

Perancangan Sistem Deteksi Objek Bola Dan Gawang Pada Robot Sepakbola Menggunakan Metode Darknet YOLO

Dwi Novianto Nugroho¹, Lilik Anifah²

^{1,2} Jurusan Teknik Elektro, Universitas Negeri Surabaya, Indonesia

¹dwi.19041@mhs.unesa.ac.id

²lilikanifah@unesa.ac.id

Abstrak— Di bidang robotika, terdapat salah satu jenis robot yang memiliki kemampuan untuk bermain sepakbola, yaitu robot sepakbola humanoid. Robot ini diharapkan memiliki kemampuan utama dalam mendeteksi objek yang ada di lapangan, terutama objek bola dan gawang. Oleh karena itu, perlu dikembangkan sistem deteksi objek bola dan gawang dengan tingkat keakuratan yang tinggi, agar robot dapat mengidentifikasi objek yang tepat untuk menjalankan tugas-tugas selanjutnya. Penelitian ini menggunakan platform robot ROBOTIS OP3 yang terintegrasi dengan webcam Logitech C920. Sistem pendeteksian yang digunakan adalah Darknet YOLO, yang merupakan salah satu penerapan dari *deep-learning*. Jaringan YOLO yang digunakan dalam penelitian ini adalah YOLOv3 Tiny, dan tahapan pelatihannya melibatkan *framework* Darknet. Pada tahap pelatihan jaringan, diperoleh nilai rata-rata loss (*avg loss*) sebesar 0,2805 dan nilai *mean Average Precision* (mAP) sebesar 96,3%. Pada tahap pengujian jaringan, dihasilkan *frame* gambar sistem pendeteksian objek yang terdeteksi dengan diberi label "bola" dan "gawang" serta *bounding box*. Pada penelitian ini sistem deteksi yang telah dikembangkan menghasilkan nilai performa jaringan dengan nilai *accuracy* sebesar 0.97, *precision* sebesar 0.99, *recall* sebesar 0.96, dan *F1-score* sebesar 0.97. Nilai-nilai performa jaringan ini diperoleh dari pengujian menggunakan 200 *frame* gambar yang digunakan sebagai *test set* dalam tahap pelatihan jaringan.

Kata Kunci— robot, *deep-learning*, deteksi objek, darknet, yolo.

I. PENDAHULUAN

Kemajuan teknologi saat ini sejalan dengan kebutuhan yang semakin kompleks dalam berbagai aspek kehidupan manusia. Oleh karena itu, kebutuhan akan teknologi robot menjadi semakin penting seiring dengan berjalannya waktu. Robot dirancang sedemikian rupa agar memiliki kemampuan yang menyerupai perilaku dan pemikiran yang dimiliki oleh manusia, sebagaimana naluri yang dimiliki oleh manusia [1]. Seiring dengan perkembangan teknologi robotika, banyak pekerjaan manusia yang dapat terbantu oleh teknologi ini. Dengan meningkatnya penggunaan perangkat teknologi, dapat diamati bahwa digitalisasi teknologi mengalami perkembangan yang signifikan. Digitalisasi merupakan suatu proses transformasi digital yang melibatkan penggunaan teknologi komputerisasi untuk mempermudah berbagai tugas manusia. Salah satu teknologi komputerisasi yang sedang menjadi perbincangan luas adalah kecerdasan buatan (*artificial intelligence*). Teknologi ini merupakan kemampuan yang dimiliki oleh sistem komputer untuk berinteraksi dengan dunia sebagaimana halnya manusia [2].

Di Indonesia sendiri terdapat suatu kompetisi robotika yang diadakan setiap tahun oleh pemerintah khusus untuk tingkat perguruan tinggi, yaitu Kontes Robot Indonesia (KRI). Kompetisi ini terdiri dari lima cabang kategori perlombaan, salah satunya adalah Kontes Robot Sepak Bola Indonesia (KRSBI), yang telah diadakan sejak tahun 2013 [3]. Dalam peraturan Kontes Robot Sepak Bola Indonesia (KRSBI), terdapat ketentuan bahwa setiap robot harus memiliki kemampuan *autonomous* atau tanpa kendali untuk merespons lingkungan sekitar. Kemampuan ini mencakup mendeteksi keberadaan bola, mendekati bola, berkomunikasi antar robot, dan menjalankan strategi permainan hingga mencetak gol [4]. Semua misi dan kemampuan yang tertera didalam peraturan diharapkan dapat dijalankan dengan menggunakan kontrol otomatis berbasis *artificial intelligence* (AI). Dengan demikian, robot dapat menjalankan tugas sesuai dengan instruksi yang diberikan [1]. Pada umumnya, robot sepak bola yang digunakan dalam kompetisi dibagi menjadi dua kategori, yakni robot sepak bola humanoid dan robot sepak bola beroda. Robot sepak bola humanoid merupakan robot yang memiliki bentuk yang menyerupai manusia, dengan dua tangan, dua kaki, bagian tubuh, dan bagian kepala [1].

Robot sepak bola humanoid merupakan salah satu kategori dari kontes robot indonesia yang akan dibahas didalam penelitian ini terutama dalam aspek kemampuan untuk mendeteksi objek bola dan gawang. Kemajuan teknologi dalam pengenalan dan pendeteksian objek terus berkembang seiring berjalannya waktu. Perkembangan ini secara garis besar dibagi menjadi dua era, yaitu era tradisional dan era penggunaan *deep-learning*. Pada era tradisional, pendeteksian objek dilakukan secara manual dengan keterlibatan manusia dalam memberikan masukan kepada sistem mengenai objek-objek yang akan dideteksi. Di sisi lain, era penggunaan *deep learning* merupakan bagian dari metode *machine learning* yang memungkinkan algoritma sistem untuk belajar dan berkembang secara mandiri melalui data yang telah dirancang dan pengalaman yang diperolehnya, tanpa memerlukan bantuan manusia yang signifikan [5]. Pendeteksi objek, atau yang sering disebut *object detection*, merupakan aplikasi komputer yang bertujuan untuk mendeteksi dan mengidentifikasi objek tertentu. Kemampuan ini termasuk dalam bidang ilmu *computer vision*, yang mempelajari bagaimana komputer dapat bekerja mirip dengan mata manusia untuk mengenali objek [6].

Oleh karena itu, diperlukan suatu sistem pendeteksian objek yang memiliki tingkat keakuratan yang tinggi agar robot dapat mengidentifikasi objek dengan tepat dan melakukan tugas-

tugas selanjutnya seperti menggiring bola, mengoper bola, atau mencetak gol tanpa kesalahan. Dalam penelitian ini, digunakan metode deteksi objek Darknet YOLO yang memiliki kemampuan untuk mendeteksi objek dengan tingkat keakuratan yang tinggi. Algoritma YOLO (*You Only Look Once*) merupakan salah satu metode dalam bidang deep-learning dan merupakan bagian dari machine learning yang saat ini sangat populer dalam pendeteksian objek atau wajah. YOLO dikenal sebagai metode deteksi yang memiliki kecepatan tinggi dan akurasi yang tinggi pula. Algoritma ini juga memiliki kemampuan klasifikasi yang serbaguna dalam melakukan pendeteksian [2]. Sedangkan Darknet merupakan kerangka kerja (*framework*) jaringan saraf bersifat open source yang ditulis dengan menggunakan bahasa pemrograman C dan CUDA. Darknet memiliki kinerja cepat, dan pemasangan yang mudah, dan mendukung komputasi baik dengan menggunakan CPU maupun GPU [7].

Dengan penerapan metode Darknet YOLO, diharapkan penelitian ini mampu mengembangkan suatu sistem deteksi objek bola dan gawang yang tepat dan akurat untuk robot sepak bola humanoid. Pada sistem yang sebelumnya telah diimplementasikan pada prototipe robot, hanya dilakukan pendeteksian bola saja dengan menggunakan metode kontur dan pengenalan bentuk bola menggunakan fungsi *HoughCircle*. Fungsi *HoughCircle* digunakan dalam metode pendeteksian objek, khususnya dalam mendeteksi objek berbentuk lingkaran, dengan menggunakan library OpenCV. Diharapkan sistem baru yang telah dikembangkan ini mampu mengenali objek bola yang berwarna orange dengan *size* 1 dan objek gawang yang berwarna putih, sesuai dengan peraturan KRSBI yang berlaku. Mengingat tingkat kesulitan yang cukup tinggi dalam mendeteksi kedua objek ini, penggunaan metode Darknet YOLO diharapkan dapat memberikan hasil yang memuaskan.

Tujuan dari penelitian ini adalah membuat suatu sistem deteksi objek pada prototipe robot sepakbola humanoid agar robot dapat mendeteksi objek bola dan gawang dengan tingkat akurasi yang baik. Penelitian ini juga termasuk dalam bagian riset robot sepakbola humanoid dalam menentukan metode yang tepat untuk pendeteksian objek, terutama objek bola dan gawang.

II. METODOLOGI PENELITIAN

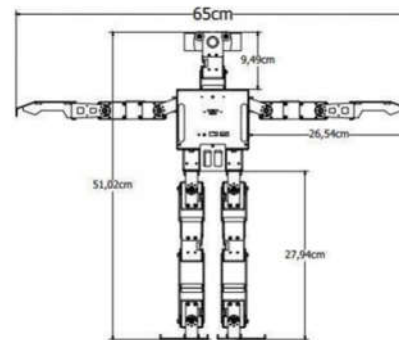
A. Desain Mekanik Robot

Prototipe robot yang digunakan merupakan salah satu platform robot humanoid dari ROBOTIS, yaitu ROBOTIS OP3 ROBOTIS OP3 adalah platform robot humanoid yang merupakan penerus dari ROBOTIS OP dan ROBOTIS OP2. Terdapat perubahan yang terlihat pada ROBOTIS OP3, salah satunya dengan mengganti aktuator Dynamixel MX28 dengan aktuator Dynamixel XM-430. Perubahan lainnya terjadi pada perangkat Mini PC, yang sebelumnya Single Board Computer (SBC) berbasis Intel Atom, sekarang menjadi Intel NUC berbasis Intel Core i3. Dengan menggunakan XM-430 yang mendukung Protocol DYNAMIXEL 2.0, robot memiliki torsi yang lebih baik dan kontrol berbasis arus serta dilengkapi dengan berbagai fitur. Mini PC Intel NUC memungkinkan

pemrosesan data yang lebih kuat dan mendukung OS 64-bit serta memiliki koneksi Bluetooth. OP3 juga dikembangkan dengan menggunakan ROS (Robot Operating System) untuk memanfaatkan berbagai library yang ada pada platform ROS. Dengan perkembangan hardware dan dukungan ROS, pengembang dapat lebih fokus pada riset dan pengembangan dibandingkan dengan robot sebelumnya.

Robot humanoid yang digunakan dalam penelitian ini memiliki struktur mekanik yang terbuat dari bahan dasar aluminium dan menggunakan 20 servo jenis Dynamixel XM-430 untuk menggerakkan setiap sendi yang menyerupai manusia seperti skema 2D yang dapat dilihat pada Gbr. 1.a. Robot ini didesain untuk dapat melakukan berbagai gerakan manusia, seperti duduk, berdiri, berjalan, berlari, menendang bola, menggerakkan kedua tangan, dan bangun saat jatuh.

Robot humanoid khususnya untuk robot sepakbola, dibangun dengan menggunakan rangkaian aktuator yang terdiri dari servo dan bracket yang berfungsi sebagai pendukung dan pengait antar servo yang menghubungkan satu bagian dengan bagian lain [8]. Bahan dasar yang diperlukan yaitu aluminium dengan ketebalan rata-rata 1,5 mm karena ketebalan tersebut cukup untuk menopang berat robot dan tidak memberatkan kinerja dari servo Dynamixel XM-430. Robot ini terdiri dari tiga bagian utama: kepala, tubuh dan tangan, dan kaki, seperti yang ditunjukkan pada Gbr. 1.b



(a)



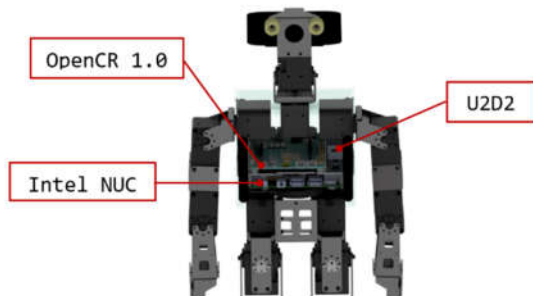
(b)

Gbr. 1 a. Skema 2D Mekanik Robot, b. Desain 3D Mekanik Robot

Bagian utama pada robot yang akan lebih difokuskan dalam penelitian ini adalah bagian awal robot, yaitu kepala, yang dilengkapi dengan servo Dynamixel XM-430. Servo ini digunakan sebagai tempat pemasangan kamera webcam sehingga robot dapat melihat dan mendeteksi objek di sekitarnya. Kemampuan maksimum pergerakan kepala/leher ke kiri atau kanan pada robot ini adalah 180 derajat seperti layaknya manusia, dan kemampuan maksimum pergerakan ke atas adalah 90 derajat [9].

B. Desain Elektrik Robot

Pada area *controller* pada bagian tubuh robot, terdapat berbagai *hardware* yang merupakan bagian elektrik dari robot. Beberapa *hardware* tersebut meliputi Mini PC Intel NUC sebagai *controller* utama, OpenCR 1.0 sebagai *controller* tambahan (*sub-controller*), dan USB Dynamixel atau U2D2. Selain itu, pada bagian kepala robot didukung oleh sebuah webcam yaitu Logitech C920. Area *controller* ditunjukkan pada Gbr. 2



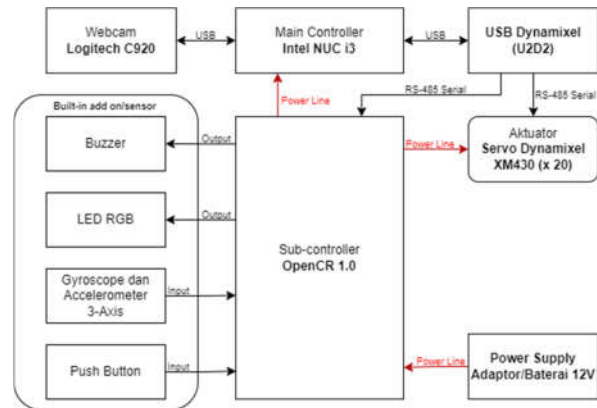
Gbr. 2 Area *controller* pada bagian tubuh robot

Prototipe robot sepak bola humanoid ini menggunakan sensor berupa gyroscope 3-axis sebagai penentu robot kondisi posisi robot, kemudian di padukan dengan sensor accelerometer 3-axis sebagai penentu robot dalam keadaan diam atau berjalan. Dari kedua sensor tersebut robot dapat mendeteksi apakah robot dalam keadaan berjalan, berdiri tegak, diam, maupun terjatuh, kedua sensor ini termasuk sensor *built-in* yang terdapat pada OpenCR 1.0.

Selain itu, webcam Logitech C920 memiliki peran penting dalam mendeteksi keberadaan objek di depan robot, baik itu bola maupun gawang. Kamera ini dihubungkan ke *main controller*, yaitu Intel NUC i3, melalui kabel USB. Selanjutnya, sinyal dari *main controller* dikirim ke *sub-controller* melalui USB yang terhubung dengan USB Dynamixel U2D2. *Sub-controller* tersebut terhubung dengan OpenCR 1.0 serta servo Dynamixel XM430 melalui koneksi Serial RS-485. Hal ini memungkinkan servo digerakkan oleh robot untuk mengatur arah pergerakan (*tracking*) sesuai kebutuhan.

Robot diperbolehkan menggunakan sensor eksternal seperti mikrofon dan speaker, serta sensor sentuhan, sensor tekanan, dan sensor suhu yang dapat ditempatkan di berbagai bagian tubuh robot [9]. *Sub-controller* OpenCR 1.0 dapat bekerja apabila menerima aliran listrik dari baterai atau adaptor untuk menghidupkannya. Untuk penggunaan buzzer dan LED RGB adalah sebagai indikator robot sedangkan *push button*

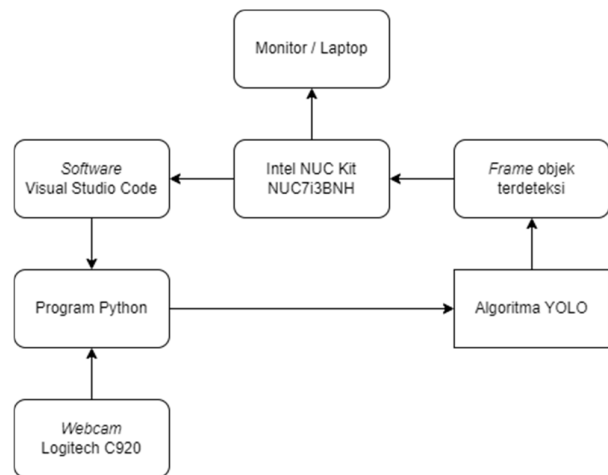
digunakan sebagai tombol untuk mengubah mode dan mereset robot. Diagram blok elektrik selengkapnya dapat dilihat pada Gbr. 3.



Gbr. 3 Diagram blok desain elektrik robot

C. Perancangan Software

Diagram blok perancangan dari *software* pada penelitian ini ditunjukkan pada gambar berikut:



Gbr. 4 Area *controller* pada bagian tubuh robot

Dalam penelitian ini, digunakan komponen hardware berupa webcam Logitech C920. Program yang digunakan untuk penelitian ini ditulis dalam bahasa pemrograman Python dan dibuat menggunakan software Visual Studio Code. Program Python tersebut akan memuat data weight yang diperoleh dari tahapan pelatihan jaringan. Data weight ini akan digunakan untuk memproses citra masukan yang diperoleh secara real-time melalui webcam atau dari file gambar yang telah ditentukan.

Pemrosesan citra tersebut dikerjakan oleh algoritma YOLO. Algoritma YOLO terdiri dari 24-layer jaringan saraf konvolusi (CNN) untuk *feature extraction*, dan dua lapisan terhubung penuh untuk memprediksi probabilitas dan koordinat objek

[10]. Setelah melalui program yang dijalankan dengan menggunakan Python dan diproses oleh Algoritma YOLO, maka layar monitor yang terhubung oleh Intel NUC sebagai Mini PC akan memunculkan window pada *Operating System* berupa frame objek yang terdeteksi.

D. Penggunaan Deep-learning YOLO

Salah satu dari beberapa teknik *deep-learning* untuk *computer vision* adalah *Convolutional Neural Network* (CNN). CNN adalah sebuah jaringan saraf konvolusi yang terdiri dari beberapa lapisan tersembunyi. Setiap lapisan tersebut melakukan perhitungan matematika pada *input* dan menghasilkan *output* yang akan menjadi *input* bagi lapisan berikutnya.

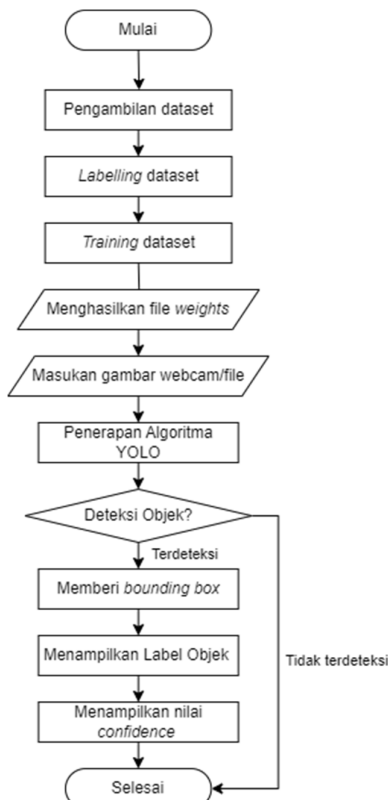
YOLO merupakan metode *Convolutional Neural Network* yang umum digunakan dalam analisis citra. Metode ini memungkinkan deteksi objek dengan menggunakan model terpadu, di mana satu jaringan konvolusi dapat secara bersamaan memprediksi beberapa kotak pembatas dan probabilitas kelas objek dalam kotak-kotak tersebut [11].

Dalam penelitian ini, Algoritma YOLO yang digunakan adalah YOLOv3 Tiny dan dilakukan training dataset dengan menggunakan *convolutional weights* yang telah dilatih menggunakan Imagenet yaitu model weights darknet53.

dilakukan pengambilan gambar dengan menggunakan kamera webcam yang terpasang pada prototipe robot dengan bantuan program Python untuk menangkap tiap frame gambar/foto. Setelah itu, dilakukan proses pelabelan pada tiap *frame* gambar dataset yang telah dikumpulkan, proses ini dilakukan dengan menggunakan aplikasi YOLO-label. *Labeling* dataset atau yang juga dikenal sebagai anotasi merupakan proses penambahan informasi pada objek seperti bola dan gawang, setiap gambar dalam dataset juga akan diberi label satu per satu untuk memperoleh koordinat dari kotak pembatas yang sesuai dengan kenyataan (*ground-truth bounding box*) pada beberapa gambar yang telah diambil sebagai dataset gambar.

Proses selanjutnya adalah melatih dataset tersebut untuk membentuk pola yang dikenal dengan istilah bobot (*weights*). Bobot tersebut akan digunakan untuk mendeteksi objek dalam gambar. Proses pelatihan akan menggunakan metode YOLO (*You Only Look Once*), yang merupakan sebuah metode *deep-learning* berbasis CNN dengan bantuan *framework* Darknet. Perangkat yang digunakan dalam proses training adalah Google Colaboratory, dengan versi config YOLO yang digunakan adalah YOLOv3 Tiny dan *weights* untuk *convolutional layers* dengan menggunakan bobot *convolutional* yang dilatih sebelumnya di Imagenet yaitu menggunakan bobot dari model darknet53. Perangkat Google Colaboratory sendiri merupakan sebuah alat yang baru dikembangkan oleh Google *Internal Research*. Alat ini dirancang untuk membantu para peneliti dalam mengolah data untuk keperluan pembelajaran di bidang pengolahan data menggunakan *machine learning* [12]. Setelah melalui tahap training dataset maka dihasilkan sebuah file bobot (*weights*) yang selanjutnya akan digunakan pada tahap pengujian sistem deteksi objek.

Pada tahap pengujian sistem deteksi objek. Pada tahap ini akan dilakukan pengujian secara tidak langsung yaitu dengan menggunakan masukan gambar ke sistem deteksi, gambar tersebut terdapat didalam folder yang berisikan 200 *frame* gambar dari dataset untuk *test set*. Pada tahap ini juga hasil dari pengujian akan ditampilkan luaran berupa bounding box sesuai dengan prediksi koordinat *bounding box* (x, y, w, h), nilai *confidence*, dan *class probability* dari objek terdeteksi



Gbr. 5 Flowchart implementasi metode Darknet YOLO

Sesuai dengan *flowchart* pada Gbr.5 dapat dideskripsikan beberapa urutan dalam penerapan dari metode Darknet YOLO ini, diawali dengan tahapan pengambilan dataset, pada tahap ini

III. HASIL DAN PEMBAHASAN

A. Perancangan Sistem

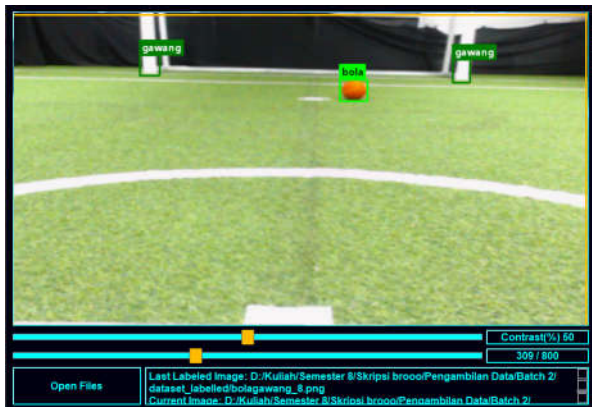
Tahapan perancangan sistem yang pertama sesuai dengan *flowchart* pada Gbr.5 yaitu pengambilan dataset, untuk dataset yang diambil merupakan gambar dari kedua objek tersebut yaitu bola dan gawang, lalu, kedua objek yang terdapat pada gambar dataset tersebut dibedakan menjadi 3 bagian yaitu untuk dataset bola, dataset gawang, dan dataset dari kedua objek tersebut. Pada tiap bagian memiliki jumlah gambar dataset yang berbeda sesuai jumlah dataset training yang telah ditentukan yaitu sebesar 1000 gambar dataset, dengan rasio pembagian 80% *train set* dan 20% *test set*. Pada training set untuk objek bola sebanyak 300 gambar, objek gawang dengan 300 gambar, dan objek gabungan bola gawang dengan 200 gambar. Seperti pada gambar berikut:



Gbr. 6 Kumpulan dataset yang telah diambil

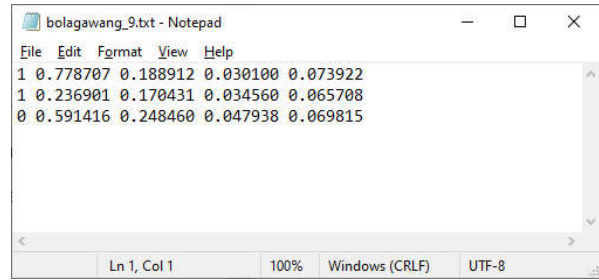
Pengambilan gambar tersebut menggunakan program python yang dibuat dengan library OpenCV. Cara kerja dari program python tersebut yaitu dengan menggunakan webcam secara real-time untuk menampilkan video pada windows kemudian dengan menekan tombol space pada keyboard maka gambar pada frame tertentu yang ada di video akan tersimpan pada folder (path) yang telah ditentukan pada program tersebut. Tahapan berikutnya yaitu dengan melakukan pelabelan gambar dataset seperti pada Gbr. 7

Dalam proses pelabelan objek bola, dilakukan dengan menandai objek bola dari sudut kiri atas hingga sudut kanan bawah. Selanjutnya, pada proses pelabelan objek gawang, dilakukan dengan memberikan label pada kedua tiang gawang bagian bawah dari objek tersebut. Jika objek gawang yang terdeteksi berbentuk utuh pada satu frame, maka akan dianggap terdapat dua objek gawang yang terdeteksi. Namun, jika hanya satu tiang gawang yang terdeteksi atau objek tidak utuh, maka akan dianggap terdapat satu objek gawang yang terdeteksi.



Gbr. 7 Proses pelabelan gambar dataset

Setelah dilakukan pelabelan, hasilnya yang berupa file anotasi dengan format .txt akan disimpan dalam folder yang juga berisi file gambar dataset objek bola dan gawang, file anotasi tersebut terdapat informasi seperti nomor kelas objek, koordinat x dan y dari kelas objek, dan lebar dan tinggi dari kelas objek. Isi dari file anotasi tersebut terdapat pada Gbr.8.



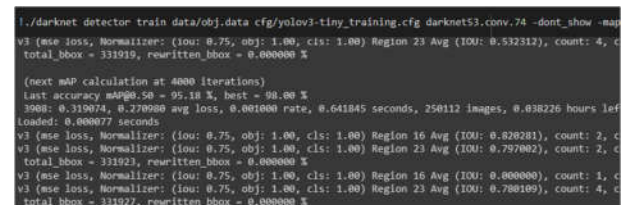
Gbr. 8 Proses pelabelan gambar dataset

Setelah selesai dilakukan pengambilan dataset dan proses pelabelan objek pada gambar untuk seluruh dataset pelatihan, langkah berikutnya adalah melakukan pelatihan atau training jaringan deteksi objek bola dan gawang dengan menggunakan YOLOv3 Tiny. Proses training akan menggunakan 800 gambar dari dataset, terdiri dari 300 gambar objek bola, 300 gambar objek gawang, dan 200 gambar objek gabungan bola dan gawang, serta bobot pretrained "darknet53.conv.74". Sebelum dilakukannya proses pelatihan jaringan diperlukan juga untuk melakukan penyesuaian untuk pelatihan jaringan YOLOv3 Tiny, yaitu dengan mengatur *hyperparameter* yang terkait dengan jaringan tersebut. Konfigurasi dari nilai *hyperparameter* dapat dilihat pada Gbr.9.



Gbr. 9 Konfigurasi *hyperparameter* YOLOv3Tiny

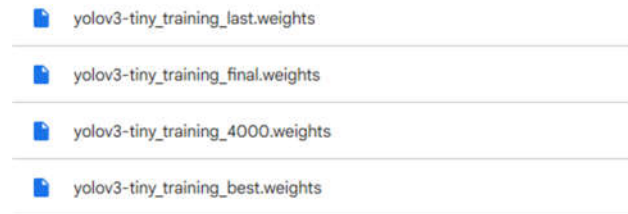
Proses training dilakukan pada platform Google Collaboratory menggunakan *framework* Darknet. Proses pelatihan yang sedang berlangsung ditunjukkan pada gambar berikut ini:



Gbr.10 Proses pelatihan jaringan YOLOv3 Tiny

Setelah proses pelatihan jaringan selesai dilakukan, maka akan menghasilkan sebuah file pada Google Drive yaitu "yolov3-tiny_training_final.weights" yang berisi bobot akhir dari jaringan yang telah dilatih, seperti yang terlihat pada Gbr.10. Apabila proses pelatihan terdapat gangguan maka dapat dilanjutkan dengan menggunakan bobot sesuai dengan iterasi terakhir yang muncul pada Google Drive, pada Gbr.10. Saat proses pelatihan jaringan sedang berlangsung, akan disimpan file "yolov3-tiny_training_last.weights" yang berisi bobot terakhir yang dilatih setiap 1000 kali epoch ke dalam path backup yang telah ditentukan di file "obj.data". Jadi, jika terjadi

gangguan atau masalah selama proses pelatihan, kita dapat menggunakan file bobot terakhir yang tersimpan



Gbr. 11 File bobot hasil proses pelatihan jaringan

Lamanya waktu pelatihan ini bervariasi tergantung pada jumlah data, total kelas keluaran, dan nilai "max_batches" yang telah ditentukan. Dalam pelatihan jaringan YOLOv3 Tiny ini, waktu yang diperlukan sekitar 3 jam tanpa gangguan.

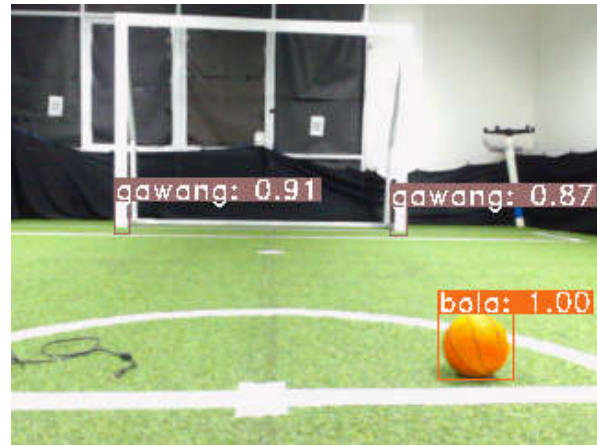
B. Pengujian Deteksi Objek Bola dan Gawang

Untuk melakukan deteksi pada objek bola dan gawang menggunakan jaringan YOLOv3 Tiny yang telah dilatih, langkah pertama adalah mengunduh file-file yang diperlukan untuk proses deteksi. File-file tersebut termasuk file "yolov3-tiny_training_final.weights" yang berisi bobot akhir jaringan YOLOv3 Tiny yang telah dilatih, file "yolov3-tiny_testing.cfg" yang berisi konfigurasi untuk digunakan pada proses pendeteksian objek, dan terakhir file "classes.txt" yang berisi nama-nama kelas objek yang diperoleh dari Google Drive.

Dalam tahap pengujian deteksi objek bola dan gawang, diperlukan program Python yang menggunakan library OpenCV. Program yang digunakan untuk pengujian deteksi objek yaitu kode program untuk mendeteksi objek pada kumpulan gambar *test set* yang terdiri dari 200 gambar dan tersimpan dalam satu folder.

Didalam program tersebut terdapat beberapa fungsi seperti fungsi yang digunakan untuk menggambar *bounding box* pada objek yang terdeteksi berdasarkan nilai x, y, w, dan h yang dihasilkan saat proses ekstraksi *bounding box*, fungsi untuk mengekstrak *bounding box*, nilai confidence, dan id kelas dari objek yang terdeteksi, fungsi untuk memprediksi dan penyesuaian untuk mendeteksi objek, dan yang terakhir fungsi 'main' digunakan untuk memuat gambar dari webcam secara real-time, dan terdapat beberapa argumen atau parameter tambahan yang digunakan untuk mengubah variabel dalam proses deteksi, seperti batas nilai confidence agar objek terdeteksi dan threshold NMS (*Non-Max Suppression*).

Pengujian dilakukan dengan menjalankan program yang telah dirancang dan dibuat dengan menggunakan software Visual Studio Code dan menghasilkan sejumlah *frame* gambar *test set* yang telah terdeteksi baik objek bola maupun gawang dan terdapat pada folder keluaran yang telah ditentukan pada program yang dibuat. Seperti yang ditunjukkan pada Gbr. 11.



Gbr. 12 Frame gambar objek bola dan gawang yang terdeteksi

C. Pengujian Performa Jaringan YOLOv3 Tiny

Pada tahapan selanjutnya dilakukan pengujian performa jaringan YOLOv3 Tiny yang telah dilatih. Dalam pengujian ini, akan dilakukan perhitungan *confusion matrix* untuk menghasilkan nilai *accuracy*, *recall*, *precision*, dan *F1-score* dari sistem deteksi objek bola dan gawang menggunakan jaringan yang telah dilatih. Pengujian performa dan perhitungan tersebut akan dilakukan secara manual dengan menggunakan 200 frame dari test set. *Confusion matrix* merupakan metode pengukuran performa untuk masalah klasifikasi dalam machine learning, di mana nilai TP (*True Positive*) adalah jumlah data yang benar-benar positif dan diprediksi dengan benar oleh sistem sebagai positif. Nilai FP (*False Positive*) adalah jumlah data yang sebenarnya negatif tetapi diprediksi sebagai positif oleh sistem. Nilai TN (*True Negative*) adalah jumlah data yang benar-benar negatif dan diprediksi dengan benar oleh sistem sebagai negatif. Sedangkan nilai FN (*False Negative*) adalah jumlah data yang sebenarnya positif tetapi diprediksi sebagai negatif oleh sistem [13].

Berdasarkan dataset *test set* yang terdiri dari 200 frame dengan pembagian 80 gambar dataset objek bola, 80 gambar dataset objek gawang, dan 40 gambar dataset objek bola dan gawang yang digunakan dalam tahap pelatihan jaringan dan hasil deteksi objek menggunakan program yang telah dibuat, maka dapat dilakukan perhitungan *confusion matrix* dari data yang diperoleh tersebut. Berikut ini adalah tabel yang menunjukkan hasil perhitungan *confusion matrix* dari pengujian performa jaringan.

TABEL I
 TABEL CONFUSION MATRIX DARI OBJEK BOLA

Nilai Prediksi	Nilai Aktual	
	Bola	Bukan Bola
Bola	168	1
Bukan Bola	2	80

TABEL III
 TABEL CONFUSION MATRIX DARI OBJEK GAWANG

Nilai Prediksi	Nilai Aktual	
	Bola	Bukan Gawang
	Bola	168
Bukan Gawang	2	80

Tabel I menunjukkan hasil perhitungan *confusion matrix* secara manual untuk objek bola dengan menggunakan gambar dari *test set*. Sementara itu, Tabel II menampilkan perhitungan manual *confusion matrix* yang diperoleh dari deteksi objek gawang menggunakan gambar dari *test set*. Dapat diketahui bahwa pada Tabel I dan Tabel II nilai TP memiliki jumlah 168 objek bola dan 164 objek gawang yang terdeteksi, kemudian untuk nilai FN adalah sebanyak 2 objek bola dan 10 objek gawang, selanjutnya, nilai FP adalah sebanyak 1 objek bola dan 2 objek gawang. Dan yang terakhir nilai TN berjumlah 80 objek bola dan 79 objek gawang, kedua nilai TN dari objek ini diperoleh dari pembagian dataset yang sudah ditentukan yaitu 80 gambar dataset objek bola dan 80 gambar dataset objek gawang, untuk 79 objek gawang tersebut dikarenakan terdapat gambar dataset objek bola yang didalam gambar tersebut juga terdeteksi objek gawang.

Berdasarkan data yang ada di kedua tabel tersebut, dapat diketahui hasil perhitungan nilai *accuracy*, *precision*, *recall*, dan *F1-score* pada pengujian performa jaringan untuk mendeteksi 200 frame gambar dari test set. Nilai *accuracy* merupakan ukuran sejauh mana model pendeteksian dapat mengidentifikasi objek dengan benar [13]. Nilai *precision* menggambarkan kemampuan model dalam mengidentifikasi objek yang relevan dengan tingkat persentase prediksi yang benar. Sedangkan, nilai *recall* menggambarkan kemampuan model dalam menemukan semua objek yang relevan dengan tingkat persentase positif yang benar dan dapat terdeteksi dari semua data *Ground Truth* yang ada [14]. Sedangkan *F1-score* digunakan untuk mengukur kombinasi antara nilai *precision* dan nilai *recall* [13].

Berikut merupakan persamaan yang digunakan untuk perhitungan nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

$$TP = 168 + 164 = 332 \quad (1)$$

$$FN = 2 + 10 = 12 \quad (2)$$

$$FP = 1 + 2 = 3 \quad (3)$$

$$TN = 80 + 79 = 159 \quad (4)$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} = \frac{491}{506} = 0.97 \quad (5)$$

$$Precision = \frac{TP}{TP+FP} = \frac{332}{335} = 0.99 \quad (6)$$

$$Recall = \frac{TP}{TP+FN} = \frac{332}{344} = 0.96 \quad (7)$$

$$F1 - score = 2 \times \left(\frac{(Precision \times Recall)}{(Precision+Recall)} \right) = 2 \times \left(\frac{(0.99 \times 0.96)}{(0.99+0.96)} \right) = 0.97(8)$$

Berdasarkan perhitungan menggunakan persamaan (1) hingga (8) diperoleh nilai TP total dari kedua objek sebesar 332 yang berarti terdapat 332 objek bola dan gawang yang berhasil terdeteksi dengan benar, nilai FN total sebesar 12 yang berarti terdapat 12 objek bola dan gawang yang gagal dideteksi, nilai

FP total sebesar 3 yang artinya terdapat 3 objek bola dan gawang yang salah dalam pendeteksian, dan nilai TN total sebesar 159 yang berarti terdapat 159 objek bola dan gawang yang memang tidak terdeteksi saat dilakukan pendeteksian diantara kedua objek. Dapat diperoleh juga dari perhitungan nilai *accuracy* adalah sebesar 0.97, nilai *precision* diperoleh sebesar 0.99, nilai *recall* sebesar 0.96, dan nilai *F1-score* adalah sebesar 0.97.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa kesimpulan yang dapat diambil, antara lain adalah sebagai berikut. Pada penelitian ini telah dikembangkan sebuah sistem pendeteksian objek bola dan gawang pada robot sepakbola humanoid ROBOTIS OP3 menggunakan metode Darknet YOLO yang pada tahap pelatihan menggunakan framework Darknet dan konfigurasi jaringan YOLOv3 Tiny. Selama proses pelatihan, kelas objek yang dilatih adalah "bola" dan "gawang". Setelah melalui proses pelatihan, file bobot yang dihasilkan diuji pada program Python dengan dukungan dari *library* OpenCV dan algoritma YOLO. Pada pengujian performa dari jaringan yang telah dilatih dengan menggunakan *test set* sebanyak 200 *frame* gambar didapatkan nilai performa jaringan yang baik yaitu dengan nilai *accuracy* sebesar 0.97, *precision* sebesar 0.99, *recall* sebesar 0.96, dan *F1-score* sebesar 0.97..

UCAPAN TERIMA KASIH

Penulis ingin menyampaikan rasa terima kasih dan apresiasi yang tinggi kepada Universitas Negeri Surabaya, Surabaya, Indonesia, terutama kepada Jurusan Teknik Elektro, Fakultas Teknik, dan pihak terkait atas kesempatan dan fasilitas yang telah diberikan untuk melaksanakan penelitian ini..

REFERENSI

- [1] R. A. Fatekha, B. S. B. Dewantara, and H. Oktavianto, "Sistem Deteksi Bola pada Robot Kiper Pemain Sepakbola Beroda," *J. Integr.*, vol. 13, no. 2, pp. 127–134, 2021, doi: 10.30871/ji.v13i2.3133.
- [2] I. Salamah, M. R. A. Said, and S. Soim, "Perancangan Alat Identifikasi Wajah Dengan Algoritma You Only Look Once (YOLO) Untuk Presensi Mahasiswa," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1492, 2022, doi: 10.30865/mib.v6i3.4399.
- [3] N. Satrio Pambudi, D. M. Wiharta, and N. Putra Sastra, "Analisa Kestabilan Gerakan Statis Pada Robot Humanoid," *J. SPEKTRUM*, vol. 5, no. 2, p. 253, 2018, doi: 10.24843/spektrum.2018.v05.i02.p32.
- [4] S. Darmawan, "Sistem Komunikasi Robot Pada Robot Krakatau Fe," *J. Portal Data*, vol. 2, no. 1, pp. 1–23, 2022, [Online]. Available: <http://portaldata.org/index.php/portaldata/article/view/64>
- [5] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, "Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 6, no. 2, p. 192, 2021, doi: 10.24114/cess.v6i2.25840.
- [6] A. Sujarwo, "Deteksi Kematangan Buah Pepaya Menggunakan Algoritma," *J. INFOKAM*, vol. Vol. XVIII, no. No. 2, 2022.
- [7] J. Redmon, "Darknet: Open Source Neural Networks in C," 2016. <https://pjreddie.com/darknet/> (accessed Feb. 12, 2023).
- [8] H. Hadriansa and D. Prayogi, "Pengenalan Citra Bola Robot Bluehuman G8," *Sebatik*, vol. 22, no. 2, pp. 188–193, 2018, doi: 10.46984/sebatik.v22i2.326.
- [9] B. Kusumoputro et al., *Panduan Kontes Robot Indonesia (KRI)*

- Tahun 2022. 2022.
- [10] S. Amraee, M. Chinipardaz, and M. Charoosaei, "Analytical study of two feature extraction methods in comparison with deep learning methods for classification of small metal objects," *Vis. Comput. Ind. Biomed. Art*, vol. 5, no. 1, 2022, doi: 10.1186/s42492-022-00111-6.
- [11] K. Khairunnas, E. M. Yuniamo, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.61622.
- [12] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan
- Algoritma YOLO (You Only Look Once)," *J. Nas. Ilmu Komput.*, vol. 2, no. 3, pp. 213–232, 2021, doi: 10.47747/jurnalnik.v2i3.534.
- [13] F. F. Sanubari and R. D. Puriyanto, "Deteksi Bola dan Gawang dengan Metode YOLO Menggunakan Kamera Omnidirectional pada Robot KRSBI-B," *Bul. Ilm. Sarj. Tek. Elektro*, vol. 4, no. 2, pp. 76–85, 2022, doi: 10.12928/biste.v4i2.6712.
- [14] A. F. Fandisyah, N. Iriawan, and W. S. Winahju, "Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3," *J. Sains dan Seni ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373520.v10i1.59312.