

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Tiga penelitian pada Tabel 2.1 dipilih karena berkaitan langsung dengan persepsi berkendara multitugas. YOLOP menjadi landasan jaringan yang menggabungkan object detection, drivable area segmentation, dan lane detection; YOLOPv2 mengembangkan konsep tersebut; sedangkan YOLOMH digunakan sebagai pembanding arsitektur multitugas yang menekankan efisiensi komputasi.

Tabel 2.1 Tiga penelitian terkait sistem persepsi visual ADV

Penelitian	Keterangan	Keterkaitan/Pembaruan
Wu et al. [2], YOLOP (2022)	Satu jaringan mengerjakan traffic object detection, drivable area segmentation, dan lane detection.	Landasan tiga keluaran persepsi yang digunakan pada penelitian.
Han et al. [3], YOLOPv2 (2022)	Mengembangkan persepsi berkendara multitugas berbasis YOLOP dengan perhatian pada akurasi dan efisiensi.	Model utama yang diimplementasikan untuk object detection, drivable area, dan lane line.
Fang et al. [4], YOLOMH (2024)	Mengusulkan arsitektur persepsi berkendara multitugas yang menekankan efisiensi komputasi.	Pembanding arsitektur multitugas; hasil kuantitatifnya tidak disetarakan dengan pengujian YOLOPv2 pada ADV.

Ketiga penelitian tersebut belum mengevaluasi rangkaian yang digunakan pada penelitian ini, yaitu kamera RGB Astra, inferensi YOLOPv2 pada mini PC CPU, pembentukan informasi obstacle dan Lane Safety, serta integrasi data melalui

ROS 2. Karena model, perangkat, dan protokol berbeda, hasil penelitian terdahulu digunakan sebagai konteks rancangan, bukan sebagai hasil pengujian ADV.

2.2 Autonomous Delivery Vehicle dan Sistem Persepsi Visual

ADV merupakan kendaraan yang dikembangkan untuk mendukung proses pengangkutan atau pengiriman barang dengan memanfaatkan sistem pengamatan lingkungan, pemrosesan informasi, perencanaan gerak, dan pengendalian kendaraan secara terintegrasi. Berbeda dengan kendaraan yang seluruh tindakannya ditentukan secara langsung oleh pengemudi, ADV memerlukan informasi dari sensor untuk membentuk representasi kondisi kendaraan dan lingkungan sebelum menentukan tindakan yang sesuai.

Secara umum, sistem kendaraan otonom dapat dipandang sebagai sejumlah subsistem yang memiliki fungsi berbeda tetapi saling berhubungan. Subsistem persepsi bertugas memperoleh dan mengolah informasi lingkungan, sedangkan subsistem lokalisasi digunakan untuk memperkirakan posisi serta orientasi kendaraan terhadap lingkungan atau peta. Informasi tersebut dapat digunakan oleh subsistem perencanaan untuk menentukan lintasan kendaraan. Selanjutnya, subsistem pengendalian menerjemahkan lintasan atau perintah gerak menjadi aksi pada aktuator kendaraan. Pembagian fungsi tersebut memungkinkan setiap subsistem dikembangkan dan dievaluasi berdasarkan keluaran yang menjadi tanggung jawabnya.

Persepsi memiliki peran penting karena kondisi lingkungan kendaraan tidak seluruhnya dapat diperoleh hanya dari informasi posisi. Kendaraan dapat menghadapi objek di depan jalur, perubahan area yang dapat dilalui, maupun perubahan batas jalur. Oleh karena itu, sistem persepsi diperlukan untuk mengubah data sensor menjadi informasi lingkungan yang dapat digunakan oleh bagian lain dari sistem kendaraan. Informasi persepsi dapat berbentuk keberadaan dan posisi objek, representasi area yang dapat dilalui, maupun batas jalur kendaraan.

Kajian *computer vision* pada kendaraan otonom mencakup pengenalan objek, pemahaman *scene*, estimasi gerak, dan pembentukan representasi lingkungan sebagai beberapa tugas yang saling berkaitan [1]. Penggunaan kamera memungkinkan informasi visual pada area depan kendaraan diperoleh dalam

bentuk citra dan kemudian diproses untuk menghasilkan representasi lingkungan yang lebih bermakna dibandingkan data piksel mentah.

Pada penelitian ini, ADV digunakan sebagai platform penerapan subsistem persepsi visual. Sistem persepsi diarahkan untuk menyediakan informasi mengenai objek dan kondisi jalur di depan kendaraan melalui pengolahan citra kamera. Keluaran tersebut diposisikan sebagai informasi lingkungan yang dapat diteruskan ke subsistem lain pada ADV, bukan sebagai bukti bahwa seluruh proses navigasi kendaraan telah berhasil.

Dengan demikian, pencapaian penelitian dibatasi pada kemampuan subsistem persepsi dalam menghasilkan informasi visual yang sesuai dengan fungsi yang dirancang serta pada karakteristik pemrosesan informasi tersebut. Keberhasilan kendaraan dalam menentukan lintasan, mencapai tujuan navigasi, maupun menghasilkan respons aktuator tidak digunakan sebagai parameter utama untuk menilai kinerja subsistem persepsi pada penelitian ini.

2.2.1 Sistem Persepsi Visual

Persepsi visual merupakan proses pengolahan citra untuk memperoleh representasi lingkungan yang memiliki makna bagi sistem. Citra yang diperoleh kamera pada dasarnya tersusun atas nilai piksel yang merepresentasikan informasi visual, tetapi nilai tersebut belum secara langsung menjelaskan keberadaan objek, area yang dapat dilalui, maupun batas jalur. Oleh karena itu, diperlukan proses pengolahan citra untuk mengubah data visual menjadi informasi yang lebih terstruktur dan dapat digunakan oleh sistem kendaraan.

Informasi yang dihasilkan sistem persepsi visual dapat memiliki bentuk yang berbeda sesuai dengan tugas pengolahannya. Pada *object detection*, sistem menghasilkan informasi mengenai keberadaan dan lokasi objek dalam bentuk *bounding box*, nilai *confidence*, dan identitas kelas. Sementara itu, pada *semantic segmentation*, setiap bagian citra diklasifikasikan pada tingkat piksel sehingga bentuk suatu area dapat direpresentasikan melalui *mask*. Perbedaan tersebut memungkinkan satu citra digunakan untuk memperoleh informasi objek sekaligus struktur lingkungan di sekitarnya.

Pada sistem persepsi kendaraan, kedua jenis informasi tersebut dapat saling melengkapi. Informasi objek memberikan keterangan mengenai keberadaan objek pada area pengamatan, sedangkan informasi segmentasi memberikan konteks mengenai kondisi ruang tempat objek tersebut berada. Dengan demikian, sistem tidak hanya mengetahui bahwa terdapat suatu objek di dalam citra, tetapi juga memperoleh informasi mengenai area yang dapat dilalui dan batas jalur kendaraan.

Dalam penelitian ini, keluaran persepsi visual terdiri atas *object detection*, *drivable area*, dan *lane line*. *Object detection* menghasilkan kandidat objek beserta informasi posisi dan *confidence*. Sementara itu, *drivable area* merepresentasikan bagian citra yang dikenali sebagai area yang dapat dilalui, sedangkan *lane line* memberikan informasi mengenai piksel yang berkaitan dengan marka atau batas jalur. Ketiga informasi tersebut menjadi keluaran dasar yang selanjutnya diproses sesuai kebutuhan sistem persepsi penelitian.

Keluaran model belum selalu dapat digunakan secara langsung sebagai informasi tingkat sistem. Hasil inferensi masih memerlukan tahapan *post-processing* untuk menyaring, menginterpretasikan, dan menyusun keluaran sesuai dengan kebutuhan aplikasi. Pada jalur deteksi, *bounding box* yang memenuhi ketentuan deteksi digunakan sebagai kandidat *obstacle*. Pada jalur segmentasi, *drivable area* dan *lane mask* digunakan untuk mengetahui hubungan antara batas jalur yang terdeteksi dan *safety boundary* yang ditentukan oleh sistem.

Dengan demikian, terdapat perbedaan antara keluaran model dan informasi hasil pengolahan sistem. *Bounding box*, *confidence*, *drivable area mask*, dan *lane mask* merupakan keluaran hasil pemrosesan model, sedangkan kandidat *obstacle* dan status *Lane Safety* merupakan informasi yang diperoleh setelah keluaran tersebut diolah lebih lanjut. Pemisahan ini penting karena nilai *confidence* suatu deteksi tidak secara langsung menunjukkan tingkat keselamatan kendaraan. Demikian pula, warna yang digunakan untuk menampilkan status *Lane Safety* bukan merupakan kelas yang diprediksi langsung oleh model, melainkan hasil interpretasi terhadap informasi jalur yang telah diproses.

Sistem persepsi visual juga mempunyai keterbatasan karena kualitas keluarannya dipengaruhi oleh kondisi citra dan lingkungan. Perubahan

pencahayaannya, oklusi objek, ukuran proyeksi objek pada citra, kondisi marka, serta perbedaan karakteristik lingkungan terhadap data yang digunakan dalam pengembangan model dapat memengaruhi hasil deteksi maupun segmentasi [1], [5]. Kondisi tersebut dapat menyebabkan objek tidak terdeteksi, *mask* tidak terbentuk secara utuh, atau informasi jalur menjadi tidak valid.

Oleh karena itu, sistem perlu membedakan antara kondisi ketika informasi visual tersedia dan dapat dievaluasi dengan kondisi ketika informasi tidak cukup untuk menghasilkan keputusan. Pada penelitian ini, prinsip tersebut digunakan terutama dalam pengolahan *Lane Safety*, sehingga data yang tidak valid tidak langsung diinterpretasikan sebagai kondisi aman. Validitas keluaran menjadi bagian penting sebelum informasi hasil persepsi diteruskan atau digunakan pada tahapan pengolahan berikutnya.

Pada penelitian ini, sistem persepsi visual kemudian direalisasikan menggunakan YOLOPv2 sebagai model utama untuk menghasilkan *object detection*, *drivable area*, dan *lane line*. Pembahasan lebih lanjut mengenai proses *object detection*, arsitektur YOLOPv2, serta pengolahan masing-masing keluarannya dijelaskan pada subbab berikutnya.

2.3 Pengolahan Citra dan Model YOLOPv2

Computer vision merupakan bidang yang mempelajari metode untuk memperoleh informasi dari citra melalui pengolahan karakteristik visual seperti bentuk, intensitas, warna, geometri, dan pola [5]. Dalam sistem persepsi visual, citra yang diperoleh kamera belum secara langsung dapat digunakan sebagai informasi mengenai kondisi lingkungan. Citra perlu melalui serangkaian proses agar informasi yang terkandung di dalamnya dapat diterjemahkan menjadi representasi objek maupun struktur area di sekitar kendaraan.

Citra digital tersusun atas piksel yang menyimpan informasi intensitas pada setiap kanal warna. Ukuran citra, urutan kanal warna, dan rentang nilai piksel yang digunakan kamera belum tentu sama dengan format masukan yang dibutuhkan oleh model. Oleh karena itu, sebelum proses inferensi dilakukan, diperlukan tahap prapemrosesan untuk menyesuaikan citra masukan dengan karakteristik model

yang digunakan. Tahap yang relevan pada penelitian ini meliputi perubahan ukuran citra, penyesuaian rasio aspek, konversi format warna, dan normalisasi nilai piksel.

Perubahan ukuran citra diperlukan karena model umumnya menggunakan ukuran masukan tertentu. Perubahan ukuran secara langsung dapat menyebabkan perbandingan panjang dan lebar objek mengalami distorsi apabila rasio aspek citra asli berbeda dengan ukuran masukan model. Salah satu metode yang dapat digunakan untuk mengurangi perubahan bentuk tersebut adalah *letterbox*. Metode ini mengubah ukuran citra dengan tetap mempertahankan rasio aspek, kemudian menambahkan *padding* pada bagian yang belum memenuhi dimensi masukan model.

Selain dimensi citra, format kanal warna juga harus disesuaikan dengan kebutuhan model. Kamera dan pustaka pengolahan citra dapat menggunakan susunan kanal yang berbeda dari susunan yang digunakan pada proses pelatihan model. Ketidaksesuaian urutan kanal dapat mengubah representasi warna yang diterima jaringan dan memengaruhi hasil inferensi. Oleh karena itu, konversi format warna dilakukan sebelum citra diteruskan ke tahap berikutnya.

Normalisasi digunakan untuk menyesuaikan rentang numerik nilai piksel dengan format masukan jaringan. Nilai piksel pada citra digital berada dalam rentang tertentu, sedangkan model memerlukan representasi numerik yang konsisten dengan kondisi saat model dikembangkan. Proses normalisasi membantu menjaga keseragaman skala nilai masukan sehingga operasi pada jaringan dapat dilakukan sesuai dengan konfigurasi model.

Setelah prapemrosesan selesai, citra digunakan sebagai masukan pada proses inferensi. Model kemudian mengekstraksi pola visual dari citra dan menghasilkan informasi sesuai dengan tugas yang dirancang. Dalam sistem persepsi penelitian ini, keluaran tersebut berkaitan dengan *object detection* serta *semantic segmentation*. *Object detection* digunakan untuk memperoleh informasi keberadaan dan posisi objek, sedangkan *semantic segmentation* digunakan untuk memperoleh representasi area pada tingkat piksel.

Tahap prapemrosesan juga memiliki hubungan langsung dengan interpretasi keluaran model. Pada proses *letterbox*, misalnya, citra mengalami perubahan skala

dan penambahan *padding*. Oleh karena itu, koordinat hasil deteksi yang dihasilkan pada ruang masukan model perlu dikoreksi kembali menggunakan informasi skala dan *padding* agar posisinya sesuai dengan citra asli. Tanpa proses tersebut, posisi *bounding box* pada hasil visualisasi dapat mengalami pergeseran atau perubahan ukuran.

Dengan demikian, konsistensi prapemrosesan merupakan bagian penting dalam sistem persepsi visual karena menentukan kesesuaian antara citra kamera, format masukan model, dan interpretasi keluaran inferensi. Pada penelitian ini, prinsip tersebut digunakan sebagai dasar pengolahan citra sebelum YOLOPv2 menghasilkan keluaran *object detection*, *drivable area*, dan *lane line*.

2.3.1 Object Detection

Object detection merupakan proses untuk menentukan keberadaan dan lokasi objek tertentu pada suatu citra. Berbeda dengan klasifikasi citra yang hanya menentukan kelas secara keseluruhan, *object detection* juga memberikan informasi posisi objek pada bidang citra. Keluaran deteksi umumnya berupa *bounding box*, nilai *confidence*, dan identitas kelas dari objek yang terdeteksi [6].

Bounding box digunakan untuk merepresentasikan lokasi objek dalam bentuk daerah persegi panjang pada citra. Informasi ini dapat dinyatakan menggunakan koordinat batas atau titik pusat beserta lebar dan tinggi kotak. Posisi *bounding box* selanjutnya dapat digunakan untuk mengetahui letak objek terhadap area pengamatan kamera serta menjadi dasar pengolahan informasi objek pada tahapan berikutnya.

Selain lokasi objek, hasil deteksi juga menghasilkan nilai *confidence*. Nilai tersebut menunjukkan tingkat keyakinan model terhadap kandidat deteksi yang dihasilkan. Kandidat dengan nilai *confidence* yang rendah dapat diabaikan menggunakan *confidence threshold*, sedangkan kandidat yang melewati nilai ambang diteruskan ke tahap pemrosesan berikutnya. Nilai *confidence* pada penelitian ini digunakan sebagai karakteristik keluaran deteksi dan tidak disamakan dengan tingkat keselamatan kendaraan.

Pada proses deteksi, satu objek dapat menghasilkan lebih dari satu kandidat *bounding box* yang saling bertumpang tindih. Oleh karena itu, diperlukan proses

penyaringan untuk mengurangi kandidat berulang. Salah satu metode yang umum digunakan adalah *Non-Maximum Suppression* (NMS). NMS mempertahankan kandidat yang memiliki skor lebih kuat dan mengurangi *bounding box* lain yang mempunyai tingkat tumpang tindih tinggi terhadap objek yang sama [6], [7].

Secara umum, tahapan hasil deteksi dapat dipandang sebagai proses pembentukan kandidat, penyaringan berdasarkan *confidence threshold*, dan pengurangan kandidat berulang melalui NMS. Hasil akhir tahap tersebut berupa kandidat objek yang telah memenuhi kriteria deteksi dan dapat digunakan sebagai masukan untuk proses interpretasi berikutnya.

Dalam penelitian ini, hasil *object detection* tidak langsung dianggap sebagai keputusan keselamatan kendaraan. *Bounding box* yang telah memenuhi ketentuan deteksi diposisikan sebagai kandidat *obstacle*. Informasi tersebut kemudian digunakan bersama proses pengolahan lain untuk membentuk keluaran sistem persepsi. Dengan demikian, terdapat perbedaan antara hasil deteksi objek pada tingkat model dengan informasi *obstacle* pada tingkat sistem.

Pengujian pada penelitian ini membatasi objek fisik yang dievaluasi pada kelas manusia. Untuk menjaga konsistensi identitas kandidat manusia selama pengujian, *pipeline obstacle* dilengkapi modul verifikasi kelas tambahan menggunakan SSDLite320 MobileNet V3 Large dengan bobot pralatih COCO [8], [9]. Modul tersebut digunakan untuk membantu memverifikasi bahwa kandidat yang dievaluasi merupakan manusia, khususnya pada tahapan pengujian deteksi dan estimasi jarak.

Penggunaan modul verifikasi tambahan tidak menggantikan peran YOLOPv2 sebagai model utama sistem persepsi. YOLOPv2 tetap digunakan untuk menghasilkan keluaran *object detection*, *drivable area*, dan *lane line*, sedangkan SSDLite digunakan sebagai pendukung verifikasi kelas manusia pada *pipeline obstacle*. Pembagian fungsi tersebut penting agar keluaran sistem tetap dapat ditelusuri sesuai dengan fungsi masing-masing model.

Evaluasi *object detection* pada penelitian ini juga tidak ditujukan sebagai *benchmark* performa YOLOPv2 secara terpisah. Oleh karena itu, penggunaan NMS tidak berarti penelitian secara otomatis melakukan evaluasi menggunakan

precision, *recall*, atau *mean Average Precision* (mAP). Evaluasi manusia lebih diarahkan pada keberhasilan kandidat manusia terdeteksi serta karakteristik *confidence* yang dihasilkan pada skenario pengujian. Parameter utama yang digunakan untuk mengevaluasi deteksi manusia adalah *detection rate* dan *confidence*, sedangkan evaluasi estimasi jarak menggunakan MAE dan RMSE.

Dengan demikian, hasil pengujian deteksi manusia diperlakukan sebagai keluaran dari *pipeline obstacle* terintegrasi dan tidak digunakan untuk menyatakan performa *object detection* YOLOPv2 secara independen.

2.3.2 You Only Look Once

YOLO merupakan pendekatan *object detection* berbasis *single-stage detector* yang melakukan proses lokalisasi dan klasifikasi objek secara langsung dalam satu alur *forward inference*. Berbeda dengan pendekatan *two-stage detector* yang terlebih dahulu membentuk kandidat wilayah objek (*region proposal*) dan kemudian melakukan klasifikasi terhadap kandidat tersebut, YOLO memproses citra secara menyeluruh dan menghasilkan prediksi posisi serta kelas objek secara langsung [7].

Secara umum, proses deteksi pada jaringan YOLO dapat dipahami melalui tiga bagian utama, yaitu *backbone*, *neck*, dan *detection head*. Backbone berfungsi mengekstraksi karakteristik visual dari citra masukan menjadi sejumlah *feature map*. Fitur yang diperoleh pada beberapa tingkat jaringan mengandung informasi dengan tingkat abstraksi dan skala yang berbeda. Selanjutnya, bagian neck menggabungkan fitur-fitur tersebut sehingga informasi posisi objek dan informasi semantik dapat digunakan bersama pada tahap deteksi. Detection head kemudian menggunakan feature map hasil penggabungan tersebut untuk menghasilkan prediksi objek.

Prediksi yang dihasilkan *detection head* mencakup informasi lokasi *bounding box*, ukuran *bounding box*, tingkat keyakinan terhadap keberadaan objek, dan informasi kelas objek. Bounding box menyatakan posisi objek pada bidang citra, sedangkan *confidence* menyatakan tingkat keyakinan model terhadap kandidat deteksi yang dihasilkan. Informasi kelas digunakan untuk menentukan kategori objek yang dikenali oleh model. Dengan demikian, satu proses inferensi

menghasilkan beberapa kandidat objek yang masing-masing memiliki informasi posisi, ukuran, kelas, dan tingkat keyakinan.

Pada model YOLO yang menggunakan mekanisme *anchor-based detection*, proses prediksi dilakukan pada beberapa skala feature map. Setiap lokasi pada feature map memiliki sejumlah *anchor* dengan ukuran atau rasio tertentu sebagai acuan awal pembentukan bounding box. Jaringan kemudian memprediksi perubahan posisi dan ukuran terhadap anchor tersebut serta probabilitas kelas objek. Penggunaan beberapa skala memungkinkan model mendeteksi objek dengan ukuran berbeda pada citra. Konsep deteksi multiskala dan anchor tersebut juga digunakan pada cabang *object detection* YOLOPv2 [3].

Kandidat bounding box yang dihasilkan jaringan belum langsung digunakan sebagai hasil akhir. Satu objek dapat menghasilkan beberapa kandidat bounding box yang saling bertumpang tindih. Oleh karena itu, tahap *post-processing* melakukan penyaringan berdasarkan nilai *confidence threshold*, kemudian menggunakan NMS untuk mempertahankan kandidat dengan tingkat keyakinan lebih tinggi dan menghilangkan kandidat lain yang merepresentasikan objek yang sama. Hasil tahap tersebut berupa deteksi objek yang telah dipilih beserta posisi bounding box, kelas, dan confidence.

Istilah *You Only Look Once* merujuk pada karakteristik bahwa prediksi objek diperoleh melalui satu proses *forward* jaringan terhadap citra masukan, bukan melalui rangkaian pencarian kandidat wilayah dan klasifikasi yang terpisah. Namun, satu kali *forward inference* tidak berarti hanya satu objek yang dapat dideteksi. Dalam satu frame, jaringan dapat menghasilkan beberapa bounding box untuk beberapa objek yang berada pada citra.

Pada aplikasi persepsi kendaraan, alur tersebut dijalankan berulang pada frame yang diperoleh dari kamera. Setiap frame yang diproses menghasilkan informasi objek baru yang dapat digunakan oleh sistem sebagai representasi kondisi lingkungan saat itu. Penelitian ini tidak melakukan perbandingan antarvarian YOLO, tetapi menggunakan prinsip tersebut sebagai dasar untuk memahami cabang *object detection* yang terdapat pada YOLOPv2.

Perlu dibedakan antara YOLOv2 dan YOLOPv2. YOLOv2 merupakan salah satu generasi model YOLO untuk tugas *object detection*, sedangkan YOLOPv2 merupakan pengembangan dari YOLOP untuk sistem persepsi berkendara multitugas. YOLOPv2 tidak hanya memiliki cabang deteksi objek, tetapi juga menghasilkan *drivable area segmentation* dan *lane detection* dalam satu jaringan [3].

2.3.3 YOLOP dan YOLOPv2

YOLOP dikembangkan sebagai jaringan persepsi berkendara multitugas yang menggabungkan tiga fungsi utama, yaitu *traffic object detection*, *drivable area segmentation*, dan *lane detection* dalam satu jaringan [2]. Berbeda dengan penggunaan tiga model terpisah, YOLOP menggunakan satu encoder bersama untuk mengekstraksi fitur citra kemudian mendistribusikan fitur tersebut ke beberapa decoder sesuai tugas masing-masing. Pendekatan tersebut memungkinkan beberapa informasi lingkungan diperoleh dari satu frame citra.

Secara umum, encoder pada jaringan persepsi multitugas berfungsi sebagai bagian yang mengekstraksi karakteristik visual dari citra. Fitur hasil ekstraksi tersebut kemudian diteruskan menuju cabang keluaran sesuai kebutuhan tugas. Cabang *object detection* menghasilkan informasi objek berupa bounding box, confidence, dan kelas. Cabang *drivable area segmentation* menghasilkan klasifikasi piksel yang menunjukkan area jalan yang dapat dilalui, sedangkan cabang *lane detection* menghasilkan informasi piksel yang berkaitan dengan garis atau batas lane.

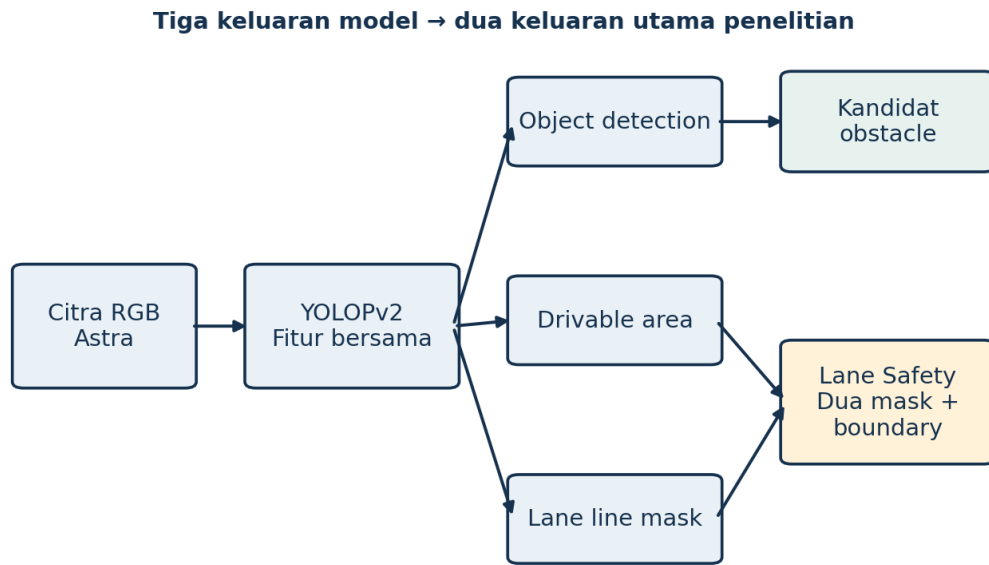
YOLOPv2 merupakan pengembangan lebih lanjut dari konsep YOLOP dengan perubahan arsitektur yang ditujukan untuk meningkatkan kemampuan dan efisiensi sistem persepsi multitugas [3]. YOLOPv2 menggunakan satu encoder bersama dan tiga cabang keluaran yang masing-masing menangani *traffic object detection*, *drivable area segmentation*, dan *lane detection*. Pada bagian encoder, informasi fitur dari beberapa tingkat jaringan diekstraksi dan digabungkan sebelum diteruskan menuju cabang tugas masing-masing.

Cabang *object detection* YOLOPv2 menggunakan mekanisme deteksi berbasis anchor pada beberapa skala. Feature map dari jaringan digunakan untuk

memprediksi posisi dan ukuran bounding box, probabilitas kelas, serta confidence kandidat objek. Penggunaan beberapa skala memungkinkan informasi objek dengan ukuran berbeda diproses pada tingkat feature map yang sesuai. Setelah inferensi, kandidat deteksi selanjutnya melalui proses penyaringan berdasarkan threshold dan NMS sebelum digunakan sebagai keluaran deteksi akhir [3].

Dua cabang lainnya bekerja pada tingkat piksel. Cabang *drivable area segmentation* memberikan informasi mengenai bagian citra yang diprediksi sebagai area yang dapat dilalui kendaraan, sedangkan cabang *lane detection* memberikan informasi mengenai posisi lane pada citra. Dengan demikian, YOLOPv2 tidak hanya menjawab pertanyaan “*objek apa dan berada di mana?*”, tetapi juga memberikan informasi “*bagian jalan mana yang dapat dilalui?*” dan “*di mana batas atau lane berada?*”.

YOLOPv2 dikembangkan dan dievaluasi oleh pengembangnya menggunakan dataset BDD100K. Dataset tersebut merupakan dataset skenario berkendara yang menyediakan variasi kondisi lingkungan dan anotasi untuk beberapa tugas persepsi. Pada penelitian asli YOLOPv2, BDD100K digunakan sebagai dataset pengembangan dan evaluasi model [3]. Penelitian Tugas Akhir ini tidak melakukan pelatihan ulang arsitektur YOLOPv2, melainkan mengimplementasikan model yang telah tersedia pada sistem ADV dan mengevaluasi keluarannya pada konfigurasi sistem penelitian.



Gambar 2.1 Hubungan tiga keluaran utama YOLOv2 dengan sistem persepsi penelitian

Sumber: Diadaptasi dari Wu dkk. [2]

Hubungan ketiga keluaran YOLOv2 dengan penelitian dapat dilihat pada Gambar 2.1. Keluaran *object detection* digunakan sebagai sumber informasi pada jalur obstacle, sedangkan *drivable area* dan *lane line* diolah lebih lanjut untuk membentuk informasi *Lane Safety*. Dengan pendekatan tersebut, satu frame RGB dari kamera dapat memberikan informasi objek sekaligus struktur lingkungan jalan yang selanjutnya digunakan oleh subsistem persepsi.

2.4 Platform Komputasi dan Backend Inferensi

Platform komputasi akhir penelitian adalah mini PC Intel NUC yang menjalankan YOLOv2 pada CPU melalui LibTorch. Model TorchScript dimuat menggunakan mekanisme PyTorch JIT [11]. Konfigurasi pengujian akhir mengacu pada backend CPU yang aktif saat pengambilan data, bukan pada backend lain yang tersedia pada repository.

NVIDIA Jetson Orin Nano digunakan sebagai platform rancangan awal sistem [12]. Platform tersebut tidak digunakan untuk menghasilkan data pengujian akhir, sehingga penelitian tidak membentuk perbandingan performa CPU dan *Graphics Processing Unit* (GPU).

Perubahan dari platform awal ke Intel NUC diperlakukan sebagai perubahan platform eksekusi. Tujuan persepsi dan fungsi utama YOLOPv2 tetap sama, sedangkan hasil performa yang dilaporkan seluruhnya berasal dari konfigurasi CPU pengujian akhir.

Performa sistem dievaluasi pada tingkat pipeline menggunakan pipeline throughput (FPS), inference time, processing time, dan frame age. Keempat parameter tersebut dibahas pada bagian parameter evaluasi.

Mini PC Intel NUC menjadi platform pengujian akhir, sedangkan Jetson Orin Nano hanya didokumentasikan sebagai platform pengembangan awal.

Mini PC yang digunakan pada pengujian akhir adalah Intel NUC 11 Pro NUC11TNHi5. Katalog Intel menempatkan NUC11TNHi5 pada keluarga NUC 11 Pro dengan prosesor Intel Core i5-1135G7 [13]. Perangkat ini digunakan untuk akuisisi kamera, inferensi YOLOPv2 melalui LibTorch CPU, ROS 2, *Graphical User Interface* (GUI), dan pencatatan data.



Gambar 2.2 Mini PC Intel NUC 11 Pro NUC11TNHi5

Sumber: <https://www.asus.com/us/displays-desktops/nucs/nuc-kits/nuc-11-pro-kit/> [22]

Konfigurasi perangkat keras dan perangkat lunak platform komputasi yang digunakan pada pengujian akhir dirangkum pada Tabel 2.2.

Tabel 2.2 Konfigurasi platform komputasi pengujian akhir

Parameter	Konfigurasi penelitian
Platform	Intel NUC 11 Pro NUC11TNHi5-WH
Kelas prosesor	Intel Core i5-1135G7 [13]

Lanjutan Tabel 2.2 Konfigurasi platform komputasi pengujian akhir

Parameter	Konfigurasi penelitian
Sistem operasi	Ubuntu 22.04
Middleware	ROS 2 Humble
Backend inferensi	LibTorch pada CPU
Model utama	YOLOPv2 (.pt)
Ukuran input model	640 × 384 piksel
Fungsi utama	Akuisisi kamera, inferensi, ROS 2, GUI, dan pencatatan data

2.5 Akuisisi Citra RGB

Kamera RGB digunakan untuk memperoleh informasi visual lingkungan dalam bentuk citra digital. Setiap piksel pada citra merepresentasikan informasi intensitas pada kanal merah (*red*), hijau (*green*), dan biru (*blue*). Kombinasi ketiga kanal tersebut membentuk informasi warna yang selanjutnya dapat digunakan sebagai masukan bagi proses pengolahan citra dan sistem persepsi visual.

Dalam sistem persepsi, kualitas informasi yang diperoleh dari kamera dipengaruhi oleh beberapa parameter akuisisi, antara lain resolusi dan *frame rate*. Resolusi menunjukkan jumlah piksel yang membentuk suatu citra. Resolusi yang lebih tinggi menyediakan informasi spasial yang lebih rinci, tetapi juga meningkatkan jumlah data yang harus diproses pada setiap *frame*. Sementara itu, *frame rate* menyatakan jumlah *frame* yang disediakan kamera dalam satu detik.

Nilai *frame rate* kamera tidak sama dengan *throughput* pemrosesan model. Kamera dapat menyediakan sejumlah *frame* dalam satu detik, tetapi tidak seluruh *frame* harus selesai diproses oleh model pada laju yang sama. Waktu prapemrosesan, inferensi, *post-processing*, serta beban komputasi lain dapat menyebabkan jumlah *frame* yang selesai diproses lebih rendah dibandingkan laju akuisisi kamera. Oleh karena itu, pada penelitian ini request frame rate kamera dan pipeline throughput diperlakukan sebagai dua parameter yang berbeda.

Akuisisi citra pada penelitian menggunakan kamera Orbbec Astra Pro sebagai sensor visual pada bagian depan kendaraan. Keluarga kamera Orbbec Astra

menyediakan informasi RGB dan depth melalui antarmuka USB [14]. Kanal RGB menghasilkan citra yang memuat informasi warna dan tekstur lingkungan, sedangkan kanal depth pada dasarnya menyediakan informasi kedalaman objek terhadap sensor. Meskipun kedua informasi tersebut tersedia pada perangkat kamera, penelitian ini menggunakan kanal RGB sebagai masukan utama sistem persepsi.

Pemilihan kanal RGB disesuaikan dengan implementasi YOLOPv2 yang digunakan pada penelitian, yaitu model menerima citra RGB yang telah melalui tahap prapemrosesan untuk menghasilkan keluaran *object detection*, *drivable area*, dan *lane line*. Dengan demikian, informasi kedalaman bukan merupakan masukan yang diperlukan oleh proses *forward inference* YOLOPv2 pada konfigurasi penelitian ini.

Kanal depth tidak digunakan sebagai sumber utama estimasi jarak karena estimasi jarak pada penelitian dirancang berdasarkan geometri *bounding box* hasil deteksi manusia. Pemisahan tersebut juga dilakukan agar metode estimasi jarak yang dievaluasi tetap konsisten dengan tujuan penelitian, yaitu mengevaluasi informasi visual yang dihasilkan dari pipeline RGB dan keluaran deteksi objek.

Selain pertimbangan ruang lingkup penelitian, penggunaan RGB saja juga mempertimbangkan sumber daya komputasi platform. Penggunaan stream depth secara bersamaan tidak hanya memerlukan akuisisi data tambahan, tetapi juga dapat membutuhkan sinkronisasi frame RGB dan depth, penyesuaian atau *alignment* antara sistem koordinat kedua citra, pemeriksaan nilai depth yang valid, serta pengolahan data kedalaman sebelum dapat digunakan oleh sistem persepsi. Tahapan tersebut merupakan proses tambahan pada pipeline di luar *forward inference* YOLOPv2.

Oleh karena itu, penggunaan kanal RGB pada penelitian ini tidak menunjukkan bahwa kemampuan depth pada kamera Astra tidak dapat dimanfaatkan, melainkan merupakan pembatasan desain sistem sesuai ruang lingkup penelitian dan kemampuan platform komputasi yang digunakan. Kanal depth dapat dikembangkan pada penelitian selanjutnya sebagai sumber informasi

kedalaman tambahan atau sebagai pembanding terhadap metode estimasi jarak berbasis geometri *bounding box*.

Frame RGB dari kamera diakuisisi melalui Video4Linux2 (V4L2) dan OpenCV [15], [16]. V4L2 berperan sebagai antarmuka akuisisi perangkat video pada sistem Linux, sedangkan OpenCV digunakan untuk menerima dan mengolah citra pada aplikasi. Citra yang telah diperoleh kemudian diteruskan ke tahapan prapemrosesan sebelum menjadi masukan bagi model YOLOPv2. Dengan demikian, proses akuisisi citra menjadi tahap awal yang menghubungkan sensor kamera dengan pipeline persepsi visual.

Selain menyediakan informasi warna, citra kamera juga memiliki karakteristik proyeksi perspektif. Pada proyeksi perspektif, objek dengan ukuran fisik yang sama cenderung menempati ukuran piksel yang semakin kecil ketika jaraknya terhadap kamera bertambah [5]. Sebaliknya, objek yang semakin dekat dengan kamera akan menempati bagian citra yang lebih besar. Hubungan tersebut menyebabkan ukuran objek pada citra dapat digunakan sebagai salah satu sumber informasi geometris untuk memperkirakan jarak.

Pada penelitian ini, prinsip perspektif tersebut digunakan pada pengujian manusia. Informasi geometris yang diperoleh dari *bounding box* hasil deteksi digunakan sebagai dasar pembentukan variabel estimasi jarak. Dengan demikian, citra RGB tidak hanya digunakan untuk menentukan keberadaan objek, tetapi juga menyediakan informasi geometris yang selanjutnya dapat diolah untuk memperkirakan jarak objek terhadap kendaraan.

Konfigurasi akuisisi pada pengujian menggunakan resolusi 1280×720 piksel dengan permintaan kamera sebesar 15 FPS. Resolusi tersebut merupakan resolusi citra yang diperoleh dari kamera, sedangkan citra yang diteruskan ke model dapat mengalami penyesuaian ukuran sesuai dengan kebutuhan input YOLOPv2. Perbedaan antara resolusi akuisisi kamera dan ukuran masukan model merupakan bagian dari tahap prapemrosesan sebagaimana dijelaskan pada Subbab 2.3.

Dengan demikian, akuisisi citra RGB pada penelitian berfungsi sebagai tahap awal untuk menyediakan informasi visual lingkungan bagi sistem persepsi. Kamera Astra menghasilkan citra area depan kendaraan, kemudian *frame* diakuisisi melalui

V4L2 dan OpenCV sebelum diproses lebih lanjut oleh pipeline YOLOPv2. Keluaran proses tersebut selanjutnya digunakan untuk membentuk informasi *object detection*, *drivable area*, dan *lane line* sesuai dengan kebutuhan sistem persepsi.



Gambar 2.3 Kamera Orbbec Astra sebagai sensor visual penelitian

Sumber: <https://shop.orbbec3d.com/Astra> [23]

Parameter akuisisi kamera RGB yang digunakan pada penelitian ini dirangkum pada Tabel 2.3.

Tabel 2.3 Konfigurasi akuisisi kamera RGB pada penelitian

Parameter	Konfigurasi penelitian
Perangkat	Orbbec Astra Pro
Kanal yang digunakan	RGB
Resolusi yang diminta aplikasi	1280 × 720 piksel
Frame rate yang diminta	15 FPS
Antarmuka	USB 2.0
Akuisisi frame	V4L2 / OpenCV
Kanal depth	Tidak digunakan sebagai masukan utama pengujian

2.6 Semantic Segmentation

Semantic segmentation merupakan proses pengelompokan citra pada tingkat piksel sehingga setiap piksel dapat diberikan label sesuai dengan kelas atau kategori yang dikenali oleh model [17], [18]. Berbeda dengan *object detection* yang

merepresentasikan objek menggunakan *bounding box*, *semantic segmentation* dapat mengikuti bentuk area secara lebih rinci karena informasi diberikan pada setiap bagian citra.

Hasil *semantic segmentation* umumnya direpresentasikan dalam bentuk *mask*. *Mask* menunjukkan lokasi piksel yang termasuk ke dalam kelas tertentu, sehingga bentuk objek atau area dapat dipertahankan lebih baik dibandingkan representasi berbentuk kotak. Karakteristik ini sesuai untuk tugas yang membutuhkan informasi mengenai struktur lingkungan, seperti permukaan jalan dan batas jalur kendaraan.

Pada sistem persepsi kendaraan, informasi segmentasi dapat digunakan untuk membedakan bagian citra berdasarkan fungsi lingkungannya. Area jalan yang dikenali dapat direpresentasikan sebagai *drivable area*, sedangkan piksel yang berkaitan dengan marka atau batas jalur dapat direpresentasikan sebagai *lane line*. Kedua informasi tersebut memberikan konteks spasial yang berbeda dibandingkan keluaran *object detection*.

Pada YOLOPv2, *drivable area* dan *lane line* dihasilkan melalui cabang segmentasi yang berbeda dan digunakan sesuai dengan fungsi masing-masing. *Drivable area* menunjukkan bagian citra yang dikenali sebagai area yang dapat dilalui, sedangkan *lane line* menunjukkan piksel yang berkaitan dengan marka atau batas jalur. Dengan demikian, satu keluaran memberikan informasi mengenai area jalan, sedangkan keluaran lainnya memberikan informasi mengenai struktur batas jalur.

Keluaran segmentasi tersebut belum langsung digunakan sebagai keputusan pada tingkat sistem. *Mask* hasil model masih memerlukan proses pengolahan lanjutan agar dapat diterjemahkan menjadi informasi yang sesuai dengan kebutuhan kendaraan. Dalam penelitian ini, *drivable area* dan *lane line* digunakan sebagai dasar untuk membentuk informasi *Lane Safety*. Proses tersebut dilakukan dengan mengevaluasi hubungan antara hasil segmentasi jalur dan batas keselamatan yang ditentukan pada sistem.

Hubungan antara kedua keluaran segmentasi juga penting karena informasi jalur tidak selalu dapat dinilai hanya dari satu *mask*. *Drivable area* memberikan

konteks mengenai keberadaan area yang dapat dilalui, sedangkan *lane line* memberikan informasi mengenai posisi batas jalur. Penggabungan kedua informasi tersebut memungkinkan sistem memperoleh gambaran yang lebih lengkap mengenai kondisi area di depan kendaraan.

Kualitas hasil segmentasi dipengaruhi oleh kondisi visual pada citra. Perubahan pencahayaan, bayangan, marka yang kurang jelas, oklusi, serta perbedaan karakteristik lingkungan terhadap data pelatihan dapat menyebabkan hasil *mask* tidak terbentuk secara sempurna. Kesalahan segmentasi dapat berupa bagian jalan yang tidak terdeteksi, piksel latar belakang yang salah diklasifikasikan sebagai jalan, atau bagian *lane line* yang terputus.

Kesalahan tersebut dapat memengaruhi informasi hasil pengolahan berikutnya. Sebagai contoh, perubahan atau terputusnya *lane mask* dapat menyebabkan jarak antara batas jalur dan *safety boundary* ikut berubah meskipun posisi kendaraan tidak mengalami perubahan. Oleh karena itu, validitas informasi segmentasi perlu diperhatikan sebelum hasilnya diinterpretasikan sebagai kondisi jalur yang aman atau tidak aman.

Pada penelitian ini, evaluasi segmentasi tidak dilakukan dengan membandingkan setiap piksel keluaran terhadap anotasi referensi untuk memperoleh metrik seperti *Intersection over Union* (IoU) atau *pixel accuracy*. Penilaian lebih diarahkan pada dampak fungsional keluaran segmentasi terhadap informasi *Lane Safety*. Dengan demikian, keberhasilan sistem dinilai berdasarkan kesesuaian status *Lane Safety* yang dihasilkan pada kondisi pengujian yang telah ditetapkan, bukan berdasarkan kesesuaian seluruh piksel hasil segmentasi terhadap anotasi *ground truth*.

Dengan demikian, *semantic segmentation* pada penelitian ini berfungsi sebagai dasar untuk memperoleh informasi *drivable area* dan *lane line* dari citra kamera. Kedua keluaran tersebut kemudian diproses lebih lanjut menjadi informasi *Lane Safety*, sedangkan mekanisme evaluasi hubungan jalur, *safety boundary*, dan penentuan status keselamatan dijelaskan pada Subbab 2.8.

2.7 Estimasi Jarak Berbasis Geometri Bounding Box

Hubungan perspektif menyebabkan ukuran proyeksi objek pada citra berubah terhadap jarak [5]. Pada penelitian ini, estimasi jarak manusia dilakukan menggunakan karakteristik geometris *bounding box* hasil deteksi. Variabel yang digunakan sebagai dasar kalibrasi adalah rasio antara lebar citra dan lebar *bounding box* manusia. Jika W_{image} merupakan lebar citra dan W_{bbox} merupakan lebar *bounding box*, maka rasio geometri dirumuskan pada Persamaan (2.1).

$$r = \frac{W_{image}}{W_{bbox}} \quad (2.1)$$

Keterangan:

r : rasio geometri citra (tanpa satuan),

W_{image} : lebar citra (piksel), dan

W_{bbox} : lebar bounding box manusia (piksel).

Berdasarkan Persamaan (2.1), semakin kecil lebar *bounding box* manusia terhadap lebar citra, semakin besar nilai rasio r . Hubungan ini digunakan sebagai variabel masukan pada proses kalibrasi. Persamaan (2.1) merupakan formulasi rasio yang digunakan dalam penelitian, sedangkan dasar hubungan perspektif antara ukuran proyeksi objek dan jarak mengacu pada [5].

Estimasi jarak selanjutnya dinyatakan menggunakan model kalibrasi linear seperti pada Persamaan (2.2).

$$\hat{d} = ar + b \quad (2.2)$$

Keterangan:

$\hat{d}$: estimasi jarak manusia (m),

a : koefisien kemiringan model kalibrasi,

r : rasio geometri,

b : konstanta atau intersep model kalibrasi.

Koefisien a dan b diperoleh dari dataset kalibrasi terpisah yang dibuat sebelum pengujian utama. Dataset kalibrasi menghubungkan nilai rasio geometri *bounding box* dengan jarak referensi yang telah diketahui. Model pada Persamaan (2.2) kemudian diterapkan pada dataset evaluasi untuk memperoleh estimasi jarak manusia.

Nilai estimasi jarak dibandingkan dengan jarak referensi untuk memperoleh *error*, MAE, dan RMSE. Definisi dan perhitungan metrik tersebut dibahas pada Subbab 2.10.1.

Model pada Persamaan (2.2) merupakan model kalibrasi empiris penelitian. Oleh karena itu, model hanya berlaku pada konfigurasi kamera, resolusi citra, posisi pemasangan, dan jenis target yang digunakan selama kalibrasi. Perubahan konfigurasi tersebut memerlukan kalibrasi ulang agar hubungan antara geometri *bounding box* dan jarak tetap sesuai dengan kondisi pengujian.

2.8 Lane Detection dan Lane Safety

Lane detection menyediakan informasi lokasi *lane line* pada citra, sedangkan *drivable area* memberikan konteks area jalan yang dikenali model. Pada penelitian ini, kedua keluaran YOLOPv2 tersebut digunakan sebagai dasar pembentukan informasi *Lane Safety* untuk menilai hubungan *outer lane* terhadap *safety boundary* [2], [3]. *Safety boundary* merupakan batas acuan yang ditetapkan pada sistem dan bukan keluaran langsung jaringan YOLOPv2.

Setelah *lane line mask* dikembalikan ke koordinat citra asli, sistem memilih bagian *outer lane* yang relevan dengan batas koridor kendaraan. Hubungan geometris antara *outer lane* dan *safety boundary* dihitung pada setiap baris citra y . Koordinat horizontal x bertambah dari kiri ke kanan citra. Subskrip s menunjukkan *safety boundary*, subskrip l menunjukkan *lane mask*, sedangkan L dan R menunjukkan sisi kiri dan kanan. Besarnya *gap* sisi kiri dan sisi kanan dinyatakan pada Persamaan (2.3) dan Persamaan (2.4).

$$g_L(y) = x_{s,L}(y) - x_{l,L}(y) \quad (2.3)$$

Keterangan:

$g_L(y)$	: <i>gap</i> sisi kiri pada baris citra y (piksel),
$x_{s,L}(y)$	: posisi horizontal <i>safety boundary</i> kiri (piksel)
$x_{l,L}(y)$	: posisi horizontal <i>outer lane</i> kiri (piksel).

$$g_R(y) = x_{l,R}(y) - x_{s,R}(y) \quad (2.4)$$

Keterangan:

- $gR(y)$: *gap* sisi kanan pada baris citra y (piksel),
 $xs,R(y)$: posisi horizontal *safety boundary* kanan (piksel)
 $xl,R(y)$: posisi horizontal *outer lane* kanan (piksel).

Persamaan (2.3) dan Persamaan (2.4) mengikuti operasi pengurangan yang digunakan pada fungsi *evaluateLaneCorridor()* dalam implementasi penelitian. Nilai *gap* positif menunjukkan *outer lane* masih berada di luar *safety boundary*. Nilai nol menunjukkan kondisi ketika *lane* menyentuh batas, sedangkan nilai negatif menunjukkan bahwa *lane* telah masuk ke bagian dalam koridor keselamatan.

Perhitungan pada satu baris citra dapat dipengaruhi oleh ketidakstabilan lokal pada *lane mask*. Oleh karena itu, sistem tidak menggunakan satu nilai *gap* secara langsung. Beberapa nilai *gap* valid dikumpulkan, diurutkan dari nilai terkecil, kemudian diringkas menggunakan metode *robust closest gap* seperti pada Persamaan (2.5) dan Persamaan (2.6).

$$\bar{g}_s = \frac{1}{q_s} \sum_{j=1}^{q_s} g_{s,(j)} \quad (2.5)$$

Keterangan:

- s : sisi L atau R
 $g_{s,(j)}$: *gap* urutan ke- j setelah nilai *gap* diurutkan menaik
 q_s : jumlah sampel pada seperempat terbawah yang digunakan dalam perhitungan
 $\bar{g}_s$: nilai *robust closest gap* sisi s (piksel).

$$q_s = \max\left(1, \left\lfloor \frac{m_s}{4} \right\rfloor\right) \quad (2.6)$$

Keterangan:

- m_s : jumlah sampel *gap* yang valid pada sisi s
 q_s : jumlah sampel pada seperempat terbawah yang digunakan dalam perhitungan.

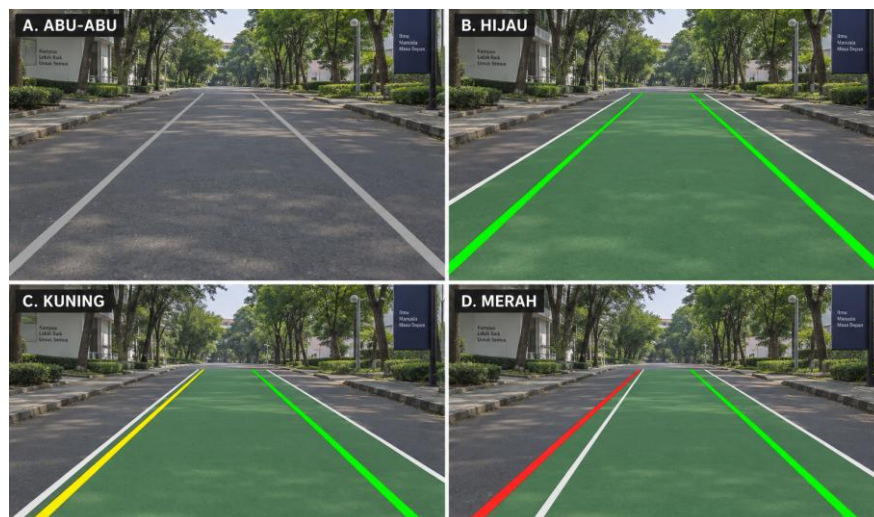
Persamaan (2.5) dan Persamaan (2.6) merepresentasikan metode yang digunakan pada fungsi *robustClosestGap()* dalam implementasi penelitian. Sistem mengambil rata-rata seperempat nilai *gap* terkecil sehingga hasil lebih

merepresentasikan bagian *outer lane* yang paling dekat terhadap *safety boundary* tanpa bergantung pada satu sampel tunggal. Implementasi hanya menilai suatu sisi valid apabila tersedia sedikitnya enam baris sampel yang memenuhi kriteria.

Sebagai contoh, apabila diperoleh delapan nilai *gap* terurut sebesar 8, 10, 12, 15, 18, 20, 22, dan 24 piksel, maka $q = 2$. Nilai representatif diperoleh dari rata-rata dua *gap* terkecil, yaitu $(8 + 10)/2 = 9$ piksel. Nilai ini menunjukkan jarak geometris pada citra dan tidak ditafsirkan sebagai lebar jalan dalam satuan meter.

Persamaan (2.3) sampai dengan Persamaan (2.6) menjadi dasar perhitungan *gap* dan penafsiran status Lane Safety. Semakin kecil nilai *gap*, semakin dekat *outer lane* terhadap *safety boundary*. Status GREEN, YELLOW, RED, dan UNKNOWN dibentuk berdasarkan nilai *gap*, validitas data, dan mekanisme histeresis.

Histeresis menggunakan ambang masuk dan keluar yang berbeda untuk mengurangi perubahan status yang berulang ketika nilai *gap* berada di sekitar batas keputusan. Pada konfigurasi penelitian, *touch margin* ditetapkan sebesar 2 piksel, *warning gap* sebesar 36 piksel, dan *release gap* sebesar 48 piksel. Nilai tersebut merupakan parameter eksperimen pada konfigurasi kamera dan resolusi citra yang digunakan, sehingga tidak diperlakukan sebagai batas keselamatan universal.



Gambar 2.4 Ilustrasi empat tingkat status Lane Safety pada koridor kendaraan

Sumber: Hasil Olahan Penulis (2026)

Gambar 2.4 merangkum interpretasi visual empat tingkat status *Lane Safety*. Warna abu-abu menunjukkan informasi jalur tidak valid atau kondisi yang tidak dapat dievaluasi, hijau menunjukkan *outer lane* berada di luar *safety boundary*

dengan margin yang memadai, kuning menunjukkan *lane* mendekati batas koridor, sedangkan merah menunjukkan kondisi sentuh atau penetrasi terhadap batas keselamatan. Penentuan status tetap mengikuti nilai *gap*, validitas *mask*, dan riwayat histeresis sebagaimana dirumuskan pada Persamaan (2.3) sampai dengan Persamaan (2.6).

Keluaran *Lane Safety* pada penelitian ini dievaluasi sebagai informasi persepsi, bukan sebagai bukti keberhasilan kendali *steering* atau respons aktuator.

2.9 Robot Operating System 2

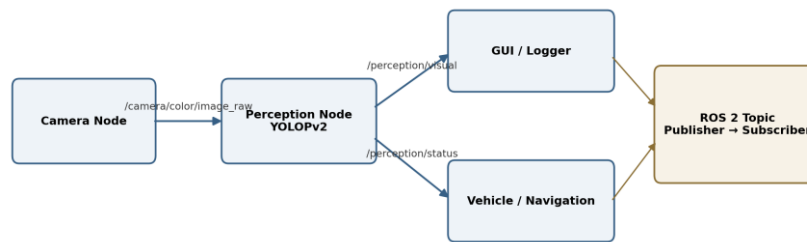
ROS 2 menyediakan mekanisme komunikasi antarkomponen melalui node, topic, publisher, dan subscriber. Arsitektur ROS 2 menggunakan middleware yang dapat berbasis *Data Distribution Service* (DDS) untuk komunikasi publish-subscribe [19].

ROS 2 dirancang sebagai kerangka robotika modular yang mendukung komunikasi, konfigurasi, dan penerapan pada beragam platform [20]. Pada penelitian ini ROS 2 digunakan sebagai middleware integrasi antara kamera, node persepsi, GUI, dan subsistem lain pada ADV.

Quality of Service (QoS) mengatur kebijakan komunikasi seperti reliability, history, dan depth [21]. Sensor data umumnya memerlukan data terbaru, sedangkan telemetry/state dapat menggunakan kebutuhan reliability yang berbeda.

Timestamp digunakan untuk menelusuri waktu data, sedangkan kontrak topic menentukan jenis informasi yang dipertukarkan. Keberadaan komunikasi ROS 2 tidak dengan sendirinya membuktikan bahwa keluaran persepsi benar; validasi tetap dilakukan melalui pengujian yang dirancang pada penelitian.

Skema komunikasi publish-subscribe ROS 2 pada aliran data persepsi ditunjukkan pada Gambar 2.5.



Gambar 2.5 Skema publish-subscribe ROS 2 pada aliran data persepsi

Sumber: Hasil Olahan Penulis (2026)

Pada Gambar 2.5, node persepsi menerima citra dan menerbitkan hasil persepsi, sedangkan GUI atau subsistem lain membaca topic yang diperlukan. Pemisahan tersebut memungkinkan akuisisi, inferensi, visualisasi, dan konsumsi data berjalan sebagai komponen yang terhubung melalui ROS 2.

2.10 Parameter Evaluasi yang Digunakan

Parameter evaluasi dipilih sesuai empat kelompok pengujian utama. Deteksi manusia dievaluasi menggunakan detection rate dan confidence; estimasi jarak menggunakan error, MAE, dan RMSE; Lane Safety menggunakan kesesuaian pasangan status; sedangkan performa pemrosesan menggunakan pipeline throughput (FPS), inference time, processing time, dan frame age.

2.10.1 Detection rate, confidence, MAE, dan RMSE jarak

Detection rate adalah persentase sampel yang menghasilkan deteksi manusia terhadap seluruh sampel pada jarak yang diuji. Confidence adalah skor keluaran detektor pada sampel yang terdeteksi. Nilai confidence tidak ditafsirkan sebagai probabilitas keselamatan.

Error jarak didefinisikan sebagai selisih estimasi jarak terhadap jarak referensi. MAE dihitung sebagai rata-rata nilai absolut error, sedangkan RMSE adalah akar rata-rata kuadrat error. MAE menunjukkan besar kesalahan tipikal, sedangkan RMSE memberi penalti lebih besar pada error yang tinggi.

2.10.2 Kesesuaian status Lane Safety

Kesesuaian Lane Safety dihitung dengan membandingkan pasangan status kiri dan kanan pada setiap capture terhadap kondisi acuan. Enam kondisi diuji masing-masing sebanyak 10 sampel. Nilai 100% pada pengujian ini berarti seluruh

pasangan status sesuai pada skenario terkontrol yang diuji, bukan berarti segmentasi sempurna untuk seluruh kondisi lingkungan.

2.10.3 FPS, waktu inferensi, waktu proses, dan frame age

Pada penelitian ini, istilah *Frames per Second* (FPS) yang digunakan sebagai parameter evaluasi performa tidak menyatakan frame rate input atau akuisisi kamera, melainkan throughput pemrosesan pipeline persepsi, yaitu laju frame yang telah menyelesaikan satu siklus pemrosesan pada pipeline YOLOPv2. Oleh karena itu, FPS yang dilaporkan pada hasil pengujian disebut sebagai *pipeline throughput* atau *pipeline FPS*.

Frame rate akuisisi kamera dan pipeline FPS merupakan dua parameter yang berbeda. Kamera Orbbec Astra Pro dikonfigurasi dengan request frame rate sebesar 15 FPS, yang menyatakan laju akuisisi citra yang diminta dari perangkat kamera. Setelah frame diperoleh, citra diproses pada Intel NUC melalui tahapan prapemrosesan, inferensi YOLOPv2, post-processing, pembentukan informasi persepsi, dan publikasi keluaran. Karena kemampuan pemrosesan pipeline pada CPU lebih rendah dibandingkan laju akuisisi citra yang diminta dari kamera, nilai throughput pipeline menjadi lebih rendah daripada request frame rate kamera.

Pipeline FPS ditentukan berdasarkan interval waktu antarpemrosesan siklus pemrosesan. Secara konseptual, pipeline FPS dinyatakan pada Persamaan (2.7).

$$\text{FPS}_{\text{pipeline}} = \frac{1}{\Delta t_{\text{completion}}} \quad (2.7)$$

Keterangan:

$\text{FPS}_{\text{pipeline}}$: throughput penyelesaian pipeline (frame/s),

$\Delta t_{\text{completion}}$: selang waktu antara selesainya satu siklus pipeline dan selesainya siklus berikutnya (s).

Nilai FPS sesaat kemudian dihaluskan menggunakan *exponential moving average* agar variasi waktu pemrosesan sesaat tidak menyebabkan perubahan nilai telemetri yang terlalu tajam. Parameter *inference_ms* menyatakan waktu yang diperlukan khusus untuk menjalankan proses *forward* model YOLOPv2. Parameter *processing_ms* menyatakan waktu pemrosesan pipeline dari awal hingga akhir

siklus sehingga mencakup proses yang lebih luas dibandingkan inference time. Sementara itu, `frame_age_ms` menunjukkan umur frame ketika frame mulai diproses dan tidak digunakan sebagai representasi latency end-to-end sensor hingga aktuator.

Dengan definisi tersebut, nilai FPS sekitar 1,6 FPS yang diperoleh pada penelitian ini merupakan throughput pemrosesan pipeline YOLOv2 pada CPU Intel NUC, bukan FPS input kamera.