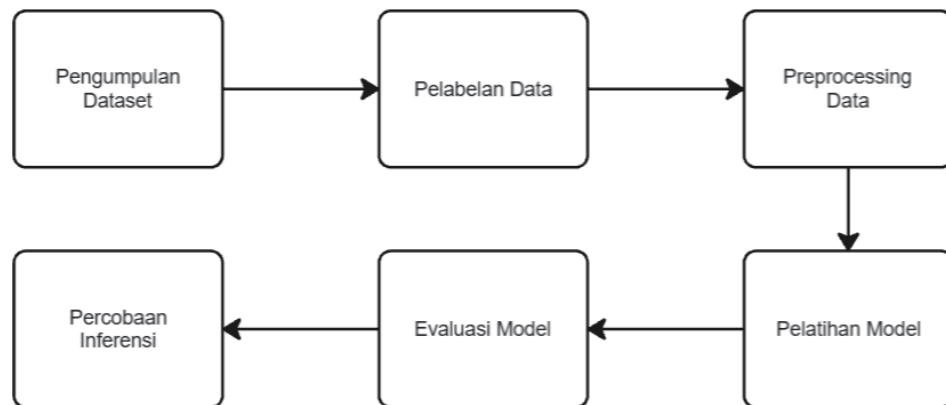


## BAB III

### METODE PENELITIAN

#### 3.1 Perancangan Solusi

Penelitian ini mencakup beberapa tahapan, diawali dengan pengumpulan dataset. Data yang diperoleh kemudian melewati proses pelabelan dan *preprocessing* agar formatnya sesuai dengan arsitektur model yang dipakai. Data tersebut selanjutnya digunakan dalam pelatihan model dan dievaluasi kinerjanya, sebelum akhirnya divalidasi melalui percobaan inferensi. Alur metode penelitian selengkapnya dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Tahapan Penelitian

##### 3.1.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini mencakup 40 WSI dari kasus PTC Varian *tall cell* dengan pewarnaan H&E. Data yang dipakai sudah dinyatakan lolos kaji etik dari institusi yang bersangkutan. Proses pengumpulan data melibatkan dokter spesialis patologi anatomi (Sp.PA). Sampel yang dipilih telah dikonfirmasi berdasarkan laporan diagnosis histopatologi pasien untuk memastikan bahwa seluruh citra yang digunakan benar-benar merepresentasikan kasus varian *tall cell* yang valid. Dataset berasal dari 7 pasien berbeda dengan distribusi *slide* berkisar antara 2 sampai 10 per pasien. Secara teknis, seluruh citra WSI didigitalisasi menggunakan *scanner* Roche

DP600 pada magnifikasi 20x. Citra yang dihasilkan memiliki resolusi spasial sebesar 0,5  $\mu\text{m}$  per piksel dan disimpan dalam format digital .bif untuk mempertahankan kualitas detail jaringan.

### 3.1.2 Pelabelan Data

Tahap pelabelan data bertujuan untuk menghasilkan *Ground Truth* yang akan digunakan sebagai referensi utama dalam pelatihan dan evaluasi model. Mengingat penentuan batas tumor membutuhkan keahlian klinis spesifik, proses anotasi dalam penelitian ini dilakukan sepenuhnya secara langsung oleh dokter spesialis patologi anatomi. Pelabelan dilakukan menggunakan *software* QuPath [20], yang dipilih karena kapabilitasnya dalam menangani citra WSI secara efisien. Dalam proses ini, dokter spesialis patologi anatomi melakukan penggambaran ROI secara manual untuk memisahkan area tumor dari jaringan non-tumor.

### 3.1.3 Preprocessing Data

Tahap *preprocessing* data bertujuan untuk mentransformasi data mentah menjadi format yang sesuai dengan spesifikasi input arsitektur model. Mengingat ukuran WSI yang sangat besar, pemrosesan tidak dapat dilakukan secara langsung pada citra utuh. Oleh karena itu, dilakukan ekstraksi fitur dengan memotong WSI menjadi bagian-bagian kecil (*patch*) berukuran 512x512 piksel untuk menyeimbangkan beban komputasi memori GPU dengan *Field of View* yang cukup luas untuk merepresentasikan struktur jaringan tumor pada perbesaran 20x.

Setelah seluruh *patch* terkumpul, dataset akan dibagi menjadi subset *training*, *validation*, dan *testing*. Pembagian dilakukan pada level pasien (*patient-level split*) dengan rasio 5:1:1. Proses alokasi pasien ke dalam setiap *subset* disusun dengan mempertimbangkan jumlah *patch* yang dimiliki oleh masing-masing individu, guna memastikan distribusi total data pada setiap subset tetap proporsional

dan seimbang. Secara teknis, pembagian ini dikelola menggunakan *file* CSV. Pendekatan ini dipilih karena menimbang volume dataset yang besar, penggunaan CSV mampu memberikan fleksibilitas eksperimen, di mana perubahan alokasi pasien antar *subset* dapat dilakukan hanya dengan memodifikasi kolom pada file tersebut. Strategi *patient-level split* diterapkan untuk mencegah terjadinya *data leakage*, di mana kemiripan karakteristik jaringan dari pasien yang sama dapat membuat hasil bias pada evaluasi. Dengan memisahkan pasien sepenuhnya antar subset, model dapat dievaluasi secara objektif untuk memastikan kemampuan generalisasinya terhadap data pasien yang belum pernah dilihat sebelumnya.

#### 3.1.4 Pelatihan Model

Tahap pelatihan model untuk tugas segmentasi tumor menggunakan arsitektur SegFormer dengan *backbone* MiT-B2. SegFormer dipilih karena pendekatannya yang berbasis *Transformer*, di mana mekanisme *self-attention* memungkinkan model untuk menangkap konteks global dari citra histopatologi secara lebih efektif [13]. Varian *backbone* MiT-B2 digunakan karena menawarkan keseimbangan optimal antara kompleksitas parameter dan performa komputasi, serta telah melalui proses *pre-training* pada dataset ImageNet-1K untuk mempercepat ekstraksi fitur. Proses pelatihan ini dilakukan menggunakan platform Google Colab Pro+ dengan dukungan GPU A100 untuk menangani beban komputasi yang diperlukan *Transformer*.

Sebelum dataset bisa digunakan untuk pelatihan model, diterapkan teknik augmentasi dan normalisasi data pada subset *training* untuk meningkatkan kemampuan generalisasi model. Rangkaian proses augmentasi meliputi:

- Rotasi dan Flip

Manipulasi geometris berupa rotasi acak serta *flip* secara horizontal maupun vertikal bertujuan untuk memperkaya variasi data, agar model mampu mengenali fitur tumor dalam berbagai orientasi.

- Color Jitter

Perubahan atribut warna yang mencakup kecerahan, kontras, saturasi, dan *hue* secara acak berfungsi untuk mensimulasikan variasi ketebalan pewarnaan H&E yang sering berbeda antar preparat, sehingga model menjadi lebih *robust* terhadap inkonsistensi visual pada data baru.

- Normalisasi

Seluruh citra input dinormalisasi menggunakan nilai rata-rata dan standar deviasi statistik dari dataset referensi ImageNet. Proses ini bertujuan menyamakan distribusi data input dengan bobot *pre-trained* pada *backbone* model, yang penting untuk mempercepat konvergensi dan menjaga stabilitas gradien selama pelatihan.

Selain pemrosesan pada citra input, dilakukan juga pemrosesan pada *mask* melalui teknik binarisasi. *Mask* yang memiliki rentang intensitas piksel standar (0, 255) dikonversi menjadi format biner menggunakan metode *thresholding*. Langkah ini memastikan bahwa area pada label terbagi menjadi dua kelas saja, yaitu area tumor (1) dan non-tumor (0).

Implementasi teknis model dibangun menggunakan *framework* PyTorch dengan memanfaatkan *library* Hugging Face Transformers untuk memanggil arsitektur SegFormer. Dalam proses *training*, diterapkan sejumlah konfigurasi *hyperparameter* yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Hyperparameter Pelatihan Model

Hyperparameter	Value
Epoch	50
Batch Size	16
Learning Rate	$6 \times 10^{-5}$
Optimizer	AdamW
Loss_fn	BCE + Dice Loss + Label Smoothing
Scheduler	ReduceLROnPlateau (Patience: 3, Factor: 0.5)
Gradient Clipping	1.0
Early Stopping	15

*Training* dibatasi hingga maksimum 50 epoch, namun dilengkapi dengan mekanisme *Early Stopping* dengan patience 15. Mekanisme ini digunakan untuk mencegah *overfitting* dengan menghentikan proses pelatihan secara otomatis jika tidak terjadi peningkatan pada *validation loss* selama 15 iterasi berturut-turut. *Batch Size* ditetapkan sebesar 16 untuk menyesuaikan kapasitas memori VRAM GPU. Proses pembaruan parameter dikelola oleh *Optimizer* AdamW dengan inisialisasi *learning rate* sebesar  $6 \times 10^{-5}$ . *Learning rate* yang rendah diterapkan untuk mengakomodasi strategi *fine-tuning*, di mana penyesuaian bobot dilakukan secara bertahap demi mempertahankan representasi fitur yang telah dipelajari dari *pre-trained weights*.

Selanjutnya, untuk menjamin konvergensi yang optimal, diterapkan mekanisme *scheduler* ReduceLROnPlateau. Algoritma ini memantau pergerakan *validation loss*, apabila terjadi stagnasi selama 3 epoch berturut-turut, *learning rate* akan direduksi sebesar 0.5. *Gradient Clipping* dengan *threshold* 1 diterapkan untuk memitigasi masalah exploding gradients yang umum terjadi pada arsitektur

berbasis *Transformer*, di mana gradien yang terlalu besar dapat menyebabkan nilai loss menjadi tidak terdefinisi (NaN) [21].

Kombinasi *Binary Cross Entropy* (BCE) dan *Dice Loss* diterapkan sebagai fungsi *loss*. BCE difokuskan untuk menangani klasifikasi pada setiap piksel individual, sedangkan *Dice Loss* berperan untuk mengatasi *class imbalance* dengan mengoptimalkan tingkat *similarity* pada area *overlap* tumor secara keseluruhan [22]. Selain itu, diterapkan teknik *Label Smoothing* sebesar 0.1. Teknik ini mengubah nilai label yang semula bernilai mutlak (0 atau 1) menjadi nilai yang sedikit lebih rendah. Penyesuaian ini bertujuan untuk mencegah model agar tidak menjadi *overconfident* terhadap data *training* yang mungkin memiliki ambiguitas pada batas tumor, sehingga meningkatkan kemampuan generalisasi.

Proses pelatihan model dimulai dengan inisialisasi *dataloader* untuk menyediakan *batch* data yang sudah disesuaikan ke dalam arsitektur Segformer. Model kemudian mempelajari pola visual area tumor secara iteratif. Pada setiap *epoch*, model menghasilkan prediksi segmentasi yang langsung dibandingkan dengan *ground truth* melalui perhitungan fungsi *loss*. Nilai *error* yang didapat kemudian menjadi dasar bagi mekanisme *backpropagation* untuk mengoreksi dan memperbarui nilai *weights* dalam jaringan. Penyesuaian berulang ini bertujuan untuk meminimalkan *loss* secara konsisten, sehingga model perlahan menjadi mahir dalam membedakan jaringan tumor dengan jaringan normal. Proses ini terus berlanjut hingga model mencapai titik konvergensi atau batas *epoch* maksimum tanpa terjebak pada *overfitting*.

### 3.2 Metode Pengujian

Pengujian dilakukan menggunakan *test* dataset yang berasal dari 1 pasien yang telah dipisahkan sejak awal. Data ini tidak pernah dilihat oleh model selama proses pelatihan maupun validasi, sehingga hasil evaluasi yang

diperoleh dapat merepresentasikan kemampuan generalisasi model secara objektif pada kasus baru. Evaluasi performa model dilakukan melalui analisis *learning curve* dan perhitungan metrik evaluasi serta percobaan inferensi.

- Learning Curve

Analisis *learning curve* dilakukan untuk memantau dinamika pelatihan model dengan memvisualisasikan perubahan nilai *loss* pada data *training* dan *validation* terhadap jumlah *epoch* yang dilalui. Grafik ini berfungsi sebagai indikator diagnostik utama untuk mendeteksi apakah model mengalami *underfitting*, *overfitting*, atau telah mencapai konvergensi yang optimal. Apabila kurva *training loss* dan *validation loss* menunjukkan pola penurunan yang stabil, konsisten, dan saling berdekatan hingga mencapai titik minimum, maka proses pelatihan dianggap konvergen dan model memiliki keseimbangan yang baik (*good fit*). Sebaliknya, indikasi *overfitting* terlihat apabila *training loss* terus menurun secara tajam sementara *validation loss* mulai stagnan atau justru meningkat, yang menandakan bahwa model hanya menghafal data latih dan gagal mengenali pola pada data baru. Kurva yang stabil pada akhir pelatihan menjadi acuan untuk memilih *checkpoint best model* yang akan digunakan dalam pengujian akhir.

- Evaluasi Metrik

Sebelum memasuki *evaluation metrics*, terdapat 4 nilai yang berperan sebagai basis perhitungannya, yaitu:

- True Positive (TP): Jumlah piksel tumor yang diprediksi dengan benar sesuai dengan *ground truth* (GT).
- False Positive (FP): Jumlah piksel non-tumor yang salah diprediksi sebagai tumor.
- False Negative (FN): Jumlah piksel tumor yang salah diprediksi sebagai non-tumor.
- True Negative (TN): Jumlah piksel non tumor yang diprediksi dengan benar sesuai dengan GT.



Berdasarkan keempat nilai dasar tersebut, kinerja model diukur menggunakan metrik-metrik berikut:

- Dice Similarity Coefficient (DSC)

Metrik DSC atau *dice score* merupakan metrik evaluasi yang umum digunakan dalam segmentasi. Metrik ini mengukur tingkat kemiripan antara dua sampel. Berbeda dengan IoU, rumus DSC memberikan *weight* ganda pada area irisan (*intersection*), sehingga nilainya cenderung sedikit lebih tinggi daripada IoU untuk prediksi yang sama. Secara matematis, DSC dihitung menggunakan persamaan berikut:

$$Dice\ Score = \frac{2 |GT \cap SR|}{|GT| + |SR|} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Dimana SR adalah area hasil prediksi model dan GT adalah *ground truth*. Nilai DSC berkisar antara 0 hingga 1 dimana nilai 0 mengindikasikan tidak ada *overlap* sama sekali, sedangkan nilai 1 menunjukkan bahwa hasil segmentasi model *overlap* secara sempurna dengan *ground truth*. Dalam konteks penelitian ini, DSC digunakan untuk menilai seberapa mirip total area prediksi model dengan *ground truth*. Metrik ini menunjukkan seberapa baik model mampu menutupi seluruh area tumor yang seharusnya dideteksi.

- Intersection over Union (IoU)

Metrik IoU atau disebut Jaccard index digunakan untuk mengukur tingkat kesesuaian antara area hasil segmentasi model dengan area sebenarnya yang dianotasi oleh ahli (*ground truth*). Metrik ini menilai seberapa besar *overlap* antara prediksi dan *ground truth*, sehingga menjadi indikator utama untuk menilai performa model segmentasi. Secara matematis, IoU dihitung dengan rumus berikut:



$$IoU = \frac{|GT \cap SR|}{|GT \cup SR|} = \frac{TP}{TP + FP + FN}$$

dengan SR menyatakan area hasil prediksi model dan GT adalah *ground truth*. Nilai IoU berada dalam rentang 0 hingga 1, di mana nilai yang mendekati 1 menunjukkan segmentasi yang sangat akurat. Dalam konteks penelitian ini, nilai IoU yang tinggi mengindikasikan bahwa model mampu memetakan batas wilayah tumor dengan presisi yang baik. Hal ini menjamin bahwa area prediksi benar-benar fokus pada jaringan tumor tanpa melebar ke jaringan sehat di sekitarnya.

- Recall

*Recall*, atau sering disebut juga sebagai Sensitivitas dalam dunia medis, berfungsi untuk mengukur kemampuan model dalam menemukan seluruh area positif. Dalam kasus ini, *recall* dihitung dengan membandingkan jumlah piksel tumor yang terdeteksi dengan benar terhadap total keseluruhan piksel tumor. Nilai *Recall* yang tinggi menjamin bahwa seluruh jaringan tumor berhasil dikenali oleh model.

$$Recall = \frac{TP}{TP + FN}$$

- Precision

*Precision* berfokus pada tingkat ketepatan atau reliabilitas dari prediksi positif yang dihasilkan model. Secara umum, metrik ini menghitung rasio antara prediksi yang benar dibandingkan dengan total seluruh prediksi yang diklaim sebagai kelas positif, termasuk adanya kesalahan prediksi. Dalam konteks penelitian ini, nilai *Precision* yang tinggi memastikan bahwa area yang ditandai sebagai tumor benar-benar merupakan jaringan tumor dan bukan

jaringan sehat yang salah dikenali. Rumus matematis *precision* dinyatakan sebagai:

$$Precision = \frac{TP}{TP + FP}$$

- Percobaan Inferensi

Tahap inferensi diawali dengan memuat kembali arsitektur model yang telah dilatih, menggunakan *best* model yang tersimpan saat *validation loss* mencapai titik minimum. Penggunaan *best* model dipilih untuk memastikan bahwa evaluasi dilakukan menggunakan iterasi model dengan performa generalisasi paling optimal. Selanjutnya, dilakukan proses inferensi pada *test* dataset secara acak yang hasilnya kemudian divisualisasikan untuk keperluan analisis. Visualisasi yang akan ditampilkan berupa citra asli, *ground truth*, hasil prediksi model, serta tampilan *overlay* antara prediksi dan *ground truth* serta *difference map* yang menampilkan perbedaan antara keduanya.

