

Sistem Pelatih Keterampilan Alat Musik dengan Pitch Detection Algorithm

Muhammad Naufal Satriandana

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132
naufal.satriandana@gmail.com

Rinaldi Munir

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132
rinaldi@staff.stei.itb.ac.id

Abstrak—Terdapat beberapa aplikasi yang dapat melatih keterampilan alat musik sudah tersedia di perangkat keras seperti ponsel dan *desktop*, namun kurangnya fleksibilitas dari hal yang diajarkan menjadi suatu masalah bagi orang yang ingin menyesuaikan latihan sesuai kebutuhannya. Karena alasan tersebut, dibutuhkan sistem pelatih keterampilan alat musik yang interaktif dan dapat dimodifikasi sesuai kebutuhan. Untuk sebuah sistem yang fleksibel, diperlukan kebebasan dari pengguna untuk merancang, memodifikasi, dan memainkan latihannya sendiri sesuai keinginan. Fitur yang dibangun dalam sistem tidak hanya harus fleksibel, namun juga bersifat interaktif sehingga dibutuhkan kemampuan untuk mengetahui nada yang dimainkan pengguna secara *real time*. Untuk hal tersebut, dimanfaatkanlah sebuah algoritma yang bernama *pitch detection algorithm* yang dapat mendeteksi nada yang dimainkan oleh pengguna secara *real time*. Algoritma ini dimanfaatkan dalam sistem yang berbentuk aplikasi berbasis desktop dengan antarmuka grafis. Sistem tersebut menggunakan *Harmonic Product Spectrum* sebagai *pitch detection algorithm*. Sistem yang dibangun dapat mendeteksi 131 dari 132 kasus uji dari 3 buah instrumen yang berbeda, yakni gitar akustik, gitar elektrik, dan piano sehingga mendapatkan akurasi deteksi sebesar 99,2%. Sistem ini dapat melatih beberapa keterampilan, antara lain adalah memainkan deretan not, menghafal not, memainkan tangga nada dalam sebuah kunci, dan melatih interval dan teori musik lainnya terkait relasi not.

Kata kunci—musik, nada, *pitch detection algorithm*, *Harmonic Product Spectrum Introduction*

I. PENDAHULUAN

Walaupun seni musik merupakan salah satu topik yang sudah diajarkan sejak dini, kesulitan memperoleh sumber yang interaktif yang baik untuk latihan alat musik seperti guru les menjadi suatu masalah dalam pembelajaran musik. Walaupun sudah terdapat banyak sumber untuk belajar musik, salah satu hal yang dirasakan adalah tidak banyak sumber yang dapat mengajarkan musik secara interaktif seperti halnya guru les. Beberapa aplikasi yang dapat melatih keterampilan bermain alat musik sudah tersedia di perangkat keras seperti ponsel dan *desktop*, contohnya adalah *Yousician*, yang menawarkan pembelajaran interaktif untuk gitar, bas, piano, ukulele, dan bernyanyi [1]. Akan tetapi, kekurangan aplikasi-aplikasi demikian adalah kurangnya fleksibilitas hal-hal yang diajarkan

dan tidak dapat disesuaikan dengan kebutuhan. Dalam hal ini, pengguna terikat pada rangkaian modul ajar yang sudah tersedia pada aplikasi dan tidak bisa merancang latihannya sendiri sesuai keinginan. Oleh karena itu, dibutuhkan aplikasi pelatih keterampilan alat musik yang interaktif dan dapat dimodifikasi sesuai kebutuhan.

Dalam *Yousician*, secara umum, pengguna dapat berlatih mulai dari cara memegang alat musik, bermain lagu dan *riff* dengan alat musik, bermain akor dengan teknik *strumming*, menghafal not, berlatih irama, tangga nada, kunci, dinamika, beberapa teori musik yang diajarkan lewat video pembelajaran (e.g. bentuk akor), hingga keterampilan spesifik untuk sebuah alat musik (e.g. slide di gitar). Keterampilan-keterampilan tersebut dapat menjadi referensi untuk sistem pelatih keterampilan alat musik yang dikembangkan.

Dalam sistem pelatih keterampilan alat musik yang interaktif, dibutuhkan kemampuan sistem untuk dapat mengetahui not yang dimainkan oleh pemain alat musik dalam suatu waktu. Untuk mendeteksi sebuah nada, perlu adanya sebuah metode, lebih tepatnya, *pitch detection algorithm* (PDA) yang merupakan algoritma yang memiliki tujuan sederhana untuk mengekstraksi frekuensi fundamental (f_0) dari sinyal suara, yang biasanya merupakan komponen frekuensi terendah dari sinyal tersebut [2].

Kebutuhan lain dari sebuah sistem pelatih keterampilan musik yang interaktif adalah kemampuan mendeteksi nada yang sedang dimainkan secara *real time*. Hal tersebut membutuhkan *pitch detection algorithm* yang tepat untuk mendeteksi nada secara *real time*. Dengan sistem tersebut, berbagai keterampilan dapat dilatih. Sebagai contoh, dapat dibuat sebuah *play along* seperti halnya karaoke hanya saja untuk alat musik. Selain itu, keterampilan yang berdasarkan teori musik seperti latihan interval, improvisasi di dalam sebuah kunci dan menghafal posisi not pada instrumen juga terpikirkan. Dengan demikian, yang menjadi masalah adalah keterampilan apa saja yang dapat dilatih dengan sistem pelatih alat musik, serta bagaimana cara memanfaatkan PDA untuk membangun sistem pelatih keterampilan alat musik.

II. METODOLOGI

A. Pitch Detection Algorithm

Dalam pengembangan sistem pelatih alat musik yang interaktif, ditentukan terlebih dahulu *pitch detection algorithm* (PDA) yang dapat digunakan dalam situasi *real-time*, yakni PDA yang dapat yakni dengan tingkat akurasi yang relatif tinggi dan tingkat eror yang relatif rendah. Selain itu, dibutuhkan pula PDA dengan latensi dan *delay* yang rendah. Ditinjau 5 PDA sebagai alternatif PDA yang digunakan dalam pengembangan sistem, antara lain *Zero Crossing Rate*, *Autocorrelation*, *Maximum Likelihood*, *Yin*, dan *Harmonic Product Spectrum*. Dari penelitian yang telah dilakukan Amado & Vieira, walaupun *Zero Crossing Rate* dan *Autocorrelation* memiliki latensi yang relatif rendah, tingkat eror yang masih relatif tinggi menjadi alasan kedua PDA tersebut tidak dipilih sebagai PDA yang digunakan dalam sistem [3]. *Maximum likelihood* tidak cocok untuk seteman tetap, seperti alat musik dawai sehingga juga tidak dijadikan PDA untuk sistem [4]. HPS dan *Yin* menjadi kandidat PDA yang digunakan dalam sistem. Akan tetapi, karena banyaknya penelitian dan sistem serupa untuk HPS seperti alat setem gitar oleh Zurstraßen, HPS menjadi algoritma yang dipilih untuk sistem pelatih keterampilan alat musik [5].

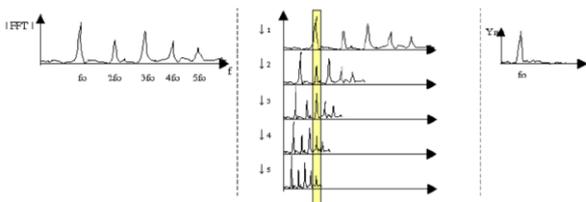
Cara kerja algoritma *Harmonic Product Spectrum* antara lain adalah mendeteksi peluang maksimum untuk frekuensi-frekuensi harmonik untuk setiap bingkai *spectral* dengan persamaan berikut.

$$Y(\omega) = \prod_{r=1}^R |X(\omega r)| \quad (1)$$

Dengan R merupakan jumlah harmonik yang akan diperhitungkan dan frekuensi ω_i adalah jangkauan frekuensi fundamental yang mungkin. Dari hasil produk harmonik, akan dicari nilai maksimum \hat{Y} , yang frekuensinya memiliki peluang terbesar sebagai frekuensi fundamental dengan persamaan berikut.

$$\hat{Y} = \max \{Y(\omega_i)\} \quad (2)$$

Algoritma HPS bekerja dengan cara men-*downsample spectra* dan *spectra* pengalinya. Hal tersebut diilustrasikan pada Gambar II.1. Untuk merepresentasikan sinyal dalam bentuk *spectra*, maka digunakan *Fast Fourier Transform* (FFT). Keunggulan dari HPS adalah biaya komputasi yang murah serta memiliki ketahanan terhadap *noise* [6].



Gambar II.1 Ilustrasi Harmonic Product Spectrum dengan puncak tertinggi hasil adalah frekuensi fundamental paling memungkinkan [6]

B. Pengembangan Sistem

Salah satu contoh sistem pelatih keterampilan alat musik adalah Yousician. Yousician merupakan perangkat lunak yang menawarkan pembelajaran interaktif untuk berbagai alat musik [1]. Dalam aplikasi Yousician, pengguna dapat belajar memainkan instrumen secara interaktif dengan bermain lagu-lagu yang tersedia pada aplikasi dengan cara mirip seperti karaoke. Pengguna dapat mengikuti path yang tersedia atau dapat juga bermain lagu yang dipilih atau melakukan latihan yang diinginkan, misalnya pelatih akor, tangga nada, hingga pelatih telinga. Fitur dan fungsionalitas yang dimiliki oleh Yousician relatif lengkap, namun penggunaannya dibatasi oleh ketersediaan latihan, kegiatan, serta daftar lagu yang sudah disediakan saja. Dalam hal ini, fitur yang tersedia tidak dapat diubah sesuai kemauan pengguna dan lagu keinginan pengguna belum tentu tersedia pada aplikasi. Pengguna yang ingin melatih beberapa not saja harus mengikuti fitur dan alat yang disediakan. Selain itu, pengguna yang memiliki keinginan latihan yang berbeda dari yang sudah tersedia oleh sistem tidak dapat merancang latihannya sendiri sehingga terdapat kekurangan aspek fleksibilitas.

Untuk solusi yang dikembangkan, digunakan perbandingan fitur dari aplikasi serupa yang sudah ada, yakni Yousician sebagai acuan. Aplikasi Yousician dijadikan acuan karena juga merupakan aplikasi yang dapat berinteraksi dengan pengguna secara *real time* serta mendeteksi not masukkan pengguna. Adapun hasil analisis fitur-fitur yang terdapat ataupun belum terdapat pada Yousician beserta fitur yang dirancang untuk difasilitasi pada pengembangan sistem tugas akhir ini tertera pada Tabel II.1.

Tabel II.1 Ketersediaan Fitur Yousician dan Sistem Rancangan

Fitur	Ketersediaan fitur	
	Yousician	Sistem rancangan
Mendeteksi not masukan	Tersedia	Tersedia
Memberikan umpan balik berupa kesesuaian not masukan dan not target	Tersedia	Tersedia
Mendeteksi akor masukan	Tersedia	Tidak tersedia
Menampilkan materi berupa video pembelajaran	Tersedia	Tidak tersedia
Pembelajaran berbentuk play along yang diiringi lagu	Tersedia	Tidak tersedia
Susunan modul-modul latihan yang berbentuk path	Tersedia	Tidak tersedia
Merancang sebuah latihan	Tidak tersedia	Tersedia
Memodifikasi latihan yang sudah ada	Tidak tersedia	Tersedia
Memberikan nama alternatif dari not yang ingin dilatih sesuai keinginan pengguna	Tidak tersedia	Tersedia

Dari PDA yang dipilih, yakni *Harmonic Product Spectrum*, sistem pelatih keterampilan alat musik yang dikembangkan merupakan sebuah aplikasi *desktop* dengan *graphical user interface*. GUI tersebut memungkinkan pengguna untuk memasukkan not target serta mendapatkan umpan balik secara langsung terkait kesesuaian not yang dimainkan dan not target. Pengguna dapat melakukan beberapa modifikasi terkait hal ingin dilatih, antara lain adalah modifikasi nama not, deskripsi latihan, serta cara pemilihan not oleh sistem. Alasan pemilihan *desktop* sebagai platform adalah karena kemampuannya untuk terhubung ke berbagai perangkat audio seperti *soundcard* dengan mudah. Aplikasi yang dibangun terdiri dari dua modul atau komponen pembangun utama. Komponen-komponen tersebut antara lain adalah komponen GUI dan komponen pendeteksi nada.

Komponen GUI merupakan antarmuka untuk interaksi pengguna dengan sistem pelatih keterampilan alat musik. Antarmuka yang dibangun akan memiliki beberapa halaman. Halaman-halaman tersebut antara lain adalah halaman utama, halaman pengaturan latihan, dan halaman latihan.

Komponen pendeteksi nada pada sistem yang dibangun merupakan komponen utama yang memungkinkan deteksi nada terjadi. *Pitch detection algorithm* yang dipilih, yakni HPS, dibangun pada komponen ini. Komponen pendeteksi nada menggunakan bahasa yang sama dengan komponen GUI, yakni Python, dan bersifat modular, yang dalam hal ini, komponen GUI dapat menggunakan fungsionalitas komponen pendeteksi nada tanpa mengetahui cara kerjanya. Pembangunan yang bersifat modular juga memudahkan pengembangan ke depannya jika diperlukan karena algoritma atau cara kerja modul pendeteksi nada dapat diubah tanpa mengubah fungsionalitas dan abstraksi modul tersebut. Selain itu, *pitch detection algorithm* yang digunakan pada komponen pendeteksi nada merupakan algoritma yang dibangun langsung pada bahasa pemrograman yang dipilih dan bukan merupakan *library*.

C. Pengujian

Pengujian terutama dilakukan terhadap komponen pendeteksi nada. Pengujian untuk performa *pitch detection algorithm* sudah pernah dilakukan pada penelitian-penelitian sebelumnya, seperti pada penelitian yang dilakukan oleh Amado & Vieira. Hal ini dapat dijadikan referensi dalam pengujian *pitch detection algorithm* pada sistem. Metode dari pengujian ini adalah sistem diberikan masukan berupa suara dari sebuah alat musik dengan not tertentu, kemudian dibandingkan kesesuaian not masukan dengan not hasil deteksi algoritma sebagai tingkat akurasi deteksi.

Selain pengujian pada *pitch detection algorithm*, pengujian lain juga dilakukan terhadap kemampuan sistem untuk melatih keterampilan alat musik. Untuk menguji hal ini, ditentukan terlebih dahulu keterampilan-keterampilan apa saja yang diuji serta metrik pengujian untuk setiap keterampilan tersebut. Berdasarkan aplikasi serupa *Yousician*, didapatkan keterampilan yang dapat diuji beserta metrik penilaian yang dapat disimpulkan dari aplikasi tersebut. Berikut merupakan keterampilan-keterampilan yang diuji dengan sistem.

- 1) Memainkan deretan not: Metrik penilaian agar sistem dikatakan dapat melatih memainkan sebuah deretan not

adalah jika diberikan sebuah deretan not kepada pengguna, sistem dapat mengetahui apakah pengguna sudah memainkan deretan not tersebut dengan benar.

- 2) Memainkan akor: Metrik penilaian agar sistem dikatakan dapat melatih memainkan akor adalah jika diberikan sebuah akor kepada pengguna, sistem dapat mengetahui apakah pengguna sudah memainkan akor tersebut dengan benar.
- 3) Menghafal not: Metrik penilaian agar sistem dikatakan dapat melatih menghafal not adalah jika diberikan sebuah not acak kepada pengguna, sistem dapat mengetahui apakah pengguna dapat menemukan not tersebut pada alat musiknya dan memainkannya.
- 4) Melatih irama: Metrik penilaian agar sistem dikatakan dapat melatih irama adalah jika diberikan sebuah deretan not beserta ketukannya kepada pengguna, sistem dapat mengetahui apakah pengguna sudah memainkan deretan not tersebut dengan ketukan yang sesuai.
- 5) Memainkan tangga nada dalam sebuah kunci: Metrik penilaian agar sistem dikatakan dapat melatih memainkan tangga nada adalah jika diberikan deretan not dalam sebuah tangga nada dalam sebuah kunci kepada pengguna, baik secara acak maupun urut dan baik dengan nama not atau sebutan not dalam tangga nada, sistem dapat mengetahui apakah pengguna sudah memainkan deretan not tersebut dengan benar.
- 6) Melatih dinamika: Metrik penilaian agar sistem dikatakan dapat melatih dinamika adalah jika sistem dapat mengetahui apakah not yang dimainkan oleh pengguna tergolong “keras” atau “lembut”.
- 7) Melatih interval dan teori musik lainnya terkait relasi not: Metrik penilaian agar sistem dikatakan dapat melatih interval adalah jika diberikan sebuah not kepada pengguna, kemudian diberikan interval tertentu secara acak dari not tersebut, sistem dapat mengetahui apakah pengguna sudah memainkan not yang dimaksud dengan benar.

Selain pengujian dengan beberapa metrik, dilakukan juga pengujian kepada pengguna dengan *Single-Ease Question (SEQ)* dan juga *System Usability Scale (SUS)*. Hal ini karena untuk sebuah sistem, dibutuhkan validasi pengguna.

III. IMPLEMENTASI SISTEM

Proses implementasi dilakukan pada sebuah perangkat komputasi dengan sistem operasi Windows. Selain itu, dalam implementasinya, digunakan juga sebuah perangkat audio beserta antarmuka audio yang digunakan untuk menangkap sinyal yang selanjutnya akan diolah oleh sistem. Adapun alat musik yang dicoba terhadap sistem antara lain adalah gitar listrik, gitar akustik, dan piano digital. Sinyal suara yang dihasilkan oleh gitar akustik dan piano digital akan ditangkap oleh perangkat audio yang terdapat pada perangkat komputasi tempat dijalankannya sistem, sedangkan sinyal yang dihasilkan oleh gitar listrik akan langsung ditangkap dengan antar muka audio yang berupa *multi effects pedal*.

Sistem yang dikembangkan menggunakan bahasa Python. Selain itu, sistem terdiri dari dua komponen, yakni komponen pendeteksi nada dan *graphical user interface*.

A. Implementasi Komponen Pendeteksi Nada

Komponen pendeteksi nada merupakan sebuah komponen yang dibuat dalam bahasa Python dan merupakan modul yang bertanggung jawab atas pengolahan sinyal suara dan mendeteksi nada dalam sinyal tersebut. Modul ini menggunakan *pitch detection algorithm* berupa *Harmonic Product Spectrum* untuk melakukan deteksi nada. Modul ini dapat digunakan dalam modul lain dalam bentuk *callback*, dalam hal ini, *callback* dapat dijadikan fungsi yang dipanggil oleh fungsi *streaming* dari *input*. Adapun library yang digunakan dalam komponen ini adalah Numpy dan SciPy.

Komponen pendeteksi nada memiliki satu *function* utama, satu *function* pendukung, serta beberapa variabel *global* yang dapat diatur oleh pengguna. Fungsi pendukung pada komponen ini bertujuan untuk menentukan not terdekat dari nada masukan. Fungsi ini menerima masukan berupa nada dalam bentuk float dengan satuan Hertz, dan mengembalikan not terdekat (e.g. A4, G#5) dan nada dari not tersebut dalam Hertz. Cara kerja fungsi ini adalah mencari tahu jarak *semitone* (n) dari nada masukan (f_i) dengan nada referensi (f_0), yang dalam hal ini merupakan *concert pitch* atau A4. Jarak *semitone* tersebut dicari dengan menggunakan persamaan berikut dengan f_0 sebagai nada referensi atau *concert pitch*, yakni 440Hz.

$$n = 12 \cdot \log_2\left(\frac{f_i}{f_0}\right) \quad (3)$$

Inti dari komponen pendeteksi nada terletak pada fungsi utama. Di dalam fungsi ini, diimplementasikan *pitch detection algorithm* yang dipilih, yakni *Harmonic Product Spectrum*.

Secara garis besar, alur dari fungsi utama mulai dari menerima sampel hingga mendeteksi nada dengan HPS adalah sebagai berikut.

- 1) Pendefinisian variabel statis: Pada saat fungsi callback pertama kali dipanggil dalam program, didefinisikan variabel sebagai representasi *window* yang merupakan sinyal yang dideteksi notnya.
- 2) Pemeriksaan status *stream*: Setiap awal siklus fungsi, diperiksa status dari *stream*, yang dalam hal ini, jika terdapat sebuah status yang menandakan error, fungsi memberitahukan bahwa tidak ada not terdeteksi dan melakukan *return*.
- 3) Penggabungan sampel ke *window*: Setiap siklus, sampel baru akan digabungkan ke akhir *window* dan menghapus sampel lama dari awal *window* dengan ukuran yang sama dengan sampel baru.
- 4) Pemeriksaan kekuatan sinyal: Jika *power* dari sinyal dalam *window* di bawah batas tertentu, fungsi memanggil memberitahukan bahwa tidak ada not terdeteksi dan melakukan *return*.

- 5) Penanganan kebocoran *spectral*: Kebocoran *spectral* pada sinyal dilakukan dengan mengalikan sinyal dengan jendela hann.
- 6) Transformasi sinyal ke bentuk *spectra*: Sinyal yang telah diolah dengan mengalikan fungsi jendela selanjutnya ditransformasi menggunakan transformasi Fourier cepat (FFT).
- 7) Pemrosesan *spectra*: Sebelum *spectra* digunakan untuk kalkulasi HPS, dilakukan pemrosesan dengan cara menghilangkan *noise*, yakni dengan memangkas frekuensi di bawah 62Hz, atau disebut *mains hum*, serta memangkas frekuensi yang memiliki energi di bawah batas tertentu yang dianggap sebagai *noise*.
- 8) Perhitungan HPS: *Spectra* yang sudah diproses kemudian diinterpolasi dan dikalikan sebanyak jumlah tertentu yang dapat didefinisikan pengguna untuk menghasilkan *spectra* akhir yang memiliki informasi terkait nada pada sinyal.
- 9) Pendeteksian not: Didapatkan frekuensi dengan nilai maksimum dari *spectra* hasil perhitungan HPS yang merupakan frekuensi dari nada yang dideteksi, kemudian not terdekat dari frekuensi tersebut dicari dengan dan dikembalikan.

B. Implementasi GUI

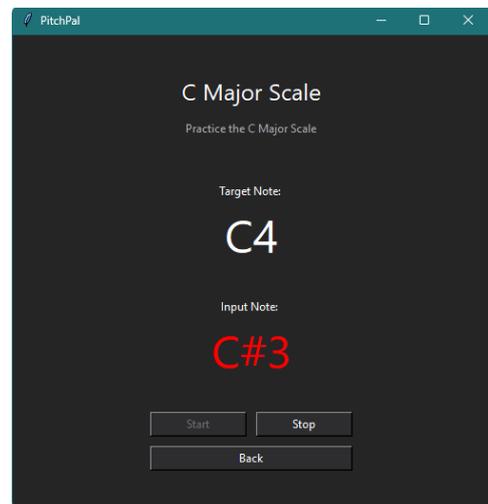
Graphical User Interface (GUI) diimplementasikan untuk sistem pelatih instrumen sebagai antarmuka untuk interaksi pengguna dengan sistem. GUI dikembangkan dalam bahasa Python dan terpisah dari komponen pendeteksi nada. *Library* utama yang digunakan untuk pengembangan GUI adalah Tkinter, yang merupakan *library* yang sudah tersedia dari Python untuk pengembangan antarmuka. Adapun *library* eksternal yang digunakan untuk pengembangan GUI antara lain adalah *sounddevice* untuk menangani perangkat audio masukan dan Numpy.

GUI dikembangkan dengan cara membuat berbagai kelas untuk tiap halaman. Selain itu, karena Tkinter bersifat *blocking*, yang dalam hal ini, tidak ada proses lain yang dapat berjalan bersamaan dengan GUI ketika ditampilkan, maka dikembangkan pula kelas yang menangani *threading* sehingga penanganan proses yang paralel dengan penampilan GUI, yakni streaming dari perangkat audio masukan, dapat dijalankan dalam thread terpisah dari proses GUI. Terdapat juga beberapa fungsi untuk mendukung proses dari kelas-kelas yang ada. Berikut merupakan kelas-kelas yang diimplementasikan dalam komponen GUI, baik sebagai halaman, ataupun sebagai kelas yang menangani *threading*.

- 1) *Threading*: Kelas ini menangani proses streaming sekaligus pendeteksian nada dalam *thread* yang terpisah dari proses GUI. Pada kelas ini, terdapat beberapa fungsi-fungsi untuk menjalankan *thread*, menghentikan *thread*, serta fungsi untuk menunggu hasil deteksi not dari komponen pendeteksi nada.
- 2) Halaman utama: Halaman utama adalah halaman yang pertama kali muncul ke pengguna ketika program pertama dijalankan. Pada halaman ini, terdapat tombol

untuk menuju halaman daftar latihan, menuju halaman pengaturan, dan tombol *about* untuk melihat informasi tentang aplikasi

- 3) Halaman pengaturan sistem: Halaman pengaturan sistem merupakan halaman yang dapat digunakan pengguna untuk mengatur berbagai parameter yang digunakan sistem. Pengguna dapat mengubah nilai masing-masing parameter dengan mengubah nilai yang ada pada kolom teks kemudian menyimpannya. Hasil modifikasi pengaturan yang dilakukan oleh pengguna disimpan ke dalam sebuah *file* dalam bentuk json dan kemudian dibaca oleh program sebagai variabel. Tujuan pembuatan halaman pengaturan sistem adalah untuk memberikan kebebasan lebih kepada pengguna.
- 4) Halaman daftar latihan: Halaman daftar latihan adalah halaman yang bertujuan untuk menampilkan daftar latihan yang sudah pernah dibuat oleh pengguna. Pada halaman ini, pengguna dapat melihat latihan apa saja yang tersedia. Pengguna juga dapat memilih salah satu latihan dari daftar latihan dan dapat memilih untuk memulai, memodifikasi, atau menghapus latihan tersebut. Daftar latihan yang tersedia tersimpan dalam sebuah *file* json dan dibaca setiap kali halaman daftar latihan dibuka.
- 5) Halaman pengaturan latihan: Halaman pengaturan latihan adalah halaman yang tampil ketika pengguna ingin membuat latihan baru atau ingin memodifikasi latihan yang sudah ada pada halaman daftar latihan. Pada halaman ini, terdapat beberapa kolom yang masing-masing merupakan atribut dari latihan. Setelah mengisi semua kolom, pengguna dapat menyimpan pengaturan latihan dengan menekan tombol *Save*.
- 6) Halaman latihan: Halaman latihan adalah halaman yang tampil ketika pengguna melakukan sebuah latihan. Pada halaman ini, pengguna dapat melihat nama latihan yang sedang dilakukan, deskripsi dari latihan tersebut, not target yang harus dimainkan, serta not yang sedang dimainkan oleh pengguna. Tampilan halaman ini berganti secara *real-time* sesuai dengan not masukan dari pengguna. Dari Gambar III.1, dapat dilihat bahwa not yang muncul pada bagian *Input Note* berwarna merah. Hal ini karena not masukan pengguna tidak sesuai dengan not target. Dalam hal ini, *C#3* tidak sesuai dengan *C4*, sehingga not berwarna merah dan not target belum berganti. Ketika not target sama dengan not masukan, tampilan not yang muncul pada bagian *Input Note* akan berwarna hijau dan not pada bagian *Target Note* akan berubah menjadi not target selanjutnya. Berdasarkan latihan yang dilakukan, tampilan dan perilaku halaman latihan bisa berbeda satu latihan dengan yang lainnya. Halaman menampilkan not alternatif pada bagian *Target* sebagai ganti dari nama not asli pada bagian *Target Note* jika latihan memiliki atribut not alternatif. Selain itu, jika latihan memiliki atribut pengambilan not secara acak, halaman menampilkan not dengan urutan yang acak dari kumpulan not target latihan dan bukan sesuai dengan urutan.



Gambar III.1 Halaman Latihan

IV. PENGUJIAN DAN HASIL

Sebagai evaluasi dari sistem pelatih keterampilan alat musik yang diimplementasikan, dilakukan pengujian terhadap sistem tersebut. Pengujian dilakukan terhadap berbagai aspek yang masing-masing memiliki metrik pengukurannya masing-masing.

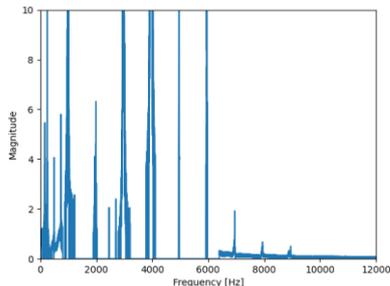
A. Pengujian Pitch Detection Algorithm

Pengujian *pitch detection algorithm* bertujuan untuk mengukur akurasi sistem dalam mendeteksi not masukan. Pengujian ini dilakukan terhadap tiga alat musik, yakni gitar akustik, gitar listrik dan piano digital. Untuk melakukan pengujian ini, ditentukan sebuah jangkauan not yang akan diperiksa akurasinya, kemudian not-not tersebut dimainkan pada setiap alat musik. Dalam pengujian ini, not-not yang diuji akurasinya merupakan semua not dari E2 hingga B5, yakni 44 not. Penentuan jangkauan not tersebut ditentukan berdasarkan jangkauan not terpendek dari ketiga alat musik yang dimainkan, yakni gitar akustik. Sebelum pengujian, dilakukan terlebih dahulu pengujian awal untuk mendapatkan nilai dari perkalian HPS yang menghasilkan akurasi maksimal untuk ketiga alat musik untuk *range* not yang ditentukan dengan perangkat audio pengujian. Dari pengujian awal, didapatkan nilai perkalian HPS yang tepat adalah 6 kali. Nilai yang besar (e.g. lebih dari 6) mengakibatkan sistem kesulitan mendeteksi not-not tinggi seperti B5 karena kurangnya puncak harmonik pada proses HPS, sedangkan nilai yang kecil (e.g. kurang dari 6) mengakibatkan sistem kesulitan mendeteksi not-not rendah.

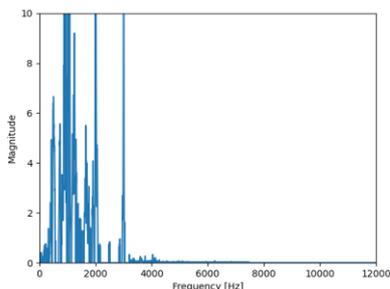
Pengujian dilakukan terhadap 132 kasus uji (44 not dan 3 alat musik). Kemudian, untuk setiap not, ditinjau hasil deteksi oleh sistem dan kesesuaiannya dengan not masukan. Dari hasil pengujian, didapatkan akurasi deteksi not untuk alat musik gitar akustik, gitar elektrik, dan piano masing-masing didapatkan sebesar 100%, 100%, dan 97,7%. Akurasi total yang didapatkan adalah sebesar 99,2% dengan kombinasi alat musik beserta not yang tepat terdeteksi sebanyak 131 dari 132 kasus uji.

Dari hasil pengujian, didapatkan kesalahan deteksi pada not uji paling tinggi pada piano (B5). Hal yang dapat memengaruhi

hasil tersebut adalah frekuensi nada yang terlalu tinggi untuk sistem sehingga frekuensi harmonik yang terlalu tinggi tidak berhasil ditangkap oleh perangkat audio sehingga menyebabkan amplitudo frekuensi fundamental pada hasil HPS dikalikan dengan nol. Hal tersebut juga bisa disebabkan oleh tidak lengkapnya frekuensi harmonik yang dihasilkan oleh alat musik seperti pada Gambar IV.1 dan Gambar IV.2. Pada kedua gambar tersebut, dapat dilihat bahwa puncak frekuensi harmonik not B5 pada gitar listrik lebih lengkap daripada piano digital.



Gambar IV.1 Representasi Spectral Not B5 pada Gitar Elektrik



Gambar IV.2 Representasi Spectral Not B5 pada Piano Digital

B. Pengujian Kemampuan Melatih Keterampilan Alat Musik

Pengujian kemampuan melatih keterampilan alat musik bertujuan untuk mengukur akurasi sistem dalam mendeteksi not masukan. Sama seperti pengujian *pitch detection algorithm*, pengujian ini dilakukan terhadap tiga alat musik, yakni gitar listrik, gitar akustik, dan piano digital. Keterampilan yang diuji merupakan keterampilan yang didefinisikan pada bagian metodologi. Untuk melakukan pengujian ini, diperlukan sebuah kasus uji yang dapat memenuhi metrik masing-masing keterampilan. Karena cara kerja sistem, semua metrik pengujian diterjemahkan menjadi apakah dapat dibuat latihan untuk sebuah keterampilan atau tidak. Sistem dikatakan dapat melatih keterampilan tersebut apabila latihan untuk suatu keterampilan dapat dibuat.

Hasil pengujian dapat dilihat pada Tabel IV.1 terdapat 4 keterampilan uji yang dapat dilatih dan 3 keterampilan uji yang tidak dapat dilatih. Keterampilan uji yang dapat dilatih berarti ada latihan yang dapat dibuat untuk keterampilan tersebut, sedangkan tidak ada latihan yang dapat dibuat untuk 3 keterampilan uji yang tidak dapat dilatih. Selama pengujian, didapatkan bahwa sistem mengalami sedikit kesulitan dalam mendeteksi dua not target yang sama secara berturut-turut. Hal ini karena jika bunyi not belum berhenti sepenuhnya, sistem

kesulitan mendeteksi kapan suatu not selesai berbunyi dan kapan not baru yang sama mulai dimainkan lagi, sehingga menganggap dua not sama yang dimainkan secara cepat menjadi satu not.

Tabel IV.1 Hasil Pengujian Kemampuan Melatih Keterampilan Alat Musik

Nama keterampilan	Hasil pengujian
Memainkan deretan not	Dapat dilatih
Memainkan akor	Tidak dapat dilatih
Menghafal not	Dapat dilatih
Melatih irama	Tidak dapat dilatih
Memainkan tangga nada dalam sebuah kunci	Dapat dilatih
Melatih dinamika	Tidak dapat dilatih

Dari pengujian ini, didapat kesimpulan bahwa keterampilan yang dapat dilatih oleh sistem pelatih keterampilan alat musik adalah memainkan deretan not, menghafal not, memainkan tangga nada dalam sebuah kunci, dan melatih interval dan teori musik lainnya terkait relasi not.

C. Pengujian Pengguna

Pengujian pengguna bertujuan untuk mengukur tingkat *usability* dari sistem. Pengujian ini dilakukan kepada 5 orang dengan perangkat dan alat musik masing-masing. Diambil 5 orang mahasiswa untuk pengujian dengan alasan kemudahan dalam proses pengujian, serta sudah cukup menjadi representasi pengguna sistem, yang dalam hal ini, bisa berupa siapa saja. Sebelum pengujian, ditanyakan terlebih dahulu keterampilan bermain alat musik pengguna serta alat musik yang digunakan untuk pengujian sebagai salah satu faktor hasil pengujian.

Untuk melakukan pengujian pengguna, dibuat 2 buah kuesioner yang berupa *Single-Ease Question* (SEQ) dan *System Usability Scale* (SUS). Tujuan dibuatnya 2 kuesioner ini adalah untuk mengukur tingkat kesulitan tugas yang diberikan serta mengukur *usability* sistem secara keseluruhan. Hasil dari kuesioner yang diberikan adalah nilai SEQ per tugas serta nilai SUS untuk sistem.

Dari hasil kuesioner, didapatkan bahwa dari skala 1 sampai 10, satu orang menjawab bahwa keterampilan bermain alat musik mereka berada di angka 5, satu orang berada di angka 4, satu orang di angka 7, dan satu orang di angka 8. Hal ini menunjukkan bahwa Sebagian besar responden setidaknya memiliki sedikit keterampilan dalam bermain alat musik.

Selama pengujian, diberikan 5 tugas kepada pengguna. Tugas-tugas dibuat dengan maksud mencakup keseluruhan fungsi dari sistem. Adapun tugas-tugas yang diterjemahkan dalam Bahasa Indonesia antara lain adalah sebagai berikut.

- 1) Ubah pengaturan "*Power Threshold*" menjadi $1e-5$
- 2) Buat sebuah latihan dengan 3 not target berbeda, tanpa nama alternatif, dan tanpa pengambilan not acak
- 3) Mulai latihan yang telah dibuat dan coba mainkan not-not target

- 4) Ubah latihan yang telah dibuat sehingga pengambilan not secara acak
- 5) Hapus latihan yang telah dibuat

Pengguna diminta untuk melakukan tugas-tugas tersebut dan menilai tingkat kemudahannya dari 1 sampai 7, yang dalam hal ini 1 berarti sangat sulit dan 7 berarti sangat mudah. Dari kuesioner tersebut, didapatkan hasil rata-rata SEQ seperti Gambar IV.3.



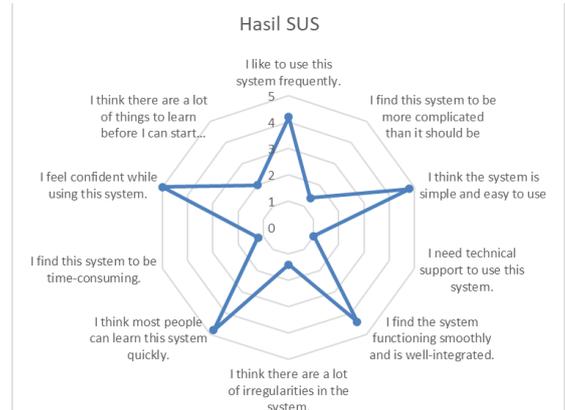
Gambar IV.3 Hasil SEQ

Didapatkan beberapa responden mengalami kesulitan dalam membuat latihan baru serta memainkannya. Hal yang dapat menyebabkan kesulitan adalah pengguna membutuhkan pengetahuan lebih terkait cara penulisan not sehingga dapat mengalami kesulitan dalam pengerjaan tugas 2.

Setelah pengguna melakukan seluruh tugas yang diberikan, pengguna mengisi kuesioner SUS untuk menilai *usability* sistem secara keseluruhan. Pertanyaan-pertanyaan yang ditanyakan kepada pengguna merupakan pertanyaan SUS yang standar digunakan. Adapun pertanyaan-pertanyaan yang diterjemahkan ke Bahasa Indonesia adalah sebagai berikut dengan masing-masing pertanyaan memiliki rentang nilai 1 sampai 5, dengan 1 berarti sangat tidak setuju dan 5 berarti sangat setuju.

- 1) Saya ingin sering menggunakan sistem ini
- 2) Saya merasa sistem ini lebih rumit dari seharusnya
- 3) Saya merasa sistem ini simpel dan mudah digunakan
- 4) Saya membutuhkan bantuan dari teknisi dalam menggunakan sistem ini
- 5) Saya merasa fitur-fitur sistem ini berjalan dengan mulus
- 6) Saya merasa ada banyak kejanggalan dalam sistem ini
- 7) Saya merasa kebanyakan orang akan memahami cara menggunakan sistem ini dengan cepat
- 8) Saya merasa sistem ini memakan waktu
- 9) Saya merasa percaya diri dalam menggunakan sistem ini
- 10) Saya perlu mempelajari berbagai hal terlebih dahulu sebelum menggunakan sistem ini

Didapatkan hasil rata-rata SUS seperti Gambar IV.4. pada pertanyaan “Saya perlu mempelajari berbagai hal terlebih dahulu sebelum menggunakan sistem ini”, rata-rata nilai yang didapat adalah 2 dengan seorang responden menjawab 4. Hal ini berarti sistem pelatih alat musik yang dikembangkan kurang ramah pengguna karena adanya banyak hal yang harus dipelajari terlebih dahulu.



Gambar IV.4 Hasil SUS

V. KESIMPULAN DAN SARAN

Sistem pelatih keterampilan alat musik berhasil dikembangkan dengan memanfaatkan *pitch detection algorithm*, yang dalam hal ini merupakan *Harmonic Product Spectrum* (HPS). Sistem ini merupakan aplikasi berbasis desktop dengan GUI sebagai antarmuka pengguna. Metode *pitch detection algorithm* (PDA) yang dapat digunakan untuk mendeteksi nada dari sebuah alat musik yang dimainkan secara *real time* adalah *Harmonic Product Spectrum* karena merupakan salah satu PDA yang paling sederhana serta membutuhkan biaya komputasi yang murah serta memiliki ketahanan terhadap *noise*. Cara memanfaatkan PDA untuk membangun sistem pelatih keterampilan alat musik adalah dengan membuat sebuah aplikasi berbasis *desktop* yang menampilkan not target yang harus dimainkan, menerima masukan berupa not yang dimainkan pada alat musik, dan memberikan umpan balik terkait kesesuaian not masukan dengan not target. Keterampilan yang dapat dilatih dengan sistem pelatih berbasis *pitch detection algorithm* antara lain adalah memainkan deretan not, menghafal not, memainkan tangga nada dalam sebuah kunci, dan melatih interval dan teori musik lainnya terkait relasi not. Dari hasil pengujian, Didapatkan akurasi deteksi not dengan menggunakan *Harmonic Product Spectrum* sebesar 99,2% yakni 131 kasus deteksi tepat dari 132 total kasus dengan satu-satunya not yang tidak dapat dideteksi dengan tepat adalah not B5 yang dimainkan pada piano. Hal ini dapat disebabkan oleh beberapa faktor, di antaranya adalah tidak lengkapnya frekuensi harmonik dari not tersebut karena frekuensi terlalu tinggi, yang menyebabkan tidak terdeteksinya frekuensi fundamental dari not tersebut.

Dari hasil implementasi dan pengujian, didapatkan bahwa masih terdapat beberapa hal yang harus dikembangkan untuk penelitian selanjutnya. Untuk penelitian selanjutnya, dapat

dikembangkan fungsionalitas sistem untuk dapat mendeteksi dua not sama yang dimainkan secara berturut-turut tanpa jeda sehingga sistem dapat mengetahui kapan sebuah not selesai dimainkan dan kapan not baru mulai dimainkan. Selain itu, *user experience* juga bisa lebih diperhalus dengan cara seperti membuat latihan yang dihentikan tidak menyebabkan aplikasi untuk *restart*. Selain itu, dapat dikembangkan fungsionalitas lainnya untuk dapat melatih keterampilan-keterampilan lainnya, seperti membuat sistem yang dapat melatih tempo, serta akurasi dari PDA dapat ditingkatkan lebih lanjut.

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, penulis telah berhasil menyelesaikan proses penyusunan *paper* dengan tepat waktu. Proses pengerjaan ini tidak akan berjalan lancar tanpa adanya bantuan dan dukungan dari pihak-pihak yang berjasa. Oleh karena itu, penulis ingin berterima kasih kepada pihak-pihak tersebut, antara lain Bapak Dr. Ir. Rinaldi Munir, M. T. selaku dosen pembimbing tugas akhir, Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. dan Bapak Dicky Prima Satya, S.T, M.T. selaku penguji tugas akhir, kedua orang tua dan keluarga yang selalu mendukung dan mendoakan penulis, seluruh dosen Institut Teknologi Bandung, terutama dosen pada program studi

Teknik Informatika yang telah mengajar dan membimbing penulis selama masa perkuliahan, serta seluruh teman dan yang telah membantu penulis dalam pengerjaan tugas akhir, terutama pada bagian pengujian.

REFERENSI

- [1] "Yousician," *Yousician*, 2018. <https://yousician.com/>
- [2] D. Gerhard, "Pitch Extraction and Fundamental Frequency: History and Current Techniques," Jan. 2003.
- [3] R. G. Amado and J. V. Filho, "Pitch detection algorithms based on zero-cross rate and autocorrelation function for musical notes," Jul. 2008, doi: <https://doi.org/10.1109/icalip.2008.4590188>.
- [4] Patricio, A. S. Master, and C. Sapp, "Efficient Pitch Detection Techniques for Interactive Music," *International Computer Music Conference*, vol. 2001, Jan. 2001.
- [5] N. Zurstraßen, "Programming a Guitar Tuner with Python," May 13, 2020. <https://www.chciken.com/digital/signal/processing/2020/05/13/guitar-tuner.html> (accessed Jul. 01, 2024)
- [6] L. Sukhostat and Y. Imamverdiyev, "A Comparative Analysis of Pitch Detection Methods Under the Influence of Different Noise Conditions," *Journal of Voice*, vol. 29, no. 4, pp. 410–417, Jul. 2015, doi: <https://doi.org/10.1016/j.jvoice.2014.09.016>.