



## PROPOSAL

# OPTIMASI GERAKAN ROBOT SEPAK BOLA MENGGUNAKAN *KALMAN FILTER* PADA DETEKSI OBJEK BERBASIS YOLO

Oleh:

**FIKRI RIVANDI**  
2207112583

PROGRAM STUDI TEKNIK INFORMATIKA S1  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS RIAU  
**2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iv</b>
<b>DAFTAR TABEL .....</b>	<b>v</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1.    Latar Belakang .....	1
1.2.    Rumusan Masalah.....	5
1.3.    Tujuan Penelitian .....	6
1.4.    Batasan Masalah.....	6
1.5.    Manfaat Penelitian .....	6
1.6.    Sistematika Penulisan .....	7
<b>BAB II LANDASAN TEORI .....</b>	<b>9</b>
2.1.    Penelitian Terdahulu .....	9
2.2.    Kompetisi Robot Sepak Bola Beroda Indonesia.....	24
2.3.    Robot Sepak Bola Beroda .....	24
2.4.    OpenCV v3.4 .....	26
2.5.    NumPy v2.3 .....	26
2.6. <i>Kalman Filter</i> .....	27
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>33</b>
3.1.    Kerangka Pikiran.....	33
3.2.    Studi Literatur .....	33
3.2.1 Algoritma Deteksi Objek (YOLO) .....	33
3.2.2 Algoritma Prediksi dan Pelacakan ( <i>Kalman Filter</i> ).....	33
3.2.3 Sistem Robotika dan Kontrol Gerak .....	34

3.3.	Identifikasi Masalah .....	34
3.4.	Persiapan Data dan Konfigurasi Model Deteksi YOLO .....	35
3.5.	Implementasi Sistem Deteksi dan Pelacakan .....	36
3.6.	Desain dan Implementasi Kontrol Gerak Robot Berbasis Penglihatan	38
<b>DAFTAR PUSTAKA .....</b>		<b>40</b>

## **DAFTAR GAMBAR**

Gambar 2.1. Robot sepak bola beroda ..... 25

## **DAFTAR TABEL**

Tabel 2.1. Penelitian terdahulu.....	17
--------------------------------------	----

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi di era modern telah memberikan dampak signifikan pada berbagai aspek kehidupan manusia. Salah satu bidang yang mengalami kemajuan pesat adalah teknologi *Artificial Intelligence* (AI) dan *machine learning* (ML) (Widodo et al., 2024). Teknologi ini memungkinkan sistem komputer tidak hanya melakukan perhitungan, tetapi juga belajar dari data, mengenali pola, serta mengambil keputusan secara mandiri (Wijaya & Yuniarto, 2024).

Salah satu cabang dari AI yang banyak dimanfaatkan adalah *computer vision*. *Computer vision* merupakan cabang dari kecerdasan buatan yang berfokus pada kemampuan komputer dan sistem untuk mengekstraksi informasi bermakna dari citra digital, video, maupun data visual lainnya. Informasi yang diperoleh kemudian dapat digunakan untuk mengambil keputusan, memberikan rekomendasi, atau menjalankan suatu tindakan tertentu. Jika kecerdasan buatan secara umum memungkinkan komputer untuk berpikir, maka *computer vision* memberikan kemampuan bagi komputer untuk melihat, mengenali, serta memahami lingkungan visual di sekitarnya (Baskoro et al., 2022).

Robotika merupakan bidang interdisipliner yang berfokus pada kajian mengenai robot, di mana ilmu pengetahuan, teknologi, dan rekayasa saling berhubungan. Hasil akhirnya berupa mesin yang dapat menirukan perilaku manusia atau menjalankan instruksi tertentu melalui pemrograman. Keberhasilan dalam bidang robotika memerlukan pemahaman dasar mengenai teknik untuk merancang dan membangun robot secara fisik, pengetahuan rekayasa untuk memahami cara kerja setiap komponennya, serta kemampuan pemrograman untuk mengatur fungsi dan perilaku robot tersebut (Hendrik & Awal, 2023).

Robot konvensional umumnya didesain untuk beroperasi pada lingkungan yang terstruktur dan terbatas dengan algoritma tugas yang telah diprogram secara statis. Akan tetapi, kenyataannya lingkungan kerja sering bersifat dinamis dan tidak terduga, sehingga dibutuhkan sistem yang lebih adaptif dan fleksibel. Pada titik

inilah kecerdasan buatan memiliki peran penting. Dengan memanfaatkan teknologi seperti *machine learning*, *computer vision*, dan kecerdasan buatan lainnya, robot modern mampu mengenali objek, memahami kondisi lingkungan, mengambil keputusan, serta belajar dari pengalaman secara mandiri (Ritonga & Hasibuan, 2025). Salah satu implementasi nyata robotika yang mendapat perhatian khusus adalah robot sepak bola, khususnya dalam Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda.

Universitas Riau memiliki klub robotika bernama *Electrical Robotic Club* (ERC) yang berfokus pada pengembangan robot KRSBI Beroda. Pada tahun 2021 dan 2022, tim ERC UNRI berhasil mencapai tingkat nasional, yang menjadi pencapaian signifikan dalam sejarah keikutsertaannya di KRSBI Beroda. Sejak berpartisipasi pertama kali pada tahun 2019, ERC UNRI telah mengalami berbagai perkembangan signifikan, baik dalam aspek perangkat keras maupun perangkat lunak. Namun, perubahan regulasi sejak tahun 2023 membuat tim harus melakukan penyesuaian ulang agar robot tetap kompetitif dan relevan.

Dalam pertandingan KRSBI Beroda, kemampuan deteksi bola dan gawang menjadi faktor yang sangat krusial. Awalnya, metode deteksi objek berbasis HSV (*Hue*, *Saturation*, *Value*) digunakan karena ringan dan dapat berjalan *real-time*. HSV adalah model warna yang merepresentasikan persepsi warna sebagaimana ditangkap oleh mata manusia. Citra dengan format HSV dimanfaatkan dalam penerapan algoritma *image thresholding* guna mendeteksi warna objek. Selanjutnya, hasil dari proses *thresholding* tersebut digunakan dalam algoritma *Convex Hull*, di mana apabila bola berhasil terdeteksi, maka sistem akan menghitung ukuran objek, menentukan titik pusatnya, serta memperoleh koordinat posisi objek (x, y) relatif terhadap posisi robot saat ini (Nanda et al., 2023).

Kelemahan metode HSV adalah ketergantungannya pada kondisi pencahayaan, sehingga akurasi deteksi menjadi tidak stabil dalam situasi lapangan yang dinamis (Nanda et al., 2023). Hal ini diperkuat oleh temuan yang menunjukkan bahwa intensitas cahaya berpengaruh signifikan terhadap akurasi deteksi objek. Misalnya, deteksi objek robot magenta mencapai akurasi di atas 75% pada intensitas 9–19 Lux dan sekitar 33 Lux, sementara robot *cyan* mampu

mencapai 80% akurasi pada rentang 9–24 Lux. Deteksi bola menunjukkan akurasi di atas 70% pada intensitas 14–33 Lux, sedangkan gawang memiliki akurasi lebih dari 70% pada kisaran 24–33 Lux, meskipun akurasinya menurun apabila terdapat objek lain di depannya. Dengan demikian, dapat disimpulkan bahwa penggunaan metode berbasis HSV sangat dipengaruhi oleh kondisi pencahayaan, yang membatasi konsistensi performa deteksi objek di lapangan.

Seiring perkembangan teknologi, metode *deep learning*, khususnya YOLO (*You Only Look Once*), mulai diterapkan untuk meningkatkan akurasi deteksi objek pada robot KRSBI Beroda milik ERC UNRI. YOLO menggunakan *single neural network* yang dapat mendeteksi beberapa objek yang terdapat pada suatu gambar secara *real-time*, YOLO secara langsung membagi gambar ke dalam grid dan memprediksi lokasi serta kelas objek dalam satu proses komputasi (F. B. Saputra et al., 2023). Hal ini membuat YOLO mampu mendeteksi objek secara *real-time* dengan akurasi tinggi.

Namun, meskipun akurat dan cepat dalam teori, penerapan YOLO pada perangkat dengan keterbatasan sumber daya, seperti robot KRSBI Beroda, menghadapi tantangan tersendiri. Model YOLO, seperti YOLOv8, membutuhkan kapasitas komputasi yang cukup tinggi untuk dapat berjalan optimal. Hal ini mencakup kebutuhan prosesor yang kuat, memori yang besar, dan sering kali dukungan GPU agar proses inferensi bisa berjalan lancar. Keterbatasan perangkat keras pada robot menyebabkan YOLO terkadang tidak dapat bekerja pada *frame rate* yang stabil. Penurunan performa ini berdampak pada keterlambatan informasi posisi bola yang diterima robot. Dengan kata lain, meskipun YOLO sangat efektif dalam menghasilkan deteksi yang akurat, efisiensinya menjadi kurang optimal pada perangkat dengan daya komputasi terbatas, sehingga menimbulkan masalah baru berupa pergerakan robot yang terlambat atau tidak stabil saat mengikuti posisi bola.

Beberapa penelitian sebelumnya membuktikan bahwa metode YOLO adalah salah satu cara efektif dalam *object detection*, tapi tidak cukup efisien untuk sumber daya perangkat keras yang digunakan, karena membutuhkan kapasitas komputasi yang mumpuni (Miharja et al., 2025). Pada penelitian lainnya juga menyebutkan bahwa menggunakan perangkat *Single Board Computer* berkinerja tinggi, seperti

NVIDIA Jetson Series, juga perlu dipertimbangkan untuk meningkatkan kecepatan deteksi (FPS) guna mencapai performa *real-time* yang lebih optimal (Firdaus & Lelono, 2025).

Selain itu, pada proses deteksi objek juga sering terjadi permasalahan seperti *occlusion* (oklusi) dan *false negative detection*. Fenomena oklusi, di mana sebagian objek tertutup oleh penghalang tertentu, merupakan masalah fundamental dalam deteksi objek. Oklusi merupakan penyebab utama yang meningkatkan *false-negative detection rate*, yang pada dapat menurunkan kinerja deteksi secara keseluruhan. Objek yang teroklusi ini dikategorikan sebagai *hard-positive examples* yang sulit dideteksi oleh model. Oleh karena itu, oklusi dan *false-negative* adalah hal yang harus diperhatikan untuk mencapai optimalitas dalam penggunaan *object detection* (Ryu & Chung, 2021).

Untuk menjawab tantangan tersebut, penelitian ini mengusulkan penggunaan *Kalman Filter* pada hasil deteksi bola berbasis YOLO. *Kalman Filter* merupakan salah satu algoritma estimasi yang banyak digunakan dalam sistem dinamis karena mampu memprediksi keadaan berikutnya dari suatu objek berdasarkan data pengamatan sebelumnya. Metode *Kalman Filter* menggunakan informasi dari objek yang terdeteksi di suatu *frame* dan status objek dari *frame* sebelumnya untuk mendapatkan status yang baru dari objek tersebut (C. Saputra, 2023).

Pada konteks robot sepak bola, *Kalman Filter* sangat relevan untuk digunakan karena bola sering mengalami perubahan posisi secara cepat, mendadak, dan terkadang tidak terduga. YOLO sebagai detektor objek hanya memberikan posisi bola pada setiap *frame*, tanpa mempertimbangkan dinamika gerakan dari bola tersebut. Akibatnya, jika bola bergerak terlalu cepat atau terjadi keterlambatan proses inferensi, robot bisa kehilangan akurasi dalam mengejar bola. Dengan *Kalman Filter*, sistem tidak hanya bergantung pada deteksi saat ini, tetapi juga dapat memprediksi posisi bola pada *frame* berikutnya, sehingga pergerakan robot menjadi lebih stabil, responsif, dan efisien.

Selain itu, *Kalman Filter* memiliki keunggulan dari sisi efisiensi komputasi. Dibandingkan dengan metode prediksi berbasis *deep learning* yang umumnya membutuhkan sumber daya tinggi, *Kalman Filter* relatif ringan dan dapat

diimplementasikan pada perangkat dengan keterbatasan komputasi seperti robot KRSBI Beroda. Penelitian sebelumnya yang dilakukan oleh Yuztiawan & Utaminingrum (2017) juga memperkuat hal ini, di mana integrasi model YOLOv8N dengan *Kalman Filter* pada kondisi lingkungan dengan banyak objek mampu meningkatkan akurasi dan stabilitas deteksi serta pelacakan hingga 91,66%. Bahkan, penggunaan *Kalman Filter* terbukti meningkatkan kinerja pelacakan sebesar 25% dibandingkan hanya menggunakan YOLOv8N saja. Menariknya, penambahan algoritma tersebut hanya memberikan tambahan waktu komputasi rata-rata sekitar 0,0076 detik per *frame* (sekitar 7,92%), sehingga sistem tetap mampu bekerja secara *real-time*. Dengan demikian, *Kalman Filter* tidak hanya meningkatkan akurasi pelacakan, tetapi juga tetap mempertahankan efisiensi komputasi yang sangat penting bagi robot kompetitif dengan keterbatasan perangkat keras.

Berdasarkan permasalahan yang telah diuraikan, peneliti tertarik untuk meneliti dengan judul “Optimasi Gerakan Robot Sepak Bola Menggunakan *Kalman Filter* Pada Deteksi Bola Berbasis YOLO”. Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem deteksi serta pelacakan bola pada robot KRSBI Beroda dengan memanfaatkan kamera *omnidirectional* dan pendekatan *deep learning* berbasis CNN. Deteksi objek akan dilakukan menggunakan algoritma YOLO, sementara prediksi pergerakan bola diperkuat dengan penerapan *Kalman Filter* untuk mengatasi keterbatasan deteksi berbasis *frame* tunggal. Dengan kombinasi ini, robot diharapkan tidak hanya mampu mengenali bola dan gawang secara akurat, tetapi juga dapat melakukan respon gerakan yang lebih mulus, stabil, dan efisien meskipun bola bergerak cepat atau kondisi lapangan penuh gangguan visual.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan dalam penelitian ini dapat dirumuskan ke dalam beberapa pertanyaan sebagai berikut:

1. Bagaimana penerapan *Kalman Filter* dapat meningkatkan akurasi prediksi posisi bola yang terdeteksi oleh YOLO?

2. Bagaimana integrasi YOLO dan *Kalman Filter* dapat mengoptimalkan gerakan robot KRSBI Beroda dalam mengikuti pergerakan bola di lapangan?
3. Sejauh mana kombinasi metode tersebut dapat meningkatkan stabilitas, efisiensi, dan kecepatan robot dalam pertandingan KRSBI Beroda?

### **1.3. Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah dikemukakan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan *Kalman Filter* pada hasil deteksi bola berbasis YOLO untuk meningkatkan akurasi prediksi posisi bola pada robot KRSBI Beroda.
2. Mengintegrasikan YOLO dengan *Kalman Filter* guna mengoptimalkan gerakan robot terhadap pergerakan bola di lapangan.
3. Mengevaluasi kinerja kombinasi YOLO dan *Kalman Filter* dalam meningkatkan stabilitas, efisiensi, dan kecepatan pergerakan robot KRSBI Beroda.

### **1.4. Batasan Masalah**

Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya membahas deteksi dan pelacakan bola menggunakan kamera *omnidirectional* pada robot KRSBI Beroda.
2. Algoritma deteksi objek yang digunakan adalah YOLO, tanpa melakukan pengembangan arsitektur baru, melainkan memanfaatkan model yang sudah ada.
3. Prediksi posisi bola dilakukan dengan menggunakan *Kalman Filter*, tanpa membandingkannya secara langsung dengan algoritma prediksi lain.
4. Fokus penelitian adalah pada optimasi gerakan robot terhadap bola, tidak mencakup aspek strategi tim, komunikasi antar robot, atau kontrol *hardware* secara mendetail.

### **1.5. Manfaat Penelitian**

Berikut adalah manfaat dilakukannya penelitian ini bagi beberapa pihak:

1. Peneliti akan memperoleh pengalaman berharga dalam mengimplementasikan algoritma YOLO yang dipadukan dengan *Kalman Filter* pada sistem robotika, khususnya dalam konteks deteksi dan prediksi objek bergerak.
2. Penelitian ini dapat membantu meningkatkan performa robot dalam pertandingan melalui sistem deteksi bola yang lebih akurat dan gerakan robot yang lebih stabil serta efisien.

## **1.6. Sistematika Penulisan**

Untuk mempermudah dalam memahami lebih jelas tentang penulisan penelitian ini, maka penelitian ini ditulis dalam beberapa bab yang masing-masing berkaitan satu sama lainnya, dengan sistematika penulisan sebagai berikut:

### **BAB I PENDAHULUAN**

Bagian ini berisi tentang deskripsi umum dari penelitian yang akan dilakukan meliputi Latar Belakang, Perumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat Penelitian dan Sistematika Penulisan.

### **BAB II LANDASAN TEORI**

Bagian ini membahas penelitian terdahulu, teori-teori dan pendapat para ahli yang berhubungan dengan penelitian yang dilakukan.

### **BAB III METODOLOGI PENELITIAN**

Bagian ini berisi tentang alat dan bahan penelitian yang dilakukan, metode dan alur penelitian, metode pengembangan sistem cerdas, metode pengumpulan data, teknik mengolah data, dan teknik menguji hasil olahan data.

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini menjelaskan tentang hasil perancangan dan analisa yang telah dilakukan sesuai dengan metodologi penelitian, sekaligus mengevaluasi hasil pengujian terhadap parameter-parameter uji yang telah ditetapkan.

### **BAB V KESIMPULAN DAN SARAN**

Bab ini menjelaskan tentang simpulan hasil penelitian yang diperoleh sesuai dengan tujuan penelitian serta memuat saran mengenai masalah dan kemungkinan pemecahannya untuk penelitian selanjutnya.

**DAFTAR PUSTAKA**

**LAMPIRAN**

## BAB II

### LANDASAN TEORI

#### 2.1. Penelitian Terdahulu

Penelitian mengenai sistem deteksi objek, khususnya dalam penggunaan model *Convolutional Neural Network* (CNN) seperti *You Only Look Once* (YOLO), telah banyak dilakukan sebelumnya. Penelitian-penelitian tersebut memberikan kontribusi penting dalam pengembangan metode deteksi objek maupun optimasi model menggunakan algoritma bantuan seperti estimator. Dengan meninjau penelitian terdahulu, penulis memperoleh wawasan terkait kelemahan metode sebelumnya, potensi pengembangan, serta peluang penerapan algoritma terbaru yang lebih efektif.

Pertama, penelitian yang berjudul “*CNN-Based Ball and Goal Detection for KRSBI Robot with Omnidirectional Camera*” dilakukan oleh Farhan & Candra (2025). Penelitian ini berfokus pada pengembangan sistem deteksi bola dan gawang berbasis *computer vision* pada robot yang mengikuti Kompetisi Robot Sepak Bola Indonesia (KRSBI). Beroda menggunakan kamera *omnidirectional*. Metode tradisional seperti HSV *color filtering* sebelumnya banyak digunakan karena implementasinya sederhana, namun performanya sangat bergantung pada intensitas cahaya sehingga tidak stabil dalam kondisi pencahayaan yang bervariasi. Untuk mengatasi kelemahan tersebut, penelitian ini mengusulkan penerapan algoritma YOLO yang berbasis CNN guna meningkatkan akurasi dan keandalan deteksi objek secara *real-time*. Dataset yang digunakan terdiri dari 1.125 citra dengan variasi pencahayaan dan posisi objek yang berbeda, kemudian dibagi menjadi 80% data pelatihan dan 20% data validasi. Model YOLOv8 dilatih menggunakan *Ultralytics* di platform Google Colab selama 100 *epoch*. Hasil pelatihan menunjukkan performa deteksi yang sangat tinggi, dengan tingkat akurasi sebesar 95,87%, *precision* mencapai 1.00 pada *confidence threshold* 0.921, *recall* sebesar 0.99, dan nilai *F1-Score* maksimum 0.97 pada *confidence* 0.149. Berdasarkan kurva *precision-confidence*, model dapat menghasilkan deteksi tanpa *false positive* pada ambang kepercayaan tinggi, sedangkan kurva *recall-confidence* menunjukkan

kemampuan model mendeteksi hampir semua objek meskipun pada nilai *confidence* yang rendah. Hasil penelitian ini membuktikan bahwa model YOLOv8 berbasis CNN mampu memberikan solusi yang tangguh dan efisien untuk sistem deteksi bola dan gawang secara real-time pada robot sepak bola beroda. Selain itu, performa model menunjukkan tingkat stabilitas yang baik terhadap perubahan pencahayaan dan posisi objek, sehingga metode ini dinilai efektif untuk diterapkan pada sistem persepsi visual robot KRSBI Beroda.

Kedua, penelitian yang berjudul “Implementasi *Object Detection* pada Robot Sepak Bola Beroda Berbasis Kamera *Omnidirectional* Menggunakan Opencv” dilakukan oleh Nanda et al. (2023). Penelitian ini mengangkat permasalahan terkait sistem deteksi pada robot KRSBI yang masih sangat dipengaruhi oleh kondisi pencahayaan. Perubahan intensitas cahaya di lapangan membuat sistem kesulitan mempertahankan akurasi deteksi, sehingga kinerja robot menjadi tidak stabil ketika kondisi cahaya berubah. Metode yang digunakan dalam penelitian ini adalah *object detection* berbasis kamera *omnidirectional* dengan bantuan pustaka OpenCV. Proses pengujian dilakukan dengan cara mengukur variasi intensitas cahaya di lapangan untuk melihat pengaruhnya terhadap kemampuan deteksi objek. Hasil yang diperoleh menunjukkan bahwa robot KRSBI mampu mengenali beberapa objek penting, seperti bola, gawang, serta robot lawan (*cyan* dan *magenta*), dengan tingkat akurasi sekitar 70%. Akan tetapi, nilai akurasi ini masih cukup dipengaruhi oleh perbedaan intensitas cahaya dari masing-masing objek. Dengan demikian, penelitian ini membuktikan bahwa penerapan *object detection* menggunakan kamera *omnidirectional* dan OpenCV dapat berjalan pada robot KRSBI, namun tetap memiliki keterbatasan signifikan terutama ketika menghadapi kondisi pencahayaan yang tidak stabil.

Ketiga, penelitian yang berjudul “Perancangan Sistem Pendekripsi Obyek Bola dengan Metode *Framework YOLO V4*” dilakukan oleh Nuralim et al. (2022). Latar belakang penelitian ini berangkat dari kebutuhan dalam KRSBI Beroda tahun 2018, di mana robot dituntut mampu melakukan navigasi serta menjalankan tugas utama, yaitu menemukan bola, menggiringnya, dan menendangnya ke arah gawang lawan. Untuk dapat melaksanakan fungsi tersebut, robot membutuhkan sistem

pendeksi bola yang cepat, akurat, dan responsif. Metode yang digunakan dalam penelitian ini adalah *object detection* berbasis *framework* YOLOv4. Pengujian sistem dilakukan menggunakan *confusion matrix* untuk mengukur performa deteksi, termasuk dalam kondisi ketika bola sebagian terhalang oleh objek lain. Hasil pengujian menunjukkan bahwa sistem mampu mendeksi bola meskipun terdapat penghalang dengan tingkat persentase 50%, 60%, hingga 70%. Namun, kemampuan deteksi menurun drastis ketika tingkat penghalang mencapai 80% ke atas. Pada kondisi tersebut (80%, 90%, hingga 100% penghalang), robot tidak lagi dapat mengenali keberadaan bola. Penelitian ini membuktikan bahwa YOLOv4 cukup efektif dalam mendeksi bola pada kondisi yang dinamis dan kompleks, termasuk ketika sebagian objek tertutup. Akan tetapi, keterbatasan tetap muncul pada tingkat *occlusion* yang tinggi, yang menunjukkan perlunya optimasi atau integrasi metode tambahan untuk meningkatkan keandalan deteksi dalam skenario pertandingan nyata.

Penelitian berikutnya dilakukan oleh Sholehurrohman et al. (2023) dengan judul “Analisis Metode *Kalman Filter*, *Particle Filter* dan *Correlation Filter* Untuk Pelacakan Objek” membahas mengenai tantangan dalam pelacakan objek (*object tracking*) pada bidang *computer vision*. Penelitian ini mengimplementasikan tiga metode, yaitu *Kalman Filter*, *Particle Filter*, dan *Correlation Filter* untuk pelacakan objek pada data video lalu lintas dan video sirkuit Naskar. Hasil penelitian menunjukkan bahwa metode *Kalman Filter* memiliki akurasi tertinggi mencapai 96,89%, sedangkan metode *Correlation Filter* lebih unggul dalam aspek performa komputasi dengan rata-rata 26,69 FPS, sementara *Particle Filter* berada di bawah *Kalman Filter* dalam hal akurasi. Kesimpulan dari penelitian ini menegaskan bahwa *Kalman Filter* sangat potensial digunakan dalam pelacakan objek yang membutuhkan akurasi tinggi, sementara *Correlation Filter* lebih sesuai untuk kebutuhan aplikasi *real-time* karena efisiensi komputasinya. Dengan demikian, penelitian ini dapat dijadikan acuan penting dalam mendukung pemanfaatan *Kalman Filter* pada penelitian terkait optimasi pergerakan robot sepak bola, khususnya dalam memprediksi pergerakan bola secara akurat.

Penelitian kelima berjudul “Implementasi Algoritma SIFT (*Scale-Invariant Feature Transform*) dan Algoritma *Kalman Filter* dalam Mendeteksi Objek Bola” yang dilakukan oleh C. Saputra (2023) berfokus pada pengembangan sistem pendekripsi bola pada robot KRSBI Beroda. Latar belakang penelitian ini adalah tuntutan agar robot mampu mendeteksi, melacak, serta menggiring bola menuju gawang lawan secara efektif dalam KRSBI Beroda. Metode pendekripsi berbasis *color filtering* sebelumnya memang dinilai cukup baik dalam mengidentifikasi objek, namun masih memiliki kelemahan dalam aspek pelacakan (*tracking*). Untuk mengatasi hal tersebut, penelitian ini menggabungkan algoritma SIFT yang berfungsi membandingkan fitur citra guna memastikan objek yang terdeteksi benar-benar bola, serta algoritma *Kalman Filter* yang berperan sebagai estimator dalam memprediksi arah pergerakan bola berdasarkan status objek pada *frame* sebelumnya. Hasil dari penelitian ini menunjukkan bahwa kombinasi algoritma SIFT dan *Kalman Filter* dapat meningkatkan akurasi serta kecepatan pendekripsi bola. *Tracking* yang dilakukan dengan *Kalman Filter* mampu memprediksi pergerakan objek dengan baik, di mana koordinat *y* akan semakin kecil jika bola bergerak ke atas *frame* dan semakin besar jika bergerak ke bawah, sedangkan koordinat *x* akan semakin kecil ketika bola bergerak ke kiri dan semakin besar saat bergerak ke kanan. Dari hasil pengujian, sistem berhasil mendeteksi objek bola dengan sempurna pada jarak tertentu, dengan rata-rata *error* pengukuran *Kalman Filter* sebesar 1,06 untuk koordinat *x* dan 7,34 untuk koordinat *y*. Dengan demikian, penelitian ini menegaskan potensi integrasi SIFT dan *Kalman Filter* untuk menghasilkan sistem deteksi dan pelacakan bola yang lebih andal dalam mendukung performa robot KRSBI.

Penelitian selanjutnya yang relevan berjudul “*Detection and Recognition of Moving Video Objects: Kalman Filtering with Deep Learning*” yang dilakukan oleh Mohammed & Hussain (2021). Penelitian ini bertujuan untuk meningkatkan akurasi dalam proses deteksi dan pengenalan objek bergerak dalam urutan video. Tujuan ini diangkat karena adanya faktor penghalang seperti jarak deteksi kamera atau kekaburan (*blurring*) gambar yang dapat mengurangi akurasi teknik yang ada. Untuk mencapai akurasi yang lebih tinggi, metode yang diusulkan adalah sistem

hibrida yang menggabungkan *Kalman Filter* dengan CNN. *Kalman Filter* diterapkan pada tahap awal deteksi untuk menghilangkan latar belakang dan memotong objek, serta berfungsi sebagai estimator rekursif yang mampu memprediksi lokasi objek di masa depan, mengurangi *noise* dari deteksi yang salah, dan mengasosiasikan multi-objek ke treknya. Setelah itu, model CNN akan memprediksi kategori objek yang sudah dideteksi dan dipotong. Hasil eksperimen menunjukkan bahwa pendekatan hibrida ini berhasil mencapai akurasi pengenalan hingga 100% pada 8 video berbeda. Hasil ini menunjukkan superioritas sistem yang diusulkan dibandingkan dengan enam algoritma lain yang ada.

Penelitian selanjutnya adalah studi berjudul "*Optimized Object Tracking Technique Using Kalman Filter*" yang dilakukan oleh Taylor et al. (2016). Penelitian ini berfokus pada pengembangan teknik pelacakan objek yang efisien untuk mengatasi masalah waktu pemrosesan yang tinggi dalam pendekripsi objek di tengah adegan yang berantakan (*cluttered scene*). Metode konvensional pelacakan berbasis fitur seperti SIFT atau SURF (*Speeded Up Robust Feature*) dinilai akurat, tetapi membutuhkan waktu pemrosesan yang lebih tinggi. Sebaliknya, metode berbasis warna memiliki waktu pemrosesan yang lebih cepat, namun akurasi yang terbatas. *Kalman Filter* digunakan dalam penelitian ini adalah untuk mengatasi kompromi tersebut dan meningkatkan efisiensi komputasi, khususnya untuk aplikasi *real-time* seperti pada sistem robotik. *Kalman Filter* berperan sebagai estimator rekursif yang efisien untuk memprediksi lokasi objek di *frame* berikutnya berdasarkan status saat ini dan model gerakan objek (diasumsikan kecepatan konstan). Prediksi lokasi ini memungkinkan sistem hanya mencari objek dalam jendela gambar yang dipotong (*cropped image*) yang ukurannya jauh lebih kecil daripada keseluruhan *frame* video. Dengan demikian, waktu pemrosesan pendekripsi objek dapat diminimalkan secara signifikan. Hasil dari penelitian ini menunjukkan bahwa integrasi *Kalman Filter* dengan teknik *cropping* secara signifikan mempercepat waktu pemrosesan sambil mempertahankan tingkat akurasi yang tinggi. Waktu pemrosesan menjadi lebih cepat ketika ukuran jendela pencarian (*search window*) diperkecil. Untuk menyeimbangkan antara waktu pemrosesan yang minimal dan kesalahan jarak (*distance error*) yang minimal,

penelitian ini menyimpulkan bahwa ukuran jendela pencarian yang optimal adalah 2.16 kali dimensi terbesar objek. Pada ukuran ini, terdapat penurunan signifikan dalam waktu pemrosesan sekaligus memastikan objek terdeteksi dengan tingkat keberhasilan yang tinggi, serta pusat objek yang terdeteksi cukup dekat dengan pusat sebenarnya.

Penelitian selanjutnya oleh Yunus Egi (2022), yang berjudul "*Basketball self training shooting posture recognition and trajectory estimation using computer vision and Kalman Filter*", secara khusus berfokus pada peningkatan performa *self-shooting* bola basket, terutama untuk pemain muda, yang sering mengalami kesulitan dan keengganan akibat postur yang salah dan tembakan yang meleset tanpa bimbingan pelatih. Penelitian ini dibuat sebagai upaya untuk menyediakan sistem umpan balik otomatis yang dapat melacak gerakan pemain, mengenali postur, dan mengestimasi lintasan proyektil secara *real-time*. Penelitian ini menggunakan teknik *computer vision* seperti pemisahan saluran warna RGB, *Median Filter*, binarisasi, dan *Area Opening* untuk mendeteksi serta melabeli objek penting, yaitu bola basket dan *T-shirt* pemain. Dengan mengestimasi lintasan proyektil, penelitian ini menunjukkan bahwa lintasan tersebut dipengaruhi secara signifikan oleh ketidakpastian lingkungan, khususnya gaya hambat udara (*air drag force*). Untuk mengatasi *noisy medium* (medium bising) ini dan mengoptimalkan lintasan yang sebenarnya, algoritma *Kalman Filter* digunakan sebagai filter rekursif canggih untuk memprediksi posisi, kecepatan, dan percepatan objek bergerak serta mengurangi kesalahan prediksi secara berulang. Hasilnya, penelitian ini berhasil menentukan bahwa sudut tembakan optimal (*best shooting angle*) yang menghasilkan skor sukses paling tinggi bagi pemain dengan tinggi 1.73 m dan kecepatan awal 9.5 m/s adalah 50° ketika gaya hambat udara diperhitungkan. Perhitungan ini menunjukkan deviasi yang signifikan; tanpa hambatan udara, jangkauan maksimum horizontal ( $x_{max}$ ) adalah 10.76 m, namun dengan hambatan udara,  $x_{max}$  turun menjadi 5.47 m, menegaskan peran krusial *Kalman Filter* dalam memprediksi lintasan secara akurat dalam kondisi dunia nyata.

Penelitian selanjutnya adalah “*ConfTrack: Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box*” yang diteliti oleh

Jung et al. (2024). Urgensi penelitian ini muncul dari kelemahan mendasar *tracker* berbasis *Kalman Filter-based Tracking-by-Detection* (KFTBD) konvensional ketika dihadapkan pada hasil deteksi yang *noisy* (kotak dengan *confidence* rendah) dalam situasi ramai. Deteksi berkepercayaan rendah ini terbukti berhubungan dengan nilai *Intersection over Union* (IoU) yang lebih rendah, yang secara langsung mengganggu kinerja pelacakan dan berpotensi menyebabkan *ID switch* atau kegagalan *track*. Untuk mengatasi masalah ini, *ConfTrack* menyarankan peningkatan fungsi *Kalman Filter* yang adaptif terhadap skor kepercayaan. Fungsi utama *Kalman Filter* dalam penelitian ini tetap sebagai estimator yang memprediksi dan memperbarui status *track*, namun ditambahkan tiga metode utama untuk menanggulangi *noise*, yaitu *Confidence Weighted Kalman-Update* (CW) untuk melakukan modifikasi pada pengukuran target *Kalman Filter*. Jika skor kepercayaan deteksi rendah, pengukuran yang digunakan akan lebih condong ke prediksi *Kalman Filter* daripada ke deteksi mentah (*noisy*). Kemudian menggunakan *Noise Scale Adaptive Kalman Filter* (NK) untuk memperkuat kovariansi *noise* ruang pengukuran yang digunakan untuk menghitung *Kalman gain* ketika skor kepercayaan rendah. Hal ini secara efektif memberikan penalti pada deteksi berkepercayaan rendah di tahap *matching*. Terakhir, menggunakan *Constant Box Prediction* (CP) untuk menstabilkan prediksi *Kalman Filter* untuk *lost track* dengan mengatur variasi ukuran kotak menjadi nol, meminimalkan dampak *noise* pada prediksi dimensi kotak. Hasil penelitian menunjukkan bahwa kombinasi metode berbasis *Kalman Filter* ini, khususnya NK, memberikan kontribusi paling konsisten dalam menaikkan metrik pelacakan. *ConfTrack* terbukti paling kuat di lingkungan ramai dan berkinerja baik dalam berbagai situasi, mencapai skor *Higher Order Tracking Accuracy* (HOTA) dan *Identification F1-Score* (IDF1) tertinggi pada *dataset Multiple Object Tracking Benchmark* 2020 (MOT20). Pencapaian IDF1 tertinggi menunjukkan bahwa *ConfTrack* sangat kuat terhadap *noise* deteksi di lingkungan ramai, memungkinkannya melacak secara stabil dengan tingkat *missing track* yang rendah dan menghasilkan *trajectory* yang stabil saat terjadi oklusi. Dengan demikian, penelitian ini menegaskan kembali

peran krusial *Kalman Filter* dalam *multi-person tracking*, khususnya melalui penyesuaian adaptif terhadap kualitas data input deteksi.

Penelitian selanjutnya adalah “*Automatic pear and apple detection by videos using deep learning and a Kalman Filter*” oleh Takura et al. (2021). Penelitian ini mengatasi tantangan krusial dalam otomatisasi pertanian, yaitu estimasi jumlah buah di kebun secara akurat. Proses penghitungan buah dibagi menjadi dua langkah penting: deteksi buah dan pelacakan buah (*tracking*) melalui urutan *frame* gambar. Urgensi pelacakan muncul karena kondisi lapangan yang tidak stabil, seperti perubahan pencahayaan di bawah kanopi pohon, serta kemungkinan buah tertutup oleh daun atau ranting (*occlusion*), yang dapat menyebabkan kegagalan deteksi di beberapa *frame*. Tanpa pelacakan yang andal, buah yang muncul kembali setelah tersembunyi dapat dihitung ganda (*double-counted*), sehingga hasil penghitungan menjadi tidak akurat. Untuk mengatasi masalah ini, metode deteksi objek berbasis *deep learning* YOLO v2 digunakan untuk mengidentifikasi buah di setiap *frame*. Kemudian, *Kalman Filter* diterapkan untuk pelacakan objek. Fungsi utama *Kalman Filter* adalah sebagai estimator yang memprediksi dan mengoreksi status (posisi dan kecepatan) buah yang sedang dilacak, bahkan ketika deteksi buah gagal di *frame* tersebut. Ini memastikan bahwa buah yang sama dapat terus diidentifikasi di *frame* yang berurutan, meskipun sempat tidak terdeteksi. Hasilnya menegaskan efektivitas integrasi ini: sistem berhasil menghitung buah pir dan apel secara otomatis dengan kesalahan absolut kurang dari 10%. Secara spesifik, dari total 234 buah pir, 226 buah berhasil dihitung dengan benar, menghasilkan nilai F1 sebesar 0.972. Performa *Kalman Filter* juga terbukti jauh lebih unggul dibanding algoritma pelacakan fitur Kanade-Lucas-Tomasi (KLT), yang menyebabkan penghitungan berlebih (*over-counted*). Dengan demikian, penelitian ini menunjukkan bahwa *Kalman Filter* sangat efektif dan esensial dalam menstabilkan proses penghitungan objek bergerak pada video di lingkungan yang tidak terkontrol.

**Tabel 2.1.** Penelitian terdahulu

No	Peneliti	Tahun	Judul	Masalah	Metode	Hasil	Perbedaan
1	T. Mohd. Farhan, Feri Chandra	2024	<i>CNN-Based Ball and Goal Detection for KRSBI Robot with Omnidirectional Camera</i>	Metode deteksi objek tradisional seperti penyaringan warna HSV memiliki kinerja yang buruk di bawah variasi kondisi pencahayaan pada robot KRSBI-Beroda	Menggunakan CNN berbasis algoritma YOLOv8 dengan kamera <i>omnidirectional</i> . Model dilatih menggunakan 1.125 gambar selama 100 epochs.	Model mencapai Akurasi 95,87%, Presisi 1.00 (pada confidence 0.921) 8, Recall 0.99, dan F1-Score 0.97 (pada confidence 0.149).	Penelitian ini fokus meningkatkan kinerja robot dari menggunakan HSV ke YOLO, sedangkan penelitian sekarang menambahkan optimasi gerakan dengan <i>Kalman Filter</i> .
2	Bagas Musamma Nanda, Simon Siregar, dan Muhammad Ikhsan Sani	2023	Implementasi <i>Object Detection</i> pada Robot Sepak Bola Beroda Berbasis Kamera <i>Omnidirectional</i> Menggunakan OpenCV	Deteksi objek yang bergantung pada intensitas cahaya, sehingga membuat hasil deteksi menjadi kurang efektif	OpenCV <i>Object Detection</i>	Robot bisa deteksi bola, gawang, robot cyan & magenta dengan akurasi ±70%, tapi hasil dipengaruhi intensitas cahaya.	Penelitian ini hanya menggunakan OpenCV sederhana, sedangkan penelitian sekarang memakai YOLO + <i>Kalman Filter</i> .

3	Jalu Nuralim, Nifty Fath, Akhmad Musafa, Sujono, Drs. Suwasti Broto	2022	Perancangan Sistem Pendeteksian Obyek Bola dengan Metode <i>Framework</i> YOLO V4	Robot butuh deteksi cepat untuk navigasi & menendang bola, namun terhambat jika bola terhalang sebagian.	YOLOv4 + <i>Confusion Matrix</i>	Robot masih bisa deteksi bola hingga 70% tertutup, gagal pada >80%.	Fokus pada akurasi YOLOv4 dengan halangan, sedangkan penelitian sekarang menambahkan prediksi pergerakan menggunakan <i>Kalman</i> <i>Filter</i> .
4	Ridho Sholehurrohman, Mochammad Reza Habibi, Igit Sabda Ilman, Rahman Taufiq, Muhaqiqin	2023	Analisis Metode <i>Kalman Filter</i> , <i>Particle Filter</i> dan <i>Correlation</i> <i>Filter</i> Untuk Pelacakan Objek	<i>Tracking</i> objek sering gagal karena oklusi dan deformasi target.	<i>Kalman Filter</i> , <i>Particle Filter</i> , <i>Correlation</i> <i>Filter</i>	<i>Kalman Filter</i> akurasi 96,89%, <i>Correlation</i> <i>Filter</i> paling cepat 26,69 FPS.	Penelitian ini membandingkan metode tracking umum, penelitian sekarang fokus <i>Kalman Filter</i> untuk robot sepak bola berbasis YOLO.
5	M. Irwan Bustami, Chindra Saputra, Desi Kisbianty, Arjuna Panji Prakarsa	2023	Implementasi Algoritma SIFT ( <i>Scale-</i> <i>Invariant</i> <i>Feature</i> <i>Transform</i> ) dan <i>Kalman</i> <i>Filter</i> dalam Mendeteksi Objek Bola	Metode <i>color</i> <i>filtering</i> saja kurang handal untuk <i>tracking</i> bola.	SIFT + <i>Kalman</i> <i>Filter</i>	SIFT mengenali objek, <i>Kalman</i> <i>Filter</i> memprediksi arah bola. <i>Error</i> rata- rata 1.06 (x) dan 7.34 (y).	Penelitian ini gabungkan SIFT + <i>Kalman Filter</i> , sedangkan penelitian sekarang pakai YOLO untuk deteksi dan <i>Kalman Filter</i> untuk prediksi.

6	Hind Rustum Mohammed & Zahir M. Hussain	2021	<i>Detection and recognition of moving video objects: Kalman Filtering with deep learning</i>	Kebutuhan akan akurasi yang lebih tinggi dalam deteksi dan pengenalan objek bergerak, terutama menghadapi kendala seperti jarak kamera atau <i>blurring</i> gambar yang menyebabkan eror pada teknik yang sudah ada.	<i>Kalman Filter</i> untuk deteksi, penghilangan latar belakang, dan <i>tracking</i> . Kemudian CNN digunakan untuk pengenalan kategori objek.	Akurasi pengenalan mencapai 100% pada 8 video berbeda. Sistem menunjukkan superioritas dibandingkan enam algoritma yang ada.	Tujuan utama adalah deteksi dan pengenalan objek bergerak secara umum. <i>Kalman Filter</i> digunakan sebagai tahap awal untuk deteksi dan tracking. Metode deteksi awal tidak menggunakan YOLO, melainkan <i>Temporal Median</i> dan <i>thresholding</i> untuk menghilangkan <i>background</i> .
7	Liana Ellen Taylor, Midriem Mirdanies, Roni Permana Saputra	2016	<i>Optimized Object Tracking Technique Using Kalman Filter</i>	Waktu pemrosesan yang tinggi pada deteksi objek, terutama untuk aplikasi <i>real-time</i> , sambil mempertahankan akurasi dalam adegan yang	<i>Kalman Filter</i> (sebagai estimator/prediktor posisi) dikombinasikan dengan <i>Color Segmentation</i> (untuk deteksi) dan teknik <i>Cropped Image</i> (pemotongan	Proses pelacakan menjadi lebih cepat secara signifikan (waktu pemrosesan berkurang) dengan mempertahankan akurasi tinggi. Ukuran jendela pencarian optimal	Penelitian ini menggunakan Segmentasi Warna dan <i>Kalman Filter</i> untuk mengoptimalkan waktu pemrosesan melalui teknik <i>Cropped Image</i> . Sedangkan penelitian sekarang menggunakan

				berantakan ( <i>cluttered scene</i> ).	gambar) untuk membatasi area pencarian.	ditemukan pada 2.16 kali dimensi terbesar objek, menyeimbangkan waktu pemrosesan dan kesalahan jarak.	YOLOv8 (model <i>deep learning</i> yang kompleks) sebagai detektor, dan <i>Kalman Filter</i> digunakan untuk stabilisasi <i>tracking</i> dan mengatasi oklusi/ <i>false negative</i> akibat beban komputasi YOLO yang tinggi. Fokus optimasi dalam penelitian ini adalah mengurangi area pencarian ( <i>cropped image</i> ), sementara penelitian sekarang adalah menstabilkan <i>output</i> detektor modern (YOLO).
8	Yunus Egi	2022	<i>Basketball self training shooting posture recognition and trajectory estimation using computer</i>	Kurangnya umpan balik dari pelatih membuat <i>self-shooting</i> menyulitkan dan memakan waktu, di mana postur salah	Menggunakan kombinasi <i>computer vision</i> tradisional (RGB <i>Channelization</i> , Median Filter, Binarisasi, <i>Area Opening</i> ) untuk deteksi <i>centroid</i>	Sistem berhasil mengklasifikasikan objek dan menentukan sudut postur terbaik. <i>Kalman Filter</i> terbukti penting karena tanpa hambatan	Penelitian ini berfokus pada analisis postur manusia dan estimasi lintasan proyektil fisik (bola basket) dalam domain olahraga/biomekanika, menggunakan CV tradisional. <i>Kalman</i>

			<i>vision and Kalman Filter</i>	menyebabkan tembakan meleset dan menimbulkan keengganahan. Diperlukan algoritma untuk memberikan <i>feedback</i> postur dan lintasan <i>real-time</i> .	bola dan <i>T-shirt</i> . Dilanjutkan dengan <i>Kalman Filter</i> untuk mengoptimalkan dan mengoreksi prediksi lintasan proyektil di lingkungan bising (adanya gaya hambat udara)	udara ( $x_{max}$ 10.76 m), hasil berbeda signifikan dari kondisi nyata dengan hambatan udara ( $x_{max}$ 5.47 m), sehingga filter berhasil mengoreksi prediksi lintasan fisik. Sudut terbaik yang terdeteksi 51°.	<i>Filter</i> berfungsi untuk mengoreksi noise fisik (gaya hambat udara) pada lintasan. Penelitian sekarang berfokus pada optimasi gerakan robot dalam domain Robot Sepak Bola, menggunakan CV berbasis YOLOv8. <i>Kalman Filter</i> berfungsi untuk menstabilkan <i>noise</i> deteksi objek (oklusi/ <i>false negative</i> ) dan mereduksi beban komputasi.
9	Hyeonchul Jung, Seokjun Kang, Takgen Kim, HyeongKi Kim	2024	<i>ConfTrack: Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box</i>	Kegagalan tracking dan <i>ID switch</i> pada <i>tracker</i> KFTBD konvensional yang disebabkan oleh hasil deteksi	Mengembangkan <i>tracker</i> <i>ConfTrack</i> berbasis KFTBD dengan penalti deteksi berkepercayaan rendah (melalui <i>Confidence Weighted</i>	<i>ConfTrack</i> mencapai skor HOTA dan IDF1 tertinggi pada dataset MOT20, membuktikan <i>robustness</i> yang tinggi terhadap <i>noise</i> deteksi di	<i>ConfTrack</i> berfokus pada <i>Multi-Person Tracking</i> di lingkungan ramai, sementara penelitian sekarang adalah <i>Single-Object Tracking</i> (Bola) untuk robot sepak bola. <i>ConfTrack</i>

				<i>(bounding box)</i> berkepercayaan rendah ( <i>noisy detections</i> ) di lingkungan yang ramai ( <i>crowd circumstances</i> )	<i>Kalman-Update</i> (CW) dan <i>Noise Scale Adaptive Kalman Filter</i> (NK)) pada <i>Kalman Filter</i> , dan strategi <i>cascading matching</i> baru (LM) untuk memanfaatkan deteksi berkepercayaan rendah.	lingkungan yang padat.	memodifikasi <i>Kalman Filter</i> untuk mempenalti/menghukum deteksi berkepercayaan rendah guna stabilitas ID. Penelitian sekarang menggunakan <i>Kalman Filter</i> untuk memprediksi pergerakan bola secara stabil, mengoptimasi gerakan robot, dan mengurangi beban komputasi pada YOLO.
10	Kenta Itakura, Yuma Narita, Shuhei Noaki, & Fumiki Hosoi	2021	<i>Automatic pear and apple detection by videos using deep learning and a Kalman Filter</i>	Penghitungan buah di kebun dari video sulit karena perubahan kondisi cahaya yang cepat dan tidak stabil, buah dapat tertutup daun/ranting ( <i>occlusion</i> ). Ini	YOLOv2 untuk deteksi objek, kemudian melakukan pelacakan menggunakan <i>Kalman Filter</i> .	Rata-rata presisi ( <i>Average Precision</i> ) deteksi pir dan apel tinggi (0.97). Penghitungan pir benar: 226 dari 234 ( $F_1 = 0.972$ ). <i>Kalman Filter</i> mampu melacak buah yang sempat	Penelitian ini fokus pada objek statis (buah) di lingkungan pertanian yang tidak terkontrol (pencahaayaan, oklusi, pergerakan kamera <i>handheld</i> ), sedangkan penelitian sekarang fokus pada objek dinamis (bola) yang bergerak cepat oleh

				menyebabkan kegagalan deteksi di beberapa <i>frame</i> dan potensi penghitungan ganda.		tidak terdeteksi ( <i>occluded</i> ).	robot di lingkungan yang lebih terkontrol. Kemudian fungsi <i>Kalman Filter</i> (KF) di penelitian ini untuk mengatasi kegagalan deteksi sesaat dan mencegah <i>double-counting</i> . Sedangkan dalam penelitian sekarang, KF digunakan untuk optimasi gerakan robot dan stabilisasi <i>computer vision</i> (mengatasi oklusi, <i>false negative</i> , dan mengurangi beban komputasi YOLOv8).
--	--	--	--	--	--	---------------------------------------	--

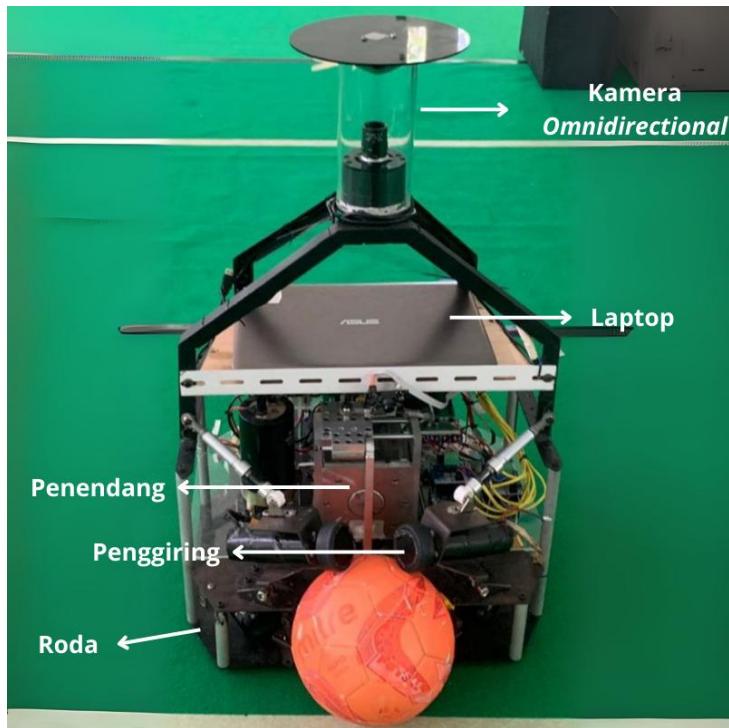
## **2.2. Kompetisi Robot Sepak Bola Beroda Indonesia**

Salah satu ajang kompetisi robotik di Indonesia adalah Kontes Robot Indonesia atau lebih sering disebut dengan KRI. KRI diselenggarakan oleh Pusat Prestasi Nasional (PUSPRESNAS) dan Kementerian Pendidikan dan Kebudayaan Republik Indonesia. KRI merupakan acara yang diadakan setiap tahun dan diikuti oleh mahasiswa dari berbagai wilayah di Indonesia mulai dari Timur, Tengah dan Barat. KRI terbagi menjadi 6 kategori, yang salah satunya adalah Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda (Nanda et al., 2023).

Kontes Robot Sepakbola Beroda Indonesia diadakan untuk meningkatkan keilmuan dan kreatifitas mahasiswa di bidang robotika. Di dalam kontes ini, mahasiswa dituntut untuk bisa mengembangkan kemampuan dalam mekanika, manufaktur, elektronika, pemrograman, *artificial intelligent, image processing*, komunikasi digital, dan strategi, sekaligus diperlukan pengembangan ke arah disiplin, toleransi, sportifitas, kerjasama, saling menghargai, kontrol emosi dan kemampuan *softskill* lainnya. (Kusumoputro et al., 2024).

## **2.3. Robot Sepak Bola Beroda**

Dalam KRSBI Beroda, robot yang digunakan pada tahap ini merupakan robot yang sama seperti yang dipakai pada pertandingan tingkat nasional, dengan beberapa ketentuan khusus. Jumlah robot yang diizinkan adalah dua unit, yaitu Robot 1 (R1) dan Robot 2 (R2), keduanya bertipe robot penyerang. Adapun spesifikasi fisik robot diatur dengan ukuran proyeksi ke lantai minimal 30 cm × 30 cm dan maksimal 52 cm × 52 cm, dengan tinggi robot antara 40 cm hingga 80 cm. Jika tinggi robot melebihi 60 cm, maka bagian tubuh robot di atas ketinggian tersebut harus berada dalam silinder berdiameter 25 cm. Berat maksimum robot ditetapkan 40 kg, sedangkan bentuk robot dibuat bebas selama sesuai regulasi, dan warna yang digunakan adalah hitam (Kusumoputro et al., 2024).



**Gambar 2.1.** Robot sepak bola beroda

Robot yang digunakan dalam Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda memiliki bentuk khas menyerupai kubus dengan berbagai komponen penting yang berfungsi mendukung pergerakan serta kemampuan bermain sepak bola. Gambar 2.1 memperlihatkan salah satu contoh robot KRSBI Beroda yang dilengkapi dengan berbagai sistem pendukung.

. Kamera *omnidirectional* ditempatkan di bagian atas robot dan berfungsi untuk mendeteksi objek seperti bola, gawang, maupun robot lawan. Konsep kamera *omnidirectional* adalah menangkap citra dari segala arah (depan, belakang, kiri, dan kanan) menggunakan teknik pantulan cermin cembung yang diarahkan ke bawah (Surya et al., 2025). Dengan cara ini, kamera mampu memperoleh citra lapangan secara menyeluruh hanya dari satu titik pengamatan. Robot dilengkapi dengan penendang berbasis solenoid. Solenoid adalah sebuah komponen elektromagnetik yang berfungsi mengubah energi listrik menjadi energi mekanis. Energi mekanis yang dihasilkan dapat berupa gerakan mendorong (*push*) maupun menarik (*pull*) (Shofi et al., 2023).

Pada bagian depan robot terdapat roda kecil yang dipasangi motor penggerak dan berputar ke arah dalam tubuh robot. Roda ini berfungsi untuk menangkap dan menahan bola ketika robot bergerak, sehingga bola tetap berada di depan robot dan tidak mudah terlepas. Bagian terakhir adalah roda yang berfungsi sebagai sistem penggerak robot. Pada robot KRSBI Beroda digunakan tiga roda utama agar pergerakan lebih fleksibel. Setiap roda terdiri dari dua komponen, yaitu motor dengan kecepatan putar sekitar 500 rpm yang menjadi sumber tenaga penggerak, serta *omni-wheel* yang memungkinkan robot dapat bergerak bebas ke berbagai arah.

#### 2.4. OpenCV v3.4

OpenCV (*Open Source Computer vision Library*) adalah sebuah pustaka perangkat lunak yang dirancang khusus untuk pengolahan citra secara *real-time*. Pustaka ini awalnya dikembangkan oleh Intel dan kini didukung oleh Willow Garage serta Itseez. OpenCV didistribusikan dengan lisensi BSD yang bersifat permisif, sehingga lebih fleksibel dibandingkan GPL dan dapat digunakan secara bebas, termasuk untuk kepentingan komersial tanpa kewajiban membuka kode sumber. Selain itu, OpenCV menyediakan antarmuka yang kompatibel dengan berbagai bahasa pemrograman seperti C++, C, Python, dan Java, serta dapat dijalankan pada beragam sistem operasi, di antaranya Windows, Linux, Mac OS, iOS, dan Android. Pustaka ini dirancang untuk efisiensi komputasi dengan fokus utama pada aplikasi berbasis real-time (C. Saputra, 2023).

Modul pustaka OpenCV dirancang dengan tingkat fleksibilitas dan ketangguhan yang tinggi untuk menangani berbagai permasalahan dalam bidang *computer vision*. Berbagai solusi telah tersedia di dalamnya, seperti pemotongan citra (*cropping*), peningkatan kualitas citra melalui penyesuaian kecerahan, ketajaman, dan kontras, pendekripsi bentuk, segmentasi citra, pelacakan objek bergerak, hingga pengenalan objek, serta beragam fungsi lainnya (Ratna, 2020).

#### 2.5. NumPy v2.3

NumPy (*Numerical Python*) adalah paket dasar dan fundamental untuk komputasi ilmiah dengan Python. Sebagai pustaka *open-source*, NumPy menyediakan objek *array* N-dimensi (*ndarray*) yang sangat kuat, yang berfungsi

sebagai struktur data universal untuk pertukaran data multi-dimensi dalam ekosistem ilmiah Python. Objek *ndarray* ini memungkinkan operasi yang cepat dan efisien pada data homogen (semua elemen harus bertipe data yang sama) dalam jumlah besar, sebuah keunggulan signifikan dibandingkan list bawaan Python, yang dicapai melalui inti kode yang dioptimalkan dalam bahasa C. Selain struktur array multidimensi, NumPy juga menawarkan koleksi lengkap fungsi matematika tingkat tinggi, termasuk aljabar linear, transformasi Fourier, dan kemampuan angka acak, menjadikannya standar *de-facto* untuk komputasi *array* dan menjadi fondasi bagi pustaka ilmu data dan *machine learning* lainnya (NumPy, 2025).

NumPy berfungsi untuk melakukan operasi vektor matriks dengan mengolah *array* dan *array* multidimensi. NumPy juga dapat digunakan untuk kebutuhan dalam analisis data (Admaja, 2021).

## 2.6. *Kalman Filter*

*Kalman Filter* (KF) merupakan algoritma yang sangat kuat untuk memperkirakan keadaan suatu sistem dinamis berdasarkan serangkaian pengukuran yang tidak lengkap dan mengandung *noise*. Algoritma ini dikembangkan oleh Rudolf E. Kálmán, dan awalnya digunakan dalam bidang navigasi penerbangan. Namun, seiring waktu, *Kalman Filter* menjadi komponen penting di berbagai bidang, seperti robotika, ekonomi, dan terutama *computer vision* (CV). *Kalman Filter* bekerja melalui dua tahap utama yang berlangsung secara berulang, yaitu tahap prediksi dan tahap pembaruan (koreksi). Pada tahap pertama, sistem memperkirakan keadaan berikutnya beserta tingkat ketidakpastiannya. Kemudian, pada tahap pembaruan, hasil prediksi tersebut diperbaiki dengan memasukkan data pengukuran terbaru. Melalui proses ini, *Kalman Filter* mampu menghasilkan estimasi keadaan objek, seperti posisi dan kecepatannya, secara halus, akurat, dan stabil, meskipun data sensor yang diterima tidak sempurna (Ultralytics, 2025).

*Kalman Filter* dinamai dari penciptanya, Rudolf Kalman. Algoritma ini bekerja dengan menerima sejumlah data yang mengandung *noise*, kemudian menyaringnya untuk meminimalkan gangguan tersebut. Karena dirancang pada ruang *linear*, *Kalman Filter* juga dikenal dengan istilah *linear quadratic estimation*

(Sholehurrohman et al., 2023). Metode *Kalman Filter* menggunakan informasi dari objek yang terdeteksi di suatu *frame* dan status objek dari *frame* sebelumnya untuk mendapatkan status yang baru dari objek tersebut (C. Saputra, 2023).

Dalam konteks AI, *Kalman Filter* banyak digunakan dalam sistem pelacakan objek (*object tracking*). Setelah model deteksi seperti Ultralytics YOLO berhasil mengidentifikasi objek pada suatu *frame*, *Kalman Filter* digunakan untuk memperkirakan posisi objek pada *frame* berikutnya. Perkiraan ini didasarkan pada model gerak (*motion model*), yang umumnya mengasumsikan bahwa objek bergerak dengan kecepatan atau percepatan konstan (Ultralytics, 2025).

Ketika *frame* berikutnya diterima, model deteksi memberikan hasil pengukuran baru berupa koordinat *bounding box* objek. *Kalman Filter* kemudian menjalankan tahap pembaruan (*update*), yaitu memperbaiki hasil prediksi awal berdasarkan data baru tersebut. Proses ini sangat efektif karena beberapa alasan penting:

- Reduksi *noise*: *Kalman Filter* mampu menghaluskan hasil deteksi yang bergetar atau tidak stabil, sehingga jalur pelacakan menjadi lebih halus dan konsisten.
- Penanganan *occlusion*: Saat objek tidak terdeteksi selama beberapa *frame* (misalnya tertutup oleh benda lain), *Kalman Filter* tetap dapat memperkirakan posisinya. Dengan demikian, ketika objek muncul kembali, sistem dapat langsung mengenaliinya tanpa kehilangan jejak.
- Estimasi keadaan (*state estimation*): Selain posisi, *Kalman Filter* juga mampu memperkirakan parameter lain seperti kecepatan dan arah gerak objek, sehingga memberikan pemahaman yang lebih menyeluruh tentang pergerakan objek.

Kemampuan *Kalman Filter* untuk memproses data secara rekursif dan efisien secara komputasi membuatnya sangat cocok untuk aplikasi *real-time*. Banyak algoritma pelacakan modern seperti SORT (*Simple Online and Realtime Tracking*) dan ByteTrack menggunakan *Kalman Filter* sebagai komponen inti prediksi gerakannya. Bahkan, model Ultralytics terbaru seperti YOLOv11 juga memanfaatkan pendekatan ini dalam mode pelacakannya (Ultralytics, 2025).

Selain *Kalman Filter* standar, terdapat beberapa varian dan metode turunan yang dikembangkan untuk menangani sistem dengan karakteristik yang lebih kompleks.

- *Extended Kalman Filter* (EKF): Merupakan pengembangan dari *Kalman Filter* klasik yang dirancang untuk menangani sistem non-linear. Jika *Kalman Filter* standar hanya bekerja optimal pada sistem linear, EKF mengatasinya dengan cara melakukan *linearization* terhadap model sistem di setiap langkah waktu. Dengan pendekatan ini, EKF mampu memberikan estimasi keadaan objek yang tetap akurat meskipun hubungan antarvariabel dalam sistem tidak bersifat linear, seperti pada kasus kendaraan yang berbelok atau robot yang bergerak dengan lintasan melengkung.
- *Unscented Kalman Filter* (UKF): Untuk sistem yang sangat non-linear, proses *linearization* pada EKF sering kali tidak cukup akurat. UKF hadir sebagai solusi yang lebih presisi dan stabil, dengan menggunakan pendekatan *unscented transform*. Metode ini tidak membutuhkan perhitungan turunan Jacobian, melainkan menggunakan sejumlah *sigma points* yang merepresentasikan distribusi keadaan secara lebih realistik sehingga hasil estimasi menjadi lebih akurat.
- *Particle Filter*: Selain kedua varian tersebut, terdapat juga metode *Particle Filter*, yang digunakan pada sistem non-linear dan non-Gaussian. Berbeda dengan *Kalman Filter* yang berasumsi distribusi Gaussian, *Particle Filter* merepresentasikan distribusi probabilitas dengan sekumpulan sampel acak (partikel). Metode ini banyak digunakan dalam robotika, khususnya untuk lokalisasi dan pemetaan, karena kemampuannya menangani ketidakpastian yang kompleks.

Dalam penerapannya, berbagai framework modern telah menyediakan implementasi *Kalman Filter*. Misalnya, Ultralytics *framework* menggunakan *Kalman Filter* dalam algoritma *object tracking*, sementara pustaka OpenCV juga menyediakan modul *Kalman Filter* yang banyak digunakan untuk pelacakan objek secara *real-time* dalam aplikasi *computer vision* (Ultralytics, 2025).

Secara sederhana, berikut merupakan persamaan matematika algoritma *Kalman Filter* pada Persamaan 1 (Yuztiawan & Utaminingrum, 2017).

$$\hat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \hat{X}_{k-1} \quad (1)$$

*Keterangan:*

$\hat{X}_k$  : Estimasi saat ini

$K_k$  : *Kalman Gain*

$Z_k$  : Nilai dari hasil pendektsian prediksi

$\hat{X}_{k-1}$  : Estimasi sebelumnya

Kalman Filter beroperasi sebagai estimator jenis *predictor-corrector* yang bekerja dalam dua langkah utama: Prediksi dan Pembaruan. Tujuannya adalah untuk memberikan estimasi keadaan sistem ( $X$ ) yang optimal dengan meminimalkan kovariansi *error*, terutama saat berhadapan dengan nilai yang bising (*noisy*).

Langkah prediksi menggunakan estimasi keadaan dari langkah waktu sebelumnya ( $k - 1$ ) untuk memproyeksikan atau memprediksi keadaan sistem saat ini ( $X_k^-$ ).

- Tujuan: Memperkirakan di mana sistem akan berada pada langkah waktu  $k$  berdasarkan dinamika sistem yang diketahui dan input kontrol yang diterapkan.
- Proses: Filter memproyeksikan mean keadaan ( $X$ ) dan kovariansi keadaan ( $P$ ) dari langkah waktu sebelumnya ke langkah waktu saat ini.
- Kovariansi: Dalam proses prediksi, ketidakpastian (diwakili oleh kovariansi  $P$ ) diperbesar oleh *noise* proses ( $Q$ ).

Fungsi `kf_predict` menghitung mean keadaan ( $X$ ) dan kovariansi ( $P$ ) yang diprediksi.

```
from numpy import dot

def kf_predict(X, P, A, Q, B, U):
    # Prediksi Keadaan: X = (A * X_sebelum) + (B * U)
    X = dot(A, X) + dot(B, U)

    # Prediksi Kovariansi: P = (A * P_sebelum * A.T) + Q
```

```

P = dot(A, dot(P, A.T)) + Q

return(X,P)

```

### Kode 2.1 Contoh kode prediksi *Kalman Filter* dengan NumPy

Setelah prediksi selesai, langkah pembaruan menggunakan nilai yang baru ( $Y_k$ ) untuk mengoreksi prediksi tersebut.

- Tujuan: Menghitung estimasi akhir yang diperbarui untuk keadaan ( $X_k$ ) dan kovariansi ( $P_k$ ) pada langkah waktu  $k$  setelah melihat pengukuran.
- Residual/Inovasi ( $V_k$ ): Langkah pertama adalah menghitung residual atau inovasi. Ini adalah perbedaan antara pengukuran aktual ( $Y_k$ ) dan pengukuran yang diprediksi (berdasarkan keadaan yang diprediksi ( $X_k^-$ )).
- *Kalman Gain* ( $K_k$ ): Residual ini kemudian dikalikan dengan *Kalman Gain* ( $K_k$ ). *Kalman Gain* bertindak sebagai "berat" (*weight*) yang menentukan seberapa besar prediksi harus dikoreksi oleh pengukuran baru. *Gain* ini mempertimbangkan kovariansi prediksi ( $P_k^-$ ) dan kovariansi *noise* pengukuran ( $R_k$ ).
  - Jika noise pengukuran ( $R$ ) tinggi, gain akan kecil, artinya filter lebih memercayai prediksinya sendiri.
  - Jika kovariansi prediksi ( $P^-$ ) tinggi (ketidakpastian prediksi besar), *gain* akan besar, artinya filter lebih memercayai pengukuran baru.
- Koreksi: Hasil koreksi ini ditambahkan ke keadaan yang diprediksi ( $X_k^-$ ) untuk mendapatkan estimasi keadaan akhir ( $X_k$ ). Kovariansi juga diperbarui ( $P_k$ ) untuk mencerminkan ketidakpastian yang telah berkurang.

Fungsi `kf_update` menghitung *gain* dan mengoreksi *mean* ( $X$ ) serta kovariansi ( $P$ ).

```

from numpy import dot, linalg
from numpy.linalg import inv

def kf_update (X, P, Y, H, R):
    # Mean Prediksi Pengukuran (IM): H * X_prediksi
    IM = dot(H, X)
    # Kovariansi Prediksi Pengukuran (IS): H * P_prediksi * H.T + R
    IS = dot(H, dot(P, H.T)) + R

    # Kalman Gain (K): P_prediksi * H.T * inv(IS)
    K = dot(P, dot(H.T, inv(IS)))

    # Pembaruan Keadaan: X_baru = X_prediksi + K * (Y - IM)

```

```
X = X + dot(K, (Y - IM))
# Pembaruan Kovariansi: P_baru = P_prediksi - K * IS * K.T
P = P - dot(K, dot(IS, K.T))

# ... (kode lain untuk menghitung LH/likelihood)
return (X, P, K, IM, IS, LH) # Mengembalikan mean dan kovariansi
yang diperbarui
```

**Kode 2.2** Contoh kode perbaruan *state Kalman Filter* dengan NumPy

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1. Kerangka Pikiran**

#### **3.2. Studi Literatur**

Tahap studi literatur dalam metodologi penelitian ini bertujuan untuk mengumpulkan, menganalisis, dan mensintesis informasi teoretis dan praktis yang relevan dan dibutuhkan untuk perancangan serta implementasi sistem optimasi gerakan robot. Proses ini memastikan bahwa metode yang dipilih, khususnya integrasi YOLO dan *Kalman Filter*, didasarkan pada prinsip-prinsip teknis yang solid dan terkini.

Studi literatur difokuskan pada tiga area utama yang secara langsung mendukung perancangan sistem:

##### **3.2.1 Algoritma Deteksi Objek (YOLO)**

- Mengidentifikasi model YOLO yang paling sesuai untuk kebutuhan *real-time* dengan keterbatasan komputasi pada robot KRSBI Beroda, yaitu YOLOv8.
- Mempelajari format *output* data deteksi dari YOLO (koordinat pusat  $x$ ,  $y$ , lebar  $w$ , dan tinggi  $h$  *bounding box*) untuk dikonversi menjadi *input* data pengukuran yang dapat diproses oleh *Kalman Filter*.

##### **3.2.2 Algoritma Prediksi dan Pelacakan (*Kalman Filter*)**

- Mengkaji prinsip dasar *Kalman Filter*, terutama dalam konteks pelacakan objek bergerak (*object tracking*).
- Mempelajari implementasi *Kalman Filter* menggunakan pustaka yang efisien, seperti NumPy dan modul yang tersedia pada OpenCV atau Ultralytics untuk memastikan kompatibilitas dan efisiensi komputasi *real-time*.

### **3.2.3 Sistem Robotika dan Kontrol Gerak**

- Mempelajari spesifikasi teknis Robot Sepak Bola Beroda, khususnya mengenai sistem penglihatan (kamera *omnidirectional*) dan aktuator yang menentukan batasan dan kemampuan sistem kontrol gerak.
- Mengumpulkan informasi mengenai konversi data posisi terprediksi (*output Kalman Filter*) menjadi perintah gerakan yang akan dieksekusi oleh robot.

## **3.3. Identifikasi Masalah**

Identifikasi masalah ini berasal dari celah (*gap*) antara metode deteksi berbasis YOLO yang memiliki beban komputasi tinggi dan kebutuhan akan gerakan robot yang stabil dan *real-time*.

Masalah-masalah teknis yang perlu dipecahkan dalam penelitian ini meliputi:

### **3.3.1. Ketidakstabilan Data Deteksi Akibat Keterbatasan Komputasi (*Frame Rate Issue*)**

Model deteksi YOLO (misalnya YOLOv8) membutuhkan kapasitas komputasi yang tinggi. Keterbatasan perangkat keras pada robot KRSBI Beroda seringkali menyebabkan penurunan frame rate atau keterlambatan proses inferensi. Masalah ini menghasilkan data posisi bola yang tidak stabil atau terlambat diterima oleh sistem kontrol, yang pada akhirnya mengakibatkan pergerakan robot menjadi tidak mulus atau tidak responsif.

### **3.3.2. Kerentanan Terhadap *Occlusion* dan *False Negative Detection***

Pada kondisi pertandingan yang dinamis, sering terjadi *occlusion* (bola tertutup sebagian) yang menyebabkan *false-negative detection* (bola tidak terdeteksi). Tanpa mekanisme prediksi, robot akan berhenti bergerak atau kehilangan jejak bola saat *occlusion* sesaat terjadi, menghambat kontinuitas permainan.

### **3.3.3. Kebutuhan untuk Prediksi Posisi Gerak Dinamis**

Detektor berbasis *frame* tunggal (YOLO) hanya memberikan posisi bola saat ini tanpa mempertimbangkan dinamika gerak bola tersebut. Untuk mengoptimalkan gerakan robot yang responsif terhadap perubahan posisi bola yang cepat dan

mendadak, dibutuhkan sebuah mekanisme prediksi yang mampu mengestimasi posisi bola di *frame* berikutnya.

Tiga masalah teknis di atas akan diselesaikan melalui implementasi dan integrasi *Kalman Filter* pada *output* deteksi YOLO. *Kalman Filter* berperan sebagai estimasi rekursif untuk memprediksi posisi, menghaluskan data yang bising (*noisy*), dan mengatasi data yang hilang (*occlusion*).

### 3.4. Persiapan Data dan Konfigurasi Model Deteksi YOLO

Model deteksi yang digunakan dalam penelitian ini telah dilatih sebelumnya menggunakan kumpulan data citra yang diambil langsung dari kamera *omnidirectional* robot KRSBI. *Dataset* ini dikumpulkan untuk merepresentasikan berbagai variasi pencahayaan dan posisi bola di lapangan, guna memastikan model memiliki ketahanan terhadap perubahan lingkungan. Proses pra-pemrosesan data melibatkan pemberian label (*labelling*) pada objek bola menggunakan *bounding box*. Keluaran dari proses pelabelan ini menghasilkan koordinat *ground truth* yang kemudian digunakan untuk pelatihan model *deep learning*.

Model yang dipilih adalah arsitektur YOLOv8, diimplementasikan menggunakan *framework* Ultralytics. Pemilihan YOLOv8 didasarkan pada keunggulannya dalam memberikan keseimbangan optimal antara kecepatan inferensi dan akurasi, yang sangat penting untuk mencapai kinerja *real-time* pada robot dengan sumber daya komputasi terbatas. Model ini bertanggung jawab untuk menghasilkan prediksi lokasi bola yang kemudian menjadi input bagi algoritma pelacakan *Kalman Filter*.

Setelah model YOLOv8 berhasil mendeteksi bola, data hasil deteksi diekstraksi. Data ini berbentuk bounding box dengan format  $(x_c, y_c, w, h, C)$ , di mana  $(x_c, y_c)$  adalah koordinat pusat bola pada citra, dan  $C$  adalah *confidence score*. Koordinat pusat  $(x_c, y_c)$  inilah yang dijadikan Vektor Pengukuran ( $Z_k$ ) yang bising (*noisy*) dan siap diolah lebih lanjut oleh *Kalman Filter*. *Confidence score* juga dapat dimanfaatkan untuk memverifikasi keandalan pengukuran saat proses pembaruan (koreksi) pada *Kalman Filter*.

### 3.5. Implementasi Sistem Deteksi dan Pelacakan

Sistem ini mengadopsi arsitektur *Tracking-by-Detection* di mana model YOLOv8 berfungsi sebagai detektor objek yang menghasilkan *measurement* (pengukuran), dan *Kalman Filter* berfungsi sebagai *tracker* (pelacak) yang melakukan prediksi dan koreksi. Keluaran dari YOLO berupa koordinat pusat bola ( $x_c, y_c$ ) pada setiap *frame* digunakan sebagai input pengukuran ( $Z_k$ ) yang bising ke *Kalman Filter*. *Kalman Filter* kemudian menjalankan siklus prediksi dan pembaruan untuk menghasilkan estimasi keadaan bola yang optimal. Keluaran *Kalman Filter* adalah posisi dan kecepatan bola yang telah dihaluskan dan diprediksi, siap diteruskan ke modul kontrol gerak robot.

Model yang dipilih adalah arsitektur YOLOv8, diimplementasikan menggunakan *framework* Ultralytics. Pemilihan YOLOv8 didasarkan pada keunggulannya dalam memberikan keseimbangan optimal antara kecepatan inferensi dan akurasi, yang sangat penting untuk mencapai kinerja *real-time* pada robot dengan sumber daya komputasi terbatas. Model ini bertanggung jawab untuk menghasilkan prediksi lokasi bola yang kemudian menjadi input bagi algoritma pelacakan *Kalman Filter*.

Setelah model YOLOv8 berhasil mendeteksi bola, data hasil deteksi diekstraksi. Data ini berbentuk bounding box dengan format  $(x_c, y_c, w, h, C)$ , di mana  $(x_c, y_c)$  adalah koordinat pusat bola pada citra, dan  $C$  adalah *confidence score*. Koordinat pusat  $(x_c, y_c)$  inilah yang dijadikan Vektor Pengukuran ( $Z_k$ ) yang bising (*noisy*) dan siap diolah lebih lanjut oleh *Kalman Filter*. *Confidence score* juga dapat dimanfaatkan untuk memverifikasi keandalan pengukuran saat proses pembaruan (koreksi) pada *Kalman Filter*.

*Kalman Filter* (KF) dikonfigurasi untuk memodelkan keadaan bola dengan 8 status (*state*) dan menerima 4 pengukuran (*measurement*).

- Vektor Keadaan ( $X$ ): Mewakili posisi, dimensi, dan kecepatan perubahan objek, didefinisikan sebagai  $[x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]^T$ . Di sini,  $x, y$  adalah koordinat pusat,  $w, h$  adalah lebar dan tinggi *bounding box*, dan  $\dot{x}, \dot{y}, \dot{w}, \dot{h}$  adalah laju perubahan (kecepatan) dari variabel-variabel tersebut.

- Vektor Pengukuran ( $Z$ ): Mewakili data yang diterima dari detektor YOLOv8 pada *frame* tertentu, didefinisikan sebagai:  $[x, y, w, h]^T$ .

Strategi *Frame Skipping* diterapkan untuk optimasi komputasi dan stabilisasi gerakan. Alur kerjanya adalah sebagai berikut:

1. Pengukuran YOLO (misal *frame* 1): Pada *frame* awal, YOLOv8 dijalankan untuk mendapatkan data pengukuran  $Z$ . Data ini digunakan untuk menginisialisasi atau memperbarui status KF.
2. Prediksi KF (*Skip Frames*): Untuk beberapa *frame* berikutnya (misalnya *frame* 2 hingga *frame* 5), sistem tidak menjalankan YOLO. Sebaliknya, KF berada dalam mode prediksi penuh, menghasilkan posisi bola yang diestimasi berdasarkan model gerakan internalnya. Hal ini mengurangi beban komputasi secara signifikan.
3. Pengukuran Ulang YOLO (*frame* 6): Pada *frame* setelah pengukuran terakhir, YOLOv8 diaktifkan kembali untuk memberikan data pengukuran baru ( $Z_k$ ) yang akan digunakan KF untuk pembaruan (*update*). Siklus ini kemudian berulang.

Strategi ini juga meningkatkan ketahanan sistem terhadap *occlusion* sesaat. Jika YOLO gagal mendeteksi bola pada *frame* yang seharusnya (misalnya *frame* 1 atau *frame* 6), KF akan diinstruksikan untuk tetap menjalankan langkah prediksi tanpa *update* dari YOLO. Untuk menjaga integritas pelacakan, batas toleransi kegagalan deteksi diatur maksimal 5 kali kejadian berturut-turut. Jika KF harus terus memprediksi lebih dari 5 *frame* tanpa input YOLO (misalnya karena bola hilang atau terhalang total), *track* (jejak) dianggap hilang (*Lost Track*), dan sistem akan kembali ke mode pencarian penuh.

Implementasi KF dilakukan menggunakan modul bawaan OpenCV, yang menyediakan fungsi-fungsi terstruktur untuk inisialisasi, prediksi, dan pembaruan, memanfaatkan operasi matriks yang dioptimalkan untuk performa *real-time*. *Output* status  $(x, y)$  yang dihaluskan dan diprediksi dari KF kemudian digunakan oleh modul kontrol robot untuk menghasilkan perintah gerakan yang stabil dan responsif.

### **3.6. Desain dan Implementasi Kontrol Gerak Robot Berbasis Penglihatan**

Perancangan sistem menggunakan pendekatan pembagian fungsi antara dua platform, yaitu:

1. Unit Pemrosesan Visi (ROS): Laptop pada robot menjalankan ROS yang berfungsi membaca citra dari kamera, melakukan deteksi objek menggunakan YOLO dan *Kalman Filter*, menentukan lokasi objek pada *grid*, dan menghasilkan perintah gerak diskrit.
2. Unit Eksekusi Gerak (Arduino Mega): Berfungsi menerima perintah gerak dari ROS melalui komunikasi serial, menerjemahkannya menjadi aksi motor, dan menjalankan manuver sesuai perintah.

Alur data utama pada sistem meliputi: kamera → deteksi YOLO → prediksi posisi Kalman → identifikasi *grid* → perintah gerak → pengiriman melalui topik “master/command” → eksekusi pada Arduino. Pendekatan distribusi seperti ini memastikan proses pengolahan citra yang berat dilakukan di ROS, sedangkan Arduino hanya menjalankan fungsi aktuasi motor.

Informasi posisi objek dikonversi menjadi instruksi gerak melalui pembagian area pengamatan menjadi *grid*. Pembagian *grid* ditentukan berdasarkan hasil kalibrasi kamera sehingga setiap kotak merepresentasikan area tertentu pada bidang pandang.

Objek yang terdeteksi (misalnya bola) akan memiliki koordinat *grid* tertentu, misalnya ( $X_{grid}, Y_{grid}$ ). Koordinat tersebut dikategorikan ke dalam beberapa zona tindakan, seperti zona kiri, zona kanan, zona tengah, atau zona jauh. Setiap zona dipetakan ke instruksi gerak tertentu, antara lain:

- Zona tengah atas → robot bergerak maju.
- Zona kiri → robot melakukan koreksi arah ke kiri.
- Zona kanan → robot melakukan koreksi ke kanan.
- Zona bawah → robot melakukan putaran pencarian.
- Objek tidak terdeteksi → robot berhenti.

Pendekatan berbasis *grid* ini membuat sistem mudah diimplementasikan karena instruksi gerak ditentukan oleh lokasi visual secara langsung.

Komunikasi antara ROS dan Arduino dilakukan melalui Serial dengan format pesan berbasis *string*, misalnya “*maju*”, “*putarKanan*”, atau “*majuPelan*”. Node ROS menerbitkan perintah ini ke topik “*master/command*”, kemudian dikirimkan ke Arduino. Arduino berfungsi membaca pesan yang diterima, melakukan *parsing*, dan memanggil fungsi gerak yang sesuai. Setiap perintah memiliki fungsi motor tersendiri, misalnya fungsi untuk bergerak maju, berbelok, atau bergerak dengan kecepatan rendah. Implementasi ini memungkinkan robot bergerak secara responsif berdasarkan keluaran pengolahan citra pada ROS.

## DAFTAR PUSTAKA

- Admaja, Y. P. (2021). *Sistem Penghitung Jumlah Pengunjung di Restoran Menggunakan Kamera Berbasis*. 3(1), 19–26.
- Baskoro, G. Y., Afrisal, H., & Sofwan, A. (2022). Perancangan Sistem Deteksi Objek Berbasis Convolutional Neural Network Menggunakan Yolov4 Dan Opencv. *Transient: Jurnal Ilmiah Teknik Elektro*, 11(4), 128–020637. <https://ejournal3.undip.ac.id/index.php/transient>
- Egi, Y. (2022). *Basketball self training shooting posture recognition and trajectory estimation using computer vision and Kalman filter*. 73, 19–27.
- Farhan, T. M., & Candra, F. (2025). CNN-Based Ball and Goal Detection for KRSBI Robot with Omnidirectional Camera. *International Journal of Electrical, Energy and Power System Engineering*, 8(2), 86–98. <https://doi.org/10.31258/ijepse.8.2.1-13>
- Firdaus, A. Z., & Lelono, D. (2025). *Sistem Klasifikasi Sampah Otomatis Berbasis Deteksi Objek Real-Time Pada Single Board Computer Dengan Algoritma YOLO*. 15(1), 49–60. <https://doi.org/10.22146/ijeis.104520>
- Hendrik, B., & Awal, H. (2023). Pengenalan Teknologi Robot Pada Anak Sekolah Dasar. *Jurmas Bangsa*, 1(1), 46–52. <https://doi.org/10.62357/jpb.v1i1.140>
- Jung, H., Kang, S., Kim, T., Kim, H., Klemove, H. L., & Korea, R. (2024). ConfTrack : Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box. *CVF Open Access*, 6583–6592.
- Kusumoputro, B., Purnomo, M. H., Rochardjo, H. S. B., Prabowo, G., Purwanto, D., Mozef, E., Indrawanto, Mutijarsa, K., & Muis, A. (2024). *Pedoman Kontes Robot Indonesia (Kri) Pendidikan Tinggi Tahun 2024*.
- Miharja, G. P., Nugraha, D. A., & Aziz, A. (2025). Analisis Perbandingan Kinerja YOLO dan Camshift Dalam Pelacakan Objek Berbasis Video. *Jurnal Riset Mahasiswa Bidang Teknologi Informasi Volume*, 5(Analisis Perbandingan Kinerja Yolo dan Camshift), 100–110.
- Mohammed, H. R., & Hussain, Z. M. (2021). *Detection and Recognition of Moving Video Objects: Kalman Filtering with Deep Learning*. 12, 154–157.

- Nanda, B. M., Siregar, S., & Sani, M. I. (2023). *Implementasi Object Detection Pada Robot Sepak Bola Beroda Berbasis Kamera Omnidirectional Menggunakan OpenCV*. 9(4), 2064–2068.
- NumPy. (2025). *NumPy*. <https://numpy.org/>
- Nuralim, J., Fath, N., Musafa, A., Sujono, & Broto, D. S. (2022). Perancangan Sistem Pendekripsi Obyek Bola Dengan Metode Framework YOLOv4. *Jurnal Maestro*, 5(2), 289–294. <https://jom.ft.budiluhur.ac.id/index.php/maestro/article/view/531>
- Ratna, S. (2020). Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm. *Technologia: Jurnal Ilmiah*, 11(3), 181. <https://doi.org/10.31602/tji.v11i3.3294>
- Ritonga, A. A., & Hasibuan, E. R. (2025). *Eksplorasi Masa Depan Robotika : Integrasi Kecerdasan Buatan dalam Sistem Automasi Adaptif Berbasis Lingkungan*. 1, 387–393.
- Ryu, S. E., & Chung, K. Y. (2021). Detection model of occluded object based on yolo using hard-example mining and augmentation policy optimization. *Applied Sciences (Switzerland)*, 11(15). <https://doi.org/10.3390/app11157093>
- Saputra, C. (2023). Implementasi Algoritma SIFT (Scale-Invariant Feature Transform) Dan Algoritma Kalman Filter Dalam Mendekripsi Objek Bola. *Jurnal PROCESSOR*, 18(1), 73–82. <https://doi.org/10.33998/processor.2023.18.1.791>
- Saputra, F. B., Kallista, M., & Setianingsih, C. (2023). Deteksi social distancing dan penggunaan masker di restoran menggunakan algoritma Residual Network (ResNet). *EProceedings of Engineering*, 10(1), 284–295.
- Shofi, M., Basuki, B. M., & Habibi, A. (2023). Rancang Bangun Penendang Bola Pada Robot Soccer Unisma Menggunakan Solenoid. *Science Electro*, 16(4), 1–7.
- Sholehurrohman, R., Habibi, M. R., Ilman, I. S., Taufiq, R., & Muhaqiqin, M. (2023). Analisis Metode Kalman Filter, Particle Filter dan Correlation Filter Untuk Pelacakan Objek. *Komputika : Jurnal Sistem Komputer*, 12(2), 21–28. <https://doi.org/10.34010/komputika.v12i2.9567>

- Surya, M., Nehemia Toscany, A., Saputra, C., Pratama, Y., & Bustami, M. I. (2025). Penggunaan Yolo Untuk Deteksi Robot Dan Gawang Pada Robot Sepak Bola Beroda. *The Indonesian Journal of Computer Science*, 14(1), 1055–1070. <https://doi.org/10.33022/ijcs.v14i1.4575>
- Takura, K. E. I., Arita, Y. U. M. A. N., & Oaki, S. H. N. (2021). *Automatic pear and apple detection by videos using deep learning and a Kalman filter*. 4(5), 1688–1695.
- Taylor, L. E., Mirdanies, M., & Saputra, R. P. (2016). *OPTIMIZED OBJECT TRACKING TECHNIQUE USING KALMAN*. 07, 57–66. <https://doi.org/10.14203/j.mev.2016.v7.57-66>
- Ultralytics. (2025). *Kalman Filter (KF)*.
- Widodo, Y. B., Sibuea, S., & Narji, M. (2024). Kecerdasan Buatan dalam Pendidikan: Meningkatkan Pembelajaran Personalisasi. *Jurnal Teknologi Informatika Dan Komputer*, 10(2), 602–615. <https://doi.org/10.37012/jtik.v10i2.2324>
- Wijaya, G. F., & Yuniarto, D. (2024). Tinjauan Penerapan Machine Learning pada Sistem Rekomendasi Menggunakan Model Klasifikasi. *Populer: Jurnal Penelitian Mahasiswa*, 3(4), 144–153. <https://doi.org/10.58192/populer.v3i4.2798>
- Yuztiawan, F. R., & Utaminingrum, F. (2017). Implementasi Metode Kalman Filter Dan Model Yolov8N Untuk Fitur Human-Following Pada Kursi Roda Pintar. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(1), 2548–2964.