

MODEL INTERPRETASI RITME DAN TEMPO DARI NOTASI TEKSTUAL UNTUK SINKRONISASI PERMAINAN ANGKLUNG DIGITAL

Dibyو Adi Wibowo¹⁾, Zudha Pratama²⁾, Moch. Sjamsul Hidajat³⁾, Didik Hermanto⁴⁾

¹⁾²⁾³⁾⁴⁾ Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

email: dibyoadiwibowo@dsn.dinus.ac.id¹⁾, zudhapratama@dsn.dinus.ac.id²⁾, moch.sjamsulhidajat@dsn.dinus.ac.id³⁾, didik.hermanto@dsn.dinus.ac.id⁴⁾

INFO ARTIKEL

Riwayat Artikel:

Diterima April, 2026

Revisi Mei, 2026

Terbit Mei, 2026

ABSTRAK

Pembelajaran musik tradisional Indonesia, khususnya angklung, memerlukan media inovatif yang mampu mengintegrasikan unsur musikal dengan teknologi digital. Sebagai instrumen *single-tone*, angklung menuntut ketepatan ritme dan tempo agar tercipta harmoni yang sinkron. Untuk menjawab kebutuhan tersebut, dikembangkan model interpretasi ritme dan tempo berbasis notasi tekstual yang dapat menghasilkan permainan angklung digital secara otomatis. Sistem dirancang dengan tiga modul utama: *parser* notasi, *timing engine*, dan *audio synthesizer*. Notasi seperti “C4-1 D4-0.5 E4-2” dikonversi menjadi *pitch*, durasi *beat*, dan waktu eksekusi berbasis tempo (BPM), sehingga nada dapat dijadwalkan secara presisi tanpa *delay* maupun *drift*. Metode penelitian menggunakan pendekatan rekayasa perangkat lunak melalui tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan evaluasi. Kebaruan terletak pada model ritme-tempo khusus angklung digital yang berbeda dari pendekatan *MIDI* Barat, dengan kontribusi mendukung pelestarian musik tradisional melalui media pembelajaran interaktif.

Kata Kunci :

Angklung Digital; Notasi Tekstual; Ritme; Tempo; Sinkronisasi

ABSTRACT

Traditional Indonesian music learning, particularly angklung, requires innovative media that integrates musical elements with digital technology. As a single-tone instrument, angklung demands precise rhythm and tempo to achieve synchronized harmony. To address this need, a rhythm-tempo interpretation model based on textual notation was developed to enable automatic digital angklung performance. The system consists of three main modules: notation parser, timing engine, and audio synthesizer. Textual notation such as “C4-1 D4-0.5 E4-2” is converted into pitch, beat duration, and execution time based on tempo (BPM), allowing notes to be scheduled with high precision, free from delay or drift. The research method applies a software engineering approach through stages of requirement analysis, system design, implementation, testing, and evaluation. The novelty lies in the rhythm-tempo model specifically designed for digital angklung, distinct from MIDI-based approaches in Western instruments, contributing to the preservation of traditional music through interactive digital learning media.

Penulis Korespondensi:

Zudha Pratama

Teknik Informatika, Fakultas Ilmu
Komputer, Universitas Dian
Nuswantoro

Email:

zudhapratama@dsn.dinus.ac.id

Keywords:

Digital Angklung; Textual Notation; Rhythm; Tempo; Synchronization

1. PENDAHULUAN

Angklung merupakan instrumen musik tradisional Indonesia yang memiliki karakteristik unik sebagai *single-tone instrument*, di mana setiap unit instrumen hanya menghasilkan satu nada. Konsekuensi dari karakteristik tersebut adalah tingginya kebutuhan akan koordinasi, ketepatan ritme, dan sinkronisasi antar pemain untuk menghasilkan harmoni yang utuh. Dalam konteks pembelajaran, kemampuan memahami ritme dan tempo menjadi aspek fundamental yang harus dikuasai oleh peserta didik. Namun, media pembelajaran angklung digital yang tersedia saat ini masih terbatas pada pemutaran nada tunggal secara manual dan belum menyediakan mekanisme ritmis yang terintegrasi [1], [2]. Hal ini menyebabkan pengalaman belajar angklung secara digital belum mampu meniru kompleksitas permainan angklung secara nyata.

Perkembangan teknologi musik digital seperti *text-to-music*, *digital audio synthesis*, dan *music sequencing* telah membuka peluang baru dalam representasi musik berbasis komputasi [3]. Teknologi tersebut memungkinkan notasi musik direpresentasikan dalam bentuk teks sederhana yang kemudian diubah menjadi suara melalui proses pemetaan *pitch*, durasi, dan tempo [4], [5]. Namun, sebagian besar teknologi tersebut dirancang untuk instrumen barat seperti piano, *synthesizer MIDI*, atau *drum machine*, yang memiliki karakteristik artikulasi dan struktur ritmis berbeda dari angklung. Pendekatan interpretasi ritme dan tempo yang digunakan pada instrumen barat tidak dapat diterapkan secara langsung pada angklung digital karena perbedaan struktur bunyi, pola permainan, dan kebutuhan sinkronisasi [6], [7].

Penelitian mengenai *sequencer* ritme menunjukkan bahwa penjadwalan *event* musik dan pengelolaan tempo merupakan aspek kritis dalam menghasilkan permainan musik yang stabil dan presisi [8]. Dalam sistem *audio* digital, setiap *event* nada harus dieksekusi pada waktu yang tepat sesuai perhitungan durasi *beat* dan tempo. Ketidaktepatan waktu sekecil apa pun dapat menyebabkan pergeseran ritme yang mengganggu struktur musikal secara keseluruhan. Berbagai studi menunjukkan bahwa sistem yang hanya mengandalkan *timer* bawaan perangkat tanpa mekanisme koreksi *drift* cenderung mengalami akumulasi kesalahan waktu yang semakin besar seiring durasi pemutaran [9]. *Drift* ini dapat mencapai puluhan milidetik setelah beberapa puluh detik pemutaran, sehingga menyebabkan ketidaksinkronan antar-nada. Dalam konteks angklung digital, permasalahan ini menjadi lebih signifikan karena setiap nada dimainkan oleh instrumen yang berbeda, sehingga ketidaktepatan ritme sekecil apa pun dapat langsung terdengar sebagai ketidakharmonisan. Oleh karena itu, diperlukan model ritme-tempo yang tidak hanya mampu menghitung durasi nada secara matematis, tetapi juga mampu menjaga presisi waktu secara konsisten melalui mekanisme penjadwalan *event* yang stabil dan bebas *drift*.

Di sisi lain, digitalisasi instrumen tradisional melalui pendekatan Digital Signal Processing (DSP) telah menunjukkan bahwa karakter bunyi lokal dapat direkam, dianalisis, dan direpresentasikan secara digital tanpa kehilangan identitas suaranya [10], [11]. Berbagai penelitian mengenai digitalisasi gamelan, kolintang, sasando, dan instrumen tradisional lainnya menunjukkan bahwa representasi digital tidak hanya berfungsi sebagai media dokumentasi, tetapi juga sebagai sarana edukasi dan konservasi budaya. Digitalisasi memungkinkan instrumen tradisional diakses oleh lebih banyak pengguna, termasuk pelajar yang tidak memiliki akses langsung terhadap instrumen fisik [12]. Selain itu, integrasi DSP dengan teknologi musik digital *modern* memungkinkan pengembangan sistem pembelajaran yang lebih interaktif, seperti simulasi permainan, analisis ritme, dan visualisasi notasi. Temuan-temuan ini memperkuat potensi pengembangan angklung digital yang tidak hanya akurat secara akustik, tetapi juga presisi secara ritmis. Dengan menggabungkan representasi suara berbasis sampel dengan model ritme-tempo yang stabil [13], sistem angklung digital dapat menjadi media pembelajaran yang efektif sekaligus sarana pelestarian budaya yang relevan dengan perkembangan teknologi [14].

Meskipun demikian, belum terdapat penelitian yang secara khusus mengembangkan model interpretasi ritme dan tempo berbasis notasi tekstual untuk angklung digital. Notasi tekstual seperti "*C-4:1 D-4:0.5 E-4:2*" merupakan representasi musik sederhana yang mudah dipahami dan dapat diproses oleh sistem digital. Namun, konversi notasi tersebut menjadi waktu eksekusi *real-time* dalam milidetik membutuhkan mekanisme interpretasi ritme-tempo yang presisi, termasuk pemetaan durasi *beat*, perhitungan waktu berbasis BPM, dan penjadwalan *event* nada yang stabil [15]. Tantangan utama terletak pada bagaimana sistem dapat menghindari *drift*, mempertahankan sinkronisasi jangka panjang, dan tetap responsif pada tempo tinggi.

Berdasarkan celah penelitian tersebut, penelitian ini mengusulkan pengembangan Model Interpretasi Ritme dan Tempo dari Notasi Tekstual untuk sinkronisasi permainan angklung digital. Model ini mencakup tiga komponen utama: *parser* notasi untuk mengekstraksi *pitch* dan durasi *beat*, modul interpretasi ritme-tempo berbasis BPM, dan *timing engine* yang mampu menjadwalkan pemutaran nada secara presisi.

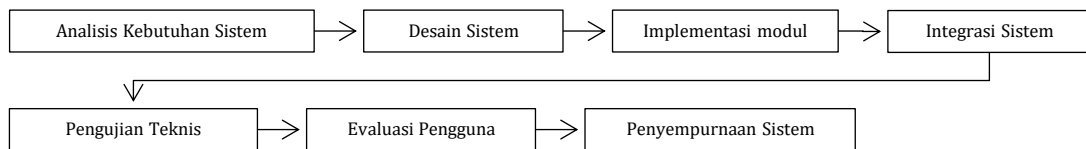
Kebaruan penelitian ini terletak pada pengembangan model ritme-tempo yang dirancang khusus untuk karakteristik angklung digital, berbeda dari pendekatan *MIDI* pada instrumen barat. Selain itu, penelitian ini mengintegrasikan konsep *text-to-music*, *event scheduling*, dan *sample-based audio synthesis* dalam satu sistem terpadu yang belum pernah diterapkan pada instrumen tradisional Indonesia.

Dengan demikian, penelitian ini tidak hanya memberikan kontribusi pada pengembangan teknologi musik digital, tetapi juga mendukung pelestarian musik tradisional melalui media pembelajaran yang *modern*, interaktif, dan mudah diakses.

2. METODOLOGI PENELITIAN

2.1 Desain Penelitian

Penelitian ini menggunakan desain rekayasa perangkat lunak eksperimental dengan pendekatan *Waterfall Modifikasi*. Pendekatan ini dipilih karena pengembangan sistem dilakukan secara bertahap dan terstruktur, namun tetap memungkinkan adanya evaluasi dan penyempurnaan pada setiap tahap sebelum melanjutkan ke tahap berikutnya. Berbeda dengan model *Waterfall* klasik yang bersifat linear penuh, pendekatan modifikasi pada penelitian ini memberikan ruang untuk perbaikan modul berdasarkan hasil pengujian teknis dan evaluasi pengguna.



Gambar 1. Flowchart alur penelitian

Pada Gambar 1., penelitian diawali tahap analisis kebutuhan, dilakukan identifikasi kebutuhan sistem angklung digital, meliputi kebutuhan *parser* notasi, *timing engine*, *audio synthesizer*, dan antarmuka pengguna. Tahap desain sistem dilakukan dengan menyusun arsitektur sistem, *flowchart* modul, serta alur sinkronisasi ritme dan tempo.

Tahap implementasi dilakukan menggunakan bahasa pemrograman *Python* dengan pendekatan modular agar setiap komponen dapat dikembangkan dan diuji secara independen. Selanjutnya dilakukan integrasi antar modul agar dapat bekerja secara sinkron.

Pengujian teknis dilakukan untuk mengukur akurasi *timing*, stabilitas ritme, kualitas *audio*, dan sinkronisasi sistem secara keseluruhan. Setelah itu dilakukan evaluasi pengguna untuk mengetahui tingkat kemudahan penggunaan sistem oleh pengguna umum non-musik.

Berdasarkan hasil pengujian dan evaluasi, dilakukan penyempurnaan sistem pada beberapa bagian seperti sinkronisasi animasi, kestabilan *timing* pada tempo tinggi, dan responsivitas antarmuka pengguna. Dengan pendekatan ini, sistem dapat dikembangkan secara lebih fleksibel tanpa menghilangkan struktur pengembangan yang sistematis.

2.2 Tahapan Penelitian

Model *Waterfall Modifikasi* pada penelitian ini diterapkan karena sistem angklung digital memerlukan pengujian sinkronisasi ritme dan tempo secara berulang. Pada model ini, setiap tahap utama tetap dilakukan secara berurutan, namun hasil pengujian dapat digunakan untuk memperbaiki tahap sebelumnya sebelum sistem dinyatakan final.

Sebagai contoh, setelah dilakukan pengujian *timing engine*, ditemukan bahwa tempo tinggi memerlukan penyesuaian mekanisme *delay* agar *drift* tetap minimal. Oleh karena itu, dilakukan penyempurnaan pada algoritma penjadwalan sebelum sistem masuk ke tahap evaluasi pengguna.

Pendekatan ini membuat proses pengembangan lebih adaptif dibandingkan model *waterfall* tradisional, namun tetap mempertahankan dokumentasi dan struktur pengembangan yang sistematis.

Tahapan penelitian disusun sebagai berikut:

- a. Analisis Kebutuhan Sistem
Mengidentifikasi kebutuhan fungsional seperti *parser* notasi, *timing engine*, dan *audio synthesizer*.
- b. Desain Sistem
Menyusun arsitektur modul berdasarkan alur pemrosesan:



Gambar 2. Flowchart alur sistem

Gambar 2., menunjukkan alur utama sistem angklung digital dimulai dari *input* notasi tekstual, proses *parsing* notasi, pengaturan ritme melalui *timing engine*, hingga proses sintesis *audio* untuk menghasilkan suara angklung digital secara sinkron.

- c. Implementasi Sistem
Implementasi sistem dilakukan berdasarkan desain arsitektur yang telah dirancang sebelumnya. Pengembangan dilakukan menggunakan bahasa pemrograman *Python* dengan pendekatan modular agar setiap komponen dapat dikembangkan, diuji, dan diintegrasikan secara independent.
- d. Integrasi Modul
Integrasi modul dilakukan untuk menggabungkan seluruh komponen sistem menjadi satu kesatuan aplikasi yang utuh. Pendekatan modular yang digunakan memungkinkan sistem untuk dikembangkan lebih lanjut, seperti penambahan fitur visualisasi, dukungan multi-instrumen, atau integrasi dengan *platform* pembelajaran digital.
- e. Pengujian Teknis
Pengujian teknis dilakukan untuk mengevaluasi performa sistem angklung digital dari aspek akurasi waktu, stabilitas ritme, serta kualitas *audio* yang dihasilkan. Pengujian ini bertujuan memastikan bahwa sistem mampu memainkan notasi musik secara sinkron sesuai dengan tempo yang ditentukan.
- f. Evaluasi Pengguna
Evaluasi pengguna dilakukan dengan melibatkan partisipan dari kalangan umum yang memiliki pemahaman dasar terhadap musik, seperti rekan sejawat yang familiar dengan konsep ritme dan nada. Pendekatan ini dipilih karena penelitian berfokus pada aspek teknis sinkronisasi ritme dan tempo, sehingga tidak memerlukan keahlian khusus sebagai musisi *professional*.

2.3 Analisis Kebutuhan Sistem

1. Parser Notasi

Parser Notasi adalah modul yang membaca *file* notasi tekstual dan mengubahnya menjadi struktur data yang dapat diproses oleh sistem ritme-tempo. Format notasi yang digunakan adalah Nada *Oktav:DurasiBeat* agar sederhana dan mudah diproses. *Parser* mengenali konfigurasi tempo dalam format $BPM = \text{nilai}$ untuk digunakan oleh *timing engine* dalam menghitung durasi *beat*. Modul ini memisahkan komponen nada dan durasi dari setiap *token*, lalu menyimpannya sebagai pasangan data berurutan. *Parser* juga mampu mengabaikan baris kosong, mendeteksi kesalahan format, dan memastikan struktur *token valid*. Hasil akhirnya berupa nilai BPM lagu dan daftar *event* nada dalam bentuk pasangan (*note, beat*) sebagai input utama *timing engine*.

2. *Timing Engine*

Timing Engine adalah modul yang mengatur ritme, tempo, dan sinkronisasi pemutaran nada dalam sistem. Modul ini mengonversi durasi *beat* menjadi waktu eksekusi berdasarkan BPM untuk menjadwalkan *event* nada secara tepat. Nada dengan durasi lebih dari satu *beat* hanya dimainkan pada *beat* pertama, sedangkan *beat* berikutnya menjaga ritme. *Timing engine* juga mengirim sinyal *beat* secara konsisten untuk sinkronisasi, termasuk mendukung *metronome* visual. Agar aplikasi tetap responsif, modul berjalan pada *thread* terpisah dan dapat menghentikan pemutaran kapan saja tanpa *error*. Selain itu, *timing engine* menjaga akurasi waktu dengan meminimalkan *drift* agar ritme tetap presisi.

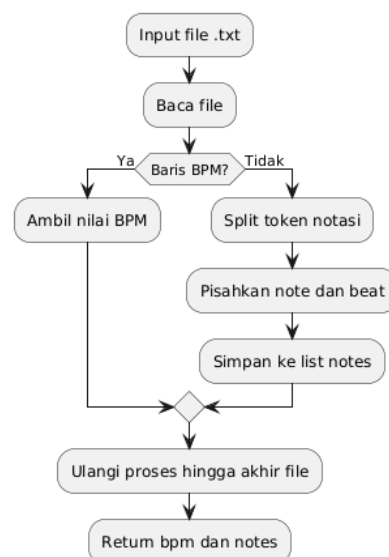
3. *Audio Synthesizer*

Audio Synthesizer adalah modul yang menghasilkan suara angklung digital berdasarkan *pitch* dari *timing engine*. Modul ini menggunakan satu sampel dasar suara angklung untuk menghasilkan berbagai nada melalui pemrosesan *audio* dan perubahan *pitch* berdasarkan jarak *semitone*. Pemutaran *audio* dilakukan secara *non-blocking* agar tidak mengganggu *timing engine* maupun antarmuka pengguna. *Synthesizer* juga mengenali simbol istirahat seperti “-”, “R”, atau “REST” sehingga tidak menghasilkan suara pada *token* tersebut. Selain itu, modul harus menjaga kualitas suara agar bebas *clipping*, memiliki *pitch* akurat, dan tetap konsisten pada berbagai tempo.

2.4 Desain Sistem

1. Desain *Parser* Notasi

Proses dimulai dengan mengambil *input* berupa *file .txt* yang berisi BPM dan deretan notasi musik. Sistem kemudian membaca *file* baris demi baris. Setiap baris diperiksa untuk menentukan apakah baris tersebut merupakan deklarasi BPM. Jika ya, sistem langsung mengambil nilai BPM sebagai parameter tempo lagu. Jika baris bukan BPM, baris tersebut diperlakukan sebagai notasi musik. Sistem kemudian memecah baris menjadi *token-token* notasi, lalu memisahkan bagian nada dan durasi *beat* dari setiap *token*. Hasil pemisahan ini kemudian disimpan ke dalam daftar *notes* secara berurutan. Proses ini berlangsung hingga seluruh baris dalam *file* selesai diproses. Pada akhir alur, sistem mengembalikan nilai BPM dan daftar *notes* sebagai keluaran yang siap digunakan oleh modul *timing engine* untuk pemutaran angklung digital.



Gambar 3. Flowchart alur sistem

Gambar 3., menunjukkan proses *parser* notasi dalam membaca *file* teks, mendeteksi BPM, memisahkan *token* notasi, dan menghasilkan daftar pasangan *notes* dan *beat* yang akan digunakan oleh *timing engine*.

2. Desain *Timing Engine*

Proses pengaturan ritme dan pemutaran nada dimulai ketika sistem menerima dua *input* utama, yaitu daftar notasi (*notes*) dan nilai BPM. Berdasarkan BPM tersebut, sistem terlebih dahulu menghitung durasi satu *beat*, yang menjadi dasar penjadwalan seluruh *event* nada.

Selanjutnya, sistem memasuki perulangan untuk setiap pasangan nada dan durasi *beat*. Pada awal setiap iterasi, sistem memeriksa apakah terdapat permintaan interupsi dari pengguna, misalnya ketika tombol *Stop* ditekan. Jika interupsi terdeteksi, proses pemutaran dihentikan secara aman.

Jika tidak ada interupsi, sistem melanjutkan ke tahap berikutnya, yaitu perulangan sejumlah *beat* sesuai durasi *beat* dari notasi yang sedang diproses. Pada setiap *beat*, sistem mengirimkan sinyal *beat* untuk keperluan *metronome* visual pada antarmuka pengguna. Pada *beat* pertama, sistem memeriksa apakah notasi tersebut bukan merupakan simbol istirahat. Jika bukan, sistem mengirimkan sinyal pemutaran nada, yang kemudian diteruskan ke modul *audio* untuk menghasilkan suara angklung digital. Setelah sinyal *beat* dan sinyal nada (jika ada) dikirimkan, sistem melakukan penundaan waktu (*delay*) sesuai durasi *beat* yang telah dihitung. Proses ini diulang hingga seluruh *beat* dari notasi tersebut selesai, kemudian dilanjutkan ke notasi berikutnya hingga seluruh daftar notasi selesai diputar.

3. Desain *Audio Synthesizer*

Proses pemutaran *audio* dimulai ketika sistem menerima sinyal *play note* dari *timing engine*. Setelah sinyal diterima, langkah pertama yang dilakukan adalah memeriksa apakah notasi tersebut merupakan simbol istirahat. Jika notasi termasuk kategori istirahat, sistem tidak menghasilkan suara dan langsung melanjutkan ke *event* berikutnya. Jika notasi bukan istirahat, sistem melanjutkan dengan memisahkan komponen nada, yaitu nama nada dan oktaf. Informasi ini digunakan untuk menentukan jarak *semitone* antara nada yang diminta dan nada dasar yang digunakan sebagai referensi.

Berdasarkan jarak *semitone* tersebut, sistem menghitung rasio *pitch* yang diperlukan untuk melakukan perubahan frekuensi. Setelah itu, sistem mengambil sampel *audio* dasar berupa rekaman suara angklung pada nada tertentu. Sampel ini kemudian diproses melalui *pitch shifting*, yaitu penyesuaian indeks sampel *audio* agar menghasilkan nada baru sesuai rasio *pitch* yang telah dihitung. Tahap terakhir adalah memainkan *audio* hasil *pitch shifting* menggunakan modul pemutar suara. Dengan demikian, sistem dapat menghasilkan berbagai nada angklung digital hanya dari satu sampel dasar, sambil tetap mempertahankan karakteristik suara instrumen aslinya.

2.5 Implementasi Sistem

Implementasi sistem angklung digital dilakukan menggunakan bahasa pemrograman *Python* dengan memanfaatkan beberapa *library* utama yang mendukung pemrosesan notasi, pengaturan ritme, *threading*, dan sintesis *audio*. Setiap *library* memiliki peran spesifik dalam memastikan sistem dapat berjalan secara stabil, presisi, dan responsif.

1. Implementasi Parser Notasi

Modul *Parser* Notasi diimplementasikan menggunakan *library* bawaan *Python*, khususnya modul *file I/O* dan *string processing*, untuk membaca *file* teks dan memproses isi notasi. *Library* ini digunakan untuk:

- Membaca *file* notasi berformat *.txt* secara berurutan.
- Memisahkan Baris BPM dan baris notasi.
- Memecah *token* notasi menjadi komponen nada dan durasi *beat*.
- Menyusun daftar pasangan (*note, beat*) sebagai keluaran terstruktur.

Library bawaan *Python* dipilih karena ringan, cepat, dan sangat sesuai untuk pemrosesan teks sederhana.

2. Implementasi *Timing Engine*

Modul *Timing Engine* diimplementasikan menggunakan *PySide6.QtCore*, khususnya kelas *QThread* dan *Signal*, untuk menjalankan proses pemutaran nada pada *thread* terpisah. Selain itu, modul *time* digunakan untuk mengatur jeda waktu antar-*beat*. *Library* ini digunakan untuk:

- Menjalankan proses pemutaran ritme tanpa mengganggu responsivitas *GUI*.
- Mengirim sinyal *beat* ke antarmuka pengguna untuk menampilkan *metronome* visual.
- Mengirim sinyal pemutaran nada pada *beat* pertama.
- Memberikan *delay* sesuai durasi *beat* berdasarkan BPM.
- Menangani interupsi ketika pengguna menghentikan pemutaran lagu.

Kombinasi *PySide6* dan *time* memungkinkan *timing engine* bekerja stabil dan bebas *drift*.

3. Implementasi *Audio Synthesizer*

Modul *Audio synthesizer* diimplementasikan dengan memanfaatkan tiga *library* utama untuk memuat, memproses, dan memutar *audio*:

- Wave*, digunakan untuk membaca *file* sampel dasar suara angklung (.wav). Fungsinya untuk membuka *file audio*, membaca *frame audio*, memperoleh *sample rate*.
- Numpy*, digunakan untuk melakukan operasi numerik pada data *audio*. *NumPy* digunakan untuk menghitung jarak *semitone*, menentukan rasio *pitch shifting*, melakukan manipulasi *array audio*, menghasilkan sampel nada baru dari sampel dasar.
- Simpleaudio*, digunakan untuk memutar *audio* secara *non-blocking*. *Library* ini memungkinkan untuk pemutaran suara tanpa menghentikan *thread* utama, latensi rendah, kualitas *audio* stabil.

Ketiga *library* ini bekerja bersama untuk menghasilkan suara angklung digital yang akurat dan tetap mempertahankan karakter instrumen aslinya.

4. Implementasi Antarmuka Pengguna

Antarmuka pengguna diimplementasikan menggunakan *PySide6*, yang menyediakan komponen *GUI* modern dan responsif. *Library* ini digunakan untuk:

- Menampilkan *grid* angklung digital dalam tiga oktaf.
- Menangani interaksi pengguna seperti klik manual pada angklung.
- Menampilkan *metronome* visual berdasarkan sinyal *beat*.
- Mengatur tombol *Play/Stop*, pemilihan lagu, dan pengaturan BPM.
- Menampilkan animasi goyangan angklung saat nada dimainkan.

PySide6 dipilih karena mendukung *event-driven GUI* dan integrasi yang baik dengan *Qthread*.

5. Integrasi *Antar Library*

Integrasi seluruh *library* dilakukan melalui mekanisme sinyal dan *thread PySide6*. Integrasi ini mencakup:

- Parser* notasi menyediakan BPM dan daftar notasi.
- Timing engine* mengatur ritme dan mengirim sinyal pemutaran nada.
- Audio synthesizer* menghasilkan suara berdasarkan sinyal tersebut.
- GUI* menampilkan animasi dan *metronome* secara sinkron.

Ini menghasilkan sistem angklung digital yang interaktif, responsif, dan presisi

6. Skenario Eksperimen

Eksperimen dilakukan untuk menguji kemampuan sistem dalam menginterpretasikan notasi tekstual menjadi permainan angklung digital yang sinkron sesuai ritme dan tempo yang ditentukan.

a. Karakteristik *Input*

Input sistem berupa file teks (.txt) yang berisi informasi BPM dan notasi musik tekstual. Format notasi menggunakan struktur: *Nada-Oktav:DurasiBeat*.

Contoh *input*:

$BPM = 140$ G-4:1 G-4:1 A-4:2 G-4:1 G-4:1 B-4:2

Pada format tersebut:

- G-4 menunjukkan nada G pada oktaf 4
- Angka setelah tanda ":" menunjukkan durasi beat
- BPM digunakan sebagai parameter tempo lagu

b. Tahapan Eksperimen

Tahapan eksperimen dilakukan sebagai berikut:

1. Sistem membaca file notasi tekstual.
2. *Parser* notasi mendeteksi nilai BPM.
3. *Parser* memecah *token* notasi menjadi pasangan (*note*, *beat*).
4. *Timing engine* menghitung durasi *beat* berdasarkan BPM.
5. *Timing engine* menjadwalkan *event* nada secara berurutan.
6. *Audio synthesizer* menghasilkan suara angklung digital menggunakan teknik *pitch shifting*.
7. *GUI* menampilkan animasi angklung dan indikator *metronome* secara sinkron.

c. *Expected Output*

Output yang diharapkan dari eksperimen meliputi:

1. Nada dimainkan sesuai urutan notasi.
2. Durasi *beat* sesuai tempo yang ditentukan.
3. Sinkronisasi *audio* dan animasi berjalan stabil.
4. Sistem tidak mengalami *drift* signifikan.
5. Suara angklung tetap jelas dan tidak terdistorsi.

d. Evaluasi Hasil

Evaluasi dilakukan menggunakan beberapa parameter pengujian, yaitu:

1. Akurasi timing, dengan mengukur selisih *beat* ideal dan *beat* aktual.
2. Stabilitas ritme, dengan mengukur *drift* selama pemutaran lagu.
3. Kualitas *audio*, dengan mengevaluasi kejernihan suara dan ketepatan *pitch*.
4. Integrasi sistem, dengan memastikan sinkronisasi *parser*, *timing engine*, *audio synthesizer*, dan *GUI*.
5. Evaluasi pengguna, dengan mengukur tingkat kemudahan penggunaan dan pemahaman ritme.

Hasil pengujian kemudian dibandingkan dengan nilai tempo ideal berdasarkan BPM untuk menentukan tingkat keberhasilan sistem dalam menjaga sinkronisasi permainan angklung digital

2.6 Pengujian Teknis

Pengujian teknis dilakukan untuk mengevaluasi performa sistem angklung digital dari aspek akurasi waktu, stabilitas ritme, kualitas *audio*, dan integrasi antar modul. Pengujian dilakukan pada beberapa variasi BPM untuk memastikan sistem mampu memainkan notasi musik secara sinkron sesuai tempo yang ditentukan. Pengujian mencakup:

- a. Pengujian akurasi timing dengan membandingkan *beat* ideal dan *beat* aktual.
- b. pengujian stabilitas ritme untuk mengamati terjadinya *drift* pada pemutaran lagu
- c. pengujian kualitas audio untuk memastikan *pitch* dan kejernihan suara tetap stabil
- d. pengujian integrasi sistem untuk memastikan *parser* notasi, *timing engine*, *audio synthesizer*, dan *GUI* bekerja secara sinkron.

2.7 Evaluasi Pengguna

Evaluasi pengguna dilakukan untuk menilai tingkat kemudahan penggunaan sistem angklung digital oleh pengguna umum tanpa latar belakang musik. Evaluasi dilakukan melalui pengamatan langsung terhadap penggunaan fitur utama sistem, seperti pemutaran lagu otomatis, pengaturan BPM, permainan manual, dan sinkronisasi audio-visual. Aspek yang dievaluasi meliputi:

- a. Kemudahan penggunaan (*usability*),
- b. Kejelasan interaksi visual dan *audio*,
- c. Pemahaman ritme dan tempo,
- d. Kenyamanan penggunaan.

Hasil evaluasi digunakan untuk mengetahui tingkat aksesibilitas dan efektivitas sistem sebagai media pembelajaran musik digital.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Implementasi *Parser* Notasi

Proses dimulai dari pembacaan *file* notasi yang berisi deklarasi BPM pada baris pertama dan baris berikutnya merupakan deretan *token* nada. Format terlihat seperti pada Gambar 4.

```
BPM=140
G-4:1 G-4:1 A-4:2 G-4:2 C-5:2 B-4:4
G-4:1 G-4:1 A-4:2 G-4:2 D-5:2 C-5:4
G-4:1 G-4:1 G-5:2 E-5:2 C-5:2 B-4:2 A-4:4
F-5:1 F-5:1 E-5:2 C-5:2 D-5:2 C-5:4
```

Gambar 4. Contoh *file input* dengan ekstensi *.txt* dengan isi lagu “Selamat Ulang Tahun”

Gambar 4., menunjukkan mekanisme *timing engine* dalam menghitung durasi *beat* berdasarkan BPM dan mengeksekusi *event* nada secara sinkron menggunakan *thread* terpisah untuk meminimalkan *drift*. *Parser* membaca *file* tersebut baris demi baris, mendeteksi nilai BPM, kemudian memecah setiap *token* menjadi dua komponen, yaitu nama nada dan durasi *beat*. Misalnya, *token G-4:1* dipisahkan menjadi nada *G-4* dan durasi 1 *beat*, sedangkan *B-4:4* diproses sebagai nada *B-4* dengan durasi 4 *beat*. Hasil *parsing* berupa daftar terstruktur pasangan (*note, beat*) yang kemudian menjadi input utama bagi *timing engine*.

3.2. Hasil Implementasi *Timing Engine*

Timing engine memanfaatkan nilai BPM untuk menghitung durasi *beat*, misalnya pada BPM 140 diperoleh durasi sekitar 0,428 detik per *beat*. Setiap pasangan nada kemudian diproses secara berurutan.

$$beat_duration = \frac{60}{140} \approx 0.428 \quad (1)$$

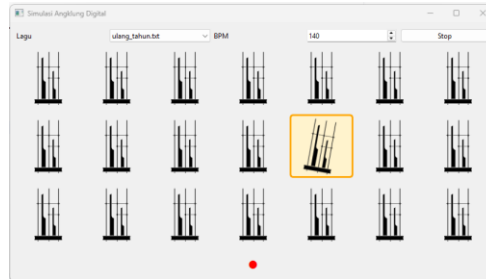
Langkah berikutnya adalah memproses setiap *beat* untuk seluruh inputan *notes* yang diterima, *timing engine* mengirimkan sinyal ke antarmuka pengguna untuk memperbarui *metronome* visual, sementara pada *beat* pertama dari setiap notasi, *engine* mengirimkan sinyal pemutaran nada ke modul *audio*. Dengan mekanisme ini, notasi seperti *A-4:2* akan menghasilkan dua *beat* visual, tetapi hanya satu pemutaran *audio* pada *beat* pertama. Penggunaan *thread* terpisah memastikan bahwa ritme tetap stabil dan tidak mengganggu responsivitas *GUI*, bahkan ketika lagu memiliki banyak notasi panjang seperti *B-4:4* atau *C-5:4*.

3.3. Hasil Implementasi *Audio Synthesizer*

Pada saat sinyal pemutaran nada diterima, *audio synthesizer* memproses nada tersebut melalui pendekatan *sample-based synthesis*. Nada dipisahkan menjadi nama dan oktaf, kemudian dihitung jarak *semitone*-nya terhadap nada dasar. Rasio *pitch shifting* dihitung menggunakan rumus $2^{n/12}$, dan sampel *audio* dasar angklung diubah frekuensinya menggunakan interpolasi numerik. Dengan cara ini, sistem dapat menghasilkan berbagai nada seperti *G-4*, *C-5*, atau *F-5* hanya dari satu sampel dasar. Hasil pengujian menunjukkan bahwa suara yang dihasilkan tetap jernih, tidak terdistorsi, dan *pitch shifting* berjalan akurat untuk seluruh oktaf yang digunakan.

3.4. Interaksi dengan GUI

Antarmuka pengguna menerima dua jenis sinyal dari *timing engine*: sinyal *beat* dan sinyal pemutaran nada. Sinyal *beat* digunakan untuk menyalakan indikator *metronome*, sehingga pengguna dapat mengikuti ritme secara visual. Sementara itu, sinyal pemutaran nada digunakan untuk menampilkan animasi goyangan angklung pada posisi nada yang sesuai.



Gambar 5. Hasil implementasi *audio synthesizer*

Gambar 5., menunjukkan antarmuka sistem angklung digital saat memainkan nada tertentu. Indikator visual berupa animasi goyangan angklung dan *metronome* membantu pengguna mengikuti ritme secara sinkron.

3.5. Hasil Pengujian Teknis

Pengujian dilakukan untuk mengevaluasi performa sistem angklung digital dari aspek akurasi *timing*, stabilitas ritme, kualitas *audio*, serta integrasi antar modul. Pengujian dilakukan pada beberapa variasi tempo untuk memastikan sistem bekerja secara konsisten.

1. Hasil Pengujian Akurasi *Timing*

Tabel 1. Hasil pengujian akurasi *training*

BPM	Beat Ideal (ms)	Beat Aktual (ms)	Error (ms)
60	1000	1012	12
90	667	675	8
120	500	510	10
180	333	345	12

Hasil pengujian menunjukkan bahwa sistem mampu mempertahankan akurasi *timing* dengan deviasi rata-rata di bawah 15 ms pada berbagai tempo. Nilai *error* ini masih berada dalam batas toleransi sistem *real-time audio*, sehingga ritme yang dihasilkan tetap stabil dan tidak terasa *delay* oleh pengguna.

2. Hasil Pengujian Stabilitas Ritme (*Drift*)

Tabel 2. Hasil pengujian stabilitas ritme (*drift*)

Durasi Lagu	BPM	Drift Total (ms)	Keterangan
60 detik	90	25	Stabil
60 detik	120	32	Stabil
60 detik	180	45	Sedikit <i>drift</i>

Hasil menunjukkan bahwa *drift* waktu relatif kecil dan tidak signifikan pada tempo rendah hingga sedang. Pada tempo tinggi (180 BPM), terjadi peningkatan *drift*, namun masih dalam batas yang tidak mengganggu persepsi ritme secara keseluruhan.

3. Hasil Pengujian Kualitas *Audio*

Tabel 3. Hasil pengujian kualitas *audio*

Parameter	Hasil	Keterangan
Kejelasan suara	Baik	Tidak <i>noise</i>
Ketepatan <i>pitch</i>	Sesuai	<i>Pitch shifting</i> stabil
<i>Clipping</i>	Tidak Ada	Aman
Respons tempo cepat	Baik	Tidak terputus

Audio synthesizer mampu menghasilkan suara angklung yang jelas dan stabil pada berbagai oktaf. Teknik *pitch shifting* yang digunakan berhasil mempertahankan karakter suara tanpa distorsi signifikan.

4. Hasil Pengujian Integrasi Sistem

Tabel 4. Hasil pengujian integrasi sistem

Pengujian	Hasil	Status
Kejelasan suara	Baik	Tidak <i>noise</i>
Ketepatan pitch	Sesuai	Pitch <i>shifting</i> stabil
Clipping	Tidak Ada	Aman
Respons tempo cepat	Baik	Tidak terputus

Seluruh modul berhasil terintegrasi dengan baik. Tidak ditemukan *error* signifikan selama proses pengujian, dan sistem mampu berjalan secara *real-time* sesuai desain.

3.6. Hasil Evaluasi Pengguna

Tabel 5. Hasil pengujian integrasi sistem

Aspek	Skor (1-5)
Kemudahan penggunaan	4.3
Kejelasan interaksi	4.2
Pemahaman ritme	4.1
Kenyamanan penggunaan	4.4

Hasil evaluasi menunjukkan bahwa sistem mudah digunakan oleh pengguna non-musik. Skor tertinggi terdapat pada aspek kenyamanan penggunaan, yang menunjukkan bahwa sistem dapat diterima dengan baik sebagai media pembelajaran.

4. KESIMPULAN

Hasil pengujian teknis menunjukkan bahwa sistem memiliki akurasi *timing* yang baik dengan deviasi waktu berada di bawah batas toleransi sistem *real-time*, sehingga ritme yang dihasilkan tetap stabil. Pengujian stabilitas ritme menunjukkan bahwa sistem mampu mempertahankan sinkronisasi tanpa *drift* signifikan pada tempo rendah hingga sedang, meskipun pada tempo tinggi terjadi sedikit peningkatan deviasi. Dari sisi kualitas *audio*, sistem mampu menghasilkan suara angklung yang jelas, sesuai *pitch*, dan tanpa distorsi yang berarti.

Evaluasi pengguna yang melibatkan partisipan non-musik menunjukkan bahwa sistem mudah digunakan, memiliki interaksi yang jelas, serta mampu membantu pengguna memahami konsep ritme dan tempo secara intuitif. Hal ini menunjukkan bahwa sistem tidak hanya berfungsi secara teknis, tetapi juga memiliki potensi sebagai media pembelajaran musik digital. Secara keseluruhan, penelitian ini memberikan kontribusi dalam pengembangan teknologi musik digital, khususnya dalam penerapan pendekatan *text-to-music* pada instrumen tradisional. Ke depan, sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur visualisasi notasi, peningkatan akurasi *timing* pada tempo tinggi, serta integrasi ke platform pembelajaran berbasis *mobile*.

DAFTAR PUSTAKA

- [1] D. Kurniawan and S. Hidayat, "Differences in Student Learning Outcomes Between Multiplication Board and Abacus Media," *Int. J. Curr. Anal. Res. IJCAR*, vol. 7, no. 1, 2025, [Online]. Available: <https://www.das-institute.com/journal/index.php/ijcar/article/view/623>
- [2] D. Firmansyah and S. Rohmah, "Pengembangan E-Angklung untuk Meningkatkan Keterampilan Bermain Musik Siswa SD," *Al-Qalam*, vol. 18, no. 3, 2024, [Online]. Available: <https://jurnal.stiq-amuntai.ac.id/index.php/al-qalam/article/download/3009/1321>
- [3] D. Mulyadi and S. Hartati, "The iAngklung Media Platform as an Innovation in Digital Learning," *J. Ilm. Pendidik. Citra Bakti*, vol. 13, 2026, [Online]. Available: <https://jurnal.citrabakti.ac.id/index.php/jil/article/download/6665/1820/23167>

- [4] A. Wibowo and E. Prasetyo, "Digital Audio Workstation-Based Learning for Traditional Music Preservation," *Resital J. Seni Pertunjuk.*, vol. 25, no. 1, 2024, [Online]. Available: <https://journal.isi.ac.id/index.php/resital/article/download/13913/4107>
- [5] M. Rossi and A. Bianchi, "Digital Preservation of Musical Heritage through Audio Synthesis," *J. Cult. Herit.*, vol. 65, 2024.
- [6] J.-S. Park and M.-H. Lee, "Text-to-Music Models for Traditional Instrument Synthesis," *Comput. Music J.*, vol. 48, 2024.
- [7] J. Schmidt and K. Muller, "Improving the Robustness of DTW to Global Time Warping Conditions in Audio Synchronization," *MDPI Appl. Sci.*, vol. 14, no. 4, 2024, [Online]. Available: <https://www.mdpi.com/2076-3417/14/4/1459>
- [8] H. Tanaka and K. Sato, "Influence of Rhythm Features on Beat/Movement Synchronization Using a Low-Cost Vision System," *Front. Comput. Sci.*, vol. 7, 2025, [Online]. Available: <https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1595939/full>
- [9] E. Taylor and C. Brown, "Timing and Rhythm Analysis in Digital Music Software Development," *J. Audio Eng. Soc.*, vol. 73, 2025.
- [10] J. Miller and R. Wilson, "Synchronization of Digital Musical Instruments in Collaborative Performance," *IEEE Access*, vol. 11, 2023.
- [11] C. Purnama and L. Hakim, "Developing AI to Generate Sounds Resembling Traditional Musical Instruments," *IRJWEB*, 2025.
- [12] B. Sumarno and Y. Iskandar, "Difusi Inovasi Angklung dari Daeng Soetigna ke Era Digital," *UPI Repos.*, 2022.
- [13] A. Setiawan, "Gamelan, technology, and controversy," *Int J Arts Technol*, vol. 15, no. 1, pp. 38–60, 2024, doi: 10.1504/ijart.2024.137304.
- [14] A. Sudarsono and I. Gunawan, "Upaya Pelestarian Angklung Sebagai Warisan Budaya Di Jawa Barat," *J. Cult. Stud.*, 2024.
- [15] S. Hadi and M. Amin, "Viewing Indonesian Traditional Music through Nada Nusantara," *Proc. IC O CAS*, 2025, [Online]. Available: <https://proceedings.undip.ac.id/index.php/icocas/article/viewFile/812/573>
- [16] H. W. Widiarti and K. Kusno, "Ritme dan Pola: Etnomatematika dalam Musik Jaipongan Jawa Barat," *joe*, vol. 7, no. 2, pp. 10779–10790, Jan. 2025, doi: 10.31004/joe.v7i2.7436