BAB II
DASAR TEORI

2.1 Tinjauan Pustaka

Setelah melakukan studi literatur terhadap topik terkait baik dari jurnal didapati beberapa penelitian yang menjadi inspirasi oleh penulis. Berikut beberapa penelitian tersebut. Rincian data dan parameter terkait disajikan pada Tabel 2.1.

Tabel 2.1 Tinjauan Pustaka

No	Jurnal	Isi Utama	Target Pembaharuan
1	YOLO-v5 Variant Selection Algorithm Coupled with Representative Augmentations for Modelling Production-Based Variance in Automated Lightweight Pallet Racking Inspection	Penelitian ini mengembangkan sistem inspeksi pallet menggunakan YOLOv5 dengan fokus pada akurasi tinggi dan model ringan. Menggunakan teknik data augmentation berbasis kondisi nyata dan algoritma pemilihan varian YOLO. Hasil terbaik: mAP 96.8% dengan model ringan (1.9M parameter).	Jangkauan kamera yang lebih luas dan dan mencakup *obstacle* dinamis dan statis
2	YOLOv7: Real-Time Object Detection	Mengembangkan YOLOv7[9] dengan konsep trainable bag-of-freebies untuk meningkatkan akurasi tanpa menambah beban komputasi. Mencapai performa tinggi dalam real-time detection (FPS tinggi + AP tinggi).	Menerapkan sistem YOLO dengan akurasi yang tinggi sebagai algoritma pendeteksi *obstacle* dinamis dan statis pada objek yang bergerak (AGV forklift)
3	Pengendalian AGV Menggunakan Raspberry Pi dan Motor DC dengan Sistem Pengenalan Lintasan Menggunakan Kamera dan Sensor Inframerah	Merancang AGV berbasis Raspberry Pi + kamera + sensor inframerah untuk navigasi dan deteksi jalur. Menggunakan image processing (HSV) untuk deteksi lintasan. Hasil: AGV mampu bergerak akurat dan efisien.	Pengkombinasian dan pengembangan sensor untuk mapping, memperluas jangkauan, dan navigasi otonom

Sumber: Diolah penulis berdasarkan tinjauan pustaka pada Subbab 2.1.

Dari ketiga pedoman jurnal diatas, fokus penelitian diatas digunakan untuk landasan pembaharuan penelitian dengan mengkombinasikan YOLOv8 dan dikombinasikan pada robot AGV yang dilengkapi dengan kamera Astra Pro untuk mendukung persepsi visual dan LiDAR sebagai sensor navigasi. Algoritma YOLOv8 digunakan untuk mendeteksi obstacle dinamis dan pallet, sedangkan Dynamic Window Approach (DWA) digunakan sebagai local planner untuk membantu penghindaran rintangan dan navigasi pada peta yang dibuat menggunakan SLAM.

2.2 Autonomous Guided Vehicle (AGV) Forklift

Perkembangan industri modern menuntut adanya sistem otomasi yang mampu meningkatkan efisiensi, produktivitas, serta keselamatan kerja, khususnya dalam proses material handling di lingkungan manufaktur, logistik, dan pergudangan. Salah satu teknologi yang banyak digunakan untuk mendukung kebutuhan tersebut adalah Automated Guided Vehicle (AGV). AGV merupakan kendaraan otomatis yang dapat bergerak mengikuti jalur tertentu atau secara otonom tanpa memerlukan operator manusia secara langsung. Sejak pertama kali diperkenalkan pada tahun 1950-an, teknologi AGV terus mengalami perkembangan signifikan dan menjadi komponen penting dalam sistem distribusi dan transportasi internal industri [10]. Visualisasi yang dibahas ditunjukkan pada Gambar 2.1.



Gambar 2.1 AGV Forklift
Sumber: Rujukan konsep AGV mengacu pada Martínez-Barberá dan Herrero-Pérez [10].

Gambar 2.1 memperlihatkan bentuk umum AGV forklift sebagai kendaraan otomatis untuk material handling. Konfigurasi tersebut menjadi konteks penelitian karena proses kerja AGV membutuhkan kemampuan navigasi, deteksi lingkungan, serta penyelarasan terhadap pallet sebelum proses pengangkatan. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

Dalam implementasinya, AGV forklift memanfaatkan berbagai sensor seperti LiDAR, kamera, encoder roda, dan Inertial Measurement Unit (IMU) untuk mendukung proses persepsi lingkungan dan lokalisasi. Sensor LiDAR umumnya digunakan untuk pemetaan dan navigasi, sedangkan kamera digunakan untuk mendukung persepsi visual. Seiring dengan perkembangan teknologi computer vision, penggunaan kamera menjadi semakin penting karena mampu melakukan deteksi dan klasifikasi objek secara real-time dengan tingkat akurasi yang tinggi. Salah satu algoritma yang banyak digunakan adalah YOLO (You Only Look Once), yang dikenal memiliki kecepatan dan performa yang baik dalam mendeteksi objek secara langsung dalam satu tahap proses [12].

Meskipun teknologi AGV forklift telah berkembang pesat, masih terdapat beberapa tantangan utama dalam implementasinya. Tantangan tersebut meliputi

kemampuan navigasi yang aman di lingkungan kerja yang dinamis, terutama yang melibatkan interaksi dengan manusia dan kendaraanlain. Selain itu, sistem juga harus mampu mendeteksi dan menghindari halangan secara real-time untuk mencegah terjadinya kecelakaan. Tantangan lain yang tidak kalah penting adalah kemampuan dalam mengidentifikasi serta menentukan posisi pallet secara akurat, serta melakukan proses docking dengan presisi tinggi agar garpu dapat masuk dengan tepat ke dalam pallet.

Berdasarkan permasalahan tersebut, penelitian ini difokuskan pada pengembangan sistem deteksi visual berbasis YOLO untuk mengidentifikasi pallet serta mendukung proses positioning dan docking pada AGV forklift. Dengan memanfaatkan metode ini, diharapkan sistem mampu meningkatkan akurasi deteksi objek serta efisiensi dalam proses pengambilan dan pemindahan pallet secara otomatis.

2.3 Jetson Orin Nano

NVIDIA Jetson Orin Nano Developer Kit merupakan salah satu platform komputasi *embedded* berbasis AI yang dikembangkan oleh NVIDIA untuk mendukung aplikasi edge computing, khususnya pada bidang robotika, computer vision, dan autonomous system. Perangkat ini dirancang untuk memberikan performa tinggi dengan konsumsi daya yang relatif rendah, sehingga sangat cocok digunakan pada sistem mobile seperti *Automated Guided Vehicle* (AGV). Visualisasi yang dibahas ditunjukkan pada Gambar 2.2.



Gambar 2.2 NVIDIA Jetson Orin Nano
Sumber: NVIDIA Corporation [13].

Gambar 2.2 menunjukkan Jetson Orin Nano yang digunakan sebagai platform komputasi utama. Perangkat ini menyediakan sumber daya pemrosesan untuk akuisisi citra, inferensi model, komunikasi ROS 2, dan pengolahan data sistem; spesifikasi teknisnya dirangkum pada Tabel 2.2. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

Tabel 2.2 Spesifikasi NVIDIA Jetson Orin Nano

Komponen	Jetson Orin Nano 4GB	Jetson Orin Nano 8GB
Arsitektur GPU	NVIDIA Ampere (512 CUDA, 16 Tensor Cores)	NVIDIA Ampere (1024 CUDA, 32 Tensor Cores)
Performa AI	Hingga 20 TOPS	Hingga 40 TOPS (67 TOPS pada pembaruan *Super*)

Komponen	Jetson Orin Nano 4GB	Jetson Orin Nano 8GB
Prosesor (CPU)	6-core Arm Cortex-A78AE v8.2 64-bit	6-core Arm Cortex-A78AE v8.2 64-bit
Memori (RAM)	4GB 64-bit LPDDR5 (34 GB/s)	8GB 128-bit LPDDR5 (68 - 102 GB/s)
Penyimpanan	Eksternal (M.2 NVMe SSD / microSD)	Eksternal (M.2 NVMe SSD / microSD)
Video Encode	1080p30 (Diproses oleh CPU)	1080p30 (Diproses oleh CPU)
Video Decode	1x 4K60 / 2x 4K30 (H.265)	1x 4K60 / 2x 4K30 (H.265)
Konsumsi Daya	5W - 10W	7W - 15W

Sumber: NVIDIA Corporation [13].

Dalam konteks AGV forklift, Jetson Orin Nano berperan sebagai unit pemrosesan utama (main controller) yang menangani berbagai tugas seperti:

- Pemrosesan citra dari kamera (object detection menggunakan YOLO)
- Integrasi sensor (LiDAR, IMU, encoder)
- Perencanaan jalur dan navigasi berbasis ROS
- Pengambilan keputusan secara real-time

Keunggulan utama dari Jetson Orin Nano dibandingkan platform embedded lainnya adalah kemampuannya dalam menjalankan model AI dengan latensi rendah, sehingga sangat mendukung implementasi sistem deteksi objek dan navigasi otonom secara simultan dalam satu perangkat[13].

2.4 Orbbec Astra Pro

Orbbec Astra Pro merupakan kamera depth berbasis teknologi RGB-D yang mampu menangkap informasi warna (RGB) sekaligus kedalaman (depth) secara bersamaan. Kamera ini banyak digunakan dalam aplikasi robotika, terutama untuk keperluan persepsi lingkungan, pemetaan, serta deteksi objek berbasis jarak. Visualisasi yang dibahas ditunjukkan pada Gambar 2.3.



Gambar 2.3 Orbbec Astra Pro
Sumber: Orbbec Inc. [14].

Gambar 2.3 menunjukkan perangkat Orbbec Astra Pro yang digunakan sebagai sensor visual. Kamera menyediakan citra yang menjadi masukan sistem perception, sedangkan karakteristik perangkat yang relevan terhadap proses akuisisi dirangkum pada Tabel 2.3. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

Tabel 2.3 Spesifikasi Orbbec Astra Pro

Fitur	Spesifikasi
Teknologi depth	Structured Light 3D
Jangkauan depth efektif	~0,6 m – 8 m

Fitur	Spesifikasi
Resolusi RGB	1280 × 720 @ 30 fps
Resolusi Depth	640 × 480 @ 30 fps
Interface	USB 2.0 (UVC/Depth via SDK)
Kompatibilitas OS	Windows, Linux, Android (melalui SDK)
Power	USB 2.0 (5 V)
Konsumsi daya	< ~2.4 W (tipikal)
Ukuran	165 × 40 × 40 mm
Berat	~0.49 kg
Field of View (Depth)	≈ 60° H × ~49.5° V
Akurasi	±1–3 mm (umumnya di 1 m, tergantung kondisi)

Sumber: Orbbec Inc. [14].

Dalam implementasi pada AGV forklift, kamera Astra Pro memiliki beberapa fungsi utama, antara lain:

- Mendukung deteksi objek (pallet dan halangan) dengan informasi kedalaman
- Membantu estimasi jarak antara robot dan objek
- Mendukung proses positioning dan docking secara lebih presisi
- Integrasi dengan sistem computer vision seperti YOLO untuk meningkatkan akurasi deteksi

Keunggulan kamera Astra Pro dibandingkan kamera konvensional adalah kemampuannya dalam memberikan informasi kedalaman secara langsung,

sehingga sangat membantu dalam aplikasi navigasi dan manipulasi objek pada robot otonom. Dengan kombinasi data RGB dan depth, sistem dapat melakukan pengambilan keputusan yang lebih akurat dalam lingkungan yang kompleks[14].

2.5 Object Detection

Object detection merupakan salah satu tugas utama dalam *computer vision* yang bertujuan untuk mengenali jenis objek sekaligus menentukan lokasi setiap objek pada suatu citra. Berbeda dengan *image classification* yang hanya memberikan informasi kelas terhadap keseluruhan citra, *object detection* menghasilkan informasi kelas dan lokasi spasial setiap objek yang terdeteksi, umumnya dalam bentuk *bounding box*. Dengan demikian, keluaran *object detection* dapat digunakan untuk mengetahui objek apa yang terdapat pada citra dan di mana posisi objek tersebut berada [16]. Perkembangan metode berbasis *deep learning* telah meningkatkan kemampuan sistem deteksi dalam menghadapi variasi ukuran, posisi, sudut pandang, serta kondisi lingkungan yang berbeda. Zou *et al*. membagi perkembangan metode *object detection* modern ke dalam berbagai pendekatan, termasuk *two-stage detector* dan *one-stage detector*, dengan pertimbangan utama berupa akurasi deteksi dan efisiensi komputasi [16].

Pada *two-stage detector*, proses pendeteksian umumnya dilakukan melalui tahap pembentukan kandidat wilayah objek yang kemudian dilanjutkan dengan proses klasifikasi dan penyempurnaan lokasi objek. Sementara itu, *one-stage detector* melakukan prediksi kelas dan lokasi objek secara langsung dalam satu alur inferensi. Pendekatan *one-stage* memiliki keunggulan dalam kecepatan pemrosesan sehingga sesuai untuk aplikasi yang membutuhkan deteksi secara *real-time*, seperti sistem persepsi pada robot bergerak. YOLO (*You Only Look Once*) merupakan salah satu metode *one-stage object detector* yang melakukan prediksi kelas dan koordinat objek dalam satu proses jaringan [12].

Pada sistem *perception* penelitian ini, proses *object detection* digunakan sebagai tahap awal untuk mengubah citra RGB yang diperoleh dari kamera Orbbec Astra Pro menjadi informasi objek yang dapat diproses oleh tahap berikutnya. Setiap hasil deteksi memuat informasi kelas objek, nilai *confidence*, dan koordinat *bounding*

box. Hasil prediksi dengan nilai *confidence* di bawah nilai ambang tidak digunakan, sedangkan beberapa *bounding box* yang mendeteksi objek yang sama diseleksi menggunakan *Non-Maximum Suppression* (NMS) berdasarkan nilai *Intersection over Union* (IoU). Tahap tersebut menghasilkan sekumpulan objek terdeteksi yang lebih representatif sebelum informasi geometrinya digunakan oleh proses *alignment*.

Sesuai dengan model yang diimplementasikan, sistem *object detection* mengenali lima kelas, yaitu floor marking, block, front, hole pallet, dan pallet. Kelas *pallet* digunakan untuk mengetahui keberadaan pallet secara keseluruhan, sedangkan kelas *front*, *hole pallet*, dan *block* menyediakan informasi bagian-bagian pallet yang diperlukan oleh proses pengolahan geometri. Pemisahan komponen pallet menjadi fitur visual yang lebih spesifik penting karena proses *positioning* tidak hanya memerlukan keberadaan pallet, tetapi juga informasi struktur bagian depan serta jalur masuk *fork* [19], [20].

Hasil *object detection* tidak secara langsung digunakan untuk menentukan bahwa AGV telah berada pada kondisi *aligned*. Informasi kelas dan *bounding box* dari tahap deteksi terlebih dahulu diteruskan ke proses pengolahan fitur pallet. Pasangan deteksi *hole pallet* digunakan untuk memperoleh posisi tengah dan orientasi visual pallet, sedangkan deteksi *block* digunakan sebagai informasi tambahan untuk memvalidasi geometri pallet. Dengan demikian, *object detection* pada penelitian ini berfungsi sebagai lapisan persepsi awal yang menyediakan informasi objek dan lokasi pada bidang citra, sedangkan estimasi titik referensi, *error lateral*, *error yaw*, dan kondisi *alignment* dilakukan pada tahap pengolahan setelah proses deteksi.

2.6 YOLOv8 (*You Only Look Once Version 8*)

YOLO (*You Only Look Once*) merupakan metode *object detection* berbasis *deep learning* yang melakukan proses lokalisasi dan klasifikasi objek dalam satu tahap inferensi. Berbeda dengan pendekatan *two-stage detector* yang memisahkan proses pembentukan kandidat objek dan klasifikasi, YOLO memproses citra secara langsung untuk menghasilkan kelas objek, nilai *confidence*, serta koordinat *bounding box*. Pendekatan tersebut membuat YOLO sesuai digunakan pada aplikasi

yang membutuhkan proses deteksi secara *real-time*, termasuk sistem persepsi visual pada robot bergerak [12]. Perkembangan keluarga YOLO hingga YOLOv8 menghasilkan arsitektur yang semakin efisien dalam menyeimbangkan kecepatan inferensi dan akurasi deteksi [11].

Pada penelitian ini, YOLOv8 digunakan sebagai metode utama untuk mengekstraksi informasi objek dari citra RGB yang diperoleh melalui kamera Orbbec Astra Pro. Citra kamera diterima oleh sistem perception, kemudian diproses oleh model YOLOv8 untuk menghasilkan informasi objek yang terdeteksi. Keluaran hasil deteksi selanjutnya tidak langsung digunakan sebagai perintah kendali kendaraan, melainkan diteruskan ke tahap pengolahan berikutnya untuk menentukan fitur geometris pallet dan kondisi alignment. Dengan demikian, fungsi YOLOv8 pada sistem ini dibatasi pada proses deteksi dan lokalisasi objek pada bidang citra**.**

Pada implementasi penelitian, setiap hasil deteksi YOLOv8 memuat identitas kelas, nilai confidence, dan koordinat bounding box. Titik pusat objek diperoleh secara programatis dari koordinat bounding box dan selanjutnya digunakan sebagai masukan pada tahap pengolahan geometri pallet. Karena operasi tersebut hanya membentuk data internal deteksi dan tidak dianalisis sebagai metrik tersendiri pada Bab IV, bentuk matematis keluaran deteksi dan perhitungan pusat bounding box tidak dituliskan sebagai persamaan bernomor. Informasi koordinat tersebut selanjutnya digunakan oleh tahap pengolahan citra berikutnya untuk memperoleh posisi relatif fitur pallet.

2.6.1 **Kelas Objek pada Model YOLOv8**

Model YOLOv8 yang digunakan pada sistem perception dilatih untuk mengenali lima kelas objek. Kelas tersebut tidak hanya digunakan untuk mendeteksi pallet secara keseluruhan, tetapi juga memisahkan beberapa fitur pallet yang diperlukan pada proses alignment. Dataset laporan juga telah membedakan pallet menjadi `block`, `hole pallet`, `pallet`, dan bagian depan pallet serta satu kelas `floor`

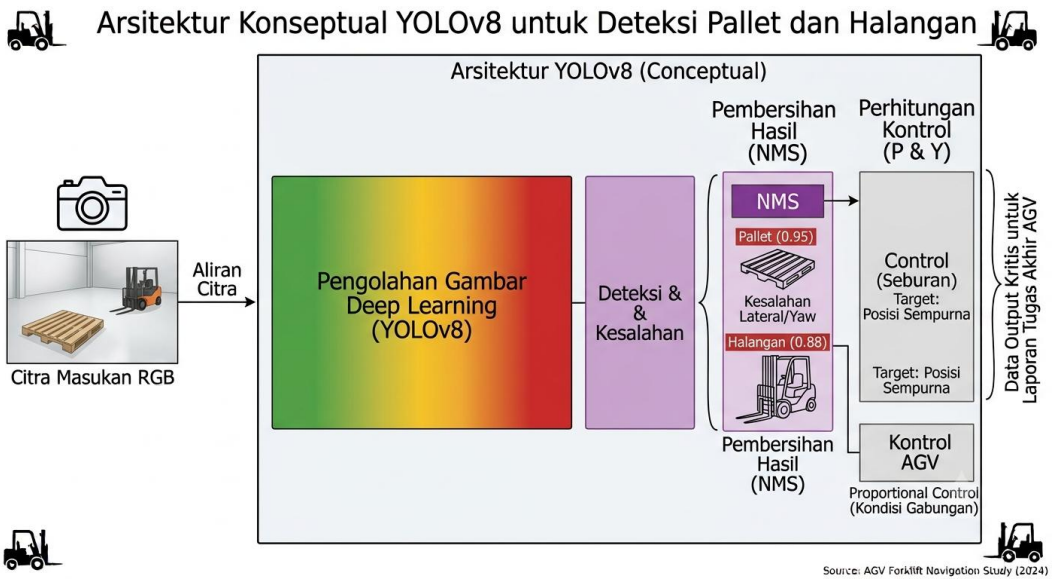
`marking`. Pada implementasi program, susunan kelas yang digunakan adalah sebagai berikut: Rincian data dan parameter terkait disajikan pada Tabel 2.4.

Tabel 2.4 Kelas Objek pada Model YOLOv8

ID Kelas	Nama kelas pada program	Fungsi dalam sistem perception
0	floor marking	Mendeteksi tanda visual pada lantai yang digunakan sebagai objek/area referensi visual
1	block	Mendeteksi bagian block pallet yang digunakan dalam validasi geometri dan penentuan posisi centerline pallet
2	front	Mendeteksi bagian depan pallet sebagai informasi visual tambahan mengenai keberadaan dan tampak pallet
3	hole pallet	Mendeteksi lubang masuk fork; pasangan hole digunakan untuk memperoleh posisi dan orientasi visual pallet
4	pallet	Mendeteksi pallet secara keseluruhan sebagai objek target utama

Sumber: Konfigurasi model penelitian dengan acuan YOLOv8 [11].

Dari lima kelas tersebut, kelas hole pallet dan block mempunyai fungsi penting dalam proses alignment. Dua hasil deteksi hole pallet digunakan untuk membentuk pasangan fitur visual yang merepresentasikan bagian kiri dan kanan lubang masuk fork. Posisi kedua fitur tersebut kemudian dapat digunakan untuk menentukan midpoint dan orientasi visual pallet. Selanjutnya, kelas block digunakan sebagai fitur tambahan untuk memvalidasi geometri pallet dan menentukan posisi yang diharapkan pada centerline pallet. Sementara itu, kelas pallet dan front menyediakan informasi mengenai keberadaan dan bagian depan pallet secara umum. Pembagian komponen pallet menjadi beberapa fitur visual juga sesuai dengan pendekatan pengenalan pallet yang memanfaatkan struktur bagian depan pallet untuk membantu estimasi posisi dan orientasi target [19]. Visualisasi yang dibahas ditunjukkan pada Gambar 2.4.



Gambar 2.4 Arsitektur Konseptual YOLOv8 untuk Deteksi Pallet dan Halangan
Sumber: Diolah penulis berdasarkan [11] dan [12].

Gambar 2.4 memperlihatkan alur konseptual YOLOv8 dari citra masukan hingga keluaran deteksi. Pada penelitian ini, keluaran berupa kelas, confidence, dan bounding box digunakan sebagai informasi awal sebelum fitur pallet diproses lebih lanjut untuk menghasilkan parameter alignment. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

Pada implementasi perangkat lunak, hasil akhir proses tersebut diterbitkan sebagai informasi objek melalui ROS 2 dan menjadi masukan bagi proses *hole-block alignment*. Oleh karena itu, YOLOv8 pada penelitian ini berfungsi sebagai lapisan persepsi awal yang mengubah citra RGB menjadi informasi kelas dan lokasi objek, sedangkan perhitungan error lateral, error yaw, validitas geometri, dan status alignment tidak dilakukan oleh YOLOv8, melainkan pada tahap pemrosesan sesudah deteksi.

2.6.2 **Proses Training YOLOv8**

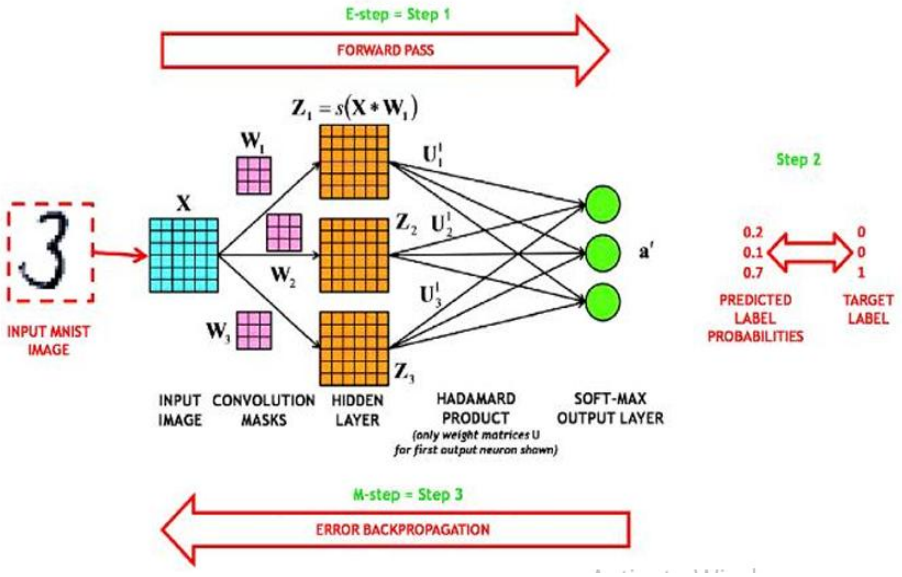
Proses *training* YOLOv8 bertujuan untuk mengoptimalkan parameter jaringan agar model mampu mengenali kelas objek dan menentukan lokasi objek secara akurat

berdasarkan data anotasi. Pada tahap awal, dataset yang telah diberi label dibagi menjadi data *training, validation,* dan *testing*. Data *training* digunakan untuk memperbarui bobot jaringan, sedangkan data *validation* digunakan untuk mengevaluasi kemampuan model selama proses pelatihan sehingga dapat diketahui apakah model mengalami peningkatan performa atau justru mengalami *overfitting*. Data *testing* selanjutnya digunakan untuk mengevaluasi performa model setelah proses pelatihan selesai [11].

Citra masukan terlebih dahulu melalui tahap *preprocessing* agar memiliki ukuran yang sesuai dengan dimensi masukan model. Setelah itu, citra diproses melalui *forward pass* pada jaringan YOLOv8. Informasi visual diekstraksi melalui bagian *backbone*, kemudian fitur dari beberapa tingkat resolusi digabungkan pada bagian *neck*. Fitur yang telah diproses selanjutnya diteruskan menuju *detection head* untuk menghasilkan prediksi kelas objek dan lokasi *bounding box*. Arsitektur multi-skala pada keluarga YOLO memungkinkan model mengenali objek dengan ukuran yang berbeda pada suatu citra [11].

Hasil prediksi dibandingkan dengan anotasi ground truth melalui fungsi loss yang mencakup komponen lokalisasi bounding box, klasifikasi, dan Distribution Focal Loss (DFL). Nilai loss digunakan dalam proses optimasi selama training dan dievaluasi berdasarkan keluaran training aktual, seperti log dan results.csv. Evaluasi difokuskan pada perubahan nilai loss selama proses pelatihan.

Nilai *loss* yang diperoleh selanjutnya digunakan pada proses *backpropagation* untuk menghitung gradien terhadap parameter jaringan. Optimizer kemudian memperbarui bobot jaringan agar nilai *loss* semakin kecil pada iterasi berikutnya. Proses tersebut dilakukan secara berulang dalam sejumlah *epoch* hingga diperoleh model dengan performa yang memenuhi kebutuhan sistem. Visualisasi yang dibahas ditunjukkan pada Gambar 2.5.



Gambar 2.5 Proses Training YOLOv8
Sumber: Diolah penulis berdasarkan Terven et al. [11].

Gambar 2.5 menunjukkan siklus pelatihan model mulai dari data masukan, perhitungan loss, backpropagation, hingga pembaruan bobot. Proses tersebut digunakan untuk memperoleh model yang mampu mengenali kelas objek yang dibutuhkan pada sistem perception AGV. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

Parameter seperti model variant, image size, jumlah epoch, batch size, learning rate, optimizer, serta konfigurasi augmentasi merupakan parameter implementasi. Nilai numeriknya mengikuti konfigurasi pelatihan yang digunakan dan disajikan pada Bab III, sedangkan Bab II membahas prinsip dasarnya.

2.6.3 **Intersection Over Union (IoU)**

Intersection over Union (IoU) merupakan metrik yang digunakan untuk mengukur tingkat kesesuaian antara *bounding box* hasil prediksi model dan *bounding box ground truth*. Nilai IoU diperoleh dengan membandingkan luas daerah perpotongan (*intersection*) antara kedua *bounding box* terhadap luas daerah gabungannya

(*union*) [17]. Metrik ini banyak digunakan dalam evaluasi *object detection* karena tidak hanya menilai apakah kelas objek berhasil dikenali, tetapi juga mengukur ketepatan posisi objek yang diprediksi.

Secara matematis, tingkat overlap antara bounding box prediksi dan ground truth dihitung menggunakan Persamaan (2.1).

$$IoU = \frac{|B_{pred} \cap B_{\square}|}{|B_{pred} \cup B_{\square}|} \quad (2.1)$$

Visualisasi yang dibahas ditunjukkan pada Gambar 2.6.

dengan:

- IoU = tingkat tumpang tindih antara bounding box prediksi dan bounding box ground truth.
- Bpred = bounding box hasil prediksi model.
- Bgt = bounding box ground truth.
- Bpred ∩ Bgt = luas daerah irisan bounding box prediksi dan ground truth.
- Bpred ∪ Bgt = luas daerah gabungan bounding box prediksi dan ground truth.
- Nilai IoU berada pada rentang 0 sampai 1.
- **Fungsi persamaan:** mengukur tingkat kesesuaian posisi bounding box hasil prediksi terhadap ground truth.

Ilustrasi hubungan antara bounding box prediksi dan ground truth ditunjukkan pada Gambar 2.6.

Gambar 2.6 Ilustrasi Intersection over Union (IoU)
Sumber: Diolah penulis berdasarkan Padilla et al. [17].

Gambar 2.6 memperlihatkan hubungan area tumpang tindih antara bounding box prediksi dan ground truth. Semakin besar proporsi irisan terhadap gabungan kedua area, semakin tinggi nilai IoU dan semakin dekat posisi prediksi terhadap anotasi acuan. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

Selain digunakan pada tahap evaluasi, konsep IoU juga digunakan pada proses *post-processing* hasil deteksi. Apabila beberapa *bounding box* mendeteksi objek yang sama dengan tingkat tumpang tindih yang tinggi, nilai IoU dapat digunakan dalam proses *Non-Maximum Suppression* (NMS) untuk menentukan *bounding box* yang dipertahankan. *Bounding box* dengan nilai *confidence* lebih tinggi dipilih, sedangkan prediksi lain yang memiliki overlap tinggi dapat dieliminasi. Dengan demikian, penggunaan IoU membantu mengurangi deteksi ganda terhadap objek yang sama dan menghasilkan keluaran deteksi yang lebih representatif.

Pada sistem perception penelitian ini, hasil deteksi setelah proses penyaringan tersebut selanjutnya diteruskan ke tahap pengolahan fitur pallet. Informasi *bounding box*, kelas objek, dan nilai *confidence* menjadi masukan untuk menentukan pasangan hole pallet, mendeteksi block, serta membentuk informasi geometris yang digunakan dalam proses alignment.

2.7 *Confusion Matrix*

Confusion Matrix merupakan salah satu metode evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan membandingkan hasil prediksi model terhadap label yang sebenarnya. Komponen yang digunakan ada 4 yaitu

- *True Positive* (TP) = merupakan kondisi ketika objek pada *ground truth* berhasil dideteksi dengan kelas yang benar dan nilai IoU memenuhi nilai ambang yang telah ditentukan
- *True Negative* (TN) = kondisi ketika model atau alat tes secara akurat memprediksi atau mengidentifikasi hasil yang bersifat negatif (tidak ada atau salah), dan kenyataannya kondisi tersebut memang benar-benar negatif
- *False Positive* (FP) = merupakan kondisi ketika model menghasilkan suatu deteksi, tetapi prediksi tersebut tidak mempunyai pasangan *ground truth* yang sesuai, kelas yang diprediksi salah, atau terjadi deteksi tambahan terhadap objek yang sama.
- *False Negative* (FN) = merupakan kondisi ketika terdapat objek pada *ground truth*, tetapi model tidak berhasil menghasilkan deteksi yang memenuhi kriteria kelas dan IoU Rincian data dan parameter terkait disajikan pada Tabel 2.5.

Tabel 2.5 Struktur Confusion Matrix

	Prediksi	
Actual	*True*	*false*
True	TP	FP
false	FN	TN

Sumber: Diolah penulis berdasarkan Padilla et al. [17].

Berdasarkan Tabel 2.5 terdapat empat nilai, yang menggambarkan sejauh mana model mampu mengklasifikasikan data benar atau salah. Confusion Matrix juga dapat digunakan untuk mengevaluasi model baik dengan dua kelas atau lebih. Selain sebagai alat ukur performa Confusion Matrix juga berfungsi sebagai visualisasi yang memudahkan analisi terhadap akurasi prediksi model. Berdasarkan confusion matrix, metrik utama yang digunakan pada penelitian ini adalah precision, recall, dan F1-Score [17], sedangkan akurasi diperlakukan sebagai informasi tambahan. Berikut merupakan penjelasan singkat dan persamaan metrik utama yang digunakan;

- Akurasi dapat digunakan sebagai informasi tambahan untuk melihat ketepatan prediksi secara umum. Namun, pada evaluasi object detection penelitian ini, metrik utama yang digunakan adalah precision, recall, F1-Score, Average Precision (AP), mean Average Precision (mAP), serta Intersection over Union (IoU).

- Presisi digunakan untuk seberapa akurat prediksi positif, persamaan berikut penting untuk menghindari terjadinya false alarm;

$$P = \frac{TP}{TP+FP} \tag{2.2}$$

dengan:

- P = precision.
- TP (True Positive) = jumlah objek yang diprediksi positif dan benar.
- FP (False Positive) = jumlah deteksi positif yang tidak sesuai dengan ground truth.
- **Fungsi persamaan:** mengukur ketepatan prediksi positif yang dihasilkan model.

- *Recall* digunakan untuk melihat seberapa banyak objek berhasil terdeteksi, persamaan ini penting untuk safety terhadap *obstacle* yang ingin di deteksi;

$$R = \frac{TP}{TP+FN} \tag{2.3}$$

dengan:

- R = recall.
- TP (True Positive) = jumlah objek yang berhasil dideteksi dengan benar.
- FN (False Negative) = jumlah objek ground truth yang tidak berhasil dideteksi.
- **Fungsi persamaan:** mengukur kemampuan model dalam menemukan objek yang sebenarnya tersedia pada data.
  - F1-Score digunakan untuk keseimbangan antara presisi & *recall;*

$$F_1 = 2\frac{PR}{P+R} \tag{2.4}$$

dengan:

- F1 = F1-Score.
- P = precision.
- R = recall.
- **Fungsi persamaan:** menunjukkan keseimbangan antara precision dan recall dalam satu nilai evaluasi.

Selain metrik precision, recall, dan F1-Score yang diperoleh dari confusion matrix, pada kasus object detection seperti penggunaan YOLOv8, evaluasi performa model tidak hanya bergantung pada klasifikasi yang benar atau salah. Model juga harus mampu menentukan lokasi objek secara tepat melalui bounding box. Oleh karena itu, diperlukan metrik evaluasi tambahan yang lebih komprehensif, yaitu Average Precision (AP) dan mean Average Precision (mAP).

Average Precision (AP) merupakan metrik yang mengukur performa model berdasarkan hubungan antara *precision* dan *recall* pada berbagai *threshold*. Nilai AP diperoleh dari luas area di bawah kurva *Precision-Recall* (PR Curve), sehingga mampu menggambarkan konsistensi model dalam mendeteksi objek dengan benar pada berbagai tingkat kepercayaan.

Selanjutnya, untuk mengevaluasi performa model secara keseluruhan pada beberapa kelas objek, digunakan *mean Average Precision* (mAP)**.** Nilai mAP merupakan rata-rata dari AP untuk seluruh kelas yang diuji. Pada implementasi object detection, perhitungan AP dan mAP juga mempertimbangkan nilai Intersection over Union (IoU) sebagai indikator kesesuaian antara bounding box prediksi dan ground truth.

Dengan demikian, kombinasi antara confusion matrix dan metrik AP serta mAP memberikan evaluasi yang lebih menyeluruh, tidak hanya dari sisi klasifikasi, tetapi juga dari ketepatan deteksi lokasi objek dalam sistem.

- *Average Precision* (AP): merupakan metrik yang mengukur kualitas model berdasarkan hubungan antara precision dan recall pada berbagai threshold confidence. Berikut merupakan persamaan;

$$AP = \int_0^1 P\,(r)dr \tag{2.5}$$

dengan:

- AP = Average Precision untuk satu kelas objek.
- P(r) = precision sebagai fungsi terhadap recall.
- r = recall.
- **Fungsi persamaan:** mengukur performa satu kelas berdasarkan hubungan precision dan recall pada berbagai threshold.

- *Mean Average Precision* (mAP): merupakan rata-rata nilai AP dari seluruh kelas, berikut merupakan persamaan yang digunakan;

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{2.6}$$

dengan:

- mAP = mean Average Precision.
- APi = Average Precision pada kelas ke-i.
- N = jumlah kelas yang dievaluasi.
- i = indeks kelas.
- **Fungsi persamaan:** menghitung rata-rata Average Precision dari seluruh kelas objek.

Untuk pengujian deteksi pada beberapa kondisi eksperimen, keberhasilan deteksi dinyatakan menggunakan detection rate. Detection rate dihitung sebagai perbandingan jumlah pengamatan yang menghasilkan deteksi valid terhadap seluruh pengamatan.

$$DR = \frac{N_{det}}{N_{total}} \times 100\% \tag{2.7}$$

dengan:

- DR = detection rate.
- Ndet = jumlah pengamatan dengan deteksi valid.
- Ntotal = jumlah seluruh pengamatan atau percobaan.
- **Fungsi persamaan:** menunjukkan persentase keberhasilan deteksi pada suatu kondisi pengujian.

Selain detection rate, nilai confidence rata-rata digunakan untuk menggambarkan tingkat keyakinan model pada sekumpulan hasil deteksi valid.

$$\acute{C} = \frac{1}{N}\sum_{i=1}^{N} C_i \tag{2.8}$$

dengan:

- $\bar{C}$ = confidence rata-rata.
- Ci = nilai confidence pada pengamatan ke-i.
- N = jumlah data confidence yang dianalisis.
- i = indeks pengamatan.
- **Fungsi persamaan:** memperoleh tingkat keyakinan rata-rata model pada sekumpulan hasil deteksi valid.

2.8 Deteksi Pallet

Pallet merupakan media penanganan material yang digunakan untuk mendukung proses penyimpanan, pengangkutan, dan pemindahan barang menggunakan forklift maupun *Automated Guided Vehicle* (AGV). Pada AGV forklift otonom, pallet tidak hanya diperlakukan sebagai objek yang harus dikenali keberadaannya, tetapi juga sebagai target visual yang memiliki struktur geometris tertentu untuk membantu proses positioning dan alignment. Oleh karena itu, sistem persepsi perlu mampu mengenali pallet serta bagian-bagian visualnya yang berhubungan dengan jalur masuk fork.

Metode berbasis visi telah banyak dikembangkan untuk melakukan pengenalan dan lokalisasi pallet pada sistem pergudangan. Wu *et al.* menunjukkan bahwa identifikasi komponen pallet dan lokalisasi slot pallet dapat digunakan untuk menentukan posisi pallet pada AGV, bahkan pada variasi jarak, sudut pandang, pencahayaan, dan latar belakang [20]. Sementara itu, Kai *et al.* menunjukkan bahwa informasi pada bagian depan pallet dapat dimanfaatkan untuk memperoleh informasi posisi dan orientasi pallet yang diperlukan oleh autonomous forklift [19].

Pada penelitian ini, deteksi pallet dilakukan menggunakan model YOLOv8 terhadap citra RGB. Model tidak hanya mendeteksi pallet sebagai satu objek utuh, tetapi membedakan beberapa bagian pallet menjadi kelas yang berbeda. Pendekatan tersebut diperlukan karena proses alignment membutuhkan informasi geometris yang lebih spesifik dibandingkan hanya mengetahui posisi *bounding box* pallet secara keseluruhan.

Dari lima kelas yang digunakan pada model, empat kelas berhubungan secara langsung dengan pallet, yaitu pallet, front, hole pallet, dan block. Sementara itu, kelas floor marking merupakan objek lingkungan dan tidak termasuk bagian struktur pallet. Informasi kelas, nilai *confidence*, dan koordinat *bounding box* dari setiap objek selanjutnya digunakan sebagai masukan pada proses pengolahan geometri pallet.

2.8.1 Struktur pallet sebagai fitur visual

Secara umum, pallet memiliki bagian permukaan atas sebagai tempat penempatan beban dan bagian bawah yang membentuk struktur penyangga serta ruang masuk fork. Struktur bagian depan pallet menjadi penting pada sistem AGV forklift karena fork harus diarahkan menuju ruang masuk pallet dengan posisi dan orientasi yang sesuai. Pada penelitian ini, bagian deckboard, stringer/block, dan fork entry diterjemahkan menjadi fitur visual yang dapat dikenali oleh model deteksi.

Kelas pallet digunakan untuk merepresentasikan pallet sebagai objek secara keseluruhan. Hasil deteksi kelas ini memberikan informasi mengenai keberadaan dan lokasi umum pallet pada bidang citra. Namun, pusat *bounding box* pallet secara

keseluruhan tidak digunakan sebagai satu-satunya referensi untuk menentukan alignment akhir karena posisi tersebut belum secara langsung menunjukkan posisi jalur masuk fork.

Kelas front digunakan untuk mengenali bagian depan pallet yang terlihat oleh kamera. Bagian depan pallet merupakan area penting karena di dalamnya terdapat struktur yang berhubungan dengan jalur masuk fork. Informasi mengenai tampak depan pallet juga relevan dalam estimasi pose pallet. Kai *et al*. menunjukkan bahwa struktur tampak depan dapat digunakan untuk memperoleh informasi posisi dan orientasi pallet pada autonomous forklift [19].

Kelas hole pallet merupakan label operasional pada model YOLOv8 yang digunakan untuk mengenali bukaan atau area masuk fork pada bagian depan pallet. Dalam sistem perception, kelas ini mempunyai peranan utama dalam proses alignment karena dua hasil deteksi hole pallet digunakan sebagai pasangan fitur kiri dan kanan. Kedua fitur tersebut selanjutnya digunakan untuk membentuk hubungan geometris yang merepresentasikan posisi tengah dan orientasi visual pallet.

Kelas block digunakan untuk mendeteksi bagian penyangga pallet yang berada dalam hubungan geometris dengan pasangan hole. Keberadaan block digunakan sebagai fitur tambahan untuk memvalidasi bahwa pasangan hole pallet yang ditemukan benar-benar berasal dari struktur pallet yang sesuai. Penggunaan informasi komponen dan slot pallet untuk positioning juga sejalan dengan penelitian Wu *et al.*, yang menunjukkan bahwa pemisahan struktur pallet menjadi beberapa komponen dapat meningkatkan kemampuan sistem dalam menentukan lokasi pallet [20].

Dengan demikian, struktur pallet yang paling relevan terhadap sistem perception dapat dikelompokkan seperti pada tabel berikut. Rincian data dan parameter terkait disajikan pada Tabel 2.6.

Tabel 2.6 Representasi Struktur Pallet sebagai Fitur Visual

Kelas YOLOv8	**Representasi visual**	**Fungsi pada sistem**
pallet	Pallet secara keseluruhan	Identifikasi keberadaan dan lokasi umum pallet

Kelas YOLOv8	Representasi visual	Fungsi pada sistem
front	Bagian depan pallet	Informasi visual bagian depan pallet
hole pallet	Bukaan/jalur masuk fork	Fitur utama pasangan kiri-kanan untuk alignment
block	Bagian penyangga/block pallet	Validasi geometri dan penentuan referensi centerline

Sumber: Diolah penulis berdasarkan [19], [20], [24], dan [29].

Pembagian tersebut membuat sistem tidak hanya menjawab pertanyaan apakah pallet terdeteksi, tetapi juga menyediakan fitur visual yang dapat digunakan untuk mengetahui apakah posisi relatif pallet sesuai dengan posisi fork. Tabel tersebut merangkum informasi utama yang digunakan pada pembahasan subbab ini. Data atau spesifikasi yang dicantumkan berfungsi sebagai acuan untuk menghubungkan dasar teori dengan konfigurasi sistem yang digunakan dalam penelitian.

2.8.2 Deteksi Komponen Pallet pada Sistem Perception

Proses deteksi komponen pallet dimulai dari citra RGB yang diproses menggunakan YOLOv8. Setiap hasil deteksi memiliki kelas, nilai *confidence*, dan koordinat *bounding box*. Deteksi yang memenuhi kriteria kemudian diteruskan ke tahap pengolahan geometri pallet. Pada tahap tersebut, sistem terutama menggunakan kelas hole pallet dan block, sedangkan kelas pallet dan front memberikan informasi semantik mengenai keberadaan pallet dan bagian depannya.

Ketika terdapat beberapa deteksi hole pallet, sistem menentukan kandidat pasangan hole kiri dan kanan. Pasangan tersebut tidak langsung dianggap valid hanya berdasarkan keberadaan dua *bounding box*. Sistem juga memeriksa kesesuaian geometrinya, seperti jarak horizontal antar-hole, perbedaan posisi vertikal, kesesuaian ukuran kedua deteksi, serta kestabilan posisinya terhadap frame sebelumnya. Validasi tersebut diperlukan untuk mengurangi kemungkinan dua deteksi yang tidak berasal dari pasangan lubang pallet yang sama digunakan dalam perhitungan alignment.

Setelah pasangan hole pallet dinyatakan valid, posisi pusat dari kedua *bounding box* digunakan untuk membentuk fitur geometris pallet. Hubungan antara kedua titik

tersebut selanjutnya digunakan untuk memperoleh titik tengah pasangan hole dan orientasi bagian depan pallet. Informasi tersebut menjadi dasar dalam menentukan centerline visual pallet dan menghitung error yaw pada tahap berikutnya.

Selain pasangan hole, sistem mencari objek kelas block pada area yang diperkirakan berdasarkan geometri pasangan hole. Deteksi block selanjutnya dibandingkan terhadap posisi block yang diharapkan. Perbedaan antara posisi aktual hasil deteksi dan posisi yang diperkirakan digunakan sebagai salah satu kriteria validasi geometri. Dengan demikian, keberadaan dua hole saja belum cukup untuk menyatakan data pallet valid; hubungan antara hole dan block juga digunakan untuk meningkatkan keandalan hasil perception.

Secara konseptual, hasil deteksi yang valid digunakan untuk memilih pasangan hole pallet, membentuk geometri bagian depan pallet, dan memvalidasi posisi block. Tahapan tersebut merupakan alur pemrosesan sehingga dijelaskan dalam bentuk narasi/diagram proses, bukan sebagai persamaan matematis.

Pendekatan berbasis komponen seperti ini sesuai dengan kebutuhan autonomous pallet handling karena ketepatan positioning tidak hanya bergantung pada keberhasilan mengenali pallet sebagai satu objek, tetapi juga pada kemampuan sistem memperoleh fitur yang berhubungan secara langsung dengan jalur masuk fork [20], [19].

Parameter numerik seperti nilai minimum confidence, batas jarak antar-hole, batas perbedaan vertikal, ukuran ROI, dan batas residual block merupakan bagian dari konfigurasi implementasi dan disajikan pada Bab III.

2.9 Titik Referensi *Pallet*

Titik referensi pallet merupakan informasi posisi visual yang digunakan untuk merepresentasikan lokasi target pada bidang citra. Setiap objek yang terdeteksi direpresentasikan oleh bounding box dengan batas kiri, kanan, atas, dan bawah. Pusat setiap bounding box dihitung secara programatis dan digunakan sebagai representasi posisi fitur pada bidang citra. Karena perhitungan pusat bounding box

hanya merupakan tahap antara dan tidak dianalisis sebagai metrik tersendiri pada Bab IV, bentuk matematisnya tidak diberi nomor persamaan secara terpisah.

Dalam proses positioning pallet, penggunaan karakteristik geometris pallet dapat memberikan informasi yang lebih relevan dibandingkan hanya menggunakan pusat bounding box pallet secara keseluruhan. Kai et al. menggunakan bounding box pallet body, left pallet hole, dan right pallet hole dari citra RGB untuk memperoleh informasi koordinat dua dimensi yang selanjutnya digunakan dalam estimasi posisi pallet [19]. Pendekatan lain juga menunjukkan bahwa identifikasi komponen pallet dan pallet slot merupakan bagian penting dalam proses positioning AGV untuk mendukung proses pengambilan pallet secara otomatis [20].

Pada penelitian ini, titik referensi alignment dibentuk dari dua hole pallet yang terdeteksi. Pusat hole kiri dinyatakan sebagai H_L = (x_L, y_L) dan pusat hole kanan sebagai H_R = (x_R, y_R). Midpoint pasangan hole kemudian dihitung menggunakan Persamaan (2.9).

$$M = \left(\frac{x_L + x_R}{2}, \frac{y_L + y_R}{2}\right) \tag{2.9}$$

dengan:

- M = midpoint atau titik tengah pasangan hole pallet.
- xL, yL = koordinat pusat hole pallet kiri.
- xR, yR = koordinat pusat hole pallet kanan.
- **Fungsi persamaan:** menentukan titik tengah geometris pasangan hole sebagai referensi pembentukan geometri pallet.

Penentuan pasangan lubang dilakukan setelah kedua deteksi memenuhi kriteria validitas yang ditetapkan oleh sistem. Kriteria tersebut meliputi tingkat kepercayaan deteksi, jarak antar-lubang, perbedaan posisi vertikal, kesesuaian ukuran kedua *bounding box*, serta perubahan posisi terhadap frame sebelumnya. Validasi tersebut diperlukan agar dua objek yang dipilih benar-benar merepresentasikan pasangan lubang pada pallet yang sama dan bukan hasil deteksi lain yang memiliki kelas serupa.

Dari pusat hole kiri dan kanan dibentuk vektor arah antar-hole, yaitu vektor yang menghubungkan kedua fitur tersebut. Agar arah vektor tidak dipengaruhi oleh panjangnya, vektor tersebut dinormalisasi menjadi vektor satuan sebagaimana dinyatakan pada Persamaan (2.10).

$$\hat{h} = \frac{h}{\|h\|} = \begin{bmatrix} h_x \\ h_y \end{bmatrix} \tag{2.10}$$

dengan:

- h = vektor arah antar-hole sebelum normalisasi.
- ||h|| = magnitudo atau panjang vektor h.
- $\hat{h}$ = vektor satuan arah antar-hole.
- hx, hy = komponen horizontal dan vertikal vektor satuan.
- **Fungsi persamaan:** memperoleh arah garis antar-hole tanpa dipengaruhi oleh panjang vektornya.

Komponen vektor satuan pada Persamaan (2.10) selanjutnya digunakan untuk membentuk vektor normal yang tegak lurus terhadap garis antar-hole, sebagaimana dirumuskan pada Persamaan (2.11).

$$n = \begin{bmatrix} -h_y \\ h_x \end{bmatrix} \tag{2.11}$$

dengan:

- n = vektor normal terhadap garis antar-hole.
- hx = komponen horizontal vektor satuan antar-hole.
- hy = komponen vertikal vektor satuan antar-hole.
- **Fungsi persamaan:** membentuk arah yang tegak lurus terhadap garis antar-hole untuk penentuan referensi geometris pallet.

Pada implementasi penelitian, arah normal dipilih menuju bagian bawah citra sehingga sesuai dengan arah geometris posisi block terhadap pasangan hole pallet.

Titik referensi utama yang digunakan dalam proses alignment selanjutnya adalah expected block center. Titik ini menggambarkan posisi pusat block yang diharapkan

berdasarkan midpoint pasangan hole, arah normal, serta offset geometri yang telah ditentukan. Hubungan tersebut dirumuskan pada Persamaan (2.12).

$$B_{exp} = M + d_{\parallel} n + d_{\perp} \hat{h} \tag{2.12}$$

dengan:

- Bexp = expected block center.
- M = midpoint pasangan hole pallet.
- n = vektor normal terhadap garis antar-hole.
- ĥ = vektor satuan arah antar-hole.
- d∥ = offset sepanjang arah normal sesuai konfigurasi geometri.
- d⊥ = offset sepanjang arah antar-hole sesuai konfigurasi geometri.
- **Fungsi persamaan:** menentukan posisi pusat block yang diharapkan berdasarkan geometri pasangan hole pallet.

dengan B_exp merupakan expected block center, M merupakan midpoint pasangan hole, d_parallel merupakan offset sepanjang arah normal n, sedangkan d_perp merupakan offset sepanjang arah vektor satuan antar-hole. Kedua offset tersebut merupakan parameter implementasi yang ditentukan melalui konfigurasi atau kalibrasi sistem, sehingga nilai numeriknya dijelaskan pada Bab III.

Deteksi objek kelas block selanjutnya digunakan sebagai validasi terhadap expected block center. Posisi pusat block hasil deteksi dibandingkan dengan posisi yang diperkirakan oleh sistem. Dengan demikian, block tidak digunakan sebagai satu-satunya pembentuk titik referensi, tetapi digunakan untuk memastikan bahwa geometri yang diperoleh dari pasangan hole pallet konsisten dengan struktur pallet yang terdeteksi.

Expected block center kemudian dibandingkan dengan referensi visual fork. Titik referensi visual ditetapkan melalui konfigurasi dan kalibrasi posisi kamera terhadap fork, sedangkan komponen horizontalnya digunakan pada perhitungan error lateral dalam Persamaan (2.15). Dengan demikian, koordinat referensi tidak perlu dituliskan sebagai persamaan tersendiri.

2.9.1 Orientasi Visual *Pallet*

Orientasi visual pallet digunakan untuk menentukan arah relatif pallet pada bidang citra selama proses *alignment*. Pada penelitian ini, orientasi visual tidak diperoleh dari Hough Line Transform, sisi luar pallet, maupun orientasi block, tetapi ditentukan berdasarkan garis yang menghubungkan pusat dua deteksi hole pallet. Pendekatan tersebut digunakan karena kedua lubang merupakan fitur geometris yang berada pada bagian depan pallet dan digunakan secara langsung dalam algoritma perception untuk membentuk arah referensi pallet.

Pusat hole kiri H_L=(x_L,y_L) dan hole kanan H_R=(x_R,y_R) digunakan secara langsung untuk menentukan orientasi garis antar-hole. Selisih koordinat horizontal dan vertikal tidak dituliskan sebagai persamaan terpisah karena hanya menjadi komponen masukan fungsi atan2 pada Persamaan (2.13).

Orientasi visual garis yang menghubungkan kedua hole dihitung menggunakan fungsi four-quadrant inverse tangent sebagaimana dinyatakan pada Persamaan (2.13).

$$\theta_h = atan2(y_R - y_L, x_R - x_L) \tag{2.13}$$

dengan:

- θh = orientasi visual garis yang menghubungkan pasangan hole pallet.
- xL, yL = koordinat pusat hole pallet kiri.
- xR, yR = koordinat pusat hole pallet kanan.
- atan2 = fungsi four-quadrant inverse tangent untuk menentukan orientasi berdasarkan selisih koordinat.
- **Fungsi persamaan:** menghitung orientasi visual pallet pada bidang citra.

dengan θ_h merupakan orientasi visual garis antar-hole terhadap sumbu horizontal citra. Jika hasil diperlukan dalam satuan derajat, konversi dari radian ke derajat dilakukan di dalam program dan tidak diperlakukan sebagai persamaan analisis tersendiri.

Penggunaan fungsi atan2 memungkinkan tanda dan kuadran sudut ditentukan berdasarkan perubahan koordinat horizontal dan vertikal kedua titik. Sistem koordinat citra menggunakan titik asal di bagian kiri atas, dengan sumbu horizontal bertambah ke arah kanan dan sumbu vertikal bertambah ke arah bawah. Oleh karena itu, konvensi tanda orientasi perlu dipertahankan secara konsisten selama proses pengolahan dan pengujian.

Orientasi visual kemudian dibandingkan dengan orientasi referensi θ_ref. Karena sudut bersifat periodik, selisih langsung perlu dinormalisasi agar tidak menimbulkan lompatan nilai ketika melewati batas ±180°.

Normalisasi error yaw tersebut dirumuskan pada Persamaan (2.14).

$$e_{yaw} = wrap_{180}(\theta_h - \theta_{ref}) \tag{2.14}$$

dengan:

- eyaw = error yaw visual.
- θh = orientasi visual garis antar-hole.
- θref = orientasi atau yaw referensi.
- wrap180 = fungsi normalisasi sudut ke rentang -180° sampai kurang dari 180°.
- **Fungsi persamaan:** menghitung penyimpangan orientasi visual pallet terhadap orientasi referensi.

Fungsi wrap_180 pada Persamaan (2.14) menjaga error yaw tetap berada pada rentang -180° hingga kurang dari 180°. Normalisasi ini diperlukan karena orientasi merupakan besaran periodik.

Sebagai ilustrasi, orientasi 179° dan −179° secara geometris hanya berbeda 2°, sehingga keduanya tidak tepat jika diperlakukan memiliki selisih sebesar 358°.

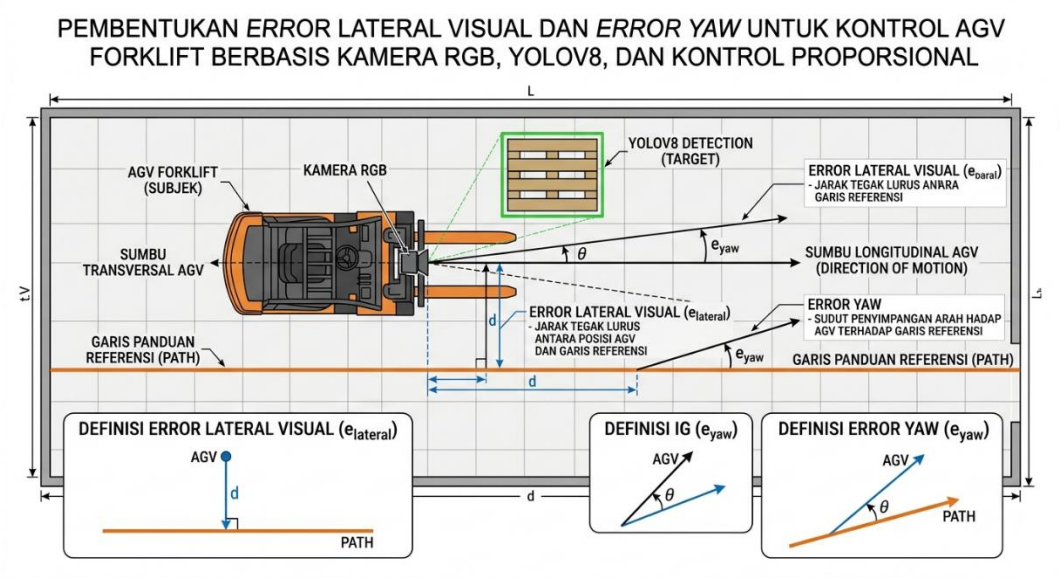
Nilai θ_ref merupakan yaw offset yang mendefinisikan orientasi visual referensi. Ketika error yaw mendekati nol, garis yang menghubungkan kedua hole telah sesuai dengan orientasi referensi; tanda positif atau negatif menunjukkan arah penyimpangan sesuai konvensi koordinat citra.

Perhitungan orientasi dilakukan hanya apabila pasangan *hole pallet* telah memenuhi kriteria validitas. Apabila salah satu lubang tidak terdeteksi, nilai *confidence* tidak memenuhi batas yang ditentukan, jarak kedua lubang berada di luar rentang yang diizinkan, atau perubahan posisi antar-frame melebihi batas yang ditetapkan, maka pasangan lubang tidak digunakan untuk menghasilkan informasi orientasi. Pendekatan tersebut mencegah hasil orientasi yang berasal dari fitur visual yang tidak valid diteruskan sebagai informasi *alignment*.

Orientasi yang diperoleh pada penelitian ini merupakan orientasi visual relatif pada bidang citra, bukan pengukuran yaw global kendaraan ataupun pose tiga dimensi pallet. Penggunaan fitur visual sebagai informasi umpan balik merupakan konsep yang umum digunakan dalam *image-based visual servoing*, yaitu konfigurasi target ditentukan berdasarkan hubungan antara fitur aktual pada citra dan fitur visual referensi [26]. Oleh karena itu, nilai error yaw pada penelitian digunakan untuk menunjukkan penyimpangan visual pallet terhadap kondisi alignment yang diinginkan dan selanjutnya dapat digunakan pada proses pengolahan *alignment*.

2.10 *Error Lateral Visual* dan *Error Yaw*

Pada proses penyelarasan (*alignment*) AGV forklift terhadap pallet, informasi hasil deteksi objek perlu diubah menjadi parameter geometris yang dapat menunjukkan perbedaan antara posisi pallet yang teramati dan kondisi penyelarasan yang diinginkan. Pada pendekatan *image-based visual servoing* (IBVS), informasi tersebut dapat dibentuk secara langsung pada bidang citra dengan membandingkan fitur visual yang terukur terhadap fitur visual referensi. Selisih antara kondisi aktual dan kondisi referensi kemudian dinyatakan sebagai *visual error* [27]. Konsep penggunaan informasi posisi dan orientasi pada bidang citra juga telah diterapkan pada proses penyelarasan robot bergerak terhadap suatu target visual [19]. Visualisasi yang dibahas ditunjukkan pada Gambar 2.7.



Gambar 2.7 Pembentukan Error Lateral Visual dan Error Yaw
Sumber: Diolah penulis berdasarkan [15], [26], dan [27].

Gambar 2.7 menggambarkan pembentukan error lateral dan error yaw dari fitur visual pallet terhadap referensi citra. Error lateral mewakili penyimpangan posisi horizontal, sedangkan error yaw mewakili penyimpangan orientasi yang digunakan sebagai masukan proses alignment. Gambar tersebut memperjelas konsep yang dibahas pada subbab terkait dan menghubungkan uraian teoritis dengan komponen atau parameter yang digunakan dalam penelitian. Interpretasinya tetap dibatasi pada fungsi yang dijelaskan dalam teks dan sumber rujukan yang dicantumkan.

2.10.1 *Error Lateral Visual*

Error lateral visual menyatakan penyimpangan posisi expected block center terhadap referensi visual fork pada arah horizontal. Expected block center dan setpoint visual telah didefinisikan pada tahap geometri dan konfigurasi sistem; hanya komponen horizontal keduanya yang digunakan pada Persamaan (2.15).

$$e_{lat,px} = x_{B,exp} - x_s \tag{2.15}$$

dengan:

- elat,px = error lateral visual dalam satuan piksel.
- xB,exp = koordinat horizontal expected block center.

- xs = koordinat horizontal setpoint visual fork.
- **Fungsi persamaan:** menghitung penyimpangan horizontal target terhadap referensi visual fork.

Tanda error lateral menunjukkan arah penyimpangan. Nilai yang mendekati nol menunjukkan centerline pallet telah sesuai dengan referensi visual, nilai negatif menunjukkan target berada di sisi kiri referensi, dan nilai positif menunjukkan target berada di sisi kanan referensi.

Dengan konvensi tanda tersebut, error lateral dapat digunakan sebagai indikator arah penyimpangan target terhadap referensi visual fork. Representasi berbasis piksel sesuai dengan prinsip image-based visual servoing karena informasi visual dapat dimanfaatkan secara langsung tanpa terlebih dahulu merekonstruksi posisi tiga dimensi [27], [19].

Setpoint visual fork tidak harus berimpit dengan pusat citra. Nilainya ditetapkan melalui konfigurasi atau kalibrasi posisi kamera terhadap fork sehingga perubahan pemasangan kamera tidak langsung diasumsikan mengubah referensi menjadi pusat frame.

Pendekatan ini menjaga definisi error lateral tetap konsisten ketika posisi pemasangan kamera atau area pemrosesan berubah, selama nilai setpoint visual diperbarui melalui proses konfigurasi atau kalibrasi yang sesuai.

Apabila diperlukan representasi dalam satuan fisik dan kalibrasi piksel terhadap jarak aktual telah dinyatakan valid, error lateral dikonversi ke satuan meter menggunakan Persamaan (2.16).

$$e_{lat,m} = k_{m/px} e_{lat,px} \tag{2.16}$$

dengan:

- elat,m = error lateral dalam satuan meter.
- km/px = faktor konversi meter per piksel.
- elat,px = error lateral visual dalam satuan piksel.

- **Fungsi persamaan:** mengubah error lateral dari domain piksel ke satuan fisik ketika kalibrasi metrik dinyatakan valid.

2.10.2 Error Yaw Visual

Error yaw visual diperoleh dari orientasi garis yang dibentuk oleh pasangan hole pallet. Pusat hole kiri dan kanan digunakan secara langsung pada Persamaan (2.13) untuk memperoleh orientasi visual θ_h. Nilai tersebut kemudian dibandingkan dengan orientasi referensi dan dinormalisasi menggunakan Persamaan (2.14). Oleh karena itu, rumus koordinat hole, selisih koordinat, konversi derajat, dan normalisasi sudut tidak diulang pada subbab ini.

Nilai error yaw yang mendekati nol menunjukkan orientasi visual pallet telah sesuai dengan referensi, sedangkan tanda positif dan negatif menunjukkan arah penyimpangan menurut sistem koordinat citra. Nilai ini merupakan orientasi relatif pada bidang citra dan bukan yaw global kendaraan atau pose tiga dimensi pallet.

Error lateral dan error yaw mempunyai fungsi yang berbeda tetapi saling melengkapi dalam menentukan kondisi alignment. Error lateral mengevaluasi posisi centerline pallet terhadap referensi fork, sedangkan error yaw mengevaluasi kesesuaian orientasi pallet. Konsep penggunaan informasi posisi dan orientasi secara simultan juga ditemukan pada sistem visual servoing untuk penyelarasan robot bergerak, di mana posisi serta orientasi visual digunakan untuk membawa robot menuju konfigurasi target [27].

Error lateral dan error yaw digunakan secara bersama untuk menentukan alignment. Kedua nilai harus berada di dalam batas toleransi yang ditetapkan pada tahap implementasi. Batas numerik disajikan pada Bab III, sedangkan penggabungan kedua syarat diformalkan melalui kriteria alignment akhir pada Persamaan (2.27).

Pada implementasinya, nilai error yang dihitung pada setiap frame dapat mengalami perubahan akibat variasi hasil deteksi objek. Oleh karena itu, nilai error mentah (raw error) diproses melalui tahapan filtering dan validasi sebelum digunakan sebagai keluaran akhir sistem perception. Pembahasan mengenai filtering,

penolakan outlier, dan kestabilan hasil deteksi ditempatkan pada subbab tersendiri agar tidak bercampur dengan definisi dasar error lateral dan error yaw.

2.10.3 Error forward

Error forward visual merupakan parameter yang digunakan untuk menunjukkan penyimpangan posisi pallet terhadap posisi referensi pada arah maju atau longitudinal AGV. Berbeda dengan error lateral yang menunjukkan pergeseran target ke arah kiri atau kanan dan error yaw yang menunjukkan perbedaan orientasi, error forward digunakan untuk mengetahui apakah posisi AGV masih terlalu jauh, sudah berada pada posisi pendekatan yang diinginkan, atau telah bergerak terlalu dekat terhadap pallet. Dengan demikian, ketiga error tersebut mempunyai fungsi yang berbeda dalam proses docking, yaitu lateral untuk posisi horizontal, yaw untuk orientasi, dan forward untuk posisi sepanjang arah gerak kendaraan.

Pada konsep *visual servoing*, posisi robot terhadap target dapat dikendalikan berdasarkan selisih antara fitur visual aktual dan fitur visual yang diinginkan pada citra. Perubahan fitur pada bidang citra dapat digunakan sebagai informasi umpan balik untuk membawa robot menuju konfigurasi target [26], [27]. Konsep tersebut digunakan pada penelitian ini untuk membentuk error forward berdasarkan perubahan posisi vertikal fitur pallet terhadap titik referensi visual yang telah ditentukan.

Sistem koordinat citra yang digunakan memiliki titik asal pada bagian kiri atas, dengan koordinat horizontal bertambah ke arah kanan dan koordinat vertikal bertambah ke arah bawah. Oleh karena itu, informasi posisi vertikal dapat digunakan untuk mengamati perubahan posisi relatif pallet selama AGV bergerak mendekati target. Ketika AGV berada lebih jauh dari pallet, fitur pallet cenderung berada lebih tinggi terhadap posisi referensi pada citra. Sebaliknya, ketika AGV mendekati pallet, posisi fitur akan bergeser terhadap referensi sesuai dengan karakteristik perspektif kamera. Penggunaan fitur visual sebagai umpan balik posisi juga umum digunakan pada sistem docking robot berbasis *visual servoing* [28].

Pada penelitian ini, titik yang digunakan untuk membentuk error forward mengikuti *expected block center* yang telah diperoleh dari geometri pasangan *hole pallet*. Jika koordinat vertikal *expected block center* dinyatakan sebagai yB,expy_{B,exp}, sedangkan koordinat vertikal referensi forward dinyatakan sebagai yf,refy_{f,ref}, maka error forward visual dalam domain piksel dapat dinyatakan sebagai:

efwd,px=yf,ref−yB,exp(2.17)e_{fwd,px}=y_{f,ref}-y_{B,exp} \tag{2.17}

dengan:

- efwd,px = error forward visual dalam satuan piksel.
- yf,ref = koordinat vertikal referensi forward pada kondisi docking yang diinginkan.
- yB,exp = koordinat vertikal expected block center hasil pengolahan geometri pallet.
- **Fungsi persamaan:** menghitung penyimpangan posisi pallet terhadap referensi forward pada bidang citra.

Konvensi tanda pada Persamaan (2.17) dibuat agar nilai error memberikan informasi arah koreksi yang mudah digunakan pada tahap berikutnya. Nilai error forward positif menunjukkan bahwa posisi visual target masih berada sebelum titik referensi sehingga AGV masih memerlukan pergerakan maju. Nilai yang mendekati nol menunjukkan bahwa target telah mencapai posisi forward yang ditentukan, sedangkan nilai negatif menunjukkan bahwa posisi target telah melewati referensi visual yang digunakan. Konvensi tanda tersebut harus dipertahankan secara konsisten pada proses kalibrasi, pencatatan data, dan pengujian sistem.

Error forward dalam piksel merupakan keluaran langsung dari sistem berbasis citra dan tidak dapat langsung dianggap sebagai jarak fisik yang sama pada seluruh daerah kerja. Hal ini disebabkan hubungan antara perpindahan objek di ruang nyata dan perpindahannya pada bidang citra dipengaruhi oleh perspektif kamera serta posisi relatif kamera terhadap target. Dalam *image-based visual servoing*, hubungan antara perubahan fitur citra dan gerakan robot memang bergantung pada konfigurasi geometris serta parameter kamera [26], [27]. Oleh sebab itu, nilai error

forward dalam satuan fisik hanya digunakan setelah dilakukan proses kalibrasi pada rentang kerja yang ditentukan.

Untuk pengujian akurasi, posisi longitudinal aktual dapat ditentukan menggunakan jarak fisik antara referensi fork dan pallet sebagai *ground truth*. Jika jarak aktual terhadap pallet dinyatakan sebagai daktuald_{aktual} dan jarak referensi docking sebagai drefd_{ref}, maka error forward aktual dapat dinyatakan sebagai:

efwd,aktual=daktual−dref(2.18)e_{fwd,aktual}=d_{aktual}-d_{ref} \tag{2.18}

dengan:

- efwd,aktual = error forward aktual sebagai ground truth pengujian.
- daktual = jarak longitudinal aktual terhadap pallet.
- dref = jarak referensi docking.
- **Fungsi persamaan:** menghitung selisih posisi longitudinal aktual terhadap jarak referensi docking.

Nilai positif menunjukkan bahwa jarak aktual masih lebih besar dibandingkan jarak yang diinginkan, nilai mendekati nol menunjukkan bahwa posisi longitudinal telah sesuai dengan referensi, sedangkan nilai negatif menunjukkan bahwa kendaraan telah bergerak melewati posisi longitudinal yang ditentukan. Nilai ini digunakan sebagai *ground truth* pada tahap pengujian untuk dibandingkan dengan error forward yang diperoleh dari sistem visual.

Penggunaan informasi posisi relatif terhadap pallet penting pada proses docking karena keberhasilan pengambilan pallet tidak hanya bergantung pada kesesuaian posisi lateral dan orientasi kendaraan. Robot juga harus berada pada jarak pendekatan yang sesuai sebelum proses pemasukan fork dilakukan. Penelitian mengenai docking robot menunjukkan bahwa informasi visual dapat digunakan sebagai umpan balik untuk mengarahkan robot menuju konfigurasi target [28], sedangkan penelitian mengenai estimasi pose pallet pada lingkungan pergudangan menunjukkan pentingnya memperoleh informasi posisi dan orientasi pallet secara andal sebelum proses pengambilan dilakukan [29].

Dengan demikian, error forward tidak digunakan untuk menggantikan error lateral maupun error yaw, tetapi melengkapi kedua parameter tersebut. Error lateral digunakan untuk mengevaluasi posisi kiri dan kanan terhadap referensi fork, error yaw digunakan untuk mengevaluasi kesesuaian orientasi, sedangkan error forward digunakan untuk mengevaluasi posisi AGV sepanjang arah pendekatan terhadap pallet. Kombinasi ketiga parameter tersebut memungkinkan kondisi docking dievaluasi berdasarkan posisi horizontal, orientasi, dan jarak pendekatan secara terpisah sebelum sistem menyatakan kendaraan siap melakukan proses *insertion*.

2.11 Kontrol Proporsional *Docking*

Kontrol proporsional merupakan metode pengendalian yang menghasilkan keluaran berdasarkan besar error yang diterima dan nilai penguatan (*gain*) yang diberikan. Pada pengendali proporsional, nilai keluaran berubah secara linear terhadap nilai error, sehingga semakin besar penyimpangan terhadap kondisi referensi maka semakin besar pula nilai koreksi yang dihasilkan [15]. Pada konsep *visual servoing*, informasi error visual dapat digunakan sebagai masukan untuk membentuk sinyal koreksi yang membawa kondisi visual aktual menuju kondisi visual yang diinginkan [26], [27].

Pada penelitian ini, kontrol proporsional digunakan sebagai pengali atau pembobot terhadap error lateral dan error yaw yang dihasilkan oleh sistem perception. Error lateral menyatakan penyimpangan posisi *centerline* pallet terhadap referensi visual fork, sedangkan error yaw menyatakan penyimpangan orientasi pallet terhadap orientasi referensi. Kedua error tersebut mempunyai karakteristik dan satuan yang berbeda, sehingga masing-masing diberikan nilai gain secara terpisah.

Keluaran koreksi proporsional untuk komponen lateral diperoleh dengan mengalikan error lateral terhadap gain lateral sebagaimana dirumuskan pada Persamaan (2.17).

$$u_{lat} = K_{lat} e_{lat} \tag{2.17}$$

dengan:

- ulat = keluaran koreksi akibat error lateral.
- Klat = gain atau faktor pengali error lateral.
- elat = error lateral hasil sistem perception.
- **Fungsi persamaan:** menghasilkan koreksi proporsional berdasarkan besar error lateral.

Dengan cara yang sama, koreksi akibat error yaw dihitung menggunakan gain yaw sebagaimana dirumuskan pada Persamaan (2.18).

$$u_{yaw} = K_{yaw} e_{yaw} \tag{2.18}$$

dengan:

- uyaw = keluaran koreksi akibat error yaw.
- Kyaw = gain atau faktor pengali error yaw.
- eyaw = error yaw hasil sistem perception.
- **Fungsi persamaan:** menghasilkan koreksi proporsional berdasarkan besar error yaw.

Nilai Klat dan Kyaw berfungsi untuk mengatur besar kontribusi masing-masing error terhadap keluaran koreksi. Penggunaan nilai gain yang berbeda diperlukan karena error lateral dan error yaw tidak selalu memiliki skala numerik maupun satuan yang sama. Error lateral pada sistem perception dapat dinyatakan dalam piksel atau satuan metrik setelah proses kalibrasi, sedangkan error yaw dinyatakan dalam satuan sudut. Oleh karena itu, kedua error tidak dapat langsung dijumlahkan tanpa terlebih dahulu diberikan faktor pengali yang sesuai.

Apabila kedua komponen digunakan secara bersamaan, koreksi total diperoleh dari penjumlahan kontribusi lateral dan yaw sebagaimana dirumuskan pada Persamaan (2.19).

$$u_{corr} = K_{lat} e_{lat} + K_{yaw} e_{yaw} \tag{2.19}$$

dengan:

- ucorr = keluaran koreksi total.
- Klat = gain lateral.
- elat = error lateral.
- Kyaw = gain yaw.
- eyaw = error yaw.
- **Fungsi persamaan:** menggabungkan kontribusi koreksi lateral dan yaw menjadi satu keluaran koreksi.

Persamaan tersebut menunjukkan bahwa kontrol proporsional pada penelitian ini tidak menghasilkan informasi posisi baru, tetapi mengubah besar error yang telah dihitung oleh sistem perception menjadi nilai koreksi berdasarkan gain masing-masing. Pendekatan serupa digunakan pada *image-based visual servoing*, di mana error antara fitur visual aktual dan fitur referensi dipetakan menjadi masukan pengendalian untuk mengurangi penyimpangan visual [26], [27]. Pemanfaatan informasi visual untuk proses penyelarasan dan *docking* juga telah diterapkan pada sistem robot bergerak berbasis *visual servoing* [28].

Tanda keluaran koreksi mengikuti tanda error yang telah didefinisikan. Ketika error lateral atau error yaw mendekati nol, kontribusi koreksi dari komponen tersebut juga semakin kecil; sebaliknya, nilai absolut error yang lebih besar menghasilkan koreksi proporsional yang lebih besar untuk gain yang sama.

Nilai gain Klat dan Kyaw merupakan parameter implementasi yang ditentukan melalui pengujian. Nilai gain tidak ditetapkan pada dasar teori karena besar gain yang sesuai bergantung pada skala error, karakteristik sistem, serta besar keluaran yang dibutuhkan oleh sistem berikutnya. Taherian *et al.* menunjukkan bahwa nilai *control gain* mempunyai pengaruh terhadap perilaku sistem *image-based visual servoing*, sehingga penentuan gain perlu disesuaikan dengan karakteristik sistem yang dikendalikan [15]. Oleh karena itu, nilai numerik Klat dan Kyaw yang digunakan pada penelitian ini dijelaskan pada tahap perancangan dan pengujian di Bab III dan Bab IV.

Secara keseluruhan, kontrol proporsional menggunakan error lateral dan error yaw yang telah divalidasi sebagai masukan, memberikan bobot melalui gain masing-

masing, lalu menggabungkan keduanya menjadi koreksi total sebagaimana Persamaan (2.19). Alur tersebut dijelaskan sebagai proses kontrol dan tidak perlu dituliskan kembali sebagai rangkaian persamaan terpisah.

2.11.1 *Gain* Proporsional

Gain proporsional menentukan sensitivitas keluaran kendali terhadap perubahan error. Pada penelitian ini hubungan matematis yang digunakan telah dinyatakan secara langsung pada Persamaan (2.17) sampai Persamaan (2.19), sehingga bentuk umum pengendali proporsional tidak dituliskan kembali sebagai persamaan terpisah.

Pada sistem docking, besar gain menentukan besarnya koreksi steering terhadap error lateral dan error yaw. Pemilihan gain dilakukan melalui pengujian karena respons akhir dipengaruhi oleh skala error, dinamika kendaraan, kecepatan operasi, dan karakteristik aktuator.

Nilai *gain* yang relatif rendah menghasilkan perubahan keluaran kendali yang lebih kecil untuk nilai *error* yang sama. Kondisi tersebut dapat menghasilkan respons koreksi yang lebih halus, tetapi waktu yang diperlukan untuk membawa sistem menuju daerah toleransi dapat menjadi lebih panjang. Sebaliknya, peningkatan *gain* menghasilkan aksi kendali yang lebih besar sehingga respons sistem dapat menjadi lebih cepat. Namun, penguatan yang terlalu agresif dapat mengurangi tingkat *robustness*, meningkatkan kecenderungan osilasi, serta menghasilkan perubahan keluaran kendali yang lebih besar [15]. Oleh karena itu, peningkatan nilai (K_p) tidak selalu menghasilkan kinerja *docking* yang lebih baik.

Batas keluaran aktuator juga dipertimbangkan pada penalaan gain. Apabila hasil koreksi melebihi rentang perintah yang diizinkan, program membatasi keluaran pada nilai minimum atau maksimum aktuator. Mekanisme saturasi tersebut menjaga keluaran kendali tetap berada dalam rentang kerja aktuator.

Pada penelitian ini, nilai *gain* tidak ditentukan berdasarkan kategori subjektif seperti “kecil”, “besar”, atau “baik”, tetapi berdasarkan nilai numerik yang diuji pada kondisi eksperimen yang sama. Beberapa kandidat (Kp) diterapkan secara bertahap, kemudian respons sistem dibandingkan menggunakan indikator yang

telah ditentukan. Pendekatan penalaan berdasarkan respons *closed-loop* merupakan salah satu cara untuk memperoleh kompromi antara kecepatan respons, performa pengendalian, dan *robustness* sistem [15].

Kategori gain rendah digunakan untuk menyatakan kandidat (Kp) yang menghasilkan aksi koreksi relatif kecil dan respons menuju kondisi referensi yang lebih lambat dibandingkan kandidat lainnya. Sementara itu, gain terpilih didefinisikan sebagai nilai (Kp) yang memberikan hasil terbaik berdasarkan kriteria evaluasi yang telah ditetapkan dalam penelitian. Dengan demikian, penetapan *gain* terpilih tidak didasarkan pada pengamatan bahwa pergerakan kendaraan terlihat lebih baik, tetapi ditentukan berdasarkan hasil pengukuran.

Parameter evaluasi yang digunakan dapat mencakup *settling time*, nilai maksimum *error*, *overshoot*, jumlah perubahan tanda *error* yang menunjukkan kecenderungan osilasi, *steady-state error*, persentase waktu keluaran berada pada kondisi saturasi, dan keberhasilan sistem mencapai serta mempertahankan daerah toleransi. Seluruh kandidat *gain* diuji menggunakan kondisi awal dan prosedur yang sama sehingga perbedaan respons dapat dikaitkan dengan perubahan nilai (Kp).

Setiap pengujian *gain* perlu dicatat bersama parameter sistem yang relevan. Data tersebut sekurang-kurangnya meliputi nilai (K

p), identitas percobaan, *timestamp* atau nomor *frame*, nilai *error*, keluaran kendali, batas saturasi aktuator, serta indikator respons yang diperoleh. Pencatatan tersebut memungkinkan proses pemilihan *gain* ditelusuri kembali dan hasil pengujian antarparameter dibandingkan secara kuantitatif.

Pada sistem berbasis kamera RGB monokular, proses penalaan *gain* juga perlu mempertimbangkan bahwa *error* kendali diperoleh dari proyeksi objek pada bidang citra. Hubungan antara titik pada ruang tiga dimensi dan koordinat pada citra ditentukan oleh model proyeksi kamera serta parameter intrinsik dan ekstrinsiknya [18]. Oleh karena itu, besar perubahan *error visual* tidak selalu memiliki hubungan fisik yang sama pada seluruh posisi dan jarak operasi.

Sebagai contoh, perubahan posisi relatif antara kamera dan *pallet* dapat mengubah ukuran maupun lokasi proyeksi fitur pada citra. Konsekuensinya, nilai *error* visual yang sama tidak diasumsikan selalu merepresentasikan besar penyimpangan fisik yang identik pada seluruh kondisi *docking*. Dalam penelitian ini, kondisi tersebut ditangani dengan menggunakan *error* sebagai ukuran penyelarasan relatif pada bidang citra dan membatasi penerapan pengendali pada rentang operasi yang telah ditetapkan melalui pengujian.

Dengan demikian, nilai *gain* proporsional akhir dipilih berdasarkan performa sistem pada rentang operasi penelitian. Nilai tersebut dianggap memenuhi kebutuhan pengendalian apabila mampu membawa *error* menuju batas toleransi dengan waktu respons yang dapat diterima, tanpa menghasilkan osilasi berlebihan dan tanpa menyebabkan keluaran aktuator berada pada saturasi secara dominan. Pendekatan ini memungkinkan pemilihan parameter kendali dijelaskan berdasarkan data eksperimen dan bukan berdasarkan penilaian subjektif operator.

2.11.2 Saturasi dan *Deadband*

Saturasi digunakan untuk membatasi steering command pada rentang minimum dan maksimum yang diizinkan oleh aktuator [15]. Sementara itu, deadband digunakan untuk mengabaikan error yang sangat kecil agar sistem tidak terus melakukan koreksi akibat noise pada hasil deteksi kamera. Nilai deadband ditetapkan lebih kecil daripada batas toleransi alignment dan diverifikasi melalui pengujian agar tidak mengganggu proses docking [26], [27].

Parameter Δmax digunakan untuk menunjukkan batas maksimum perubahan *steering command* dalam satu periode kendali. Nilai tersebut perlu ditentukan secara numerik dan dicatat selama pengujian agar setiap percobaan dapat dilakukan dengan kondisi yang sama. Pencatatan meliputi nilai Δmax, waktu pengujian, nilai *error*, dan keluaran *steering*. Dengan demikian, evaluasi sistem dapat dilakukan berdasarkan data pengujian dan tidak hanya berdasarkan pengamatan visual.

Parameter clip menunjukkan kondisi ketika nilai keluaran kendali telah mencapai batas maksimum atau minimum yang ditentukan. Kondisi *clip* dicatat dari keluaran program selama proses pengujian. Informasi tersebut digunakan untuk

mengetahui seberapa sering aktuator bekerja pada batas kemampuannya. Penggunaan batas kendali seperti ini penting karena aktuator memiliki keterbatasan fisik yang harus diperhitungkan dalam perancangan sistem kendali [15].

Pada sistem *docking* berbasis kamera RGB dan YOLOv8, setiap bagian memiliki fungsi yang berbeda. YOLOv8 digunakan untuk mendeteksi keberadaan dan posisi *pallet* pada citra, kemudian informasi tersebut digunakan untuk membentuk *error lateral* dan *error yaw*. Selanjutnya, kontrol proporsional menggunakan kedua *error* tersebut untuk menentukan koreksi *steering*. Pendekatan *visual servoing* juga menggunakan informasi visual sebagai masukan untuk menghasilkan aksi kendali secara *closed-loop* [26], [27]. Oleh karena itu, apabila terjadi kegagalan *docking*, penyebabnya perlu dianalisis berdasarkan tahapan deteksi, pembentukan *error*, atau pengendalian sehingga sumber masalah dapat ditentukan dengan lebih tepat

2.11.3 Parameter Respons Sistem

Respons sistem *docking* dievaluasi berdasarkan settling time, overshoot, final error, dan kestabilan. *Settling time* merupakan waktu yang diperlukan sistem hingga *error* masuk dan tetap berada dalam batas toleransi yang telah ditentukan. *Overshoot* menunjukkan besarnya respons yang melewati nilai target, sedangkan *final error* menunjukkan sisa kesalahan antara kondisi akhir dan nilai referensi [22], [23]. Parameter-parameter tersebut umum digunakan untuk menilai kecepatan, ketepatan, dan karakteristik respons sistem kendali.

Pada penelitian ini, settling time ditentukan sebagai waktu sejak proses pengendalian dimulai hingga *error* masuk dan tetap berada dalam batas toleransi. Overshoot dihitung berdasarkan penyimpangan maksimum respons yang melewati nilai target, sedangkan final error diperoleh dari nilai *error* pada akhir pengujian. Kestabilan diamati dari perubahan *error* setelah mendekati target, yaitu apakah *error* semakin kecil dan stabil atau masih mengalami osilasi. Penggunaan parameter waktu respons seperti *settling time*, *overshoot*, dan *steady-state error* dapat digunakan sebagai dasar evaluasi kinerja sistem kendali [22], [23].

Setiap parameter dicatat berdasarkan hasil pengujian sehingga penilaian kinerja tidak hanya berdasarkan pengamatan visual. Data yang digunakan meliputi

nilai *error*, waktu pengujian, dan keluaran kendali. Seluruh pengujian dilakukan dengan kondisi awal dan prosedur yang sama agar hasil setiap percobaan dapat dibandingkan secara objektif. Dengan demikian, *gain* yang digunakan dapat dipilih berdasarkan parameter respons yang memberikan kinerja *docking* terbaik.

Keberhasilan keseluruhan proses docking pada beberapa kali percobaan dinyatakan menggunakan success rate. Nilai ini menunjukkan persentase jumlah percobaan yang memenuhi seluruh kriteria keberhasilan terhadap jumlah total percobaan sebagaimana dirumuskan pada Persamaan (2.20).

$$SR = \frac{N_{berhasil}}{N_{total}} \times 100\% \tag{2.20}$$

dengan:

- SR = success rate.
- Nberhasil = jumlah percobaan yang memenuhi kriteria keberhasilan docking.
- Ntotal = jumlah seluruh percobaan.
- **Fungsi persamaan:** menunjukkan persentase keberhasilan proses docking dari seluruh percobaan.

2.12 Filtering dan Validasi Temporal

Hasil deteksi objek berbasis citra dapat mengalami perubahan antar-*frame* akibat variasi posisi *bounding box*, perubahan pencahayaan, *noise* pada proses deteksi, maupun perubahan sudut pandang kamera. Apabila nilai posisi dan orientasi hasil deteksi digunakan secara langsung tanpa proses penyaringan, perubahan kecil pada hasil deteksi dapat menyebabkan nilai *error lateral* dan *error yaw* ikut berfluktuasi. Pada sistem *visual servoing*, keandalan informasi visual merupakan aspek penting karena hasil ekstraksi fitur pada citra digunakan sebagai dasar dalam menentukan kondisi relatif antara robot dan target [26], [27]. Selain itu, sistem berbasis kamera bekerja secara diskrit berdasarkan *frame*, sehingga perubahan informasi visual antarwaktu perlu diperhatikan dalam proses pengolahan [21].

Pada sistem *perception* penelitian ini, hasil pengukuran visual tidak langsung digunakan sebagai keluaran akhir. Data terlebih dahulu melalui proses filtering, penolakan outlier, dan validasi temporal. Proses tersebut digunakan untuk

mengurangi pengaruh perubahan sesaat pada hasil deteksi serta mencegah sistem menyatakan kondisi *alignment* hanya berdasarkan satu *frame*. Secara umum, alur pengolahan data dapat dinyatakan sebagai

```
deteksi objek→validasi geometri→filtering→penolakan outlier→validasi
 temporal.
```

Dengan tahapan tersebut, informasi yang diteruskan sebagai hasil *perception* merupakan data yang telah memenuhi kriteria geometris dan mempunyai kestabilan terhadap waktu.

2.12.1 Exponential Moving Average

Proses filtering menggunakan Exponential Moving Average (EMA) untuk memperhalus perubahan nilai antar-frame. EMA memberikan bobot pada data terbaru sesuai smoothing factor α. Pada penelitian ini, EMA diterapkan pada error lateral dan error yaw sebagaimana dinyatakan pada Persamaan (2.21) sampai Persamaan (2.23).

Nilai α menentukan tingkat respons *filter* terhadap data baru. Nilai α yang relatif besar menyebabkan keluaran lebih cepat mengikuti perubahan pengukuran, sedangkan nilai yang lebih kecil menghasilkan keluaran yang lebih halus tetapi memberikan respons yang lebih lambat terhadap perubahan. Oleh karena itu, nilai α perlu ditentukan berdasarkan karakteristik perubahan hasil deteksi dan kebutuhan respons sistem.

Pada algoritma *perception* penelitian ini, proses *smoothing* diterapkan pada informasi posisi fitur *hole pallet*, *error lateral*, dan *error yaw*. Penyaringan posisi fitur digunakan untuk mengurangi perubahan posisi titik referensi yang disebabkan oleh variasi *bounding box*, sedangkan penyaringan terhadap *error lateral* dan *error yaw* digunakan untuk menghasilkan nilai penyimpangan yang lebih stabil sebelum digunakan pada proses berikutnya.

Untuk error lateral, proses filtering menggunakan Exponential Moving Average (EMA) sebagaimana dirumuskan pada Persamaan (2.21).

$$\hat{e}_{lat}(k) = \alpha_{lat} e_{lat}(k) + (1 - \alpha_{lat}) \hat{e}_{lat}(k-1) \quad (2.21)$$

dengan:

- elat[k] = error lateral hasil filtering pada frame ke-k.
- elat raw[k] = error lateral mentah pada frame ke-k.
- elat[k-1] = error lateral hasil filtering pada frame sebelumnya.
- αlat = smoothing factor untuk error lateral.
- k = indeks frame atau waktu diskrit.
- **Fungsi persamaan:** memperhalus perubahan error lateral antar-frame menggunakan Exponential Moving Average.

Error yaw merupakan besaran periodik, sehingga selisih sudut antar-frame terlebih dahulu dinormalisasi untuk mencegah loncatan semu ketika nilai melewati batas 180° atau -180°. Selisih sudut ternormalisasi dihitung menggunakan Persamaan (2.22).

$$\Delta\theta_k = wrap_{180}\left(e_{yaw}(k) - \hat{e}_{yaw}(k-1)\right) \quad (2.22)$$

dengan:

- Δθk = perubahan error yaw ternormalisasi pada frame ke-k.
- eyaw[k] = error yaw pada frame ke-k.
- eyaw[k-1] = error yaw pada frame sebelumnya.
- wrap180 = fungsi normalisasi sudut.
- **Fungsi persamaan:** memperoleh perubahan yaw antar-frame tanpa loncatan semu pada batas ±180°.

Berdasarkan selisih sudut tersebut, filtered yaw error diperbarui menggunakan Persamaan (2.23).

$$\hat{e}_{yaw}(k) = wrap_{180}\left[\hat{e}_{yaw}(k-1) + \alpha_{yaw}\Delta\theta_k\right] \quad (2.23)$$

dengan:

- eyaw[k] = error yaw hasil filtering pada frame ke-k.
- eyaw[k-1] = error yaw hasil filtering pada frame sebelumnya.
- αyaw = smoothing factor untuk error yaw.

- $\Delta\theta k$ = perubahan yaw ternormalisasi.
- **Fungsi persamaan:** memperhalus perubahan error yaw dengan mempertahankan sifat periodik sudut.

Penggunaan normalisasi tersebut mencegah kondisi dua nilai sudut yang sebenarnya berdekatan, misalnya 179° dan −179°, dianggap mempunyai perubahan sebesar 358°.

Nilai numerik α_lat dan α_yaw serta parameter smoothing posisi fitur merupakan parameter implementasi yang ditetapkan berdasarkan konfigurasi program dan disajikan pada Bab III.

2.12.2 Penolakan Outlier

Selain *filtering*, sistem menggunakan mekanisme penolakan *outlier* untuk mencegah perubahan hasil deteksi yang tidak wajar diteruskan sebagai informasi posisi atau orientasi. *Outlier* dapat muncul akibat kesalahan deteksi sementara, perubahan *bounding box* secara mendadak, objek lain yang terklasifikasi sebagai fitur pallet, maupun hilangnya sebagian fitur visual. Permasalahan *outlier* pada fitur citra merupakan salah satu faktor yang dapat menurunkan keandalan sistem *visual servoing*[32], sehingga informasi visual yang tidak konsisten perlu ditangani sebelum digunakan pada proses berikutnya. Pendekatan *robust visual servoing* juga menekankan pentingnya mengurangi pengaruh kesalahan dan *outlier* pada hasil pengolahan citra[32].

Konsistensi temporal diperiksa berdasarkan perubahan posisi pasangan hole, expected block center, error yaw, dan error lateral antar-frame. Kandidat yang berubah melebihi threshold tidak digunakan untuk memperbarui hasil filter. Nilai threshold ditetapkan sebagai parameter implementasi dan disajikan pada Bab III.

Dengan mekanisme tersebut, *filtering* dan *outlier rejection* mempunyai fungsi yang berbeda. *Filtering* digunakan untuk memperhalus variasi data yang masih dianggap valid, sedangkan *outlier rejection* digunakan untuk menolak perubahan yang berada di luar batas konsistensi yang telah ditentukan.

Nilai TH, TB, Tyaw,jump, dan Tlat,jump merupakan parameter implementasi yang nilainya disajikan pada Bab III.

2.12.1 Validasi Kestabilan Antar-Frame

Hasil pengukuran yang memenuhi toleransi pada satu *frame* belum cukup untuk menyatakan bahwa kondisi visual telah stabil. Deteksi objek dapat secara sesaat menghasilkan posisi yang tepat akibat fluktuasi *bounding box*, meskipun kondisi pada *frame* berikutnya kembali berubah. Oleh karena itu, penelitian ini menggunakan mekanisme stable frame untuk mengevaluasi konsistensi kondisi selama beberapa *frame* secara berturut-turut.

Validasi temporal dilakukan hanya pada data yang telah memenuhi validitas geometri. Error lateral dan error yaw hasil filtering kemudian dibandingkan dengan batas toleransinya. Kondisi gabungan tersebut dinyatakan sebagai C_tol, yaitu kondisi ketika kedua error berada di dalam batas yang ditetapkan. Definisi formal validitas geometri, stable-frame counter, dan kondisi stabil dirangkum pada Persamaan (2.24) sampai Persamaan (2.26).

Nilai N_req menyatakan jumlah minimum frame berturut-turut yang harus memenuhi validitas geometri dan toleransi sebelum data dinyatakan stabil.

Mekanisme tersebut menyebabkan penghitung kestabilan kembali ke nol apabila salah satu kondisi tidak terpenuhi, misalnya pasangan *hole pallet* hilang, geometri *block* tidak valid, *error lateral* keluar dari toleransi, atau *error yaw* melebihi batas yang ditentukan. Pada implementasi program, hilangnya pasangan *hole pallet* juga menyebabkan keadaan *filter* yang berkaitan dengan posisi dan error di-*reset*. Hal tersebut mencegah data lama digunakan kembali ketika fitur visual baru terdeteksi setelah kehilangan target.

Dengan demikian, kondisi *detection stable* tidak ditentukan hanya berdasarkan keberhasilan mendeteksi pallet pada satu citra, tetapi berdasarkan konsistensi fitur geometris dan error selama sejumlah frame secara berturut-turut. Hasil filtering dan validasi temporal selanjutnya digunakan sebagai masukan dalam penentuan status *alignment* pada subbab berikutnya.

Secara keseluruhan, filtering dan validasi temporal membentuk alur: data deteksi divalidasi secara geometris, disaring, diperiksa terhadap outlier, kemudian diuji kestabilannya antar-frame. Alur ini dijelaskan sebagai urutan proses dan tidak memerlukan persamaan tambahan.

Alur tersebut membuat keputusan penyelarasan lebih tahan terhadap noise, perubahan bounding box sesaat, dan kesalahan pada satu frame.

2.13 Kriteria *Alignment* dan *Docking*

Kriteria *alignment* digunakan untuk menentukan apakah posisi relatif AGV forklift terhadap pallet telah memenuhi kondisi penyelarasan yang diperlukan sebelum proses berikutnya dilakukan. Pada sistem berbasis visi, kondisi *alignment* tidak cukup ditentukan hanya berdasarkan keberhasilan model dalam mendeteksi pallet. Informasi hasil deteksi perlu menunjukkan bahwa fitur geometris pallet dapat diidentifikasi secara konsisten, posisi lateral telah berada dalam batas toleransi, orientasi pallet telah mendekati orientasi referensi, dan kondisi tersebut bertahan selama beberapa frame secara berurutan. Dalam *visual servoing*, konfigurasi yang diinginkan direpresentasikan melalui fitur visual referensi dan sistem berusaha memperkecil selisih antara fitur aktual dan fitur referensi tersebut [26].

Pada penelitian ini, penentuan *alignment* memanfaatkan fitur visual pada bagian depan pallet. Informasi tersebut diperoleh dari hasil deteksi kelas hole pallet dan `block`. Penggunaan fitur bagian pallet untuk memperoleh informasi posisi relatif sesuai dengan penelitian mengenai *pallet positioning* dan *pose estimation*, yang menunjukkan bahwa identifikasi bagian atau slot pallet merupakan komponen penting untuk menentukan posisi pallet secara akurat sebelum proses pengambilan oleh AGV forklift [20], [29].

Tahap pertama dalam penentuan alignment adalah memastikan bahwa data geometris pallet memenuhi kriteria validitas. Validitas dasar mencakup keberhasilan mendeteksi hole pallet kiri dan kanan, kesesuaian pasangan kedua hole, keberadaan block, serta kesesuaian posisi block terhadap geometri yang diprediksi. Seluruh kondisi tersebut digabungkan dalam Persamaan (2.24).

$$C_{geom} = C_{HL} \wedge C_{HR} \wedge C_{pair} \wedge C_{block} \wedge C_{block_{geom}} \quad (2.24)$$

dengan:

- Cgeom = kondisi validitas geometri pallet secara keseluruhan.
- CHL = kondisi hole pallet kiri valid.
- CHR = kondisi hole pallet kanan valid.
- Cpair = kondisi pasangan kedua hole valid.
- Cblock = kondisi block berhasil dideteksi.
- Cblock_geom = kondisi geometri block sesuai dengan referensi.
- ∧ = operator logika AND.
- **Fungsi persamaan:** memastikan seluruh fitur geometris yang diperlukan tersedia sebelum data digunakan untuk alignment.

Nilai C_{geom}=1 menunjukkan bahwa informasi visual yang diperlukan untuk menghitung posisi dan orientasi pallet telah tersedia. Sebaliknya, apabila salah satu kondisi tersebut tidak terpenuhi, data tidak digunakan untuk menyatakan pallet telah berada pada kondisi *alignment*. Pendekatan ini diperlukan karena keberhasilan deteksi objek secara individual belum menjamin bahwa kombinasi fitur tersebut merepresentasikan geometri pallet yang benar.

Selain validitas geometri, sistem mengevaluasi error lateral dan error yaw terhadap batas toleransi. Nilai toleransi merupakan parameter implementasi yang dijelaskan pada Bab III. Kedua syarat tidak dituliskan sebagai persamaan terpisah karena sudah menjadi bagian langsung dari kondisi alignment akhir pada Persamaan (2.27).

Penggunaan toleransi posisi dan orientasi secara bersamaan penting karena kondisi pallet yang telah terpusat secara horizontal belum tentu menunjukkan orientasi yang benar. Demikian pula, orientasi pallet dapat mendekati nol tetapi posisi centerline pallet masih bergeser terhadap referensi fork. Penelitian tentang *pallet positioning* menunjukkan bahwa keberhasilan proses pengambilan membutuhkan informasi posisi pallet atau slot yang akurat pada berbagai jarak dan sudut pengamatan [20]. Oleh karena itu, kedua komponen error perlu memenuhi batas toleransi secara simultan.

Hasil estimasi visual pada satu frame juga belum cukup untuk menyatakan *alignment* karena deteksi objek dapat mengalami fluktuasi akibat pencahayaan, perubahan sudut pandang, variasi *bounding box*, atau ketidakstabilan hasil inferensi. Dalam sistem *visual servoing*, kestabilan fitur visual dan perilaku konvergensi merupakan aspek penting dalam menentukan bahwa konfigurasi target benar-benar telah dicapai [26]. Untuk mengurangi kemungkinan sistem menyatakan *aligned* hanya akibat satu frame yang kebetulan memenuhi toleransi, digunakan mekanisme *stable frame*.

Kestabilan temporal ditentukan dari jumlah frame berturut-turut yang memenuhi validitas geometri dan toleransi error. Jika seluruh kondisi terpenuhi, stable-frame counter bertambah satu; jika tidak, penghitung kembali ke nol. Mekanisme tersebut dirumuskan pada Persamaan (2.25).

$$N_{stable}(k) = \begin{cases} N_{stable}(k-1) + 1, & C_{geom} = 1 \wedge C_{tol} = 1 \\ 0, & \text{lainnya} \end{cases} \tag{2.25}$$

dengan:

- Nstable[k] = jumlah frame stabil berturut-turut pada frame ke-k.
- Nstable[k-1] = jumlah frame stabil pada frame sebelumnya.
- Cgeom = kondisi validitas geometri.
- Ctol = kondisi error berada di dalam batas toleransi.
- **Fungsi persamaan:** menambah penghitung ketika geometri dan toleransi terpenuhi, serta mengembalikannya ke nol ketika syarat tidak terpenuhi.

Data dinyatakan stabil ketika jumlah frame valid telah mencapai jumlah minimum yang dipersyaratkan, sebagaimana dinyatakan pada Persamaan (2.26).

$$C_{stable} = \left(N_{stable} \geq N_{req}\right) \tag{2.26}$$

dengan:

- Cstable = kondisi kestabilan temporal.
- Nstable = jumlah frame stabil berturut-turut.
- Nreq = jumlah minimum frame stabil yang dipersyaratkan.
- **Fungsi persamaan:** menentukan bahwa data telah stabil setelah jumlah frame valid mencapai batas minimum.

Apabila error lateral dikonversi ke satuan meter, faktor kalibrasi hanya digunakan ketika kalibrasi telah divalidasi dan target berada pada daerah kerja kalibrasi. Kondisi validitas tersebut direpresentasikan oleh C_metric pada Persamaan (2.27), sehingga relasi logikanya tidak perlu ditulis sebagai persamaan tambahan.

Setelah validitas geometri, kestabilan temporal, validitas kalibrasi metrik, dan toleransi kedua error terpenuhi, kondisi alignment akhir ditentukan menggunakan Persamaan (2.27).

$$C_{align} = C_{geom} \wedge C_{stable} \wedge C_{metric} \wedge \left(\left|e_{lat,m}\right| \leq T_{lat,m}\right) \wedge \left(\left|e_{yaw}\right| \leq T_{yaw}\right) \quad (2.27)$$

dengan:

- Calign = kondisi alignment akhir.
- Cgeom = validitas geometri pallet.
- Cstable = kondisi kestabilan temporal.
- Cmetric = kondisi validitas kalibrasi metrik.
- elat,m = error lateral dalam satuan meter.
- Tlat,m = batas toleransi error lateral metrik.
- eyaw = error yaw.
- Tyaw = batas toleransi error yaw.
- ∧ = operator logika AND.
- **Fungsi persamaan:** menetapkan status alignment hanya ketika validitas geometri, kestabilan, kalibrasi, dan toleransi error terpenuhi.

Persamaan tersebut menunjukkan bahwa status aligned hanya diberikan apabila data geometri pallet valid, hasil pengukuran telah stabil selama jumlah frame yang ditentukan, error lateral metrik valid dan berada dalam toleransi, serta error yaw berada dalam toleransi. Konsep ini sesuai dengan kebutuhan pallet pose estimation pada sistem warehouse automation, di mana posisi dan orientasi pallet perlu diestimasi secara andal sebelum robot melakukan operasi manipulasi atau pengambilan pallet [29].

Dalam implementasi ROS 2, kondisi alignment akhir digunakan untuk membentuk status proses UNKNOWN, APPROACH, FINE_ALIGN, atau ALIGNED. Perpindahan status tersebut merupakan logika program, bukan persamaan matematis, sehingga dituliskan dalam narasi atau diagram alir.

2.14 Robot Operating System 2 (ROS 2)

Robot Operating System 2 (ROS 2) merupakan *framework* perangkat lunak untuk pengembangan sistem robotika yang mendukung arsitektur modular dan komunikasi terdistribusi. Sistem dalam ROS 2 dibentuk oleh sejumlah *node* yang memiliki fungsi tertentu dan saling bertukar data melalui antarmuka komunikasi. Arsitektur tersebut memungkinkan proses akuisisi sensor, pengolahan citra, deteksi objek, estimasi posisi relatif, dan fungsi robot lainnya dikembangkan sebagai modul yang terpisah namun tetap terintegrasi [30]. ROS 2 dirancang untuk mendukung kebutuhan sistem robot modern yang memerlukan arsitektur modular, skalabel, dan dapat digunakan pada berbagai jenis perangkat komputasi [30].

Komunikasi data yang digunakan pada sistem perception penelitian ini terutama menggunakan mekanisme *topic* dengan pola *publisher-subscriber*. Pada mekanisme tersebut, *publisher* mengirimkan *message* ke suatu *topic*, sedangkan satu atau lebih *subscriber* dapat menerima data yang diterbitkan pada *topic* tersebut. Antarmuka *topic* sesuai untuk data yang diperbarui secara terus-menerus, seperti citra kamera, hasil deteksi objek, maupun status sistem robot [31].

Pada implementasi perception AGV forklift, fungsi sistem dibagi menjadi tiga *node* utama, yaitu astra_rgb_v4l2_node, obstacle_detector_node, dan hole_block_alignment_node. Pembagian tersebut membuat proses akuisisi citra, deteksi objek, dan estimasi alignment dapat dijalankan serta diuji secara terpisah.

Node astra_rgb_v4l2_node berfungsi sebagai antarmuka kamera Orbbec Astra Pro. Node tersebut mengambil citra RGB dari kamera dan menerbitkannya melalui *topic*

```
/camera/color/image_raw.
```

Citra RGB tersebut menjadi masukan bagi obstacle_detector_node. Pada implementasi yang digunakan, data depth tidak menjadi bagian dari aliran perception aktif sehingga komunikasi kamera menuju proses deteksi didasarkan pada citra RGB.

Node obstacle_detector_node bertugas menjalankan inferensi YOLOv8 terhadap citra yang diterima dari kamera. Hasil deteksi berupa kelas objek, nilai *confidence*, koordinat *bounding box*, dan posisi pusat objek kemudian dibentuk menjadi *message* hasil deteksi dan diterbitkan melalui *topic*

```
/obstacle_detection/obstacles.
```

Selain data numerik hasil deteksi, node tersebut juga menyediakan citra hasil visualisasi melalui

```
/obstacle_detection/visualization.
```

Pemisahan antara data hasil deteksi dan data visualisasi memungkinkan informasi numerik digunakan oleh proses berikutnya tanpa bergantung pada tampilan antarmuka.

Hasil deteksi dari YOLOv8 selanjutnya digunakan oleh hole_block_alignment_node. Node ini menerima dua jenis masukan, yaitu citra RGB dari

```
/camera/color/image_raw
```

dan informasi objek dari

```
/obstacle_detection/obstacles.
```

Data tersebut digunakan untuk melakukan validasi pasangan hole pallet, menentukan midpoint pasangan hole, menghitung orientasi pallet, menentukan *expected block center*, melakukan validasi geometri block, serta menghitung error lateral dan error yaw. Setelah melalui proses filtering dan validasi kestabilan, node menerbitkan informasi hasil alignment melalui

```
/fork_alignment/state.
```

Node alignment juga menghasilkan citra visualisasi melalui

```
/fork_alignment/image.
```

Dengan demikian, aliran data perception pada sistem dapat direpresentasikan sebagai

2.14.1 Message Hasil Deteksi dan Alignment

ROS 2 menggunakan *message type* untuk menentukan struktur data yang dikirimkan melalui *topic*. Selain menggunakan tipe pesan yang telah tersedia, ROS 2 memungkinkan pengembang membuat *custom message* apabila data sistem membutuhkan struktur khusus [30]. Pada penelitian ini, informasi hasil alignment disusun dalam AlignmentState.msg sehingga data yang dikirim tidak hanya berupa satu nilai error, tetapi mencakup sejumlah informasi kondisi perception. ROS 2 memang mendukung pembuatan antarmuka kustom .msg untuk pertukaran data antar-node sesuai kebutuhan aplikasi.

Informasi utama yang diterbitkan pada /fork_alignment/state meliputi kondisi deteksi pallet, kestabilan deteksi, nilai *confidence*, error lateral, error yaw, validitas data, serta status alignment. Selain itu, message menyediakan indikator apakah error lateral dan yaw telah berada dalam toleransi serta kondisi ready_for_insertion.

Secara konseptual, status alignment dibagi menjadi beberapa keadaan, yaitu:

- UNKNOWN, ketika informasi visual yang diperlukan belum tersedia;
- APPROACH, ketika pallet atau sebagian fitur pallet telah terdeteksi tetapi data belum memenuhi seluruh kriteria alignment;
- FINE_ALIGN, ketika data geometris telah valid dan proses penyelarasan berlangsung;
- ALIGNED, ketika seluruh kriteria alignment telah terpenuhi.

Perubahan status tersebut dapat dinyatakan sebagai

$$\text{UNKNOWN} \rightarrow \text{APPROACH} \rightarrow \text{FINE ALIGN} \rightarrow \text{ALIGNED}.$$

Status ALIGNED kemudian digunakan untuk membentuk kondisi ready_for_insertion, yaitu kondisi bahwa hasil perception telah memenuhi kriteria penyelarasan yang telah ditetapkan.

2.14.2 Hubungan Publisher dan Subscriber

Hubunagan antar-node pada sistem perception dapat dirangkum sebagai berikut: Rincian data dan parameter terkait disajikan pada Tabel 2.7.

Tabel 2.7 Hubungan Publisher dan Subscriber pada Sistem Perception

Node	Fungsi	Subscriber	Publisher
astra_rgb_v4l2_node	Akuisisi citra RGB		/camera/color/image_raw
obstacle_detector_node	Deteksi objek YOLOv8	/camera/color/image_raw	/obstacle_detection/obstacles, /obstacle_detection/visualization
hole_block_alignment_node	Estimasi alignment pallet	/camera/color/image_raw, /obstacle_detection/obstacles	/fork_alignment/state, /fork_alignment/image

Sumber: Diolah penulis berdasarkan dokumentasi ROS 2 [30], [31] dan implementasi sistem.

Struktur tersebut membuat data citra hanya perlu diterbitkan sekali oleh node kamera dan selanjutnya dapat digunakan oleh beberapa node sesuai kebutuhan. Pola *publish-subscribe* bersifat asinkron dan sesuai untuk aliran data sensor yang diperbarui secara kontinu [31].

Dalam penelitian ini, ROS 2 tidak secara langsung melakukan proses deteksi maupun perhitungan alignment. ROS 2 berfungsi sebagai middleware yang mengatur pertukaran informasi antara node kamera, YOLOv8, dan node estimasi alignment. Dengan pemisahan tersebut, perubahan pada algoritma deteksi tidak harus mengubah node kamera, dan perubahan pada metode alignment tidak harus mengubah proses inferensi YOLOv8. Arsitektur modular semacam ini merupakan salah satu alasan ROS 2 banyak digunakan pada sistem robot modern yang membutuhkan integrasi sensing, perception, planning, dan autonomy [30].