

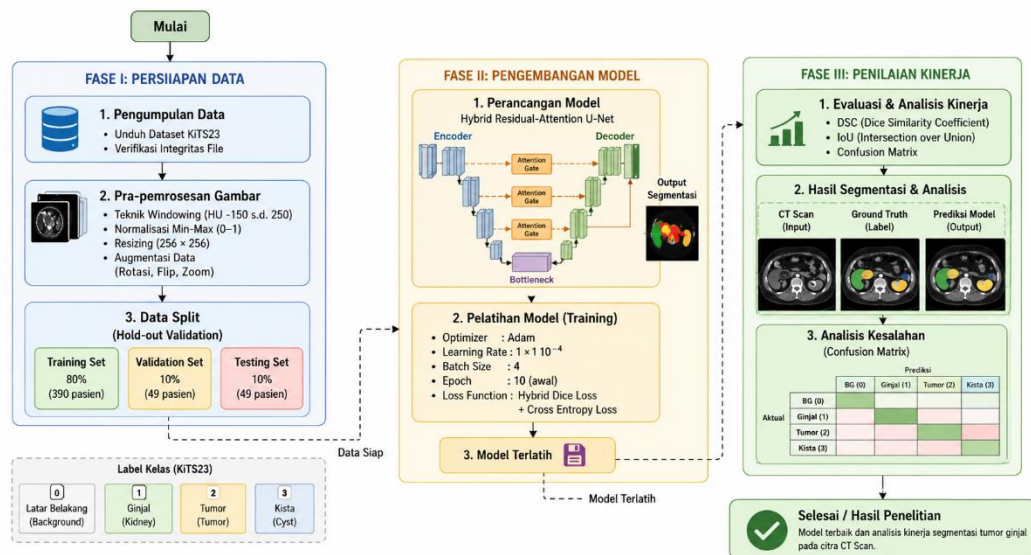
BAB III METODOLOGI PENELITIAN

3.1 Desain dan Strategi Penelitian

Strategi penelitian dirancang untuk mengukur efektivitas metode usulan, yaitu Hybrid Residual-Attention U-Net, dalam melakukan segmentasi semantik pada citra medis. Variabel yang dimanipulasi adalah arsitektur model (penggunaan blok *Residual* dan *Attention Gate*), sedangkan variabel yang diukur adalah akurasi segmentasi (Dice Score, IoU) serta analisis kesalahan klasifikasi tingkat piksel menggunakan Confusion Matrix.

3.2 Tahapan Penelitian

Untuk menjamin proses penelitian yang terstruktur dan sistematis, tahapan pelaksanaan penelitian disusun mengikuti alur kerja (*pipeline*) *Deep Learning* standar. Secara visual, tahapan penelitian mulai dari pengumpulan data hingga evaluasi model digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Berdasarkan Gambar 3.1, tahapan penelitian meliputi:

1. Pengumpulan Data: Mengunduh dataset KiTS23 dan verifikasi integritas file.
2. Pra-pemrosesan: Teknik *windowing* dan normalisasi untuk standarisasi input citra.

3. Perancangan Model: Implementasi kode arsitektur *Hybrid Residual-Attention U-Net*.
4. Pelatihan Model: Proses *training* menggunakan akselerasi GPU di *cloud*.
5. Evaluasi & Analisis: Pengukuran kuantitatif menggunakan metrik DSC, IoU, dan *confusion matrix*.

3.3 Tahap Pengumpulan Data

Tahap pertama adalah pengumpulan data yang menjadi bahan utama pembelajaran model.

3.3.1 Spesifikasi Dataset

Penelitian ini menggunakan data sekunder dari repositori publik *The 2023 Kidney and Kidney Tumor Segmentation Challenge* (KiTS23) (Heller, Isensee, Trofimova, Tejpau, & al., 2023). Dataset ini dipilih karena menawarkan kebaruan (*novelty*) berupa variabilitas klinis yang lebih kompleks dibandingkan pendahulunya (KiTS19), mencakup fase kontras yang beragam dan struktur tumor yang heterogen.

Karakteristik teknis dataset ini meliputi:

1. Format Data: Citra volumetrik 3D disimpan dalam format NIfTI (.nii.gz) yang mempertahankan metadata medis dan nilai intensitas asli dalam satuan *Hounsfield Unit* (HU).
2. Label Semantik: *Ground truth* dianotasi oleh pakar medis ke dalam kelas: (0) Latar Belakang, (1) Ginjal, (2) Tumor, dan (3) Kista. Sesuai batasan masalah, penelitian ini berfokus pada segmentasi Ginjal dan Tumor.

3.3.2 Pembagian Data (*Data Splitting*)

Untuk menghindari bias dan *overfitting*, dataset dibagi menggunakan skema *Hold-out Validation* dengan proporsi sebagai berikut:

1. Data Latih (*Training Set*) - 80%: Digunakan untuk melatih bobot model (390 pasien).
2. Data Validasi (*Validation Set*) - 10%: Digunakan untuk memantau kinerja saat pelatihan dan menyetel *hyperparameter* (49 pasien).

3. Data Uji (*Testing Set*) - 10%: Digunakan untuk evaluasi akhir model pada data yang belum pernah dilihat sebelumnya (49 pasien).

3.4 Tahap Pra-pemrosesan Data (*Preprocessing*)

Data mentah CT Scan memiliki rentang intensitas yang sangat luas dan artefak yang dapat mengganggu model. Oleh karena itu, tahap pra-pemrosesan dilakukan untuk menstandarisasi input citra.

3.4.1 *Windowing* (Pengaturan Kontras)

Citra CT Scan asli memiliki rentang intensitas -1000 HU hingga >3000 HU. Teknik *intensity clipping* diterapkan pada rentang -150 HU hingga 250 HU (*Soft Tissue Window*). Langkah ini bertujuan untuk:

1. Menghilangkan informasi yang tidak relevan seperti udara (sangat gelap) dan tulang (sangat terang).
2. Meningkatkan kontras visual pada jaringan lunak ginjal dan area tumor agar fitur tekstur lebih menonjol bagi model.

3.4.2 Normalisasi Min-Max

Setelah *windowing*, nilai intensitas piksel ditransformasi ke dalam rentang desimal [0, 1] menggunakan normalisasi Min-Max. Proses ini krusial untuk menjaga stabilitas numerik dan mempercepat konvergensi gradien stokastik selama proses pelatihan (*training*).

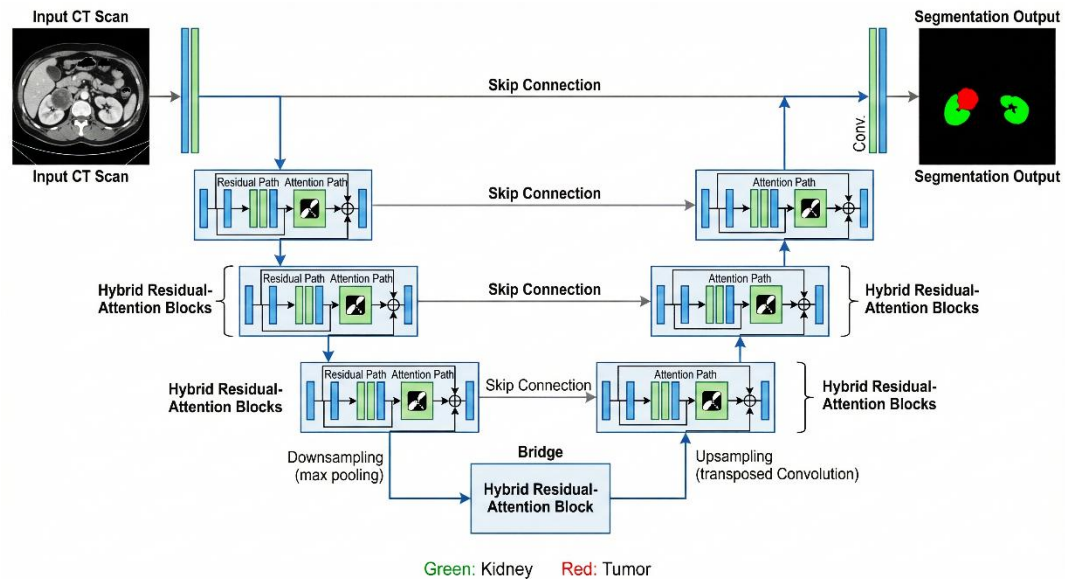
$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

3.4.3 *Resizing* dan *Augmentasi*

Untuk efisiensi komputasi mengingat keterbatasan memori GPU, resolusi spasial citra irisan diseragamkan menjadi 256×256 piksel. Selanjutnya, untuk memperkaya variasi data latih dan meningkatkan generalisasi model, diterapkan teknik augmentasi data berupa rotasi ($\pm 10^\circ$), pencerminan horizontal (*horizontal flip*), dan perbesaran (*zooming*).

3.5 Tahap Perancangan Model

Pada tahap ini, arsitektur *Hybrid Residual-Attention U-Net* dibangun dengan mengintegrasikan modul-modul canggih ke dalam kerangka dasar U-Net. Desain ini ditujukan untuk mengatasi kelemahan U-Net standar dalam menangkap detail batas tumor yang kompleks .



Gambar 3.2 Arsitektur Hybrid Residual-Attention U-Net Usulan

Komponen utama arsitektur meliputi:

1. Jalur *Encoder* menggunakan blok Residual (*ResBlock*) untuk menggantikan lapisan konvolusi biasa guna mengatasi masalah degradasi pada jaringan yang dalam (Jin et al., 2021).
2. Attention Gates (AGs): Ditempatkan pada jalur penghubung (*skip connection*) antara Encoder dan Decoder. Modul ini berfungsi menekan sinyal fitur dari latar belakang yang tidak relevan dan memfokuskan bobot model pada area target tumor (Jin et al., 2021).
3. Jalur Decoder: Melakukan *Upsampling* untuk mengembalikan resolusi citra menjadi peta segmentasi utuh.

3.6 Tahap Pelatihan Model (*Training*)

Proses pelatihan (*training*) model dieksekusi menggunakan akselerasi GPU dengan konfigurasi *hyperparameter* yang telah disesuaikan dengan keterbatasan

memori dan kompleksitas dataset KiTS23. Berikut adalah rincian spesifikasi pelatihannya:

1. Epoch: Pelatihan dijalankan selama 10 epoch awal untuk memantau dinamika konvergensi model.
2. Batch Size: Ditetapkan sebesar 4, yang merupakan ukuran optimal untuk memaksimalkan VRAM GPU (16 GB) saat memproses citra resolusi 256×256 piksel.
3. Optimizer: Menggunakan algoritma Adam (*Adaptive Moment Estimation*) karena kemampuannya dalam menyesuaikan *learning rate* secara dinamis.
4. Learning Rate: Diinisialisasi pada nilai yang kecil, yaitu 1×10^{-4} (0.0001), guna mencegah model mengalami *overshooting* saat mencari titik minimum global.
5. Loss Function: Menggunakan kombinasi hibrida antara *Cross-Entropy Loss* (untuk akurasi klasifikasi piksel) dan *Dice Loss* (untuk mengoptimalkan tumpang tindih area geometris). Kombinasi $L_{\text{total}} = \alpha \cdot L_{\text{Dice}} + \beta \cdot L_{\text{CE}}$ ini diterapkan untuk menangani masalah ketidakseimbangan kelas (*class imbalance*) ekstrem pada dataset.

3.7 Tahap Evaluasi dan Analisis

Setelah model selesai dilatih, dilakukan pengujian menggunakan data uji (*testing set*) yang belum pernah dilihat model sebelumnya. Evaluasi dilakukan secara kuantitatif untuk membuktikan hipotesis penelitian.

3.7.1 Metrik Akurasi

Kinerja model diukur menggunakan metrik standar segmentasi medis:

1. Dice Similarity Coefficient (DSC): Mengukur persentase tumpang tindih (*overlap*) antara prediksi model dan *ground truth*. Nilai mendekati 1 menunjukkan akurasi tinggi.
2. Intersection over Union (IoU): Memberikan penalti yang lebih ketat terhadap kesalahan prediksi. Metrik ini sangat sensitif dalam mengukur ketepatan batas (*boundary precision*) area tumor.

3.7.2 Analisis Kesalahan (*Confusion Matrix*)

Analisis mendalam dilakukan pada tingkat piksel untuk mengidentifikasi jenis kesalahan model:

1. False Positive (FP): Kasus *over-segmentation*, di mana jaringan sehat terdeteksi sebagai tumor.
2. False Negative (FN): Kasus *under-segmentation*, di mana area tumor gagal dideteksi. Analisis ini digunakan untuk mengetahui kelemahan model dalam membedakan tumor dengan organ sekitarnya.

3.8 Instrumen Penelitian

Mengingat kompleksitas komputasi dataset KiTS23, penelitian ini menggunakan infrastruktur berbasis *Cloud Computing*:

1. Perangkat Keras (*Hardware*):
 - a. Platform: Google Colab.
 - b. GPU: NVIDIA Tesla T4 dengan 16 GB GDDR6 VRAM untuk akselerasi komputasi matriks *Deep Learning*.
 - c. System RAM: 51 GB RAM untuk memfasilitasi pemuatan dan pra-pemrosesan dataset volumetrik KiTS23.
 - d. Penyimpanan: Google Drive (terintegrasi dengan Colab) untuk menyimpan dataset volumetrik dan model.
2. Perangkat Lunak (*Software*):
 - a. Bahasa: Python 3.x.
 - b. Framework Deep Learning: PyTorch / TensorFlow.
 - c. Library Medis: NiBabel (untuk membaca format .nii.gz) dan NumPy.