

Pengembangan Kursi Roda Otonom Berbasis YOLOv8 Untuk Penghindaran *Obstacle*

Dr. Eko Mulyanto Yuniarno, S.T., M.T. Dr. Arief Kurniawan, S.T., M.T. I Gst Ngr Agung Hari Vijaya Kusuma
Departemen Teknik Komputer *Departemen Teknik Komputer* *Departemen Teknik Komputer*
Institut Teknologi Sepuluh Nopember *Institut Teknologi Sepuluh Nopember* *Institut Teknologi Sepuluh Nopember*
Surabaya, Indonesia 60111 Surabaya, Indonesia 60111 Surabaya, Indonesia 60111
ekomulyanto@ee.its.ac.id arifku@ee.its.ac.id 07211940000073@student.its.ac.id

Abstrak—Pengembangan kursi roda otonom telah menjadi semakin penting dalam meningkatkan mobilitas bagi individu dengan mobilitas terbatas. Studi ini mengusulkan pengembangan sistem kursi roda otonom berbasis YOLOv8 untuk menghindari obstacle, khususnya fokus pada deteksi obstacle manusia. Dengan memanfaatkan kemampuan deteksi objek yang canggih dari YOLOv8, sistem yang diusulkan bertujuan untuk mendeteksi dan menghindari obstacle manusia secara efektif. Sistem tersebut mendeteksi manusia melalui video menggunakan Intel NUC dan Kamera. Obstacle yang terdeteksi membuat NUC mengirim perintah ke ESP32 untuk menjalankan motor untuk melakukan manuver penghindaran. Pengujian performa keberhasilan penghindaran dilakukan dengan 30 kali percobaan pada objek manusia yang diam. Hasil pengujian menunjukkan bahwa kursi roda berhasil menghindar sebanyak 30 kali tanpa gagal, memberikan tingkat keberhasilan sebesar 100%. Hal ini menunjukkan bahwa sistem kursi roda otonom yang dirancang mampu melakukan penghindaran rintangan dengan sangat baik.

Kata kunci—Kursi Roda Otonom, YOLOv8, Intel NUC, ESP32, Deteksi Manusia, Bantuan Mobilitas

I. PENDAHULUAN

Mobilitas merupakan aspek penting dalam kehidupan sehari-hari yang memungkinkan individu untuk melakukan berbagai aktivitas dengan mandiri. Namun, bagi individu yang mengalami kelumpuhan atau gangguan mobilitas lainnya, ketergantungan pada alat bantu seperti kursi roda menjadi hal yang tidak terhindarkan. Menurut Kamus Besar Bahasa Indonesia (KBBI), lumpuh adalah kondisi kehilangan fungsi gerak pada bagian tubuh tertentu yang dapat disebabkan oleh berbagai faktor, seperti cedera atau penyakit saraf [1]. Dalam konteks medis, kelumpuhan dapat terjadi akibat kerusakan pada sistem saraf, baik itu saraf pusat maupun saraf tepi, yang mengakibatkan hilangnya kontrol motorik pada anggota tubuh [2].

Perkembangan teknologi dalam bidang ini telah memberikan kontribusi signifikan dalam meningkatkan kualitas hidup penyandang disabilitas. Salah satu inovasi yang menonjol adalah pengembangan kursi roda elektrik yang dikendalikan melalui joystick. Penelitian oleh Choi, Chung, dan Oh menunjukkan bahwa kontrol gerak kursi roda elektrik yang diintegrasikan dengan joystick dapat meningkatkan keselamatan dan kenyamanan pengguna [3]. Meskipun demikian, tantangan tetap ada dalam hal navigasi dan penghindaran rintangan, terutama di lingkungan yang dinamis dan kompleks.

Seiring dengan kemajuan teknologi kecerdasan buatan, deep learning telah menjadi alat yang sangat efektif dalam mendeteksi, mengidentifikasi, dan melacak objek secara real-time. Lecrosnier dalam penelitiannya mengungkapkan bahwa penggunaan deep learning dalam deteksi dan pelokalisasi objek pada kursi roda pintar dapat meningkatkan mobilitas dan keselamatan pengguna di lingkungan kesehatan [4]. Teknologi ini memungkinkan kursi roda untuk secara otomatis menghindari rintangan, termasuk manusia, sehingga pengguna dapat bergerak dengan lebih aman dan efisien.

Pengembangan kursi roda otonom menjadi langkah vital untuk meningkatkan kemandirian dan kualitas hidup individu dengan keterbatasan mobilitas. Dengan perkembangan teknologi sensor dan pemrosesan gambar, aplikasi metode deteksi objek seperti YOLO (You Only Look Once) menjadi inti dari inovasi solusi mobilitas otonom.

Dengan latar belakang ini, proyek ini bertujuan untuk meningkatkan navigasi kursi roda otonom melalui integrasi YOLOv8 dalam deteksi dan penghindaran rintangan, serta memastikan keselamatan dan kenyamanan pengguna dalam berbagai situasi.

II. TINJAUAN PUSTAKA

A. Object Detection

Deteksi objek secara real-time telah muncul sebagai komponen kritis dalam berbagai aplikasi, meliputi berbagai bidang seperti kendaraan otonom, robotika, pengawasan video, dan realitas tertambah. Di antara berbagai algoritma deteksi objek, kerangka kerja YOLO (You Only Look Once) telah menonjol karena keseimbangan kecepatan dan akurasi yang luar biasa, memungkinkan identifikasi objek yang cepat dan dapat dianalisa dalam gambar. Sejak diperkenalkan, keluarga YOLO telah berkembang melalui beberapa iterasi, setiap versi membangun atas versi sebelumnya untuk mengatasi keterbatasan dan meningkatkan kinerja.

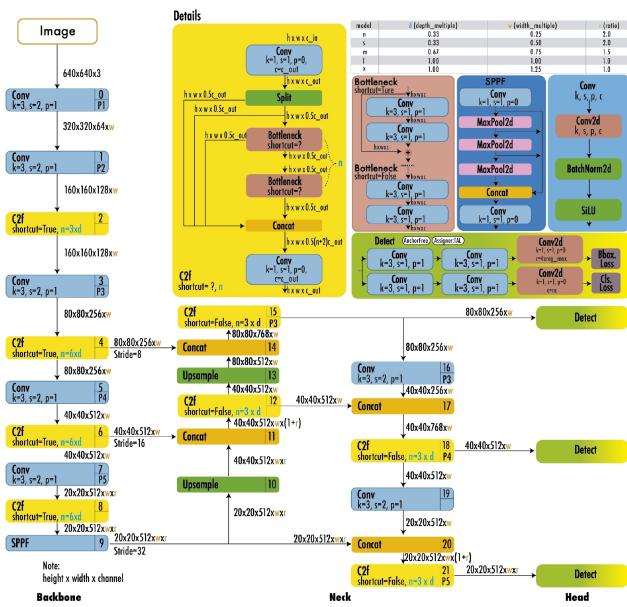
B. YOLO (You Only Look Once)

YOLO oleh [5]. Joseph Redmon untuk pertama kalinya memperkenalkan pendekatan *End to end* pada deteksi objek secara *Real Time*. Nama YOLO, yang merupakan singkatan dari "You Only Look Once" (Anda Hanya Melihat Sekali),

merujuk pada kemampuannya untuk menyelesaikan tugas deteksi dengan hanya satu kali laluan jaringan, berbeda dengan pendekatan sebelumnya yang menggunakan jendela geser diikuti oleh pengklasifikasi yang harus dijalankan ratusan atau ribuan kali per gambar atau metode yang lebih canggih yang membagi tugas menjadi dua langkah, di mana langkah pertama mendeteksi kemungkinan daerah dengan objek atau proposal daerah dan langkah kedua menjalankan pengklasifikasi pada proposal tersebut. Selain itu, YOLO menggunakan keluaran yang lebih sederhana berdasarkan hanya regresi untuk memprediksi keluaran deteksi sebagai lawan dari Fast R-CNN yang menggunakan dua keluaran terpisah, sebuah klasifikasi untuk probabilitas dan regresi untuk *box* koordinat[5]

C. YOLOv8

YOLOv8 diluncurkan pada Januari 2023 oleh *Ultralytics*, perusahaan yang mengembangkan YOLOv5. YOLOv8 menyediakan lima versi skala: YOLOv8n (nano), YOLOv8s (kecil), YOLOv8m (sedang), YOLOv8l (besar), dan YOLOv8x (ekstra besar). YOLOv8 mendukung berbagai tugas visi seperti deteksi objek, segmentasi, estimasi pose, pelacakan, dan klasifikasi[6]. Arsitektur YoloV8 terdiri dari lapisan Convolution 2D, C2f(Cross Stage Partial Network), SPPF, Upsampling dan Concat. Gambar 1 merupakan detail arsitektur YOLOv8.



Gambar 1: Arsitektur YOLOv8.

D. Estimasi Pose

Estimasi pose adalah tugas menggunakan model pembelajaran mesin (*ML*) untuk memperkirakan pose seseorang dari gambar atau video dengan mengestimasi lokasi spasial sendi tubuh utama (titik kunci atau *keypoints*). Estimasi pose merujuk pada teknik visi komputer yang mendeteksi sosok manusia dalam gambar dan video, sehingga seseorang dapat menentukan, misalnya, di mana siku seseorang muncul dalam gambar. Penting untuk menyadari bahwa estimasi pose hanya

memperkirakan di mana sendi tubuh kunci berada dan tidak mengenali siapa yang ada dalam gambar atau video.[7]

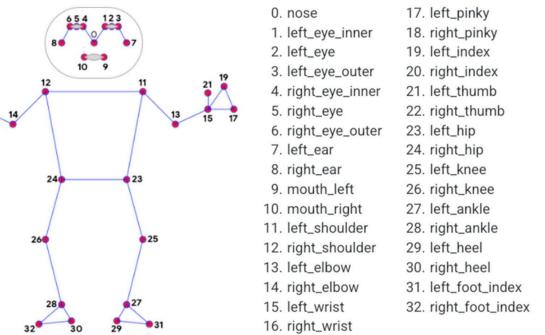
E. MediaPipe

MediaPipe adalah sebuah kerangka kerja yang dirancang oleh Google untuk membangun pipeline persepsi secara real-time. MediaPipe memungkinkan pengembang untuk mengintegrasikan berbagai jenis data sensor seperti video, dan data lainnya dalam satu platform yang efisien dan dapat dijalankan di berbagai perangkat, mulai dari mobile hingga desktop [8]

Kerangka kerja ini menggunakan konsep "graph" di mana setiap node di dalam graph merupakan sebuah "calculator" yang melakukan tugas spesifik, seperti deteksi objek, pelacakan pose, atau segmentasi gambar. Setiap node ini dapat dikonfigurasi melalui GraphConfig, yang mendeskripsikan topologi serta fungsionalitas dari node-node tersebut.

F. MediaPipe Pose

MediaPipe Pose (MPP), sebuah kerangka kerja lintas platform sumber terbuka yang disediakan oleh Google, digunakan untuk mendapatkan perkiraan koordinat sendi manusia 2D dalam setiap bingkai gambar. MediaPipe Pose membangun pipa dan memproses data kognitif dalam bentuk video menggunakan pembelajaran mesin (*machine learning - ML*). MPP menggunakan BlazePose yang mengekstrak 33 landmark 2D pada tubuh manusia seperti yang ditunjukkan pada Gambar 2. BlazePose adalah arsitektur pembelajaran mesin ringan yang mencapai kinerja real-time pada telepon seluler dan PC dengan inferensi CPU. Ketika menggunakan koordinat yang dinormalisasi untuk estimasi pose, rasio invers harus dikalikan dengan nilai piksel sumbu-y. [9]



Gambar 2: Pose MediaPipe

Precision

Precision merupakan rasio dimana TP (true positive) dimana jumlah positif benar dan FP (false positive) jumlah positif palsu sebagai metrik evaluasi dalam konteks machine learning, memberikan ukuran terhadap rasio prediksi positif yang tepat dibandingkan dengan seluruh prediksi positif yang diberikan oleh model. Dengan kata lain, precision 19 memberikan wawasan seberapa akurat model dalam membuat prediksi positif. Lebih rinci, precision mencerminkan seberapa sering model berhasil mengklasifikasikan instance sebagai positif dengan

benar dalam keseluruhan dataset. Nilai precision dapat mengindikasikan sejauh mana model mampu memberikan prediksi yang benar dalam konteks positif. Rentang nilai precision berada antara 0 dan 1, di mana nilai 1 menunjukkan bahwa semua prediksi positif model adalah benar, sementara nilai 0 menunjukkan bahwa tidak ada prediksi positif yang benar.

$$\frac{TP}{TP + FP} \quad (1)$$

Pada persamaan diatas menyajikan perbandingan antara prediksi positif yang tepat dengan total prediksi positif yang diberikan oleh model, memberikan pandangan yang lebih mendalam terkait kemampuan model dalam menghasilkan hasil yang benar dalam kategori yang diinginkan

Recall

Recall merupakan metrik yang digunakan untuk mengukur rasio dari data positif yang benar yang ditemukan dari seluruh data positif. Recall memberikan informasi tentang seberapa baik model machine learning menemukan semua data positif. Nilai recall berkisar antara 0 dan 1. Recall yang tinggi menunjukkan bahwa kelas yang dikenali dengan benar banyak, atau false negative yang didapatkan sedikit. Rumus dari recall dapat dilihat pada persamaan dibawah

$$\frac{TP}{TP + FN} \quad (2)$$

Recall merupakan rasio dimana TP (true positif) adalah jumlah positif benar dan FN (false negative) jumlah negatif palsu

Mean Average Precision (mAP)

Mean Average Precision (mAP) adalah sebuah metrik akurasi yang dihasilkan dari menghitung rata-rata dari Average Precision (AP) atau presisi rata-rata. AP sendiri diperoleh melalui perhitungan precision dan recall. Oleh karena itu, mAP dapat dianggap sebagai metrik evaluasi yang sangat informatif dalam mengevaluasi kinerja suatu sistem.

$$AP = \sum ((Recall_{n+1} - Recall_n) \times Precision_{interp}) \quad (3)$$

$$\times Recall_{n+1}) \quad (4)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (5)$$

G. Intersection over Union (IoU)

Intersection over Union, atau IoU, adalah metrik yang digunakan untuk mengevaluasi keakuratan posisi objek yang dideteksi oleh model dalam pemrosesan gambar. Prinsipnya adalah dengan menghitung area persinggungan antara kotak deteksi yang dihasilkan oleh model dengan kotak referensi yang merupakan standar emas atau Ground Truth. Rasio ini didapat dengan membandingkan area irisan kedua kotak tersebut terhadap keseluruhan area yang mereka cakup secara bersamaan. Jika kita membayangkan kedua kotak tersebut

sebagai satu kesatuan, maka IoU memberikan kita sebuah skor yang mengukur seberapa baik model kita dalam memprediksi lokasi objek sebenarnya. Semakin besar area persinggungan relatif terhadap total area gabungan, semakin tinggi nilai IoU, yang menandakan keakuratan prediksi yang lebih baik. Secara Sistematis, hal ini dituliskan sebagai :

$$Intersection over Union (IoU) = \frac{|A \cap B|}{|A \cup B|}. \quad (6)$$

H. Intel NUC

Intel NUC (Next Unit of Computing) adalah solusi komputasi yang kompak dan kuat yang dirancang oleh Intel untuk memenuhi berbagai kebutuhan komputasi, mulai dari hiburan rumah hingga gaming dan tugas profesional. Intel NUC dengan fitur prosesor Intel Core generasi dalam form factor kompak 4x4 inci. Dirancang untuk menawarkan kombinasi ukuran, kinerja, keberlanjutan, dan keandalan yang dibutuhkan oleh bisnis modern. Intel NUC model tertentu juga menyertakan teknologi Intel vPro® Enterprise dengan keamanan yang ditingkatkan. Mini PC ini dapat diupgrade dan diperbaiki, menjadikannya pilihan serbaguna untuk berbagai aplikasi bisnis termasuk komputasi klien, komputasi edge, dan digital signage.

I. ESP32 Devkit V1

ESP32 Devkit V1 adalah salah satu development board yang dibuat oleh DOIT untuk menjalankan modul ESP-WROOM-32 buatan Espressif. ESP32 Devkit dikenal dengan Development board yang kaya fitur dengan konektivitas Wi-Fi dan Bluetooth terintegrasi untuk beragam aplikasi. Devkit ini memiliki banyak pin yang memungkinkannya untuk diprogram dengan banyak tugas.

J. Motor Driver H-Bridge

Driver motor H-Bridge adalah rangkaian elektronik yang digunakan untuk mengontrol arah dan kecepatan motor DC. Cara kerjanya berdasarkan empat switch yang membentuk jembatan H (H-Bridge), yang mana dengan mengatur pembukaan dan penutupan switch-switch ini, kita dapat mengatur arah arus yang mengalir ke motor. Dengan demikian, kita bisa mengubah arah putaran motor DC. Driver Motor H-Bridge tersusun oleh sekumpulan transistor yang berfungsi sebagai pengendali motor, terutama yang memerlukan arus serta tegangan yang cukup besar. Selain itu, Rangkaian H-Bridge juga dapat memberikan fungsi pengereman pada motor dengan menghubungkan kedua terminal motor sehingga motor dapat berhenti lebih cepat. [10]

K. Kursi Roda Elektrik KY-123

Kursi roda Elektrik KY-123 merupakan alat bantu mobilitas yang terdiri dari struktur dasar kursi roda, sistem pengendalian gerak, mesin elektrik, dan modul baterai. Keunggulan alat ini terletak pada kemampuannya untuk dikendalikan dengan mudah dan nyaman, meminimalkan usaha fisik yang diperlukan pengguna dibandingkan dengan kursi roda manual. Ini sangat bermanfaat bagi individu dengan kondisi hemiplegia,

memungkinkan pengoperasian dengan satu tangan. Selain itu, kursi roda elektrik ini juga memberikan solusi mobilitas yang lebih baik bagi lansia yang mengalami keterbatasan dalam bergerak.

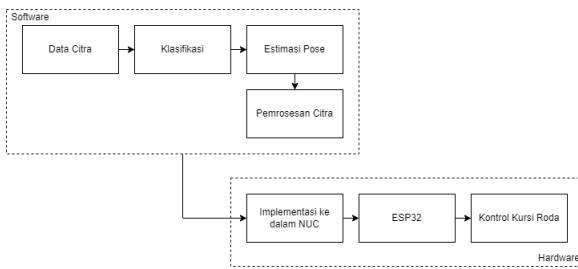


Gambar 3: Gambar Kursi Roda Elektrik KY-123

III. DESAIN DAN IMPLEMENTASI

A. Deskripsi Sistem

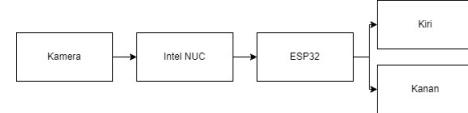
Penelitian dan pembuatan sistem ini diterapkan sesuai dengan desain dan implementasi pada bab ini. Desain sistem ini mencakup konsep pembuatan, perancangan, alur, dan implementasi infrastruktur yang dibuat dalam Blok Diagram. Desain dan penerapan diilustrasikan melalui penggunaan Gambar dan akan dijelaskan mulai dari pengumpulan data berupa citra, analisa dari model yang telah dibuat untuk mendeteksi objek Manusia, serta sistem yang menggunakan model tersebut seperti Gambar 4 dan dirincikan pada tiap subbab.



Gambar 4: Blok Diagram Sistem

B. Hardware

Perancangan hardware dilakukan sesuai dengan alur yang akan dideskripsikan pada sub- bab ini. Perangangan ini akan dipresentasikan dengan blok diagram alur yang telah merepresentasikan alur perancangan hardware ini. Gambar 5 merupakan blok diagram hardware yang akan ditampilkan sebagai berikut



Gambar 5: Blok Diagram Hardware

C. Software

Perancangan software dilakukan sesuai dengan alur yang akan dideskripsikan pada subbab ini. Perangangan ini akan dipresentasikan dengan blok diagram alur yang telah merepresentasikan alur perancangan software ini. Gambar 6 merupakan blok diagram alur akan ditampilkan sebagai berikut :

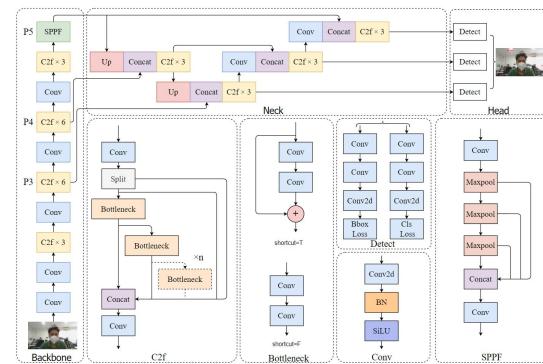


Gambar 6: Diagram Blok Software

D. Klasifikasi YoloV8

Dalam proses pengklasifikasian, setiap citra yang telah melalui proses Labeling akan dikenali dengan menggunakan YoloV8 yang telah ditraining untuk mengenali Manusia yang terdapat pada citra. Model akan mendapatkan memberikan output sesuai dengan kelas manusia, nantinya hasil dari klasifikasi akan dijadikan acuan posisi pada grid deteksi.

Model yang digunakan memiliki output kelas yaitu Manusia, adapun nilai yang diberikan oleh model yaitu berupa *Bounding Box* dan Nilai Konfiden (*Confidence Score*). Alur prosesnya dapat dilihat pada Gambar 7 yang merupakan blok diagram arsitektur.

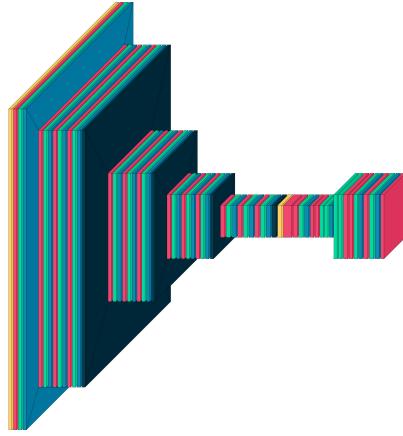


Gambar 7: Visualisasi Arsitektur YoloV8

Pada gambar 8 Warna biru menunjukkan lapisan convolutional yang ada dalam backbone, neck, dan head. Warna kuning menunjukkan C2f blok residual dengan convolution. Warna Hijau Muda menunjukkan SPPF lapisan Spatial Pyramid Pooling Fast. Merah Muda Menunjukkan lapisan UpSampling. Warna Ungu menunjukkan lapisan Concatenate untuk menggabungkan peta fitur.

TABEL I
TABEL KEYPOINT

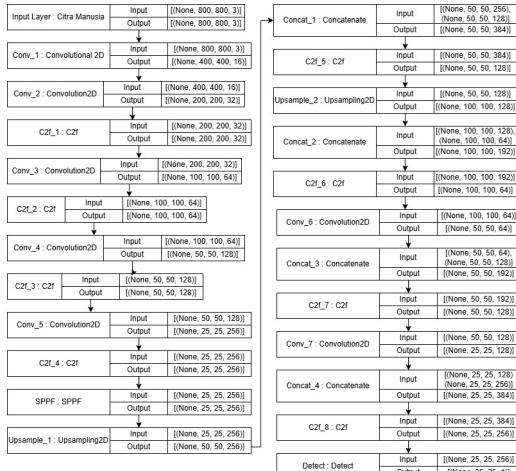
Nomor	Keypoint	Nama Keypoint
11		RIGHT_SHOULDER
12		LEFT_SHOULDER
14		RIGHT_ELBOW
16		RIGHT_WRIST



Gambar 8: Visualisasi Dengan VisualKeras

YoloV8 menggunakan Convolutional Neural Networks (CNN) sebagai dasar arsitekturnya. Dalam YOLO, jaringan CNN digunakan untuk ekstraksi fitur dari gambar input dan kemudian menerapkan jaringan lain untuk memprediksi Box unding Box dan kelas objek langsung dari gambar tersebut.

Berdasarkan salah satu pelatihan yang telah dilakukan, model ini memiliki 225 *Layer* atau lapisan, 3011043 parameter, dan 8.2 GFLOPs. Terdapat lapisan *Convolution 2D* (*Conv2D*), Blok C2f, Blok SPPF, Lapisan Upsampling, dan Lapisan Concat. Gambar 9 merepresentasikan setiap jenis layer dengan input output.



Gambar 9: Arsitektur YoloV8

E. Estimasi Pose MediaPipe

Dalam penelitian ini, beberapa landmark yang dianggap relevan adalah keypoint pada siku, lengan bawah, dan bahu kanan serta kiri. Titik-titik keypoint ini dipilih berdasarkan visibilitas dan konsistensi dalam deteksi. Tabel oxy menunjukkan nomor dan nama keypoint yang digunakan dalam estimasi pose.

F. Perhitungan Jarak

Salah satu cara untuk menghitung jarak dengan menggunakan bounding box adalah melalui konsep *focal length pixel* (f_p).

Fokus panjang dalam piksel, atau *focal length pixel*, adalah konversi dari fokus panjang lensa kamera yang biasanya diukur dalam milimeter ke dalam satuan piksel. Ini merupakan konsep kunci dalam fotogrametri dan visi komputer yang digunakan untuk menghubungkan informasi visual dari kamera ke ukuran fisik di dunia nyata.

Dalam konteks tugas akhir ini, fokus panjang dalam piksel digunakan untuk mengkonversi ukuran obstacle dari unit piksel menjadi unit meter. Hal ini penting karena kursi roda otonom perlu memahami jarak nyata ke obstacle untuk mengambil keputusan navigasi yang tepat. Rumusannya dapat dilihat pada Persamaan 7 :

$$f_p = \frac{D \times h_b}{H_o} \quad (7)$$

Di mana:

- f_p adalah *focal length* dalam piksel,
- D adalah jarak sebenarnya dari kamera ke objek,
- h_b adalah tinggi bounding box dalam piksel,
- H_o adalah tinggi sebenarnya dari objek.

Saat menghitung jarak ke objek, digunakan tinggi dari bounding box yang terdeteksi oleh YOLO, dikombinasikan dengan tinggi nyata objek yang diketahui dan fokus panjang dalam piksel. Sehingga membentuk Persamaan 8 sebagai berikut:

$$D = \frac{f_p \times H_o}{h_b} \quad (8)$$

Di sini:

- H_o merupakan tinggi rata-rata manusia atau objek lain yang diidentifikasi sebagai obstacle.

Pendekatan lain dalam menentukan jarak dengan menggunakan pose adalah melalui metode Jarak Euclidean. Jarak Euclidean dalam piksel adalah metode untuk mengukur jarak lurus antara dua titik dalam ruang gambar, yang biasanya diukur dalam piksel. Dalam konteks tugas akhir ini, yang melibatkan kursi roda otonom dengan integrasi MediaPipe, pengukuran ini sangat penting untuk berbagai fungsi, terutama dalam analisis pose dan penilaian proporsi objek dalam citra yang dihasilkan oleh kamera. Rumusannya dapat dilihat pada Persamaan 9

$$d_p = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times s_f \quad (9)$$

Di mana:

- d_p adalah jarak Euclidean dalam piksel,
- (x_1, y_1) dan (x_2, y_2) adalah koordinat dua titik yang diukur dalam piksel,
- s_f adalah faktor skala.

Rumus ini menghasilkan jarak antara dua titik dalam satuan yang sama dengan satuan koordinat x_1, y_1 dan x_2, y_2 . Biasanya, jika koordinat-koordinat ini diukur dalam piksel, maka jarak yang dihasilkan juga akan dalam piksel.

Dari perhitungan di atas, masih didapatkan jarak yang bernilai piksel. Dalam konteks perhitungan jarak, perlu menggunakan nilai standar dalam meter agar perhitungan tersebut sesuai dengan kaidah yang berlaku pada umumnya. Agar nilai piksel tersebut dapat diubah menjadi meter, perlu dilakukan kalibrasi menggunakan nilai K . Nilai K merupakan nilai kalibrasi berdasarkan pengukuran eksperimental. Sehingga didapat Persamaan 10 sebagai berikut:

$$D_m = \left(\frac{K}{d_p} \right) \quad (10)$$

Di mana:

- D_m adalah jarak dalam meter,
- K adalah konstanta kalibrasi,
- d_p adalah jarak dalam piksel.

Nilai K menentukan seberapa besar pengaruh jarak dalam piksel terhadap jarak dalam meter. Nilai yang lebih besar atau lebih kecil akan secara langsung mempengaruhi hasil perhitungan jarak. Misalnya, nilai K yang lebih besar akan menghasilkan jarak yang lebih kecil untuk jumlah piksel yang sama. Nilai ini digunakan dalam konteks yang sangat spesifik di mana parameter tersebut menggambarkan hubungan langsung antara ukuran piksel dan jarak atau dimensi nyata, berdasarkan asumsi spesifik tentang geometri scene dan karakteristik kamera.

Nilai ini harus dikalibrasi secara akurat agar sesuai dengan karakteristik spesifik kamera dan setup yang digunakan. Kalibrasi yang tidak tepat akan menghasilkan pengukuran jarak yang tidak akurat, yang dapat berdampak pada keputusan navigasi kursi roda otonom. Berikut rumus untuk pengkalibrasian nilai K menggunakan Persamaan 11:

$$K = J_o \times U_p \quad (11)$$

Di mana:

- J_o adalah jarak nyata objek,
- U_p adalah ukuran objek dalam piksel.

Dalam perhitungan lebar obstacle, digunakan perhitungan jarak Euclidean, namun dengan konteks aplikasi yang berbeda. Kedua pendekatan ini berbeda dalam aplikasi dan konteksnya. Perbedaan tersebut dapat dilihat pada Persamaan 15

$$l_p = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times s_f \quad (12)$$

Nilai output yang didapatkan adalah lebar manusia dalam satuan piksel. Untuk memetakan lebar dari piksel ke meter, perlu dibuat rumus konversi yang menentukan ukuran dalam piksel (yang merupakan ukuran digital dan relatif) ke ukuran dunia nyata (meter). Rumusannya dapat dilihat pada Persamaan 16:

$$l_m = l_p \times s_f \quad (13)$$

Dengan mengalikan lebar objek dalam piksel dengan faktor skala, hasilnya adalah lebar objek dalam meter. Rumus ini berguna dalam aplikasi di mana perlu mengetahui dimensi fisik objek dalam dunia nyata untuk membuat keputusan atau pengukuran yang tepat.

Faktor skala adalah nilai yang mengonversi ukuran dari unit piksel ke unit meter. Nilai ini diperoleh melalui proses kalibrasi. Faktor skala menentukan berapa meter yang diwakili oleh setiap piksel dalam gambar, berdasarkan jarak kamera ke objek dan pengaturan kamera lainnya seperti fokus panjang. Selain itu, perlu diketahui bahwa faktor skala yang digunakan pada rumus ini berbeda dengan rumus jarak yang sebelumnya dijabarkan. Berikut rumus untuk mengkalibrasi faktor skala dalam konteks lebar objek pada Persamaan 17:

$$s_f = \frac{D_n}{U_p} \quad (14)$$

Di mana:

- D_n adalah dimensi nyata rata-rata,
- U_p adalah ukuran dalam piksel rata-rata.

Dalam perhitungan lebar obstacle, digunakan perhitungan jarak Euclidean, namun dengan konteks aplikasi yang berbeda. Kedua pendekatan ini berbeda dalam aplikasi dan konteksnya. Perbedaan tersebut dapat dilihat pada Persamaan 15

$$l_p = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times s_f \quad (15)$$

Nilai output yang didapatkan adalah lebar manusia dalam satuan piksel. Untuk memetakan lebar dari piksel ke meter, perlu dibuat rumus konversi yang menentukan ukuran dalam piksel (yang merupakan ukuran digital dan relatif) ke ukuran dunia nyata (meter). Rumusannya dapat dilihat pada Persamaan 16:

$$l_m = l_p \times s_f \quad (16)$$

Dengan mengalikan lebar objek dalam piksel dengan faktor skala, hasilnya adalah lebar objek dalam meter. Rumus ini berguna dalam aplikasi di mana perlu mengetahui dimensi fisik objek dalam dunia nyata untuk membuat keputusan atau pengukuran yang tepat.

Faktor skala adalah nilai yang mengonversi ukuran dari unit piksel ke unit meter. Nilai ini diperoleh melalui proses kalibrasi. Faktor skala menentukan berapa meter yang diwakili oleh setiap piksel dalam gambar, berdasarkan jarak kamera ke objek dan pengaturan kamera lainnya seperti fokus panjang. Selain itu, perlu diketahui bahwa faktor skala yang digunakan pada rumus ini berbeda dengan rumus jarak yang sebelumnya dijabarkan. Berikut rumus untuk mengkalibrasi faktor skala dalam konteks lebar objek pada Persamaan 17:

$$s_f = \frac{D_n}{U_p} \quad (17)$$

Di mana:

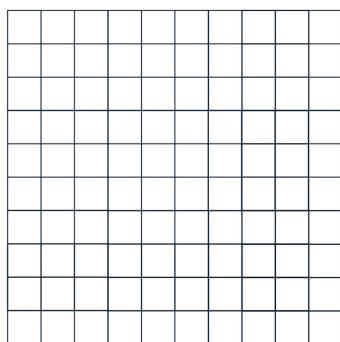
- D_n adalah dimensi nyata rata-rata,
- U_p adalah ukuran dalam piksel rata-rata.

Nilai-nilai ini dapat digunakan dalam kode untuk mengonversi ukuran dalam piksel ke ukuran nyata berdasarkan pengukuran yang telah dikalibrasi. Kalibrasi harus dilakukan dengan baik dan benar untuk mendapatkan hasil yang akurat.

G. Implementasi Grid

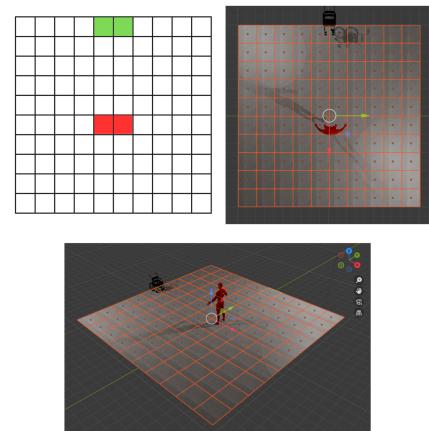
Dalam tugas akhir ini, kursi roda otonom harus dapat mengetahui posisi obstacle yang akan dilaluinya. Oleh karena itu, diperlukan sebuah peta yang dapat digunakan sebagai acuan kursi roda untuk mengambil tindakan berdasarkan jarak obstacle dan lebar obstacle yang akan menjadi acuan untuk menjawab permasalahan, yaitu di titik mana kursi roda harus menghindar dan apakah obstacle berhasil dihindari. Grid menjadi salah satu pendekatan terbaik dalam memetakan hasil deteksi. Penggunaan grid tidak hanya mempermudah pengambilan tindakan, tetapi juga memberikan visualisasi yang mudah dimengerti. Ukuran grid juga dapat disesuaikan sesuai dengan kebutuhan, baik dimensi maupun tampilannya.

Setelah perhitungan rumus di atas diimplementasikan dalam kode, akan didapatkan beberapa variabel penting yang akan digunakan dalam memetakan posisi maupun ukuran obstacle dalam grid. Sebelum itu, berikut adalah tampilan grid yang digunakan dapat dilihat pada Gambar 10 sebagai berikut:



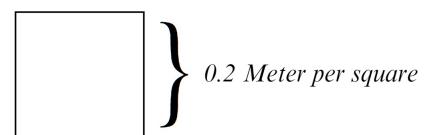
Gambar 10: Grid 10x10.

Grid dibuat menggunakan library OpenCV. Library ini digunakan atas dasar implementasinya yang mudah dan ringan. Agar Lebih mudah dimengerti, penjelasan akan disertai dengan visualisasi 3d melalui blender pada Gambar 11.



Gambar 11: Visualisasi Grid dengan blender.

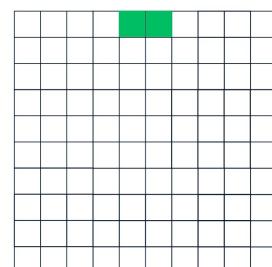
Penggunaan grid 10x10 didasari atas hasil citra dan performa model. Baik bounding box maupun pose memiliki keterbatasan di mana hasil perhitungannya tidak melebihi parameter yang akan ditetapkan pada grid. Parameter tersebut dapat dilihat pada Gambar 12 sebagai berikut:



Gambar 12: Parameter untuk setiap kotak grid.

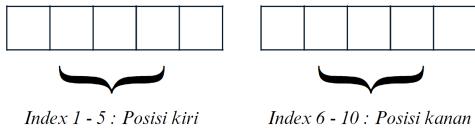
Dari gambar di atas, setiap kotak dalam grid bernilai 0.2 meter baik dalam posisi vertikal maupun horizontal. Penentuan nilai 0.2 meter per kotak ini didasarkan pada hasil kalibrasi dan pengukuran eksperimental yang memastikan ukuran ini optimal untuk deteksi obstacle yang akurat dan efisien.

Untuk memetakan objek dengan baik dalam grid, perlu dibuat indeks untuk mengetahui posisi relatif kiri dan kanan objek terhadap kamera. Untuk 10 kotak horizontal, akan dibagi menjadi indeks kiri dan kanan, di mana indeks 1 sampai 5 dikategorikan sebagai indeks kiri, dan indeks 6 hingga 10 dikategorikan sebagai indeks kanan. Pengambilan keputusan ini terkait dengan posisi kursi roda statis pada grid, yang digambarkan pada Gambar 13 sebagai berikut:



Gambar 13: Posisi Kursi Pada grid.

Dapat dilihat bahwa posisi kursi roda mengambil 2 kotak grid hijau pada bagian (5,1) dan (6,1). Keputusan ini di dasarkan pada lebar kursi roda yang sesuai dengan ukuran grid. Posisi atas menentukan indeks mana yang merupakan bagian kiri maupun kanan dalam grid. Dengan posisi atas yang ditetapkan sebagai posisi konstan kursi roda dan hasil input citra yang didapatkan berlawanan dengan hasil deteksi (mirror), maka indeks dapat digambarkan sesuai Gambar 14 sebagai berikut:



Gambar 14: Kategori Posisi berdasarkan Index.

Pengkategorian ini berperan penting dalam pengambilan keputusan belok kursi roda, di mana nantinya hasil deteksi yang berupa jarak, lebar, serta posisi relatif akan ditampilkan pada grid.

H. Navigasi Penghindaran Kursi Roda

1) Hasil Tanpa Deteksi: Pada kondisi ini tidak ada objek yang terdeteksi, yang berarti tidak ada obstacle yang menghalangi jalan kursi roda oleh karena itu kursi roda akan terus bergerak maju. Pada kondisi ini bounding box, pose, grid dan lainnya tidak ditampilkan karena tidak ada manusia yang terdeteksi. dapat dilihat pada contoh gambar 15



Gambar 15: Contoh kondisi tidak terdeteksi Manusia.

Dapat dilihat pada gambar 15, terdapat teks "Manusia Tidak Terdeteksi" yang berarti tidak ada objek manusia yang terdeteksi.

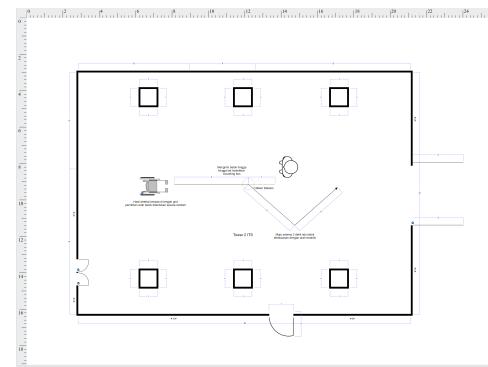
2) Hasil deteksi objek pada grid menunjukkan Index Kiri Lebih besar dari Index kanan: Pada kondisi ini posisi hasil deteksi menunjukkan nilai yang lebih besar pada Index kiri (Index 1-5) yang berarti bahwa objek sedang berada pada sebelah kiri posisi terhadap kursi roda. Dapat dilihat pada contoh gambar dibawah.



Gambar 16: Contoh kondisi Index Kiri >Kanan .

Dapat dilihat pada gambar 16, grid nilainya lebih mengarah index kiri dari pada kanan. Dimana dilihat dari pertimbangan posisi tersebut maka kursi roda akan menghindar ke Kanan yang merupakan belokan yang lebih aman ketimbang belokan ke kiri.

Kursi roda baru dapat dikatakan menghindar jika dapat kembali ke arah asal, dengan demikian Gambar 17 adalah gambar skematis penghindaran yang akan dilalui dalam kondisi ini.



Gambar 17: Skematik penghindaran pada kondisi Index Kiri >Kanan.

Dapat dilihat pada Gambar 17 saat kursi roda sudah pada jarak 1 meter deteksi, maka akan berbelok sesuai dengan kondisi index dan menyimpan arah belokan ini. Apabila sudah tidak tergambar bounding box maka kursi roda akan maju selama 4 detik, lalu mengecek arah belokan terakhir. Setelah itu kursi roda akan Belok lagi sesuai dengan arah terakhir selama 5 detik, lalu mereset kondisi arah terakhir. Terakhir pada kondisi ini kursi roda akan masuk ke kondisi tanpa deteksi, sehingga ia akan terus bergerak maju hingga deteksi selanjutnya.

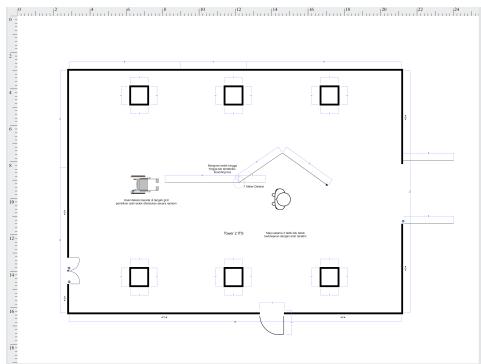
3) Hasil deteksi objek pada grid menunjukkan index Kanan lebih besar dari Index kiri: Pada kondisi ini posisi hasil deteksi menunjukkan nilai yang lebih besar pada index kanan (6-10) yang berarti bahwa objek sedang berada pada sebelah kanan posisi terhadap kursi roda. dapat dilihat pada contoh gambar 18



Gambar 18: Contoh kondisi Index Kanan >kiri

Dapat dilihat pada gambar 18, grid nilainya lebih mengarah index kanan dari pada kiri. Dimana dilihat dari pertimbangan posisi tersebut maka kursi roda akan menghindar ke kiri yang merupakan belokan yang lebih aman ketimbang belokan ke kanan.

Kursi roda baru dapat dikatakan menghindar jika dapat kembali ke arah asal, dengan demikian Gambar 19 adalah gambar skematis penghindaran yang akan dilalui dalam kondisi ini.



Gambar 19: Skematis penghindaran pada Contoh kondisi Index Kanan >Kiri.

Dapat dilihat pada Gambar 19 saat kursi roda sudah pada jarak 1 meter deteksi, maka akan berbelok sesuai dengan kondisi index dan menyimpan arah belokan ini. Apabila sudah tidak tergambar bounding box maka kursi roda akan maju selama 4 detik, lalu mengecek arah belokan terakhir. Setelah itu kursi roda akan Belok lagi sesuai dengan arah terakhir selama 5 detik, lalu mereset kondisi arah terakhir. Terakhir pada kondisi ini kursi roda akan masuk ke kondisi tanpa deteksi, sehingga ia akan terus bergerak maju hingga deteksi selanjutnya.

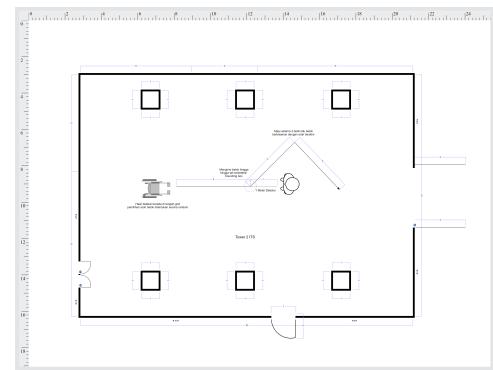
4) Hasil deteksi objek pada grid menunjukkan posisi Linear terhadap kursi roda: Pada kondisi ini perlu ditambahkan sebuah perintah untuk pengambilan keputusan dimana posisi belok pada kondisi ini baik melalui kanan maupun kiri tidak akan memiliki kelebihan/kekurangan karena dalam posisi ini nilai index sama besarnya baik kanan maupun kiri. sehingga perlu dilakukan pendekatan yang baik agar pengambilan keputusan ini tidak menimbulkan kesalahan. Pada tugas proyek ini pengambilan keputusan ini didasari pada nilai random. Sehingga keputusan yang diambil bisa ke kanan maupun ke kiri.



Gambar 20: Contoh kondisi Linier.

Dapat dilihat pada gambar 20, gridnya sejajar dengan posisi kursi roda. Sehingga penggunaan random akan sangat berguna untuk mengambil keputusan apabila menghadapi kasus seperti ini.

Kursi roda baru dapat dikatakan menghindar jika dapat kembali ke arah asal, dengan demikian Gambar 21 adalah gambar skematis penghindaran yang akan dilalui dalam kondisi ini.



Gambar 21: Skematis penghindaran pada Contoh kondisi Linier.

Dapat dilihat pada Gambar 21 saat kursi roda sudah pada jarak 1 meter deteksi, maka akan berbelok sesuai dengan kondisi index dan menyimpan arah belokan ini. Apabila sudah tidak tergambar bounding box maka kursi roda akan maju selama 4 detik, lalu mengecek arah belokan terakhir. Setelah itu kursi roda akan Belok lagi sesuai dengan arah terakhir selama 5 detik, lalu mereset kondisi arah terakhir. Terakhir pada kondisi ini kursi roda akan masuk ke kondisi tanpa deteksi, sehingga ia akan terus bergerak maju hingga deteksi selanjutnya.

IV. HASIL DAN PEMBAHASAN

A. Pengujian FPS

Dapat dilihat pada tabel II pada bagian kanan, didapatkan pada pengujian nilai fps laptop ini nilai rata - rata FPS sebesar 13.140. Dimana nilai FPS tertinggi adalah sebesar 13.23 dan FPS paling rendah ialah 13.05. Selain itu dapat dilihat pada tabel II, didapatkan pada pengujian ini nilai rata - rata FPS pada intel NUC sebesar 6.111. Dimana nilai FPS tertinggi yang didapatkan ialah 6.47 dan FPS paling rendah ialah 5.54.

TABEL II
HASIL PERBANDINGAN FPS PADA LAPTOP DAN NUC

	Laptop	NUC
Rata-rata FPS	13,14	6,11
FPS Maksimum	13,23	6,47
FPS Minimum	13,05	5,54

B. Pengujian Berdasarkan Response Time

Berdasarkan output diatas dapat dihitung Response Time sistem yang akan jabarkan pada tabel III Hasil Response Time akan diuji untuk mendapatkan waktu yang dibutuhkan untuk melakukan pendekatan dengan model yang kemudian diklasifikasi dan dikirim ke ESP32 hingga motor kursi roda mulai bergerak. Pengujian ini dilakukan secara real time pada perangkat NUC, perhitungan delay didapatkan dari mulai dikirim hingga berhentinya motor pada kursi roda. sedangkan perhitungan inference time dimulai dari dimulainya proses prediksi hingga didapatkan hasil dari proses klasifikasi. Rata - rata waktu delay yang didapatkan adalah 0.2494 detik dari data yang sudah didapatkan dari pengujian NUC, adapun hasilnya dapat dilihat pada tabel dibawah. Adapun nilai inference rata-rata yang didapatkan adalah 139.4899 ms atau 0.1394899 detik

TABEL III
HASIL DELAY

	per second
Rata-rata Delay	0,249
Delay Maksimum	0,379
FPS Minimum	0,145

TABEL IV
HASIL INFERENCE

	per milisecond
Rata-rata Inference	139,489
Inference Maksimum	181,100
Inference Minimum	123,100

C. Pengujian Kesesuaian Jarak Deteksi

Pengujian ini dilakukan pengetesan terhadap model dalam menghasilkan jarak berdasarkan hasil perhitungan pada *Bounding Box* dan pose. Pengujian ini dilakukan dengan membandingkan jarak objek asli dengan jarak yang dihasilkan sistem pada Intel NUC terhadap manusia yang berdiri tegak. Kalibrasi telah dilakukan pada jarak 150cm, jarak ini diambil atas dasar visibilitas pose dan bounding box. Sehingga didapatkan nilai Focal Length sebesar 480, serta nilai K1 dan K2 sebesar 10.922 dan 24.222. Nilai-nilai tersebut akan digunakan dalam pengujian kesesuaian jarak pada 150cm, 100cm dan 50cm. Adapun tujuan dilakukan pengujian ini untuk menguji kemampuan sistem dalam mengukur jarak.

Pengujian ini dilakukan menggunakan alat ukur meteran yang ditancapkan pada kamera dan ditarik menuju peneliti untuk mendapatkan jarak. Nilai tersebut akan dihitung untuk diambil nilai rata rata *difference* atau perbedaan yang dihasilkan dari sistem terhadap pengukuran nyata. Berikut merupakan nilai rata-rata selisih jarak dari masing masing pengujian.

TABEL V
RINGKASAN HASIL PENGUJIAN KESESUAIAN JARAK DETEKSI

Distance	Yolo Bbox	MediaPipe Shoulder	MediaPipe Hand
150cm	3.2 cm	5.06 cm	23.8 cm
100cm	20.8 cm	2.2 cm	3.53 cm
50cm	69.8 cm	14.8 cm	1.93 cm

Pada tabel V didapatkan pada masing - masing jarak nilai error terbesar pada 150cm ialah Landmark lengan dengan presentase 15,86%, dan terkecil pada Yolo Bbox pada 2,13%. Nilai error terbesar pada jarak 100cm adalah Yolo Bbox dengan presentase 20,8% , dan terkecil pada landmark Bahu dengan presentase 2,2%. Nilai error terbesar pada jarak 50cm adalah Yolo Bbox dengan presentase sebesar 139%, dan terkecil landmark lengan dengan presentase 3,86%.

TABEL VI
TABEL PERFORMA KEBERHASILAN PENGHINDARAN

Percobaan	Hasil
1	Kursi Roda Berhasil Menghindar
2	Kursi Roda Berhasil Menghindar
3	Kursi Roda Berhasil Menghindar
4	Kursi Roda Berhasil Menghindar
5	Kursi Roda Berhasil Menghindar
6	Kursi Roda Berhasil Menghindar
7	Kursi Roda Berhasil Menghindar
8	Kursi Roda Berhasil Menghindar
9	Kursi Roda Berhasil Menghindar
10	Kursi Roda Berhasil Menghindar
11	Kursi Roda Berhasil Menghindar
12	Kursi Roda Berhasil Menghindar
13	Kursi Roda Berhasil Menghindar
14	Kursi Roda Berhasil Menghindar
15	Kursi Roda Berhasil Menghindar
16	Kursi Roda Berhasil Menghindar
17	Kursi Roda Berhasil Menghindar
18	Kursi Roda Berhasil Menghindar
19	Kursi Roda Berhasil Menghindar
20	Kursi Roda Berhasil Menghindar
21	Kursi Roda Berhasil Menghindar
22	Kursi Roda Berhasil Menghindar
23	Kursi Roda Berhasil Menghindar
24	Kursi Roda Berhasil Menghindar
25	Kursi Roda Berhasil Menghindar
26	Kursi Roda Berhasil Menghindar
27	Kursi Roda Berhasil Menghindar
28	Kursi Roda Berhasil Menghindar
29	Kursi Roda Berhasil Menghindar
30	Kursi Roda Berhasil Menghindar

Pada Tabel VI Hasil yang didapatkan Penghindaran berhasil sebanyak 30 kali. Sehingga didapatkan presentase keberhasilan yang didapatkan dari hasil uji ini sebesar 100%. Hasil ini membuktikan sistem yang dibuat mampu untuk mendekteksi dan menghindari manusia dengan baik.

D. Performa Akurasi Penghindaran

Pengujian akurasi jarak penghindaran dilakukan dengan mengukur perbandingan antara jarak pengukuran saat kursi roda melakukan penghindaran yang dihasilkan sistem dan jarak kursi roda saat penghindaran terhadap Manusia pada dunia nyata. Jarak yang ditetapkan dalam Penghindaran ialah

pada 100 cm atau 1 meter, yang berarti kursi roda harus menghindar pada jarak yang ditentukan apabila ada Manusia yang terdeteksi.

TABEL VII
RINGKASAN HASIL PERFORMA AKURASI PENGHINDARAN

<i>Distance set</i>	<i>Real Error</i>	<i>System Error</i>
100cm	33.1 cm	29.3 cm

Dapat dilihat pada Tabel VII didapatkan rata- rata error pengukuran nyata sebesar 33,1 cm dan rata rata error pengukuran sistem sebesar 29,3. Dimana hasil yang didapatkan tidak sesuai dengan jarak yang ditetapkan yaitu 100cm atau 1 meter, dan cenderung mengalami penurunan akurasi.

Penurunan ini disebabkan oleh beberapa faktor, yaitu posisi kamera yang terguncang seiring pengujian, delay pada sistem, laptop yang tidak di charge membuat pemakaian gpu menjadi terbatas dan penurunan performa laptop selama pengujian yang diakibatkan oleh panas laptop yang meningkat seiring dengan waktu pengujian menyebabkan menurunnya FPS yang didapatkan.

E. Performa Keberhasilan Penghindaran dengan Dua Obstacle

TABEL VIII

TABEL PERFORMA KEBERHASILAN PENGHINDARAN DUA OBSTACLE

Percobaan	Hasil
1	Kursi Roda Berhasil Menghindar
2	Kursi Roda Berhasil Menghindar
3	Kursi Roda Berhasil Menghindar
4	Kursi Roda Berhasil Menghindar
5	Kursi Roda Berhasil Menghindar
6	Kursi Roda Berhasil Menghindar
7	Kursi Roda Berhasil Menghindar
8	Kursi Roda Berhasil Menghindar
9	Kursi Roda Berhasil Menghindar
10	Kursi Roda Berhasil Menghindar

Berdasarkan Hasil yang didapatkan Penghindaran berhasil sebanyak 10 kali. Sehingga didapatkan persentase yang didapatkan dari hasil uji ini sebesar 100%. Hasil ini membuktikan sistem yang dibuat mampu untuk mendeteksi dan menghindari dua obstacle manusia.

F. Performa Keberhasilan Penghindaran dengan Tiga Obstacle

TABEL IX

TABEL PERFORMA KEBERHASILAN PENGHINDARAN TIGA OBSTACLE

Percobaan	Hasil
1	Kursi Roda Berhasil Menghindar
2	Kursi Roda Berhasil Menghindar
3	Kursi Roda Berhasil Menghindar
4	Kursi Roda Berhasil Menghindar
5	Kursi Roda Berhasil Menghindar

Berdasarkan Hasil yang didapatkan Penghindaran berhasil sebanyak 5 kali. Sehingga didapatkan persentase yang didapatkan dari hasil uji ini sebesar 100%. Hasil ini membuktikan sistem yang dibuat mampu untuk mendeteksi dan menghindari tiga obstacle manusia.

G. Performa Keberhasilan Penghindaran dengan Empat Obstacle

TABEL X
TABEL PERFORMA KEBERHASILAN PENGHINDARAN
EMPAT OBSTACLE

Percobaan	Hasil
1	Kursi Roda Berhasil Menghindar
2	Kursi Roda Berhasil Menghindar
3	Kursi Roda Berhasil Menghindar

Berdasarkan Hasil yang didapatkan Penghindaran berhasil sebanyak 3 kali. Sehingga didapatkan persentase yang didapatkan dari hasil uji ini sebesar 100%. Hasil ini membuktikan sistem yang dibuat mampu untuk mendeteksi dan menghindari tiga obstacle manusia.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

- 1) Model dengan Metrics tertinggi yang telah di-training dengan berbagai konfigurasi adalah model dengan skor mAP di IoU 0.5 tertinggi sebesar 81.85%. Adapun nilai yang didapatkan ini sudah cukup baik untuk melakukan penghindaran dilihat pada hasil performa penghindaran yang sangat baik.
- 2) Performa NUC dalam pengujian FPS menghasilkan Nilai yang lebih rendah ketimbang Laptop pribadi Penulis dengan selisih 7.029 fps.
- 3) Rata - rata delay yang didapatkan pada pengujian adalah sekitar 0.2494 detik dan rata- rata nilai inference yang didapatkan 139.4899 ms atau 0.1394 detik.
- 4) Hasil menunjukkan bahwa deteksi menggunakan *Bounding box* dan Landmark bahu lebih akurat pada jarak yang lebih jauh (150 cm dan 100cm), sedangkan landmark lengan lebih akurat pada jarak yang lebih dekat (50 cm). Dimana rata-rata *difference* terbaik bounding box yaitu pada jarak 150cm sebesar 3.2 cm, rata-rata *difference* landmark bahu terbaik yaitu pada jarak 100cm sebesar 2.2 cm dan rata-rata *difference* landmark lengan terbaik yaitu pada jarak 50cm sebesar 1.93 cm.
- 5) Hasil Performa Deteksi menunjukkan hasil yang memuaskan dalam 30 sampel pengujian. Dengan persentase keberhasilan sebesar 100% yang menunjukkan bahwa sistem yang dibuat dapat menghindari manusia dengan sangat baik.

VI. SARAN

Untuk pengembangan lebih lanjut pada penelitian selanjutnya, adapun saran yang bisa diberikan antara lain:

- 1) Variasi dataset yang lebih ditingkat untuk meningkatkan performa deteksi
- 2) Menggunakan Device yang memiliki performa yang lebih baik untuk fps yang lebih tinggi
- 3) Meningkatkan performa grid deteksi dengan melakukan penyesuaian yang lebih detail untuk pemetaan yang lebih baik lagi.
- 4) Menggunakan pendingin device saat melakukan pengujian di ruangan terbuka agar tidak mengalami penurunan performa.

PUSTAKA

- [1] K. Daring, “Kbbi vi daring,” 2016. [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/lumpuh>
- [2] P. Pansawira, “Kelumpuhan - gejala, penyebab, dan mengobati - alodokter,” 4 2022. [Online]. Available: <https://www.alodokter.com/kelumpuhan#:~:text=Kondisi%20ini%20dapat%20disebabkan%20oleh,Penanganan%20kelumpuhan%20tergantung%20pada%20penyebabnya>
- [3] J. H. Choi, Y. Chung, and S. Oh, “Motion control of joystick interfaced electric wheelchair for improvement of safety and riding comfort,” *Mechatronics*, vol. 59, pp. 104–114, 2019.
- [4] L. Lecrosnier, R. Khemmar, N. Ragot, B. Decoux, R. Rossi, N. Kefi, and J.-Y. Ertaud, “Deep learning-based object detection, localisation and tracking for smart wheelchair healthcare mobility,” *International journal of environmental research and public health*, vol. 18, no. 1, p. 91, 2021.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [6] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, p. 1680–1716, Nov. 2023. [Online]. Available: <http://dx.doi.org/10.3390/make5040083>
- [7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [8] C. Lugaressi, J. Tang, H. Nash, C. McClanahan, E. Ubowea, M. Hays, F. Zhang, C. Chang, M. G. Yong, J. Lee, W. Chang, W. Hua, M. Georg, and M. Grundmann, “Mediapipe: A framework for building perception pipelines,” *CoRR*, vol. abs/1906.08172, 2019. [Online]. Available: <http://arxiv.org/abs/1906.08172>
- [9] J.-W. Kim, J.-Y. Choi, E.-J. Ha, and J.-H. Choi, “Human pose estimation using mediapipe pose and optimization method based on a humanoid model,” *Applied Sciences*, vol. 13, no. 4, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/4/2700>
- [10] Muhammad, “Analisis driver h-bridge menggunakan relay pada motor bldc konstruksi axial flux (celah ganda),” 2018. [Online]. Available: <https://repository.unej.ac.id/bitstream/handle/123456789/89217/Muhammad%20-%202016191020115.pdf?sequence=1&isAllowed=y>