

# Dokumen Desain Teknis & Konseptual

## Proyek

### Aethera Pixelis: Aplikasi Transformasi Citra Digital Berbasis Pipeline Formula

#### 1. Ringkasan Konsep

**Aethera Pixelis** adalah aplikasi web interaktif berbasis Streamlit yang dirancang untuk melakukan transformasi, pemulihan, dan peningkatan kualitas citra digital melalui serangkaian proses terstruktur yang disebut **Pipeline Formula**. Tujuan utama proyek ini adalah mendemonstrasikan pemahaman mendalam terhadap integrasi algoritma *filtering*, *restoration*, dan *enhancement* modern dalam satu kesatuan sistem yang fleksibel.

Keunikan Aethera Pixelis terletak pada konsep *Formula Pipeline* yang memungkinkan pengguna merangkai, mengatur parameter, dan menjalankan urutan operasi pengolahan citra secara modular. Nilai akademik proyek ini sangat tinggi, mencakup implementasi algoritma klasik (seperti *Bilateral Filter* dan *Unsharp Mask*) hingga integrasi model *Deep Learning* modern (seperti ESRGAN/SwinIR) untuk *Super Resolution*, serta desain *Human-Computer Interaction (HCI)* yang baik untuk visualisasi proses yang kompleks. Aplikasi ini tidak hanya menghasilkan output visual yang unggul, tetapi juga menyajikan dokumentasi langkah demi langkah dari proses yang terjadi.

#### 2. Fitur Utama Aethera Pixelis

Nama Fitur	Deskripsi Fungsi	Input & Output	Metode & Algoritma Inti	Parameter Utama	Metrik Evaluasi
Super Resolution 4K Enhance	Meningkatkan resolusi citra input beresolusi rendah menjadi citra tajam dan realistis hingga skala 4x	Input: Citra (Low Resolution, ~720p). Output: Citra (High Resolution, ~4K).	Denoising (Bilateral/Non-Local Means) → Upscaling (ESRGAN/SwinIR) → Sharpening (Unsharp Mask).	scale_factor, denoise_sigma, sharpen_amount.	PSNR, SSIM, LPIPS (untuk kualitas perseptual), Visual Assessment (ketajaman & artefak).

	(resolusi 4K), dengan fokus pada pengurangan <i>noise</i> pra-pemrosesan.				
<b>Creative Filmic Effect</b>	Menerapkan efek artistik sinematik atau retro film vintage melalui manipulasi tonal, <i>color grading</i> , dan penambahan tekstur simulasi butiran film.	Input: Citra (Color/Gray scale). Output: Citra (Color graded/Artistik).	Tone Mapping (LUT/Tone Curve) → Color Grading (Histogram Matching) → Texture Simulation (Film Grain Overlay).	LUT_preset, contrast_boost, grain_intensity, color_temp.	Visual Assessment (estetika), Histogram Analysis (distribusi warna), Qualitative Feedback.
<b>Selective Texture Enhancement</b>	Meningkatkan detail tekstur secara lokal pada area tertentu (misalnya kulit, kain, atau batu) sambil menjaga area halus (seperti langit atau <i>out-of-focus</i> ) dari peningkatan	Input: Citra (Color/Gray scale). Output: Citra (Enhanced Texture, Low Global Noise).	Detail Extraction (Laplacian Pyramid/Guided Filter) → Adaptive Sharpening (RL/Wiener) → Local Contrast Stretching.	radius_local, edge_threshold, contrast_limit.	SSIM Lokal, Edge Response Analysis, Visual Assessment (detail & noise).

	n noise yang berlebihan.				
--	--------------------------------	--	--	--	--

### 3. Formula Pipeline (Langkah-langkah proses)

Konsep inti dari Aethera Pixelis adalah rangkaian langkah pemrosesan yang terdefinisi.

#### 3.1. Super Resolution 4K Enhance

**Tujuan:** Mengurangi noise sebelum upscaling (prasyarat model DL), kemudian melakukan upscaling cerdas, dan terakhir mempertajam hasil tanpa memperkenalkan *ringing artifact*.

##### Pipeline Formula Penuh:

Input Image (Low Resolution)

→ Pre-Denoising(Bilateral Filter,  $d=9$ ,  $\sigma_{\text{Color}}=75$ ,  $\sigma_{\text{Space}}=75$ )

{Fungsi: Menghaluskan permukaan tanpa mengaburkan tepi. Alasan: Noise akan diperbesar 4x oleh model SR; harus dieliminasi di awal.}

→ Upscaling (ESRGAN/SwinIR,  $\text{scale}=4$ ,  $\text{model}=\text{'x4plus'}$ )

{Fungsi: Rekonstruksi detail resolusi tinggi menggunakan jaringan saraf. Alasan: Memberikan hasil \*super resolution\* yang realistis dan superior dibandingkan interpolasi klasik.}

→ Post-Sharpening (Unsharp Mask,  $\text{amount}=1.2$ ,  $\text{radius}=2.0$ )

{Fungsi: Meningkatkan kontras lokal dan ketajaman yang mungkin sedikit hilang pada tahap upscaling. Alasan: Mengkompensasi hasil yang kadang terlalu halus dari model DL.}

→ Output Enhanced Image (4K Resolution)

##### Simplified Pseudo-Pipeline:

Input Image

→ Denoise(BF)

→ Upscale(ESRGAN)

→ Sharpen(UM)

→ Output

#### 3.2. Creative Filmic Effect

**Tujuan:** Menggeser palet warna dan tonalitas citra agar menyerupai tampilan film analog atau sinema digital tertentu.

##### Pipeline Formula Penuh:

Input Image (Asli)

→ Tonal Mapping (LUT 3D Transform, preset='Cinematic\_Ektar100')

{Fungsi: Memetakan nilai RGB input ke nilai RGB baru yang sudah dikalibrasi. Alasan: LUT (Look-Up Table) adalah metode tercepat dan paling efisien untuk \*color grading\* non-linear.}

→ Color Correction (Histogram Matching, target='Teal\_Orange\_Target')

{Fungsi: Menyesuaikan histogram input agar sesuai dengan histogram referensi target film.

Alasan: Memastikan distribusi warna mengikuti pola estetik yang diinginkan (misalnya, \*shadow\* menjadi \*teal\*, \*highlight\* menjadi \*orange\*).}

→ Texture Simulation (Film Grain Overlay, intensity=0.15, size=2.0)

{Fungsi: Menambahkan lapisan noise frekuensi tinggi Gaussian/Perlin noise. Alasan:

Mensimulasikan butiran film (grain) untuk memberikan kesan otentik dan tekstural pada hasil akhir.}

→ Output Filmic Image

### **Simplified Pseudo-Pipeline:**

Input Image

→ Tone(LUT)

→ Color(HistMatch)

→ Texture(Grain)

→ Output

## **3.3. Selective Texture Enhancement**

**Tujuan:** Meningkatkan detail pada tekstur lokal secara cerdas tanpa meningkatkan noise secara global.

### **Pipeline Formula Penuh:**

Input Image (Asli)

→ Detail Extraction (Guided Filter, radius=30, epsilon=0.01)

{Fungsi: Memisahkan citra menjadi lapisan dasar (base layer) dan lapisan detail (detail layer).

Alasan: Guided Filter efektif dalam pemisahan tepi/detail secara halus dan cepat.}

→ Adaptive Sharpening (Detail Layer Boost, gain=1.5, clip\_limit=2.0)

{Fungsi: Peningkatan kontras hanya pada lapisan detail, seringkali menggunakan teknik seperti Reinard Local Contrast/CLAHE adaptif. Alasan: Membatasi peningkatan hanya pada frekuensi tinggi (tekstur) tanpa mempengaruhi frekuensi sangat tinggi (noise).}

→ Fusion & Local Adjustment (Laplacian Pyramid Fusion)

{Fungsi: Menggabungkan kembali lapisan dasar dan lapisan detail yang sudah ditingkatkan.

Alasan: Teknik fusi multi-skala memastikan detail yang ditingkatkan terintegrasi kembali dengan mulus tanpa artefak tepi yang tajam.}

→ Output Textured Image

Simplified Pseudo-Pipeline:

- Input Image
- Extract(GuidedFilter)
  - Enhance(AdaptiveSharpen)
  - Merge(LaplacianFusion)
  - Output

4. Rekomendasi Metode & Algoritma

Nama Metode	Fungsi & Keunggulan	Rumus Dasar	Alasan Akademik Pemilihan	Kompleksitas Waktu/Prosesing
Bilateral Filter	Edge-Preserving Smoothing. Menghaluskan citra dengan mempertahankan tepi. Lebih baik dari Gaussian Blur untuk denoising awal.	$I_p = \frac{1}{W_p} \sum_{q \in S} I_q \cdot G_{\sigma_s}( p-q ) \cdot G_{\sigma_r}(\dots)$	$I_p - I_q$	)\$
ESRGAN (Enhanced Super-Resolution Generative Adversarial Networks)	Deep Learning Super Resolution. Menghasilkan citra 4x resolusi lebih tinggi dengan detail yang realistis, dilatih pada metrik perseptual (LPIPS).	Arsitektur GAN dengan Residual-in-Residual Dense Block (RRDB) dan fungsi perceptual loss.	Representasi dari algoritma state-of-the-art (SOTA) di bidang Deep Learning untuk restorasi citra.	Tinggi, membutuhkan GPU (GFLOPs). Fallback CPU sangat lambat, gunakan untuk demonstrasi saja.

<b>Unsharp Mask (UM)</b>	<i>Sharpening</i> klasik melalui peningkatan kontras tepi. Bekerja dengan mengurangi versi blur dari citra asli.	$I_{\text{sharpened}} = I_{\text{original}} + \alpha \cdot (I_{\text{original}} - I_{\text{blurred}})$	Sederhana, cepat, dan efektif sebagai tahap pasca-pemrosesan (Post-Sharpening) yang ringan.	$O(N)$ (sangat cepat).
<b>3D Look-Up Table (LUT)</b>	<i>Color Grading</i> non-linear. Mengubah skema warna citra secara cepat dan konsisten dengan hasil yang sudah divalidasi oleh profesional.	Memetakan triplet (R, G, B) input ke (R', G', B') output melalui kubus 3D $N^3$ .	Konsep pemetaan warna (color mapping) yang efisien, menunjukkan pemahaman <i>color science</i> dalam DIP.	$O(1)$ (lookup time, sangat cepat) setelah tabel di-load.
<b>Guided Filter</b>	Filter <i>edge-aware</i> yang digunakan untuk ekstraksi detail. Dapat memisahkan lapisan dasar ( <i>base</i> ) dan lapisan detail ( <i>detail</i> ) dengan sangat baik.	Meminimalkan fungsi energi yang melibatkan citra input dan citra panduan ( <i>guidance image</i> ).	Pemahaman terhadap filter kontekstual yang menghindari artefak gradien yang disebabkan oleh filter non-linear sederhana lainnya.	$O(N)$ (cepat, dapat diimplementasikan menggunakan integral image).

## 5. Desain Aplikasi Streamlit

### Layout UI

Aethera Pixelis akan mengadopsi layout *modern, clean*, dan berorientasi pada *dashboard* data visual.

- **Sidebar (st.sidebar):**

- **Header:** Logo>Nama "Aethera Pixelis" dan deskripsi singkat.
- **Input Area:** st.sidebar.file\_uploader untuk mengunggah citra (mendukung format JPG, PNG).
- **Pipeline Selector:** st.sidebar.selectbox untuk memilih salah satu dari 3 fitur inti.
- **Action Button:** st.sidebar.button "Proses Citra" (diposisikan di bawah parameter).
- **Main Content (st.columns):**
  - **Header Utama:** Judul fitur yang dipilih.
  - **Pipeline Editor/Viewer (Top Section):** Area teks (*read-only*) yang menampilkan **Formula Pipeline** yang sedang aktif.
  - **Parameter Slider (Mid Section):** st.slider dan st.number\_input yang diorganisir menggunakan st.expander untuk setiap langkah (Pre-Denoising, Upscaling, Sharpening, dll.).
  - **Visual Output Panel (Bottom Section):**
    - Panel kiri (st.column 1): Citra Input Asli.
    - Panel kanan (st.column 2): Citra Output Hasil Pemrosesan, ditampilkan menggunakan *Before-After Slider* (jika memungkinkan) atau berdampingan.
- **Footer/Log:** st.expander di bagian bawah untuk menampilkan metrik evaluasi (PSNR/SSIM) dan log proses (*status logs*).

## Interaksi Pengguna (User Flow Diagram Naratif)

1. **Mulai & Unggah:** Pengguna membuka aplikasi. Menggunakan st.session\_state untuk menjaga state.
2. **Input:** Pengguna mengunggah gambar melalui st.sidebar.file\_uploader.
3. **Pemilihan Pipeline:** Pengguna memilih fitur (misalnya, *Super Resolution 4K Enhance*) dari st.sidebar.selectbox.
4. **Konfigurasi Parameter:** Parameter default dimuat. Pengguna menyesuaikan st.slider (misalnya, scale\_factor, denoise\_sigma, sharpen\_amount).
5. **Eksekusi:** Pengguna menekan st.sidebar.button ("Proses Citra").
6. **Visual Feedback:** st.spinner atau st.progress (progress bar) muncul, menunjukkan proses sedang berjalan.
7. **Hasil:** Citra hasil dan input ditampilkan berdampingan menggunakan st.columns dan st.image. Metrik hasil (PSNR/SSIM) ditampilkan di footer log.
8. **Output:** Tombol st.download\_button memungkinkan pengguna menyimpan hasil.

## Warna & Tema Rekomendasi

- **Tone:** Elegan, Modern, Clean.
- **Warna:**
  - Primer (Accent): Biru Tua/Navy (melambangkan "Aethera"/langit malam, stabilitas, dan teknologi). #0F4C75
  - Sekunder (Highlight): Teal atau Cyan (melambangkan "Pixelis", *digital* dan *enhancement*). #3282B8
  - Background: Light Gray atau Putih bersih untuk Streamlit, dengan *dark theme*

sebagai alternatif.

## 6. Struktur Kode (Pseudocode / Skeleton)

Struktur kode modular ini memisahkan logika antarmuka (Streamlit) dari logika pemrosesan citra (Pipeline) dan utilitas (IO/Metriks).

/aethera\_pixelis

```
|
|— app.py
|— pipeline/
|   |— filters.py      # Implementasi Bilateral, Unsharp Mask, Guided Filter
|   |— restoration.py  # Implementasi ESRGAN/SwinIR loader, Denoising
|   |— enhancement.py  # Implementasi LUT, Hist Matching, Grain Simulation
|
|— utils/
|   |— io_utils.py     # Fungsi load/save citra, konversi BGR/RGB
|   |— metrics.py      # Fungsi perhitungan PSNR, SSIM, LPIPS
|
|— assets/
|   |— sample_images/ # Citra uji default
|   |— style_presets/ # File LUT (.cube) atau preset histogram
```