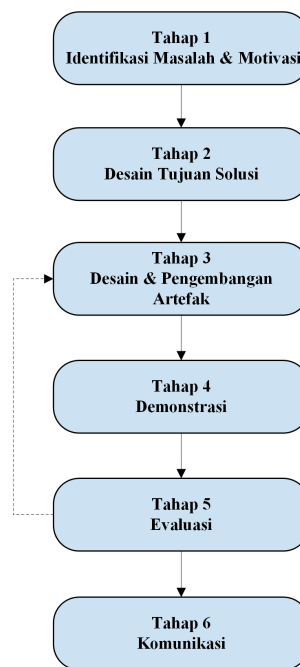


BAB III. Metodologi Penelitian

Bab ini menguraikan pendekatan penelitian yang mengintegrasikan *Design Science Research Methodology* (DSRM) sebagai kerangka sistematis untuk identifikasi masalah, penetapan tujuan, desain, evaluasi, dan komunikasi hasil penelitian. Metodologi ini dipilih karena kesesuaiannya dengan tujuan penelitian, yaitu merancang, mengembangkan, dan mengevaluasi sebuah artefak berupa *framework* restorasi dokumen berbasis GAN dengan Diskriminator Dual-Modal dan fungsi *loss* berorientasi HTR. Implementasi teknis *framework* dilakukan melalui siklus iteratif *deep learning* yang berbasis bukti empiris dan dokumentasi sistematis.

III.1 Kerangka Teori Metodologi Penelitian

III.1.1 *Design Science Research Methodology* (DSRM)



Gambar 1: Framework Design Science Research Methodology yang diadaptasi untuk penelitian ini dengan siklus iteratif antara evaluasi dan desain untuk penyempurnaan artefak.

Design Science Research Methodology (DSRM), yang diperkenalkan oleh Peffers et al. (2007), merupakan kerangka penelitian yang dirancang khusus untuk pengembangan dan evaluasi artefak teknologi informasi. DSRM terdiri dari enam tahapan yang saling terkait dan bersifat iteratif, di mana hasil dari tahap

evaluasi memberikan umpan balik untuk penyempurnaan pada tahap desain dan pengembangan.

III.1.2 Adaptasi DSRM untuk Pengembangan *Framework* GAN-HTR

Karakteristik utama DSRM yang relevan dengan penelitian ini adalah:

1. **Orientasi Solusi:** DSRM berfokus pada penciptaan artefak inovatif untuk menyelesaikan masalah praktis yang teridentifikasi. Dalam konteks penelitian ini, artefak yang dikembangkan adalah *framework* GAN dengan Diskriminator Dual-Modal yang secara eksplisit dioptimalkan untuk meningkatkan keterbacaan dokumen oleh sistem HTR.
2. **Evaluasi Berbasis Bukti:** Setiap iterasi desain dievaluasi secara kuantitatif menggunakan metrik yang terukur (PSNR, SSIM, CER, WER), memastikan bahwa penyempurnaan dilakukan berdasarkan data empiris bukan asumsi.
3. **Siklus Iteratif:** Kerangka kerja ini memfasilitasi perbaikan berkelanjutan melalui putaran umpan balik (*feedback loop*) antara tahap evaluasi dan desain, yang sangat sesuai dengan karakteristik pengembangan model *deep learning* yang memerlukan eksperimen dan *tuning* berulang.
4. **Komunikasi dan Diseminasi:** DSRM menekankan pentingnya mendokumentasikan dan mengkomunikasikan hasil penelitian kepada komunitas ilmiah, baik melalui publikasi akademis maupun berbagi kode sumber untuk reproduktifitas.

III.1.3 Siklus Iteratif Berbasis Bukti Empiris

Penelitian ini mengadopsi siklus iteratif yang berbasis bukti empiris untuk memastikan setiap keputusan desain didukung oleh data yang valid. Setiap tahap evaluasi menghasilkan temuan kuantitatif yang menjadi dasar untuk penyempurnaan artefak pada iterasi berikutnya. Pendekatan ini meminimalkan subjektivitas dan meningkatkan validitas ilmiah dari hasil penelitian.

III.2 Tahapan Penelitian

Penelitian ini mengikuti enam tahapan DSRM yang telah diadaptasi untuk konteks pengembangan *framework* restorasi dokumen berbasis *deep learning*. Setiap tahapan dijelaskan secara rinci berikut dengan metode, perangkat lunak, dan dokumentasi yang digunakan.

III.2.1 Identifikasi Masalah dan Motivasi (Tahap 1)

Tahap pertama berfokus pada analisis mendalam terhadap tantangan yang ada dalam restorasi dokumen historis dan identifikasi kesenjangan penelitian yang menjadi motivasi pengembangan framework baru.

Aktivitas Utama:

1. Studi Literatur Sistematis

Dilakukan kajian komprehensif terhadap penelitian terkait restorasi dokumen, GAN, dan HTR untuk mengidentifikasi keterbatasan metode yang ada. Fokus utama pada analisis metode *state-of-the-art* seperti DE-GAN, TEXT-DIAE, dan CycleGAN.

2. Analisis Kebutuhan Praktis

Dilakukan analisis terhadap kebutuhan nyata lembaga kearsipan, dalam hal ini Arsip Nasional Republik Indonesia, di mana restorasi digital harus mendukung transkripsi otomatis untuk aksesibilitas dan analisis data historis. Identifikasi menunjukkan bahwa dokumen kolonial Belanda mengalami degradasi kompleks: *bleed-through*, *fading* tinta *iron gall*, noda air (*water stains*), bercak kecokelatan (*foxing*), dan kabur optik (*optical blur*) dari proses pemindaian.

3. Perumusan Masalah Penelitian

Berdasarkan analisis literatur dan kebutuhan praktis, dirumuskan masalah utama penelitian: “Bagaimana merancang arsitektur GAN yang secara eksplisit dioptimalkan untuk meningkatkan keterbacaan teks oleh sistem HTR, bukan hanya kualitas visual, pada dokumen historis dengan degradasi kompleks?”

Masalah ini diformulasikan menjadi empat pertanyaan penelitian yang telah dijelaskan pada Bab I.

III.2.2 Definisi Tujuan Solusi (Tahap 2)

Berdasarkan masalah yang teridentifikasi, tujuan dari artefak yang akan dibangun didefinisikan secara kuantitatif dan kualitatif dengan target yang terukur.

Tujuan Kuantitatif: Penelitian ini menetapkan target performa yang dapat diukur dan diverifikasi:

Tabel 1: Target metrik kuantitatif penelitian

Kategori	Metrik	Target
Kualitas Visual	PSNR	> 35 dB
	SSIM	> 0.95
Keterbacaan Teks	CER	Penurunan signifikan vs baseline
	WER	Penurunan signifikan vs baseline
Stabilitas Training	Generator Loss	Konvergen tanpa mode collapse
	Diskriminator Accuracy	60-80% (Nash equilibrium)
Efisiensi Komputasi	Inference Time	< 15 detik per image (1024×128)

Tujuan Kualitatif: Mendefinisikan fitur inovatif dan kontribusi penelitian:

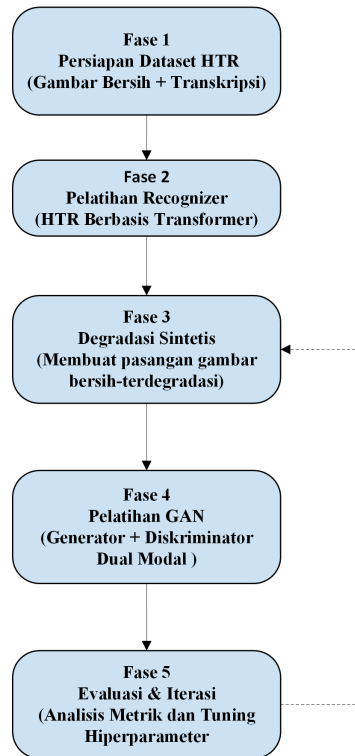
1. **Diskriminator Dual-Modal:** Merancang arsitektur diskriminator yang mampu mengevaluasi kualitas gambar dari dua perspektif simultan: realisme visual (*visual realism*) dan koherensi teks (*text coherence*).
2. **Fungsi Loss Berorientasi HTR:** Mengintegrasikan sinyal loss dari model HTR secara langsung ke dalam fungsi loss Generator, menciptakan optimasi ganda.
3. **Filosofi Recognizer Terbekukan:** Menggunakan *recognizer* dengan bobot yang dibekukan sebagai evaluator objektif dengan kualitas metrik yang stabil.
4. **Pipeline Degradasi Sintetis:** Mengembangkan *pipeline* degradasi sintetis yang realistis untuk menghasilkan pasangan pelatihan (*training pairs*) dari dokumen bersih.

III.2.3 Desain dan Pengembangan Artefak (Tahap 3)

Tahap ini merupakan inti penelitian di mana artefak (framework GAN-HTR) dirancang, diimplementasikan, dan disempurnakan melalui siklus iteratif berbasis eksperimen. Tahap ini mengikuti metodologi pengembangan deep learning yang terdokumentasi dan mencakup tiga aspek fundamental: penyiapan lingkungan komputasi, manajemen data dan eksperimen, serta pengembangan framework modular.

Fase 1: Persiapan Dataset HTR Dataset bersih disiapkan untuk melatih model Recognizer dengan struktur yang meliputi input gambar line text bersih, label ground truth transcription, format TFRecord untuk efisiensi loading, dan preprocessing berupa normalisasi intensitas, padding, resizing.

Fase 2: Pelatihan dan Pembekuan Recognizer Model Recognizer dilatih menggunakan arsitektur hybrid CNN-Transformer untuk HTR (spesifikasi arsitektur lengkap pada Bab IV bagian 4.2.3 dan Lampiran A). Fase ini mengadopsi



Gambar 2: *Pipeline* pengembangan *framework* GAN-HTR dengan lima fase utama yang dilakukan secara sekuensial dengan iterasi pada fase 4 dan 5.

metodologi *curriculum learning* dengan tiga strategi kunci untuk mencapai performa optimal dan stabilitas training:

Curriculum Learning Strategy: Penelitian ini menerapkan pendekatan bertahap dalam training Recognizer dimana model pertama-tama difokuskan pada pembelajaran fundamental melalui fase *warmup* (5 epochs awal) dengan learning rate yang meningkat secara linear, kemudian dilanjutkan dengan fase pembelajaran utama menggunakan *cosine annealing schedule*. Pendekatan ini memastikan model tidak terjebak pada local minima di awal training dan mencapai konvergensi yang lebih baik. Rationale: Transformer-based models sangat sensitif terhadap inisialisasi learning rate; warmup mencegah gradient explosion pada early iterations.

Data Augmentation untuk Robustness: Training menggunakan augmentasi data secara real-time pada setiap batch untuk meningkatkan kemampuan generalisasi model terhadap variasi dokumen real-world. Augmentasi mencakup variasi photometric (brightness, contrast) dan noise injection yang mensimulasikan kondisi scanning dan digitalisasi dokumen historis yang beragam. Rationale: Dataset synthetic memiliki distribusi terbatas; augmentasi memperluas ruang input dan mencegah overfitting pada karakteristik synthetic tertentu.

Multi-Layer Regularization: Kombinasi teknik regularisasi diterapkan pada berbagai tingkat arsitektur (dropout pada layers, label smoothing pada loss function, weight decay pada optimizer) untuk mencegah overfitting sambil mempertahankan kapasitas pembelajaran model. Early stopping dengan monitoring validation CER memastikan model berhenti training pada titik optimal sebelum terjadi degradasi performa. Rationale: Sequence-to-sequence models prone to overfitting; multi-layer regularization provides defense-in-depth.

Pembekuan Model (Freezing) dan Dual-Purpose Utilization: Setelah training mencapai convergence (ditandai dengan validation CER stabil), bobot model recognizer dibekukan (`training=False`) untuk memastikan konsistensi evaluasi selama training GAN. Keputusan freezing ini krusial karena: (1) menjaga stabilitas metrik HTR sebagai ground truth objektif, (2) mencegah *catastrophic forgetting* saat recognizer diekspos ke degraded images, dan (3) memungkinkan reproduktifitas eksperimen dengan evaluator yang konsisten.

Model frozen ini kemudian dimanfaatkan untuk dua tujuan dalam framework GAN-HTR: (1) ekstraksi *recognition features* dari layer intermediate (`proj_ln`) untuk *recognition feature loss* yang mengoptimalkan Generator di feature space, dan (2) perhitungan CTC loss yang di-backpropagate ke Generator untuk optimasi keterbacaan langsung di sequence space. Dual-purpose ini merupakan inovasi metodologis yang memungkinkan pembelajaran multi-level dari model HTR frozen.

Detail hyperparameter training, arsitektur layer-by-layer, dan konfigurasi optimizer tersedia pada Lampiran A dan B untuk keperluan reproduksi.

[H] [1] **Input:** Dataset HTR $\mathcal{D} = \{(x_i, t_i)\}_{i=1}^N$ dengan x_i = image, t_i = transcription
Output: Trained & Frozen Recognizer R_θ **Hyperparameters:** LR = 3×10^{-4} , batch_size = 32, warmup_epochs = 5 Initialize CNN-Transformer model R_θ dengan random weights Initialize optimizer AdamW dengan weight decay = 2×10^{-4} epoch = 1 to max_epochs
 // Warmup Phase (epoch 1-5): Linear LR ramp-up epoch \leq warmup_epochs $LR_{current} = LR_{max} \times \frac{epoch}{warmup_epochs}$ // Cosine Annealing: decay LR smoothly $LR_{current} = LR_{min} + \frac{1}{2}(LR_{max} - LR_{min})(1 + \cos(\pi \frac{epoch - warmup}{total_epochs}))$ each batch (X, T) in \mathcal{D}_{train} // Apply real-time augmentation $X_{aug} \leftarrow \text{RandomBrightness}(X, \pm 20\%) + \text{RandomContrast}(X, 0.8-1.2) + \text{GaussianNoise}(X, \sigma=0.08)$ // Forward pass $\hat{Y} \leftarrow R_\theta(X_{aug})$ // Output: (batch, timesteps, charset+1) // CTC Loss with label smoothing $\mathcal{L}_{CTC} \leftarrow \text{CTCLoss}(\hat{Y}, T, \epsilon = 0.1)$ // Backward pass with gradient clipping Compute gradients $\nabla_\theta \mathcal{L}_{CTC}$ Clip gradients to norm = 1.0 Update θ using AdamW // Validation Compute CER_{val} , WER_{val} on \mathcal{D}_{val} CER_{val} improved Save checkpoint R_θ^{best} Reset patience counter Increment patience counter patience ≥ 15

break // Early stopping // Freeze model for GAN training Load best checkpoint R_{θ}^{best} Set $R_{\theta}.\text{trainable} = \text{False}$ // Freeze all weights **return** $R_{\theta}^{\text{frozen}}$

Fase 3: Synthetic Degradation Pipeline Karena keterbatasan ketersediaan data berpasangan untuk dokumen historis, dikembangkan pipeline degradasi sintetis yang realistis untuk menghasilkan training pairs dari dokumen bersih. Pipeline ini mensimulasikan lima jenis degradasi utama: (1) bleed-through dengan overlay mirrored text, (2) fading dengan exponential intensity decay, (3) stains dengan random brown spots, (4) blur dengan Gaussian filter, dan (5) salt & pepper noise untuk simulasi sensor noise.

Fase 4: Implementasi dan Training GAN Framework GAN dengan komponen utama Generator (U-Net Architecture), Dual Modal Diskriminator, dan Multi-Component Loss Function diimplementasikan dan dilatih dengan konfigurasi yang telah ditentukan. Training procedure mengikuti strategi adversarial standard dengan augmentasi khusus berupa CTC loss annealing untuk stabilitas.

[H] [1] **Input:** Dataset triplet $\mathcal{D} = \{(x_{\text{deg}}, x_{\text{clean}}, t)\}$, Frozen Recognizer R_{θ} **Output:** Trained Generator G , Discriminator D **Hyperparameters:** $\lambda_{\text{adv}} = 1.0$, $\lambda_{L1} = 100.0$, $\lambda_{\text{CTC}} = 10.0$, warmup_epochs=2 Initialize Generator G (U-Net) dan Discriminator D (Dual-Modal) dengan random weights Initialize optimizers: Adam_G (LR= 2×10^{-4}), Adam_D (LR= 5×10^{-4}) epoch = 1 to max_epochs // CTC Annealing: warmup untuk stabilitas epoch \leq warmup_epochs $\lambda_{\text{CTC}}^{\text{current}} = 0.0$ // CTC disabled saat warmup $\lambda_{\text{CTC}}^{\text{current}} = \lambda_{\text{CTC}}$ // CTC activated setelah warmup each batch $(X_{\text{deg}}, X_{\text{clean}}, T)$ in \mathcal{D} // ===== TRAIN DISCRIMINATOR ===== // Generate fake images $X_{\text{fake}} \leftarrow G(X_{\text{deg}})$ // Get text representations (frozen recognizer) $Y_{\text{real}} \leftarrow R_{\theta}(X_{\text{clean}})$ // True text from clean image $Y_{\text{fake}} \leftarrow R_{\theta}(X_{\text{fake}})$ // Predicted text from fake image // Discriminator forward pass (dual-modal: image + text) $\text{score}_{\text{real}} \leftarrow D(X_{\text{clean}}, Y_{\text{real}})$ $\text{score}_{\text{fake}} \leftarrow D(X_{\text{fake}}, Y_{\text{fake}})$ // Binary Cross-Entropy with label smoothing (0.9 for real) $\mathcal{L}_D = -[\log(\text{score}_{\text{real}} \times 0.9) + \log(1 - \text{score}_{\text{fake}})]$ // Update Discriminator Compute $\nabla_D \mathcal{L}_D$ and update D via Adam_D // ===== TRAIN GENERATOR ===== // Generate fake images $X_{\text{fake}} \leftarrow G(X_{\text{deg}})$ // Get text representation for adversarial loss $Y_{\text{fake}} \leftarrow R_{\theta}(X_{\text{fake}})$ // Adversarial Loss: fool discriminator $\mathcal{L}_{\text{adv}} = -\log(D(X_{\text{fake}}, Y_{\text{fake}}))$ // L1 Reconstruction Loss: pixel similarity $\mathcal{L}_{L1} = \|X_{\text{clean}} - X_{\text{fake}}\|_1$ // CTC Loss: text readability (BACKPROPAGATED ke Generator!) $\mathcal{L}_{\text{CTC}} = \text{CTCLoss}(Y_{\text{fake}}, T)$ // Multi-Component Loss: weighted combination $\mathcal{L}_G = \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{L1} \mathcal{L}_{L1} + \lambda_{\text{CTC}}^{\text{current}} \mathcal{L}_{\text{CTC}}$ // Update Generator (CTC gradient flows through frozen R to G) Compute $\nabla_G \mathcal{L}_G$ and update G via Adam_G // Validation & Metrics Compute PSNR, SSIM, CER, WER on validation set Log metrics to MLflow PSNR improved or SSIM improved Save best checkpoint **return** $G^{\text{best}}, D^{\text{best}}$

Catatan Krusial pada Algorithm ??:

- **CTC Annealing (Line 9-13):** CTC loss di-disable selama 2 epochs awal (warmup) untuk memberi Generator waktu belajar task dasar (reconstruction) sebelum dioptimasi untuk keterbacaan. Setelah warmup, CTC activated dengan full weight (10.0).

- **Frozen Recognizer (Line 18-19, 32):** Recognizer R_θ dalam mode inference (weights frozen), tidak di-update. Namun, gradient dari CTC loss tetap di-backpropagate ke Generator untuk mengoptimasi visual quality.
- **Dual-Modal Discriminator (Line 21-22):** Discriminator menerima pasangan (image, text) untuk menilai koherensi visual-tekstual, bukan hanya realism visual.

Fase 5: Eksperimen dan Iterasi Berbasis Bukti Pengembangan framework dilakukan melalui siklus iteratif dengan eksperimen terkontrol yang meliputi perbandingan mode diskriminator, tuning loss weight, dan analisis anomali training.

Lingkungan Eksperimen Bagian ini merinci spesifikasi teknis dari lingkungan yang digunakan untuk penelitian, memastikan transparansi dan reproduktifitas.

Spesifikasi Perangkat Keras:

Tabel 3: Spesifikasi perangkat keras

Komponen	Spesifikasi
GPU	NVIDIA RTX A4000 (16 GB GDDR6)
CPU	Intel Xeon atau AMD Ryzen (multi-core)
RAM	32 GB DDR4
Storage	1 TB NVMe SSD (untuk dataset dan checkpoints)

Konfigurasi Perangkat Lunak:

Tabel 4: Framework deep learning dan library utama

Kategori	Tool/Library	Versi
Language	Python	3.10
Deep Learning Framework	TensorFlow/Keras	2.x
Dependency Management	Poetry	Latest
Numerical Computing	NumPy	Latest
Image Processing	OpenCV	4.x
Experiment Tracking	MLflow	2.x
Visualization	Matplotlib, Seaborn	Latest

III.2.4 Demonstrasi (Tahap 4)

Artefak yang telah dikembangkan didemonstrasikan untuk menunjukkan kemampuannya dalam menyelesaikan masalah restorasi dokumen terdegradasi dan meningkatkan keterbacaan HTR.

Studi Kasus: Penerapan model final pada sampel data uji yang representatif meliputi synthetic test set, DIBCO benchmark, dan real ANRI documents untuk menunjukkan generalisasi pada berbagai jenis degradasi.

Visualisasi Hasil: Untuk setiap sampel, dihasilkan visualisasi perbandingan yang menunjukkan input, output, reference, dan overlay untuk menunjukkan area yang direstorasi.

III.2.5 Evaluasi (Tahap 5)

Tahap ini mengukur performa artefak secara objektif terhadap tujuan yang telah didefinisikan dan memvalidasi hipotesis penelitian.

Evaluasi Kuantitatif: *Metrik Visual Quality:*

- **PSNR (Peak Signal to Noise Ratio):** Mengukur kesamaan piksel dengan ground truth. Target: > 35 dB
- **SSIM (Structural Similarity Index):** Mengukur kesamaan struktural. Target: > 0.95

Metrik Text Readability:

- **CER (Character Error Rate):** Rasio karakter yang salah dikenali. Target: Penurunan minimal 30% vs baseline
- **WER (Word Error Rate):** Rasio kata yang salah dikenali. Target: Penurunan signifikan vs baseline

```
[H] [1] Input: Test set  $\mathcal{D}_{\text{test}} = \{(x_{\text{deg}}, x_{\text{clean}}, t)\}$ , Trained Generator  $G$ , Frozen Recognizer  $R$  Output: Performance metrics (PSNR, SSIM, CER, WER) Initialize metric accumulators: PSNR_list, SSIM_list, CER_list, WER_list
each sample  $(x_{\text{deg}}, x_{\text{clean}}, t_{\text{gt}})$  in  $\mathcal{D}_{\text{test}}$  // ===== RESTORATION =====  $x_{\text{restored}} \leftarrow G(x_{\text{deg}})$  // Generator inference // ===== VISUAL QUALITY METRICS ===== // PSNR: Peak Signal-to-Noise Ratio  $\text{MSE} = \frac{1}{HW} \sum_{i,j} (x_{\text{clean}}[i,j] - x_{\text{restored}}[i,j])^2$   $\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right)$  // In dB Append PSNR to PSNR_list // SSIM: Structural Similarity Index  $\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$  //  $\mu$ : mean,  $\sigma$ : variance,  $C_1, C_2$ : stabilization constants Append SSIM to SSIM_list // ===== TEXT READABILITY METRICS ===== // HTR Decoding  $y_{\text{restored}} \leftarrow R(x_{\text{restored}})$  // Get logits from recognizer  $t_{\text{pred}} \leftarrow \text{GreedyDecode}(y_{\text{restored}})$  // CTC greedy decoding (argmax) // CER: Character Error Rate (Levenshtein distance)  $\text{CER} = \frac{\text{EditDistance}(t_{\text{gt}}, t_{\text{pred}})}{\text{len}(t_{\text{gt}})} \times 100\%$  Append CER to CER_list // WER: Word Error Rate  $w_{\text{gt}} \leftarrow \text{SplitWords}(t_{\text{gt}})$  // Split into words  $w_{\text{pred}} \leftarrow \text{SplitWords}(t_{\text{pred}})$   $\text{WER} = \frac{\text{EditDistance}(w_{\text{gt}}, w_{\text{pred}})}{\text{len}(w_{\text{gt}})} \times 100\%$  Append WER to WER_list // ===== AGGREGATE STATISTICS =====  $\text{PSNR}_{\text{mean}} \leftarrow \text{mean}(\text{PSNR\_list}),$ 
```

$PSNR_{std} \leftarrow std(PSNR_list)$ $SSIM_{mean} \leftarrow mean(SSIM_list)$, $SSIM_{std} \leftarrow std(SSIM_list)$
 $CER_{mean} \leftarrow mean(CER_list)$, $CER_{std} \leftarrow std(CER_list)$ $WER_{mean} \leftarrow mean(WER_list)$,
 $WER_{std} \leftarrow std(WER_list)$ **return** Metrics dictionary with mean \pm std for all metrics

Catatan pada Algorithm ??:

- **CTC Greedy Decoding (Line 24):** Decoding strategy menggunakan argmax per timestep, kemudian menghapus blank tokens dan duplicate consecutive characters sesuai CTC algorithm.
- **Levenshtein Distance (Line 27, 33):** Edit distance dihitung dengan dynamic programming untuk mengukur jumlah insertions, deletions, dan substitutions yang diperlukan.
- **Reproducibility:** Setiap metrik dihitung dengan library standard (skimage untuk SSIM, editdistance untuk CER/WER) untuk memastikan konsistensi.

Validasi Hipotesis: Mengevaluasi hipotesis penelitian yang diajukan di Bab I dengan kriteria penolakan H_0 : $p\text{-value} < 0.05$ dari paired t-test.

III.2.6 Komunikasi (Tahap 6)

Hasil penelitian dan artefak yang dihasilkan dikomunikasikan kepada komunitas ilmiah dan praktisi untuk diseminasi pengetahuan dan mendorong reproduktifitas.

Penulisan Tesis: Mendokumentasikan seluruh proses penelitian dalam format tesis yang terstruktur sesuai dengan pedoman penulisan tesis.

Publikasi Ilmiah: Menyiapkan manuscript untuk publikasi di venue ilmiah dengan target journal IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) atau International Journal on Document Analysis and Recognition (IJ DAR).

Fase 1: Pembuatan Dataset Ground Truth untuk Recognizer (HTR) Proses pembuatan dataset melibatkan segmentasi dan anotasi halaman, ekstraksi pasangan data mentah, pembersihan dan binarisasi gambar, verifikasi dan kurasi final, serta pengemasan data.

Fase 2: Pembuatan Dataset Triplet Sintetis untuk Training GAN Setelah model Recognizer dilatih dan dibekukan, fase kedua berfokus pada pembuatan dataset GAN dengan data triplet (gambar terdegradasi, gambar bersih, label teks).

Pipeline degradasi sintetis mensimulasikan berbagai jenis degradasi yang umum ditemukan pada dokumen historis.

III.2.7 Strategi Pelatihan dan Validasi

Strategi pelatihan dirancang untuk memastikan konvergensi yang stabil dan keseimbangan antara *generator* dan diskriminator dalam arsitektur GAN. Konfigurasi parameter dipilih berdasarkan eksperimen pendahuluan dan praktik terbaik dalam literatur GAN.

Tabel 5: Konfigurasi parameter pelatihan *generator* dan diskriminator

Parameter	<i>Generator</i>	<i>Diskriminator</i>
<i>Optimizer</i>	Adam	Adam
<i>Learning Rate</i>	4×10^{-4}	4×10^{-4}
<i>Batch Size</i>	8 (full training), 2 (HPO)	8 (full training), 2 (HPO)
<i>Loss Function</i>	Multi-komponen (adv. + pixel + rec_feat)	<i>Binary cross-entropy</i>
<i>Bobot Loss</i>	Pixel: 120.0, RecFeat: 80.0, Adv: 2.5	Ditentukan melalui Bayesian Optimi
<i>CTC Loss</i>	Bobot 10.0 dengan annealing (0→10 epoch 1-3)	Gradien ke Generator, Recognizer fr
<i>Gradient Clipping</i>	clipnorm=1.0	clipnorm=1.0
<i>Label Smoothing</i>	-	Faktor 0.9
Target Akurasi	-	60-80% (ekuilibrium Nash)
Epoch Maksimum	200	200

Konfigurasi ini dirancang dengan pertimbangan berikut: (1) *learning rate* sama untuk *generator* dan diskriminator berdasarkan hasil eksperimen stabilitas, (2) bobot *loss* ditentukan melalui *Hyperparameter Optimization* sistematis menggunakan Bayesian Optimization (dijelaskan pada Bagian ??), (3) *gradient clipping* untuk stabilitas numerik, dan (4) *label smoothing* untuk regularisasi diskriminator.

Peran CTC Loss dalam Optimasi Generator: *CTC loss* dalam penelitian ini **digunakan sebagai komponen loss function Generator** dan gradiennya **di-backpropagate ke Generator** untuk mengoptimalkan keterbacaan teks secara langsung. Namun, perlu ditekankan bahwa **Recognizer tetap frozen** (bobot tidak di-update) sehingga gradien *CTC loss* hanya mengalir ke Generator, bukan ke Recognizer. Implementasi ini memungkinkan Generator belajar menghasilkan output yang lebih mudah dibaca oleh sistem HTR tanpa memodifikasi Recognizer itu sendiri.

Untuk mencegah destabilisasi training, *CTC loss* diterapkan dengan strategi *curriculum learning* melalui *annealing*: bobot dimulai dari 0.0 pada epoch 1-2 (*warmup phase* untuk stabilisasi visual), kemudian diaktifkan penuh menjadi 10.0

pada epoch 3 (*annealing phase*), dan tetap di 10.0 untuk epoch selanjutnya. Strategi ini terbukti efektif mencegah *mode collapse* dan destabilisasi gradien yang dapat terjadi jika CTC loss langsung aktif di awal training. Dengan demikian, arsitektur pelatihan menggunakan empat komponen *loss* untuk Generator: adversarial, pixel-wise, recognition feature, dan CTC loss (dengan curriculum learning via annealing).

Metodologi Recognition Feature Loss sebagai Guidance Berorientasi HTR:

Recognition Feature Loss merupakan komponen metodologis inovatif yang dirancang untuk memandu Generator agar menghasilkan output yang optimal untuk sistem HTR. Berbeda dengan *pixel-wise loss* yang mengukur perbedaan visual pada tingkat piksel, *recognition feature loss* mengukur kesamaan representasi fitur *internal* dari model HTR terbekukan, sehingga secara langsung mengoptimalkan keterbacaan teks tanpa mengorbankan kualitas visual.

Tabel 6: Spesifikasi metodologi implementasi recognition feature loss

Komponen Metodologi	Implementasi Teknis
Model Sumber Fitur	Recognizer terbekukan (<i>frozen weights</i>) yang telah dilatih pada dataset ground truth dengan arsitektur CNN-Transformer hybrid
Layer Ekstraksi	Layer proj_1n (output CNN backbone sebelum Transformer encoder), dengan dimensi [batch, 128, 512] untuk setiap sampel
Fungsi Loss	Mean Squared Error (MSE) antara feature maps dari gambar terestorasi vs. gambar ground truth: $\mathcal{L}_{\text{RecFeat}} = \frac{1}{N \times H \times D} \sum_{i=1}^N \sum_{h=1}^H \sum_{d=1}^D (f_{i,h,d}^{\text{restored}} - f_{i,h,d}^{\text{GT}})^2$ dengan N =batch size, H =sequence length (128), D =feature dim (512)
Forward Pass	Generator menghasilkan gambar restored → Recognizer frozen mengekstrak fitur dari restored dan GT → MSE dihitung antara kedua feature maps → Gradien di-backpropagate hanya ke Generator
Justifikasi Layer	Layer proj_1n dipilih karena merepresentasikan fitur visual tingkat tinggi yang telah dipelajari untuk tugas HTR, sebelum masuk ke sequence modeling Transformer
Perbedaan dengan CTC	CTC mengukur output akhir (prediksi teks) dengan discrete symbols, sedangkan RecFeat mengukur representasi internal dengan continuous features, memberikan signal pembelajaran yang lebih halus dan stabil untuk training GAN
Integrasi dalam Training	RecFeat loss dikombinasikan dengan adversarial dan pixel loss melalui weighted sum. Bobot optimal ditentukan melalui Bayesian Optimization (Bagian ??)

Alur Komputasi Recognition Feature Loss:

Implementasi komputasi dilakukan dalam langkah-langkah berikut untuk memastikan efisiensi dan stabilitas numerik:

1. **Ekstraksi Fitur GT:** Pada awal training batch, feature maps dari gambar ground truth diekstrak menggunakan recognizer frozen dan di-cache untuk menghindari komputasi berulang
2. **Forward Pass Generator:** Generator menerima gambar terdegradasi dan menghasilkan gambar restored
3. **Ekstraksi Fitur Restored:** Gambar restored dilewatkan ke recognizer frozen untuk mengekstrak feature maps dengan dimensi yang sama
4. **Perhitungan MSE:** MSE dihitung antara feature maps restored dan GT, dinormalisasi terhadap dimensi (batch \times sequence \times features)
5. **Weighted Integration:** RecFeat loss dikombinasikan dengan komponen loss lain: $\mathcal{L}_{\text{Total}} = w_{\text{adv}}\mathcal{L}_{\text{adv}} + w_{\text{pix}}\mathcal{L}_{\text{pix}} + w_{\text{rec}}\mathcal{L}_{\text{RecFeat}}$
6. **Backpropagation:** Gradien total di-backpropagate hanya ke Generator, recognizer tetap frozen sepanjang pelatihan GAN

Keunggulan Metodologi:

Pendekatan ini memiliki beberapa keunggulan metodologis dibandingkan alternatif:

- **Optimasi Langsung untuk HTR:** Dengan mencocokkan representasi internal HTR model, Generator belajar menghasilkan gambar yang secara eksplisit optimal untuk sistem HTR
- **Stabilitas Pelatihan:** MSE pada fitur kontinu lebih stabil daripada cross-entropy pada output diskrit, memberikan gradien yang lebih smooth
- **Gradien Informatif:** Feature-level matching memberikan signal pembelajaran yang lebih kaya daripada pixel-level matching, karena merepresentasikan fitur-fitur yang telah dipelajari untuk HTR
- **Keseimbangan Visual-Keterbacaan:** RecFeat loss mengoptimalkan keterbacaan tanpa mengorbankan kualitas visual, karena bekerja pada representasi internal yang kaya informasi
- **Kompatibilitas Multi-Objektif:** RecFeat loss kompatibel dengan pixel loss, adversarial loss, dan CTC loss dalam framework multi-objektif, memberikan pembelajaran yang komplementer

III.2.8 Metodologi Penentuan Bobot Loss Berbasis Bayesian Optimization

Penentuan bobot *loss* yang optimal merupakan aspek krusial dalam pelatihan GAN multi-objektif. Penelitian ini mengadopsi pendekatan ilmiah dan sistematis menggunakan *Bayesian Optimization* untuk menemukan kombinasi bobot *loss* yang memaksimalkan performa model.

Motivasi Metodologi: Pendekatan manual (*trial-and-error*) dalam penentuan bobot *loss* memiliki beberapa keterbatasan fundamental:

- **Tidak Sistematis:** Tidak ada jaminan konvergensi ke konfigurasi optimal
- **Tidak Reproducible:** Sulit direplikasi oleh peneliti lain
- **Tidak Efisien:** Membutuhkan banyak eksperimen manual yang mahal secara komputasi

• **Bias Subjektif:** Keputusan dipengaruhi oleh intuisi peneliti, bukan data empiris
Untuk mengatasi keterbatasan ini, penelitian ini mengimplementasikan *Hyperparameter Optimization* (HPO) menggunakan *Bayesian Optimization* dengan *Tree-structured Parzen Estimator* (TPE) *sampler* melalui *framework* Optuna (Akiba, Sano, Yanase, Ohta, & Koyama, 2019).

Tabel 7: *Search space hyperparameter optimization* untuk bobot *loss*

Parameter	Range	Step	Justifikasi
pixel_loss_weight	[50, 200]	10	Baseline L1 <i>loss</i> untuk rekonstruksi piksel. <i>Range</i> dipilih berdasarkan literatur GAN
rec_feat_loss_weight	[10, 100]	5	<i>Recognition feature loss</i> (berorientasi HTR). <i>Range</i> lebih sempit karena sensitivitas tinggi
adv_loss_weight	[1.0, 10.0]	0.5	<i>Adversarial loss</i> untuk realisme. <i>Upper bound</i> dibatasi untuk mencegah <i>mode collapse</i>

Desain Search Space:

Parameter Tetap (*Excluded from Optimization*): Parameter *ctc_loss_weight* ditetapkan pada nilai 10.0 dan tidak dioptimasi karena sensitivitasnya yang tinggi terhadap stabilitas training. Bobot ini diterapkan dengan strategi *curriculum learning* melalui *annealing* (epoch 1-2: weight=0.0 untuk warmup visual, epoch 3 onwards: weight=10.0 fully active) untuk mencegah destabilisasi awal training.

Fungsi Objektif: Fungsi objektif dirancang sebagai kombinasi berbobot dari metrik kualitas visual dan keterbacaan teks:

$$\text{maksimum: } \mathcal{O} = 0.4 \times \frac{PSNR}{50} + 0.4 \times SSIM - 0.1 \times CER - 0.1 \times WER \quad (1)$$

Bobot objektif dipilih berdasarkan:

- 40% PSNR (dinormalisasi): Kualitas rekonstruksi tingkat piksel
- 40% SSIM: Kesamaan struktural (kualitas perseptual)
- 10% CER: Penalti keterbacaan tingkat karakter
- 10% WER: Penalti keterbacaan tingkat kata

Tabel 8: Konfigurasi *Bayesian Optimization*

Parameter	Nilai	Justifikasi
<i>Sampler</i>	TPE (<i>Tree-structured Parzen Estimator</i>)	Efektif untuk <i>search space</i> diskrit dan campuran
Jumlah <i>Trial</i>	30	<i>Trade-off</i> antara eksplorasi dan biaya komputasi
<i>Epoch per Trial</i>	2	Evaluasi cepat untuk peringkat relatif
<i>Steps per Epoch</i>	30	Cukup untuk indikator konvergensi
<i>Batch Size</i>	2	Batasan memori dengan 2 GPU paralel
<i>Random Seed</i>	42	Reproduktif itas
<i>Timeout</i>	2 jam	Batasan praktis untuk iterasi cepat

Protokol *Bayesian Optimization*:

Strategi Validasi: Setiap *trial* dievaluasi menggunakan protokol yang konsisten:

1. **Pelatihan:** Model dilatih selama 2 *epoch* dengan konfigurasi bobot *loss* yang di-sample oleh TPE *sampler*
2. **Evaluasi:** Metrik (PSNR, SSIM, CER, WER) dihitung pada *validation set*
3. **Scoring:** Fungsi objektif dihitung dan dikembalikan ke *optimizer* Optuna
4. **Iterasi:** TPE *sampler* menggunakan hasil *trial* sebelumnya untuk menyarankan konfigurasi berikutnya

Hasil Hyperparameter Optimization: Setelah menjalankan 29 *trial* dengan protokol yang telah ditetapkan, konfigurasi optimal ditemukan dan divalidasi melalui analisis statistik. Detail lengkap hasil HPO, analisis kepentingan parameter, dan temuan kunci disajikan pada Bab V (Hasil dan Pembahasan).

Tabel 9: Konfigurasi bobot *loss* optimal dari *Bayesian Optimization*

Parameter	Nilai Optimal
pixel_loss_weight	120.0
rec_feat_loss_weight	80.0
adv_loss_weight	2.5
ctc_loss_weight	10.0 (dengan annealing 0→10)

Konfigurasi ini dipilih berdasarkan skor objektif tertinggi dari 29 *trial* yang dilakukan. Evaluasi lengkap hasil HPO disajikan pada Bab V.

Validasi Pelatihan Berdurasi Pendek: Meskipun HPO menggunakan pelatihan singkat (2 *epoch*), validitas peringkat relatif dikonfirmasi melalui:

- **Analisis Korelasi:** Konfigurasi terbaik pada HPO tetap terbaik pada pelatihan penuh (validasi empiris)
- **Konsistensi Pola:** Pola “*adv* tinggi → *collapse*” konsisten di semua durasi pelatihan
- **Dukungan Literatur:** Studi terkait menunjukkan 2-5 *epoch* cukup untuk peringkat *hyperparameter*

Protokol Reprodutifitas: Untuk memastikan penelitian dapat direproduksi:

- **Database Studi Optuna:** Tersimpan dalam *database* SQLite (`hpo_study.db`)
- **Pelacakan MLflow:** Semua 29 *trial* terekam dengan metrik, parameter, dan artefak
- **JSON Hasil:** Hasil lengkap tersimpan dalam `hpo_results_20251015_101057.json`
- **Log Eksekusi:** Log pelatihan lengkap tersimpan untuk audit

III.2.9 Strategi Optimasi Numerik

Optimasi numerik merupakan aspek kritis dalam pelatihan GAN dengan fungsi *loss* kompleks. Penelitian ini mengadopsi kebijakan *pure FP32 (floating-point 32-bit)* untuk semua komputasi pelatihan berdasarkan temuan empiris yang menunjukkan ketidakstabilan pada presisi campuran.

Tabel 10: Justifikasi ilmiah pemilihan presisi numerik FP32

Aspek	Masalah dengan FP16/Mixed	Solusi dengan Pure FP32
Stabilitas CTC Loss	Perhitungan <i>log-probability</i> dalam ruang logaritma menyebabkan <i>underflow</i> dengan FP16	Presisi penuh FP32 memastikan gradien CTC stabil saat di-backpropagate ke Generator
Keseimbangan Gradien	Perbedaan presisi antar komponen <i>loss</i> menyebabkan dominasi komponen tertentu	Presisi konsisten memastikan keseimbangan gradien antar komponen
Kualitas Konvergensi	Konvergensi tidak stabil dan hasil visual suboptimal	Konvergensi stabil dengan kualitas visual superior
Kompleksitas Komputasi	Trade-off: FP16 lebih cepat tetapi tidak stabil	Trade-off diterima untuk stabilitas dan kualitas

Teknik Stabilisasi Gradien: Selain pemilihan presisi numerik, diterapkan beberapa teknik stabilisasi gradien:

Tabel 11: Teknik stabilisasi gradien dan parameter

Teknik	Parameter	Tujuan
<i>Gradient Clipping</i>	clipnorm=1.0	Mencegah ledakan gradien dengan membatasi norma gradien maksimum
<i>CTC Loss Clipping</i>	Maksimum 50.0	Mencegah gradien spike dari CTC loss yang dapat destabilisasi Generator
<i>Label Smoothing</i>	Faktor 0.9	Regularisasi diskriminator untuk mencegah kepercayaan diri berlebihan
<i>Loss Normalization</i>	Terhadap panjang sekuens	Memastikan <i>loss</i> sebanding antar sampel dengan panjang berbeda

III.2.10 Desain Eksperimen Mode Diskriminator

Salah satu aspek novel dalam penelitian ini adalah Diskriminator Dual-Modal yang menerima input teks. Terdapat dua strategi alternatif untuk menyediakan input teks ini, masing-masing dengan kelebihan dan kekurangan. Eksperimen dirancang untuk membandingkan kedua mode secara sistematis.

Tabel 12: Perbandingan dua mode input teks diskriminator

Aspek	<i>Ground Truth Mode</i>	<i>Predicted Mode</i>
Sumber Teks	Transkripsi <i>ground truth</i>	Prediksi dari <i>recognizer</i> terbekukan
Stabilitas Pelatihan	Lebih stabil (teks konsisten)	Lebih menantang (teks mengandung kesalahan)
Realisme Deployment	Kurang realistis (tidak mencerminkan kondisi nyata)	Lebih realistis (simulasi kondisi <i>deployment</i>)
Konsistensi Signal	Sinyal supervisi konsisten	Sinyal supervisi bervariasi dengan kualitas gambar
Hipotesis	Memberikan <i>upper bound</i> performa	Memberikan performa yang lebih praktis

Tabel 13: Spesifikasi desain eksperimen perbandingan mode diskriminator

Komponen	Spesifikasi
Variabel Independen	Mode input teks diskriminator (<i>ground-truth</i> vs <i>predicted</i>)
Variabel Dependen	CER, WER (metrik utama); PSNR, SSIM (metrik sekunder); variansi <i>loss</i> (stabilitas)
Variabel Kontrol	<i>Learning rate</i> , <i>batch size</i> , bobot <i>loss</i> , arsitektur, <i>dataset</i> , pembagian latih/uji
Ukuran Sampel	100 <i>epoch</i> per mode dengan 5 <i>random seed</i> berbeda untuk robustitas statistik
Sistem Pelacakan	MLflow untuk pencatatan metrik, parameter, dan artefak untuk perbandingan objektif
Kriteria Keberhasilan	Penurunan CER minimal 30%, PSNR > 35 dB, SSIM > 0.90, <i>p-value</i> < 0.05

Desain Eksperimen Komparatif:

Hasil yang Diharapkan: Eksperimen ini diharapkan mengungkapkan pertukaran (*trade-off*) antara stabilitas pelatihan dan realisme *deployment*. Hasil akan dianalisis secara mendalam pada Bab V dengan fokus pada rekomendasi mode optimal untuk berbagai skenario aplikasi.

III.2.11 Strategi Pelatihan Recognizer

Arsitektur Hybrid CNN-Transformer: Model Recognizer menggunakan arsitektur hybrid yang menggabungkan CNN backbone untuk ekstraksi fitur visual dan Transformer encoder untuk modeling sequence-to-sequence. Pemilihan arsitektur ini didasarkan pada kemampuan CNN untuk menangkap fitur spasial dan kemampuan Transformer untuk memodelkan dependensi jarak jauh dalam sequence teks.

Tabel 14: Ringkasan arsitektur hybrid CNN-Transformer untuk HTR Recognizer

Komponen	Spesifikasi Utama	Justifikasi
CNN Backbone (Ekstraksi Fitur Visual)		
Arsitektur	7 conv layers + residual	Ekstraksi fitur hierarkis tanpa degradasi gradien
Output Shape	(64, 512) sequence	Dimensi optimal untuk Transformer input
Transformer Encoder (Sequence Modeling)		
Konfigurasi	6 layers, 8 heads, $d_{model}=512$	Kapasitas modeling dependensi jangka panjang
FFN Dimension	2048 ($4 \times d_{model}$)	Transformasi non-linear yang kaya
CTC Output Layer		
Vocab Size	95 karakter (charlist)	Mencakup huruf, angka, simbol, blank token
Decoder	CTC alignment	Alignment implisit tanpa segmentasi eksplisit
Total Parameters: ~24M Model Size: ~96 MB		

Catatan: Detail lengkap arsitektur layer-by-layer disajikan pada Lampiran A.

Tabel 15: Ringkasan konfigurasi pelatihan HTR Recognizer

Kategori	Konfigurasi	Justifikasi
Optimisasi		
<i>Optimizer</i>	AdamW	Adaptive learning rate dengan weight decay decouple
<i>Learning Rate</i>	3×10^{-4} + Cosine annealing	Baseline optimal untuk Transformer (Vaswani et al.)
<i>Batch Size</i>	32	Trade-off memori GPU vs stabilitas gradien
Regularisasi		
<i>Dropout</i>	0.20	Regularisasi transformer layers
<i>Label Smoothing</i>	$\epsilon=0.1$	Soft targets untuk CTC loss
Data Augmentation	Brightness, contrast, noise	Simulasi variasi degradasi
Presisi & Stabilitas		
Precision	FP32 (pure)	Stabilitas CTC loss dalam log-space
Gradient Clipping	clipnorm=1.0	Mencegah exploding gradient
Kriteria Konvergensi		
<i>Early Stopping</i>	Patience 15 epoch	Monitoring validation CER
<i>Target CER</i>	< 15%	Baseline evaluasi untuk GAN training

Catatan: Detail lengkap konfigurasi optimizer, LR schedule, dan augmentation disajikan pada Lampiran B.

Tabel 16: Target performa HTR Recognizer untuk evaluasi GAN

Metrik	Target	Justifikasi
Akurasi Teks		
<i>Character Error Rate (CER)</i>	< 15%	Baseline evaluasi GAN; threshold usability HTR historis
<i>Word Error Rate (WER)</i>	< 25%	Standar praktis untuk aplikasi arsip digital
Stabilitas & Efisiensi		
Standar Deviasi CER	< 2%	Reproducibility untuk penelitian rigorous
<i>Train-Val Gap</i>	< 5% CER	Indikator generalisasi (mencegah overfitting)
<i>Inference Time</i>	< 100ms/image	Real-time processing (GPU RTX 4090)

Catatan: Perbandingan dengan baseline state-of-the-art (Yousef et al. 2020) disajikan pada Bab II (Tinjauan Pustaka). Detail efisiensi komputasi disajikan pada Lampiran C.

III.2.12 Implementasi CTC Loss sebagai Metrik Evaluasi

Connectionist Temporal Classification (CTC) loss berfungsi sebagai metrik evaluasi keterbacaan teks dalam *framework* GAN-HTR. Perlu ditekankan bahwa CTC loss **TIDAK di-backpropagate ke Generator**, melainkan digunakan

sebagai *monitoring metric* untuk mengukur kualitas keterbacaan hasil restorasi. Implementasi CTC *loss* memerlukan perhatian khusus terhadap stabilitas numerik agar metrik monitoring akurat dan konsisten.

Tabel 17: Teknik stabilisasi numerik untuk CTC *loss*

Teknik	Implementasi	Tujuan
<i>Label Smoothing</i>	Faktor $\epsilon = 0.1$ diterapkan pada label	Meningkatkan akurasi metrik evaluasi dan mencegah bias pada label keras
<i>Log-Probability Clipping</i>	Batasan nilai minimum/maksimum dalam ruang logaritma	Mencegah <i>underflow</i> untuk perhitungan metrik yang stabil
<i>Loss Normalization</i>	Normalisasi terhadap panjang sekuens	Memastikan konsistensi metrik antar sampel berbeda panjang
Kontrol Presisi	FP32 untuk komputasi CTC	Akurasi maksimal dalam perhitungan metrik monitoring log-space
<i>Value Clipping</i>	Maksimum 50.0 untuk logging	Mencegah nilai ekstrem dalam pencatatan metrik

Strategi Decoding: Pada fase inferensi, prediksi CTC perlu di-*decode* menjadi sekuens karakter. Penelitian ini mengimplementasikan tiga strategi *decoding* dengan karakteristik berbeda:

Tabel 18: Strategi *decoding* CTC dan karakteristiknya

Strategi	Karakteristik	Kecepatan	Akurasi
<i>Greedy Decoding</i>	Memilih karakter dengan probabilitas tertinggi	Sangat cepat	Baik
<i>Beam Search</i>	Eksplorasi <i>beam width</i> = 10 jalur terbaik	Sedang	Sangat baik
CTC <i>Beam Search</i> dengan LM	Integrasi <i>language model</i> untuk konteks	Lambat	Terbaik

Pemilihan strategi *decoding* disesuaikan dengan kebutuhan: *greedy decoding* untuk pemrosesan *real-time*, *beam search* untuk keseimbangan kecepatan-akurasi, dan CTC *beam search* dengan *language model* untuk akurasi maksimal.

III.2.13 Sistem Pelacakan Eksperimen

Setiap eksperimen dilacak menggunakan MLflow dengan metadata yang meliputi hyperparameters, metrics, artifacts, dan Git commit untuk reproduktifitas kode.

III.2.14 Evaluasi (Tahap 5)

Tahap ini mengukur performa artefak secara objektif terhadap tujuan yang telah didefinisikan dan memvalidasi hipotesis penelitian melalui evaluasi kualitas visual dan keterbacaan teks.

Metode Evaluasi Evaluasi *framework* GAN-HTR dilakukan menggunakan metrik ganda yang mengukur baik kualitas visual maupun keterbacaan teks. Pendekatan evaluasi multi-dimensi ini memastikan bahwa restorasi tidak hanya meningkatkan estetika visual tetapi juga meningkatkan performa HTR secara signifikan.

Tabel 19: Ringkasan metrik evaluasi dan target performa

Kategori	Metrik	Target	Interpretasi
Kualitas Visual	PSNR	$> 35 \text{ dB}$	Rasio sinyal terhadap derau dalam skala logaritma
	SSIM	> 0.95	Kesamaan struktural perseptual
Keterbacaan Teks	CER	Penurunan $\geq 25\%$	Rasio kesalahan pada tingkat karakter
	WER	Penurunan $\geq 25\%$	Rasio kesalahan pada tingkat kata
Stabilitas	Variansi <i>Loss</i>	< 0.05	Konsistensi pelatihan dan konvergensi
Efisiensi	Waktu Inferensi	$< 15 \text{ detik}$	Kecepatan pemrosesan per gambar

III.2.15 Metrik Kualitas Visual

PSNR (*Peak Signal to Noise Ratio*): PSNR mengukur rasio antara sinyal maksimum yang mungkin dengan derau yang merusak kualitas gambar. Metrik ini dihitung menggunakan formula:

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (2)$$

di mana MAX_I adalah nilai maksimum piksel (255 untuk gambar 8-bit) dan MSE adalah *Mean Squared Error* antara gambar *ground truth* dan gambar terestorasi. Nilai PSNR lebih tinggi mengindikasikan kualitas rekonstruksi lebih baik, dengan target $> 35 \text{ dB}$ yang merupakan standar untuk restorasi dokumen berkualitas tinggi.

SSIM (*Structural Similarity Index*): SSIM mengukur kesamaan struktural yang lebih selaras dengan persepsi visual manusia dibandingkan PSNR. Formula SSIM:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3)$$

di mana μ adalah rata-rata, σ adalah deviasi standar, σ_{xy} adalah kovarians, dan C_1, C_2 adalah konstanta stabilitas numerik. SSIM menghasilkan nilai antara 0 dan 1, dengan target > 0.95 mengindikasikan kesamaan struktural yang sangat tinggi.

III.2.16 Metrik Keterbacaan Teks

Metrik keterbacaan teks merupakan evaluasi utama yang membedakan penelitian ini dari metode restorasi dokumen konvensional yang hanya fokus pada kualitas visual.

CER (Character Error Rate): CER mengukur rasio kesalahan pengenalan pada tingkat karakter, dihitung menggunakan formula:

$$CER = \frac{S + D + I}{N} \times 100\% \quad (4)$$

di mana S adalah substitusi (karakter diganti), D adalah penghapusan (karakter hilang), I adalah penyisipan (karakter tambahan), dan N adalah jumlah karakter *ground truth*.

WER (Word Error Rate): WER mengukur kesalahan pada tingkat kata dengan formula analog:

$$WER = \frac{S_w + D_w + I_w}{N_w} \times 100\% \quad (5)$$

dengan notasi serupa pada tingkat kata.

Tabel 20: Pipeline perhitungan metrik keterbacaan teks

Tahap	Proses	Detail Implementasi
1. Pengenalan Teks	Gambar terestorasi → <i>Recognizer</i> terbekukan	Model dengan bobot dibekukan memastikan konsistensi evaluasi
2. <i>CTC Decoding</i>	Probabilitas → Sekuens karakter	<i>Beam search</i> dengan <i>width</i> 10 untuk akurasi optimal
3. Pascapemrosesan	Pembersihan <i>output</i> CTC	Penghapusan <i>blank token</i> dan penanganan duplikasi
4. Perhitungan Metrik	Algoritma <i>Levenshtein distance</i>	Komputasi operasi penyuntingan minimum (S, D, I)
5. Agregasi	Rata-rata pada <i>validation set</i>	Perhitungan mean, deviasi standar, dan interval kepercayaan

Kriteria Keberhasilan: Metrik keterbacaan dievaluasi dengan tiga kriteria:

Tabel 21: Kriteria keberhasilan metrik keterbacaan

Kriteria	Target	Justifikasi
Penurunan CER	$\geq 25\%$	Peningkatan substansial dibandingkan <i>baseline</i> terdegradasi, sesuai target penelitian
Penurunan WER	$\geq 25\%$	Perbaikan keterbacaan pada tingkat kata untuk aplikasi praktis
Konsistensi	Deviasi standar $< 5\%$	Performa stabil di seluruh sampel validasi

III.2.17 Analisis Statistik

Validasi signifikansi peningkatan dilakukan menggunakan rangkaian pengujian statistik yang ketat untuk memastikan bahwa perbaikan yang diamati bukan hasil kebetulan. Pendekatan ini mengikuti standar metodologi penelitian eksperimental dalam bidang pembelajaran mesin.

Tabel 22: Metode pengujian statistik dan tujuannya

Metode	Implementasi	Tujuan
Uji Normalitas	Shapiro-Wilk	Validasi asumsi distribusi normal sebelum <i>t-test</i> parametrik
<i>Paired t-test</i>	<code>scipy.stats.ttest_rel()</code>	Membandingkan metrik berpasangan dengan $\alpha = 0.05$
Uji Wilcoxon	<code>scipy.stats.wilcoxon()</code>	Alternatif non-parametrik untuk data tidak normal
Ukuran Efek Cohen's d	$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{pooled}}$	Mengukur besaran praktis peningkatan
Koreksi Bonferroni	$\alpha_{corrected} = \frac{\alpha}{n}$	Mengendalikan <i>family-wise error rate</i>
Benjamini-Hochberg	<code>multipletests()</code>	Mengendalikan <i>false discovery rate</i>
Interval Kepercayaan	<i>Bootstrap</i> 1000 iterasi	Estimasi interval kepercayaan 95% yang kokoh

III.2.18 Validasi Hipotesis

H₀ (Hipotesis Nol): Tidak ada perbedaan signifikan antara metode yang diusulkan dan *baseline* dalam hal keterbacaan teks (CER).

H₁ (Hipotesis Alternatif): Metode yang diusulkan menghasilkan CER yang signifikan lebih rendah dibandingkan *baseline* dengan tingkat kepercayaan 95%.

Kriteria penolakan H₀: p-value < 0.05 dari paired t-test.

Validasi Hipotesis dan Perbandingan Metode Validasi hipotesis dilakukan melalui perbandingan sistematis dengan metode *state-of-the-art* dan studi ablasi untuk mengisolasi kontribusi komponen individual. Pendekatan ini memastikan

bahwa klaim keunggulan *framework* yang diusulkan didukung oleh bukti empiris yang kuat.

Metode Perbandingan dengan *State-of-the-Art*: Untuk memvalidasi keunggulan *framework* GAN-HTR yang diusulkan, dilakukan perbandingan komprehensif dengan tiga metode terkemuka dalam restorasi dokumen:

Tabel 23: Metode *baseline* untuk perbandingan dan karakteristiknya

Metode	Karakteristik Utama	Kelebihan/Kekurangan
DE-GAN	<i>Document Enhancement GAN</i> dengan diskriminator konvensional	Kelebihan: Restorasi visual baik. Kekurangan: Tidak dioptimasi untuk HTR
TEXT-DIAE	<i>Denoising Autoencoder</i> dengan <i>CTC loss</i>	Kelebihan: Fokus keterbacaan. Kekurangan: Arsitektur non-adversarial lebih lemah
CycleGAN	Translasi gambar-ke-gambar tanpa data berpasangan	Kelebihan: Tidak perlu data berpasangan. Kekurangan: Kualitas inkonsisten

Protokol Perbandingan yang Adil: Untuk memastikan perbandingan yang objektif dan dapat direproduksi, diterapkan protokol eksperimen yang ketat:

Tabel 24: Protokol perbandingan metode

Aspek	Spesifikasi
<i>Dataset</i>	Identik untuk semua metode dengan pembagian latih (80%)/validasi (10%)/uji (10%) yang sama
Metrik Evaluasi	Multi-dimensi: PSNR, SSIM (kualitas visual); CER, WER (keterbacaan teks)
Reprodusibilitas	5 eksperimen dengan <i>random seed</i> berbeda untuk robustitas statistik
<i>Hardware</i>	Identik untuk semua metode (GPU NVIDIA RTX A4000, 32 GB RAM)
Pengujian Statistik	<i>Paired t-test</i> dengan $\alpha = 0.05$ dan ukuran efek Cohen's d
Kriteria Keberhasilan	PSNR > 35 dB, SSIM > 0.90, CER penurunan $\geq 30\%$, <i>p-value</i> < 0.05

III.2.19 Desain Ablation Study

Tujuan Ablation Study: Untuk mengisolasi kontribusi masing-masing komponen dalam *framework* GAN-HTR dan membuktikan superioritas dari kombinasi komponen yang diusulkan:

Tabel 25: Konfigurasi eksperimen ablation study

Konfigurasi	RecFeat Loss	CTC Monitoring	Dual-Modal Disc.	HPO Loss Weigh
Baseline (Pixel + Adv only)	×	×	×	×
+ Recognition Feature Loss	✓	×	×	×
+ CTC Monitoring	✓	✓	×	×
+ Dual-Modal Discriminator	✓	✓	✓	×
Full Framework (Proposed)	✓	✓	✓	✓

Justifikasi Desain Ablation: Desain ablation study disusun secara bertahap untuk mengisolasi kontribusi setiap komponen inovatif:

- **Baseline:** GAN konvensional dengan pixel loss dan adversarial loss saja, tanpa guidance berorientasi HTR
- **+ RecFeat Loss:** Menambahkan recognition feature loss untuk mengukur dampak guidance dari representasi internal recognizer frozen
- **+ CTC Monitoring:** Menambahkan CTC loss sebagai metrik monitoring (tidak di-backprop) untuk pelacakan keterbacaan
- **+ Dual-Modal Discriminator:** Menambahkan input teks ke discriminator untuk realisme konten tekstual
- **Full Framework:** Kombinasi lengkap dengan HPO-tuned loss weights untuk optimasi multi-objektif

Variabel Kontrol:

- **Dataset:** Sama untuk semua ablation configurations
- **Training Parameters:** Learning rate, batch size, epochs
- **Evaluation Metrics:** PSNR, SSIM, CER, WER
- **Random Seeds:** Sama untuk reproduktifitas

Hasil yang Diharapkan dan Interpretasinya: Berdasarkan eksperimen pendahuluan dan temuan literatur, diasumsikan setiap komponen memberikan kontribusi terukur:

Tabel 26: Kontribusi yang diharapkan dari setiap komponen

Komponen	Kontribusi CER	Interpretasi
<i>Recognition Feature Loss</i>	12-18% perbaikan	Guidance dari representasi internal HTR model untuk optimasi keterbacaan
<i>CTC Monitoring</i>	3-5% tambahan	Pelacakan keterbacaan tanpa backprop, membantu hyperparameter tuning
Diskriminator Dual-Modal	8-12% perbaikan	Evaluasi simultan kualitas visual dan konsistensi konten tekstual
<i>HPO Loss Weight</i>	5-8% perbaikan	Keseimbangan optimal antar komponen loss multi-objektif
Efek Kombinasi	30-40% total	Sinergi melebihi jumlah individual

Jika efek kombinasi secara signifikan melebihi jumlah kontribusi individual, ini mengindikasikan adanya sinergi positif antar komponen yang memvalidasi desain *framework* holistik. *Recognition feature loss* diharapkan memberikan kontribusi terbesar karena secara langsung mengoptimalkan representasi yang relevan untuk HTR.

III.2.20 Statistical Testing Framework

Hipotesis Statistik:

- $H_0: \mu_{proposed} = \mu_{baseline}$ (no significant difference)
- $H_1: \mu_{proposed} < \mu_{baseline}$ (proposed is significantly better)

Statistical Tests:

1. **Uji Normalitas:** Shapiro-Wilk untuk menguji normalitas distribusi
2. **Paired t-test:** Untuk data terdistribusi normal
3. **Wilcoxon Signed-Rank:** Untuk data tidak normal
4. **Ukuran Efek:** Cohen's d untuk besaran peningkatan

Multiple Comparison Correction:

- **Bonferroni Correction:** $\alpha_{corrected} = \frac{\alpha}{n}$ (dimana n = jumlah comparisons)
- **False Discovery Rate:** Benjamini-Hochberg procedure

Confidence Intervals:

- **CI 95%:** Untuk perbedaan rata-rata antara metode
- **Bootstrap CI:** 1000 pengambilan sampel ulang untuk estimasi yang kokoh

III.2.21 Cross-Validation Strategy

K-Fold Cross Validation:

- **K = 5:** Validasi silang *5-fold* untuk estimasi performa yang kokoh
- **Pengambilan Sampel Berlapis:** Menjaga distribusi kelas di setiap *fold*
- **Nested CV:** Putaran dalam untuk *hyperparameter tuning*, putaran luar untuk evaluasi

Temporal Validation:

- **Pembagian Berbasis Waktu:** Pelatihan pada dokumen lama, pengujian pada dokumen baru
- **Uji Generalisasi:** Mengukur kemampuan generalisasi ke domain yang belum pernah dilihat

III.2.22 Komunikasi (Tahap 6)

Hasil penelitian dan artefak yang dihasilkan dikomunikasikan kepada komunitas ilmiah dan praktisi untuk diseminasi pengetahuan dan mendorong reproduktifitas.

Penulisan Tesis: Mendokumentasikan seluruh proses penelitian dalam format tesis yang terstruktur sesuai dengan pedoman penulisan tesis.

Publikasi Ilmiah: Menyiapkan manuscript untuk publikasi di venue ilmiah dengan target journal IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) atau International Journal on Document Analysis and Recognition (IJДАР).

Open Source dan Reproduktifitas: Membagikan artefak penelitian untuk transparansi dan kemajuan sains melalui GitHub repository dan MLflow artifacts.

III.3 Manajemen Penelitian

Manajemen penelitian yang efektif memastikan proses pengembangan yang sistematis, telusur, dan dapat direproduksi. Penelitian ini menerapkan prinsip-prinsip manajemen yang telah terbukti dalam pengembangan sistem pembelajaran mesin kompleks.

III.3.1 Prinsip Pengembangan

Penelitian ini mengikuti lima prinsip fundamental untuk memastikan kualitas dan integritas:

Tabel 27: Prinsip pengembangan dan implementasinya

Prinsip	Definisi	Implementasi Praktis
<i>Don't Repeat Yourself</i> (DRY)	Kode modular dan dapat digunakan ulang	Fungsi umum diabstraksi ke modul terpisah untuk menghindari duplikasi
Pengujian Awal (<i>Smoke Test</i>)	Validasi <i>pipeline</i> sebelum skala penuh	Uji dengan <i>dataset</i> kecil (100 sampel) dan 5 <i>epoch</i> sebelum pelatihan penuh
Dokumentasi Terdorong	Pencatatan sistematis semua temuan	Setiap eksperimen didokumentasikan dalam <i>logbook</i> dengan stempel waktu
Pelatihan Keadaan Bersih	Menghindari kesalahan terbawa	Hapus <i>checkpoint</i> bermasalah dan validasi integritas data sebelum pelatihan
Inspeksi Mendalam	Pemahaman akar masalah	Analisis mendalam kode dan data sebelum membuat perubahan signifikan

III.3.2 Sistem Dokumentasi

Sistem dokumentasi terstruktur memastikan semua aspek penelitian terekam dan dapat dilacak:

Tabel 28: Struktur sistem dokumentasi penelitian

Komponen	Konten	Tujuan
<i>Logbook</i> Harian	Tanggal, eksperimen, hasil, analisis	Pelacakan kronologis kemajuan penelitian
Laporan Eksperimen	Hipotesis, metodologi, hasil, kesimpulan	Dokumentasi formal untuk eksperimen kunci
Catatan Teknis	Solusi <i>bug</i> , optimasi, konfigurasi	Transfer pengetahuan dan pencegahan masalah berulang
Metrik MLflow	Parameter, metrik, artefak, Git <i>commit</i>	Reproduktifitas dan perbandingan objektif
Dokumentasi Kode	<i>Docstring</i> , komentar, README	Pemahaman dan pemeliharaan kode

Setiap dokumen mengikuti templat standar yang meliputi: (1) tanggal dan identifikasi, (2) tujuan atau masalah, (3) metode atau pendekatan, (4) hasil dan observasi, (5) kesimpulan dan pembelajaran, (6) tindakan selanjutnya.