

Implementasi Metode *Long Short-Term Memory* dalam Pembuatan Musik Sintetis

Alfarabi Muzli, Chastine Fatichah, dan Nanik Suciati
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: chastine@if.its.ac.id

Abstrak—Penggunaan teknologi dalam bidang seni telah membuka berbagai peluang baru untuk meningkatkan kinerja dan kreativitas. Dalam industri musik, perkembangan pesat kecerdasan buatan telah memberikan kemungkinan baru dalam pembuatan musik. Namun, keberadaan teknologi ini juga memiliki implikasi terhadap penggunaan musik dan proses transkripsi manual. Musik menjadi sumber daya penting bagi para musisi, telah mengalami perubahan dari bentuk manuskrip tangan menjadi versi digital yang dapat diakses melalui internet. Pelatihan pembuatan musik menggunakan Recurrent Neural Network yaitu Long Short-Term Memory dilakukan dengan dataset terdiri dari tiga orang komponis ternama dalam bidang musik klasik. Dataset ini dibagi menjadi empat bagian. Selain itu, ada dua skenario yang akan diuji, dan hasilnya akan dibandingkan dengan model lain yaitu Gated Recurrent Unit. Untuk pengolahan data musik, digunakan library python bernama Music21. Model dievaluasi dengan empat cara yaitu dengan melihat loss pelatihan model, persebaran statistik notasi, perhitungan kesamaan dan penilaian dari ahli musik. Pada skenario kedua, model LSTM menunjukkan hasil yang lebih baik dibandingkan dengan skenario pertama, dan juga lebih unggul daripada model GRU.

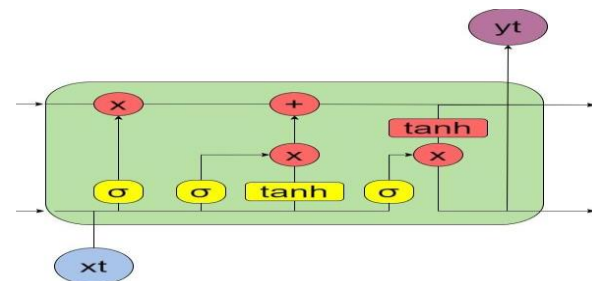
Kata Kunci—*Long Short-Term Memory*, Music21, Musik, Musik Klasik, *Recurrent Neural Network*, Teknologi Dalam Seni.

I. PENDAHULUAN

DALAM bidang seni, teknologi telah memainkan peran yang signifikan dalam meningkatkan performa kerja dan menginspirasi kreativitas pengguna. Keberadaan teknologi telah mengarah pada pengembangan algoritma cerdas yang dapat mengotomatisasi tugas-tugas repetitif dalam seni. Dalam industri musik, teknologi telah digunakan untuk membantu pembuatan musik, dan perkembangan kecerdasan buatan telah membawa dampak yang signifikan. Salah satu area yang menunjukkan kemajuan yang pesat adalah dalam pembuatan dan analisis musik [1].

Seiring dengan semakin luasnya komunitas musik yang melibatkan berbagai kelompok musisi di seluruh dunia, ada kebutuhan yang terus berkembang untuk sumber daya musik yang mudah diakses dan dapat dipertukarkan. Lembaran musik adalah salah satu alat penting dalam mempelajari dan membawakan musik. Namun, proses transkripsi manual dari sumber musik menjadi lembaran musik yang dapat dipertukarkan masih memakan waktu yang lama. Untuk mengatasi tantangan ini, terobosan dalam kecerdasan buatan telah memungkinkan pengembangan Optical Music Recognition (OMR), sebuah pendekatan untuk mentranskripsikan secara otomatis notasi musik dari materi yang dipindai menjadi format digital terstruktur [2].

Namun, kompleksitas struktural notasi musik dan variasi gaya penulisan yang berbeda membuat OMR menjadi tantangan yang sulit. Untuk mengatasi kendala ini, studi terbaru telah memanfaatkan kemajuan dalam bidang



Gambar 1. Arsitektur LSTM.

kecerdasan buatan untuk meningkatkan kemampuan sistem komputasi dalam pengenalan, penciptaan, dan analisis musik. Misalnya, menggunakan arsitektur model seperti Long Short-Term Memory (LSTM) Neural Network, para peneliti telah berhasil menghasilkan musik sintetik secara otomatis.

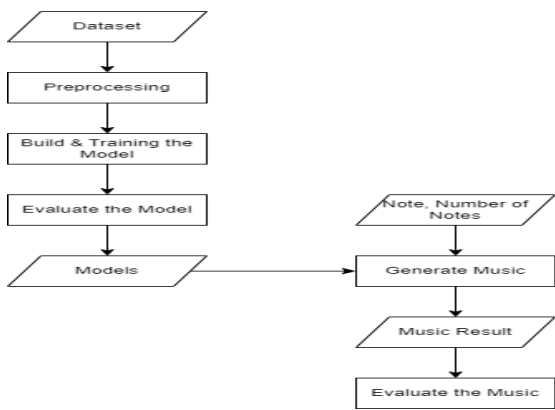
Namun, masih ada ruang untuk eksplorasi lebih lanjut mengenai potensi kecerdasan buatan dalam transkripsi musik otomatis. Dalam konteks ini, penelitian ini bertujuan untuk mengeksplorasi dan mengembangkan metode baru yang menggunakan kecerdasan buatan untuk transkripsi musik secara otomatis. Kami akan memperkenalkan metode yang menggabungkan arsitektur LSTM Neural Network dengan algoritma optimasi Adam dan fungsi aktivasi softmax. Implementasi dan pengolahan data dilakukan menggunakan bahasa pemrograman Python dan platform Google Collaboratory.

II. TINJAUAN PUSTAKA

A. Long Short-Term Memory

Long Short-Term Memory (LSTM) dikenalkan pada pertengahan 90-an oleh peneliti dari Jerman sebagai variasi dari Recurrent Net. LSTM merupakan langkah yang lebih maju dari Recurrent Neural Network (RNN) tradisional. Salah satu permasalahan utama pada RNN adalah masalah vanishing gradient, dan LSTM dirancang dengan tujuan mengatasi permasalahan ini. LSTM mengadopsi pendekatan yang selektif dalam memproses dan memfilter informasi yang ada. Dibandingkan dengan RNN tradisional, LSTM memiliki kemampuan yang lebih baik dalam mengingat pola dalam jangka waktu yang lama [3]. Dalam LSTM, terdapat tiga jenis gerbang yang bertanggung jawab untuk mengatur aliran informasi di dalamnya, yaitu gerbang input (input gate), gerbang lupa (forget gate), dan gerbang keluaran (output gate). Gerbang input memutuskan informasi baru mana yang harus masuk ke dalam sel memori, gerbang lupa menentukan informasi mana yang harus dihapus dari sel memori, dan gerbang keluaran mengendalikan bagaimana informasi yang ada di dalam sel memori dikeluarkan.

Dengan adanya gerbang-gerbang ini, LSTM dapat secara efektif memproses dan mengingat informasi yang relevan



Gambar 2. Flowchart generasi musik.

Tabel 1. Spesifikasi Dataset Awal perkasus

Dataset	Banyak Notasi
1	141259 Notasi
2	4145 Notasi
3	81312 Notasi
4	55802 Notasi

dalam jangka waktu yang lebih panjang. Selain itu, struktur LSTM juga memiliki mekanisme yang disebut dengan cell state, yang bertindak sebagai "pipa" yang membawa informasi melalui urutan waktu. Cell state bertugas untuk menyimpan dan membawa informasi yang relevan sepanjang urutan waktu. Dengan adanya cell state, LSTM dapat menjaga informasi yang penting dan mencegah hilangnya informasi yang berharga selama proses pembelajaran [4]. Arsitektur LSTM dapat dilihat pada Gambar 1.

B. Keras

Keras, sebagai API dari deep learning, memiliki kemampuan yang sangat berguna dalam pengembangan model machine learning menggunakan TensorFlow di lingkungan Python. Dikembangkan dengan fokus pada eksperimen yang cepat, Keras menawarkan pendekatan yang mempermudah pengguna manusia untuk bekerja dengan model neural network. Melalui pengurangan cognitive load, Keras memberikan pengguna API yang sederhana dan konsisten, sehingga memudahkan proses pengembangan dan iterasi model. Sebagai wrapper TensorFlow, Keras tidak hanya menyederhanakan penggunaan neural network, tetapi juga menyediakan banyak implementasi dari blok bangunan neural network yang umum digunakan, termasuk lapisan (layers), target, fungsi aktivasi, pengoptimal, dan alat-alat lainnya. Terlebih lagi, Keras juga menyediakan berbagai fitur yang mempermudah pemrosesan data gambar dan teks, serta secara signifikan mengurangi kompleksitas pengkodean yang diperlukan untuk menulis model deep neural network yang kompleks. Dalam kombinasi dengan Tensorflow, Keras menjadi alat yang sangat berharga dalam mengembangkan model deep learning yang efisien dan kuat.

C. TensorFlow

TensorFlow adalah open-source library perangkat lunak untuk machine learning, terutama pada Deep learning. TensorFlow dibuat oleh Google dan dapat digunakan dengan berbagai bahasa pemrograman. TensorFlow menggunakan aliran data dan pemrograman yang dapat dibedakan untuk melakukan berbagai tugas dengan tensor, yang merupakan array atau matriks multidimensi. Ini menawarkan tools,

Tabel 2. Statistik data hasil dari setiap dataset pada skenario 1

Dataset	Statistik Hasil	Nilai
1	Rata-rata frekuensi notasi	3.03
	Notasi dengan jumlah terbesar	9
	Notasi dengan jumlah terkecil	1
2	Rata-rata frekuensi notasi	100
	Notasi dengan jumlah terbesar	100
	Notasi dengan jumlah terkecil	100 (0)
3	Rata-rata frekuensi notasi	4.17
	Notasi dengan jumlah terbesar	29
	Notasi dengan jumlah terkecil	1
4	Rata-rata frekuensi notasi	9.09
	Notasi dengan jumlah terbesar	63
	Notasi dengan jumlah terkecil	1

Tabel 3. Hasil perhitungan Kesamaan skenario 1

Dataset	Similarity Notasi (%)	Similarity Akor (%)
1	0.01	0.00
2	0.03	0.00
3	0.01	0.00
4	0.02	0.00

Tabel 4. Penilaian ahli pada skenario 1

Dataset	Nilai	Komentar
1	57	Pola nada beberapa yang kurang pas didengar
2	20	Tidak bagus didengar karena hanya 1 notasi saja
3	80	Pola beberapa notasi bagus didengar
4	28	Banyak notasi yang sama, terlalu banyak nada yang sama

library, dan sumber daya komunitas bagi para peneliti dan pengembang untuk membangun dan menerapkan aplikasi machine learning di berbagai lingkungan.

D. Python

Dalam konteks Machine Learning, Python telah menjadi bahasa pemrograman yang sangat populer. Python menyediakan berbagai library dan framework yang kuat untuk membangun dan mengimplementasikan model Machine Learning. Beberapa library populer untuk Machine Learning di Python termasuk NumPy, Pandas, Scikit-learn, TensorFlow, dan PyTorch. Library-library ini menyediakan fungsi-fungsi yang siap pakai untuk memproses data, membangun model, melatih model, dan mengevaluasi hasilnya. Selain itu, Python juga digunakan secara luas untuk deployment model Machine Learning. Setelah model Machine Learning dikembangkan, Python dapat digunakan untuk membuat antarmuka atau API yang memungkinkan pengguna lain untuk mengakses dan menggunakan model tersebut. Python juga dapat digunakan untuk mengintegrasikan model Machine Learning dengan sistem yang lebih besar atau aplikasi yang sedang berjalan. keunggulan Python adalah kemampuannya sebagai bahasa pemrograman umum yang dapat digunakan untuk menyelesaikan berbagai jenis masalah di sistem komputer modern. Python menyediakan sintaksis yang mudah dibaca dan jelas, sehingga lebih mudah bagi pengembang untuk mengekspresikan ide dan solusi pemrograman [5].

E. Music21

Selain itu, Music21 juga memberikan kemampuan untuk melakukan analisis musik secara mendalam. Dengan menggunakan toolkit ini, pengguna dapat menjawab pertanyaan-pertanyaan yang kompleks mengenai struktur

Tabel 5.

Statistik data hasil pada setiap dataset pada skenario 2

Dataset	Statistik Hasil	Nilai
1	Rata-rata frekuensi notasi	2.50
	Notasi dengan jumlah terbesar	10
	Notasi dengan jumlah terkecil	1
2	Rata-rata frekuensi notasi	3.23
	Notasi dengan jumlah terbesar	9
	Notasi dengan jumlah terkecil	1
3	Rata-rata frekuensi notasi	2.38
	Notasi dengan jumlah terbesar	11
	Notasi dengan jumlah terkecil	1
4	Rata-rata frekuensi notasi	7.69
	Notasi dengan jumlah terbesar	62
	Notasi dengan jumlah terkecil	1

Tabel 6.

Hasil perhitungan Kesamaan skenario 2

Dataset	Similarity Notasi (%)	Similarity Akor (%)
1	0.03	0.05
2	0.08	2.72
3	0.05	0.07
4	0.03	0.05

Tabel 7.

Penilaian ahli pada skenario 2

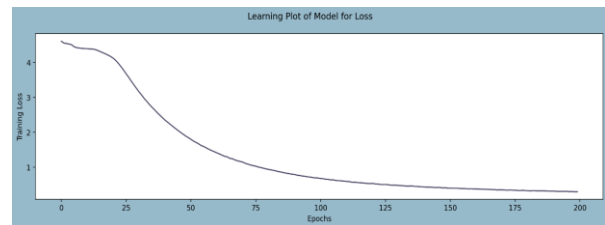
Dataset	Nilai	Komentar
1	85	Memiliki inovasi notasi dan bagus didengar
2	75	Menggunakan harmoni, walaupun masih ada notasi yang tidak cocok
3	86	Pola notasi bagus untuk didengar dan mudah diingat
4	29	Bagian awal sudah bagus, tetapi di pertengahan mulai menggunakan 1 notasi sampai akhir

musik, harmoni, melodi, ritme, dan elemen-elemen lain yang ada dalam sebuah komposisi musik. Salah satu keunggulan utama dari Music21 adalah kemampuannya dalam mempelajari dataset musik yang besar. Dengan menggunakan algoritma dan metode yang disediakan oleh toolkit ini, pengguna dapat menganalisis dataset musik yang terdiri dari ribuan atau bahkan jutaan lagu. Hal ini memungkinkan para peneliti musik untuk mengidentifikasi pola-pola umum, tren, atau karakteristik khas dari suatu genre musik, periode waktu tertentu, atau bahkan dari komposer-komposer tertentu.

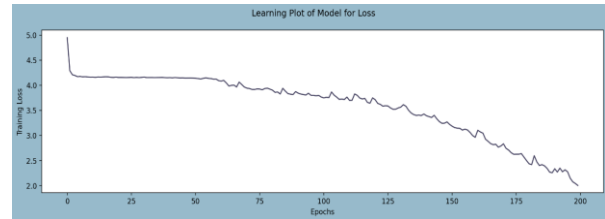
Selain sebagai alat analisis, Music21 juga dapat digunakan untuk menghasilkan lembaran-lembaran musik. Dengan memanfaatkan fitur komposisi dari toolkit ini, pengguna dapat menciptakan musik baru secara otomatis atau dengan menggabungkan elemen-elemen yang ada dalam dataset musik yang telah ada. Hal ini sangat berguna bagi para komposer, pengaransemen, atau musisi yang ingin menciptakan musik dengan berbagai gaya, genre, atau karakteristik yang berbeda. Sebagai alat pembelajaran, Music21 memberikan pendekatan yang unik dalam pengajaran dasar teori musik. Dengan menggunakan pendekatan komputasional, toolkit ini dapat membantu pemula dalam memahami konsep-konsep dasar seperti notasi musik, akor, skala, modulasi, dan sebagainya.

F. Evaluasi

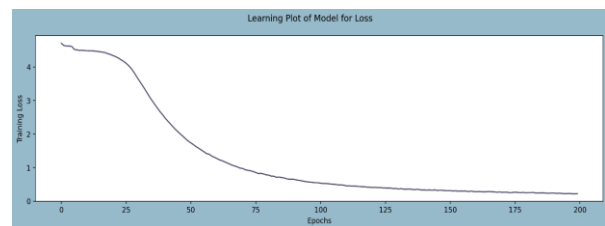
Untuk meningkatkan kualitas evaluasi model, terdapat empat pendekatan yang akan digunakan. Pertama-tama, evaluasi akan dilakukan dengan mempertimbangkan loss yang dihasilkan oleh model. Loss ini merupakan metrik yang menggambarkan sejauh mana model dapat mempelajari pola



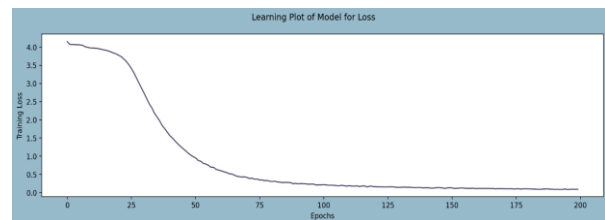
Gambar 3. Grafik loss dataset 1 skenario 1.



Gambar 4. Grafik loss dataset 2 skenario 2.



Gambar 5. Grafik loss dataset 3 skenario 1.



Gambar 6. Grafik loss dataset 4 skenario 1.

dan struktur musik yang diinputkan. Semakin rendah nilai loss yang didapatkan, semakin baik pula kualitas model tersebut.

Evaluasi selain menggunakan loss akan menggunakan perbandingan kesamaan antara dataset dengan musik yang telah digenerasi. Hal ini untuk menunjukkan bagaimana hasil dipengaruhi oleh dataset berdasarkan persebaran notasi dan akornya. Evaluasi ini dilakukan dengan membagi himpunan irisan notasi antara dataset awal dan musik yang dihasilkan dengan panjang dataset awal. Selain itu, evaluasi akan melibatkan analisis statistik notasi pada musik yang dihasilkan oleh model. Dengan melihat persebaran notasi rata-rata, persebaran notasi terbesar dan terkecil berdasarkan jumlah kemunculan mereka, dapat diperoleh pemahaman tentang kualitas dari musik tanpa harus mendengar musiknya terlebih dahulu.

Pendekatan ketiga dalam evaluasi model adalah melibatkan pendapat ahli musik. Ahli ini merupakan seorang lulusan fakultas musik dan merupakan seorang guru piano. Ahli musik akan melakukan penilaian langsung terhadap musik yang dihasilkan oleh model, memberikan komentar dan memberikan nilai-nilai sebagai ukuran objektif. Rentang nilai yang digunakan oleh ahli musik dalam penilaian ini adalah dari 0 hingga 100, dimana nilai 0 menunjukkan musik yang tidak layak didengar sedangkan nilai 100 menunjukkan sebuah karya yang sangat bagus.

Evaluasi perhitungan kesamaan, perhitungan statistik persebaran notasi dalam musik, dan evaluasi penilaian ahli

merupakan bentuk lain dari penilaian model. Jika hasil loss kecil, tetapi berdasarkan kesamaan, statistik, dan ahli menyatakan bahwa musik bernilai tinggi maka model memiliki kinerja yang bagus. Jadi tidak hanya terpaku pada satu evaluasi saja untuk menilai apakah model tersebut bekerja dengan baik atau tidak..

III. METODOLOGI

A. Metode yang Dirancang

Gambar 2 menunjukkan bagaimana aliran penelitian ini dilakukan. Dimulai dari pengumpulan dataset yang akan digunakan sebagai bahan latih model untuk generasi musik. Dilanjutkan dengan tahap preprocessing yang mana akan mengolah dataset mentah menjadi dataset yang siap digunakan, ada beberapa tahap yang dilalui seperti ekstrak notasi, pendefinisian skenario, pemetaan notasi ke bilangan bulat, dan pembagian data input dan outputnya. Selanjutnya model dibangun dengan menggunakan spesifikasi yang telah ditentukan dan menggunakan dataset yang telah disiapkan untuk melatih model. Setelah itu, dilakukan evaluasi awal terhadap kinerja model dengan loss, dan model akan digunakan untuk melakukan generasi musik. Sebelum menggenerasi musik, pengguna atau user melakukan input spesifikasi musik yang diinginkan seperti dimulai dari notasi apa dan seberapa banyak notasi yang akan di generasi dalam musik. Musik yang dihasilkan akan dievaluasi dengan beberapa cara yaitu perhitungan kesamaan antara dataset dan hasil musik, perhitungan persebaran statistik notasi dalam musik, dan penilaian secara subjektif dari seorang ahli musik.

B. Dataset

Untuk melatih model Long Short-Term Memory (LSTM) yang dimaksud, diperlukan adanya dataset yang memadai. Dataset yang digunakan dalam konteks ini adalah dataset file MIDI yang berasal dari tiga komposer terkenal, yaitu Johann Sebastian Bach, Ludwig van Beethoven, dan Wolfgang Amadeus Mozart. Setiap komposer menyumbangkan berbagai jumlah file MIDI yang berbeda-beda, menciptakan keragaman musik yang diperlukan dalam proses pelatihan. MIDI sendiri merupakan singkatan dari Musical Instrument Digital Interface yang berguna untuk menciptakan musik dengan bentuk partitur, mengolah partitur digital, dan sebagainya yang berhubungan dengan musik dan partitur.

Dalam rangka melatih model LSTM, akan dibentuk empat dataset yang akan digunakan. Dataset pertama merupakan gabungan dari seluruh karya musik dari ketiga komposer tersebut. Dataset ini dipakai karena mengandung semua gaya dan karakteristik musik dari Johann Sebastian Bach, Ludwig van Beethoven, dan Wolfgang Amadeus Mozart. Sementara itu, dataset kedua, ketiga, dan keempat mewakili masing-masing komposer secara individual, dengan dataset kedua berisi karya-karya Johann Sebastian Bach, dataset ketiga berisi karya-karya Ludwig van Beethoven, dan dataset keempat berisi karya-karya Wolfgang Amadeus Mozart.

C. Praproses

Setelah melakukan pembacaan dataset, langkah selanjutnya adalah melakukan pemisahan data-notasi dan akor yang ada. Untuk melakukan pemisahan tersebut, digunakan fungsi yang disediakan oleh library Python music21. Fungsi tersebut terdiri dari dua bagian, yaitu fungsi

"note" untuk mengelola data-notasi dan fungsi "chord" untuk mengelola data akor. Penerapan fungsi-fungsi ini dilakukan untuk setiap dataset yang ada. Hasil dari pemisahan data-notasi adalah sebuah list notasi, yang umumnya disajikan dalam bentuk Scientific Pitch Notation. List notasi ini berisi kumpulan notasi-notasi seperti ['E5', 'F#5', 'C4', 'D5'].

Dalam Scientific Pitch Notation, setiap notasi memiliki dua komponen utama: nama notasi dan oktaf. Sebagai contoh, notasi 'E5' mengindikasikan bahwa notasi tersebut adalah E dan berada pada oktaf 5 (dengan catatan bahwa oktaf default adalah oktaf 4). Sementara itu, data akor memiliki bentuk angka, seperti contohnya '5.7.11'. Angka-angka ini mewakili kumpulan notasi-notasi yang membentuk sebuah akor, dan penentuan arti dari setiap angka tersebut telah ditentukan oleh library music21. Spesifikasi dataset awal perkasus dapat dilihat pada Tabel 1.

Visualisasi persebaran notasi pada setiap dataset dilakukan untuk menunjukkan bahwa terdapat variasi jumlah notasi yang berbeda-beda. Beberapa notasi memiliki jumlah di bawah 100, sedangkan yang lain memiliki jumlah di atas 100. Untuk mengkaji pengaruh jumlah notasi dan variasi notasi terhadap pelatihan model dan hasil musik yang dihasilkan, dataset tersebut akan dibagi menjadi dua skenario.

Skenario 1 akan terdiri dari kumpulan notasi dengan jumlah di atas 100. Dalam skenario ini, model akan dilatih dengan fokus pada frekuensi atau kemunculan setiap notasi didalam dataset. Tujuannya adalah untuk melihat bagaimana model bereaksi terhadap frekuensi yang lebih besar dalam jumlah notasi dan apakah hal tersebut mempengaruhi hasil musik yang dihasilkan. Skenario 2, di sisi lain, akan mencakup kumpulan notasi dengan jumlah di bawah 100. Dalam skenario ini, model akan dilatih dengan memperhatikan variasi dari notasinya dikarenakan kemunculan yang sedikit tetapi bervariasi. Hal ini bertujuan untuk mempelajari apakah dengan berfokus pada variasi notasi akan menghasilkan musik yang berbeda atau memiliki kualitas yang berbeda dibandingkan dengan berfokus pada jumlah kemunculan setiap notasi.

D. Membangun model dan melatih model

Setiap model akan memiliki 2 layer LSTM/GRU dengan masing-masing 512 dan 256 neurons. Pada layer pertama juga akan didefinisikan input shape nya yang diambil dari shape setiap training data nya. Selain itu, pada setiap model ditambahkan layer Dropout dan Dense serta menggunakan fungsi aktivasi softmax. Selanjutnya melakukan compile model dengan loss menggunakan 'categorical_crossentropy' dan optimizer nya adalah Adamax (dengan learning rate 0.01). Setelah model dibangun, selanjutnya melakukan training model dengan parameter data training X dan y, epoch sebesar 200 dan batch_size sebesar 256.

E. Menggenerasi Lagu

Setelah mengevaluasi, model akan digunakan untuk membuat lagu baru berdasarkan data yang telah di training. Melakukan generasi untuk menentukan notasi berikutnya dengan cara membaca input data dan mengeluarkan output berupa notasi berikutnya yang memiliki suasana atau genre yang sama atau melodi yang mirip dengan input. Pada tahap ini tidak mengeluarkan output yang memiliki genre atau aliran musik yang berbeda dengan input. Jika input memiliki aliran musik klasik, maka output juga berupa aliran musik

klasik. Dan, karena pada dataset merupakan notasi berbentuk piano, maka semua output yang didapat dalam bentuk notasi piano. Setelah menemukan output sesuai, output akan dikonversi kedalam file MIDI.

IV. HASIL UJI COBA DAN EVALUASI

Seperti yang sebelumnya telah disebutkan bahwa terdapat 4 dataset dan 2 skenario. Berikut merupakan hasil ujicoba untuk setiap skenario.

A. Skenario 1

Dataset sebelumnya melalui proses pra proses dimana data akan diambil yang memiliki jumlah diatas 100 sehingga berisikan notasi-notasi yang mendominasi jumlahnya pada satu *Corpus*. Tabel 2 menunjukkan perhitungan statistik berupa rata-rata frekuensi dari setiap notasi, notasi dengan jumlah terbanyak, dan notasi dengan jumlah terkecil untuk setiap dataset.

Tabel 2 menunjukkan bagaimana persebaran notasi pada suatu *Corpus* musik baru. Didapat bahwa dengan dataset kecil seperti dataset 2, akan mendapatkan nilai yang cukup jelek karena memiliki persebaran notasi berjumlah 100 dimana sebelumnya musik hanya digenerasi sebesar 100 notasi saja.

Tabel 3 menunjukkan bagaimana hasil perhitungan kesamaan musik yang digenerasi terhadap dataset. Hal ini dilakukan untuk melihat apakah musik hasil generasi mengadopsi gaya-gaya penulisan maupun pola-pola notasi yang membentuk dataset.

Tabel 4 juga menunjukkan hasil penilaian dari ahli yang memiliki bidang pada seni musik. Musik akan dinilai apakah termasuk bagus apa tidak dengan nilai dalam rentang 0-100.

B. Skenario 2

Dataset sebelumnya melalui proses pra proses dimana data akan diambil yang memiliki jumlah dibawah 100 sehingga berisikan notasi-notasi dengan jumlah sedikit tetapi memiliki varian banyak. Tabel 5 menunjukkan hasil perhitungan statistik berupa rata-rata frekuensi dari setiap notasi, notasi dengan jumlah terbanyak, dan notasi dengan jumlah terkecil berdasarkan setiap dataset.

Tabel 5 menunjukkan bagaimana persebaran notasi didalam *Corpus* musik yang telah digenerasi. Pada dataset 2, hasilnya sangat berbeda pada hasil tabel 2 yang mana memiliki nilai 100 untuk persebaran notasinya. Dataset 4 untuk kedua skenario, persebarannya tidak terlalu berbeda.

Tabel 6 menunjukkan bagaimana hasil perhitungan kesamaan musik yang digenerasi terhadap dataset. Hal ini dilakukan untuk melihat apakah musik hasil generasi mengadopsi gaya-gaya penulisan maupun pola-pola notasi yang membentuk dataset.

Tabel 7 juga menunjukkan hasil penilaian dari ahli yang memiliki bidang pada seni musik. Musik akan dinilai apakah termasuk bagus apa tidak dengan nilai dalam rentang 0-100.

C. Evaluasi Model

Performa model LSTM pada dataset 1 untuk skenario 1 sudah terbukti sangat memuaskan, dengan mencapai nilai loss sebesar 0.30. Hasil ini mengindikasikan bahwa model tersebut memiliki kemampuan yang sangat baik dalam mengidentifikasi dan mempelajari pola-pola musik yang ada,

serta memungkinkan untuk menghasilkan keluaran musik yang cukup baik. Di sisi lain, LSTM untuk skenario 2 juga menunjukkan kinerja yang baik dengan nilai loss sebesar 0.37. Meskipun demikian, jika dibandingkan dengan skenario 1, dapat disimpulkan bahwa model LSTM pada skenario 1 memiliki performa yang lebih unggul berdasarkan perbandingan nilai loss yang diperoleh. Hal ini menunjukkan bahwa model LSTM pada skenario 1 mampu secara lebih akurat menangkap esensi dan pola-pola musik yang ada dalam dataset. Gambar 3 menunjukkan grafik *loss* pada dataset 1 skenario 1.

Pada dataset 2 untuk skenario 1, LSTM menunjukkan performa yang kurang memuaskan dengan nilai loss yang didapat sebesar 3.43, mengindikasikan bahwa model sulit untuk memahami pola musik secara keseluruhan. Salah satu faktor penyebabnya adalah jumlah dataset yang terbatas, sehingga model tidak dapat bekerja secara maksimal dan menghasilkan prediksi yang akurat. Dengan dataset yang lebih besar, model LSTM kemungkinan akan mampu menangkap pola musik dengan lebih baik. Hal yang serupa juga terjadi pada skenario 2, meskipun performanya sedikit lebih baik dibandingkan dengan skenario 1, namun tetap terdapat kekurangan dalam performa model dengan nilai loss mencapai 2.00. Hal ini menunjukkan bahwa model LSTM masih belum sepenuhnya efektif dalam menghasilkan prediksi yang memuaskan dalam kasus dataset 2 ini. Selain itu, ketika melihat hasil generasi musik untuk skenario 1, hanya terdapat 1 notasi yang dihasilkan dari awal hingga akhir. Gambar 4 menunjukkan hasil grafik *loss* pada dataset 2 skenario 2.

Pada dataset 3 skenario 1, model LSTM telah menunjukkan performa yang sangat baik dibandingkan dengan model-model sebelumnya. Dalam skenario ini, model berhasil mencapai nilai loss sebesar 0.22, menunjukkan kemampuannya yang luar biasa dalam memahami pola musik dan memungkinkan untuk menghasilkan musik yang cukup bagus. Keberhasilan ini dapat diatribusikan pada beberapa faktor, salah satunya adalah penggunaan dataset yang tepat secara jumlah. Untuk skenario 2, model cukup memuaskan walaupun tidak melebihi skenario 1 dengan loss sebesar 0.32. Gambar 5 menunjukkan grafik *loss* pada dataset 3 skenario 1.

Terkait dengan dataset terakhir, pada skenario 1 terlihat bahwa model yang dihasilkan menunjukkan performa yang sangat baik. Dengan nilai loss sebesar 0.09, model pada skenario ini berhasil mencapai hasil yang paling bagus dibandingkan dengan model-model sebelumnya. Performa yang lebih baik pada skenario ini memberikan keyakinan bahwa musik yang dihasilkan oleh model tersebut memiliki kualitas yang tinggi. Selanjutnya, pada skenario 2 juga terdapat perkembangan yang signifikan. Meskipun tidak sebaik skenario 1, model pada skenario 2 berhasil mencapai tingkat kehilangan sebesar 0.21, yang menjadikannya sebagai model terbaik kedua dalam percobaan tersebut. Hasil ini menunjukkan peningkatan kinerja yang cukup signifikan dibandingkan dengan model-model sebelumnya. Gambar 6 menunjukkan grafik *loss* pada dataset 4 skenario 1.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dalam pengerjaan penelitian ini, setelah melalui tahap

perancangan, implementasi metode, serta uji coba, diperoleh kesimpulan sebagai berikut:(1)Data musik yang digunakan telah diolah pada tahap praproses untuk dibersihkan datanya atau data yang layak digunakan. Data musik juga telah dibagi menjadi dua skenario yang memiliki spesifikasinya masing-masing. Pada setiap skenarionya, data musik juga telah di *mapping* kedalam bentuk bilangan bulat agar mudah diolah oleh model;(2)Hasil eksperimen menunjukkan bahwa model LSTM dan variannya mampu menghasilkan musik sintetis yang memiliki struktur melodi dan harmoni yang baik. Model yang digunakan telah melalui proses pelatihan dengan menggunakan data musik yang telah diolah;(3)Telah dilakukan evaluasi kinerja dari model yang diimplementasikan. Metode evaluasi dengan *loss function* untuk melihat bagaimana kinerja model dalam mempelajari pola notasi musik yang diberikan, dan didapat model dari dataset musik 4 pada kedua skenario memiliki kinerja yang bagus. Perhitungan kesamaan telah dilakukan pada hasil musik, dan didapatkan pada skenario 1 semua musik memiliki nilai yang cukup besar dibandingkan dengan skenario 2. Perhitungan persebaran statistik notasi juga telah dilakukan, dan didapatkan bahwa hasil pada skenario 2 lebih bagus dan berdasarkan pendapat ahli musik yang menyatakan bahwa hasil musik pada skenario 2 lebih bagus.

B. Saran

Saran yang diberikan untuk Pengembangan penelitian ini

sebagai berikut;(1)Pengembangan pembuatan musik dengan cara mempertimbangkan durasi dari setiap notasi pada dataset, sehingga dapat menghasilkan musik yang lebih beragam;(2)Pengembangan pembuatan musik dengan menggunakan toolkit python lainnya, seperti PyDub agar pengolahan data tidak terfokus pada notasi saja tetapi berfokus pada sinyal suara musiknya;(3)Pengembangan pembuatan musik dengan menggunakan genre musik lainnya, seperti musik jazz dan musik pop.

DAFTAR PUSTAKA

- [1] J. W. Hong, K. Fischer, Y. Ha, and Y. Zeng, "Human, I wrote a song for you: An experiment testing the influence of machines' attributes on the AI-composed music evaluation," *Comput Human Behav*, vol. 131, Jun. 2022, doi: 10.1016/j.chb.2022.107239.
- [2] F. J. Castellanos, C. Garrido-Munoz, A. Ríos-Vila, and J. Calvo-Zaragoza, "Region-based layout analysis of music score images," *Expert Syst Appl*, vol. 209, Dec. 2022, doi: 10.1016/j.eswa.2022.118211.
- [3] M. Dua, R. Yadav, D. Mangai, and S. Brodiya, "An Improved RNN-LSTM based novel approach for sheet music generation," *Procedia Comput Sci*, vol. 171, pp. 465–474, 2020, doi: 10.1016/j.procs.2020.04.049.
- [4] M. Alizamir *et al.*, "Improving the accuracy of daily solar radiation prediction by climatic data using an efficient hybrid deep learning model: Long short-term memory (Lstm) network coupled with wavelet transform," *Eng Appl Artif Intell*, vol. 123, Aug. 2023, doi: 10.1016/j.engappai.2023.106199.
- [5] K. A. Lambert, *Fundamental of Python: First Programs*, 2nd ed. USA: Cengage, 2017. ISBN: 978-1337-56009-2.