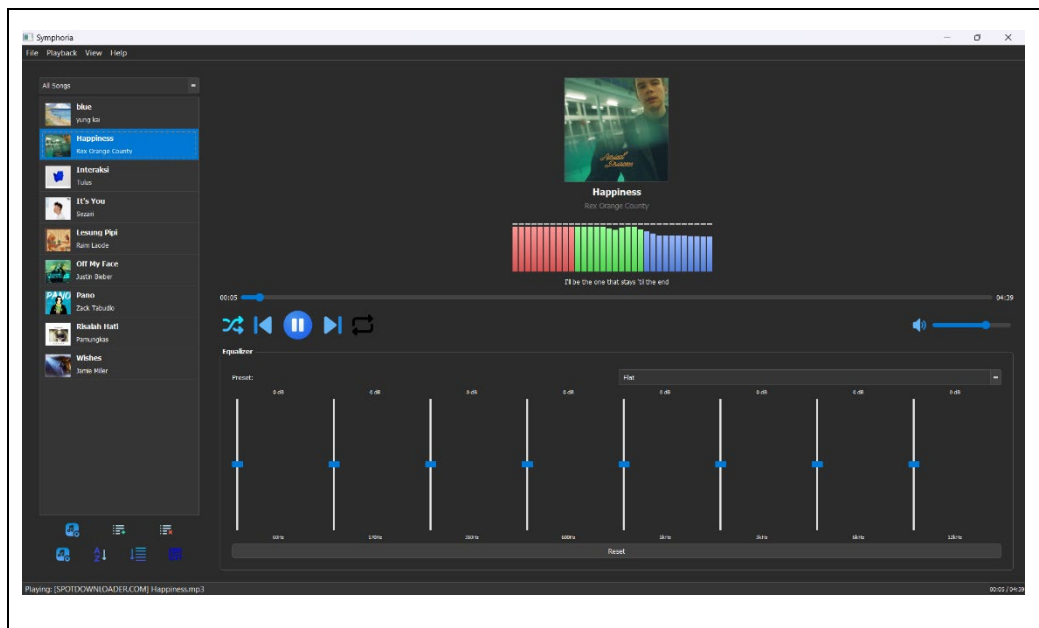


- Preset Treble Boost

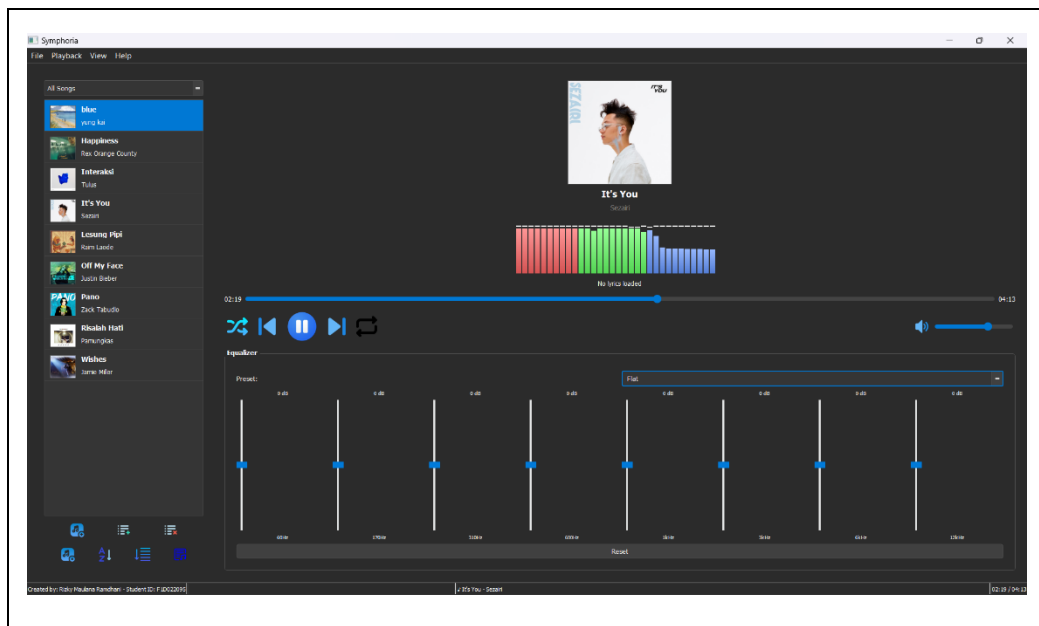


4. Tampilan Lirik:

- Lagu Dengan Lirik



- Lagu Tanpa Lirik



F. Kesimpulan

Symphoria Music Player berhasil menghadirkan pengalaman pemutaran musik yang kaya fitur dengan antarmuka yang ramah pengguna. Desain modular memastikan kemudahan perawatan, dan basis data SQLite mendukung pengelolaan data yang efisien. Fitur seperti visualisasi audio dan lirik tersinkronisasi menjadikannya alat serbaguna. Peningkatan di masa depan dapat mencakup dukungan format audio tambahan, integrasi streaming online, dan efek audio canggih seperti reverb.