BAB 1

PENDAHULUAN

1.1 Latar belakang

Pendeteksian objek sudah sangat sering kita temukan dalam kehidupan sehari-hari yang mempermudah setiap kegiatan dalam proses pengenalan objek, misalnya terdapat pada bidang militer, transportasi cerdas, deteksi wajah, bidang robotika, dan lainnya. Deteksi target adalah salah satu hotspot penelitian di bidang visi komputer. Lokasi dan kategori target dapat ditentukan dengan menggunakan deteksi target. Saat ini, deteksi target telah diterapkan di banyak bidang salah satunya termasuk segmentasi citra (Ju et al., 2019).

Kompetisi Robot Indonesia (KRI) merupakan kontes yang berfokus pada perancangan dan pembuatan robot di bidang robotika yang diadakan setiap tahun secara nasional. Acara ini diselenggarakan oleh Direktorat Jenderal Pembelajaran dan Kemahasiswaan di bawah Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Salah satu kategori dalam Kompetisi Robotika Indonesia adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI-B).

KRSBI-B adalah Robot sepak bola beroda yang dapat melakukan gerakan seperti menggiring bola, mengoper bola, dan mencetak gol. Robot sepak bola beroda sendiri memiliki komponen-komponen utama untuk melakukan hal tersebut, seperti motor dan roda sebagai penggerak, mikrokontroler sebagai otak untuk memberikan perintah, dan kamera sebagai alat untuk melihat. Salah satu komponen utama sebagai pendeteksi objek adalah kamera *omnidirectional* yang

digunakan dalam proses pendeteksian bola, gawang, dan robot atau *obstacle* yang diterapkan pada robot KRSBI-B. Dengan menggunakan kamera *omnidirectional*, pada saat proses pendeteksian objek sering sekali mengalami *error* dan FPS *drop* dalam melakukan pendeteksian objek. Hal tersebut menyebabkan seringnya terjadi bergonta-gantinya penggunaan algoritma dalam mendeteksi objek yang diterapkan pada robot KRSBI-B.

Pada penggunaan YOLO V7 - RAR untuk mendeteksi objek bergerak seperti sebuah kendaraan dengan menggunakan CUDA v11.2, Pytorch v1.10, NVIDIA GeForce RTX3090, *memory* 25,4 Gb, dan Python versi 3.9 dengan mengguanakan *backbone* DarkNet. Akurasi deteksi rata-rata dari algoritma YOLO V7 – RAR mencapai 95,1%; 2,4% (96 FPS) lebih tinggi dari pada algoritma asli (Zhang et al., 2023).

Pada penelitian terdahulu dilakukan pendeteksian sebuah objek dengan menggunakan YOLO V1 dengan hasil tingkat akurasi 69,70% (60fps) untuk deteksi objek dan klasifikasi. YOLO v2 tingkat presisi rata-rata 81% untuk deteksi objek bergerak. YOLO v3 tingkat presisi diatas 90% pendeteksian dan perhitungan pada jaringan distribusi. YOLO v4 tingkat 97,31% untuk pendeteksian objek hidup (Diwan et al., 2023).

Pengujian kecepatan running YOLO dengan menggunakan Darknet-53. Kecepatan running meningkat secara signifikan (sekitar 442% lebih cepat dari varian YOLO sebelumnya) dalam mendeteksi objek yang di tangkap, tetapi akurasi deteksi berkurang. Dengan menggunakan *pooling layer* dan mengurangi angka

untuk *convolution layer*. Ini memprediksi tensor tiga dimensi yang berisi skor objektivitas, kotak pembatas, dan prediksi kelas pada dua skala yang berbeda (Adarsh et al., 2020).

Sambungan pada di Tinier – YOLO memberikan solusi untuk membantu meningkatkan akurasi deteksi dan kinerja *realtime* karena propagasi fitur diperkuat dan aliran informasi maksimum dalam jaringan dipastikan. Ukuran model YOLO-Tiny 2x lebih kecil dari SqueezeNet SSD dan MobileNet SSD. Nilai BFLOP/s YOLO-Tiny sekitar 2x lebih kecil dari dua model lainya, serta YOLO-Tiny hanya membutuhkan waktu 135,2ms untuk mendeteksi gambar, sedangkan MobileNet SSD menggunakan 483,7ms dan SqueezeNet SSD menggunakan 595,7ms (Fang et al., 2020).

You only look once (YOLO) adalah algoritma yang dapat melakukan pendeteksian objek secara *realtime*, YOLO sendiri selalu mendapatkan perkembangan dan peningkatan dari versi-versi sebelumnya. YOLO V8 adalah jenis YOLO dari versi terbaru setelah YOLO V7 dimana YOLO V8 ini dapat memanfaatkan pendeteksian objek secara signifikan dengan kualitas FPS yang tinggi dan juga memiliki kelebihan dengan hanya sedikit *dataset* untuk belajar mengenali sebuah objek. YOLO V8 adalah implementasi baru dari *Deep Learning* yang menghubungkan *input* (citra asli) dengan *output*. Jenis algoritma YOLO V8 ini menggunakan arsitektur *A deep dive*, dibantu dengan CNN dan *backbone* yang baru dimana menggunakan *convolusional layer* untuk *pixelnya* yang bila digambarkan akan berbentuk seperti sebuah piramida.

Pada penelitian ini bertujuan untuk mendeteksi objek seperti bola, gawang, robot atau *obstacle* dengan menggunakan algoritma YOLO V8, dikarenakan pada algoritma YOLO V8 dapat dengan cepat belajar mengenali objek dengan sedikitnya *dataset* yang digunakan dalam proses *training*, dapat memberikan tingkat FPS yang tinggi dan dapat mengurangi jumlah *error* yang terdapat pada saat proses pendeteksian objek secara *realtime*.

1.2 Identifikasi masalah

Berdasarkan latar belakang dan penelitian sebelumnya, beberapa masalah yang dapat diidentifikasi, di antaranya:

1. *Frame Rate Persecond* (FPS) pada saat pendeteksian objek rendah.
2. Sistem ketepatan dalam pendeteksian masih terdapat *error* dalam mengenali objek yang akan di deteksi.
3. Jenis kamera untuk pendeteksian objek pada robot KRSBI-B dapat dengan menggunakan CamHD Venda Logitech dan Webcam Logitech HD C920, masih kurang optimal dikarenakan rendahnya resolusi tangkapan kamera pada saat pengambilan gambar.
4. Metode pendeteksian objek pada robot KRSBI-B menggunakan metode *Convolutional Neural Network* (CNN), *Hue Saturation Value* (HSV) dimana ketika suatu objek terdiri dari dua warna atau lebih akan mengakibatkan pendeteksian suatu objek menjadi tidak sempurna.

1.3 Batasan masalah

Pada Penelitian ini dibuat batasan masalah sebagai berikut :

1. Dengan menggunakan kamera *omnidirectional* membuat proses pengambilan gambar lebih susah karena tingkat objek makin terlihat lebih kecil dan jauh saat menggunakan lensa cembung sebagai *omnidirectional*-nya.
2. Menggunakan *You only look once* (YOLO) versi 8 untuk mendeteksi objek, dikarenakan YOLO V8 masih sangat baru, membuat terbatasnya informasi mengenai penggunaan dan kelebihan yang masih banyak belum diketahui.
3. Menggunakan bola berwarna *orange*, yang harus menyesuaikan kontras ruangan dan yang ada dikamera. Menggunakan lapangan yang terbuat dari karpet berwarna hijau.
4. Menggunakan lapangan yang terbuat dari karpet berwarna hijau, agar objek yang diatas lapangan dapat dikenali dengan mudah.
5. Menggunakan gawang berwarna putih, robot atau *obstacle* berwarna hitam.
6. Menggunakan 9 buah lampu LED karena memerlukan pencahayaan yang stabil, karena perubahan dan tingkat intensitas cahaya dapat mempengaruhi tampilan pada objek.
7. Dalam penelitian ini fitur YOLO terhadap tekstur tidak digunakan karena pada penelitian ini bentuk dan warna pada sebuah objek sudah cukup untuk menjadi fitur yang membantu dalam pendeteksian objek.

1.4 Rumusan masalah

1. Bagaimana cara mengurangi FPS *drop* pada saat deteksi objek menggunakan metode YOLO Versi8?

2. Bagaimana menghindari *eror* deteksi objek melalui metode YOLO Versi 8 pada saat ada benda lain yang membuat *noise* pendeteksian?

1.5 Tujuan penelitian

Berdasarkan latar belakang yang ada tujuan dari penelitian ini adalah:

1. Mengurangi tingkat *error* pada saat melakukan pendeteksian objek yang sudah di *training* dan dipelajari oleh algoritma YOLO V8.
2. Meningkatkan FPS pada deteksi objek menggunakan YOLO V8, dimana pada versi sebelumnya deteksi objek memiliki tingkat FPS hanya sampai 30%.
3. Dapat mendeteksi objek selain bola, seperti objek lain yang ada disekitar saat proses perlombaan, contohnya gawang, *obstacle*, atau robot lain yang ada disekitarnya..

1.6 Manfaat penelitian

Penelitian ini diharapkan dapat memberikan maanfaat sebagai berikut:

1. Pendeteksian Objek tertentu selain bola dapat lebih optimal dengan menggunakan metode *you only look once* (YOLO).
2. Peforma dan kemampuan akurasi dalam mendeteksi objek pada kamera *omnidirectional* meningkat.
3. *You only look once* Versi 8 menjadi metode algoritma pendeteksian yang dapat diterapkan dimasa mendatang.