BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil eksperimen dan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan utama yang menjawab rumusan masalah penelitian sebagai berikut:

1) Modifikasi arsitektur *Real-Time Detection Transformer* (RT-DETR) berhasil dilakukan untuk mendukung tugas *instance segmentation* dengan mengadopsi dan mengintegrasikan tiga komponen kunci dari arsitektur *MaskDINO*, yaitu *mask branch*, *hybrid matching*, dan mekanisme *unified denoising*. Integrasi ini menghasilkan sebuah arsitektur hibrida baru bernama *Insta-RT-DETR* yang mampu melakukan deteksi dan segmentasi dalam satu alur inferensi secara *end-to-end*.
2) Model *Insta-RT-DETR* menunjukkan performa akurasi yang sangat kompetitif pada dataset validasi COCO 2017. Dalam hal *instance segmentation*, model ini berhasil mencapai *Mask AP* sebesar 42.5%. Sementara itu, untuk tugas *object detection*, model ini mencatatkan *Box AP* sebesar 50.2%. Meskipun terdapat sedikit penurunan pada performa deteksi dibandingkan dengan arsitektur *RT-DETR*, hal ini merupakan *trade-off* yang wajar akibat penambahan tugas segmentasi secara simultan.
3) Model yang diusulkan berhasil mencapai *trade-off* yang optimal antara akurasi (AP) dan kecepatan (FPS), serta terbukti mampu beroperasi secara *real-time*. Dengan kecepatan inferensi mencapai 35.6 FPS pada GPU NVIDIA A100 dan 31.8 FPS pada GPU NVIDIA V100, *Insta-RT-DETR* secara signifikan lebih cepat dibandingkan model-model berakurasi tinggi seperti *MaskDINO* (14.2 FPS). Lebih penting lagi, saat dibandingkan dengan sesama model *real-time, Insta-RT-DETR* menunjukkan akurasi yang unggul, melampaui *FastInst-D3* (40.5% Mask AP) dan *YOLACT++* (34.1% Mask AP). Capaian ini memposisikan *Insta-RT-DETR* sebagai sebuah solusi yang sangat seimbang dan unggul terutama dalam kelas *real-time instance segmentation*.

5.2. Saran

Meskipun penelitian ini telah berhasil mencapai tujuannya, terdapat beberapa keterbatasan yang teridentifikasi yang dapat menjadi landasan untuk penelitian selanjutnya. Berikut adalah beberapa saran untuk pengembangan di masa mendatang:

1) Penelitian ini berhasil menunjukkan bahwa penambahan *mask branch, hybrid matching,* dan *unified denoising* pada RT-DETR menghasilkan kemampuan *instance segmentation* real-time dengan akurasi yang kompetitif. Namun, penelitian masih terbatas pada penggunaan backbone ResNet-50. Untuk penelitian berikutnya, dapat dipertimbangkan penggunaan backbone dengan kapasitas representasi lebih tinggi seperti ResNet-101 atau Swin Transformer, agar diperoleh pemahaman lebih dalam mengenai pengaruh kapasitas backbone terhadap kualitas segmentasi tanpa mengorbankan kecepatan. Selain itu, eksplorasi terhadap variasi desain encoder, misalnya pengembangan lebih lanjut dari AIFI atau CCFF, juga berpotensi meningkatkan efisiensi komputasi model.
2) Evaluasi model pada penelitian ini dilakukan menggunakan dataset COCO 2017, yang memang menjadi standar benchmark untuk tugas *object detection* dan *instance segmentation*. Namun, cakupan dataset ini terbatas pada skenario umum, sehingga masih perlu diuji apakah arsitektur Insta-RT-DETR mampu beradaptasi pada domain lain. Oleh karena itu, penelitian selanjutnya dapat menguji performa model pada dataset khusus seperti Cityscapes untuk skenario perkotaan, LVIS untuk jumlah kategori yang lebih luas, atau bahkan domain khusus seperti citra medis dan satelit. Langkah ini akan memperluas pemahaman mengenai kemampuan generalisasi arsitektur dan potensi penerapannya pada kasus nyata yang lebih spesifik.
3) Hasil penelitian ini menunjukkan bahwa Insta-RT-DETR mampu mencapai keseimbangan yang baik dengan capaian Mask AP sebesar 42.5% pada kecepatan 35.6 FPS. Meskipun demikian, kompromi antara akurasi dan efisiensi masih dapat ditingkatkan. Penelitian berikutnya dapat mengeksplorasi teknik optimasi yang lazim digunakan di bidang *computer*

vision, misalnya *model pruning* untuk mengurangi kompleksitas parameter, *quantization* untuk memperkecil presisi komputasi tanpa mengorbankan akurasi secara signifikan, serta pemanfaatan *lighter backbone* seperti *MobileNet* atau *EfficientNet* agar model lebih ramah terhadap deployment di perangkat dengan keterbatasan sumber daya. Pendekatan ini berpotensi mempercepat inferensi pada perangkat edge atau sistem real-time tanpa kehilangan presisi yang telah dicapai.