

Eksplorasi Deep Learning Menghasilkan Karya Musik Menggunakan Metode Generative Adversarial Networks (GANs) (Kasus Musik Genre Pop)

Noviyanti. P¹, Yuliana², Listra Firgia³, Veneranda Rini Hapsari⁴

^{1,2,3,4}Institut Shanti Bhuana, Bengkulu, Indonesia

e-mail: noviyanti@shantibhuana.ac.id¹, yuliana@shantibhuana.ac.id²,

listrafirgia@shantibhuana.ac.id³, rini@shantibhuana.ac.id⁴

Abstract

Music artistry is an enduring form of artistic expression that continues to evolve across various genres. Among these genres, pop music stands out as particularly popular. Creating musical compositions is a challenging endeavor, requiring a profound understanding of musical notation, a skill possessed by select individuals, such as musicians. Even for musicians, a wealth of references is necessary to produce fresh compositions that can be appreciated by a wide audience. This study aims to explore the creation of new pop genre music using Generative Adversarial Networks (GANs). GANs, a widely adopted method, demonstrate the capability to generate novel works by leveraging two distinct components: the Generator and the Discriminator. These models engage in a competitive interplay, with the Generator striving to produce synthetic datasets that closely resemble authentic ones, while the Discriminator endeavors to discern between datasets generated by the Generator and genuine ones. Based on the conducted research, it is evident that GANs have the capacity to generate a diverse range of new music based on acoustic piano instrument notations, employing a dataset of 50 music files in .mid format.

Keywords: artwork of music, pop music genre, GANs, generator and discriminator, google colab.

Abstrak

Karya seni musik merupakan karya seni yang tidak akan pernah hilang. Musik dapat terus dikembangkan dengan berbagai genre. Salah satu genre musik yang cukup populer adalah musik genre pop. Menghasilkan karya musik merupakan hal yang tidak mudah. Perlu pemahaman lebih terkait not musik dan hanya orang-orang tertentu yang dapat melakukannya, misalnya musisi. Namun, musisi sekalipun perlu banyak referensi untuk dapat menghasilkan musik karya baru yang tentu saja dapat dinikmati oleh masyarakat luas. Penelitian ini bertujuan melakukan eksplorasi dalam menghasilkan musik baru genre pop menggunakan metode GANs. Metode GANs cukup populer dan mampu menghasilkan karya baru dengan mengandalkan 2 komponen unik yang dimiliki oleh metode GANs, yaitu Generator dan Diskriminator. Kedua model tersebut saling beradu, Generator berusaha menghasilkan dataset palsu namun menyerupai dataset asli, sedangkan Diskriminator berusaha untuk melawan Generator dengan membedakan dataset yang dihasilkan oleh Generator dengan dataset asli. Berdasarkan penelitian yang telah dilakukan, metode GANs dapat menghasilkan musik baru yang cukup bervariasi berdasarkan pada not-not instrumen piona akustik menggunakan 50 dataset file musik dengan format .mid.

Kata kunci: karya seni musik, musik genre pop, GANs, generator dan diskriminator, google colab.

1. PENDAHULUAN

Deep Learning merupakan bagian dari *Machine Learning* yang memanfaatkan jaringan saraf tiruan (*Neural Networks*). *Deep Learning* memiliki struktur yang kompleks dan terdiri dari banyak lapisan (*deep layers*). Lapisan lapisan atau biasa disebut *deep layers* bertujuan untuk melakukan pembelajaran pada data hingga menghasilkan suatu model. *Deep Learning* dapat menghasilkan suatu model yang pendekatannya mirip dengan cara kerja otak manusia untuk memproses suatu informasi. *Deep Learning* saat ini populer di beberapa kasus, seperti pengenalan wajah, deteksi objek, pengenalan suara, pemrosesan bahasa alami, dan sebagainya [1]. *Deep learning* memiliki beberapa metode, salah satunya metode Generative Adversarial Networks (GANs). Metode GANs merupakan metode yang melibatkan dua (2) komponen yang saling bertarung untuk menghasilkan sebuah model. Dua (2) komponen yang dimaksud adalah Generator dan Diskriminator. Hal menarik dari metode GANs ini adalah dapat menghasilkan data baru menyerupai data asli. Generator dan Diskriminator saling bersaing dan berkolaborasi dalam menghasilkan data baru tersebut. Model pertama disebut Generator, bertugas untuk menghasilkan data baru yang menyerupai data asli dalam artian menghasilkan data palsu atau tiruan, sedangkan model kedua disebut Discriminator, bertugas untuk membedakan antara data asli dan data palsu yang dihasilkan oleh Generator. Melalui proses latihan yang terus-menerus, kedua model ini dapat meningkatkan kemampuan mereka dalam menghasilkan data baru yang semakin realistis [2].

Kasus yang ingin diselesaikan pada penelitian ini, yaitu musik *genre* pop. Adapun tujuan penelitian ini untuk menghasilkan sebuah model menggunakan metode GANs. Model tersebut akan digunakan untuk menghasilkan suatu data baru dan data baru tersebut berupa musik yang baru dan tidak ada pada dataset pelatihan. Musik merupakan karya seni yang perlu untuk dikembangkan dan di-*update* berdasarkan musik yang sudah ada saat ini. Musik terus dikembangkan dengan berbagai *genre* dan bertujuan untuk memberikan kenyamanan bagi penikmat musik. Musik dengan *genre* pop menjadi salah satu *genre* musik yang disukai oleh para penikmat musik. Karya seni musik dapat memiliki berbagai bentuk dan gaya yang berbeda-beda, tergantung dari asal-usul, budaya, dan era di mana karya seni musik tersebut dibuat. Karya seni musik dapat digunakan untuk tujuan hiburan, budaya, komunikasi, atau bahkan terapi. Karya seni musik juga dapat diekspresikan melalui berbagai bentuk, seperti lagu, syair, atau orkestra. Sebuah karya seni musik dapat menjadi refleksi dari perasaan, pikiran, atau bahkan narasi tertentu yang ingin disampaikan oleh penciptanya. Sebagai salah satu bentuk seni, karya seni musik juga memiliki nilai estetika dan dapat diapresiasi secara subjektif oleh para penggemarnya. Dalam beberapa konteks, karya seni musik juga dapat menjadi simbol dari sebuah budaya, identitas, atau bahkan gerakan sosial [3].

Untuk dapat menghasilkan karya seni musik bukanlah hal yang mudah. Diperlukan pengetahuan dan tentunya seni tersendiri dibidang musik. Namun, era teknologi saat ini memungkinkan banyak hal dapat dilakukan dengan mengandalkan perkembangan teknologi. Salah satu pengembangan teknologi saat ini, yaitu penerapan Deep Learning menggunakan metode GANs.

Penelitian terkait yang telah dilakukan oleh [4] membahas tentang menghasilkan karya seni gambar abstrak menggunakan metode Generative Adversarial Networks (GANs). Pada penelitian ini menggunakan 1369 data lukisan alam, pemandangan, maupun bunga. Metode GANs memiliki 2 alur dalam menghasilkan suatu karya seni gambar abstrak dari data yang dimasukkan, yaitu generator dan diskriminator. Penelitian yang dilakukan menggunakan 3 skenario pengujian dengan membagi iterasi, yaitu 10 iterasi, 100 iterasi, dan 1000 iterasi untuk proses pelatihan (training). Hasil pengujian yang telah dilakukan diperoleh dengan menggunakan 1000 iterasi dapat menghasilkan gambar abstrak dengan baik.

Penelitian yang dilakukan oleh [5] bertujuan untuk mengubah wajah manusia menjadi wayang orang dengan menambahkan makeup dan aksesoris menggunakan Generative Adversarial Network (GANs). Penelitian ini menggunakan dataset yang terdiri dari 1216 data latih dan 240 data uji. Pengujian kuantitatif pada penelitian ini digunakan untuk mengevaluasi kualitas citra yang dihasilkan dari generator. Hasil eksperimen menunjukkan bahwa UGATIT memiliki hasil yang lebih baik daripada DCLGAN berdasarkan nilai dari Inception Score, FID, dan KID, dengan skor sebagai berikut: 2.414, 0.924, dan 4.357.

Penelitian selanjutnya dilakukan oleh [6] membahas tentang penelitian menciptakan karakter anime secara otomatis menggunakan Generative Adversarial Networks (GANs) dengan menggunakan *library* PyTorch sebagai alat pelatihan. GANs mampu menghasilkan karakter anime baru dengan error yang cukup tinggi untuk data trainingnya, namun memiliki error yang lebih rendah dan skor yang lebih tinggi pada data testingnya. Data training terdiri dari 63.565 citra dan data uji terdiri dari 10.000 citra. Penelitian ini bertujuan untuk menciptakan karakter anime secara otomatis menggunakan teknologi Computer Vision, dan hasilnya menunjukkan GANs dapat menghasilkan karakter anime baru secara efektif.

Penelitian yang dilakukan oleh [7] membahas tentang implementasi Generative Adversarial Networks (GANs) pada sistem presensi berbasis deteksi wajah (SIDEWA). Tujuan dari penelitian ini adalah untuk meningkatkan efektivitas sistem presensi dengan mengimplementasikan GANs untuk memperbaiki kualitas citra deteksi wajah, yang kemudian akan diproses oleh sistem presensi SIDEWA. Penelitian ini menggunakan dataset wajah sebanyak 500 gambar dan dilakukan beberapa tahap pelatihan untuk menghasilkan model GANs yang terbaik. Hasil penelitian menunjukkan bahwa dengan menggunakan GANs pada sistem presensi SIDEWA, mampu meningkatkan akurasi deteksi wajah sebesar 10,2%.

Penelitian selanjutnya dilakukan oleh [8] membahas tentang penggunaan Deep Convolutional Generative Adversarial Network (DCGAN) untuk mentransformasikan citra malam menjadi citra siang. Penelitian ini bertujuan untuk mengatasi masalah ketidakjelasan citra malam yang menyulitkan dalam mendeteksi objek dan menghasilkan citra yang lebih jelas dan mudah dibaca. Penelitian menggunakan dataset yang terdiri dari 1000 pasang citra siang dan malam dan melatih model menggunakan arsitektur DCGAN. Hasil eksperimen menunjukkan bahwa model DCGAN yang dihasilkan berhasil mentransformasikan citra malam menjadi citra siang dengan baik, dilihat dari nilai PSNR, SSIM, dan NIQE yang lebih baik daripada model baseline.

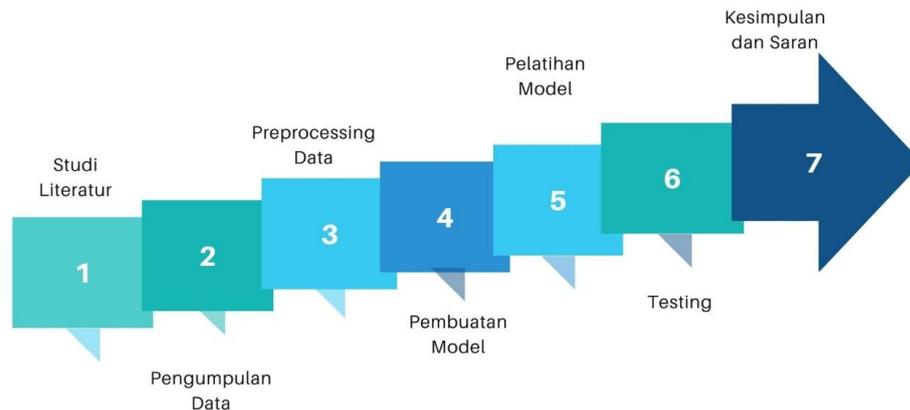
Penelitian yang dilakukan oleh [9] membahas tentang sintesis audio menggunakan metode Generative Adversarial Networks (GANs). Penelitian untuk mengolah musik menggunakan metode GANs merupakan tugas yang tidak mudah karena musik memiliki gelombang audio magnitudo dan frekuensi yang memerlukan kemampuan yang kompleks untuk memodelkannya. Penelitian ini menggunakan dataset Nsynth. Berdasarkan penelitian yang telah dilakukan, metode GANs dapat digunakan untuk membangun model generator dan menghasilkan audio baru dengan memberikan ketepatan baseline WaveNet.

Penelitian yang dilakukan oleh [10] membahas tentang pembuatan musik menggunakan transformer dan GANs. Melalui eksperimen, ditemukan bahwa pengolahan akord yang dilaporkan dalam penelitian ini tidak ideal, sebagaimana tercermin dalam proporsi akord yang tinggi dan nada sumbang yang berlebihan. Kamus karakter yang diproses dengan metode ini dibangun sesuai dengan nada dan akor dalam musik sesungguhnya. Untuk mengurangi ukuran kamus dan meningkatkan efek prediksi, frekuensi kata digunakan sebagai dasar untuk pemfilteran, dan beberapa nada dan akord dengan frekuensi rendah, nada ekstrim, dan akor kompleks dilindungi. Namun, jumlah label akor jauh lebih banyak daripada jumlah not individual. Meskipun jumlah nada tunggal dalam data pelatihan jauh lebih besar daripada frekuensi terjadinya akord, transformator tidak peka terhadap frekuensi dalam proses perhitungan model, terlepas dari probabilitas relatif nada tunggal dan akord. Dataset yang dipilih dalam penelitian ini didasarkan pada komposisi piano yang mengandung banyak akord dan not yang berbeda. Menggunakan akord mayor dan minor saja tidak cukup untuk mengekspresikan musik dalam kumpulan data ini. Oleh karena itu, preprocessing dataset musik dan penggunaan algoritma untuk meringkas aturan kemunculan nada secara efektif harus menjadi tujuan utama pembuatan musik di masa mendatang.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode Generative Adversarial Networks (GANs). Metode GANs melibatkan 2 komponen utama yang saling beradu untuk menghasilkan model yang sesuai dengan kemampuan komponen

masing – masing. Untuk sumber data dihasilkan dari dataset audio yang nantinya akan dikumpulkan untuk proses melakukan training data. Sedangkan untuk teknik pengumpulan data dapat dilakukan dengan mengumpulkan dataset audio dari genre musik pop, kemudian memproses dataset audio tersebut ketipe file midi. Adapun alur penelitian yang dilakukan dapat dilihat pada Gambar 1



Gambar 1. Alur Penelitian

a) Studi Literatur

Studi literatur dilakukan untuk mengidentifikasi masalah. Mengumpulkan materi dan referensi terkait dari berbagai sumber, seperti jurnal penelitian untuk menentukan solusi dari permasalahan yang ada.

b) Pengumpulan Data

Langkah selanjutnya adalah melakukan pengumpulan data. Data yang dikumpulkan berupa audio musik dengan genre musik pop.

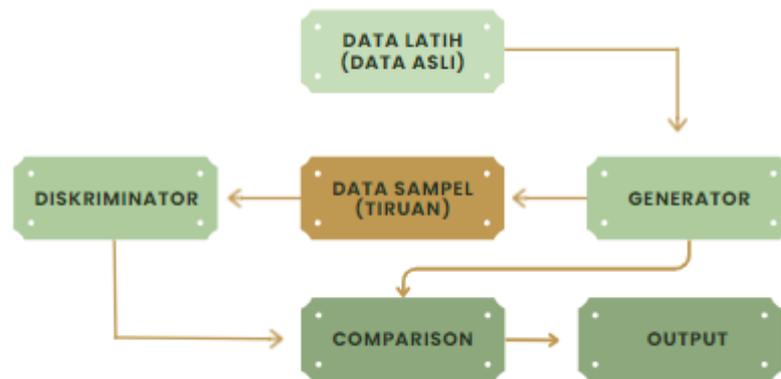
c) Preprocessing Data

Preprocessing data diperlukan untuk menghasilkan data yang akurat dengan melakukan normalisasi data, penghapusan data yang tidak relevan, dan menyesuaikan format data yang dapat dibaca oleh metode Generative Adversarial Networks (GANs). Pada kasus penggunaan metode GANs untuk audio berektensi mp3 harus diubah menjadi ekstensi file.mid.

d) Pembuatan Model

Terdapat 2 model pada metode GANs, yaitu generator dan diskriminator. Input yang diberikan untuk model Generator adalah data latih berupa audio musik genre pop, kemudian Generator akan menghasilkan data sampel berupa audio tiruan yang menyerupai audio asli. Setelah Generator menghasilkan data sampel berupa audio tiruan, selanjutnya diskriminator bekerja dengan melakukan comparison, yaitu membandingkan antara data yang dihasilkan oleh generator dengan data latih yang ada hingga

menghasilkan output yang diinginkan. Setelah generator dan diskriminator saling beradu, langkah inti dari penelitian ini, yaitu menghasilkan audio baru. Diagram alir deep learning menggunakan metode GANs dapat dilihat pada Gambar 2.



Gambar 2. Diagram Alir Penelitian menggunakan Metode GANs.

e) Pelatihan Model

Pelatihan model diperlukan agar pertarungan antara Generator dan Diskriminator dapat terus meningkatkan kualitas dan menghasilkan keaslian audio yang lebih baik. Hasil penilaian diskriminator akan digunakan untuk memperbaiki generator agar menghasilkan audio yang lebih baik dan lebih mirip dengan audio asli. Untuk menghasilkan output yang diinginkan harus menyesuaikan dengan parameter yang digunakan, seperti *batch size*, *latent dim*, *output shape*, dan *epoch* dalam melakukan pelatihan model.

f) Testing

Setelah pembuatan model dan pelatihan model telah dilakukan, maka langkah selanjutnya dapat dilakukan testing atau pengujian model untuk melihat keakuratan model dalam menghasilkan audio baru. Output yang dihasilkan dari penelitian ini adalah berupa model generatif yang dapat digunakan untuk menghasilkan audio baru untuk membangun musik *genre* pop yang bervariasi.

g) Kesimpulan

Langkah terakhir dari penelitian yang telah dilakukan adalah menyimpulkan hasil penelitian dan menjawab perumusan masalah yang ingin diselesaikan dari penelitian yang dilakukan.

3. HASIL DAN PEMBAHASAN

Penelitian yang dilakukan menggunakan Google Colab dengan menggunakan pemrograman Python. Google Colab merupakan salah satu layanan *cloud* yang disediakan oleh Google untuk melakukan pengembangan pemrograman Python. Langkah pertama melakukan penelitian deep learning adalah menyiapkan dataset. Pada penelitian ini menggunakan dataset musik

genre pop yang dapat diakses secara gratis pada situs FMA (Free Music Archive) yang memiliki berbagai variasi musik genre pop. Setelah dataset siap, musik yang tersedia dengan format mp3 tidak dapat dibaca oleh metode GANs sehingga perlu melakukan konversi file. Salah satu file musik yang dapat dibaca oleh metode GANs adalah file dengan format .mid, sehingga perlu melakukan konversi file musik format .mp3 ke format .mid. Konversi dilakukan dengan menggunakan Google Colab. Setelah berhasil melakukan konversi, dataset dapat digunakan untuk diolah menggunakan metode GANs. Untuk menggunakan Google Colab pada penelitian ini langkah awal yang harus dilakukan adalah install library dan import library. Terdapat beberapa library yang diperlukan untuk dapat menjalankan metode GANs menghasilkan musik genre pop baru

3.1. Konversi File Musik Format .mp3 ke File Midi

Proses awal melakukan konversi pada file musik format .mp3 ke format .mid menggunakan library mido pydub, sehingga perlu melakukan instalasi pada Google Colab dengan menggunakan kode berikut:

```
!pip install mido pydub
```

Selanjutnya melakukan import pustaka yang dibutuhkan untuk kelangsungan proses konversi dengan menggunakan kode berikut:

```
from pydub import AudioSegment  
from pydub.playback import play  
from mido import Message, MidiFile, MidiTrack  
import os
```

Langkah selanjutnya adalah fungsi untuk melakukan konversi dari format .mp3 ke format .mid menggunakan kode berikut:

```
def mp3_to_midi(mp3_path, midi_path):  
    audio = AudioSegment.from_mp3(mp3_path)  
    notes = []  
    for i in range(0, len(audio), 100):  
        segment = audio[i:i+100]  
        dbfs = segment.dBFS  
        if dbfs > -30:  
            note = 60  
        elif dbfs > -40:  
            note = 62  
        else:  
            note = 64  
        notes.append(note)  
  
    mid = MidiFile()  
    track = MidiTrack()
```



```
mid.tracks.append(track)

for note in notes:
    track.append(Message('note_on', note=note, velocity=64, time=100))
    track.append(Message('note_off', note=note, velocity=64, time=100))

mid.save(midi_path)
```

Apabila ketika kode dijalankan tidak ada informasi error maka dapat dilanjutkan ke proses berikutnya. Pada penelitian ini setiap dataset maupun output akan terakses di google drive karena google colab dapat dengan mudah dihubungkan dengan google drive menggunakan kode berikut:

```
# Langkah 1: Hubungkan Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
# lokasi dataset di google drive
mp3_folder_path = '/content/drive/My Drive/fma_pop_musik/'
```

Setelah berhasil menghubungkan Google Colab dengan Google Drive, masukkan lokasi dataset yang akan digunakan untuk konversi.

```
import os

# Pastikan folder ada
if os.path.exists(mp3_folder_path):
    # Coba lihat isi dari folder
    mp3_files = os.listdir(mp3_folder_path)
    print(mp3_files)
else:
    print("Folder tidak ditemukan.")
```

Kode diatas berfungsi untuk memastikan bahwa lokasi dataset sudah sesuai dan memiliki dataset yang diperlukan, apabila dataset tidak ditemukan maka kemungkinan terjadi kesalahan ketika menghubungkan Google Colab dengan Google Drive. Kemudian tentukan lokasi file konversi menggunakan kode berikut:

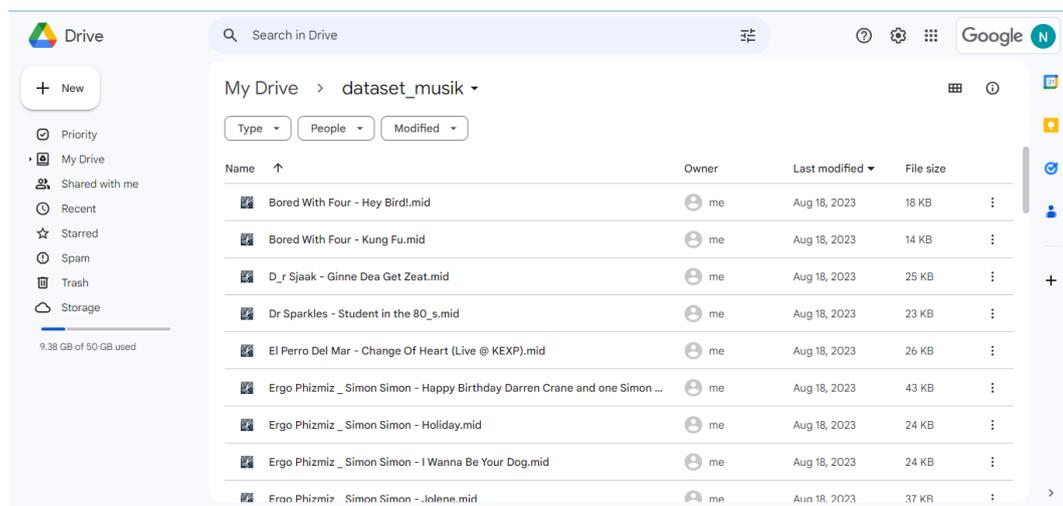
```
# lokasi hasil konversi tersimpan
midi_folder_path = '/content/drive/My Drive/dataset_musik/'

os.makedirs(midi_folder_path, exist_ok=True)
```



```
# proses konversi format file .mp3 ke format .mid
for mp3_filename in os.listdir(mp3_folder_path):
    if mp3_filename.endswith('.mp3'):
        mp3_path = os.path.join(mp3_folder_path, mp3_filename)
        midi_filename = mp3_filename[:-4] + '.mid'
        midi_path = os.path.join(midi_folder_path, midi_filename)
        mp3_to_midi(mp3_path, midi_path)
```

Dataset yang berhasil dikonversi akan tersimpan di Google Drive dengan nama folder `dataset_musik`.



Gambar 3. Dataset Musik yang telah dikonversi.

3.2. Menghasilkan Musik Baru menggunakan Metode GANs (Generative Adversarial Networks)

Setelah dataset siap maka lanjut pada proses menghasilkan musik baru. Hubungkan terlebih dahulu Google Colab dengan Google Drive menggunakan kode berikut:

```
# Langkah 1: Hubungkan Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

Pada penelitian ini membutuhkan beberapa library yang mendukung metode GANs agar dapat bekerja dengan maksimal.

```
# Langkah 2: Instalasi pustaka
!pip install mido
!pip install tensorflow
!pip install pretty_midi
!pip install midiutil
!pip install pydub
```

Library yang digunakan memiliki fungsi untuk menghasilkan musik baru menggunakan format file midi, seperti mido, pretty_midi, dan midiutil, sedangkan tensorflow merupakan library yang mendukung untuk metode GANs dan pydub untuk mendukung audio. Selanjutnya melakukan import pustaka yang diperlukan.

```
# Langkah 3: Impor pustaka yang diperlukan
```

```
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, Model
from mido import MidiFile, MidiTrack, Message
import os
import midiutil
import shutil
from pydub import AudioSegment
import pretty_midi
```

Pustaka yang diperlukan harus dipastikan ter-import dengan baik agar kode-kode dapat dijalankan dengan lancar dan sesuai. Langkah selanjutnya adalah melakukan praproses data.

```
# Langkah 4: Tentukan lokasi file MIDI dan muat data
```

```
midi_folder = "/content/drive/MyDrive/dataset_musik/"
midi_files = os.listdir(midi_folder)

midi_data = []

for file_name in midi_files:
    if file_name.endswith(".mid"):
        midi_path = os.path.join(midi_folder, file_name)
        midi_data.append(pretty_midi.PrettyMIDI(midi_path))
```

```
# Periksa apakah ada file MIDI yang dimuat
```

```
if len(midi_data) == 0:
    print("Tidak ada file MIDI yang dimuat.")
else:
```

```
# Langkah 5: Persiapkan dataset (prapemrosesan)
```

```
def preprocess_midi(midi_data, output_shape):
```

```
    # Inisialisasi matriks untuk menyimpan representasi MIDI
```

```
    midi_representation = np.zeros((len(midi_data), output_shape))
```

```
    for i, midi in enumerate(midi_data):
```

```
        for note in midi.instruments[0].notes: # Hanya mengambil catatan dari instrumen pertama
```



```
pitch = note.pitch
start_time = note.start
end_time = note.end
duration = end_time - start_time
velocity = note.velocity

# Hitung indeks tempat catatan dimulai dan berakhir
start_index = int(start_time / (480 / output_shape))
end_index = int(end_time / (480 / output_shape))

# Masukkan informasi catatan ke dalam matriks
midi_representation[i, start_index:end_index] = pitch

return midi_representation

# Panggil fungsi prapemrosesan dataset
output_shape = 128 # Sesuaikan dengan panjang atau durasi MIDI yang diinginkan
preprocessed_data = preprocess_midi(midi_data, output_shape)

# Contoh: memastikan data yang dihasilkan
print(preprocessed_data[0]) # Melihat data dari file MIDI pertama
```

Kode diatas akan merupakan praproses atau kegiatan menginisiasi data dan proses menyiapkan dataset untuk dapat diolah oleh metode GANs. Tahap penting dari penelitian ini, yaitu pembentukan model GANs menggunakan Generator dan Diskriminator. Kode pembangun model Generator dan Diskriminator adalah sebagai berikut:

```
# Langkah 6: Pilih atau persiapkan model GAN
def build_generator(latent_dim, output_shape):
    model = tf.keras.Sequential([
        layers.InputLayer(input_shape=(latent_dim,)),
        layers.Dense(128),
        layers.LeakyReLU(alpha=0.2),
        layers.Dense(256),
        layers.LeakyReLU(alpha=0.2),
        layers.Dense(output_shape, activation='sigmoid')
    ])
    return model

def build_discriminator(input_shape):
    model = tf.keras.Sequential([
        layers.InputLayer(input_shape=input_shape),
        layers.Dense(256),
```



```
layers.LeakyReLU(alpha=0.2),
layers.Dense(128),
layers.LeakyReLU(alpha=0.2),
layers.Dense(1, activation='sigmoid')
])
return model

latent_dim = 50
output_shape = 128
input_shape = output_shape

generator = build_generator(latent_dim, output_shape)
discriminator = build_discriminator(input_shape)

discriminator.compile(optimizer='adam', loss='binary_crossentropy')
discriminator.trainable = False

gan_input = layers.Input(shape=(latent_dim,))
x = generator(gan_input)
gan_output = discriminator(x)
gan = tf.keras.models.Model(gan_input, gan_output)

gan.compile(optimizer='adam', loss='binary_crossentropy')
```

Model Generator dan Diskriminator harus menyesuaikan latent dim dan output shape yang diinginkan agar model dapat dibangun sesuai dengan yang diinginkan pada penelitian ini. Penelitian ini menggunakan latent dim 50 dan batch size 64, serta menggunakan epoch 10.000 dan output shape dengan nilai 128. Untuk kode pelatihan model GANs adalah sebagai berikut:

```
# Langkah 7: Bangun model GAN
batch_size = 64
epochs = 10000

for epoch in range(epochs):
    noise = np.random.randn(batch_size, latent_dim)
    generated_music = generator.predict(noise)

    real_music = preprocessed_data[np.random.randint(0, preprocessed_data.shape[0],
batch_size)]
    x_combined = np.concatenate([real_music, generated_music])
    y_combined = np.concatenate([np.ones((batch_size, 1)), np.zeros((batch_size, 1))])

    d_loss = discriminator.train_on_batch(x_combined, y_combined)
```



```
noise = np.random.randn(batch_size, latent_dim)
y_mislabeled = np.ones((batch_size, 1))

a_loss = gan.train_on_batch(noise, y_mislabeled)

if (epoch + 1) % 100 == 0:
    print(f"Epoch: {epoch + 1}, D Loss: {d_loss}, G Loss: {a_loss}")
```

Pelatihan model GANs dari kode diatas akan menampilkan iterasi proses pelatihan yang berlangsung sebanyak 10.000 iterasi dan menampilkan loss D dan loss G. Terlihat nilai loss D dan loss G pada epoch ke 10.000, yaitu:

Epoch: 10000, D Loss: 1.4088327588979155e-06, G Loss: 12.790088653564453

Berdasarkan nilai loss D diatas dapat dijelaskan bahwa model diskriminator dapat membedakan data asli dan data palsu yang dihasilkan oleh generator, sedangkan nilai loss G menunjukkan seberapa baik model dalam menghasilkan musik baru. Hal ini dapat dijelaskan berdasarkan kualitas musik yang didapatkan apakah sudah sesuai atau masih perlu pelatihan model yang lebih lagi. Untuk tiap nilai pada loss D dan loss G bergantung pada kebutuhan penelitian yang dilakukan. Langkah selanjutnya adalah menghasilkan musik baru.

```
# Langkah 8: Hasilkan musik baru dan simpan sebagai file MIDI
random_latent_vectors = np.random.randn(1, latent_dim) # Membuat vektor laten acak
generated_music = generator.predict(random_latent_vectors) # Menghasilkan musik baru

# Normalisasi nilai-nilai MIDI (antara 0 dan 127)
generated_music = (generated_music * 127).astype(int)

# Buat objek PrettyMIDI untuk menyimpan hasil musik
generated_midi = pretty_midi.PrettyMIDI()

# Tambahkan instrumen (misalnya, piano)
piano_program = pretty_midi.instrument_name_to_program('Acoustic Grand Piano')
piano = pretty_midi.Instrument(program=piano_program)

# Tambahkan catatan berdasarkan hasil dari generator
start_time = 0
end_time = 0.5 # Tentukan durasi catatan

for note_value in generated_music[0]:
    note_instance = pretty_midi.Note(velocity=100, pitch=note_value, start=start_time,
end=end_time)
    piano.notes.append(note_instance)
```

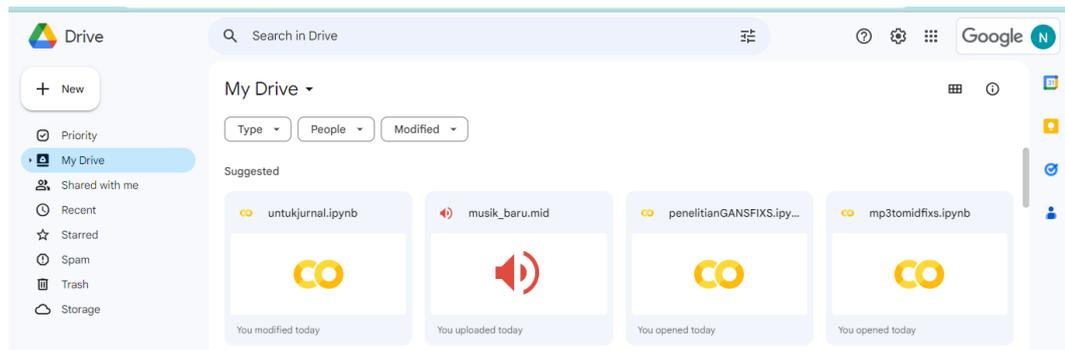


```
start_time = end_time
end_time += 0.5 # Pindahkan ke catatan berikutnya

# Tambahkan instrumen ke PrettyMIDI
generated_midi.instruments.append(piano)

# Simpan hasil musik sebagai file MIDI
output_midi_path = '/content/drive/MyDrive/musik_baru.mid'
generated_midi.write(output_midi_path)
```

Kode diatas merupakan kode untuk menghasilkan musik baru berdasarkan pada proses yang telah dilalui dari menyiapkan dataset, membangun model, hingga proses pelatihan model. Musik baru yang dihasilkan tersimpan pada google drive dengan nama musik_baru.



Gambar 4. Musik Baru.

Musik baru yang dihasilkan memberikan hasil audio yang cukup bervariasi berdasarkan not-not baru dan dapat diadopsi bagi para musisi untuk menjadi referensi dalam mencari ide menghasilkan musik baru berbasis pada variasi not yang lebih dominan pada instrumen piano akustik.

4. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa metode GANs (Generative Adversarial Networks) dapat digunakan untuk menghasilkan musik baru. Hasil musik baru yang diperoleh pada penelitian ini memberikan audio dengan format file .mid yang cukup baik berdasarkan pada 50 dataset musik file .mid. Keberhasilan pembelajaran mendalam (deep learning) dipengaruhi oleh beberapa faktor, yaitu dataset, arsitektur model GANs, dan mode pelatihan. Penelitian menghasilkan musik baru ini merupakan penelitian yang cukup kompleks sehingga diperlukan pemahaman yang lebih agar dapat menghasilkan musik baru yang lebih baik lagi dan kunci pembelajaran mendalam (deep learning) adalah melakukan eksperimen dan menyiapkan dataset yang bervariasi.

DAFTAR PUSTAKA

- [1] A. Gupta, "Introduction to deep learning: Part 1," *Chem. Eng. Prog.*, vol. 114, no. 6, pp. 22–29, 2018.
- [2] H. Yaya and T. Wahyono, *Dasar-Dasar Deep Learning dan Implementasinya*. Bandung: Gava Media, 2021.
- [3] Sayidiman and I. Andi, *Seni Musik*. Jakarta: PT Grafindo Media Pratama, 2010.
- [4] E. S. Purba, T. Informatika, F. T. Informasi, U. Kristen, and S. Wacana, "Implementation of Generative Adversarial Networks for Penerapan Generative Adversarial Networks Dalam Pembuatan," vol. 3, no. 3, pp. 707–715, 2022.
- [5] C. Nurdenara and W. Fawwaz, "Translasi Citra Antara Manusia Dan Wayang Orang Menggunakan Generative Adversarial Network," *eProceedings ...*, vol. 8, no. 5, pp. 11307–11319, 2021, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15617>.
- [6] V. V. Pramansah, D. I. Mulyana, T. Silfia, and R. T. Jaya, "Penciptaan Karakter Anime Otomatis Dengan Menggunakan Generative Adversarial Networks," *J. Tek. Elektro dan Komputasi*, vol. 4, no. 1, pp. 21–29, 2022.
- [7] T. Suprpti, D. A. Kurnia, D. Anggara, R. D. Rian, and A. Setiawan, "Implementasi Model Algoritma Generative Adversarial Network (Gan) Pada Sistem Presensi Berbasis Deteksi Wajah (SIDEWA)," *Temat. J. Teknol. Inf. Komun.*, vol. 9, no. 2, pp. 231–236, 2023, doi: 10.38204/tematik.v9i2.1048.
- [8] P. Y. Santosa, E. Rachmawati, T. Agung, and B. Wirayuda, "Translasi Citra Malam Menjadi Siang Menggunakan Deep Convolutional Generative Adversarial Network," *eProceedings Eng.*, vol. 8, no. 1, 2021.
- [9] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial Neural Audio Synthesis," pp. 1–17, 2019.
- [10] J. Min, Z. Liu, L. Wang, D. Li, M. Zhang, and Y. Huang, "Music Generation System for Adversarial Training Based on Deep Learning," *Processes*, vol. 10, no. 12, 2022, doi: 10.3390/pr10122515.