

# Laporan Final Project Kecerdasan Buatan (Lanjut)

*Implementasi Neural Style Transfer untuk Transformasi Gaya Artistik pada Citra Digital Menggunakan VGG-19.*



Kelompok 10

Anggota Kelompok:

23.11.5571 | Aulia Resty Nur Aini

23.11.5603 | Nabillah Alimuddin

23.11.5604 | M. Dian Fauzi

23.11.5620 | Ledyvia Audiz Coranov

## 1. Latar Belakang

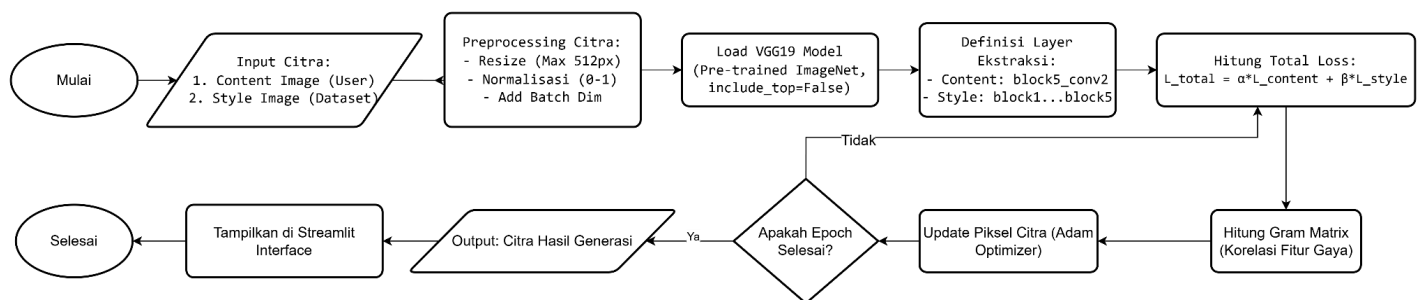
Perkembangan teknologi *Artificial Intelligence* (AI), khususnya dalam domain *Generative Modelling*, telah menghadirkan paradigma baru dalam penciptaan karya seni digital. Salah satu tantangan utama dalam industri kreatif adalah menggabungkan konten visual realistis (seperti fotografi) dengan estetika artistik yang kompleks (seperti gaya lukisan impresionisme) tanpa memerlukan proses pelukisan manual yang memakan waktu lama. Metode konvensional dalam pengolahan citra digital seringkali hanya memanipulasi warna atau filter sederhana, namun gagal menangkap esensi "gaya" atau tekstur dari sebuah karya seni.

Pada tahun 2016, Gatys et al. memperkenalkan metode *Neural Style Transfer* (NST) yang memanfaatkan *Convolutional Neural Networks* (CNN) untuk memisahkan dan merekomposisi konten dan gaya dari dua citra yang berbeda [1]. Pendekatan ini memungkinkan komputer untuk "melukis ulang" sebuah foto dengan gaya seniman ternama. Penelitian serupa juga telah dikembangkan di Indonesia, seperti yang dilakukan oleh Yulian dkk. (2023) yang menerapkan NST dan *Convolutional Autoencoder* untuk pelestarian motif Batik [5]. Mengacu pada penelitian tersebut, penggunaan arsitektur *Deep Learning* terbukti efektif dalam merepresentasikan fitur visual artistik yang kompleks.

Dalam proyek akhir ini, kami mengembangkan sistem bernama **"The Art Museum Generator"**. Sistem ini mengimplementasikan arsitektur VGG-19 yang telah dilatih sebelumnya (*pre-trained*) pada dataset ImageNet. Tujuannya adalah untuk mendemokratisasi akses terhadap penciptaan seni digital, memungkinkan pengguna awam untuk mentransformasi foto pribadi mereka menjadi karya seni dengan gaya pelukis maestro dunia (seperti Van Gogh atau Monet) melalui antarmuka berbasis web yang interaktif.

## 2. Metode:

### 1) Alur Pengerjaan (System Flowchart)



Tahapan pengerjaan proyek ini dimulai dari pengumpulan data hingga *deployment* aplikasi, dengan rincian sebagai berikut:

**a. Pengumpulan Dataset:** Mengurasi citra gaya (*style images*) dari repositori seni publik (Kaggle: Best Artworks of All Time).

**b. Preprocessing Citra:**

- i. *Resizing*: Mengubah dimensi citra konten dan gaya menjadi maksimum 512 piksel (sisi terpanjang) untuk menjaga efisiensi komputasi GPU.
- ii. *Normalization*: Mengonversi nilai piksel ke rentang [0, 1] dan menambahkan dimensi *batch* agar sesuai dengan input tensor VGG-19.

**c. Pembangunan Model (VGG19):** Memuat model VGG-19 tanpa *top layer* (klasifikasi), dan mendefinisikan *intermediate layers* untuk ekstraksi fitur:

- i. *Content Layer*: block5\_conv2
- ii. *Style Layers*: block1\_conv1, block2\_conv1, block3\_conv1, block4\_conv1, block5\_conv1.

**d. Proses Optimasi (Training Loop):**

- i. Menghitung *Content Loss* (perbedaan fitur konten) dan *Style Loss* (perbedaan korelasi Gram Matrix).
- ii. Menggunakan *Adam Optimizer* dengan *learning rate* 0.02 untuk memperbarui piksel pada citra target (bukan bobot model).

**e. Deployment:**

Mengintegrasikan model ke dalam antarmuka pengguna menggunakan *Streamlit* dan *Cloudflare Tunnel* agar dapat diakses secara publik.

## 2) Algoritma dan Arsitektur Model

Algoritma utama yang digunakan adalah *Neural Style Transfer* berbasis optimasi. Kami menggunakan arsitektur **VGG19** (*Visual Geometry Group 19-layer*), sebuah jaringan saraf konvolusional yang sangat dalam [2].

- Mekanisme Loss Function: Total Loss dihitung dari penjumlahan terbobot antara Content Loss dan Style Loss:

$$L_{total} = \alpha L_{content} + \beta L_{style}$$

- Dimana  $\alpha$  (bobot konten) diset sebesar  $10^4$  dan  $\beta$  (bobot gaya) diset sebesar  $10^{-2}$ .

- **Gram Matrix:** Untuk merepresentasikan gaya, kami menghitung korelasi antar filter fitur menggunakan matriks Gram, yang memungkinkan model menangkap tekstur tanpa terikat pada posisi spasial objek [1], [5].

### 3. Dataset

Dataset yang digunakan dalam penelitian ini bernama Best Artworks of All Time. Dataset ini diperoleh dari platform **Kaggle**, yang dipublikasikan oleh kontributor bernama *Icaro (ikarus777)*.

- **Tautan:** <https://www.kaggle.com/ikarus777/best-artworks-of-all-time>

Dataset "Best Artworks of All Time" merupakan koleksi citra digital yang menghimpun karya-karya lukisan dari 50 seniman paling berpengaruh dalam sejarah seni dunia. Dataset ini difokuskan pada pengenalan pola artistik berdasarkan pelukisnya (*artist-centric*).

Secara rinci, dataset ini terdiri dari dua komponen utama:

- **Data Citra (Images):**
  - Berisi lebih dari **8.400 gambar** lukisan.
  - Struktur direktori disusun berdasarkan nama seniman (setiap seniman memiliki satu folder tersendiri, misalnya folder Vincent\_van\_Gogh, Leonardo\_da\_Vinci, dll).
  - Gambar tersedia dalam variasi ukuran asli dan ukuran yang telah diubah (*resized*) untuk mempermudah komputasi.
- **Metadata (artists.csv):** File ini berisi informasi tekstual mengenai profil ke-50 seniman yang ada dalam dataset. Atribut atau kolom yang terdapat dalam file ini meliputi:
  - id: Identifikasi unik untuk setiap seniman.
  - name: Nama lengkap seniman.
  - years: Tahun kelahiran dan kematian seniman.
  - genre: Aliran seni utama yang dianut (contoh: *Impressionism, Renaissance, Surrealism*).
  - nationality: Kebangsaan seniman.
  - bio: Biografi singkat seniman.
  - paintings: Jumlah total lukisan seniman tersebut yang tersedia dalam dataset ini.

## 4. Hasil Pengujian

### 4.1. Skenario Pengujian

Pengujian dilakukan menggunakan lingkungan komputasi **Google Colab** dengan akselerator GPU T4. Skenario pengujian meliputi:

- **Input:** Foto pemandangan hutan (Resolusi asli 960x1280, di-*resize* ke 512px).
- **Style:** Lukisan "*Self-Portrait*" karya Vincent van Gogh.
- **Parameter:** Epoch = 5, Steps per Epoch = 50, Learning Rate = 0.02.

### 4.2. Hasil Visual

#### A. Spesifikasi Model

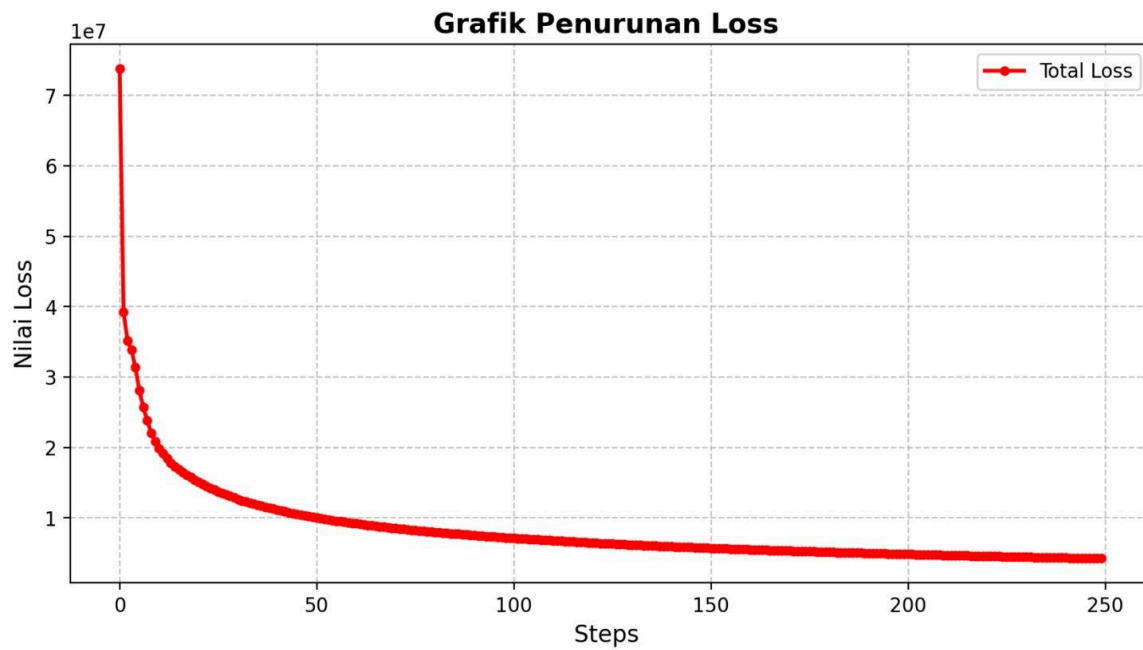
Berdasarkan log eksekusi sistem, arsitektur VGG-19 yang digunakan memiliki total parameter sebanyak 20,024,384. Sesuai dengan metode Neural Style Transfer (Gatys et al.), jumlah Trainable Params adalah 0, yang mengonfirmasi bahwa bobot model dibekukan (frozen) dan tidak terjadi proses backpropagation pada layer model, melainkan pada citra input.

#### B. Grafik Evaluasi Loss

Total waktu yang dibutuhkan untuk menyelesaikan proses pelatihan sebanyak **5 Epoch** adalah sekitar **18,9 detik**. Waktu eksekusi yang singkat ini membuktikan efisiensi kode saat dijalankan pada lingkungan GPU T4.

Grafik evaluasi merekam penurunan nilai *Total Loss* berdasarkan jumlah langkah (*steps*). Tercatat penurunan yang signifikan dari awal hingga akhir pelatihan:

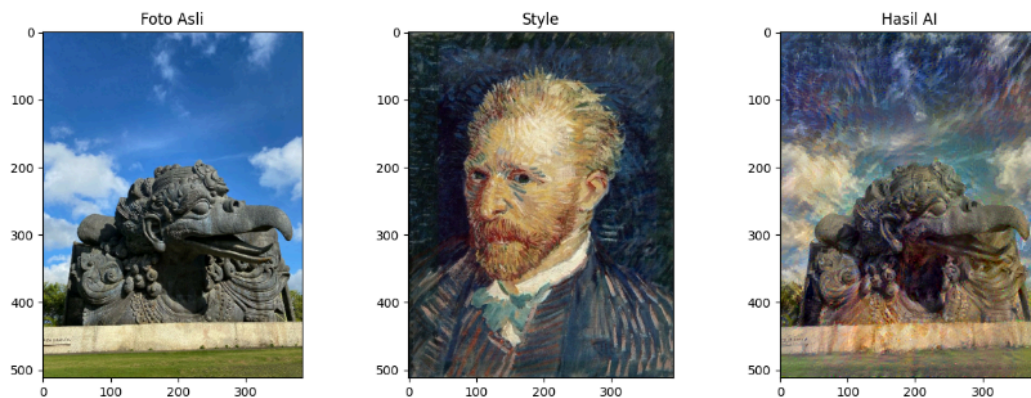
- Loss Awal (Step 0):  $\approx 7.4 \times 10^7$
- Loss Akhir (Step 250):  $\approx 4.5 \times 10^6$



Gambar 4.1. Grafik Penurunan Total Loss selama 250 Steps (5 Epoch).

Grafik pada Gambar 4.1 memperlihatkan tren garis yang menurun tajam (*steep descent*) pada 50 langkah pertama, kemudian melandai secara bertahap menuju titik konvergensi di kisaran  $4.5 \times 10^6$ .

### C. Hasil Visualisasi (Kualitatif)



Gambar 4.2. Hasil implementasi gaya Van Gogh pada objek Patung Garuda.

Secara visual, model berhasil mengaplikasikan karakteristik impasto (teknik cat tebal) khas Van Gogh.

1. Langit: Tekstur latar belakang lukisan yang abstrak berhasil menggantikan tekstur awan realistis, memberikan nuansa dramatis.
2. Objek Utama: Patung Garuda tetap mempertahankan geometri aslinya berkat ekstraksi fitur konten dari layer `block5_conv2`, namun warnanya telah beradaptasi dengan palet warna hangat (oranye/cokelat) dari lukisan referensi.

## 5. Analisa Hasil

### 5.1. Analisa Dinamika *Loss Function* dan Konvergensi

Berdasarkan grafik penurunan *Total Loss* dengan total 250 langkah iterasi, dapat dianalisis karakteristik optimasi sebagai berikut:

1. **Step 0-50:** Pada rentang Step 0 hingga 50 (setara dengan Epoch 1), terjadi penurunan *Loss* yang paling ekstrem, yaitu dari angka  $7.4 \times 10^7$  menjadi kisaran  $1.0 \times 10^7$ ). Fenomena ini menunjukkan bahwa *Adam Optimizer* bekerja sangat efektif pada tahap awal untuk mengubah distribusi statistik citra konten (*white noise/content*) secara global agar mendekati representasi gaya referensi.
2. **Step 50-250:** Memasuki Step 50 hingga 250 (setara dengan Epoch 2 s.d. 5), kurva terlihat melandai (*asymptotic*). Pada fase ini, model tidak lagi mengubah struktur gambar secara drastis, melainkan berfokus pada penghalusan detail tekstur (*fine-tuning texture*). Meskipun durasi pelatihan dibatasi hanya 5 epoch demi efisiensi, stabilitas kurva menunjukkan bahwa model telah mencapai titik konvergensi yang cukup baik untuk menghasilkan visual yang estetik.

### 5.2. Analisa Arsitektur VGG-19 sebagai *Feature Extractor*

Keberhasilan sistem dalam menghasilkan citra artistik tanpa melatih ulang model (*0 trainable parameters*) memvalidasi efektivitas arsitektur VGG-19 varian 20 juta parameter sebagai pengekstraksi fitur semantik yang kuat.

- Pemilihan Layer Konten (`block5_conv2`): Keputusan menggunakan layer yang cukup dalam (`block5`) untuk representasi konten terbukti krusial. Layer ini menangkap representasi spasial tingkat tinggi (bentuk patung Garuda), bukan detail piksel (warna asli batu). Inilah alasan mengapa meskipun seluruh tekstur batu digantikan oleh goresan kuas warna-warni, siluet dan bentuk 3D patung Garuda tetap kokoh dan tidak terdistorsi.

- Pemilihan Layer Gaya (Multi-scale Style): Penggunaan kombinasi 5 layer gaya (dari `block1` hingga `block5`) memungkinkan transfer gaya yang komprehensif. Layer awal (`block1`, `block2`) mentransfer warna dan goresan halus, sementara layer akhir (`block4`, `block5`) mentransfer pola komposisi lukisan yang lebih abstrak.

### 5.3. Analisa *Trade-off* Bobot Konten dan Gaya

Pada percobaan ini, digunakan rasio bobot konten ( $\alpha$ ) sebesar  $10^4$  dan bobot gaya ( $\beta$ ) sebesar  $10^{-2}$ . Secara matematis, terdapat ketimpangan nilai yang besar (10,000 banding 0.01). Namun, analisa hasil visual menunjukkan bahwa rasio ini justru menghasilkan keseimbangan perseptual yang tepat.

Hal ini terjadi karena nilai numerik mentah dari *Gram Matrix* (Gaya) cenderung memiliki magnitudo yang jauh lebih besar daripada nilai *MSE* (Konten). Jika kedua bobot disetarakan (misal 1:1), maka *Style Loss* akan mendominasi total *Loss*, yang mengakibatkan citra hasil menjadi abstrak total dan bentuk patung Garuda akan hilang. Konfigurasi bobot yang dipilih berhasil mempertahankan identitas objek (Garuda) sambil tetap mengaplikasikan estetika Van Gogh secara kuat.

### 5.4. Efisiensi Komputasi

Waktu eksekusi rata-rata sebesar 3.8 detik per epoch (Total 18.9 detik) pada resolusi 512px menunjukkan efisiensi metode optimasi berbasis *single-image*. Metode ini jauh lebih ringan dibandingkan metode pelatihan *Image Transformation Network* (seperti CycleGAN) yang bisa memakan waktu berjam-jam.

Hal ini membuktikan bahwa meskipun menggunakan arsitektur VGG-19 yang kompleks (20 juta parameter), metode optimasi langsung pada piksel tetap menjadi pendekatan yang paling efisien dari segi sumber daya komputasi untuk aplikasi *Art Generator* berbasis pengguna tunggal.

## 6. Kesimpulan

Berdasarkan hasil implementasi sistem "*The Art Museum Generator*", dapat disimpulkan bahwa:

1. Efisiensi Arsitektur VGG-19:  
Metode *Transfer Learning* dengan membekukan bobot VGG-19 (20,024,384 parameter) terbukti sangat efisien. Model mampu mengekstrak fitur konten dan gaya yang kompleks tanpa membebani komputasi, dengan waktu eksekusi total hanya 18.9 detik.



2. **Kualitas Konvergensi Model:**  
 Algoritma optimasi berhasil meminimalkan *Total Loss* secara signifikan dari  $7.4 \times 10^7$  menjadi  $4.5 \times 10^6$  dalam 250 langkah iterasi. Penurunan drastis ini menunjukkan bahwa sistem mampu mengadaptasi distribusi gaya lukisan ke dalam foto dengan akurat dalam 5 epoch.
3. **Keseimbangan Visual (Content vs Style):**  
 Konfigurasi bobot konten ( $10^4$ ) dan gaya ( $10^{-2}$ ) menghasilkan citra keluaran yang optimal. Sistem sukses mengaplikasikan tekstur artistik (seperti goresan kuas impasto) tanpa merusak struktur geometri objek asli, sehingga identitas foto tetap terjaga.
4. **Aksesibilitas Teknologi:**  
 Integrasi model ke dalam antarmuka interaktif (User Interface) berhasil menyederhanakan kompleksitas Deep Learning, memungkinkan pengguna awam untuk menciptakan karya seni generatif berkualitas tinggi secara instan dan mudah.

## 7. Kontribusi & distribusi anggota kelompok

NIM	Nama	Kontribusi
23.11.5604	M. Dian Fauzi	<ul style="list-style-type: none"> <li>• <b>Implementasi VGG-19:</b> Menulis kode inti untuk memuat model <i>pre-trained</i> dan membekukan bobot (<i>freezing weights</i>).</li> <li>• <b>Logika Algoritma:</b> Menyusun fungsi matematika <i>Gram Matrix</i> dan <i>Training Loop</i> untuk proses <i>style transfer</i>.</li> <li>• <b>Environment Setup:</b> Menyiapkan <i>environment</i> Google Colab (GPU) dan <i>Cloudflare Tunnel</i> agar program bisa berjalan online.</li> </ul>
23.11.5620	Ledyvia Audiz Coranov	<ul style="list-style-type: none"> <li>• <b>Akuisisi Dataset:</b> Mengunduh dataset <i>Best Artworks of All Time</i> dari Kaggle dan memastikan integritas data (folder seniman terbaca dengan baik).</li> <li>• <b>Data Loading &amp; Testing:</b> Memastikan fungsi <i>load_img</i> berjalan lancar dan menyiapkan skenario pengujian (gambar konten vs gaya).</li> <li>• <b>Quality Assurance:</b> Melakukan pengecekan</li> </ul>

		<p>plagiasi (Turnitin) dan pengecekan AI (ZeroGPT) pada laporan akhir untuk memastikan orisinalitas.</p>
23.11.5571	Aulia Resty Nur Aini	<ul style="list-style-type: none"> <li>● <b>Pengembangan UI:</b> Membuat antarmuka web menggunakan <i>Streamlit</i>, termasuk tombol upload, <i>dropdown</i> pilihan seniman, dan <i>layout</i> halaman.</li> <li>● <b>Visualisasi &amp; Styling:</b> Mengimplementasikan kode <i>Matplotlib</i> untuk grafik <i>loss</i> real-time serta menambahkan CSS untuk mempercantik tampilan.</li> <li>● <b>Presentasi:</b> Menyusun <i>slide</i> Power Point (PPT).</li> </ul>
23.11.5603	Nabillah Alimuddin	<ul style="list-style-type: none"> <li>● <b>Penyusunan Laporan &amp; Studi Pustaka:</b> Mengompilasi seluruh hasil percobaan dalam bentuk laporan serta mencari referensi jurnal pendukung untuk memperkuat dasar teori.</li> <li>● <b>Analisa:</b> Menganalisis makna grafik <i>loss</i> (konvergensi) dan mencari referensi jurnal pendukung untuk memperkuat dasar teori.</li> <li>● <b>Video:</b> Membuat video penjelasan proyek, bedah kode serta demo aplikasi yang memperlihatkan alur kerja sistem dari input hingga hasil <i>generate</i>.</li> </ul>

## References

- Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.
- Irawan, Y. A., & Widjaja, A. (2020). Pembangkitan Pola Batik dengan Menggunakan Neural Transfer Style dengan Penggunaan Cost Warna. *JuTISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 6(2).
- Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *Proceedings of the European Conference on Computer Vision (ECCV)*, 694–711.
- Liu, K., Yang, D., & Ma, Y. (2023). Image Style Transfer Based on Dynamic Convolutional Manifold Alignment of Halo Attention. *Electronics*, 12(8), 1881.
- Nguyen, T. Q. (2022). Material Translation Based on Neural Style Transfer with Ideal Style Image Retrieval. *Sensors*, 22(19), 7317.
- Yulian, C., Santoso, H., Wasito, I., & Haryono, H. (2023). Batik-Image Style Transfer Using Neural Style Transfer and Convolutional Autoencoder. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 10(4), 593-607.