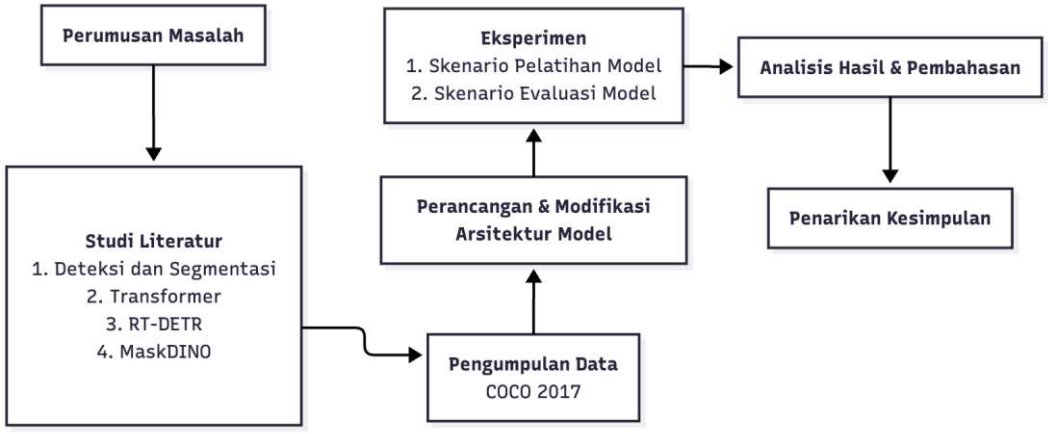
BAB III
METODE PENELITIAN

Bab ini akan menjelaskan secara rinci tentang langkah-langkah metodologis yang dilakukan dalam penelitian, mulai dari perancangan arsitektur, pengumpulan data, hingga proses eksperimen dan evaluasi model. Metodologi ini disusun secara sistematis untuk menjawab rumusan masalah yang telah dijabarkan pada bab sebelumnya.

3.1. Desain Penelitian

Desain penelitian merupakan kerangka kerja yang disusun secara sistematis untuk memandu jalannya penelitian agar dapat menjawab pertanyaan penelitian secara valid dan objektif. Desain ini mencakup semua tahapan, mulai dari perumusan masalah, pemilihan metode, hingga analisis data dan penarikan kesimpulan. Dengan adanya desain yang terstruktur, penelitian dapat dilaksanakan secara logis, koheren, dan terarah. Seluruh tahapan penelitian tersebut diilustrasikan dengan diagram alur pada Gambar 3.1



Gambar 3.1 Desain Penelitian

Pada bagian selanjutnya akan dijelaskan secara rinci dari desain penelitian yang akan dilakukan berdasarkan Gambar 3.1.

3.1.1. Perumusan Masalah

Tahap perumusan masalah adalah fondasi konseptual dari seluruh penelitian. Tahap ini didefinisikan sebagai proses untuk mengidentifikasi adanya kesenjangan (*research gap*) dan menetapkan tujuan yang jelas. Pada tahap ini, hal yang akan dilakukan adalah menganalisis kondisi terkini di mana model deteksi objek *real-time* seperti RT-DETR belum memiliki kemampuan *instance segmentation*, sementara model segmentasi yang akurat seperti Mask DINO tidak efisien untuk penggunaan *real-time*. Berdasarkan analisis tersebut, dirumuskan serangkaian pertanyaan penelitian yang spesifik untuk dijawab, yang memandu arah dan ruang lingkup penelitian secara keseluruhan.

3.1.2. Studi Literatur

Tahap studi literatur adalah proses membangun landasan teoritis dan kontekstual untuk penelitian. Pada tahap ini, akan dilakukan kajian pustaka secara mendalam terhadap berbagai sumber akademis. Langkah yang dilakukan meliputi penelaahan konsep-konsep fundamental seperti arsitektur Transformer dan berbagai jenis segmentasi gambar, serta pemahaman mendalam mengenai arsitektur kunci yang menjadi acuan, yaitu RT-DETR dan Mask DINO. Tujuan dari tahap ini adalah untuk memahami teknologi yang ada, mengidentifikasi metode yang relevan untuk diadopsi, dan memetakan posisi penelitian ini terhadap karya-karya sebelumnya (*state-of-the-art*).

3.1.3. Pengumpulan Data

Tahap ini didefinisikan sebagai proses penyiapan bahan utama yang akan digunakan dalam eksperimen. Langkah yang dilakukan pada tahap ini adalah memilih dan mengumpulkan dataset yang sesuai dengan tujuan penelitian. Sesuai dengan batasan yang telah ditetapkan, akan digunakan dataset COCO 2017, yang merupakan tolok ukur standar untuk tugas deteksi dan segmentasi. Setelah dikumpulkan, dataset tersebut akan melalui proses persiapan awal untuk memastikan data siap digunakan pada tahap pelatihan dan evaluasi model.

3.1.4. Perancangan dan Modifikasi Arsitektur Model

Tahap perancangan dan modifikasi arsitektur merupakan inti dari kontribusi teknis penelitian ini. Tahap ini didefinisikan sebagai proses merancang sebuah

solusi komputasional baru untuk menjawab masalah yang telah dirumuskan. Langkah utama yang akan dilakukan adalah membangun sebuah model baru bernama *Insta-RT-DETR*. Hal ini akan dicapai dengan memodifikasi arsitektur dasar *RT-DETR* dan mengintegrasikan tiga komponen kunci yang diadopsi dari *Mask DINO*, yaitu *mask branch*, *hybrid matching*, dan mekanisme *denoising*. Tujuan dari perancangan ini adalah untuk menciptakan sebuah arsitektur yang mampu melakukan *instance segmentation* secara efisien.

3.1.5. Eksperimen

Tahap eksperimen pelatihan adalah proses implementasi untuk mengoptimalkan parameter model yang telah dirancang. Pada tahap ini, akan dilakukan proses *training* di mana model *Insta-RT-DETR* dilatih menggunakan data latih dari COCO 2017. Langkah yang dilakukan meliputi konfigurasi skenario pelatihan, termasuk pengaturan *hyperparameter* seperti *learning rate* dan *batch size*, serta menjalankan semua proses komputasi intensif ini pada hardware GPU. Tujuan dari tahap ini adalah agar model dapat mempelajari pola-pola dari data sehingga mampu melakukan prediksi yang akurat.

Setelah pelatihan dilakukan, untuk mengukur kinerja model dibutuhkan tahap evaluasi. Tahap evaluasi merupakan proses pengukuran kinerja model yang telah dilatih secara objektif dan kuantitatif. Pada tahap ini, akan dilakukan serangkaian pengujian terhadap model menggunakan data validasi COCO 2017 yang belum pernah dilihat sebelumnya. Langkah yang akan dijalankan adalah mengukur performa model menggunakan metrik standar industri, yaitu *Average Precision* (AP) untuk menilai akurasi dan *Frames Per Second* (FPS) untuk menilai efisiensi komputasi. Tujuan tahap ini adalah untuk mendapatkan data kuantitatif yang valid mengenai seberapa baik performa model yang diusulkan.

3.1.6. Analisis Hasil dan Pembahasan

Tahap analisis hasil dan pembahasan adalah proses interpretasi terhadap data kuantitatif yang diperoleh dari tahap evaluasi. Pada tahap ini, akan dilakukan analisis mendalam terhadap hasil performa model. Langkah utama meliputi analisis komparatif, di mana kinerja *Insta-RT-DETR* akan dibandingkan dengan model-model lain yang relevan. Selain itu, akan dilakukan analisis terhadap *trade-off*

antara akurasi dan kecepatan. Tujuan dari tahap ini adalah untuk memahami keunggulan, kelemahan, dan posisi kontribusi mo del yang diusulkan dalam lanskap penelitian yang ada.

3.1.7. Penarikan Kesimpulan

Tahap terakhir dari desain penelitian adalah penarikan kesimpulan, yang merupakan proses kompilasi dari seluruh temuan penelitian. Pada tahap ini, akan dirumuskan jawaban yang konklusif untuk setiap pertanyaan penelitian yang diajukan di awal. Langkah yang dilakukan adalah merangkum hasil dan kontribusi utama dari penelitian, serta mengidentifikasi keterbatasan yang ada. Berdasarkan hal tersebut, akan diberikan saran-saran yang konstruktif untuk pengembangan atau penelitian selanjutnya di masa mendatang.

3.2. Alat dan Bahan Penelitian

Bagian ini menjelaskan alat dan bahan yang digunakan selama penelitian berlangsung.

3.2.1. Alat Penelitian

Penelitian ini menggunakan alat-alat meliputi perangkat keras, perangkat lunak, dan platform komputasi *serverless* dalam upaya untuk mencapai tujuan penelitian. Berikut adalah daftar perangkat dan data yang akan digunakan dalam jalannya penelitian:

1. Perangkat keras yang digunakan yaitu dua laptop dengan spesifikasi:
    a. Apple M1 chip (8-core CPU, 7-core GPU)
    b. 8 GB *Unified Memory*
    c. Penyimpanan SSD 256 GB
2. Perangkat lunak yang digunakan:
    a. Visual Studio Code
    b. Jupyter Notebook
    c. Kaggle Notebook
    d. Terminal
    e. Google Chrome
    f. WPS

g. GitHub
h. Mermaid Chart

3. Platform komputasi *serverless* yang digunakan adalah:
   a. Modal.com: Digunakan sebagai platform komputasi *serverless* untuk eksperimen yang membutuhkan GPU intensif, *training* dan evaluasi model menggunakan GPU NVIDIA L40S. Pengujian inferensi *Frame Per Second* (FPS) Menggunakan GPU NVIDIA A100 untuk mengukur performa (FPS) inferensi model dalam berbagai skenario sesuai dengan penelitian *Mask DINO* (Li et al., 2022).
   b. vast.ai: Juga digunakan sebagai platform komputasi *serverless* untuk eksperimen yang membutuhkan GPU, yaitu pengujian *Frame Per Second* (FPS) inferensi dengan menggunakan GPU NVIDIA V100 sesuai dengan benchmark yang dilakukan pada penelitian *FastInst* (He et al., 2023).

3.2.2. Bahan Penelitian

Bahan utama yang digunakan dalam penelitian ini adalah dataset publik COCO 2017 (Common Objects in Context). Dataset ini dipilih karena merupakan tolok ukur (benchmark) standar untuk tugas *object detection* dan *instance segmentation*, serta menyediakan citra dengan adegan yang kompleks dan representatif untuk pengujian model (Lin et al., 2014). Dataset COCO 2017 terdiri dari 118,287 gambar untuk pelatihan dan 5,000 gambar untuk validasi, yang mencakup 80 kategori objek.

Setiap objek dalam dataset ini telah dianotasi secara manual dengan *bounding box* dan *segmentation mask* tingkat instans (*per-instance*). Anotasi yang detail ini memungkinkan model untuk dilatih tidak hanya untuk mengenali dan melokalisasi objek, tetapi juga untuk memisahkan setiap *instance* objek hingga ke tingkat piksel, yang merupakan syarat fundamental untuk tugas *instance segmentation*. Format anotasi yang digunakan adalah JSON, yang memuat semua informasi terkait kategori, posisi, dan kontur segmentasi untuk setiap objek di dalam gambar. Penggunaan dataset yang kaya dan kompleks ini memastikan bahwa evaluasi performa model yang diusulkan dapat dilakukan secara robust dan komprehensif.

3.3. Prosedur Penelitian

Bagian ini menguraikan langkah-langkah teknis yang dilakukan secara sistematis selama pelaksanaan penelitian. Prosedur ini mencakup seluruh alur kerja eksperimental, mulai dari persiapan data, perancangan arsitektur, hingga skenario pelatihan dan evaluasi model untuk memperoleh hasil yang valid dan dapat dipertanggungjawabkan.

3.3.1. *Pre-Processing* dan *Load* Data

Prosedur pertama dalam penelitian ini adalah persiapan data. Tahap ini krusial untuk memastikan bahwa data yang digunakan bersih, bervariasi, dan siap untuk diproses oleh model. Bahan utama penelitian, yaitu dataset COCO 2017 (Lin et al., 2014), pertama-tama melalui tahap pra-pemrosesan. Prosedur ini melibatkan penerapan serangkaian teknik augmentasi data untuk meningkatkan variasi visual dan melatih model yang *robust* terhadap perubahan kondisi gambar. Teknik augmentasi yang diterapkan meliputi distorsi warna (*color distort*), perluasan (*expand*), pemotongan (*crop*), pembalikan horizontal (*flip*), dan yang terakhir, pengubahan ukuran (*resize*) semua gambar ke resolusi input standar 640x640 piksel. Data augmentasi tersebut sama persis mengikuti data augmentasi dari *RT-DETR* (Lv et al., 2023).

Selain augmentasi tersebut, diimplementasikan pula strategi pelatihan multi-skala (*multi-scale training*) yang dinamis selama *data loading*. Sesuai dengan metodologi *RT-DETR* (Lv et al., 2023), gambar input tidak diubah ukurannya ke resolusi tetap, melainkan secara acak diubah ukurannya pada setiap iterasi pelatihan. Ukuran resolusi dipilih dari serangkaian skala yang telah ditentukan, mulai dari 480x480 hingga 800x800 piksel. Prosedur ini berfungsi sebagai augmentasi data tambahan yang memaksa model untuk belajar mengenali objek dalam berbagai skala dan ukuran, yang kemudian dimuat ke dalam model menggunakan 4 *workers* secara paralel.

3.3.2. Skenario Pelatihan Model

Skenario pelatihan model *Insta-RT-DETR* dijalankan untuk mengoptimalkan parameter model yang telah dirancang. Pelatihan dilakukan selama 50 *epochs* dengan total waktu *training* mencapai 75 jam. Skenario pelatihan

diatur dengan konfigurasi *hyperparameter* yang spesifik, seperti yang dirinci pada Tabel 3.1, termasuk penggunaan *optimizer* AdamW dengan *base learning rate* 1e-4 dan *weight decay* 0.0001, serta *batch size* 16. Untuk meningkatkan efisiensi komputasi dan stabilitas selama proses belajar, dua teknik penting diaktifkan: *Automatic Mixed Precision* (AMP) untuk mempercepat komputasi, dan *Exponential Moving Average* (EMA) untuk menghaluskan pembaruan bobot model.

Sebagai bagian krusial dari skenario pelatihan, diterapkan strategi *multi-scale training* yang diadopsi langsung dari metodologi RT-DETR (Lv et al., 2023). Seperti yang telah dijelaskan pada tahap *pre-processing*, resolusi gambar input diubah secara dinamis pada setiap iterasi. Ukuran resolusi dipilih secara acak dari serangkaian skala yang telah ditentukan, mulai dari 480x480 hingga 800x800 piksel. Tujuan dari penerapan strategi ini dalam skenario pelatihan adalah untuk meningkatkan kemampuan generalisasi model, melatihnya agar menjadi lebih *robust* dalam mendeteksi dan melakukan segmentasi objek pada berbagai ukuran dan skala yang mungkin ditemui pada kondisi dunia nyata. Seluruh proses komputasi yang intensif ini dieksekusi pada platform komputasi *serverless* Modal.com dengan memanfaatkan sumber daya GPU NVIDIA L40S.

Tabel 3.1 Konfigurasi *Hyperparameter* yang digunakan pada *training*

Hyperparameter	**Value**
Input size	640
Data Augmentation	*color distort, expand, crop, flip, resize (640, 640)*
Backbone	*ResNet-50*
Epochs	50
Batch size	16
optimizer	AdamW
base learning rate	1e-4

Hyperparameter	Value
learning rate of backbone	1e-5
linear warm-up start factor	0.001
linear warm-up steps	2000
weight decay	0.0001
clip gradient norm	0.1
ema decay	0.9999
number of AIFI layers	1
number of RepBlocks	3
embedding dim	256
feedforward dim	1024
nheads	8
number of feature scales	4
number of decoder layers	6
number of queries	300
decoder npoints	4
class cost weight	2.0
α in class cost	0.25
γ in class cost	2.0
Bbox L1 cost weight	5.0
GIoU cost weight	2.0
dice cost weight	5.0

Hyperparameter	Value
mask bce cost weight	5.0
class loss weight	1.0
α in class loss	0.75
γ in class loss	2.0
Bbox L1 loss weight	5.0
GIoU loss weight	2.0
dice loss weight	5.0
mask bce loss weight	5.0
denoising number	100
label noise ratio	0.5
box noise scale	0.4

3.3.3. Skenario Evaluasi Model

Setelah proses pelatihan selesai, model dievaluasi secara kuantitatif untuk mengukur performanya pada dataset validasi COCO 2017. Skenario evaluasi dirancang untuk memastikan perbandingan yang adil dan komprehensif dengan model-model lain. Pengukuran performa dilakukan melalui dua aspek utama. Aspek akurasi diukur menggunakan metrik standar COCO, yaitu *Box* AP untuk *object detection* dan *Mask* AP untuk *instance segmentation*. Metrik turunan seperti *AP50*, *AP75*, serta *AP* untuk objek berukuran kecil (*APs*), medium (*APm*), dan besar (*APl*) juga dilaporkan untuk memberikan analisis yang lebih mendalam. Sementara itu, aspek efisiensi diukur dengan *Frames Per Second* (FPS) untuk mengetahui kecepatan *inference* model, disertai pencatatan jumlah parameter dan GFLOPS sebagai indikator kompleksitas komputasi.

Untuk memastikan perbandingan yang setara (*apple-to-apple comparison*) dengan hasil yang dilaporkan pada paper rujukan, protokol pengujian FPS

dilakukan secara ketat. Prosedur ini melibatkan pengukuran waktu komputasi rata-rata dengan *batch size* 1 untuk seluruh set data validasi COCO. Agar perbandingan relevan, pengujian ini dieksekusi pada dua jenis GPU yang berbeda sesuai dengan *hardware* yang digunakan oleh model pembanding. NVIDIA A100 digunakan untuk perbandingan dengan model-model umum seperti MaskDINO (Li et al., 2022), sedangkan NVIDIA V100 digunakan secara spesifik untuk perbandingan dengan model-model yang dioptimalkan untuk *real-time*, seperti FastInst (He et al., 2023).