

BAB III METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Penelitian ini bertujuan untuk mengumpulkan data citra jalanan yang representatif dari kondisi nyata di Surabaya, Indonesia menggunakan teknik web scraping. Teknik ini dilakukan pada Google Street View dengan bantuan alat otomatis Selenium. Pemilihan teknik web scraping menggunakan Selenium bukan tanpa alasan. Dibandingkan dengan dataset sintetis yang sering digunakan dalam penelitian sejenis, web scraping memungkinkan pengambilan data yang lebih bervariasi, mencerminkan kondisi dunia nyata yang sulit diprediksi. Teknik ini memberikan kemudahan dalam menangkap citra dari berbagai kondisi jalan yang berbeda, seperti jalan utama, jalan sempit, hingga area deng(Yılmaz & Selvi, 2023)g dinamis(Yılmaz & Selvi, 2023).

3.1.1 Metode yang digunakan

Selenium WebDriver digunakan sebagai alat utama untuk mengakses Google Street View dan mengambil gambar jalanan secara otomatis. Dengan menggunakan Python untuk messngendalikan Selenium, peneliti dapat mengarahkan pengambilan gambar ke lokasi-lokasi tertentu di Surabaya yang mewakili berbagai kondisi jalan yang ingin diamati. Selenium memungkinkan navigasi dinamis pada halaman web, sehingga dapat menyesuaikan posisi kamera dan menangkap citra dari berbagai sudut dan arah(Umar et al., 2023).

Langkah-langkah yang dilakukan dalam pengumpulan data adalah sebagai berikut:

1. Inisialisasi dan Navigasi Lokasi:

- Skrip otomatis menggunakan **Selenium** untuk membuka **Google Street View** dan mengarahkan pandangan ke titik koordinat tertentu di Surabaya. Koordinat yang dipilih bukan hanya berdasarkan posisi geografis, tetapi juga berdasarkan jenis jalan yang ada, seperti jalan utama, jalan sempit, atau area yang lebih kompleks dengan kegiatan manusia, kendaraan parkir sembarangan, dan pedagang kaki lima.
- Lokasi-lokasi ini dipilih untuk mencakup beragam kondisi jalan yang dapat terjadi dalam kehidupan sehari-hari, mencakup semua elemen yang

berpotensi memengaruhi visualisasi citra, seperti keberadaan kendaraan dan pejalan kaki.

2. Pengambilan Gambar:

- Gambar-gambar diambil langsung dari elemen visual di Google Street View dengan menggunakan fitur screenshot untuk menangkap gambar dari sudut pandang tertentu. Pengambilan gambar dilakukan pada sudut 90° dan 270° agar mendapatkan variasi sudut pandang yang lebih lengkap.
- Mengapa memilih sudut 90° dan 270° ? Sudut 90° mengarah ke kiri (atau kanan) dari posisi kamera, sedangkan 270° mengarah ke sisi berlawanan, memberikan pandangan yang lebih luas terhadap situasi di sekeliling jalan yang diambil. Kedua sudut ini memastikan bahwa gambar yang dihasilkan menggambarkan kondisi jalan secara keseluruhan, memberikan pandangan lebih lengkap mengenai keberadaan objek seperti kendaraan, pejalan kaki, serta elemen-elemen lain di jalan.

3. Penyimpanan Data:

Setelah pengambilan gambar selesai, gambar disimpan dalam folder yang terstruktur berdasarkan **arah pandang** dan **waktu pengambilan gambar**. Penyimpanan gambar yang terstruktur memungkinkan manajemen data yang lebih efisien.

3.1.2 Struktur Folder dan Data Hasil Scraping

Setelah proses scraping selesai, gambar yang dihasilkan disusun dalam struktur folder berdasarkan **arah pandang** dan **kategori kondisi jalan**. Berikut adalah contoh struktur folder yang digunakan untuk menyimpan gambar:

- outdoor_90deg: Gambar yang diambil dengan sudut pandang 90° (menghadap ke arah kiri atau kanan jalan).
- outdoor_270deg: Gambar yang diambil dengan sudut pandang 270° (menghadap ke arah berlawanan, biasanya sisi kanan atau kiri jalan dari sudut pandang sebelumnya).

Struktur ini dirancang agar mudah diakses dan diorganisir berdasarkan arah pandang, memudahkan peneliti dalam proses pemilihan gambar untuk analisis lebih lanjut.

3.1.3 Labeling Dataset dengan Roboflow

Setelah gambar terkumpul, langkah selanjutnya adalah **labeling** untuk menandai objek-objek penting yang ada dalam gambar. Proses ini dilakukan dengan menggunakan **Roboflow**, sebuah platform berbasis web yang memungkinkan peneliti untuk memberikan anotasi visual pada objek-objek dalam gambar, seperti **jalan, kendaraan, pejalan kaki, rambu lalu lintas**, dan objek lainnya.

Langkah-langkah dalam proses labeling adalah sebagai berikut:

- Menandai objek: Peneliti memberi label pada objek-objek yang relevan dalam gambar, seperti kendaraan, pejalan kaki, atau objek lainnya. Setiap kategori objek diberi warna yang berbeda untuk mempermudah identifikasi visual (Haq, 2024).
- Eksport data: Data yang telah diberi label kemudian diekspor dalam format yang sesuai untuk digunakan dalam pelatihan model segmentasi semantik.

Proses pelabelan dataset dilakukan menggunakan platform Roboflow dengan pendekatan anotasi pixel-level untuk setiap kelas objek pada citra. Roboflow menyediakan antarmuka berbasis web yang memudahkan proses anotasi dan pengelolaan dataset segmentasi semantik. Contoh proses pelabelan citra jalanan menggunakan Roboflow ditunjukkan pada Gambar 3.1, di mana setiap piksel citra diberi label sesuai kelas objek seperti road, building, rider, dan traffic light.



Gambar 3. 1 Proses Pelabelan Citra Jalanan Menggunakan Roboflow

Gambar 3.1 memperlihatkan bahwa anotasi segmentasi semantik memiliki tingkat kompleksitas yang tinggi, terutama pada area objek yang saling tumpang tindih dan objek berukuran kecil. Objek seperti rider dan traffic light memerlukan ketelitian tinggi dalam penentuan batas piksel agar tidak terjadi kesalahan anotasi antar kelas.

Roboflow menyediakan dua pendekatan untuk proses labeling: AI-assisted labeling dan manual labeling. Masing-masing pendekatan memiliki kelebihan dan kelemahan yang perlu dipertimbangkan dalam konteks segmentasi semantik citra jalanan.

AI-Assisted Labeling pada Roboflow

AI-assisted labeling memanfaatkan model pembelajaran mesin untuk memberikan saran anotasi otomatis yang dapat mempercepat proses pelabelan.

Kelebihan AI-Assisted Labeling:

- Mempercepat proses anotasi untuk objek-objek besar dengan batas yang jelas seperti building, road, dan vegetation, sehingga mengurangi waktu labeling secara signifikan.
- Konsistensi anotasi yang lebih tinggi untuk kelas dominan karena algoritma AI menggunakan kriteria yang sama dalam setiap prediksi.
- Mengurangi beban kerja annotator dengan menyediakan mask awal yang dapat dikoreksi, sehingga annotator tidak perlu membuat anotasi dari awal.
- Efisien untuk dataset berskala besar dengan ratusan hingga ribuan citra yang memerlukan anotasi cepat.

Kelemahan AI-Assisted Labeling:

- Akurasi rendah untuk objek berukuran kecil dan dinamis seperti rider, bicycle, dan traffic light, di mana fitur AI sering gagal mengenali atau menghasilkan mask yang tidak presisi.
- Kesulitan menangani objek dengan batas ambigu seperti sidewalk yang memiliki tekstur dan warna mirip dengan road, sehingga saran anotasi sering tidak akurat.

- Ketergantungan pada kualitas model AI yang digunakan, yang mungkin belum dilatih dengan data konteks jalanan Indonesia sehingga kurang familiar dengan karakteristik visual lokal.
- Memerlukan koreksi manual yang ekstensif untuk objek kompleks, yang dapat mengurangi efisiensi waktu yang seharusnya diperoleh dari pendekatan otomatis.

Manual Labeling pada Roboflow

Manual labeling adalah proses anotasi yang sepenuhnya dilakukan oleh annotator manusia tanpa bantuan AI, memberikan kontrol penuh terhadap setiap piksel yang dilabeli.

Kelebihan Manual Labeling:

- Presisi tinggi dalam menentukan batas objek, terutama untuk objek kompleks dengan detail halus seperti rider yang memerlukan anotasi pada bagian helm, tangan, dan kaki.
- Fleksibilitas penuh dalam menangani kasus-kasus sulit seperti objek yang tumpang tindih, okludasi parsial, atau objek dengan pencahayaan yang tidak biasa.
- Kemampuan untuk membuat keputusan kontekstual berdasarkan pemahaman visual manusia yang tidak dapat dilakukan oleh AI, seperti membedakan sidewalk dari road berdasarkan fungsi area tersebut.

Kelemahan Manual Labeling:

- Waktu labeling yang sangat lama, terutama untuk dataset dengan lebih dari 900 citra dan multiple kelas objek yang harus dianotasi secara pixel-level.
- Risiko ketidakkonsistenan anotasi akibat faktor manusia seperti kelelahan, interpretasi subjektif terhadap batas objek, dan perubahan kriteria labeling seiring waktu.
- Beban kerja yang tinggi untuk satu annotator yang harus menangani seluruh dataset, meningkatkan potensi kesalahan dan penurunan kualitas anotasi pada citra-citra yang dilabeli di tahap akhir.

- Subjektivitas dalam penentuan batas objek, terutama pada area dengan transisi gradual seperti batas antara sidewalk dan road atau antara vegetation dan building.

Meskipun Roboflow menyediakan fitur AI-assisted labeling, pada penelitian ini fitur tersebut tidak digunakan secara penuh karena hasil anotasi otomatis belum mampu menghasilkan batas objek yang presisi pada konteks citra jalanan Indonesia. Contoh area anotasi yang sulit dilabeli secara otomatis ditunjukkan pada Gambar 3.2, di mana objek berukuran kecil dan dinamis menunjukkan batas yang ambigu sehingga tetap memerlukan koreksi manual.



Gambar 3. 2 Contoh Tantangan Pelabelan pada Objek Berukuran Kecil

Proses anotasi dilakukan secara manual terhadap lebih dari 900 citra oleh satu annotator. Kondisi ini menyebabkan proses labeling menjadi sangat memakan waktu dan berpotensi menimbulkan ketidakkonsistenan anotasi, khususnya pada kelas minor dengan jumlah data terbatas. Keterbatasan dalam proses labeling ini berdampak pada kualitas ground truth dan distribusi data antar kelas, yang selanjutnya dianalisis pada bab hasil dan pembahasan melalui evaluasi kegagalan segmentasi.

3.1.4 Variasi Data yang Dikumpulkan

Proses scraping ini menghasilkan data yang mencakup beragam kondisi jalan di Surabaya. Variasi data ini penting untuk memastikan model segmentasi semantik dapat mengatasi kondisi dunia nyata yang beragam. Beberapa variasi dalam dataset ini meliputi:

- Jenis jalan: Jalan utama yang lebar, jalan sempit, hingga gang-gang kecil.
- Permukaan jalan: Jalan yang memiliki berbagai permukaan, seperti aspal kasar, beton, dan kerikil.
- Aktivitas jalan: Keberadaan kendaraan yang parkir sembarangan atau pedagang kaki lima yang menghalangi jalan atau trotoar.

3.2 Pengolahan Data

Pengolahan data adalah bagian kritis dari pipeline penelitian ini yang mencakup pemrosesan citra infrastruktur jalan Indonesia untuk melatih model segmentasi semantik **SegFormer-R101**.

3.2.1 Preprocessing Data (Persiapan Gambar)

Tahapan **preprocessing** bertujuan untuk menyiapkan gambar mentah yang dikumpulkan melalui **web scraping** menggunakan Selenium agar sesuai dengan persyaratan input untuk model segmentasi semantik yang digunakan. Proses ini tidak hanya mencakup perubahan ukuran gambar, tetapi juga pemrosesan tambahan yang memungkinkan model untuk mengenali berbagai jenis objek dalam citra jalanan yang bervariasi (Kim et al., 2023).

1. Perubahan Ukuran Gambar:

Gambar-gambar yang dikumpulkan memiliki berbagai macam dimensi yang tidak konsisten, yang dapat menyebabkan gangguan pada proses pelatihan model. Untuk memastikan pemrosesan yang efisien, gambar-gambar tersebut diubah ukurannya menggunakan **resizing dengan interpolasi bicubic**. Teknik interpolasi bicubic mempertahankan struktur visual dan mengurangi distorsi yang dapat terjadi selama perubahan skala. Setiap gambar akan diubah menjadi ukuran yang sesuai dengan input layer model seperti **512x512** atau **1024x1024** (Virbukaitė & Bernatavičienė, 2024).

2. Pemotongan Gambar:

Tile-based cropping adalah teknik yang digunakan untuk memotong gambar besar menjadi patch-atau-tiles yang lebih kecil. Teknik ini memungkinkan model untuk memproses citra secara lebih fokus dan mengurangi dampak noise dari objek yang tidak relevan. Dimensi patch bisa disesuaikan, misalnya, menjadi patch berukuran 256x256 piksel.

Proses ini juga membantu dalam menangani overfitting, karena model tidak hanya berfokus pada satu area besar citra, tetapi juga pada bagian-bagian lebih kecil yang relevan(Jenkinson & Arandjelović, 2024).

3. Normalisasi:

Data gambar yang dimasukkan ke dalam model harus dalam bentuk **nilai piksel yang dinormalisasi** dalam rentang tertentu, seperti **[0, 1]** atau **[-1, 1]**, untuk menghindari masalah perbedaan skala antara input dan bobot model(Mukhametzhanov, 2023). Normalisasi ini dilakukan dengan membagi nilai RGB setiap piksel dengan 255 atau dengan menerapkan mean dan standar deviasi gambar yang dihitung dari dataset pelatihan, menggunakan rumus standar:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

Di mana x adalah nilai piksel gambar, μ adalah nilai rata-rata piksel untuk dataset, dan σ adalah deviasi standar piksel.

3.2.2 Domain Adaptation

Domain Adaptation mengacu pada teknik dalam **transfer learning** yang digunakan untuk mengatasi masalah perbedaan antara **domain sumber (synthetic data)** dan **domain target (real-world data)** yang dihadapi oleh model segmentasi semantik. Secara lebih rinci, proses ini melibatkan teknik-teknik yang memungkinkan **penyesuaian distribusi** dan **karakteristik visual** antara kedua domain agar model yang telah dilatih pada domain sintesis tetap efektif saat diterapkan pada data dunia nyata.

Kerangka Kerja Domain Adaptation

1. Transfer Learning:

Transfer learning adalah fondasi dasar dari domain adaptation. Proses ini memanfaatkan pengetahuan yang telah diperoleh model dari dataset **sumber** (biasanya dataset sintesis) untuk diterapkan pada dataset **target** (data dunia nyata yang memiliki karakteristik yang lebih kompleks)(Pyliaididis et al., 2024). Model yang awalnya dilatih pada data sintesis (misalnya **GTA5** atau **SYNTHIA**) memiliki representasi fitur yang baik untuk objek yang relatif sederhana. Namun, karena perbedaan visual

yang signifikan, seperti pencahayaan, tekstur, atau keberagaman objek, model memerlukan penyesuaian untuk bekerja pada **data jalanan Indonesia** yang jauh lebih bervariasi.

2. **Optimal Transport (OT):**

Salah satu teknik yang digunakan untuk menyeimbangkan distribusi antara domain sumber dan target adalah Optimal Transport (OT). OT adalah teori matematis yang digunakan untuk mengukur jarak atau perbedaan distribusi antara dua himpunan distribusi probabilitas, yang sering digunakan dalam domain adaptation untuk menyelaraskan distribusi fitur. Secara khusus, OT memungkinkan kita untuk memetakan data sintetis (dalam bentuk distribusi fitur) ke dalam domain target, dengan tujuan agar representasi fitur yang diperoleh dari domain sumber lebih mirip dengan fitur dalam domain target (Guo et al., 2024).

$$D_{OT}(P, Q) = \min_{\pi \in \Pi(P, Q)} \int_{x \times y} c(x, y), \pi(x, y), dx, dy$$

Di mana P dan Q adalah distribusi sumber dan target, π adalah transport map yang mencocokkan elemen-elemen x dan y dalam domain sumber dan target, dan $c(x, y)$ adalah biaya transportasi antara x dan y . Proses ini memastikan distribusi yang lebih seragam antara domain-domain tersebut.

3. **Fine-Tuning:**

Setelah **transfer learning** dan **penyelarasan distribusi** menggunakan **Optimal Transport**, model kemudian di **fine-tune** menggunakan subset data dunia nyata untuk menyesuaikan parameter model lebih lanjut. Pada tahap ini, model yang sudah dilatih pada data sintetis diupdate dengan data dunia nyata, meskipun dalam jumlah yang lebih kecil, untuk memperbaiki representasi fitur yang lebih relevan dengan domain target (Sun et al., 2023).

Model segmentasi semantik yang digunakan dalam penelitian ini adalah SegFormer-R101, sebuah arsitektur berbasis Vision Transformer yang menggunakan encoder hierarkis dan decoder yang sederhana berupa Multilayer Perceptron (MLP). Model ini dipilih karena mampu bekerja dengan baik meskipun ada perbedaan antara data sintetis dan data nyata, terutama dalam hal

ciri visual, serta lebih efisien dalam penggunaan komputer dibandingkan model segmentasi yang menggunakan konvolusi biasa.

SegFormer-R101 tidak memerlukan kode secara eksplisit dan menggunakan metode embedding patch yang tumpang tindih, sehingga mampu menangkap konteks secara global dan lokal sekaligus.

Sifat ini membuat SegFormer-R101 cocok untuk digunakan dalam segmentasi citra jalan di Indonesia yang memiliki variasi cahaya, tekstur, dan kondisi lingkungan yang beragam.

Untuk informasi lebih lengkap mengenai struktur SegFormer-R101, termasuk cara kerja encoder, decoder, dan alur pengolahan data dari gambar masukan hingga hasil segmentasi, akan dijelaskan pada Subbab 3.3.

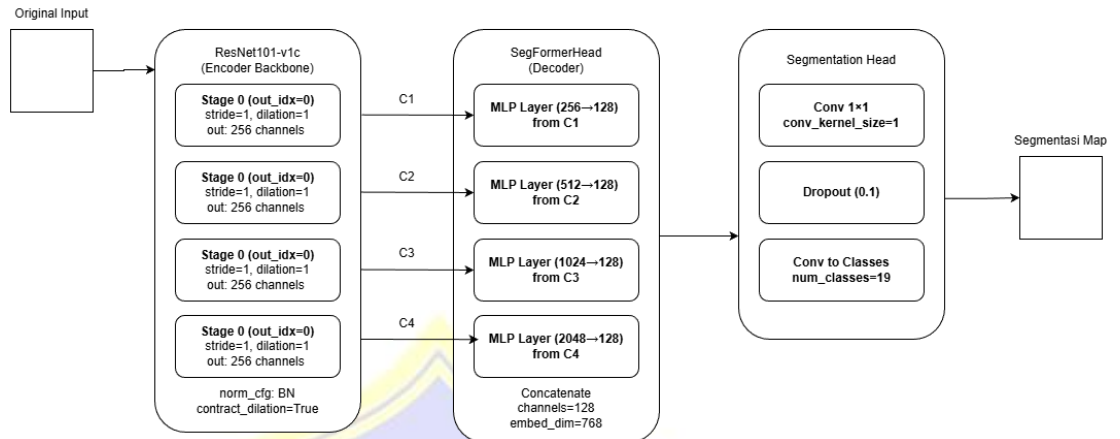
3.3 Model Segmentasi Semantik

3.3.1 Arsitektur SegFormer-R101

Arsitektur model yang digunakan dalam penelitian ini dirancang untuk menyelesaikan permasalahan segmentasi semantik pada citra jalanan, yaitu tugas klasifikasi setiap piksel citra ke dalam kelas semantik tertentu. Berbeda dengan klasifikasi citra atau deteksi objek, segmentasi semantik menuntut model untuk menghasilkan prediksi yang **presisi secara spasial** dan **konsisten secara semantik**, karena setiap kesalahan satu piksel dapat memengaruhi kualitas peta segmentasi secara keseluruhan.

Untuk memenuhi kebutuhan tersebut, digunakan arsitektur **encoder-decoder** yang terdiri dari **ResNet-101** sebagai **encoder** dan **SegFormer Head** sebagai **decoder**. Encoder bertugas mengekstraksi representasi fitur visual dari citra masukan, sedangkan decoder bertugas menggabungkan fitur-fitur tersebut dan menghasilkan peta segmentasi akhir. Arsitektur ini dipilih karena mampu mengintegrasikan **detail lokal** dan **konteks global**, yang merupakan dua aspek krusial dalam segmentasi citra jalanan.

Berikut diagram arsitektur dari SegFormer-R101



Gambar 3. 3 Arsitektur SegFormer-Resnet101(Xie et al., 2021)

1. Input Citra

Citra masukan merupakan citra digital berwarna dengan format RGB yang terdiri dari tiga kanal warna, yaitu merah, hijau, dan biru. Setiap piksel pada citra masukan menyimpan informasi intensitas warna yang merepresentasikan kondisi visual lingkungan jalanan, seperti permukaan jalan, bangunan, kendaraan, serta objek pendukung lainnya.

Resolusi citra masukan dipertahankan pada tahap awal pemrosesan. Tidak dilakukan proses penurunan resolusi atau pengaburan spasial sebelum citra diproses oleh encoder. Keputusan ini didasarkan pada karakteristik tugas segmentasi semantik yang sangat bergantung pada informasi spasial. Jika detail spasial hilang pada tahap awal, maka model akan kesulitan merekonstruksi batas objek yang tajam pada tahap decoding.

Dengan demikian, citra masukan berfungsi sebagai representasi visual mentah yang menyediakan seluruh informasi awal bagi encoder untuk membentuk feature map bertingkat.

2. Encoder: ResNet-101

Encoder pada arsitektur ini menggunakan ResNet-101, yaitu jaringan convolutional residual dengan kedalaman 101 lapisan.

Encoder berfungsi sebagai komponen utama yang mengekstraksi fitur visual dari citra masukan secara bertahap dan hierarkis.

Keunggulan utama ResNet-101 terletak pada penggunaan residual connection, yaitu mekanisme penjumlahan antara input dan output suatu blok konvolusi. Mekanisme ini memungkinkan jaringan mempelajari representasi fitur yang sangat dalam tanpa mengalami degradasi performa, sekaligus memastikan bahwa informasi dari lapisan awal tetap dapat dimanfaatkan oleh lapisan yang lebih dalam.

Encoder ini dimodifikasi agar sepenuhnya bersifat convolutional, sehingga cocok digunakan untuk tugas dense prediction seperti segmentasi semantik. Encoder menghasilkan empat feature map utama, yaitu C1, C2, C3, dan C4, yang masing-masing merepresentasikan level abstraksi fitur yang berbeda.

3. Stage 1 – Feature Map C1 (Fitur Tingkat Rendah)

Stage pertama menghasilkan feature map C1 dengan resolusi spasial yang relatif tinggi, yaitu sekitar seperempat dari resolusi citra masukan. Feature map ini mengandung informasi visual tingkat rendah yang bersifat lokal dan detail.

Informasi yang terkandung pada C1 meliputi:

- Tepi objek
- Perubahan intensitas warna
- Perbedaan tekstur antar area citra

Fitur-fitur ini belum bersifat semantik, namun sangat penting untuk menjaga ketepatan batas objek. Dalam segmentasi semantik, batas antar kelas seperti jalan dan trotoar atau bangunan dan langit sering kali ditentukan oleh perbedaan tekstur dan tepi visual yang halus. Jika informasi ini tidak dipertahankan, hasil segmentasi akan terlihat kabur dan tidak presisi.

Oleh karena itu, C1 berfungsi sebagai penyedia detail spasial yang menjadi acuan utama dalam proses decoding, khususnya untuk memperbaiki batas objek pada peta segmentasi akhir.

4. Stage 2 – Feature Map C2 (Fitur Tingkat Menengah Awal)

Stage kedua menghasilkan feature map C2 dengan resolusi sekitar satu per delapan dari resolusi citra masukan. Pada tahap ini, jaringan mulai membentuk representasi fitur tingkat menengah yang menggabungkan detail spasial dengan struktur objek.

Feature map C2 mulai menangkap:

- Bentuk dasar objek
- Pola visual yang berulang
- Struktur geometris sederhana

Representasi fitur pada tahap ini sudah mulai mengarah ke pemahaman objek, meskipun masih mempertahankan cukup banyak detail spasial. C2 berperan sebagai penghubung penting antara fitur detail tingkat rendah (C1) dan fitur semantik tingkat tinggi (C3 dan C4).

Keberadaan C2 membantu model dalam mengenali objek berukuran sedang dan meningkatkan konsistensi segmentasi pada area yang memiliki variasi bentuk dan tekstur.

5. Stage 3 – Feature Map C3 (Fitur Semantik Tingkat Tinggi)

Stage ketiga menghasilkan feature map C3 dengan resolusi sekitar satu per enam belas dari resolusi citra masukan. Feature map ini mengandung representasi fitur semantik tingkat tinggi yang berfokus pada struktur global objek dan hubungan antar objek dalam suatu adegan.

Pada tahap ini, receptive field jaringan menjadi lebih luas, sehingga model mampu memahami konteks visual yang lebih besar. C3 memungkinkan model untuk:

- Membedakan kelas dengan kemiripan visual
- Memahami hubungan spasial antar objek
- Menentukan kelas berdasarkan konteks lingkungan

Sebagai contoh, perbedaan antara jalan dan trotoar sering kali tidak hanya ditentukan oleh tekstur, tetapi juga oleh posisi relatif dan konteks lingkungan. C3 menyediakan informasi semantik yang dibutuhkan untuk pengambilan keputusan tersebut.

6. Stage 4 – Feature Map C4 dengan Dilated Convolution (Konteks Global)

Stage keempat menghasilkan feature map C4 dengan menggunakan dilated convolution. Pada tahap ini, resolusi spasial feature map dipertahankan, namun receptive field jaringan diperluas secara signifikan.

Penggunaan dilated convolution memungkinkan model menangkap konteks global tanpa melakukan downsampling tambahan. Hal ini sangat penting dalam segmentasi semantik citra jalanan, karena banyak kelas objek ditentukan oleh konteks global, bukan hanya oleh ciri lokal.

C4 menyediakan pemahaman menyeluruh terhadap adegan, seperti distribusi objek besar, struktur lingkungan, dan hubungan skala besar antar elemen visual. Feature map ini menjadi sumber utama informasi konteks global bagi decoder.

7. Decoder: SegFormer Head sebagai Mekanisme Integrasi Fitur

Decoder pada arsitektur ini menggunakan SegFormer Head, yang dirancang untuk mengintegrasikan feature map multi-skala dari encoder secara efisien. Decoder ini menggunakan pendekatan berbasis Multi-Layer Perceptron (MLP) untuk menyatukan informasi dari berbagai level abstraksi.

Pendekatan ini berbeda dari decoder konvolusional klasik yang umumnya menggunakan serangkaian upsampling dan konvolusi. SegFormer Head lebih menekankan pada penyamaan dimensi fitur dan penggabungan informasi secara global, sehingga lebih ringan dan stabil.

8. Penyamaan Dimensi Fitur Antar Level

Setiap feature map keluaran encoder memiliki jumlah kanal yang berbeda. Jika langsung digabungkan, feature map dengan dimensi besar akan mendominasi representasi gabungan. Oleh karena itu, dilakukan penyamaan dimensi kanal agar setiap level fitur memiliki kontribusi yang seimbang.

Tahap ini memastikan bahwa:

- Informasi detail dari C1 tetap berpengaruh
- Informasi konteks dari C4 tidak mendominasi secara berlebihan
- Proses fusi fitur berlangsung secara adil dan stabil

9. Penyesuaian Resolusi Spasial

Karena setiap feature map memiliki resolusi spasial yang berbeda, dilakukan proses penyesuaian resolusi agar seluruh feature map memiliki ukuran spasial yang sama. Proses ini bertujuan untuk menjaga korespondensi spasial antar fitur sebelum digabungkan.

Tanpa penyesuaian resolusi, penggabungan fitur akan menghasilkan ketidaksesuaian posisi piksel, yang dapat menyebabkan kesalahan pada batas objek dalam peta segmentasi akhir.

10. Penggabungan Fitur Multi-Skala

Setelah dimensi kanal dan resolusi spasial diseragamkan, seluruh feature map digabungkan menjadi satu representasi terpadu. Proses ini mengintegrasikan:

- Detail spasial dari fitur tingkat rendah
- Informasi struktural dari fitur tingkat menengah
- Konteks global dari fitur tingkat tinggi

Penggabungan ini memungkinkan model untuk menghasilkan segmentasi yang akurat pada berbagai skala objek.

11. Penyatuan Representasi dan Penstabilan Fitur

Feature map hasil penggabungan kemudian diproses untuk menyatukan informasi dari berbagai skala dan menstabilkan distribusi fitur. Tahap ini penting untuk memastikan bahwa representasi fitur yang dihasilkan siap digunakan untuk proses klasifikasi piksel.

Stabilisasi fitur pada tahap ini berkontribusi langsung terhadap konsistensi hasil segmentasi dan kestabilan proses pelatihan.

12. Segmentation Head

Segmentation head merupakan komponen akhir dari arsitektur yang bertugas mengubah representasi fitur hasil decoder menjadi peta

kelas segmentasi. Pada tahap ini, setiap piksel diklasifikasikan berdasarkan informasi fitur yang telah dikombinasikan dari berbagai level abstraksi.

Segmentation head dirancang untuk mempertahankan kesesuaian spasial antara peta segmentasi dan citra masukan, sehingga hasil segmentasi dapat merepresentasikan struktur visual secara akurat.

13. Peta Segmentasi sebagai Keluaran Akhir

Keluaran akhir dari arsitektur ini berupa peta segmentasi semantik, di mana setiap piksel pada citra masukan diberi label kelas tertentu. Peta segmentasi ini merepresentasikan pemahaman menyeluruh model terhadap isi visual citra jalanan.

Peta ini selanjutnya digunakan sebagai dasar evaluasi performa model dan analisis kualitas segmentasi.

3.4 Strategi Pelatihan dan Optimisasi Model

Pelatihan model segmentasi semantik dirancang secara sistematis dengan mempertimbangkan kompleksitas visual citra jalanan perkotaan di Surabaya yang mencakup variasi kondisi pencahayaan, kepadatan objek, serta perbedaan skala antar objek. Lingkungan jalan perkotaan memiliki karakteristik dinamis yang menuntut model untuk mampu mengekstraksi fitur kontekstual global sekaligus mempertahankan presisi pada objek berukuran kecil.

Oleh karena itu, proses pelatihan tidak hanya difokuskan pada pencapaian nilai akurasi tinggi, tetapi juga pada stabilitas konvergensi model serta kemampuan generalisasi terhadap data dunia nyata yang tidak sepenuhnya terstruktur.

3.4.1 Pemilihan Optimizer dan Skema Learning Rate

Optimizer yang digunakan dalam pelatihan model adalah AdamW dengan learning rate awal sebesar 1×10^{-4} . Pemilihan AdamW didasarkan pada kemampuannya dalam mengoptimalkan parameter model dengan memisahkan mekanisme weight decay dari pembaruan gradien, sehingga mampu mencegah overfitting secara lebih efektif dibandingkan optimizer Adam konvensional. Hal ini menjadi sangat penting mengingat arsitektur SegFormer-R101 memiliki

jumlah parameter yang besar serta memanfaatkan pretrained backbone berbasis ResNet-101.

Untuk mengatur dinamika pembaruan parameter selama pelatihan, digunakan learning rate scheduler polynomial decay dengan strategi warmup pada fase awal pelatihan. Warmup dilakukan selama 500 iterasi pertama untuk meningkatkan nilai learning rate secara bertahap dari nilai kecil menuju learning rate awal yang telah ditentukan. Strategi ini bertujuan untuk menghindari instabilitas gradien yang sering terjadi pada awal pelatihan, terutama karena bagian decode head model diinisialisasi secara acak sementara backbone menggunakan bobot hasil pretraining.

3.4.2 Durasi Pelatihan dan Strategi Iterasi

Total iterasi pelatihan ditetapkan sebanyak 30.000 iterasi. Penentuan jumlah iterasi ini didasarkan pada pertimbangan keseimbangan antara waktu komputasi dan kebutuhan model untuk mencapai konvergensi yang optimal. Jumlah iterasi yang terlalu sedikit berisiko menyebabkan model belum sepenuhnya mempelajari distribusi data, sementara iterasi yang terlalu banyak berpotensi menyebabkan overfitting terhadap data pelatihan.

Evaluasi performa model dilakukan secara berkala setiap 500 iterasi menggunakan dataset validasi. Evaluasi berkala ini memungkinkan pemantauan perkembangan performa model dari waktu ke waktu serta membantu mengidentifikasi fase-fase kritis dalam proses pembelajaran, seperti fase pembelajaran cepat di awal pelatihan dan fase konvergensi di akhir pelatihan. Selain itu, penyimpanan checkpoint model dilakukan setiap 1.000 iterasi untuk mendokumentasikan evolusi performa model dan memungkinkan pemulihan pelatihan jika terjadi gangguan teknis.

3.5 Rancangan atau Design yang diusulkan

Rancangan atau desain yang diusulkan dalam penelitian ini adalah suatu alur kerja yang mendeskripsikan langkah-langkah dari pengumpulan data, persiapan data, pelatihan model, hingga evaluasi hasil segmentasi semantik. Rancangan ini akan menggunakan pendekatan **domain adaptation** untuk

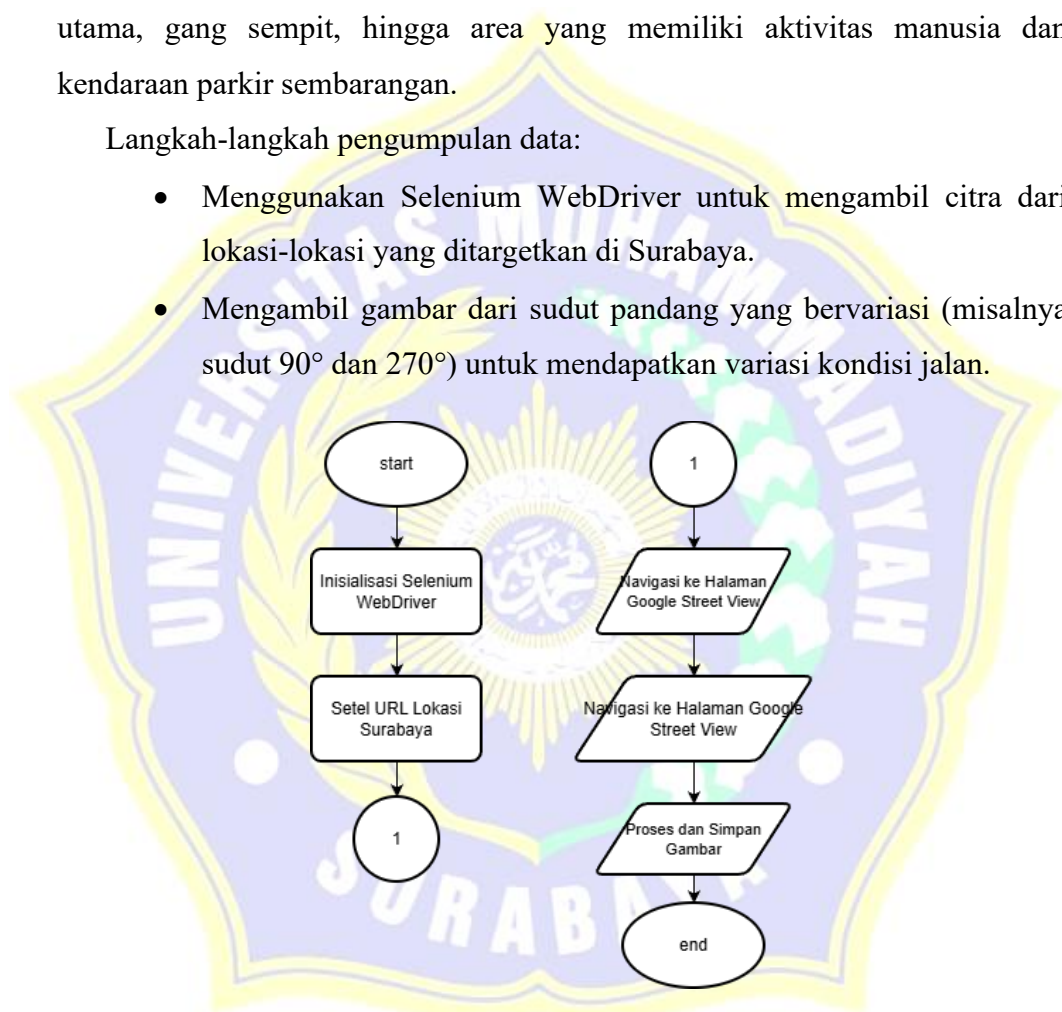
mengatasi perbedaan karakteristik antara data sintesis (seperti GTA5 dan Synthia) dan data dunia nyata (seperti citra jalan Surabaya).

1. Pengumpulan Data

Alur kerja dimulai dengan **pengumpulan data citra jalanan** yang representatif dari Surabaya. Data ini akan dikumpulkan melalui teknik **web scraping** menggunakan alat otomasi Selenium untuk mengambil gambar dari **Google Street View** yang mencakup berbagai kondisi jalan, seperti jalan utama, gang sempit, hingga area yang memiliki aktivitas manusia dan kendaraan parkir sembarangan.

Langkah-langkah pengumpulan data:

- Menggunakan Selenium WebDriver untuk mengambil citra dari lokasi-lokasi yang ditargetkan di Surabaya.
- Mengambil gambar dari sudut pandang yang bervariasi (misalnya sudut 90° dan 270°) untuk mendapatkan variasi kondisi jalan.



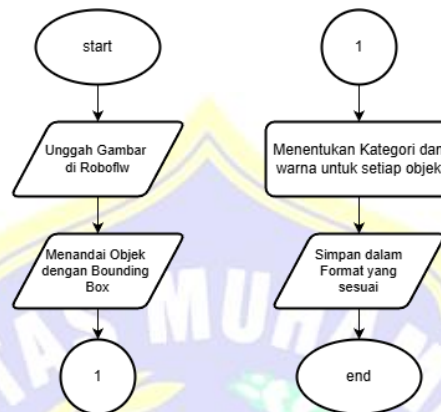
Gambar 3. 4 Flowchart Pengumpulan Data

2. Labeling Dataset

Setelah citra terkumpul, tahap berikutnya adalah **labeling dataset**, di mana objek-objek penting dalam citra (seperti jalan, kendaraan, pejalan kaki, dan rambu lalu lintas) diberi label menggunakan platform **Roboflow**.

Langkah-langkah labeling dataset:

- Menandai objek-objek pada citra dan memberikan kategori warna berbeda untuk setiap kelas objek.
- Mengekspor data yang telah dilabeli ke dalam format yang dapat digunakan untuk pelatihan model.



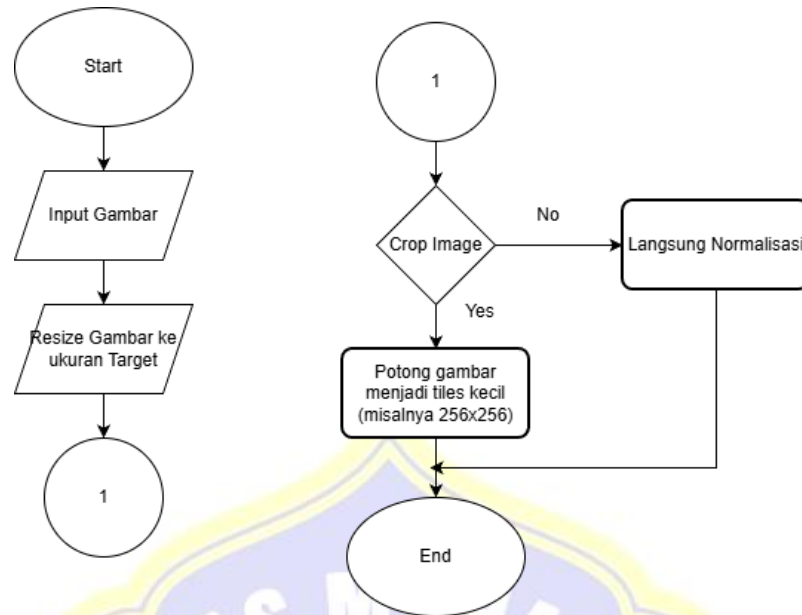
Gambar 3. 5 Flowchart Labeling Dataset

3. Preprocessing Data

Data yang telah dikumpulkan dan diberi label harus melalui tahap **preprocessing** untuk menyiapkan gambar agar sesuai dengan format yang dibutuhkan oleh model segmentasi semantik.

Langkah-langkah preprocessing:

- **Pengubahan Ukuran Gambar:** Mengubah ukuran gambar menjadi format yang konsisten, misalnya 512x512 piksel atau 1024x1024 piksel, agar sesuai dengan input model.
- **Pemotongan Gambar:** Memotong gambar besar menjadi **tiles** yang lebih kecil (misalnya 256x256 piksel), untuk mempercepat pemrosesan dan mengurangi noise.
- **Normalisasi:** Menormalisasi piksel gambar agar berada dalam rentang nilai $[0, 1]$ atau $[-1, 1]$.

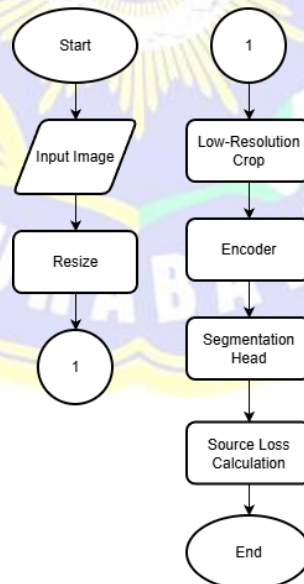


Gambar 3. 6 Flowchart Preprocessing Data

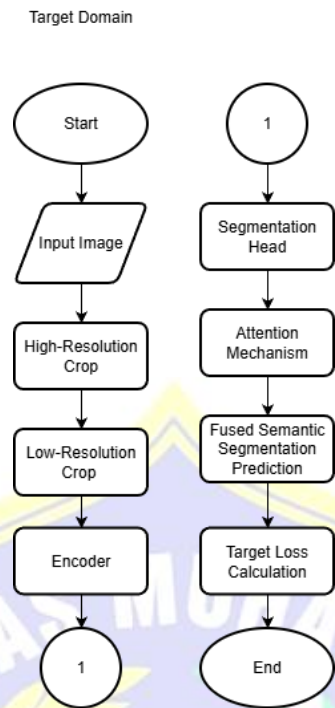
4. Pelatihan Model Segmentasi

Setelah data dipersiapkan, tahap selanjutnya adalah **pelatihan model segmentasi semantik**. Model utama yang akan diuji dalam penelitian ini adalah **SegFormer-R101**.

Source Domain



Gambar 3. 7 Flowchart Train Source Domian



Gambar 3. 8 Flowchart Train Target Domain

Pada flowchart yang menggambarkan proses pelatihan untuk Target Domain dan Source Domain, kedua alur ini mengikuti prinsip dasar HRDA (High-Resolution Domain Adaptation), namun dengan beberapa perbedaan penting dalam cara data diproses dan hasil yang diharapkan.

Untuk Target Domain, proses dimulai dengan memasukkan gambar dari domain yang dituju. Gambar tersebut kemudian dipotong dengan dua jenis crop, yaitu High-Resolution (HR) Crop dan Low-Resolution (LR) Crop. HR Crop digunakan untuk mengambil detail yang halus dan penting, terutama untuk mengenali objek kecil atau rumit, sedangkan LR Crop memberikan gambaran umum konteks gambar tanpa fokus pada detail spesifik. Setelah dipotong, gambar dianalisis oleh encoder untuk mengekstrak fitur yang relevan. Fitur tersebut kemudian diproses oleh Segmentation Head untuk menghasilkan prediksi segmentasi. Selanjutnya, Attention Mechanism diterapkan untuk memberi perhatian lebih pada bagian gambar yang penting, sehingga hasil prediksi menjadi lebih akurat. Setelah proses segmentasi selesai, hasil prediksi digabungkan untuk membuat prediksi segmentasi yang lebih lengkap dan akurat. Hasil Target Loss Calculation dibuat untuk

membandingkan prediksi dengan gambar benar (ground truth), dan hasil ini digunakan untuk memperbaiki model.

Sementara itu, untuk Source Domain, gambar dari domain sumber diproses dengan metode yang lebih sederhana. Setelah dimasukkan ke dalam model, ukuran gambar diperkecil agar lebih mudah diproses. Gambar lalu dipotong dengan LR Crop, yang hanya memberikan gambaran umum tanpa mengambil detail yang halus. Potongan gambar ini kemudian diproses melalui encoder untuk mengekstrak fitur dan diteruskan ke Segmentation Head untuk membuat prediksi segmentasi. Setelah itu, perhitungan Source Loss Calculation dilakukan untuk membandingkan hasil prediksi dengan gambar benar dari domain sumber, sehingga model dapat dioptimalkan. Berbeda dengan Target Domain, pada Source Domain tidak menggunakan Attention Mechanism, yang berarti model hanya fokus pada informasi umum dan tidak memberi perhatian khusus pada bagian tertentu dari gambar.

Perbedaan utama antara kedua proses ini adalah dalam penggunaan LR Crop dan Attention Mechanism pada Target Domain, yang memungkinkan model menangkap detail gambar secara lebih rinci dan memperhatikan bagian yang lebih penting. Sementara itu, pada Source Domain hanya menggunakan LR Crop dengan fokus pada konteks keseluruhan gambar, sehingga menghasilkan prediksi yang umum. Selain itu, pada Target Domain, hasil prediksi segmentasi digabungkan untuk meningkatkan akurasi, sedangkan pada Source Domain, perhitungan loss dilakukan setelah segmentasi selesai. Dengan demikian, proses untuk Target Domain lebih kompleks dan memberikan hasil segmentasi yang lebih presisi dan akurat dibandingkan dengan Source Domain.

5. Domain Adaptation

Untuk mengatasi perbedaan karakteristik antara data sintesis dan data dunia nyata, teknik domain adaptation akan diterapkan. Pendekatan ini melibatkan dua langkah utama:

- **Transfer Learning:** Model yang telah dilatih pada dataset sintesis akan ditransfer untuk digunakan pada data dunia nyata. Penyesuaian dilakukan agar model dapat beradaptasi dengan kondisi visual jalanan Surabaya.

- Optimal Transport (OT): Teknik ini akan digunakan untuk menyelaraskan distribusi fitur antara dataset sintetis dan data dunia nyata, memastikan bahwa representasi fitur pada data dunia nyata lebih mirip dengan data pelatihan sintetis.

6. Evaluasi Model

Setelah pelatihan selesai, model yang telah dilatih akan dievaluasi menggunakan dua metrik utama: **Akurasi** dan **Mean Intersection over Union (mIoU)**. Evaluasi ini dilakukan untuk mengukur kemampuan model dalam mengenali objek di citra jalan Surabaya, seperti kendaraan, pejalan kaki, dan rambu lalu lintas.

Langkah-langkah evaluasi model:

- Menggunakan data uji yang terdiri dari gambar yang belum pernah dilihat sebelumnya oleh model.
- Menghitung mIoU untuk mengukur seberapa baik model dalam yang ada dalam gambar (Plaksyvyi et al., 2023).
- mendeteksi dan membedakan objek-objek dalam citra.
- Menghitung akurasi pixel untuk mengukur persentase piksel yang terklasifikasi dengan benar oleh model.
-

3.6 Rencana Evaluasi

Pada subbab ini, dijelaskan mengenai rencana evaluasi yang akan dilakukan untuk mengukur kinerja model segmentasi semantik yang diterapkan pada citra infrastruktur jalan di Surabaya. Evaluasi ini bertujuan untuk menilai sejauh mana model dapat beradaptasi dengan kondisi dunia nyata, khususnya mengatasi pergeseran domain antara data sintetis dan data nyata. Evaluasi ini juga akan mengukur seberapa baik model dalam mengenali objek di citra jalanan dengan karakteristik visual yang berbeda.

1. Evaluasi dengan Akurasi (Accuracy)

Akurasi mengukur sejauh mana model berhasil mengklasifikasikan setiap piksel dalam citra dengan benar. Pada evaluasi ini, akurasi dihitung dengan cara membandingkan jumlah piksel yang terklasifikasi dengan benar oleh model dengan total piksel

Rumus Akurasi:

$$\left[\text{Akurasi} = \frac{\text{Jumlah Piksel yang Terklasifikasi dengan Benar}}{\text{Jumlah Piksel Total}} \right]$$

Penjelasan Sederhana:

- Piksel yang Terklasifikasi dengan Benar adalah jumlah piksel yang diprediksi dengan benar oleh model (misalnya, kendaraan yang terdeteksi sebagai kendaraan).
- Piksel Total adalah semua piksel dalam gambar yang dianalisis (misalnya, jumlah total piksel di gambar jalanan yang diambil).

2. Evaluasi dengan Mean Intersection over Union (mIoU)

mIoU mengukur kinerja model dalam memprediksi area yang benar sesuai dengan kelas yang ada pada gambar. mIoU dihitung dengan cara mengukur tumpang tindih antara prediksi model dengan ground truth, dibagi dengan total area yang seharusnya terdeteksi. Nilai mIoU yang lebih tinggi menunjukkan bahwa model dapat mendeteksi objek dengan lebih baik dan lebih akurat (Terekovskiy et al., 2023).

Rumus mIoU:

$$\left[\text{mIoU} = \frac{\text{Jumlah Area yang Benar Diprediksi}}{\text{Jumlah Area yang Benar Diprediksi} + \text{Area yang Salah Diprediksi}} \right]$$

Penjelasan Sederhana:

- Jumlah Area yang Benar Diprediksi adalah bagian gambar yang berhasil terdeteksi dengan benar oleh model.
- Jumlah Area yang Salah Diprediksi adalah bagian gambar yang seharusnya terdeteksi oleh model tetapi tidak terdeteksi atau terdeteksi dengan salah.

3.6.1 Pemantauan Proses Pelatihan dan Visualisasi Grafik

Untuk memastikan proses latihan model berjalan sesuai dengan metode yang sudah ditentukan, dilakukan pengawasan secara teratur terhadap beberapa parameter utama selama seluruh tahapan latihan. Pengawasan ini menjadi bagian penting dari metode penelitian, bukan hanya sebagai penunjang visualisasi, sehingga setiap perubahan kinerja model dapat ditelusuri secara berurutan dan dianalisis secara objektif.

Pengawasan tidak hanya fokus pada hasil akhir kinerja model, tetapi juga pada cara model belajar sepanjang proses, termasuk kecepatan konvergensi, kestabilan optimisasi, serta cara model merespons pengaturan laju pembelajaran. Dengan demikian, proses latihan dapat dinilai secara menyeluruh, baik dari aspek angka maupun dari pola perubahan kinerja selama proses berlangsung.

Selama latihan, beberapa parameter dicatat secara rutin setiap beberapa iterasi, seperti nilai loss pelatihan, laju pembelajaran, serta indikator evaluasi pada dataset validasi, yaitu mean Intersection over Union (mIoU), mean Accuracy (mAcc), dan overall Accuracy (aAcc). Pencatatan parameter ini dilakukan secara teratur untuk setiap tahap pelatihan agar data bisa dibandingkan antar iterasi. Data hasil pencatatan kemudian ditampilkan dalam bentuk grafik untuk membantu analisis, sehingga peneliti bisa mengidentifikasi tren dan pola pembelajaran.

Grafik ini memungkinkan pengamatan terhadap penurunan atau stagnasi loss pelatihan, konsistensi perubahan laju pembelajaran dengan rencana yang telah dibuat, serta perkembangan kinerja segmentasi pada dataset validasi. Dengan menganalisis grafik ini, fase-fase penting dalam pelatihan, seperti fase belajar cepat di awal dan fase konvergensi di akhir, bisa diidentifikasi dengan lebih jelas.

3.6.2 Analisis Grafik Training Loss dan Learning Rate

Grafik training loss digunakan untuk menunjukkan bagaimana nilai fungsi kerugian berubah selama proses pelatihan model setiap kali model melakukan iterasi. Nilai loss menunjukkan seberapa besar kesalahan antara hasil yang diperoleh model dengan hasil yang benar. Jika nilai loss terus menurun secara stabil, berarti proses optimisasi bekerja dengan baik dan model sedang belajar untuk mengenali fitur yang relevan dari data pelatihan. Namun, jika nilai loss mengalami fluktuasi besar atau tidak berubah selama beberapa iterasi, itu bisa menunjukkan adanya masalah dalam pelatihan, seperti gradien yang tidak stabil, learning rate terlalu besar, atau pengaturan hyperparameter yang tidak tepat. Oleh karena itu, analisis grafik training loss sangat penting untuk mengevaluasi kinerja pelatihan model dan mendeteksi masalah sejak awal.

Grafik learning rate digunakan untuk memastikan bahwa cara menentukan learning rate yang digunakan sesuai dengan rencana yang sudah ditentukan, yaitu menggunakan strategi polynomial decay dengan warmup pada awal pelatihan.

Melalui visualisasi learning rate, peneliti dapat memverifikasi bahwa learning rate meningkat secara perlahan selama fase warmup tanpa menyebabkan ketidakstabilan dalam pembaruan parameter, terutama saat model masih dalam tahap awal penyesuaian. Selain itu, penurunan learning rate secara perlahan dalam fase berikutnya membantu model mencapai konvergensi yang stabil tanpa terjadi pembelajaran yang terlalu cepat atau kekacauan. Dengan demikian, grafik learning rate merupakan alat penting untuk menilai bagaimana strategi optimisasi berjalan selama proses pelatihan.

3.6.3 Grafik Metrik Evaluasi pada Dataset Validasi

Selain grafik hasil pelatihan dan laju pembelajaran, grafik metrik evaluasi pada dataset validasi juga dianalisis secara rapi untuk mengawasi bagaimana performa model berkembang selama proses pelatihan. Metrik yang dipakai adalah mean Intersection over Union (mIoU), mean Accuracy (mAcc), dan overall Accuracy (aAcc), masing-masing memiliki fungsi sendiri dalam menilai kualitas segmentasi secara menyeluruh.

Grafik mIoU menjadi indikator utama untuk menilai kinerja segmentasi karena metrik ini mengukur seberapa cocok antara area yang diprediksi dengan data sebenarnya untuk setiap kelas secara seimbang. Dengan mengamati perubahan nilai mIoU dari satu iterasi ke iterasi berikutnya, peneliti dapat mengetahui seberapa baik model meningkatkan kualitas segmentasi untuk berbagai jenis objek, baik yang besar maupun kecil. Peningkatan perlahan pada nilai mIoU menunjukkan bahwa model tidak hanya mengingat pola dari data pelatihan, tetapi juga bisa menggeneralisasi terhadap data validasi yang belum pernah dilihat sebelumnya.

Grafik mAcc dianalisis untuk menilai kemampuan model dalam mengklasifikasikan setiap kelas objek secara merata. Berbeda dengan aAcc yang bersifat menyeluruh, mAcc menghitung rata-rata akurasi per kelas, sehingga sangat berguna untuk menemukan kelas yang sulit dipelajari oleh model. Perbedaan pola antara mAcc dan mIoU dapat menunjukkan apakah kesalahan

model terjadi karena ketidakmampuan mengenali keberadaan objek atau karena ketidakakuratan batas segmentasi.

Sementara itu, grafik aAcc menunjukkan akurasi keseluruhan pada tingkat piksel dan memberikan gambaran umum mengenai kinerja klasifikasi model. Namun, karena metrik ini sangat dipengaruhi oleh kelas dengan jumlah piksel yang besar seperti jalan atau langit, nilai aAcc perlu dianalisis dengan hati-hati dan tidak digunakan sebagai satu-satunya penilaian keberhasilan model.

3.6.4 Peran mIoU, mAcc, dan aAcc

Evaluasi kinerja model segmentasi semantik menggunakan tiga metrik, yaitu mean Intersection over Union (mIoU), mean Accuracy (mAcc), dan overall Accuracy (aAcc). Ketiga metrik ini digunakan bersamaan guna memperoleh gambaran yang menyeluruh mengenai kualitas segmentasi, baik dari sisi ketepatan prediksi tiap kelas maupun akurasi keseluruhan pada tingkat piksel.

Metrik mIoU digunakan sebagai penilaian utama untuk menilai kualitas segmentasi semantik karena mampu mengukur kesesuaian antara hasil prediksi model dan data sebenarnya (ground truth) pada setiap kelas secara seimbang. mIoU menghitung rasio antara area yang sama (intersection) dan area gabungan (union) antara prediksi dan anotasi ground truth, lalu diambil rata-ratanya untuk seluruh kelas. Pendekatan ini membuat mIoU tidak tergantung pada jumlah piksel suatu kelas, sehingga sangat cocok untuk dataset dengan klasifikasi tidak seimbang, seperti dataset jalan kota di mana kelas jalan dan bangunan dominan, sedangkan kelas seperti rider dan lampu lalu lintas memiliki jumlah piksel yang lebih sedikit.

Sementara itu, metrik mAcc digunakan untuk mengukur akurasi prediksi rata-rata pada tiap kelas secara terpisah. mAcc memberikan informasi tentang sejauh mana model mampu mengenali masing-masing kelas, tanpa terpengaruh oleh kelas yang dominan. Dengan demikian, mAcc penting untuk mengidentifikasi kelas yang sulit dipelajari model, terutama kelas minor atau objek kecil yang memiliki variasi visual tinggi dan data terbatas. Perbedaan signifikan antara nilai mIoU dan mAcc pada suatu kelas bisa menunjukkan bahwa model sudah bisa mendeteksi keberadaan objek, tetapi masih kurang tepat dalam menentukan batasnya.

Metrik aAcc merepresentasikan akurasi keseluruhan model pada tingkat piksel dengan menghitung rasio jumlah piksel yang dikenali benar terhadap total piksel pada seluruh gambar. Metrik ini memberikan gambaran umum mengenai performa model secara keseluruhan, namun cenderung dipengaruhi oleh kelas yang dominan. Karena itu, nilai aAcc yang tinggi tidak selalu mencerminkan akurasi baik pada semua kelas, terutama kelas minor.

Dengan menggunakan ketiga metrik ini secara bersamaan, evaluasi kinerja model menjadi lebih objektif dan menyeluruh. mIoU digunakan sebagai indikator utama kualitas segmentasi, mAcc digunakan untuk menganalisis performa model pada tiap kelas, sedangkan aAcc memberikan konteks mengenai akurasi global pada tingkat piksel.

