



PROPOSAL

OPTIMASI GERAKAN ROBOT SEPAK BOLA MENGUNAKAN KALMAN FILTER PADA DETEKSI BOLA BERBASIS YOLO

Oleh:

FIKRI RIVANDI

2207112583

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

DAFTAR ISI

COVER	i
HALAMAN JUDUL	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL	v
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Tujuan Penelitian	5
1.4. Batasan Masalah.....	5
1.5. Manfaat Penelitian	6
1.6. Sistematika Penulisan	6
BAB II LANDASAN TEORI	8
2.1. Penelitian Terdahulu	8
2.2. Kompetisi Robot Sepak Bola Beroda Indonesia.....	17
2.3. Robot Sepak Bola Beroda	17
2.4. Bola dan Gawang	18
2.5. OpenCV	18
2.6. <i>Machine Learning</i>	19
2.7. <i>Deep Learning</i>	19
2.8. <i>Convolutional Neural Network</i>	19
4.8.1. <i>Convolution Layer</i>	20
4.8.2. <i>Pooling Layer</i>	22

4.8.3.	<i>Fully Connected Layer</i>	23
2.9.	<i>You Only Look Once (YOLO)</i>	23
3.9.1.	<i>Original YOLO</i>	25
3.9.2.	<i>YOLO-v2/9000</i>	25
3.9.3.	<i>YOLO-v3</i>	26
3.9.4.	<i>YOLO-v4</i>	26
3.9.5.	<i>YOLO-v5</i>	26
3.9.6.	<i>YOLO-v6</i>	27
3.9.7.	<i>YOLO-v7</i>	27
3.9.8.	<i>YOLO-v8</i>	27
2.10.	<i>Labelling</i>	28
2.11.	<i>Confusion Matrix</i>	29
BAB III METODOLOGI PENELITIAN		31
3.1.	<i>Kerangka Pikiran</i>	31
3.2.	<i>Studi Literatur</i>	31
3.3.	<i>Identifikasi Masalah</i>	32
3.4.	<i>Pengumpulan Data Objek Bola dan Gawang</i>	32
3.5.	<i>Preprocessing Data</i>	33
3.6.	<i>Pelatihan</i>	34
3.7.	<i>Pengujian Data Uji</i>	34
DAFTAR PUSTAKA		36

DAFTAR GAMBAR

Gambar 2.2. Cara Kerja <i>Convolutional Neural Network</i> (Faiz et al., 2020).....	20
Gambar 2.3. Proses pada <i>convolution layer</i> (Sarkar, 2022).....	21
Gambar 2.3. Contoh proses <i>zeropadding</i>	21
Gambar 2.4. Contoh proses <i>stride</i>	22
Gambar 2.5. Contoh operasi pada <i>pooling layer</i> (Rao et al., 2020).....	22
Gambar 2.6. Arsitektur sederhana YOLO (Carolina & Lina, 2023).....	24
Gambar 2.7. Arsitektur YOLOv1 (Hussain, 2023).....	25
Gambar 2.8. Perbandingan YOLO-v8 (Hussain, 2023).....	28
Gambar 2.4. Proses <i>labelling</i>	29
Gambar 3.1. Kerangka Pikiran Penelitian.....	31
Gambar 3.3. Contoh gambar gawang menggunakan kamera <i>omnidirectional</i>	33
Gambar 3.4. Proses <i>labelling</i> menggunakan <i>labelImg</i>	34

DAFTAR TABEL

Tabel 2.1. Penelitian terdahulu.....	12
Tabel 2.2. Contoh tabel <i>Confusion Matrix</i>	29
Tabel 3.1 <i>Confusion Matrix</i>	35

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 kecerdasan buatan (*Artificial Intelligence/AI*) dan pembelajaran mesin (*machine learning*). Teknologi ini memungkinkan sistem komputer tidak hanya melakukan perhitungan, tetapi juga belajar dari data, mengenali pola, serta mengambil keputusan secara mandiri.

Salah satu cabang dari AI yang banyak dimanfaatkan adalah *computer vision*, yaitu teknologi yang memungkinkan komputer menangkap gambar, memahami dan menginterpretasikan informasi visual dari dunia nyata. Perkembangan *deep learning*, khususnya melalui *Convolutional Neural Network* (CNN), telah meningkatkan kemampuan sistem dalam melakukan deteksi objek secara akurat, bahkan dalam kondisi lingkungan yang kompleks.

Kemajuan teknologi ini membuka peluang besar dalam pengembangan robotika, yang merupakan bidang ilmu interdisipliner melibatkan elektronika, mekanika, dan ilmu komputer. Robotika modern tidak lagi terbatas pada sekadar otomasi sederhana, melainkan sudah mampu mengambil keputusan berbasis data visual melalui integrasi AI. 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*. Namun, kelemahan utama metode ini adalah ketergantungannya pada kondisi pencahayaan, sehingga akurasi deteksi menjadi tidak stabil dalam situasi lapangan yang dinamis.

Seiring perkembangan teknologi, metode *deep learning*, khususnya YOLO (*You Only Look Once*), mulai diterapkan untuk meningkatkan akurasi deteksi objek. YOLO merupakan algoritma deteksi objek berbasis *single-shot detection*, di mana proses deteksi dilakukan hanya dalam sekali pengolahan citra. Berbeda dengan metode konvensional yang membutuhkan dua tahap (*region proposal* dan klasifikasi), YOLO secara langsung membagi gambar ke dalam grid dan memprediksi lokasi serta kelas objek dalam satu proses komputasi. Hal ini membuat YOLO mampu mendeteksi objek secara *real-time* dengan akurasi tinggi.

Keunggulan lain dari YOLO adalah kemampuannya mendeteksi beberapa objek sekaligus dalam sebuah *frame*, bahkan ketika objek mengalami variasi seperti rotasi, skala, ataupun kondisi pencahayaan yang berubah-ubah. Hal ini menjadikan YOLO lebih *robust* dibandingkan metode berbasis warna seperti HSV, yang sangat bergantung pada kestabilan pencahayaan.

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 merespons posisi bola.

Beberapa penelitian sebelumnya menunjukkan 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).

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. Dengan sifatnya yang iteratif dan berbasis probabilitas, Kalman Filter dapat memperhalus data pengukuran yang memiliki *noise* serta memperkirakan posisi objek secara lebih akurat.

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. Hal ini menjadikannya solusi yang seimbang antara kebutuhan akurasi dan keterbatasan perangkat keras.

Algoritma Kalman Filter terbukti efisien dalam estimasi posisi objek target yang hilang saat pendeteksian secara *real-time*. Implementasi metode Kalman Filter, memiliki persentase akurasi 96,89% untuk tugas *tracking* objek pada video lalu lintas dan sirkuit Naskar, dimana metode tersebut lebih baik dalam segi akurasi dibanding kedua kompetitornya yaitu Partikel Filter dan Korelasi Filter (Sholehurrohman et al., 2023).

Berdasarkan permasalahan yang telah diuraikan, 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.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam peningkatan performa robot KRSBI Beroda, khususnya pada aspek persepsi visual dan perencanaan gerakan, sekaligus menjadi dasar bagi pengembangan metode deteksi dan prediksi berbasis kecerdasan buatan di bidang robotika kompetitif.

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 respons 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.
3. Hasil penelitian ini dapat menjadi referensi bagi mahasiswa dan peneliti lain yang ingin mengembangkan metode deteksi objek berbasis *deep learning* dan optimasi pergerakan menggunakan algoritma prediksi.
4. Penelitian ini berpotensi memberikan kontribusi dalam pengembangan sistem persepsi visual dan perencanaan gerakan yang dapat diadaptasi pada berbagai aplikasi lain, seperti kendaraan otonom, sistem pengawasan, maupun robot industri.

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

Perancangan Sistem Deteksi Bola dan Gawang dengan Algoritma *Convolutional Neural Network* pada Robot KRSBI Menggunakan Kamera *Omnidirectional* didukung oleh beberapa referensi yang merupakan penelitian sebelumnya. Referensi-referensi tersebut memberikan kontribusi yang sangat bernilai dalam memberikan pedoman dan masukan bagi penelitian ini. Analisis dari penelitian terdahulu membuat penulis mendapat wawasan lebih tentang hal-hal yang relevan, temuan-temuan menarik, dan Melalui analisis dari penelitian terdahulu, penulis dapat memperoleh wawasan yang mendalam tentang isu-isu yang relevan, temuan-temuan penting, dan pendekatan yang telah diuji sebelumnya. Dengan mempertimbangkan informasi dan pengetahuan yang telah dikumpulkan melalui referensi-referensi tersebut, penelitian ini dapat mengambil langkah yang lebih terinformasi, memperluas pemahaman, dan memperkaya hasil penelitian yang diharapkan.

Pertama, yaitu penelitian berjudul “Implementasi *Object Detection* pada Robot Sepak Bola Beroda Berbasis Kamera *Omnidirectional* Menggunakan *Opencv*” yang dilakukan oleh Bagas Musamma Nanda, Simon Siregar, dan Muhammmad Ikhsan Sani (2023). Masalah yang diangkat penelitian ini yaitu sistem deteksi pada robot KRSBI tersebut masih dipengaruhi oleh intensitas cahaya objek yang dideteksi, sehingga jika ada perubahan intensitas cahaya dari suatu objek, maka hasil deteksinya menjadi tidak efektif. Metode yang digunakan pada penelitian ini adalah *object detection* dari kamera dengan *OpenCV library*. Pengujian dilakukan dengan cara mengukur intensitas cahaya di lapangan. Setelah pengujian, didapatlah hasil bahwa robot KRSBI dapat mendeteksi objek bola, gawang, robot *cyan* dan *magenta* dengan akurasi 70% dan dengan intensitas cahaya berbeda tiap objeknya. Penelitian ini dapat membuktikan bahwa *object detection* bisa diterapkan pada robot KRSBI dengan kamera *omnidirectional*.

Penelitian kedua dengan judul “Sistem Deteksi Bola pada Robot Kiper Pemain Sepak Bola Beroda” karya Rifqi Amalya Fatekha, Bima Sena Bayu Dewantara, dan Hary Oktavianto pada tahun 2021. Masalah pada penelitian ini adalah robot yang pergerakannya terbatas sehingga tidak dapat memainkan bola dengan baik. Metode yang dipilih pada penelitian ini adalah metode deteksi warna dan objek dengan melakukan eliminasi objek-objek yang tidak diperlukan saat menangkap gambar sehingga hanya tersisa objek penting seperti robot dan bola. Proses evaluasi yang dilakukan adalah dengan membandingkan 2 metode deteksi objek yaitu *HoughCircle* dan *approxPolyDP*. Hasilnya deteksi bola menggunakan metode *approxPolyDP* memiliki kecepatan yang baik bahkan 15.8ms lebih cepat dari metode *HoughCircle*. Penelitian ini membuktikan bahwa robot bisa mendeteksi objek bola dengan deteksi warna dan bentuk.

Penelitian ketiga berjudul “Rancang Bangun *Object Detection* Pada Robot Soccer Menggunakan Metode *Single Shot Multibox Detector* (SSD MobileNetV2)” yang dilakukan oleh Cokorda Gde Wahyu Permana, Duman Carev Khrisne, dan Nyoman Putra Sastra (2021). Penelitian ini mengangkat masalah tidak efisiennya deteksi objek robot yang menggunakan *color filtering* (HSV) karena bergantung pada kondisi lingkungan yang berubah-ubah. Untuk mengatasi masalah tersebut, dipilihlah algoritma *Artificial Neural Network* (ANN) dengan metode *Single Shot Multibox Detector* (SSD MobileNet V2) dengan harapan model deteksi yang dihasilkan memberikan kecepatan dan keakuratan yang lebih baik. Hasilnya robot dapat mendeteksi objek sengan jarak tertentu dengan baik bahkan jika objek tersebut sedikit dihalangi oleh objek lainnya.

Penelitian keempat berjudul “Perancangan Sistem Deteksi Objek Bola dan Gawang pada Robot Sepakbola Menggunakan Metode *Darknet YOLO*.” Penelitian ini dilakukan oleh dua orang yaitu Dwi Novianto Nugroho dan Lilik Anifah (2023). Pada penelitian ini, terdapat sebuah robot bola humanoid yang menggunakan *webcam* diharapkan dapat mendeteksi bola dan gawang dengan akurasi yang sangat tinggi. Sehingga dibutuhkan sebuah metode deteksi yang menggunakan ilmu *computer vision*. Selanjutnya, dibuatlah sebuah model deteksi bola dan gawang menggunakan *Darknet YOLO* yang dikenal memiliki kecepatan serta akurasi yang

tinggi. Hasilnya, didapat akurasi yang tinggi dari deteksi bola tersebut dengan tingkat *accuracy* sebesar 0.97, *precision* sebesar 0.99, *recall* sebesar 0.96, dan F1-*score* sebesar 0.97.

Penelitian kelima dengan judul "Perancangan Sistem Pendeteksian Obyek Bola dengan Metode Framework YOLO V4" dibuat oleh Jalu Nuralim, Nifty Fath, Akhmad Musafa, dan Drs. Suwandi Broto (2022). Pada penelitian ini diketahui bahwa dalam Kontes Robot Sepakbola Indonesia Beroda tahun 2018 mengharuskan robot untuk dapat bernavigasi dan melakukan tugasnya, yakni menemukan, menggiring, dan menendang bola ke arah gawang, sehingga perlu respon yang cepat agar robot dapat menuju gawang lawan. Penelitian ini menggunakan metode YOLOv4 dan akan diuji dengan metode *Confusion Matrix*. Hasilnya, robot bisa membaca objek bola dengan dihalangi penghalang yang berpersentase 50%, 60%, dan 70 %. Sedangkan robot sudah tidak bisa mendeteksi bola yang terhalang dengan persentase halangan 80%, 90%, dan 100%.

Penelitian berikutnya berjudul "Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19." Penelitian ini dibuat oleh tiga orang yaitu Faizal Indaryanto, Anan Nugroho, dan Alfa Faridh Suni. Penelitian ini muncul karena banyaknya masyarakat yang tidak menaati protokol *social distancing* padahal saat itu sedang masa wabah Covid-19. Program *social distancing detector* pun dibuat menggunakan algoritma YOLOv3. Hasilnya, sistem dapat mendeteksi objek manusia dengan tingkat akurasi diatas 90%.

Penelitian ketujuh berjudul " Sistem Pengenalan Wajah Menggunakan Metode YOLO." Penelitian ini dilakukan oleh Muhammad Syarif Hidayatulloh. Penelitian ini dilakukan karena kemajuan teknologi yang sangat masif sehingga muncul metode identifikasi yang mutakhir seperti deteksi wajah. Penelitian ini akan dilakukan dengan metode YOLO untuk deteksinya. Dari pengujian yang dilakukan dengan jarak yang berbeda-beda, sistem dapat mendeteksi wajah dengan akurasi 100%.

Penelitian kedelapan berjudul "Pendeteksi Senjata Berbahaya pada Percobaan Tindakan Kriminal Dengan Menggunakan Metode YOLO (You Only Look Once)". Penelitian ini dibuat oleh Ade Luthfi pada tahun 2021. Kasus yang

diangkat pada penelitian ini adalah tidak efektifnya CCTV dalam mencegah kejahatan yang meningkat pada era Covid-19. Penelitian ini dibuat menggunakan metode YOLO dan menghasilkan hasil deteksi yang memiliki tingkat akurasi sebesar 66% pada *webcam*.

Penelitian kesembilan berjudul "Implementasi Algoritma *You Only Look Once* (YOLO) Untuk Deteksi Korban Bencana Alam." Penelitian ini diterbitkan pada tahun 2021 yang dibuat oleh Moehammad Sarosa dan Nailul Muna. Penelitian ini muncul untuk membantu penanganan evakuasi korban bencana alam agar korban tersebut lebih cepat ditemukan. Metode deteksi yang digunakan dalam penelitian ini adalah YOLOv3. Setelah diuji dengan 100 data *train* dan 100 data *validation*, didapatlah hasil deteksi korban dengan *F1 Score* mencapai 95,3%.

Penelitian kesepuluh dengan judul "Implementasi Algoritma YOLO (*You Only Look Once*) Untuk Deteksi Alat Rias Nusantara" dibuat oleh Zuanita Syifaul Jannah dan Felix Andreas Sutanto. Penelitian ini dipublikasikan pada tahun 2022. Masalah yang diangkat pada penelitian ini adalah banyaknya ragam rias adat di Indonesia . Untuk membantu membedakan masing masing rias adat tersebut, dilibatkanlah teknologi masa kini yakni *machine learning*. Algoritma yang digunakan dalam penelitian ini adalah YOLOv4-Tiny. Setelah melatih 1478 citra dataset yang terdiri dari 14 jenis rias adat nusantara, didapatkannya hasil deteksi dengan nilai akurasi rata-rata sebesar 95,20%.

Tabel 2.1. Penelitian terdahulu

No	Peneliti	Tahun	Judul	Masalah	Metode	Hasil	Perbedaan
1	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 <i>Opencv</i>	Sistem deteksi bergantung pada intensitas cahaya yang membuat hasil deteksi tidak efektif	<i>Object Detection</i> dengan <i>library OpenCV</i>	Hasil akurasi deteksi objek bola, gawang, robot <i>magenta</i> dan robot <i>cyan</i> mencapai 70%	Menggunakan <i>Object Detection</i> berbasis warna
2	Rifqi Amalya Fatekha, Bima Sena Bayu Dewantara, dan Hary Oktavianto	2021	Sistem Deteksi Bola pada Robot Kiper Pemain Sepak Bola Beroda	Pergerakan robot terbatas sehingga tidak dapat memainkan bola dengan baik	Deteksi warna dan objek dengan melakukan eliminasi serta 2 sub metode deteksi objek yaitu <i>HoughCircle</i> dan <i>approxPolyDP</i>	Deteksi warna dan objek dengan sub metode <i>approxPolyDP</i> menghasilkan kecepatan yang baik	Menggunakan metode deteksi objek <i>HoughCircle</i> dan <i>approxPolyDP</i>
3	Cokorda Gde Wahyu Permana, Carev Khrisne, dan	2021	Rancang Bangun <i>Object Detection</i> Pada Robot <i>Soccer</i> Menggunakan Metode <i>Single Shot</i>	Robot dengan deteksi yang menggunakan <i>color filtering</i> (HSV) tidak efisien karena	<i>Artificial Neural Network</i> (ANN) dengan	Robot dapat mendeteksi objek dengan jarak idela tertentu bahkan jika	Menggunakan metode SSD MobilenetV2

	Nyoman Putra Sastra		<i>Multibox Detector</i> (SSD MobileNetV2)	sangat tergantung lingkungan yang bisa berubah-ubah.	metode <i>Single Shot Multibox Detector</i> (SSD MobileNetV2)	objek dihalangi oleh objek lainnya	
4	Dwi Novianto Nugroho, Lilik Anifah	2023	Perancangan Sistem Deteksi Objek Bola dan Gawang pada Robot Sepakbola Menggunakan Metode <i>Darknet</i> YOLO	Sebuah robot sepakbola humanoid diharapkan dapat mendeteksi objek bola dan gawang dengan sangat akurat. Hal ini dibutuhkan supaya robot dapat menggiring bola, mengoper bola, dan mencetak gol tanpa masalah	<i>Darknet</i> YOLO dengan jaringan YOLOv3, serta pelatihannya menggunakan <i>framework Darknet</i> .	Didapat hasil nilai <i>accuracy</i> 0.97, <i>precision</i> sebesar 0.99, <i>recall</i> sebesar 0.96, dan <i>F1-score</i> sebesar 0.97.	Menggunakan <i>framework Darknet</i> . Robot yang digunakan adalah robot bola humanoid, dan kameranya menggunakan <i>webcam</i>
5	Jalu Nuralim, Nifty Fath, Akhmad Musafa, Sujono, Drs. Suwasti Broto	2022	Perancangan Sistem Pendeteksian Obyek Bola dengan Metode <i>Framework</i> YOLO V4	Pada KRSBI 2018 yang diselenggarakan DIKTI, robot bola harus bisa bernavigasi dan mengontrol bola kearah gawang. Karenanya, dibutuhkan respon	Menggunakan metode YOLOv4. Pengujian menggunakan <i>Confusion Matrix</i> .	Pengujian dengan halangan dilakukan dengan berbagai tingkatan. Dengan halangan 50%, 60%, dan 70% sistem dapat mendeteksi objek bola yang terhalang objek lainnya.	Menggunakan YOLOv4, objek yang di deteksi hanya bola.

				yang cepat supaya robot bisa mengetahui posisi gawang lawan		Sedangkan dengan persen halangan 80%, 90%, dan 100% sistem tidak dapat mendeteksi objek bola yang terhalang.	
6	Faizal Indaryanto, Anan Nugroho, Alfa Faridh Suni	2021	Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19	Salah satu cara untuk mencegah penyebaran Covid-19 adalah <i>social distancing</i> , namun masyarakat masih banyak yang lalai dan tidak menaati protokol kesehatan tersebut. Untuk mengatasi hal tersebut, dibuatlah sebuah aplikasi <i>social distancing detector</i> menggunakan library YOLOv3.	Algoritma YOLOv3	Berdasarkan pengujian menggunakan gambar <i>testing</i> , sistem dapat mendeteksi orang dengan tingkat akurasi rata-rata sebesar 90,04%	Objek yang dideteksi adalah manusia
7	Muhammad Syarif Hidayatulloh	2021	Sistem Pengenalan Wajah Menggunakan Metode YOLO	Perkembangan teknologi yang masif membuat munculnya cara	Algoritma YOLO	Pengujian dilakukan dengan jarak berbeda-beda,	Objek yang di deteksi adalah wajah

				baru untuk absensi, salah satunya pendeteksi wajah.		menghasilkan akurasi 100%	
8	Ade Luthfi	2021	Pendeteksi Senjata Berbahaya pada Percobaan Tindakan Kriminal Dengan Menggunakan Metode YOLO (<i>You Only Look Once</i>)	Angka kriminalitas meningkat selama era Covid-19. Namun, keamanan seperti CCTV masih belum efektif untuk mengatasi tingkat kejahatan.	Algoritma YOLO	Nilai akurasi pada <i>webcam</i> sebesar 66%, sedangkan pada video hasil rekaman CCTV sebesar 56%	Objek yang dideteksi adalah senjata yang berpotensi untuk terjadinya tindakan kriminal.
9	Moehammad Sarosa, Nailul Muna	2021	Implementasi Algoritma <i>You Only Look Once</i> (Yolo) Untuk Deteksi Korban Bencana Alam	Bencana alam merupakan suatu kejadian yang dapat menimbulkan kerusakan. Namun, bencana alam tidak bisa diprediksi tempat dan waktunya yang mengakibatkan berpotensi jatuh korban yang banyak.	Algoritma YOLOv4	Didapatkan <i>F1 Score</i> mencapai 95,3% menggunakan YOLOv3 dengan 100 data <i>train</i> dan 100 data <i>validation</i>	Objek yang dideteksi adalah korban bencana alam
10	Zuanita Syaiful Jannah, Felix	2022	Implementasi Algoritma YOLO	Tata rias dan busana Indonesia memiliki makna	Algoritma YOLOv4	Aplikasi yang dibangun dapat	Objek yang dideteksi

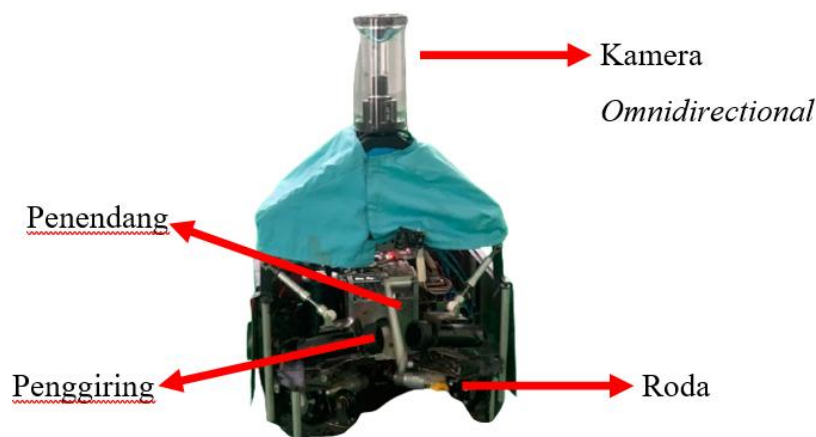
	Andreas Sutanto		(<i>You Only Look Once</i>) Untuk Deteksi Rias Adat Nusantara	dan lambang tertentu di tiap daerah dan kepercayaan masing-masing. Dengan kemajuan teknologi, jenis rias tersebut dapat dibedakan dengan bantuan deteksi objek menggunakan <i>machine learning</i>		mendeteksi jenis rias dengan akurasi rata-rata 95,20% dalam rata-rata waktu deteksi selama 327ms (<i>milli-seconds</i>)	adalah rias adat nusantara
--	--------------------	--	---	--	--	---	-------------------------------

2.2. Kompetisi Robot Sepak Bola Beroda Indonesia

Merujuk ke jurnal yang berjudul "Sistem Pergerakan Robot Kiper Beroda Menggunakan Metode Wall Follower Berbasis *Image Processing*," Kompetisi Robot Sepak Bola Beroda Indonesia (KRSBI) adalah kompetisi robot yang melakukan permainan bola dan dibagi menjadi tiga peran yaitu robot kiper, robot pertahanan, dan robot penyerang. (Risfendra et al., 2020)

Kontes Robot Sepakbola Beroda Indonesia diselenggarakan dengan tujuan untuk meningkatkan pengetahuan dan kreativitas mahasiswa dalam bidang robotika. Dalam kontes ini, mahasiswa diharapkan untuk mengembangkan keterampilan dalam berbagai bidang, termasuk mekanika, manufaktur, elektronika, pemrograman, kecerdasan buatan, pengolahan citra, komunikasi digital, dan strategi. Selain itu, kontes ini juga mendorong pengembangan sikap disiplin, toleransi, sportivitas, kerjasama, saling menghargai, kontrol emosi, dan berbagai keterampilan softskill lainnya. (Kusumoputro et al., 2024)

2.3. Robot Sepak Bola Beroda



Gambar 2.1. Robot sepak bola beroda

Untuk robot yang dikompertisikan di Kontes Robot Sepak Bola Indonesia Beroda bisa dilihat pada gambar 3.1. Tampak sebuah robot berbentuk kubus yang memiliki sepasang roda di depan yang merupakan penggiring. Sedangkan dibagian atas terdapat kamera *Omnidirectional* yang berguna untuk mendeteksi objek.

Kamera *Omnidirectional* adalah sebuah konsep penggunaan kamera untuk mengambil citra atau gambar yang menampilkan berbagai arah, baik depan, belakang, kiri, dan kanan. Pada gambar 3.2. bisa dilihat bahwa kamera *omnidirectional* menggunakan teknik mengarahkan kamera ke cermin cembung yang dihadapkan ke bawah atau permukaan lantai sehingga gambar yang ditangkap oleh kamera adalah pantulan dari cermin cembung tersebut. (Darmawan, 2018)

Berikutnya ada penendang yang menggunakan *kicker* berbasis solenoid yang mana daya listrik akan dihantarkan kepada kumparan kawat yang mengelilingi sebuah batang logam lalu akan melepaskan gaya dorong yang kuat ke arah yang ditentukan.

Untuk penggiring berupa roda kecil yang diberi motor lalu berputar ke arah dalam robot sehingga roda ini akan menangkap dan menjaga bola saat robot bergerak agar bola tidak terlepas dari robot.

Terakhir ada roda yang berfungsi agar robot bisa bergerak. Sebuah robot memiliki 3 roda agar gerak menjadi fleksibel. Roda ini terdiri dari dua bagian. Yang pertama ada motor dengan kecepatan putar 500rpm untuk menggerakkan robot. Kedua ada *omni wheel* yang berfungsi agar robot bisa bergerak ke semua arah

2.4. Bola dan Gawang

Dalam matematika, khususnya geometri, bola didefinisikan sebagai himpunan semua titik yang berjarak sama dari suatu titik tetap yang disebut pusat bola.

Sedangkan gawang adalah sebuah rintangan yang dibuat oleh manusia ataupun alamiah. Dalam pertandingan sepakbola, gawang menjadi sarana yang amat penting untuk menentukan gol atau tidaknya bola yang ditendang ke arah gawang. Gawang akan menjadi penentu golnya bola saat bola melewati garis gawang yang ditarik dari dua tiang gawang tersebut. (Ardi Gabelatama, 2018)

2.5. OpenCV

Open Source Computer Vision Library atau yang biasa disebut OpenCV merupakan pustaka perangkat lunak yang bertujuan untuk mengolah citra secara

real-time. Pustaka ini awalnya dibuat oleh Intel, hingga sekarang didukung oleh Itseez dan Willow Garage. (Pratama et al., 2022)

Pustaka OpenCV ini dirancang dengan sangat kuat dan fleksibel untuk menangani berbagai masalah dalam penglihatan komputer yang solusinya sudah tersedia, seperti pemotongan citra, peningkatan kualitas citra dengan mengatur kecerahan, ketajaman, dan kontras, serta melakukan deteksi bentuk, segmentasi citra, deteksi objek bergerak, pengenalan objek, dan banyak lagi. (Ratna, 2020)

2.6. Machine Learning

Machine Learning adalah salah satu pengaplikasian dari *Artificial Intelligent* (AI) yang fokus kepada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus diprogram berulang kali. *Machine Learning* membutuhkan sebuah data sebagai proses *learning* sebelum menghasilkan sebuah model. Singkatnya, *Machine Learning* adalah pemrograman komputer untuk mencapai kriteria/performa tertentu dengan menggunakan sekumpulan data *training* atau pengalaman di masa lalu. (Chazar & Erawan, 2020)

2.7. Deep Learning

Deep learning adalah cabang dari *machine learning* yang fokus pada pengembangan dan penggunaan algoritma *neural networks* yang kompleks untuk meniru cara kerja otak manusia dalam memproses informasi dan belajar dari data. *Neural networks* dalam konteks *deep learning* terdiri dari banyak lapisan (layer) yang memungkinkan model untuk memahami representasi data yang semakin abstrak dan kompleks. (Santoso & Ariyanto, 2018). Seiring berjalannya waktu, muncul beberapa algoritma baru yang mengimplementasikan *deep learning*, salah satunya adalah *Convolutional Neural Network* (CNN).

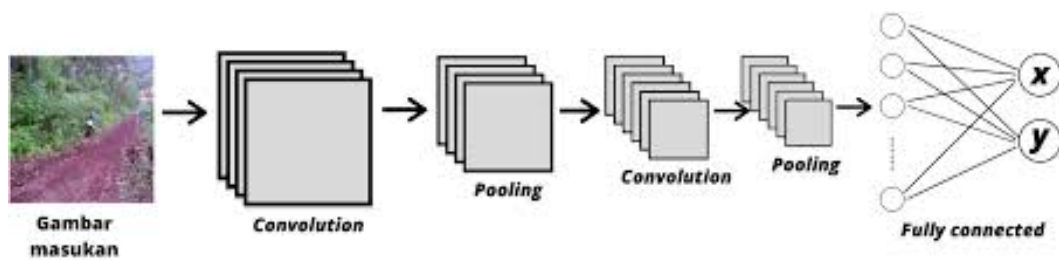
2.8. Convolutional Neural Network

Convolutional Neural Network atau yang biasa disingkat dengan CNN merupakan sebuah algoritma *object detection* yang menerapkan ilmu klasifikasi objek yang menyerupai jaringan saraf dengan menggunakan konvolusi setidaknya

satu pada tiap *layer* untuk mengganti perkalian matriks umum. CNN memiliki kemampuan untuk mengklasifikasikan data berupa citra dengan akurasi yang tinggi karena bisa mengatasi berbagai jenis deformasi gambar seperti *translation*, *rotation* dan *scale*. (Andika et al., 2019)

Secara teknis, CNN merupakan sebuah arsitektur yang bisa untuk dilatih dan memiliki beberapa tahap. *Input* dan *Output* dari tahap-tahap tersebut terdiri atas beberapa *array* yang disebut *feature map*. (Nurhikmat, 2018)

Berikut merupakan jaringan arsitektur *Convolutional Neural Network* :

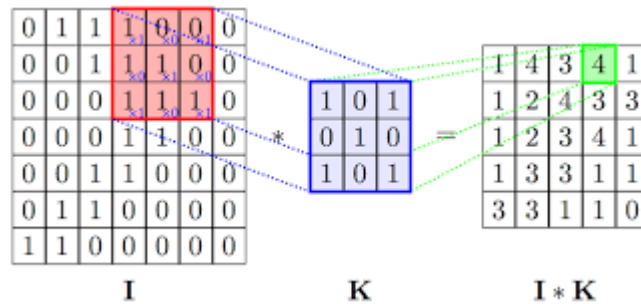


Gambar 2.2. Cara Kerja *Convolutional Neural Network* (Faiz et al., 2020)

Pada gambar 2.2. bisa dilihat bahwa pada dasarnya CNN bekerja melalui tiga jenis lapisan sebelum mengeluarkan output.

4.8.1. *Convolution Layer*

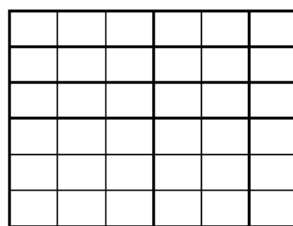
Lapisan ini berfungsi untuk melakukan proses konvolusi pada citra yang dimasukkan. Konvolusi sendiri merupakan istilah matematis yang mengaplikasikan sebuah fungsi pada sebuah *output* fungsi lain terus-menerus. Untuk kasus pengolahan citra, konvolusi mengaplikasikan sebuah kotak kuning yang disebut dengan *kernel* pada gambar atau citra di semua *offset* yang memungkinkan. Ini akan menjadikan citra input diambil sebagian untuk dijadikan sebuah citra baru agar fitur pada citra bisa di ekstraksi. (Suartika E. P, I Wayan, Wijaya Arya Yudhi, 2016)



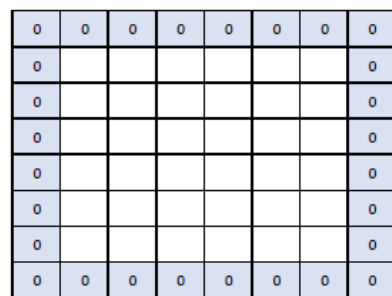
Gambar 2.3. Proses pada *convolution layer* (Sarkar, 2022)

Gambar 2.3. menunjukkan contoh proses pada *convolution layer*. Terlihat bahwa *value* dari gambar *input* 7x7 diberi filter dengan matriks 3x3 sehingga menghasilkan output dengan *value* matriks 5x5.

Pada proses *convolution*, terdapat parameter penting yang berpengaruh pada prosesnya, yaitu *padding* dan *stride*. *Padding* merupakan penambahan batas pada setiap sisi gambar *input* agar proses *convolution* tidak melewati sedikitpun bagian dari gambar *input*. Umumnya, jenis *padding* yang sering digunakan adalah *zeropadding*, yaitu menambahkan elemen 0 pada setiap tepi gambar *input*. Untuk contoh *padding* dengan menggunakan *zeropadding* bisa dilihat pada gambar 2.3.(Luthfi, 2021)



Gambar 6x6

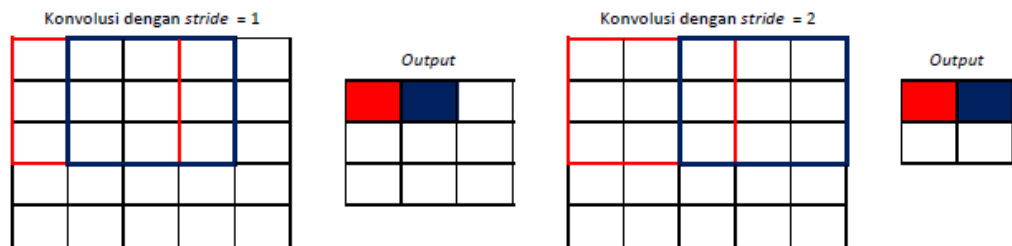


Gambar 6x6 dengan 1 lapisan *zeropadding*

Gambar 2.3. Contoh proses *zeropadding*

Sedangkan *stride* merupakan parameter yang menentukan seberapa banyak filter bergeser melewati input pada saat proses

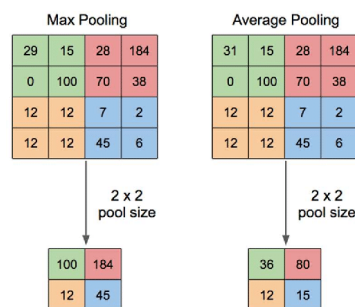
convolution. Sebagai contoh pada gambar 2.4., jika nilai *stride* adalah 1, maka filter akan bergeser sebanyak 1 kotak pada input. Sedangkan jika nilai *stride* adalah 2, maka filter akan bergeser sebanyak 2 kotak pada *input*. Hal ini akan mempengaruhi akurasi dan ukuran output pada saat selesai proses konvolusi. (Luthfi, 2021)



Gambar 2.4. Contoh proses *stride*

4.8.2. *Pooling Layer*

Pooling Layer adalah lapisan yang terdiri dari *filter* dengan ukuran dan *stride* tertentu yang bergerak melintasi seluruh peta fitur. *Layer* ini ditambahkan setelah *convolutional layer* dan berfungsi untuk menyusun layer dalam CNN, yang dapat diulang satu atau lebih kali dalam suatu model. Secara umum, *filter* pada *pooling layer* menggunakan ukuran 2×2 dengan *stride* 2, yang diterapkan pada setiap bagian dari input. Operasi pooling ini bersifat tetap dan tidak dipelajari. (Azmi et al., 2023)



Gambar 2.5. Contoh operasi pada *pooling layer* (Rao et al., 2020)

Gambar 2.5. menunjukkan bahwa ada jenis *pooling* pada CNN. Yang pertama disebut dengan *max pooling*. *Max pooling* mengambil nilai maksimal pada filter 2x2. Berikutnya ada *average pooling*. *Average pooling* mengambil nilai rata-rata dari filter 2x2. (Nurhikmat, 2018)

4.8.3. **Fully Connected Layer**

Fully Connected Layer merupakan lapisan yang dimana semua *neuron* pada lapisan setelahnya sama dengan halnya dengan *neural network* biasa. Lapisan ini biasa digunakan pada sebuah MLP (*Multi Layer Perceptron*) yang bertujuan untuk bertransformasi pada suatu dimensi data agar dapat diklasifikasikan secara linear.

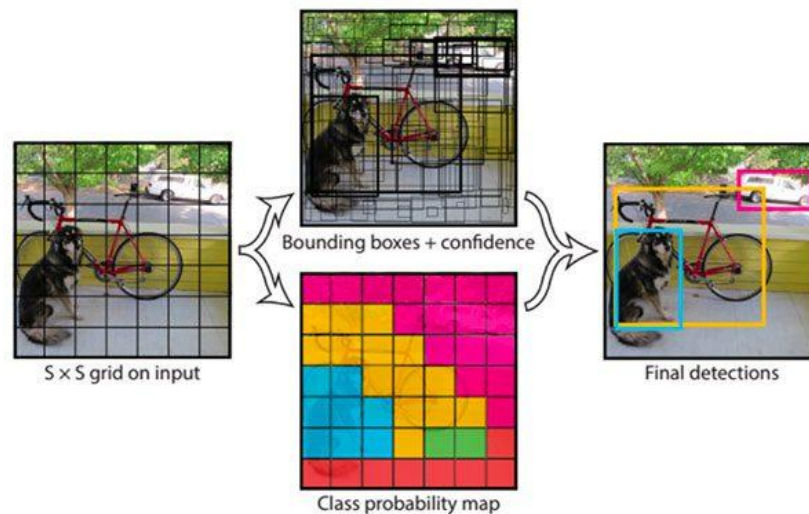
Fully Connected Layer memiliki perbedaan dengan *Convolution Layer*, yakni terletak pada ruang lingkupnya. *Convolution Layer* hanya menghubungkan *neuron* ke daerah tertentu pada suatu *input*, sedangkan *neuron* pada *Fully Connected Layer* mencakup semua area pada input. (Nurhikmat, 2018)

Deteksi objek dalam CNN memiliki setidaknya dua jalur yang berbeda, yakni deteksi satu tahap dan deteksi dua tahap. Deteksi dua tahap membagi proses deteksi kedalam dua tahap, yakni Ekstraksi fitur yang diikuti dengan regresi, lalu klasifikasi untuk mendapatkan *output*. Hal ini mungkin dapat memberikan akurasi yang tinggi, namun perlu komputasi yang cukup kompleks dan tidak terlalu efektif untuk deteksi objek *real-time*. Pada deteksi satu tahap, proses klasifikasi dan regresi dilakukan secara bersamaan sehingga meringankan beban komputasi. Ada beberapa algoritma satu tahap yang ada, seperti *Single-shot Detector* (SSD), *Deconvolutional Single-shot Detector* (D-SSD), RetinaNet, dan *You Only Look Once* (YOLO). Dari keempat algoritma ini, YOLO lebih disukai karena kompabilitasnya yang sangat tinggi dengan kebutuhan deteksi, seperti akurasi yang tinggi, ringan, serta penerapannya yang ramah lingkungan. (Hussain, 2023)

Berdasarkan simpulan paragraf sebelumnya, penelitian ini akan menggunakan *You Only Look Once* (YOLO) sebagai algoritma deteksi objeknya.

2.9. **You Only Look Once (YOLO)**

YOLO merupakan salah satu varian model yang dimiliki *Convolutional Neural Network* (CNN). YOLO menggunakan pendekatan terbaru dalam objek, yang mana pemrosesan data bisa dilakukan secara *realtime*. Pengolahan gambar dengan YOLO cukup mudah simpel. YOLO mengolah gambar dengan melakukan *resize* pada gambar masukan, lalu menjalankan *single convolutional network* pada gambar *input*, dan *setting threshold* untuk hasil deteksi dengan nilai *confidence* dari model (Redmon et al., 2016)



Gambar 2.6. Arsitektur sederhana YOLO (Carolina & Lina, 2023)

Gambar 2.6. menunjukkan arsitektur sederhana dari algoritma YOLO. *Input* pada YOLO awalnya dibagi menjadi *grid* $S \times S$. Setiap sel *grid* mempunyai tanggung jawab untuk memprediksi suatu *bounding box* B , nilai *confidence* untuk kotak tersebut, dan probabilitas kelas C . Nilai *confidence* ini akan menentukan seberapa akurat sebuah model mendeteksi objek didalam *bounding box*. Dalam sebuah *bounding box*, terdapat 5 nilai prediksi yang akan dikeluarkan: x , y , w , h , dan *confidence*. Koordinat x dan y akan merepresentasikan titik pusat dari kotak terhadap batas kotak, w dan h merepresentasikan ukuran gambar *input*, dan *confidence score* menggambarkan seberapa yakin suatu *bounding box* memiliki objek didalamnya. Prediksi pada YOLO biasa dikodekan sebagai $[S, S, B*5+C]$. (Redmon et al., 2016)

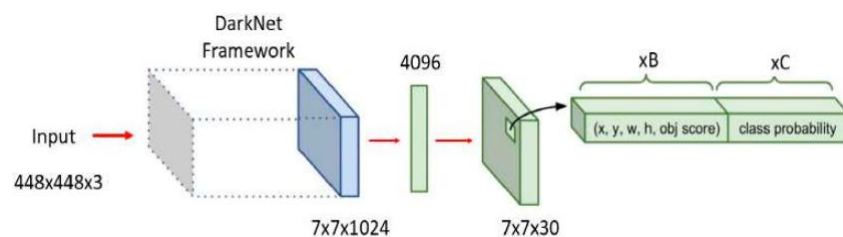
Tujuan utama YOLO adalah mendeteksi dan menentukan lokasi objek menggunakan kotak pembatas. Untuk itu, diperlukan dua set vektor kotak

pembatas: vektor y sebagai representasi *ground truth* dan vektor y' sebagai vektor prediksi. Untuk mengatasi beberapa kotak pembatas yang tidak mengandung objek atau mengandung objek yang sama, YOLO menerapkan *non-maximum suppression* (NMS). Dengan menetapkan nilai ambang batas untuk NMS, semua kotak pembatas yang diprediksi dan memiliki tumpang tindih dengan IoU yang lebih rendah dari nilai NMS yang telah ditetapkan akan dihapus. (Hussain, 2023)

Sejak pertama kali dikenalkan pada tahun 2016, YOLO sudah memiliki beberapa versi hingga sekarang dengan perkembangan yang pesat. Berikut merupakan kilas singkat mengenai tiap-tiap versi YOLO yang pernah ada:

3.9.1. *Original YOLO*

Original YOLO atau sebut saja YOLOv1 merupakan versi YOLO pertama kali yang diluncurkan pada tahun 2015 oleh Joseph Redmon. Prinsip utama yang dibawa oleh YOLOv1 adalah memaksakan *grid cell* dengan ukuran $s \times s$ pada gambar *input*.



Gambar 2.7. Arsitektur YOLOv1 (Hussain, 2023)

YOLOv1 memiliki beberapa celah yang perlu diperhatikan. Arsitektur ini memiliki nilai *recall* yang lebih rendah dan *localization error* yang lebih tinggi dari metode CNN lain yakni R-CNN. YOLOv1 juga cukup sulit untuk mendeteksi objek yang terlalu dekat posisinya. (Hussain, 2023)

3.9.2. *YOLO-v2/9000*

YOLO-v2/9000 dikenalkan oleh Joseph Redmon pada tahun 2016. Versi ini muncul untuk mengatasi setidaknya sebagian kekurangan yang ada pada YOLOv1 namun dengan tetap mempertahankan kecepatannya. Hal ini dilakukan dengan memperkenalkan *Batch Normalization*, yang mana hal ini

menghasilkan proses *training* yang lebih cepat dan mengurangi *overfitting*. Efektivitasnya dapat dibuktikan dengan meningkatnya mAP sebesar 2% dibanding YOLOv1. (Hussain, 2023)

Namun, YOLOv2 masih memiliki masalah. Algoritma ini masih sedikit kesulitan jika berhadapan dengan objek yang kecil pada gambar *input*.

3.9.3. YOLO-v3

YOLO-v3 menawarkan sebuah arsitektur *hybrid* yang mengambil dan memperhitungkan berbagai aspek dari YOLO-v2, Darknet-53, dan konsep ResNet dari *residual network*. ResNet sendiri merupakan sebuah konsep *skip-connection* yang menjaga informasi dan membuatnya tetap tersedia pada *layer* ke *layer* arsitektur. (Hussain, 2023)

Hal ini memungkinkan untuk mendukung fitur *fine-grained* ke *layer* yang lebih dalam. Selain itu, YOLO-v3 juga bisa melakukan deteksi *multi-scale* yang membuat performanya meningkat dan dapat mendeteksi objek yang lebih kecil. (Hussain, 2023)

3.9.4. YOLO-v4

YOLO-v4 adalah varian pertama yang sudah tidak dirilis oleh Redmon. Versi ini dirilis pada tahun 2020 oleh Alexey Bochkovsky. Pada dasarnya versi ini merupakan distilasi dari berbagai teknik deteksi objek untuk diuji dan disempurnakan hingga menjadi detektor objek yang ringan dan dapat digunakan secara *real-time*.

Backbone yang digunakan dalam YOLO-v4 adalah CSPDarknet-53. Hal ini terjadi setelah dilakukan berbagai macam eksperimen dengan fitur-fitur lainnya. (Hussain, 2023)

3.9.5. YOLO-v5

YOLO-v5 memiliki kemiripan dengan YOLO-v4. Yakni sama sama menggabungkan teknik *computer vision* untuk meningkatkan performa. Versi ini dirilis 2 bulan setelah YOLO-v4 oleh Glenn Jocher. Namun bedanya, versi ini tidak dirilis dalam Darknet, melainkan PyTorch. Hal ini dilakukan karena Pythorch memiliki ekosistem yang mapan, dan komunitas yang lebih luas untuk mendukung pada *mobile device* dibandingkan Darknet. (Hussain, 2023)

Selain itu, YOLO-v5 membawa konsep *automatet anchor box learning*. Konsep ini membuat koneksi secara otomatis mempelajari *anchor box* unik pada *dataset* COCO dan mengambil yang paling sesuai untuk mempercepat proses pemodelan. (Hussain, 2023)

3.9.6. YOLO-v6

YOLO-v6 dirilis pada tahun 2022 oleh Tim Teknis Meituan yang berasal dari Tiongkok. Versi ini lebih berfokus untuk detektor objek yang berorientasi pada industri. Sehingga, versi ini memiliki beberapa varian yang menyesuaikan dengan perangkat yang digunakan agar berjalan optimal.

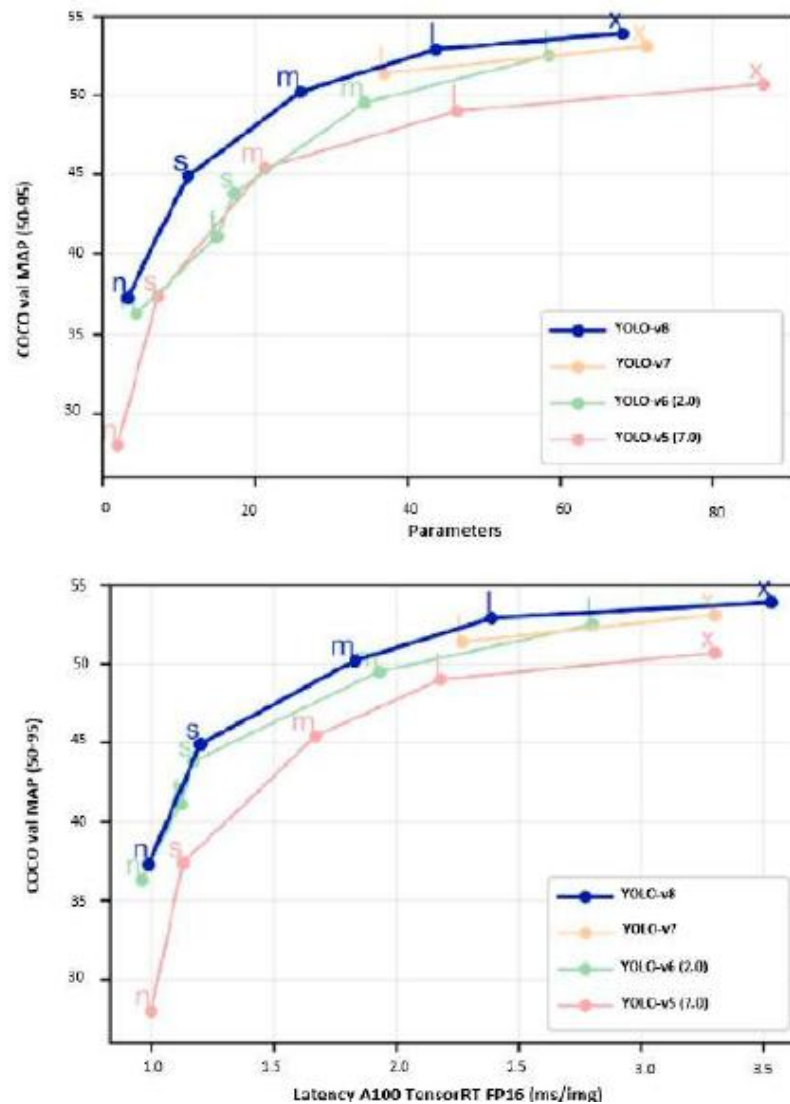
YOLO-v6 memilih untuk menggunakan pendekatan tanpa *anchor*, menghasilkan kecepatan 51% lebih cepat dibandingkan yang menggunakan *anchor*. (Hussain, 2023)

3.9.7. YOLO-v7

Satu bulan setelah YOLO-v6 dirilis, YOLO-v7 muncul untuk meningkatkan kecepatan GPU pada saat proses pemodelan. YOLO-v7 melakukan beberapa reformasi arsitektur untuk meningkatkan akurasi dan kecepatan deteksi yang tinggi. (Hussain, 2023)

3.9.8. YOLO-v8

Januari 2023, YOLO-v8 pun muncul dengan keunggulannya dibandingkan pendahulunya. Versi ini dirilis oleh Ultralytics.



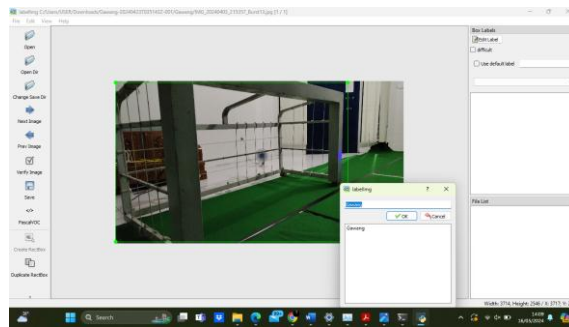
Gambar 2.8. Perbandingan YOLO-v8 (Hussain, 2023)

Pada gambar 2.8. bisa dilihat perbandingan antara YOLO-v8 dengan versi-versi sebelumnya yang dilatih pada gambar dengan resolusi 640p, semua varian YOLO-v8 menunjukkan *throughput* yang lebih baik dengan jumlah parameter yang sama. (Hussain, 2023)

Dari berbagai versi YOLO yang sudah dijabarkan, YOLO-v8 adalah versi yang paling cocok dengan penelitian ini karena menunjang deteksi *real-time* dan memiliki performa yang lebih baik dibandingkan versi pendahulunya (YOLO-v5, YOLO-v6, YOLO-v7) yang juga sama-sama menunjang *real-time*.

2.10. Labelling

Labelling merupakan sebuah proses pemberian label pada sekumpulan *pixel* pembentuk objek pada suatu citra. Objek yang berbeda akan diberi label yang berbeda. Proses *labelling* akan sangat penting untuk proses penganalisaan bentuk dan pengenalan pola pada pengolahan citra. (Mozef, 2003)



Gambar 2.4. Proses *labelling*

2.11. *Confusion Matrix*

Confusion Matrix merupakan sebuah metode perhitungan akurasi yang biasa digunakan pada proses *data mining*. *Confusion matrix* dipresentasikan sebagai sebuah tabel yang menyatakan jumlah data yang benar dan salah setelah diklasifikasikan. (Rahman et al., 2017)

Tabel 2.2. Contoh tabel *Confusion Matrix*

Matrix		<i>Predicted</i>	
		True	False
<i>Actual</i>	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Berdasarkan tabel 2.2. yang mencontohkan tabel *Confusion Matriks*, didapat data sebagai berikut:

- True Positive* (TP) yang bermakna jumlah data positif yang diklasifikasi dengan benar oleh sistem
- True Negative* (TN) yang bermakna jumlah data negatif yang diklasifikasi dengan benar oleh sistem

- c. *False Positive* (FP) yang bermakna jumlah data positif yang diklasifikasikan dengan salah oleh sistem
- d. *False Negative* (FN) yang bermakna jumlah data negatif yang diklasifikasikan dengan salah oleh sistem

Dari keempat data yang sudah didapat, bisa diambil beberapa nilai dengan rumus sebagai berikut:

- Nilai akurasi yang merupakan perbandingan antara data yang diklasifikasikan benar dengan seluruh data yang ada. Nilai ini dapat diperoleh dengan persamaan :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

- Nilai presisi yang merupakan jumlah data positif yang diklasifikasikan benar dengan seluruh jumlah data positif. Nilai ini diperoleh dengan persamaan:

$$\text{Presisi} = \frac{TP}{TP+FP} \times 100\%$$

- Nilai *recall* yang merupakan persenan data positif yang diklasifikasikan benar oleh sistem. Nilai ini diperoleh dengan persamaan :

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\%$$

- Nilai *error* yang merupakan nilai yang menunjukkan seberapa besar kesalahan sistem dalam mengidentifikasi data. Nilai ini diperoleh dengan persamaan :

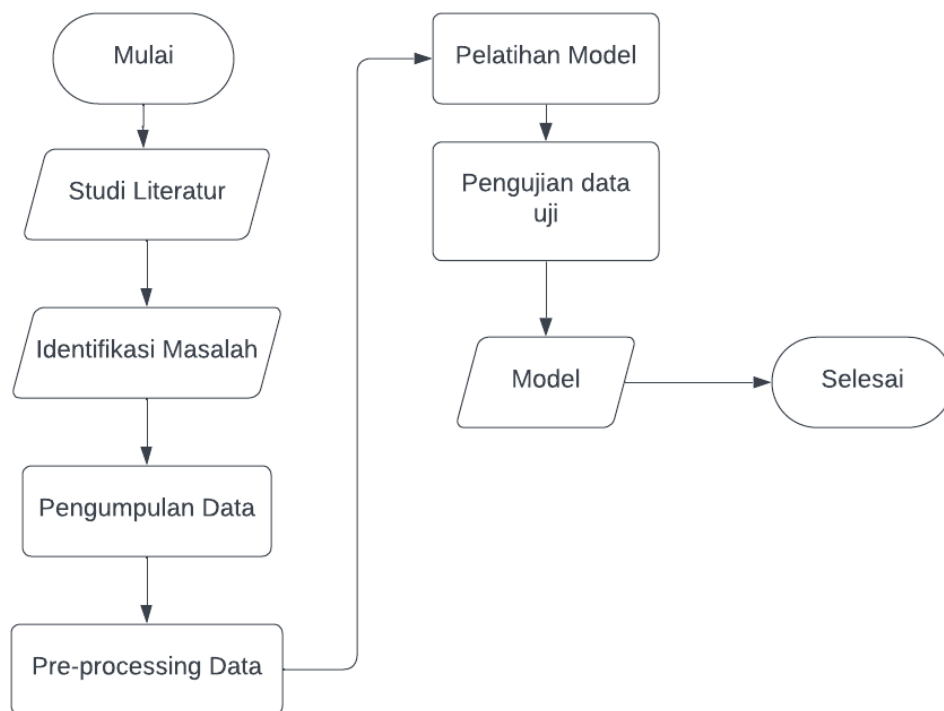
$$\text{Error} = \frac{FP}{TP} \times 100\% \quad (\text{Pratiwi, 2020})$$

BAB III

METODOLOGI PENELITIAN

3.1. Kerangka Pikiran

Pada penelitian ini akan dilakukan beberapa tahap yang digambarkan pada gambar 3.1, yaitu dimulai dengan studi literatur, lalu mengidentifikasi masalah, mengumpulkan data yang dibutuhkan, menyunting data, membuat model dari data yang sudah diatur, menguji model yang dibuat, dan selesai.



Gambar 3.1. Kerangka Pikiran Penelitian

3.2. Studi Literatur

Studi Literatur dilakukan untuk mengumpulkan informasi tentang implementasi *deep learning* melalui metode YOLO dan beberapa teori pendukungnya dikarenakan YOLO memiliki beberapa versi yang berbeda. Selain itu, studi literatur juga dilakukan untuk mendapatkan informasi tentang objek bola dan gawang yang akan diidentifikasi oleh robot. Studi literatur dilakukan juga untuk

mengumpulkan penelitian-penelitian terdahulu yang terkait dengan penelitian ini sehingga dijadikan acuan penelitian. Penelitian terdahulu yang dikumpulkan berupa sistem deteksi bola dan gawang pada robot dengan metode yang berbeda-beda.

3.3. Identifikasi Masalah

Robot harus mendeteksi objek bola dan gawang dengan baik agar bisa berjalan sesuai yang diinginkan. Saat kamera mendeteksi objek bola dan gawang menggunakan metode HSV, seringkali objek tidak terbaca karena intensitas cahaya yang selalu berubah-ubah membuat warna yang ditangkap kamera juga berubah-ubah. Hal ini dapat mengganggu jalannya permainan bola oleh robot sehingga membuat hasil kompetisi tidak didapat dengan baik. Bisa saja operator robot mengganti *range* warna dalam setiap waktu tertentu agar objek bisa dibaca dengan baik, namun tidak akan efisien dari segi waktu.

3.4. Pengumpulan Data Objek Bola dan Gawang

Data yang akan digunakan didalam penelitian ini ada 2 (dua) jenis, yaitu bola dan gawang. Pengumpulan data bola dan gawang dilakukan dengan cara pengambilan data secara langsung. Pengambilan data secara langsung dilakukan dengan menggunakan kamera *Omnidirectional* yang terdapat pada robot KRSBI.

Untuk pengambilan gambar secara langsung melalui kamera *Omnidirectional*, didapat gambar bola sebanyak 100 gambar dan gawang sebanyak 100 gambar.



Gambar 3.2. Contoh gambar bola menggunakan kamera *omnidirectional*

Gambar 3.2 menunjukkan salah satu contoh gambar bola yang diambil menggunakan kamera *omnidirectional*. Terlihat bahwa ada sebuah bola berwarna oranye yang terletak pada samping kanan kamera *omnidirectional* terhadap gambar.



Gambar 3.3. Contoh gambar gawang menggunakan kamera *omnidirectional*

Gambar 3.3 menunjukkan salah satu contoh gambar gawang yang diambil menggunakan kamera *omnidirectional*. Terlihat bahwa ada sebuah bola berwarna putih yang terletak pada samping kanan kamera *omnidirectional* terhadap gambar.

3.5. *Preprocessing Data*

Setelah data dikumpulkan, berikutnya adalah melakukan *Preprocessing Data*. Pada langkah ini objek baik bola dan gawang yang terdapat pada gambar akan di beri label agar terdapat informasi tentang objek pada setiap gambar. Untuk melakukan pemberian label, tiap gambar akan diberi sebuah *box* tepat pada objek bola atau gawang lalu diberikan nama kelas yang merupakan nama objek yang di beri *box* pada gambar tersebut. Proses *labelling* pada gambar bisa dilihat pada gambar 3.4 berikut.



Gambar 3.4. Proses *labelling* menggunakan *labelling*

Setelah semua gambar diberi label, gambar akan dibagi menjadi 2 kelompok, yakni 80% untuk *training* dan 20% untuk *validation*.

3.6. Pelatihan

Setelah dilakukan *labeling* pada semua *dataset* gambar maka didapat pula informasi yang diperlukan yaitu berupa file berekstensi txt yang memuat informasi data citra yang akan digunakan pada proses *training*. Proses *training* sendiri memerlukan beberapa *library* pendukung agar dapat berjalan dengan efektif. *Library* tersebut adalah sebagai berikut:

- a. Ultralytics
- b. NumPy 1.16.2
- c. OpenCV-Python 4.0.0.21

Library OpenCV disertakan untuk membantu dalam *preprocessing* citra pada *dataset*. Sedangkan *Library Ultralytics* sangat efisien untuk deteksi objek dalam implementasi *real-time*, termasuk mendeteksi bola dan gawang.

3.7. Pengujian Data Uji

Pengujian dilakukan untuk menguji tingkat akurasi model agar sesuai yang diharapkan. Proses pengujian ini akan menggunakan *dataset* yang sama seperti untuk membuat model. Namun bedanya, *dataset* ini tidak akan melalui

preprocessing. Berikutnya data akan diuji menggunakan *confusion matrix*. Model akan memprediksi *dataset* yang telah dikumpulkan lalu memasukkannya kedalam tabel *confusion matrix* Seperti pada tabel 3.1.

Tabel 3.1 *Confusion Matrix*

Matrix		<i>Predicted</i>	
		Bola	Gawang
<i>Actual</i>	Bola		
	Gawang		

Setelah mendapat data pada tabel 3.1, barulah dapat dihitung nilai akurasi, presisi, dan *recall*-nya. Diharapkan penelitian ini dapat menghasilkan model dengan akurasi setidaknya 80%

DAFTAR PUSTAKA

- Andika, L. A., Pratiwi, H., & Handajani, S. S. (2019). *Lingga Aji Andika 1 , Hasil Pratiwi 2 , and Sri Sulistijowati Handajani 3 1*. 331–340.
- Ardi Gabelatama. (2018). *PENGEMBANGAN MODIFIKASI GAWANG (GOAL SCORE) SEBAGAI INSTRUMEN LATIHAN DALAM PERMAINAN SEPAKBOLA*. 7, 1–25.
- Azmi, K., Defit, S., & Sumijan, S. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 28–40. <https://doi.org/10.52072/unitek.v16i1.504>
- Carolina, A., & Lina, L. (2023). Sistem Penjualan Dengan Pengenalan Produk Secara Otomatis Menggunakan Metode Yolo. *Simtek: Jurnal Sistem Informasi Dan Teknik Komputer*, 8(2), 271–275. <https://doi.org/10.51876/simtek.v8i2.224>
- Darmawan, A. (2018). *Sistem Pembacaan Bola Menggunakan Omnidirectional Camera Untuk Pergerakan Robot Sepak Bola Beroda*. 21–74.
- Faiz Nashrullah, Suryo Adhi Wibowo, & Gelar Budiman. (2020). The Investigation of Epoch Parameters in ResNet-50 Architecture for Pornographic Classification. *Journal of Computer, Electronic, and Telecommunication*, 1(1), 1–8. <https://doi.org/10.52435/complete.v1i1.51>
- 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>
- Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7). <https://doi.org/10.3390/machines11070677>
- 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*.
- Luthfi, A. (2021). Pendeteksi Senjata berbahaya Pada Percobaan Tindakan Kriminal Dengan Menggunakan Metode YOLO (You Only Look Once).

Teknik Informatika, Fakultas Sains Dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, 135.

- 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.
- Mozef, E. (2003). *ALGORITMA LABELING CITRA BINER DENGAN PERFORMANSI OPTIMAL PROCESSOR-TIME*. 67–77.
- Nurhikmat. (2018). *IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA WAYANG GOLEK*. 93.
- Pratama, M. R., Bhayangkara, E. P., & Ishlah, J. M. (2022). Model Aplikasi Document Scanner Menggunakan Operator Canny Dan Contour Pada Open Cv Berbasis Desktop. *JUTEKIN (Jurnal Teknik Informatika)*, 10(2).
<https://doi.org/10.51530/jutekin.v10i2.635>
- Pratiwi, B. P. (2020). Pengukuran Kinerja Sistem Kualitas Udara. *Jurnal Informatika UPGRIS*, 6(2), 66–75.
- Rahman, M. F., Alamsah, D., Darmawidjadja, M. I., & Nurma, I. (2017). Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN). *Jurnal Informatika*, 11(1), 36.
<https://doi.org/10.26555/jifo.v11i1.a5452>
- Rao, M., Fernandez, C. J., & K., S. (2020). Analysis of Melanoma Lesion Images using Feature Extraction & Classification Algorithms. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(6), 2613–2618.
<https://doi.org/10.35940/ijrte.f8612.038620>
- 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>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 779–

788. <https://doi.org/10.1109/CVPR.2016.91>

- Risfendra, R., Asfinaldi, A., Habibullah, H., & Julisardi, J. (2020). Sistem Pergerakan Robot Kiper Beroda Menggunakan Metode Wall Follower Berbasis Image Processing. *Elkha*, 12(1), 1. <https://doi.org/10.26418/elkha.v12i1.35245>
- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(1), 15–21. <https://doi.org/10.23917/emitor.v18i01.6235>
- Sarkar, M. (2022). *Salt Detection Using Segmentation of Seismic Image*. 2–7.
- 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>
- Suartika E. P, I Wayan, Wijaya Arya Yudhi, S. R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101. *Jurnal Teknik ITS*, 5(1), 76.