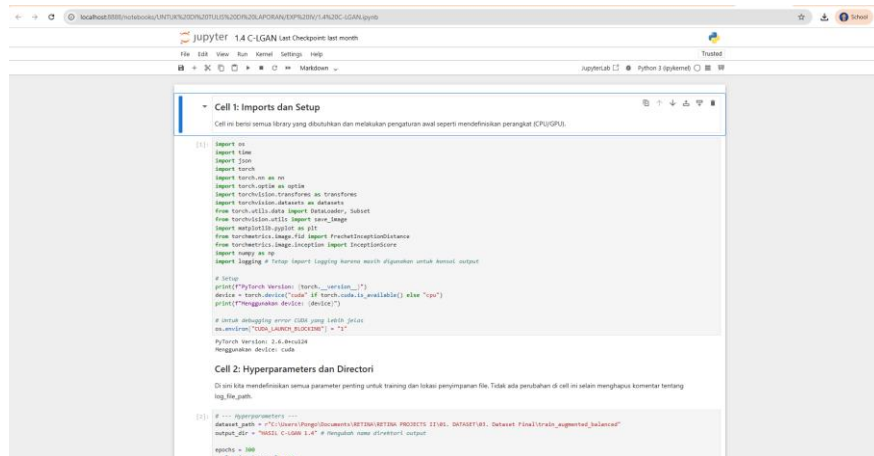


LAMPIRAN

Lampiran 1. Tangkapan Layar Kode GANs



```
[1]: import os
import time
import json
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision.transforms as transforms
import torchvision.datasets as datasets
from torch.utils.data import DataLoader, Subset
from torchvision.utils import save_image
import matplotlib.pyplot as plt
from torchvision.models.inception import inception_v3
import numpy as np
import logging # Untuk logging karena masih digunakan untuk Annot output

# Setup
print(f"PyTorch Version: {torch.__version__}")
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Menggunakan device: {device}")

# Untuk menangkap error jika yang salah terjadi
os.environ["CUDA_LAUNCH_BLOCKING"] = "1"

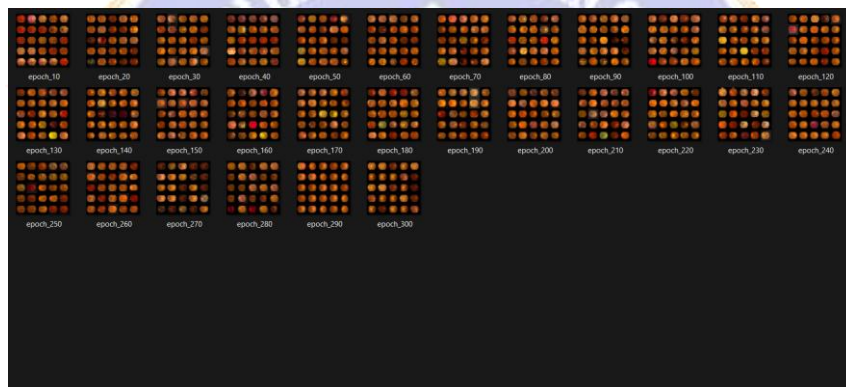
PyTorch Version: 2.0.0
Menggunakan device: cuda

Cell 2: Hyperparameters dan Directori
Di sini kita mendefinisikan semua parameter penting untuk training dan lokasi penyimpanan file. Total ada perubahan di cell ini selain menghapus komentar tentang log file path.

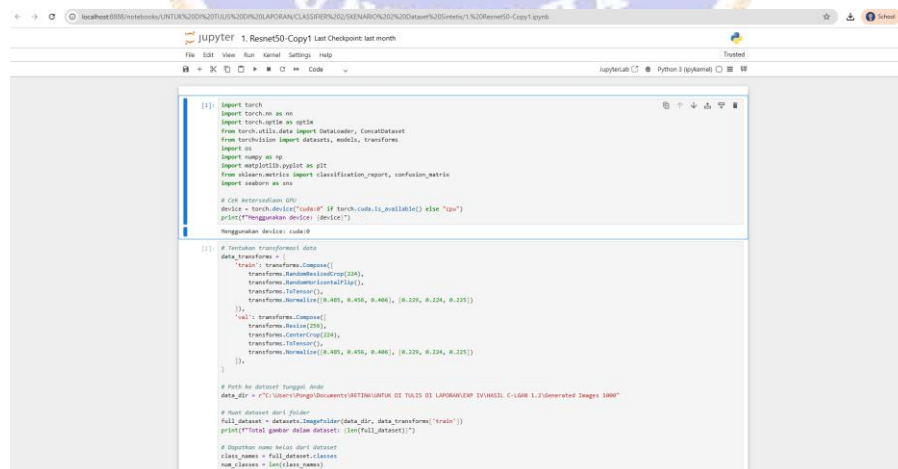
[2]: # --- Hyperparameters ---
dataset_path = r"C:\Users\Prang\Documents\SETI\SETI\PROJECTS\11\SETI_DATASET\01_Dataset\Final\train_augmented_balanced"
output_dir = "SETI_0-1000_2-4" # Mengubah name directory output

epochs = 100
validation_interval = 10
```

Lampiran 2. Citra Hasil Generasi



Lampiran 3 Tangkapan Layar Kode Klasifikasi



```
[1]: import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, ConcatDataset
from torchvision import datasets, models, transforms
import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
import random as rnd

# Cek ketersediaan GPU
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(f"Menggunakan device: {device}")

# Mengkonversi device ke cuda:0

[2]: # Tentukan transformasi data
data_transform = [
    transforms.Compose([
        transforms.RandomHorizontalFlip(0.5),
        transforms.RandomVerticalFlip(0.5),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ]),
    transforms.Compose([
        transforms.Resize(224),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ])
]

# Path ke dataset yang akan di-load
data_dir = r"C:\Users\Prang\Documents\SETI\UNTUK DI TULIS DI LAPORAN\EXP\UNTUK_0-1000_2-4\Generated Images 1000"

# Buat dataset dari folder
full_dataset = datasets.ImageFolder(data_dir, data_transform=[transform])
print(f"Total gambar dalam dataset: {len(full_dataset)}")

# Definisikan nama kelas dari dataset
class_names = full_dataset.classes
map_classes = {int(i):i for i in class_names}
print(f"Nama kelas: {class_names}")
```