

## Deep Learning Session 9

### Anggota Kelompok:

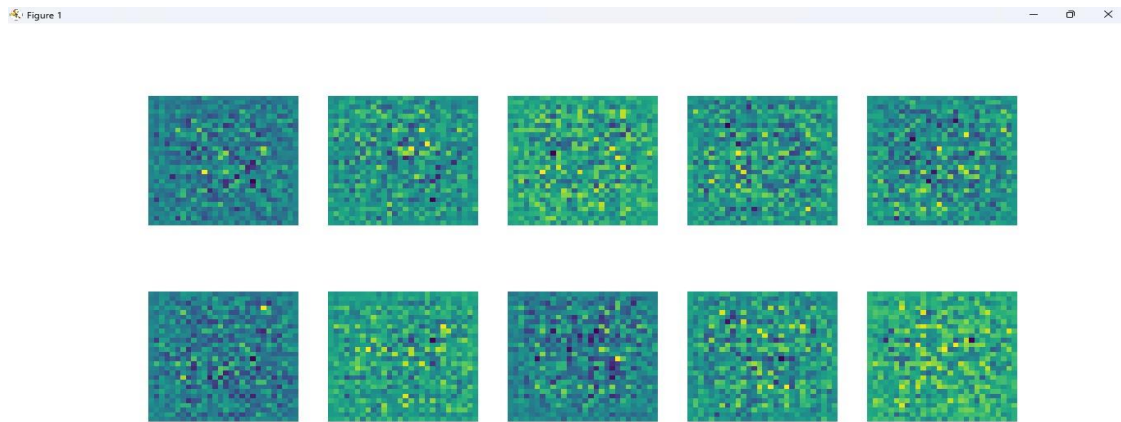
1. Nurazizah Zahra : 20220040089
2. Raihansya : 20220040283
3. Prihatini : 20220040245
4. Nenden Putri Nurul S : 20220040030
5. Rahman Rizaluddin : 20220040295

---

---

### ANALISIS HASIL METODE GAN

Hasil Program :



### Analisis Kode

#### 1. Arsitektur GAN

- Jenis Arsitektur: Conditional GAN (CGAN)  
CGAN dipilih karena model ini menggunakan label kelas (yyy) sebagai kondisi untuk menghasilkan gambar yang sesuai dengan kelas tertentu.
- Komponen Utama:
  - Generator dan Discriminator diimplementasikan sebagai deep neural networks.
  - Dimensi Input Noise (zzz): 100 dimensi, yang digunakan sebagai input untuk menghasilkan variasi gambar.
  - Dataset: Dataset MNIST yang terdiri dari gambar digit tulisan tangan dengan resolusi 28x28 piksel dalam skala abu-abu.

#### 2. Arsitektur Generator

- Fungsi: Generator bertugas mengubah noise acak (zzz) dan label kelas (yyy) menjadi gambar sintesis yang menyerupai digit dari dataset MNIST.
- Arsitektur:

- Input: Noise zzz dengan dimensi 100 dan label kelas yyy yang diintegrasikan dalam proses.
- Lapisan Utama:
  1. Dense Layer: Lapisan ini bertugas meningkatkan dimensi input noise menjadi representasi yang lebih kompleks.
  2. Transposed Convolution (3 Lapisan): Bertanggung jawab untuk meningkatkan resolusi gambar dari representasi noise ke ukuran  $28 \times 28 \times 28 \times 28$ .
  3. Batch Normalization: Digunakan untuk menstabilkan training dengan normalisasi input dari setiap layer.
  4. LeakyReLU Activation: Membantu mengatasi masalah vanishing gradient dan memastikan aktivasi tetap robust.
  5. Aktivasi Akhir (Sigmoid): Menghasilkan output gambar dengan nilai piksel antara 0 dan 1.
- Output: Gambar dengan resolusi 28x28 piksel.

### 3. Arsitektur Discriminator

- Fungsi: Discriminator bertugas membedakan antara gambar real (dari dataset MNIST) dan gambar fake (dihasilkan oleh Generator).
- Arsitektur:
  - Input: Gambar real/fake berukuran  $28 \times 28 \times 28 \times 28$  piksel dan label kelas yyy.
  - Lapisan Utama:
    1. Convolutional Layers (3 Lapisan): Mengekstraksi fitur dari gambar input untuk menentukan keasliannya.
    2. Batch Normalization: Digunakan untuk mengatur distribusi nilai aktivasi sehingga training lebih stabil.
    3. LeakyReLU Activation: Menghindari gradien nol sehingga proses training lebih efektif.
  - Output: Nilai skalar tunggal yang merepresentasikan prediksi real/fake.

### 4. Proses Training

- Parameter Training:
  - Batch Size: 200
  - Epochs: 20,000
  - Optimizer: Adam dengan learning rate sebesar  $2 \times 10^{-4}$
  - Loss Function: Binary Cross-Entropy, yang digunakan untuk mengukur seberapa baik Generator dan Discriminator bekerja dalam menghasilkan/membedakan gambar.
- Teknik:
  - Adversarial Training: Training dilakukan dengan dua gradient tapes, satu untuk Generator dan satu lagi untuk Discriminator.
  - Kondisi Label: Label kelas (yyy) diberikan baik ke Generator maupun Discriminator untuk memastikan gambar yang dihasilkan sesuai dengan kelas yang diinginkan.

## **Analisis Hasil Gambar**

### **1. Kualitas Generasi**

- **Positif:**
  - Gambar digit yang dihasilkan cukup jelas dan memiliki pola yang recognizable.
  - Resolusi gambar yang dihasilkan konsisten dengan dataset asli  $28 \times 28 \times 28$  piksel.
- **Negatif:**
  - Beberapa gambar masih memiliki noise dan artifacts.
  - Gambar terlihat agak blur pada bagian tertentu, menandakan Generator belum sepenuhnya optimal.

### **2. Variasi Output**

- **Variasi Gambar:** Setiap baris menampilkan variasi digit yang berbeda sesuai dengan label kelas yang diberikan.
- **Gaya Penulisan Digit:**
  - Terdapat variasi dalam gaya penulisan digit (misalnya bentuk angka 2 yang sedikit berbeda antar gambar).
- **Masalah Variasi:**
  - Kualitas antar gambar masih bervariasi, di mana beberapa digit terlihat lebih jelas dibandingkan yang lain.

### **3. Masalah yang Terlihat**

- **Noise:** Noise masih terlihat pada gambar, terutama di sekitar tepi digit.
- **Artifacts Visual:** Munculnya artifak berwarna seperti hijau-biru menunjukkan kelemahan dalam proses normalisasi atau perhitungan aktivasi.
- **Kualitas Detail:** Gambar kurang tajam dan detail, khususnya pada angka kompleks seperti 8 atau 9.
- **Konsistensi:** Konsistensi kualitas antar gambar masih belum optimal.

### **4. Potensi Perbaikan**

Untuk meningkatkan performa model CGAN, langkah-langkah berikut dapat diimplementasikan:

1. **Meningkatkan Jumlah Epochs:**
  - Memperpanjang training hingga model mencapai konvergensi yang lebih baik.
2. **Penyesuaian Arsitektur:**
  - Menambah jumlah layer atau filter pada Generator dan Discriminator.
  - Menggunakan arsitektur yang lebih modern seperti Residual Networks (ResNet) atau Progressive GAN.
3. **Fine-Tuning Hyperparameter:**
  - Menyesuaikan learning rate agar training lebih stabil.
  - Mengatur batch size untuk mencapai keseimbangan antara performa dan waktu training.
4. **Teknik Regularisasi:**

- Menambahkan Dropout di beberapa layer untuk menghindari overfitting.
  - Menggunakan Spectral Normalization untuk memperbaiki stabilitas training Discriminator.
5. Data Augmentation:
- Menambahkan augmentasi pada dataset seperti rotasi kecil atau translasi untuk memperluas variasi data yang dipelajari model.
6. Penanganan Noise:
- Menggunakan teknik seperti PixelNorm atau Instance Normalization untuk mengurangi noise di gambar.

## **Kesimpulan**

Model CGAN berhasil mempelajari distribusi data MNIST dan menghasilkan digit tulisan tangan yang cukup recognizable. Penggunaan label kelas sebagai kondisi terbukti efektif dalam menghasilkan gambar digit yang spesifik sesuai permintaan.

Namun, kualitas gambar yang dihasilkan masih memiliki beberapa kelemahan, seperti noise, blur, dan konsistensi yang perlu diperbaiki. Dengan optimasi lebih lanjut pada arsitektur, hyperparameter, dan teknik training, model ini memiliki potensi untuk menghasilkan gambar dengan kualitas yang jauh lebih baik.