

Penerapan Logits Processing Pada Teknologi Transformer untuk Penciptaan Melodi Berbentuk Notasi ABC dalam Pengembangan Game Indie

Muhammad Faishal Ali Dhiaulhaq*, Arif Akbarul Huda, Arifiyanto Hadinegoro

¹Fakultas Ilmu Komputer, Informatika, Universitas Amikom Yogyakarta, Sleman, Indonesia

Email: ¹faishalali22@students.amikom.ac.id, ²arif.akbarul@amikom.ac.id, ³arifiyanto@amikom.ac.id

Email Penulis Korespondensi: ¹faishalali22@students.amikom.ac.id

Submitted: 07/01/2025; Accepted: 26/01/2025; Published: 01/03/2025

Abstrak—Teknologi kecerdasan buatan generatif (Gen AI) saat ini banyak dimanfaatkan oleh pelaku kreatif, termasuk pemusik dan pengembang game. Banyak pengembang game sekarang yang mengandalkan aset musik dari sumber terbuka atau berbayar, namun variasi musik yang tersedia cenderung terbatas. Penelitian ini bertujuan untuk membantu pengembang game dalam menyediakan aset musik berupa notasi ABC. Metode penelitian mencakup pengumpulan data notasi ABC, pemrosesan data, pengembangan model, dan evaluasi metrik. Data dikumpulkan dengan mengambil notasi ABC beserta elemen musik yang khas. Pemrosesan data meliputi penanganan missing value dan seleksi fitur, sementara persiapan data melibatkan pelabelan dan tokenisasi. Model yang digunakan adalah GPT-2 berbasis arsitektur Transformer, yang telah dilatih dengan dataset umum. Integrasi model dengan data notasi ABC dilakukan dengan pembaruan Logits Processing untuk meningkatkan kontrol hasil keluaran. Hasil evaluasi menunjukkan bahwa teknologi transformer dapat menghasilkan pola pitch yang sesuai dengan data validasi, dengan distribusi nilai EMD terkonsentrasi di rentang 1.0–1.5 dan rata-rata 1.60. Meskipun ada beberapa outlier dan perbedaan distribusi pitch antara data validasi dan hasil generasi, genre Horror dengan mood Joyful dan emosi Excitement menghasilkan nilai combined fitness tertinggi 0.528. Model masih perlu penyempurnaan untuk menghasilkan distribusi pitch yang lebih konsisten. Penelitian ini menunjukkan potensi penggunaan teknologi transformer dalam pembuatan aset musik untuk game, namun penelitian lanjutan diperlukan untuk meningkatkan akurasi dan konsistensi hasil.

Kata Kunci: Deep Learning; Transformers; Notasi ABC; Logits Processing; Generative AI

Abstract—Generative Artificial Intelligence (Gen AI) technology is increasingly being used by creative professionals, including musicians and game developers. Many game developers now turn to open or paid music assets, but the variety of options is usually quite limited. This research aims to assist game developers in generating music assets in ABC notation format. The research methods include data collection in the form of ABC notation, data processing, model development, and metric evaluation. The data was collected by extracting ABC notation along with the characteristic musical components of each item. Data processing involved handling missing values and feature selection, while data preparation included labeling and tokenization. The model used was GPT-2 based on the Transformer architecture, pretrained on a general dataset. Integration of the model with ABC notation data was enhanced using Logits Processing to improve output control. The evaluation results show that Transformer technology can generate pitch patterns consistent with the validation data, with the EMD values concentrated in the range of 1.0–1.5 and an average of 1.60. Although there are some outliers and differences in pitch distribution between the validation data and generated results, the Horror genre with a Joyful mood and Excitement emotion achieved the highest combined fitness score of 0.528. The model still requires further refinement to produce more consistent pitch distributions. This research demonstrates the potential of Transformer technology in generating music assets for games, but further studies are needed to improve accuracy and consistency in the results.

Keywords: Deep Learning; Transformers; ABC Notation; Logits Processing; Generative AI

1. PENDAHULUAN

Musik merupakan salah satu cabang seni yang sangat digemari dan memiliki tempat yang istimewa dalam kehidupan manusia. Sejak zaman dahulu, kehidupan manusia selalu diiringi oleh musik yang tak hanya hadir sebagai hiburan, tetapi juga sebagai bagian penting dari budaya, upacara, dan ekspresi emosi. Seiring berjalannya waktu, musik terus berkembang dan mengalami perubahan, dengan berbagai variasi dan genre yang semakin beragam. Dari musik tradisional yang diwariskan turun temurun hingga musik modern yang dipengaruhi oleh kemajuan teknologi, musik telah menjadi elemen yang tak terpisahkan dalam kehidupan manusia [1]. Hal ini tercermin dalam bagaimana musik menjadi bagian dari berbagai aspek kehidupan, mulai dari media hiburan, ritual budaya, hingga sebagai bentuk terapi emosional [2].

Selain itu, musik juga menjadi sumber inspirasi yang tak ternilai bagi para pelaku kreatif, baik itu pemusik, penulis, seniman, maupun pengembang aplikasi. Terutama bagi para pengembang *game*, musik menjadi elemen yang sangat penting dalam menciptakan pengalaman interaktif yang mendalam. Dalam konteks pengembangan *game*, musik tidak hanya berfungsi sebagai latar belakang, tetapi juga sebagai pembangun atmosfer dan pemicu emosi bagi pemain [3]. Oleh karena itu, keberadaan musik dalam dunia *game* sangat krusial dalam membentuk nuansa permainan dan memperkaya pengalaman yang dirasakan oleh pemain. Namun, tidak semua pelaku kreatif, khususnya pengembang *game*, dapat menciptakan musik mereka sendiri. Hal ini terutama berlaku bagi pengembang *game Indie*, yang sering kali harus mengelola hampir seluruh aspek pengembangan *game* secara mandiri. Mulai dari *coding*, desain visual, hingga pembuatan aset-aset pendukung lainnya. Dengan keterbatasan sumber daya dan keahlian, tidak semua pengembang *game Indie* memiliki keahlian dalam membuat musik yang sesuai dengan visi permainan mereka. Oleh

karena itu, mereka lebih memilih untuk mencari musik dari pihak ketiga yang menyediakan musik yang bisa disesuaikan dengan kebutuhan *game* yang sedang mereka kembangkan. Dalam banyak kasus, pengembang *game Indie* akan mengandalkan penyedia musik khusus yang menawarkan berbagai jenis musik yang bisa diintegrasikan dalam *game* mereka tanpa harus membuatnya sendiri [4].

Meskipun, keterbatasan variasi musik yang tersedia sering kali menjadi tantangan bagi para pengembang *game*. Terkadang musik yang tersedia tidak dapat memenuhi kesesuaian kebutuhan pengembang untuk *game* yang sedang dikembangkan. Masalah keterbatasan musik menjadi alasan utama penelitian ini, yaitu dengan memberikan solusi untuk mengatasi keterbatasan musik yang tepat untuk pengembangan *game*. Penelitian ini bertujuan untuk mengembangkan solusi berbasis kecerdasan buatan (AI) dengan menggunakan arsitektur Transformer untuk generasi musik yang dapat diterapkan dalam pengembangan *game*. Kontribusi utama dari penelitian ini adalah memperkenalkan metode baru yang memungkinkan pengembang *game*, khususnya pengembang *game Indie*, untuk menciptakan variasi musik yang lebih sesuai dengan tema dan suasana permainan tanpa dibatasi oleh keterbatasan variasi aset musik yang ada. Dengan pendekatan ini, penelitian ini tidak hanya membuka kemungkinan baru dalam pembuatan musik untuk *game*, tetapi juga dapat meningkatkan pengalaman pemain dengan musik yang lebih dinamis dan kontekstual.

Salah satu solusi yang dapat diberikan yaitu dengan memanfaatkan teknologi kecerdasan buatan, khususnya arsitektur Transformer. Teknologi ini telah terbukti ampuh untuk menangani proses pengenalan data sekuensial dalam memaksimalkan nilai posibilitas dari data. Cara kerja dari arsitektur Transformer memanfaatkan metode *autoregressive* yang memprediksi *token* selanjutnya dari *token* data sekuensial yang diberikan sebelumnya sebagai konteks utamanya [5][6]. Metode ini membantu dalam memprediksi data teks kontekstual sebagai hasil keluaran generatif. Hal ini menjadi relevan saat generasi musik berbentuk teks melodi dalam format notasi ABC, di mana setiap rangkaian notasi diprediksi berdasarkan konteks notasi-notasi sebelumnya. Transformer juga dapat mengatasi tantangan dalam mempertahankan koherensi dan keindahan musik, sehingga melodi yang dihasilkan masih terdengar alami dan sesuai dengan rasa musikal yang diinginkan [7].

Seperti halnya pada penelitian oleh R. Soelistijadi et al. menggunakan LSTM dalam meregenerasi teks lirik musik dari Alm. Didi Kempot. Dalam percobaan generasi teks 6 klaster sebanyak 3 kali maka panjang kata dalam setiap baris yang dianjurkan sebanyak 3-4 kata agar memperoleh lirik lagu dengan kesesuaian arti [2].

Selanjutnya, pada penelitian yang dilakukan oleh Q. Huang et al, menggunakan model Noise2Music yang mengadopsi kerangka kerja dua tahap. Tahap pertama melibatkan model generator yang bertanggung jawab untuk menghasilkan audio atau spektrogram dengan kualitas rendah (low-fidelity), sementara tahap kedua melibatkan model cascader yang digunakan untuk meningkatkan kualitas audio tersebut menjadi lebih tinggi (high-fidelity). Pada hasil evaluasi, model ini mencapai skor *Fréchet Audio Distance* (FAD) sebesar 0,518 untuk *model waveform* dan 0,459 untuk model spektrogram. Angka ini mengindikasikan kualitas audio yang lebih baik dibandingkan model *baseline* seperti Riffusion dan Mubert [8].

Selanjutnya, pada penelitian yang dilakukan oleh Igor Cardoso et al. menggunakan model *Controllable Music Transformer* (CMT) dalam melibatkan 140 klip video memperoleh hasil pola ritme yang lebih dekat dengan komposisi manusia, dengan nilai *Grooving Pattern Similarity* sebesar 0.821. Untuk variasi kelas nada dalam musik mendapatkan nilai rata-rata *Number of Unique Pitch Classes* sebesar 9.840. Kedua metrik memperoleh hasil yang mirip dengan buatan manusia [4].

Selanjutnya, penelitian oleh Yi-Jen S. et al, telah mengembangkan Theme Transformer yang menggunakan pendekatan *contrastive learning* dan *clustering* dalam proses ekstraksi tema musik dari potongan-potongan musik. Pendekatan ini dapat menghasilkan musik yang lebih konsisten dengan tema yang diberikan daripada metode *prompt-based*, terutama pada aspek *polyphonic* pada musik pop piano. Hasil tercerminkan dari pola-pola repetisi dan variasi tema yang menyerupai karya manusia [9].

Selain itu, penelitian Xin L. et al, berupaya meningkatkan ketahanan model CNN terhadap serangan adversarial dengan *Probabilistically Compact* (PC) *loss* dan *logit constraints*. Pendekatan ini telah meningkatkan ketahanan model terhadap gangguan kecil pada data input, tanpa mengurangi kinerja model pada data yang bersih [10].

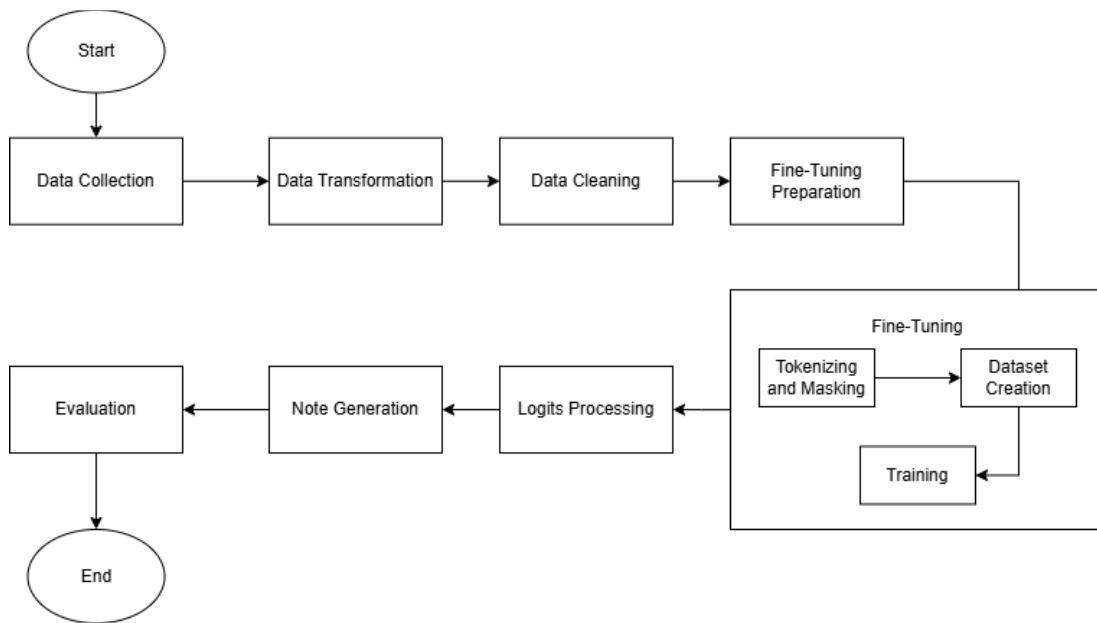
Meskipun penelitian sebelumnya telah mengeksplorasi generasi musik menggunakan AI, sebagian besar masih berfokus pada optimasi kualitas audio tanpa memperhatikan aplikasi spesifik dalam pengembangan *game*, terutama untuk pengembang *game Indie* dengan sumber daya terbatas. Namun, penelitian-penelitian tersebut belum membahas bagaimana AI dapat disesuaikan untuk menghasilkan musik yang sesuai dengan konteks dan atmosfer *game*, serta memberikan variasi yang diperlukan untuk genre *game* tertentu. Penelitian ini mengisi kekosongan tersebut dengan mengembangkan solusi berbasis arsitektur Transformer yang tidak hanya meningkatkan variasi dan kualitas musik, tetapi juga kemampuan adaptasi dinamis dengan genre *game* yang dikembangkan, memberikan opsi musik yang lebih kaya dan relevan bagi pengembang *game Indie*.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dirancang melalui rangkaian langkah yang terorganisir, sistematis, dan terencana dengan baik untuk memastikan hasil yang diperoleh memiliki tingkat keakuratan dan keandalan yang sesuai. Pada Gambar 1 terlihat

tahapan penelitian, dimulai dengan pengumpulan data, transformasi data, hingga pembersihan data. Proses tersebut diikuti oleh persiapan *fine-tuning*, yang melibatkan tokenisasi, *masking*, pembuatan dataset, dan pelatihan model untuk menghasilkan performa optimal. Hasil logit kemudian diproses untuk mendukung tahap generasi musik, diakhiri dengan evaluasi komprehensif guna mengkaji keluaran yang sesuai dengan tujuan penelitian.



Gambar 1. Tahapan Penelitian

2.2 Pengumpulan Data

Data diambil dari platform Kaggle dengan nama dataset “ABC Notation Music for RNN”. Dataset berisi 46 musik berbentuk notasi ABC dengan detail seperti judul musik, ritme, birama, panjang notasi, kunci dasar, dan sumber atau pencipta. Dataset berbentuk teks (.txt) yang masih mentah. Lalu data yang penulis gunakan dapat diakses di tautan berikut <https://www.kaggle.com/datasets/juansebm/abc-notation-music-for-rnn>.

2.3 Transformasi Data

Proses yang dilakukan pada tahap transformasi data yaitu mengubah bentuk dataset menjadi lebih detail untuk memperkaya informasi yang dapat membantu dalam proses *fine-tuning*. Perubahan bentuk dataset dengan menggunakan *library* Pandas serta penggunaan *regular expression* untuk mendeteksi pola teks. Langkah yang dilakukan penulis untuk mentransformasi data yaitu dengan menggunakan pola *pitch* dengan bantuan kunci dasar [11]. Pemberian label pola *pitch* dilakukan pada setiap *bar* yang memisahkan antar sekuensial notasi. Selain itu, hasil akhir juga dilakukan rekonstruksi dataset agar lebih mudah diproses untuk persiapan *fine-tuning*.

Untuk aturan pemberian pola *pitch* atau *pitch pattern* didasarkan pada metode yang disebut metode analisis pola nada relatif. Pada kasus penulis, detail dari penerapan metode tersebut sudah disediakan pada **Tabel 1**.

Tabel 1. Dasar Pembentukan Kategori Pola *Pitch*

Kategori	Kondisi	Deskripsi
Up	Semua nilai nada naik secara bertahap: $pitch_values[i] > pitch_values[i-1]$ untuk semua i .	Pola nada menunjukkan peningkatan tanpa penurunan atau nada tetap.
Down	Semua nilai nada turun secara bertahap: $pitch_values[i] < pitch_values[i-1]$ untuk semua i .	Pola nada menunjukkan penurunan tanpa peningkatan atau nada tetap.
Flat	Salah satu dari: 1. Satu nada: Bar terlalu pendek (hanya satu nilai). 2. Nada sama: Semua nilai nada tetap sama: $pitch_values[i] == pitch_values[i-1]$. 3. Pola campuran: Kombinasi nada naik, turun, atau tetap (tidak konsisten)	1. Pola nada tidak konsisten atau tidak ada perubahan berarti. 2. Tidak cukup panjang untuk membentuk pola. 3. Tidak ada perubahan nada dalam bar. 4. Pola menunjukkan kombinasi yang tidak konsisten antara naik, turun, dan tetap.

Penjelasan pada Tabel 1 merupakan pendekatan untuk mengategorikan pola perubahan *pitch* dalam notasi musik yang direpresentasikan dengan format ABC berdasarkan analisis pola nada relatif yang dilakukan. Setiap *bar* berisi urutan nada yang dihitung menjadi nilai *pitch*, lalu dikategorikan ke dalam tiga kelompok: *Up*, *Down*, dan *Flat*.

Kategori *Up* diberikan jika nilai *pitch* menunjukkan kenaikan bertahap, *Down* untuk penurunan bertahap, dan *Flat* jika tidak ada perubahan signifikan atau terdapat pola campuran antara kenaikan dan penurunan.

Pendekatan tersebut memungkinkan analisis lebih lanjut terhadap pola-pola musik dalam dataset, yang berguna untuk aplikasi seperti klasifikasi, prediksi, dan pemahaman struktur musik dalam konteks penelitian. Di atas, maka ada contoh kasus yang disediakan oleh penulis yang ada di Tabel 2.

Tabel 2. Contoh Kasus Pemberian Pola *Pitch*

Bar	Nilai <i>Pitch</i>	Kategori	Penjelasan
[C D E F]	[2, 3, 4, 5]	Up	Semua nilai nada naik secara bertahap.
[G F E D]	[6, 5, 4, 3]	Down	Semua nilai nada turun secara bertahap.
[C C C C]	[2, 2, 2, 2]	Flat	Semua nilai nada tetap sama.
[C D C D]	[2, 3, 2, 3]	Flat	Pola campuran antara naik dan turun.
[C]	[2]	Flat	Bar terlalu pendek (hanya memiliki satu nilai nada).
[C D E D C]	[2, 3, 4, 3, 2]	Flat	Pola campuran: nada naik (2 → 3 → 4) dan turun (4 → 3 → 2).

Contoh pada Tabel 2 mengategorikan pola perubahan *pitch* dalam notasi musik menjadi tiga kategori: *Up*, *Down*, dan *Flat*. Kategori *Up* diberikan jika nilai *pitch* meningkat secara bertahap, seperti pada urutan [C D E F]. *Down* digunakan jika nilai *pitch* menurun, seperti pada [G F E D], sementara *Flat* diterapkan pada pola yang tidak menunjukkan perubahan signifikan atau mencakup kombinasi naik-turun, seperti pada [C D C D]. Pendekatan ini memungkinkan pengelompokan pola *pitch* dalam dataset musik untuk analisis lebih lanjut.

2.4 Pembersihan Data

Pembersihan data dilakukan pengisian nilai *null*, dengan tujuan untuk membersihkan atau mengisi data yang hilang atau tidak sesuai. Pengisian nilai *null* pada fitur panjang notasi dengan cara menghitung rata-rata banyak notasi. Selain fitur itu, diisi dengan nilai *default*. Setelah pengisian nilai *null*, nilai yang masih memiliki makna ganda akan diselaraskan untuk mempermudah proses *encoding* nantinya.

Setelah proses pengisian nilai *null*, selanjutnya yaitu menyelaraskan inputan yang tidak seragam. Penyelarasan data bertujuan untuk memperbanyak varian data pada kategori tertentu.

2.5 Persiapan *Fine-Tuning*

Sebelum proses *fine-tuning*, ada beberapa proses yang dilakukan penulis untuk mengubah data agar dapat diterima oleh model. Hal yang dilakukan pertama kali adalah proses *encoding* data yang bersifat kategorikal atau berbentuk teks untuk diubah menjadi bentuk numerik. Proses *encoding* menggunakan *library* dari bahasa pemrograman Python.

2.6 *Fine-Tuning*

Pada proses *fine-tuning*, penulis menggunakan *pretrained model* dari Transformer yaitu GPT-2. Yang dapat digunakan oleh penulis dalam menyesuaikan dan melakukan *fine-tuning* yaitu GPT-2 Tokenizer dan GPT-2 Model. Lalu, proses yang dilakukan adalah melakukan tokenisasi pada data yang sudah berbentuk numerik.

Selanjutnya, proses yang dilakukan oleh penulis yaitu mengambil *id input* dari hasil tokenisasi data sebelumnya. Lalu, data itu diberikan nilai *attention mask* untuk memastikan bahwa setiap informasi itu penting [12]. Dengan memberikan nilai 1 (satu) pada setiap elemen dalam *attention masks*, maka menunjukkan bahwa tidak ada token yang diabaikan oleh model selama proses *self-attention*.

$$Attention\ Mask[i] = \begin{cases} 1, & \text{jika token ke-}i \text{ relevan} \\ 0, & \text{jika token ke-}i \text{ padding atau tidak relevan} \end{cases} \quad (1)$$

Setelah pemberian *attention mask*, langkah yang penulis lakukan selanjutnya adalah mempersiapkan dataset yang berisi pasangan *id input* dan *attention masks* dalam format yang dapat diterima oleh model. Lalu, dataset akan dimasukkan ke dalam *data loader* untuk proses *batching*. Dataset tersebut akan dipecah dalam beberapa *batch* yang setiap *batch* berisi 1 (satu) contoh, yang memungkinkan untuk menghindari penggunaan *padding* dalam proses pelatihan.

Pada proses pelatihan model berbasis Transformer, penulis memilih algoritma optimasi Adam dengan *learning rate* 5×10^{-5} . Algoritma Adam akan meminimalkan fungsi kerugian yang mengukur kesalahan antara prediksi dan label yang benar. Untuk mengatur *learning rate* secara dinamis, maka penulis menerapkan pendekatan Linear Decay sebagai *learning rate scheduler* [13].

$$lr_t = lr_{start} \times \left(1 - \frac{t}{T}\right) \quad (2)$$

Selain itu, penulis juga menerapkan metode *mixed precision training* menggunakan GradScaler, yang mengurangi penggunaan memori tanpa mengorbankan akurasi. Selama proses pelatihan, gradien dihitung menggunakan pendekatan *backpropagation* dan *model parameter* diperbarui berdasarkan rumus optimasi *gradient descent* [14][15].

$$\theta_t = \theta_{t-1} - \eta \cdot \nabla_{\theta} \ell(\theta) \quad (3)$$

Langkah terakhir yaitu melakukan proses *fine-tuning* pada dataset yang sudah disiapkan dengan model yang sudah dibentuk.

2.7 Logits Processing

Dalam upaya untuk mengontrol keluaran model berbasis token, penulis mengimplementasikan pendekatan pemrosesan *logits* kustom yang membatasi pemilihan token hanya pada token yang valid [16]. Misalkan *scores* adalah nilai *logits* yang dihitung untuk setiap token dalam distribusi, dan *valid tokens* adalah indeks token yang diperbolehkan, maka untuk setiap token yang tidak valid, nilai *logits*-nya akan diatur menjadi sangat rendah, yaitu $-\infty$, sehingga token tersebut tidak dipilih dalam langkah berikutnya.

$$scores[i] = \begin{cases} -\infty, & \text{jika } i \notin \text{valid tokens} \\ \text{original score}[i], & \text{jika } i \in \text{valid tokens} \end{cases} \quad (4)$$

Dengan demikian, maka model akan hanya menghasilkan token yang sesuai dengan aturan yang telah ditetapkan, serta meningkatkan relevansi keluaran model terhadap konteks yang diinginkan.

2.8 Generasi Notasi

Proses generasi notasi dilakukan melalui langkah-langkah seperti pemfilteran data berdasarkan masukan pola *pitch*, ritme, dan kunci yang ingin digenerasi. Setelah itu, *bar* yang relevan akan digabungkan untuk membentuk urutan yang lebih panjang. *Bar* yang tergabung akan diberikan *padding* jika diperlukan.

Dengan menggunakan *logits processor*, model dikontrol hanya untuk menghasilkan token yang valid, sesuai dengan aturan yang telah ditetapkan. Maka dengan begitu, integritas dari notasi yang dihasilkan dapat terjaga. Contoh generasi notasi ABC yang dihasilkan oleh model GPT-2 seperti yang ada di **Tabel 7**.

2.9 Evaluasi

Evaluasi kualitas musik yang dihasilkan dilakukan menggunakan berbagai metrik yang mencakup *Average Smoothness*, *Average Consonance*, *Average Rhythmic Variety*, *Average Pitch Range*, *Average Repetition Rate*, dan *Average Tonality* [17]. Metrik-metrik tersebut dikemas menjadi satu formula *Average Combined Fitness*. Pada formula, variabel p adalah setiap *pitch* yang dihasilkan dari proses generasi notasi ABC.

$$Smoothness = \frac{1}{1 + \sum_{i=1}^{n-1} |p_i - p_{i+1}|} \quad (5)$$

$$Consonance = \frac{\sum_{i=1}^{n-1} 1_{\{|p_i - p_{i+1}| \in \{0,7,12\}\}}}{n-1} \quad (6)$$

$$Rhythmic Variety = \frac{\text{Jumlah Durasi Unik}}{\text{Jumlah Durasi Total}} \quad (7)$$

$$Pitch Range = \max(p) - \min(p) \quad (8)$$

$$Repetition Rate = \frac{\sum_{i=1}^{n-1} 1_{\{p_i = p_{i+1}\}}}{n} \quad (9)$$

$$Tonality Score = \frac{\sum_{i=1}^n 1_{\{p_i \bmod 12 \in T\}}}{n} \quad (10)$$

Selain itu, metrik *Earth Mover's Distance* (EMD) juga telah digunakan untuk mengevaluasi perbedaan distribusi *pitch* dalam struktur musik, yang memberikan perspektif tambahan terkait kesesuaian pola *pitch* dengan karakteristik *genre* dan *mood* yang ditargetkan [18][19]. Dengan menggabungkan berbagai metrik ini, analisis evaluasi menjadi lebih menyeluruh dalam menilai kualitas dan kecocokan musik terhadap konteks yang diinginkan. Menambahkan perhitungan distribusi *pitch* juga untuk menambah wawasan terkait sebaran *pitch* yang muncul dengan komparasi pada data validasi.

$$EMD(P, Q) = \min_{\gamma \in \mathcal{R}(P, Q)} \int_X c(x, y) d\gamma(x, y) \quad (11)$$

$$EMD(P, Q) = \sum_{i=1}^N \sum_{j=1}^M |P_i - Q_j| \times D(i, j) \quad (12)$$

$$EMD_{\text{per lagu}} = EMD(P_{\text{validasi}}, P_{\text{generated}}) \quad (13)$$

$$Average Pitch Distribution = \frac{1}{N} \sum_{i=1}^N 1_{\{pitch_i\}} \text{ dengan } 1_{\{pitch_i\}} = \frac{\text{Jumlah } Pitch_i}{\text{Total Entri}} \quad (14)$$

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas hasil penelitian mengenai generasi musik berbasis GPT-2 menggunakan dataset notasi ABC. Analisis dilakukan dengan membandingkan hasil generasi model terhadap data validasi berdasarkan kategori *Genre*, *Mood*, dan *Emotional* yang telah diinputkan setiap kali akan melakukan generasi notasi. Hasil evaluasi model generasi notasi dijelaskan secara mendetail, mencakup performa dalam menghasilkan notasi dengan pola *pitch Up*,

Down, dan *Flat*, serta kesesuaian dengan ritme dan kunci yang sudah melekat pada inputan *Genre*, *Mood*, dan *Emotional*.

Selain itu, dilakukan perbandingan hasil generasi model dengan data validasi menggunakan metrik evaluasi seperti *Average Smoothness*, *Average Consonance*, *Average Rhythmic Variety*, *Average Pitch Range*, *Average Repetition Rate*, dan *Average Tonality*. Metrik tambahan seperti *Earth Mover's Distance* (EMD) juga digunakan untuk mengevaluasi distribusi *pitch*, memastikan konsistensi karakteristik pola *pitch* terhadap data validasi. Setiap hasil dievaluasi berdasarkan kesesuaian dengan kategori *input* yang telah ditentukan, sehingga model dapat dinilai baik dari segi akurasi, relevansi musikalitas, maupun integritas hasil generasi.

3.1 Pengumpulan Data

Data yang dikumpulkan dan digunakan pada penelitian ini diambil dari sumber penyedia data terbuka yaitu Kaggle. Dataset yang digunakan berjudul “ABC Notation Music for RNN” yang berisi data notasi. Dataset ini mencakup 46 lagu dengan informasi detail seperti judul lagu (T), birama (M), panjang notasi (L), sumber atau kolektor (N), transkriptor (Z), kunci dasar (K), dan notasi ABC lengkap untuk setiap lagu. Sebagai contoh, salah satu data dalam dataset memiliki detail seperti terlihat pada Tabel 3 berikut:

Tabel 3. Contoh Isi Dataset Asli

Contoh Isi dari Dataset
X: 1
T: The Enchanted Valley
M: 2/4
L: 1/16
B: "O'Neill's 1"
N: "Very slow" "collected by J. O'Neill"
N:
Z: "Transcribed by Norbert Paap, norbertp@bdu.uva.nl"
Z:
K:Gm
G3-A (Bcd=e) f4 (g2dB) ({d}c3-B) G2-E2 F4 (D2=E^F)
G3-A (Bcd=e) f4 d2-f2 (g2a2 b2).g2 {b}(a2g2 f2).d2
(d2{ed}c2) B2B2 (A2G2 {AG}F2).D2 (GABc) (d2{ed}c>A) G2G2 G2z
G B2c2 (dcAB) G2G2 G3G B2d2 (gfdc) d2g2 (g3ga)
(bagf) (gd)d>c (B2AG) F-D.D2 (GABc) d2d2 (bgfd) cA.F2
G2A2 (B2{cB}AG) A3-G F2-D2 (GABc) (d2{ed}c>A) G2G2 G2z2

Dataset berbentuk file teks (.txt) yang kemudian diolah lebih lanjut menggunakan teknik pembersihan dan transformasi data untuk memastikan konsistensi dan kelengkapan informasi. Data ini menjadi basis utama dalam proses pelatihan dan evaluasi model generasi musik berbasis GPT-2.

3.2 Transformasi Data

Proses transformasi data pada penelitian ini berhasil menghasilkan dataset yang lebih terstruktur dan informatif untuk mendukung proses *fine-tuning* model GPT-2. Transformasi dilakukan dengan menambahkan pola *pitch* berdasarkan analisis pola nada relatif, yang mengelompokkan nada menjadi kategori *Up*, *Down*, dan *Flat*. Bentuk dari dataset yang sudah dilakukan transformasi data dapat dilihat pada Tabel 4.

Tabel 4. Contoh Bentuk Dataset Setelah Proses Transformasi Data

Index	Judul Lagu	Nomor Bar	Bar	...	Pola Pitch	Ritme	Kunci
1	The Enchanted Valley	1	G3-A (Bcd=e)	...	Flat	Very Slow	Gm
2	The Enchanted Valley	2	f4 (g2dB)	...	Flat	Very Slow	Gm
3	The Enchanted Valley	3	(({d}c3-B) G2-E2	...	Flat	Very Slow	Gm

Dataset diubah menjadi bentuk tabel agar lebih mudah diproses, menggunakan *library* Pandas dari bahasa pemrograman Python. Dengan format tabel, fitur-fitur yang ada dalam dataset menjadi lebih jelas dan terstruktur, serta memudahkan analisis lebih lanjut.

3.3 Pembersihan Data

Proses pembersihan data pada penelitian ini memastikan dataset memiliki kualitas yang memadai untuk digunakan dalam pelatihan model. Pembersihan dilakukan dengan mengisi nilai kosong (*null*) pada fitur-fitur penting, seperti panjang notasi, yang diisi menggunakan nilai rata-rata pada kategori musik yang sama, serta menyelaraskan inputan yang tidak seragam untuk memperkaya variasi data. Sebagai hasilnya, dataset menjadi lebih konsisten, dengan format yang terstandarisasi pada fitur-fitur seperti birama, ritme, panjang notasi, dan kunci dasar. Langkah ini penting untuk

memastikan bahwa model dapat memahami pola data dengan baik, mengurangi risiko kesalahan akibat data yang tidak lengkap atau tidak sesuai. Untuk contoh data awal dan hasil penyesuaian nilai ada seperti di **Tabel 5**.

Tabel 5. Contoh Bentuk Nilai Setelah Proses Penyesuaian

Nama Kolom	Data Sebelum Penyesuaian	Data Setelah Penyesuaian
Birama	3/4 3/4 3/8 3/4 3/8 3/4	3/4
Panjang Note	1/2.3065693430656933	1/2
Ritme	moderate	Moderate
Kunci	G Major	G

Proses ini dilakukan penyesuaian nilai pada setiap baris data untuk menjaga konsistensi data. Kolom Birama yang awalnya berisi variasi seperti "3/4 3/4 3/8 3/4 3/8 3/4" diselaraskan menjadi "3/4" untuk menyederhanakan format dan memastikan keseragaman birama. Begitu pula dengan kolom Panjang Note, nilai yang awalnya lebih panjang ("1/2.3065693430656933") diselaraskan menjadi "1/2" untuk memudahkan perbandingan. Selain itu, nilai pada kolom Ritme dan Kunci juga disesuaikan agar lebih konsisten dalam format penulisan, seperti penyesuaian "moderate" menjadi "Moderate" dan "G Major" menjadi "G". Proses penyesuaian ini dilakukan untuk mempermudah pemrosesan data lebih lanjut.

3.4 Proses-Proses pada *Fine-Tuning*

Proses persiapan *fine-tuning* dilakukan untuk memastikan dataset dapat diterima dengan baik oleh model GPT-2. Tahap ini mencakup *encoding* data kategorikal menjadi bentuk numerik, seperti mengonversi notasi ABC, panjang notasi, pola *pitch*, birama, ritme, dan kunci dasar ke dalam representasi angka. Hasil dari proses *encoding* adalah seperti pada Tabel 6.

Tabel 6. Contoh Bentuk Data Setelah *Encoding*

Nama Kolom	Sebelum Proses <i>Encoding</i>	Setelah Proses <i>Encoding</i>
Bar	G3-A (Bcd=e).	[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
Panjang Note	1/16	12
Pola Pitch	Flat	1
Birama	2/4	2
Ritme	Very Slow	31
Kunci	Gm	33

Pada Tabel 6 menunjukkan bahwa saat ini nilai-nilai di dalam dataset kini telah dikonversi menjadi format numerik, sehingga siap diproses oleh model.

Setelah data berhasil dienkoding, selanjutnya proses *fine-tuning*. Langkah pertama adalah tokenisasi data numerik, diikuti dengan pemberian *attention mask* untuk memastikan model fokus pada elemen yang relevan selama pelatihan. Dataset dibagi menjadi *batch* untuk efisiensi, serta agar menghindari penggunaan *padding* yang tidak perlu. Algoritma optimasi Adam digunakan untuk memperbaiki model dengan mengurangi kesalahan, sementara *learning rate* disesuaikan secara dinamis dengan metode Linear Decay. Untuk menghemat memori tanpa mengurangi akurasi, digunakan metode *mixed precision training*. Lalu, *Gradien* dihitung dengan *backpropagation* untuk memperbarui parameter model dengan cara yang lebih efektif, sehingga model dapat belajar pola data dengan lebih baik. Pada tahap *logits processing*, *token* yang tidak relevan diberikan nilai sangat rendah ($-\infty$), sehingga model hanya memilih *token* yang valid dan sesuai dengan aturan yang diinginkan. Ini memungkinkan kontrol lebih baik terhadap keluaran model, menjamin notasi musik yang dihasilkan tetap sesuai dengan konteks dan memenuhi kebutuhan penelitian.

3.5 Generasi Notasi

Proses generasi notasi musik dilakukan dengan memanfaatkan model GPT-2 yang telah melalui tahap *fine-tuning*. Model ini menghasilkan notasi musik berbentuk notasi ABC berdasarkan input pola *pitch*, ritme, dan kunci yang telah ditentukan sebelumnya. Setiap bar pada notasi yang dihasilkan dirancang untuk mencerminkan karakteristik input dengan menjaga integritas struktur musik melalui penggunaan *logits processor*. Dengan memfilter *token* yang valid, model mampu menghasilkan urutan notasi yang tidak hanya sesuai dengan aturan musik tetapi juga relevan dengan konteks input. Hasil generasi menunjukkan kemampuan model dalam mengintegrasikan elemen-elemen musikal secara konsisten, memberikan keluaran yang mendekati data validasi baik dari segi pola *pitch*, ritme, maupun struktur keseluruhan. Contoh generasi notasi ABC yang dihasilkan oleh model GPT-2 seperti yang ada di Tabel 7.

Tabel 7. Contoh *Output* Generasi Notasi ABC dari GPT-2

Input	Hasil Notasi ABC
Pola Pitch = Slow Ritme = Slow Kunci = Gm	(B,>CD/2E/2)(d>cB/2A/2)(g> ab/2a/2)g2 b3/4-a/4(B,>CD/2E/2)f d fF D G/2-A/2.g/2.b/2.a/2.g/2.f/2.e/2(B> cd/2=e/2)\ B> (c d/4=e/4^f/2)(GAG)g f g/2-a/2G2d g {b}

Pada Tabel 7 menunjukkan contoh input-input yang digunakan oleh model untuk menghasilkan notasi ABC. Input tersebut mencakup pola *pitch*, ritme, dan kunci dasar, yang menjadi dasar dalam pembentukan karakter notasi ABC yang akan dihasilkan.

3.6 Evaluasi dan Visualisasi Data

Proses evaluasi dilakukan dengan membandingkan hasil generasi model dengan data validasi yang telah disiapkan sebelumnya. Perbandingan ini didasarkan pada nilai rata-rata dari hasil perhitungan setiap kategori input, baik pada data validasi maupun data hasil generasi. Pendekatan ini memungkinkan analisis yang lebih komprehensif untuk menilai sejauh mana model mampu mereplikasi pola dan karakteristik musikal dari data validasi, sehingga memastikan kesesuaian antara hasil generasi dengan konteks yang diinginkan

a. Evaluasi Kualitas Musik Berdasarkan Metrik Nada

Hasil evaluasi kualitas musik yang dihasilkan oleh model GPT-2 diukur menggunakan beberapa metrik nada yang mencakup *Average Smoothness*, *Average Consonance*, *Average Rhythmic Variety*, *Average Pitch Range*, *Average Repetition Rate*, dan *Average Tonality Score*. Setiap metrik memberikan perspektif unik dalam menilai kesesuaian notasi yang dihasilkan dengan karakteristik musik yang diharapkan.

Pada penelitian ini, setiap metrik diberi bobot sesuai dengan pentingnya aspek yang dinilai dalam menciptakan hasil dari model [20]. Besaran bobot pada setiap metrik dapat dilihat pada Tabel 8.

Tabel 8. Penerapan Bobot pada Combined Fitness

Nama Metrik	Jumlah Bobot
Smoothness Score	0.3
Consonance Score	0.2
Rhythmic Variety	0.2
Range Score	0.1
Repetition Rate	0.1
Tonality Score	0.1

Smoothness mendapat bobot terbesar karena transisi nada yang mulus menciptakan musik yang lebih harmonis, yang sangat penting untuk *game Indie*. *consonance* dan *rhythmic variety* juga mendapat bobot tinggi, karena harmoni dan variasi ritme berperan besar dalam menjaga keterlibatan pendengar. Sementara itu metrik *range*, *repetition rate*, dan *tonality* diberi bobot lebih kecil, namun tetap signifikan untuk mencerminkan keragaman, pola pengulangan, dan kesesuaian tonalitas dalam komposisi.

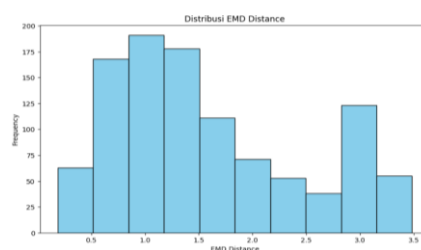
Tabel 9. Hasil Evaluasi dengan Average Combined Fitness

Genre	Mood	Emotional	Avg. Rhythmic Variety	Avg. Consonance	Avg. Smoothness	Avg. Combined Fitness
Horror	Joyful	Excitement	0.791	0.421	0.243	0.528
Puzzle	Epic	Excitement	0.54	0.587	0.424	0.511
Puzzle	Serene	Peaceful	0.877	0.403	0.299	0.495
Horror	Relaxing	Fear	0.444	0.615	0.399	0.492
RPG	Epic	Challenge	0.438	0.575	0.407	0.48
RPG	Joyful	Excitement	0.69	0.384	0.234	0.477

Pada Tabel 9 menunjukkan hasil evaluasi, nilai *combined fitness* tertinggi diperoleh pada *genre Horror* dengan kategori *mood Joyful* dan kategori tingkat *emotional Excitement*, dengan nilai *combined fitness* sebesar 0.528. Hal ini menunjukkan bahwa kombinasi elemen-elemen musik pada konfigurasi tersebut menghasilkan kualitas musik yang paling optimal dibandingkan dengan *genre* dan *mood* lainnya. Kontribusi metrik seperti *rhythmic variety* yang tinggi (0.791) dan *consonance* (0.421) menjadi faktor utama yang mendukung hasil ini.

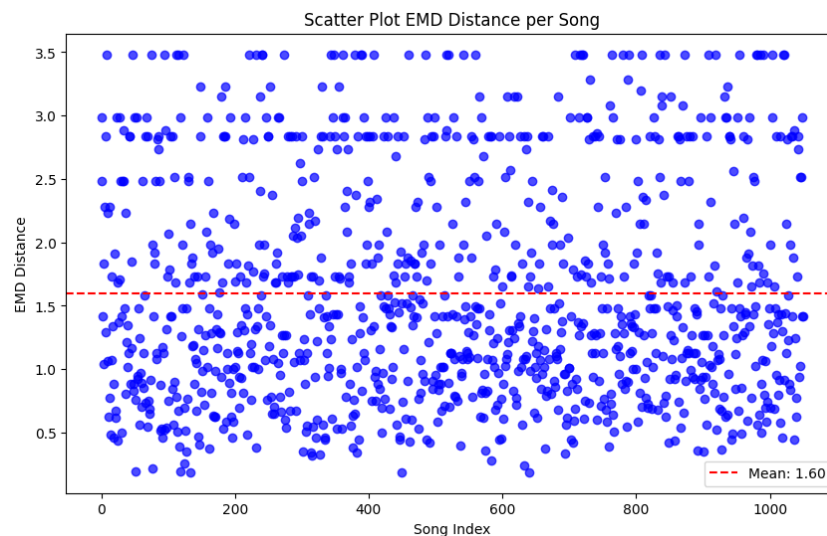
b. Evaluasi Distribusi Pitch dengan Earth Mover's Distance (EMD)

Pada uji evaluasi dengan pendekatan *Earth Mover's Distance* (EMD). Proses evaluasi menggunakan metrik tersebut dilakukan pada hasil generasi teks berbentuk notasi ABC pada salah satu sampel kategori inputan. Yaitu pada *genre Horror*, dengan *mood Joyful*, dan *emotional Peaceful*. Evaluasi dilakukan pada data validasi dengan kategori-kategori yang sama dengan yang diinputkan pada proses generasi.



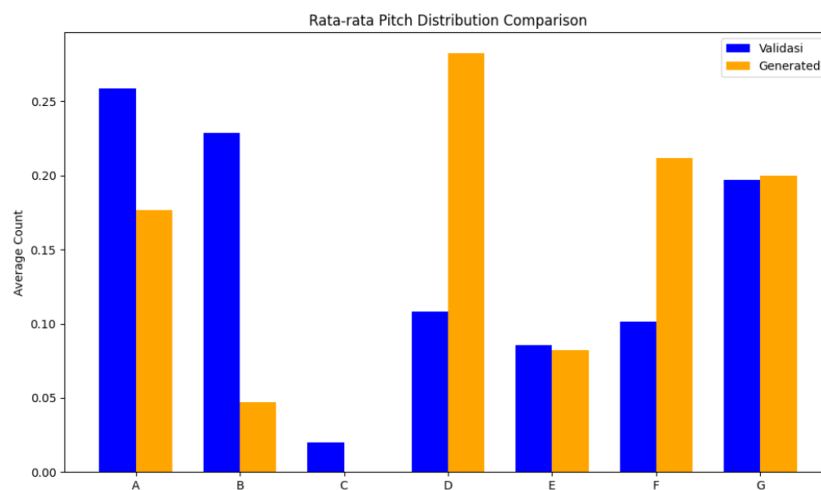
Gambar 2. Visualisasi Distribusi Jarak EMD

Terlihat pada hasil visualisasi di atas, distribusi nilai jarak EMD menunjukkan pola yang terkonsentrasi pada rentang 1.0 hingga 1.5. Hal ini mengindikasikan bahwa mayoritas pola *pitch* yang dihasilkan memiliki kesesuaian yang relatif tinggi dengan data validasi. Namun, terdapat beberapa kasus dengan nilai EMD yang lebih tinggi, yang menunjukkan adanya perbedaan yang signifikan pada pola *pitch* untuk beberapa lagu tertentu. Selain itu juga dilakukan proses visualisasi deteksi *outlier* pada data validasi.



Gambar 3. Visualisasi Deteksi *Outlier* dengan EMD

Scatter plot memperlihatkan variasi jarak EMD antar lagu, dengan beberapa lagu menunjukkan nilai EMD yang jauh di atas rata-rata ($mean = 1.60$, ditandai dengan garis merah). Lagu-lagu dengan nilai EMD di atas rata-rata terindikasi *outlier*, yang mengindikasikan bahwa model mengalami kesulitan dalam menghasilkan pola *pitch* yang sesuai untuk lagu-lagu tersebut.



Gambar 4. Visualisasi Distribusi *Pitch* Hasil Generasi

Penelitian ini mengevaluasi distribusi *pitch* rata-rata pada data validasi dan hasil generasi. Distribusi *pitch* dihitung berdasarkan kemunculan rata-rata *pitch* A hingga G. Sumbu X mewakili label *pitch* (A-G), sedangkan sumbu Y menunjukkan rata-rata frekuensi kemunculan *pitch*.

Hasil visualisasi menunjukkan perbedaan pola distribusi antara data validasi (warna biru) dan hasil generasi (warna oranye). *Pitch* tertentu, seperti D dan F, memiliki distribusi yang lebih tinggi pada hasil generasi, sedangkan *pitch* seperti A dan B lebih dominan pada data validasi. Perbedaan ini mengindikasikan adanya ketidaksesuaian distribusi rata-rata *pitch* antara data validasi dan generasi.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan teknologi Transformer untuk generasi melodi berbentuk notasi ABC berhasil menghasilkan pola *pitch* yang cukup sesuai dengan data validasi, meskipun terdapat tantangan yang harus diatasi. Hasil evaluasi menunjukkan kesesuaian yang baik dengan *Earth Mover's Distance* (EMD) yang terkonsentrasi pada rentang 1.0–1.5 dan rata-rata 1.60. Model juga menunjukkan performa yang baik dalam genre tertentu, seperti

Horror dengan *mood Joyful* dan *Emotional Excitement*, yang didukung oleh variasi ritme dan harmoni yang konsisten. Namun, model ini masih menghadapi keterbatasan dalam menghasilkan distribusi *pitch* yang lebih konsisten, dengan adanya *outlier* yang menunjukkan perbedaan signifikan pada pola *pitch* tertentu. Selain itu, terdapat ketidaksesuaian dalam distribusi *pitch* antara data validasi dan hasil generasi, yang perlu diperbaiki. Oleh karena itu, meskipun teknologi Transformer menunjukkan potensi besar, penelitian lebih lanjut diperlukan untuk meningkatkan akurasi distribusi *pitch* dan kesesuaian pola untuk mencapai hasil yang lebih seragam dan harmonis di berbagai kategori input.

REFERENCES

- [1] W.J. Baltzell, *A Complete History of Music*, 1st ed. Frankfurt am Main, Germany: Outlook Verlag GmbH, 2020. Accessed: Jan. 11, 2025. [Online]. Available: <https://www.gutenberg.org/ebooks/54392>
- [2] R. Soelistijadi, "Prototipe Model Generatif dengan LSTM untuk Penciptaan lagu Campur Sari Didi Kempot," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 4, pp. 3687–3700, Dec. 2022, doi: 10.35957/jatisi.v9i4.2186.
- [3] W. Gibbons and M. Grimshaw-Aagaard, *The Oxford Handbook of Video Game Music and Sound*. Oxford University Press, 2024. doi: 10.1093/oxfordhb/9780197556160.001.0001.
- [4] I. Cardoso and L. N. Ferreira, "The NES Video-Music Database: A Dataset of Symbolic Video Game Music Paired with Gameplay Videos Video-Music Database: A Dataset of Symbolic Video Game Music Paired with Gameplay Videos," *FDG '24: Proceedings of the 19th International Conference on the Foundations of Digital Games*, pp. 1–6, 2024, doi: 10.1145/3649921.
- [5] A. Muhamed *et al.*, "Symbolic Music Generation with Transformer-GANs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 408–417, May 2021, doi: 10.1609/aaai.v35i1.16117.
- [6] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," Sep. 2021, [Online]. Available: <http://arxiv.org/abs/2109.00663>
- [7] H. Wang, S. Hao, C. Zhang, X. Wang, and Y. Chen, "Motif Transformer: Generating Music with Motifs," *IEEE Access*, vol. 11, pp. 63197–63204, 2023, doi: 10.1109/ACCESS.2023.3287271.
- [8] Q. Huang *et al.*, "Noise2Music: Text-conditioned Music Generation with Diffusion Models," Feb. 2023, [Online]. Available: <http://arxiv.org/abs/2302.03917>
- [9] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Müller, and Y.-H. Yang, "Theme Transformer: Symbolic Music Generation With Theme-Conditioned Transformer," *IEEE Trans Multimedia*, vol. 25, pp. 3495–3508, Nov. 2023, doi: 10.1109/TMM.2022.3161851.
- [10] X. Li, X. Li, D. Pan, and D. Zhu, "Improving Adversarial Robustness via Probabilistically Compact Loss with Logit Constraints," pp. 8482–8490, Dec. 2020, Accessed: Dec. 24, 2024. [Online]. Available: <http://arxiv.org/abs/2012.07688>
- [11] A. Amrullah, B. Wiratama, and A. A. Huda, "Performance Analysis of Deep Neural Network based Gamelan Musical Instruments Separation," in *2023 6th International Conference on Information and Communications Technology (ICOIACT)*, IEEE, Nov. 2023, pp. 338–342. doi: 10.1109/ICOIACT59844.2023.10455860.
- [12] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-Attention Attribution: Interpreting Information Interactions Inside Transformer," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, pp. 12963–12971, May 2021, doi: 10.1609/aaai.v35i14.17533.
- [13] Y. Shao *et al.*, "An Improvement of Adam Based on a Cyclic Exponential Decay Learning Rate and Gradient Norm Constraints," *Electronics (Switzerland)*, vol. 13, no. 9, May 2024, doi: 10.3390/electronics13091778.
- [14] J. von Oswald *et al.*, "Transformers Learn In-Context by Gradient Descent," 2023. doi: 10.5555/3618408.3619872.
- [15] R. Yusuf and A. A. Huda, "Deteksi Emosi Wajah Menggunakan Metode Backpropagation," *Journal Automation Computer Information System*, vol. 3, no. 2, pp. 103–114, Sep. 2023, doi: 10.47134/jacis.v3i2.60.
- [16] D. Rothman, *Transformers for Natural Language Processing and Computer Vision*, 3rd ed. Birmingham, UK: Packt Publishing Ltd., 2024. Accessed: Jan. 12, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10162340>
- [17] X. Jin, W. Zhou, J. Wang, D. Xu, Y. Rong, and S. Cui, "An Order-Complexity Model for Aesthetic Quality Assessment of Symbolic Homophony Music Scores," *2023 IEEE International Conference on Multimedia and Expo (ICME)*, Jan. 2023, [Online]. Available: <http://arxiv.org/abs/2301.05908>
- [18] A. S. Subramanian, C. Weng, S. Watanabe, M. Yu, and D. Yu, "Deep Learning based Multi-Source Localization with Source Splitting and its Effectiveness in Multi-Talker Speech Recognition," *Comput Speech Lang*, Feb. 2021, [Online]. Available: <http://arxiv.org/abs/2102.07955>
- [19] L.C. Castilho, R. Dias, and J.F. Pinho, *Perspectives on Music, Sound and Musicology*, vol. 10. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-78451-5.
- [20] K. A. Yuana, A. Hadinegoro, Indarto, Deendarlianto, and E. P. Budiana, "Developing from 2D to 3D Droplet Modeling and Simulation Using Lattice Boltzmann Method (LBM)," in *ICOIACT 2021 - 4th International Conference on Information and Communications Technology: The Role of AI in Health and Social Revolution in Turbulence Era*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 175–179. doi: 10.1109/ICOIACT53268.2021.9564006.