BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

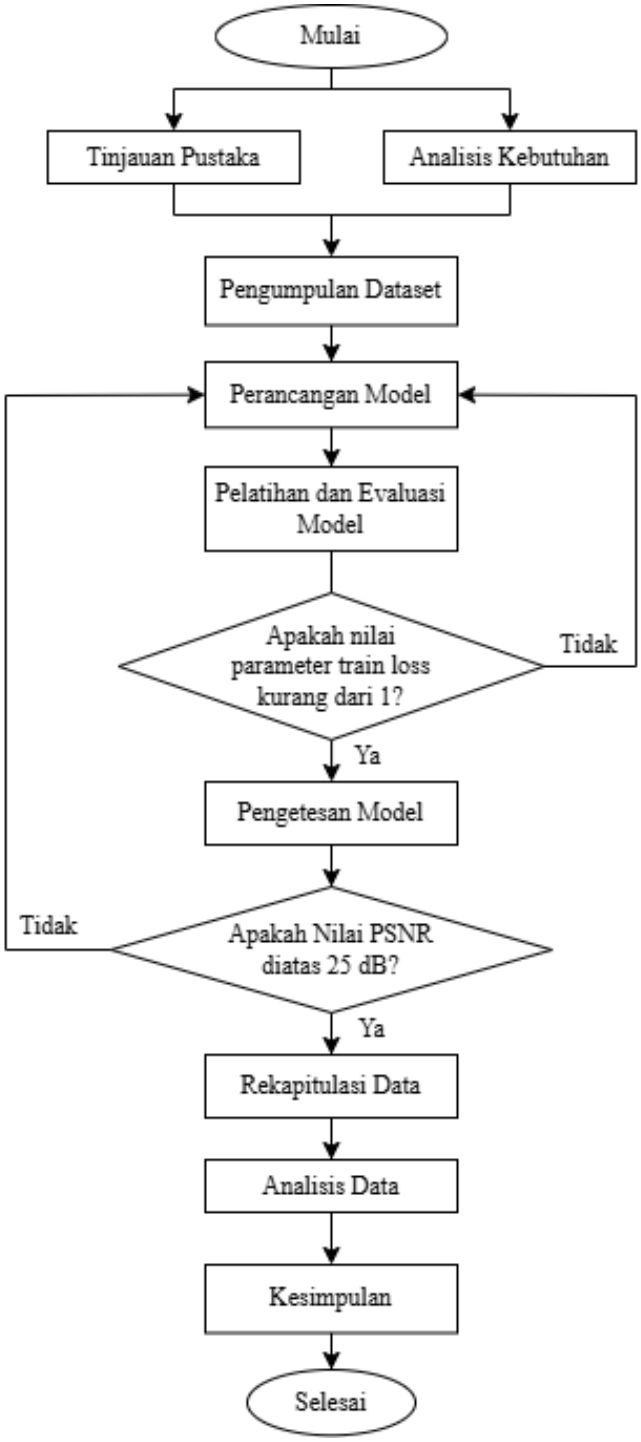
Penelitian ini menggunakan model penelitian ADDIE (*Analisys, Design, Development, Implementation, Evaluation*), yang berfungsi sebagai metodologi sistematis untuk desain, pengembangan, dan evaluasi. metode ini dipilih karena kemampuan beradaptasinya dalam memastikan bahwa setiap fase pengembangan model CNN untuk kompresi dan rekonstruksi gambar dilakukan dengan cara yang terstruktur dan dapat diskalakan. Selama fase Analisis, persyaratan yang berkaitan dengan efisiensi kompresi gambar dibedakan untuk lingkungan komunikasi yang ditandai dengan *bandwidth* terbatas. Selain itu, fase Desain mencakup perumusan arsitektur CNN yang optimal, dengan mempertimbangkan integrasi algoritma kompresi tambahan. Pada fase Pengembangan, implementasi model dijalankan melalui pelatihan dengan kumpulan data yang disiapkan, sambil melakukan evaluasi untuk mengukur kinerjanya. Fase Implementasi melibatkan simulasi transmisi data melintasi saluran komunikasi *bandwidth* rendah untuk mengevaluasi efisiensi model dalam kondisi yang meniru lingkungan dunia nyata. Pada akhirnya, fase Evaluasi dilakukan melalui analisis komprehensif kinerja model berdasarkan metrik evaluasi seperti PSNR, dimensi *latent space*, dan estimasi durasi transmisi, dengan tujuan menentukan konfigurasi optimal yang selaras dengan persyaratan penelitian.

3.2 Alur Penelitian

Diagram alir penelitian berikut menyajikan representasi proses yang dilakukan pada penelitian ini, yang menggunakan skema ADDIE. Diagram ini menjelaskan langkah-lanhkah mulai dari analisis kebutuhan dan tinjauan literatur hingga perancangan, pegujian dan evaluasi model. Setiap langkah dibuat dengan memastikan bahwa model CNN berbasis arsitektur *encoder-decoder* yang direkayasa memenuhi persyaratan untuk efisiensi transmisi gambar pada saluran komunikasi bandwidth rendah, seperti LoRa, tanpa berkorban banyak terkait

Muhammad Iqbal Fauzan, 2025
OPTIMALISASI LATENT SPACE PADA CNN ENCODER-DECODER UNTUK KOMPRESI DAN EFISIENSI TRANSMISI GAMBAR PADA KENDARAAN LISTRIK OTONOM
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

kualitas rekonstruksi gambar. Diagram ini tidak hanya menggambarkan lintasan penelitian tetapi juga mendukung metodologi berulang yang digunakan untuk mencapai hasil optimal melalui berbagai proses evaluatif dan pengujian model.



Gambar 3.1 Alur Penelitian

3.2.1 Tinjauan Pustaka dan Analisis Kebutuhan

Tahap ini merupakan langkah awal yang penting dalam metodologi penelitian, yang bertujuan untuk memastikan bahwa semua kebutuhan penelitian yang diperlukan dapat diidentifikasi dengan baik. Proses analitis dimulai dengan melakukan tinjauan pustaka untuk memahami kemajuan terbaru dalam teknologi kompresi gambar dan arsitektur CNN berbasis *encoder-decoder*. Selanjutnya, analisis persyaratan teknis dilakukan untuk merumuskan model di lingkungan komunikasi khusus untuk kendaraan otonom. Hal ini juga terkait dengan karakteristik dari saluran komunikasi LoRa, dengan batasan *bandwidth* sekitar 37,5 Kbps, maka ada syarat yang dibuat:

1. Model harus memiliki kemampuan untuk mengompres gambar input dimensi 80 × 80 piksel ke dalam *latent space* yang diminimalkan secara substanțial untuk mengoptimalkan efisiensi transmisi.
2. Integritas kualitas rekonstruksi gambar harus ditegakkan, dengan nilai target PSNR target yang ditentukan melebihi 25 dB.
3. Optimalisasi algoritma tambahan untuk kompresi *latent space*, seperti Blosc dan Zlib, sangat penting untuk memastikan pelestarian integritas data selama tahap transmisi.
4. Evaluasi efektivitas model harus mencakup analisis perubahan antara dimensi *latent space*, kualitas rekonstruksi, dan durasi transmisi.

Penilaian kebutuhan ini selanjutnya memerlukan identifikasi hambatan teknis selama tahap pelatihan dan potensi distorsi gambar dalam hasil rekonstruksi saat menggunakan kompresi Blosc.

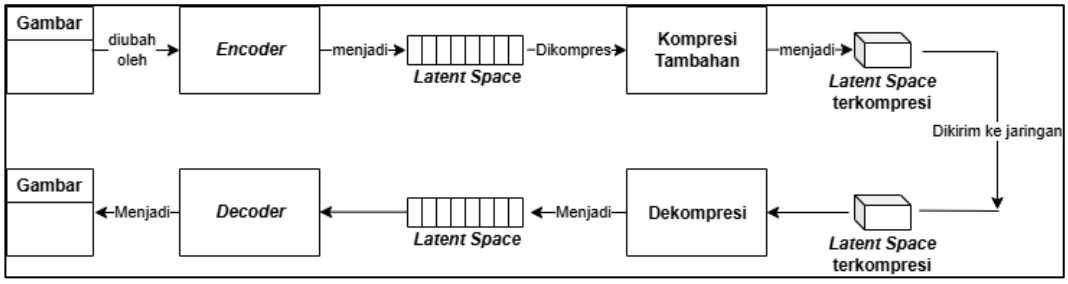
3.2.2 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari kumpulan gambar dengan resolusi 240x240 piksel, yang dikumpulkan dari proses perekaman suasana jalan dan lingkungan disekitarnya. Dataset ini dibagi menjadi dua bagian, yaitu data pelatihan dan data pengujian, untuk digunakan dalam proses pelatihan dan validasi model CNN *encoder-decoder*. Proses praproses data meliputi normalisasi nilai

piksel, penyesuaian ukuran, dan augmentasi data jika diperlukan untuk memperkaya variasi citra dalam pelatihan.

3.2.3 Perancangan Model

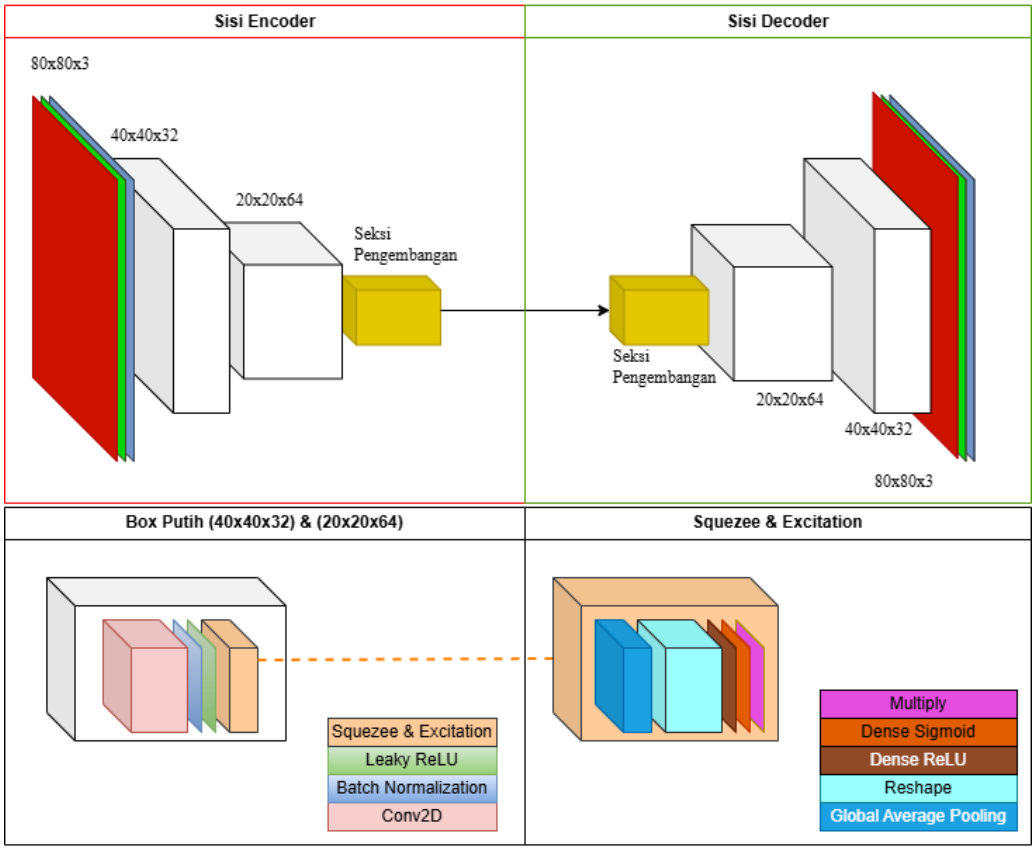
Desain arsitektur model CNN *encoder-decoder* yang dirancang dalam penelitian ini bertujuan untuk mengurangi dampak dari tantangan transmisi gambar pada kanal komunikasi dengan bandwidth terbatas. Sebelum menjelaskan arsitektur internal model secara lebih rinci, perlu disampaikan terlebih dahulu gambaran umum mengenai alur sistem kompresi dan rekonstruksi yang diusulkan dalam penelitian ini. Gambar berikut menyajikan representasi umum dari proses yang berlangsung mulai dari input citra, transformasi oleh model, hingga proses kompresi tambahan dan pengiriman data.



Gambar 3.2 Gambaran umum alur sistem

Gambar 3.2 tersebut memperlihatkan alur sistem secara keseluruhan, yang dimulai dari proses pengubahan citra input oleh modul *encoder* menjadi representasi berdimensi lebih rendah yang disebut sebagai *latent space*. Representasi ini kemudian diproses lebih lanjut oleh algoritma kompresi tambahan Blosc atau Zlib, menghasilkan data *latent* yang lebih ringkas dalam bentuk latent space terkompresi. Data hasil kompresi inilah yang kemudian dikirimkan melalui jaringan, dengan asumsi kanal komunikasi yang memiliki bitrate terbatas. Setelah diterima oleh sisi penerima, data *latent* yang telah dikompresi terlebih dahulu didekompresi untuk mengembalikan bentuk *latent space* aslinya. Proses ini dilanjutkan dengan pengubahan kembali *latent space* menjadi citra oleh modul *decoder*, yang diharapkan dapat menghasilkan citra rekonstruksi dengan kualitas visual yang mendekati citra aslinya.

Arsitektur model yang digunakan secara garis besar terbagi menjadi komponen utama *encoder*, dan *decoder* yang dilengkapi dengan blok pendukung. Gambar 3.3 berikut berisi komposisi dari arsitektur yang digunakan.



Gambar 3.3 Arsitektur Model Dasar *Encoder-Decoder*

Encoder bertanggung jawab untuk mengekstraksi fitur penting dari gambar input berukuran 80x80x3. Pada tahap awal, gambar diolah dengan lapisan konvolusi yang menghasilkan keluaran berukuran 40x40x32, dilanjutkan dengan proses konvolusi kedua yang menghasilkan keluaran berukuran 20x20x64. Lapisan konvolusi ini berfungsi untuk menangkap pola spasial dan tekstur dari gambar masukan. Filter yang digunakan dalam lapisan pertama berjumlah 32, sementara lapisan kedua menggunakan 64 filter untuk memperkaya informasi fitur. Selain lapisan konvolusi, proses *encoder* ini turut dilengkapi dengan komponen normalisasi dan aktivasi untuk meningkatkan stabilitas pelatihan dan efisiensi komputasi. *Batch normalization* diterapkan guna menstabilkan distribusi aktivasi selama pelatihan.

Komponen box putih dalam diagram mewakili blok konvolusi yang memiliki susunan elemen konvolusi, *batch normalization*, aktivasi Leaky ReLU, dan *Squeeze-and-Excitation* (SE) *block*. Blok SE ini diawali dengan Global Average Pooling yang menghitung rata-rata nilai setiap kanal fitur untuk mendapatkan informasi global. Selanjutnya, Dense layer dengan aktivasi ReLU mengolah informasi global tersebut agar lebih representatif, yang kemudian diteruskan ke Dense layer dengan aktivasi sigmoid untuk menghasilkan bobot perhatian adaptif. Bobot tersebut kemudian diterapkan pada fitur awal melalui proses *reshaping* dan multiplikasi untuk memperkuat saluran yang penting dan melemahkan yang kurang signifikan.

Segmen pengembangan yang ditunjukkan oleh blok berwarna kuning dalam diagram berfungsi sebagai *bottleneck* yang memampatkan representasi fitur menjadi bentuk yang lebih kompak untuk mendukung efisiensi transmisi data. Segmen ini menjadi titik penghubung antara *encoder* dan *decoder* yang menjaga agar informasi inti tidak hilang selama proses *encoding* dan *decoding*.

Decoder bertugas merekonstruksi gambar dari *latemt space* yang telah dikompresi oleh *encoder*. Tahapan pertama pada *decoder* melibatkan lapisan konvolusi yang mengubah fitur berukuran 20x20x64 menjadi 40x40x32, sebelum akhirnya mengembalikan dimensi gambar ke ukuran asli yaitu 80x80x3 dengan berusaha meminimalisr dampak dari proses pemampatan yang ekstrim.

Untuk memperoleh konfigurasi optimal dalam proses kompresi dan rekonstruksi citra, dilakukan serangkaian eksperimen dengan mengeksplorasi berbagai variasi desain pada segmen pengembangan dalam arsitektur *encoder-decoder*. Segmen ini berfungsi sebagai tahap pengubahan yang menentukan efisiensi representasi fitur sebelum diteruskan ke tahap *decoding*. Dalam penelitian ini, dikembangkan 20 model dengan variasi struktur pada segmen pengembangan, yang bertujuan untuk mengevaluasi dampak perubahan dimensi fitur laten terhadap kualitas rekonstruksi citra. Tabel 3.1 berikut menyajikan perbedaan konfigurasi pada masing-masing model, mencakup perubahan pada sisi *encoder* dan *decoder* untuk menyesuaikan arsitektur dengan desain segmen pengembangan yang diujikan.

Tabel 3.1 Rincian Segmen Pengembangan

Nomor Model	Spesifikasi Model	
	Segmen Pengembangan	**Loss Function**
1	Encoder: 10x10x128 → 5x5x256 → **2x2x32** Decoder: **2x2x32** → 5x5x256 → 10x10x128	MSE
2		MAE
3		SSIM
4		MSE + Gradient (3:7)
5		Charbonnier
6	Encoder: 10x10x128 → 5x5x256 → **2x2x48** Decoder: **2x2x48** → 5x5x256 → 10x10x128	MSE
7		MAE
8		SSIM
9		MSE + Gradient (3:7)
10		Charbonnier
11	Encoder: 10x10x128 → 5x5x256 → **2x2x64** Decoder: **2x2x64** → 5x5x256 → 10x10x128	MSE
12		MAE
13		SSIM
14		MSE + Gradient (3:7)
15		Charbonnier
16	Encoder: 10x10x128 → **5x5x8** Decoder: **5x5x8** → 10x10x128	MSE
17		MAE
18		SSIM
19		MSE + Gradient (3:7)
20		Charbonnier

Berdasarkan tabel 3.1, model-model yang digunakan dalam penelitian ini disusun berdasarkan struktur dasar *encoder-decoder* berbasis CNN yang ditujukan untuk melakukan kompresi dan rekonstruksi citra. Arsitektur umum dari setiap model terdiri atas tiga komponen utama, yaitu *encoder*, *latent space*, dan *decoder*. *Encoder* menerima *input* berupa gambar berukuran 80×80 piksel dan melakukan transformasi ke dalam bentuk representasi berdimensi lebih rendah, yaitu 20×20×64, yang kemudian diteruskan ke *latent space*. *Decoder* kemudian mengambil representasi tersebut dan merekonstruksi kembali menjadi citra berdimensi 80×80 piksel. Proses ini bersifat simetris, dengan tujuan mempertahankan sebanyak mungkin informasi visual yang terkandung dalam citra awal setelah proses kompresi.

Perbedaan utama antar model terletak pada segmen pengembangan, yaitu bagian tambahan di antara *encoder* dan *decoder* yang berfungsi untuk memodifikasi struktur jaringan guna mengatur tingkat kedalaman, jumlah filter, serta ukuran fitur

spasial. Segmen ini didesain untuk mengeksplorasi sejauh mana kompleksitas arsitektur berpengaruh terhadap hasil rekonstruksi gambar dan ukuran *latent space*. Dalam penelitian ini, sebanyak 20 model dibangun dengan 4 kelompok utama berdasarkan konfigurasi segmen pengembangan pada sisi *encoder* dan *decoder*.

Model 1 hingga 5 menggunakan konfigurasi tiga lapisan konvolusi berturut-turut pada sisi encoder, yaitu 10×10×128, 5×5×256, dan 2×2×32. Decoder dari model ini mengadopsi struktur simetris dengan urutan 2×2×32, 5×5×256, dan 10×10×128. Model 6 hingga 10 menggunakan konfigurasi yang serupa, tetapi dengan perbedaan pada lapisan terdalam *encoder* dan *decoder*, yaitu filter 2×2×48. Model 11 hingga 15 melanjutkan tren ini dengan peningkatan jumlah filter pada lapisan terdalam menjadi 2×2×64. Sementara itu, Model 16 hingga 20 dibangun dengan pendekatan yang lebih sederhana, di mana hanya terdapat dua lapisan konvolusi pada *encoder*, yaitu 10×10×128 dan 5×5×8, tanpa menyertakan lapisan 2×2. *Decoder* pada kelompok ini juga terdiri dari dua lapisan yang secara struktur mencerminkan *encoder*-nya.

Setiap konfigurasi segmen pengembangan diterapkan pada lima model yang berbeda, masing-masing dilatih dengan fungsi *loss* yang berbeda. Fungsi *loss* yang digunakan mencakup lima jenis yang banyak digunakan dalam literatur rekonstruksi citra: MSE, MAE, SSIM, kombinasi Gradient Loss dan MSE dalam rasio 3:7, serta Charbonnier Loss. Pemilihan kelima fungsi *loss* ini dilakukan untuk mengevaluasi pengaruh karakteristik fungsi kerugian terhadap dinamika pelatihan dan hasil rekonstruksi. Masing-masing fungsi *loss* memiliki karakteristik dan implikasi visual yang berbeda, sehingga penerapannya diharapkan dapat memberikan variasi dalam performa setiap model ketika diuji dengan dataset yang sama.

Dengan kombinasi antara empat konfigurasi arsitektur dan lima jenis fungsi *loss*, total 20 model independen dikembangkan. Setiap model dilatih secara terpisah dengan parameter pelatihan yang konsisten, menggunakan data citra yang telah disesuaikan resolusinya. Tujuan dari variasi ini bukan untuk menentukan konfigurasi terbaik, melainkan untuk memperoleh pemahaman yang lebih luas mengenai hubungan antara kompleksitas arsitektur, jenis fungsi *loss*, dan

dampaknya terhadap kualitas hasil rekonstruksi serta efisiensi representasi data. Data yang dihasilkan dari proses pelatihan dan pengujian selanjutnya digunakan dalam evaluasi kuantitatif maupun kualitatif untuk mengkaji potensi penerapan sistem ini dalam skenario transmisi gambar pada kanal komunikasi berbitrate rendah.

3.2.4 Pelatihan dan Evaluasi Model

Pada tahap ini, model dilatih menggunakan dataset yang telah disiapkan. Model dikompilasi dengan optimasi berbasis Adam yang dilengkapi dengan mekanisme *gradient clipping*. Skema penjadwalan *learning rate* diterapkan untuk menyesuaikan nilai pembelajaran seiring bertambahnya jumlah *epoch*, dengan tujuan menjaga stabilitas dan konvergensi model. Berikut adalah spesifikasi pelatihan yang digunakan dalam penelitian ini:

Tabel 3.2 Ketetapan nilai parameter untuk melatih model

Parameter	**Konfigurasi**
Batch Size 1	1
Optimizer	Adam
Gradient Clipping	Clip Norm: 1.0, Clip Value: 0.5
Loss Function	Charbonnier Loss (ε = 1e-3)
Initial Learning Rate	10^{-4}
Learning Rate Scheduler	
- Epoch 0-29	10^{-4}
- Epoch 30-59	10^{-5}
- Epoch 60-89	10^{-6}
- Epoch 90+	10^{-7}
Training Epochs	100
Callbacks	TimeHistory (mencatat waktu tiap epoch & LR)
Validation	Menggunakan dataset pengujian pada setiap epoch

Ditunjukkan pada tabel 3.2, model dilatih dengan *batch size* sebesar 1 dan menggunakan *optimizer* Adam, yang dilengkapi dengan mekanisme *gradient clipping* untuk menjaga stabilitas pelatihan, dengan *clip norm* sebesar 1.0 dan *clip*

value sebesar 0.5. Fungsi *loss* yang digunakan adalah Charbonnier Loss dengan nilai epsilon sebesar 1e-3. Proses pelatihan dilakukan selama 100 *epoch*, dengan skema penyesuaian *learning rate* yang ditentukan berdasarkan jumlah *epoch* yang telah dilalui. Pada awal pelatihan, nilai *learning rate* ditetapkan sebesar 1e-4, kemudian dikurangi secara bertahap menjadi 1e-5 setelah 30 *epoch*, 1e-6 setelah 60 *epoch*, dan 1e-7 setelah 90 *epoch* untuk memastikan model dapat beradaptasi dengan baik. Selain itu, *callback TimeHistory* digunakan untuk mencatat waktu pelatihan pada setiap *epoch* serta memantau perubahan *learning rate* sepanjang proses pelatihan. Validasi dilakukan pada setiap *epoch* menggunakan dataset pengujian untuk mengevaluasi kinerja model selama proses pelatihan berlangsung.

Pelatihan seluruh model dalam penelitian ini dilakukan secara lokal menggunakan perangkat keras dan perangkat lunak yang telah disesuaikan untuk mendukung proses komputasi intensif pada CNN. Berikut ini adalah spesifikasi perangkat dan versi *library* yang dipasangkan pada *environment* pelatihan model

Tabel 3.3 Spesifikasi perangkat dan *environment* pelatihan model

Perangkat/*Library*	**Jenis/versi**
Central Processing Unit (CPU)	12th Gen Intel(R) Core(TM) i5-12450HX
Random Access Memory (RAM)	12 GB
Graphical Processing Unit (GPU)	NVIDIA GeForce RTX 3050 6GB Laptop GPU
Tensorflow	Tensorflow-GPU versi 2.10.0
CUDA Deep Neural Network (cuDNN)	Versi 8.1

Berdasarkan Tabel 3.3 Perangkat keras yang digunakan meliputi prosesor 12th Gen Intel(R) Core(TM) i5-12450HX, unit pemrosesan grafis NVIDIA GeForce RTX 3050 6GB Laptop GPU, serta memori utama (RAM) sebesar 12 GB. Konfigurasi ini dipilih karena dirasa cukup mampu menangani beban komputasi menengah yang diperlukan dalam pelatihan model encoder-decoder berbasis CNN dengan ukuran gambar yang relatif kecil dan jumlah parameter yang masih terkontrol. Untuk mendukung percepatan komputasi GPU, pelatihan dilakukan menggunakan pustaka TensorFlow versi 2.10.0 dengan dukungan TensorFlow-

GPU serta integrasi cuDNN versi 8.1. Seluruh proses pelatihan dijalankan dalam lingkungan Python dan dilengkapi dengan manajemen memori GPU untuk memastikan proses berjalan secara efisien. Penggunaan perangkat ini memungkinkan seluruh model dilatih secara seri dan berurutan tanpa memerlukan infrastruktur komputasi berskala besar.

3.2.5 Pengujian Model

Pada tahap evaluasi, beberapa parameter diukur untuk memahami kinerja model dalam proses kompresi dan rekonstruksi gambar. Selama proses pelatihan, nilai *loss* diamati secara bertahap untuk memantau bagaimana model belajar dari data yang diberikan dan apakah proses konvergensi berjalan secara stabil. Setelah seluruh model selesai dilatih, dilakukan pengujian menggunakan 50 gambar uji untuk mengevaluasi performa rekonstruksi citra. Pengujian ini dilakukan dalam tiga skema pengujian berbeda, yaitu: tanpa kompresi tambahan (*baseline*), dengan kompresi Blosc, dan dengan kompresi Zlib. Ketiga skema pengujian tersebut dimaksudkan untuk mengkaji apakah penerapan algoritma kompresi tambahan pada *latent space* dapat meningkatkan efisiensi penyimpanan dan transmisi tanpa menyebabkan degradasi signifikan pada kualitas hasil rekonstruksi.

Beberapa metrik utama yang diukur dalam tahap uji coba meliputi ukuran file *latent space* setelah proses encoding dalam satuan byte, yang mencerminkan tingkat kompresi awal yang dicapai oleh model. Selanjutnya, dilakukan perbandingan ukuran file citra hasil rekonstruksi dengan gambar aslinya untuk menilai sejauh mana informasi dapat dipertahankan. Kualitas rekonstruksi dievaluasi menggunakan nilai *Peak Signal-to-Noise Ratio* (PSNR), yang secara kuantitatif membandingkan kesamaan antara citra asli dan citra hasil rekonstruksi. Di samping itu, dilakukan estimasi waktu pengiriman dengan asumsi bitrate sebesar **37,5 kbps**, yang disesuaikan dengan karakteristik kanal transmisi LoRa. Estimasi ini dinyatakan dalam satuan milidetik (ms) dan dihitung berdasarkan ukuran akhir dari *latent space* setelah diterapkan skema kompresi masing-masing. Melalui serangkaian pengujian ini, diharapkan dapat diperoleh pemahaman yang lebih menyeluruh mengenai efektivitas masing-masing model dalam menghasilkan

representasi citra yang efisien, baik dari sisi kualitas visual maupun potensi penerapannya dalam sistem transmisi berkecepatan rendah.

3.2.6 Rekapitusi Data

Rekapitulasi data mencakup rangkuman hasil pelatihan dan evaluasi model dengan metrik, termasuk perbandingan metrik PSNR dengan rumus:

$$PSNR = 20 \times log_{10}\left(\frac{Maks_1}{\sqrt{MSE}}\right)$$

Dengan $Maks_1$adalah Nilai maksimum piksel gambar, dan MSE merupakan Mean Squared Error antara gambar input dan gambar hasil rekonstruksi. Selain itu juga terdapat parameter ukuran file *latent space*, dan estimasi durasi transmisi secara hitungan dengan rumus:

$$Waktu(ms) = (\frac{Ukuran\ file\ (byte)}{37.500/8}) \times 1000$$

. Data dari pengujian 20 model kemudian dicatat untuk tujuan analisis tambahan untuk menentukan konfigurasi terbaik untuk kompresi gambar yang efisien dalam pengiriman, namun masih bisa meminimalkan dampak dari kompresi ekstrimnya.

3.2.7 Analisis Data

Analisis data pada penelitian ini dilakukan dengan membahas temuan-temuan yang didapatkan dati rekapitulasi hasil pengujian terhadap 20 model. Setiap temuan tersebut dilakukan analisis serta diidentifikasi faktor-faktor yang turut berpengaruh. Kemudian dilakukan penelusuran terhadap dampak dari faktor-faktor tersebut terhadap parameter performa yang telah didefinisikan. Analisis ini kemudian bisa digunakan sebagai dasar untuk melakukan perbandingan antar model. Pada tahap ini juga dilakukan seleksi terhadap model-model yang menunjukkan hasil terbaik dengan pertimbangan keseimbangan antara nilai kualitas rekonstruksi dan efisiensi representasi data.