

PENGEMBANGAN KURSI RODA OTONOM BERBASIS YOLOV8 UNTUK PENGHINDARAN OBSTACLE

I Gst Ngr Agung Hari Vijaya Kusuma
Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
0721194000073@student.its.ac.id

Dr. Eko Mulyanto Yuniarno, S.T.,M.T.
Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
ekomulyanto@ee.its.ac.id

Abstrak—Pengembangan kursi roda otonom telah menjadi semakin penting dalam memberikan mobilitas dan kemandirian yang ditingkatkan 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 Single Based Computer Intel NUC dan Kamera. Obstacle yang dideteksi membuat SBC mengirim perintah ke ESP32 untuk menjalankan motor untuk melakukan manuver penghindaran. Evaluasi sistem meliputi pengujian komprehensif di berbagai lingkungan dan konteks navigasi untuk memvalidasi efektivitas dan kehandalan dalam aplikasi dunia nyata. Penelitian ini berkontribusi pada kemajuan teknologi kursi roda otonom, dan keselamatan bagi individu dengan keterbatasan mobilitas.

Kata kunci—Deep Learning

I. PENDAHULUAN

Lumpuh, sebagaimana didefinisikan oleh Kamus Besar Bahasa Indonesia, adalah keadaan di mana fungsi anggota badan melemah sehingga tidak bertenaga atau tidak dapat digerakkan lagi sebagaimana mestinya [1]. Otot, tulang, saraf, dan jaringan penghubung antara ketiganya memiliki peran yang krusial dalam mengendalikan gerak tubuh manusia. Gangguan pada salah satu dari komponen ini bisa menyebabkan kelumpuhan, baik yang bersifat sementara maupun permanen.

Beberapa penyakit dan kondisi dapat memicu kelumpuhan, termasuk stroke yang menyebabkan kelumpuhan pada bagian wajah, lengan, dan kaki sebelah, *Bell's Palsy* yang menyebabkan kelumpuhan pada satu sisi wajah tanpa melibatkan anggota tubuh lainnya, trauma kepala yang menyebabkan kelumpuhan pada area tubuh yang sesuai dengan kerusakan otak, serta polio yang menyerang lengan, kaki, dan otot pernapasan, di antara banyak kondisi lain yang dapat memicu kelumpuhan [2].

Individu yang mengalami kelumpuhan sering kali menghadapi tantangan mobilitas dalam kehidupan sehari-hari. Mereka umumnya membutuhkan alat bantu, seperti kursi roda, untuk beraktivitas. Saat ini, kursi roda elektrik yang dioperasikan dengan joystick telah tersedia, namun alat ini sering kali tidak

memadai untuk orang yang mengalami kelumpuhan lengan [3].

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. Penelitian oleh Lecrosnier mengenai aplikasi YOLOv3 dalam pengembangan kursi roda otonom untuk menghindari pintu menunjukkan keberhasilan metode ini dalam memperbaiki navigasi dan keamanan pengguna [4].

YOLOv8, dengan kemampuan deteksi objek yang ditingkatkan dan akurasi yang tinggi dalam real-time, menawarkan kemungkinan besar untuk pengembangan lebih lanjut dalam kursi roda otonom. Oleh karena itu, penelitian dengan judul "Pengembangan Kursi Roda Otonom Berbasis YOLOv8 untuk Penghindaran Obstacle" diharapkan bisa mengurangi risiko kecelakaan yang berhubungan dengan kursi roda, memberikan solusi efektif untuk navigasi dan keamanan.

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.

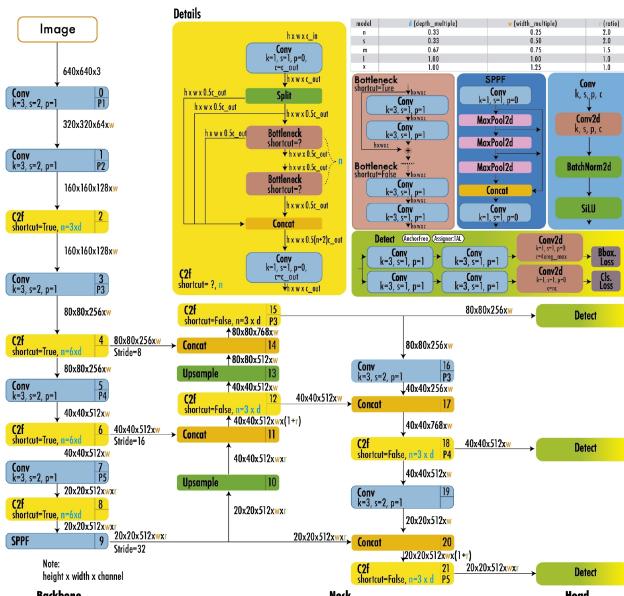
B. YOLO (You Only Look Once)

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]



Gambar 1: Arsitektur YOLOv8.

D. Estimasi Pose

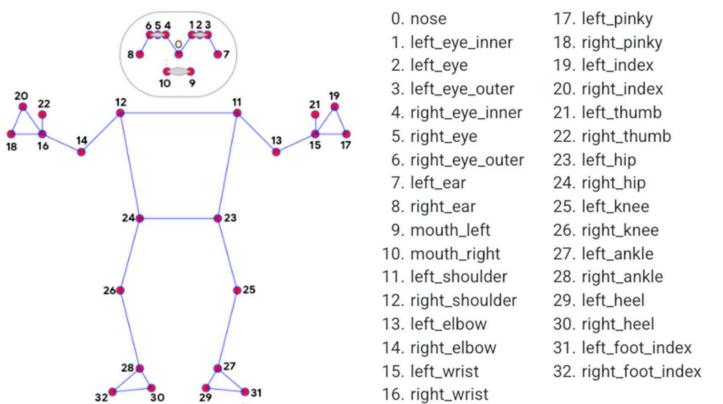
Estimasi pose adalah tugas menggunakan model pembelajaran mesin (*ML*) untuk memperkirakan pose seseorang dari gambar atau video dengan mengestimasi lokasi spasial sendi tubuh utama (titik kunci atau *keypoints*). Estimasi pose merujuk pada teknik visi komputer yang mendeteksi sosok manusia dalam gambar 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 kerangka kerja untuk membangun pipa yang melakukan inferensi dari data sensorik apa pun. Dengan MediaPipe, pipa persepsi dapat dibangun sebagai graf dari komponen modular, termasuk inferensi model, algoritma pengolahan media, dan transformasi data, dll. Data sensorik seperti aliran audio dan video memasuki graf, dan deskripsi yang dirasakan seperti aliran lokalizasi objek dan landmark wajah keluar dari graf. MediaPipe dirancang untuk praktisi pembelajaran mesin (*ML*) (*machine learning*), termasuk peneliti, mahasiswa, dan pengembang perangkat lunak, yang mengimplementasikan aplikasi ML yang siap produksi, menerbitkan kode yang menyertai karya penelitian, dan membangun prototipe teknologi. Kasus penggunaan utama untuk MediaPipe adalah prototipe cepat dari pipa persepsi dengan model inferensi dan komponen yang dapat digunakan kembali lainnya. MediaPipe juga memfasilitasi penyebaran teknologi persepsi ke dalam demo dan aplikasi di berbagai platform perangkat keras yang berbeda. MediaPipe memungkinkan peningkatan bertahap pada pipa persepsi melalui bahasa konfigurasi yang kaya dan alat evaluasi. [8]

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 2.2. 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.

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

$$mAP = \frac{1}{n} \sum_n^{i=1} AP_i \quad (4)$$

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 :

$$Intersection over Union (IoU) = \frac{|A \cap B|}{|A \cup B|}. \quad (5)$$

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 NCU 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 singage.

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 pengereaman pada motor

dengan menghubungkan kedua terminal motor sehingga motor dapat berhenti lebih cepat. fibriananalisis

K. Kursi Roda Elektrik KY-123

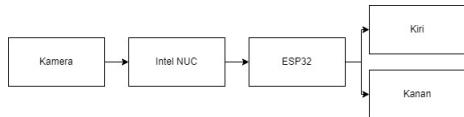
Kursi roda bertenaga listrik merupakan alat bantu mobilitas yang terdiri dari struktur dasar kursi roda, sistem pengendalian gerak, mesin elektrik, dan modul baterai. Keunggulan alat ini terletak pada kemampuannya untuk dikendalikan dengan mudah dan nyaman, meminimalkan usaha fisik yang diperlukan pengguna 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.

III. DESAIN DAN IMPLEMENTASI

Penelitian ini dilakukan sesuai dengan desain sistem berikut beserta implementasinya. Desain sistem adalah konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk alur yang harus dikerjakan

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 3.1 berikut dan dirincikan pada tiap subbab.



Gambar 3: Blok Diagram Hardware

B. Kamera

Pada tugas akhir ini pendekstian menggunakan data citra yang diolah untuk mendapatkan output berupa deteksi manusia yang akan dianalisa jaraknya. Kamera menjadi input utama untuk mendapatkan citra yang akan dipasangkan dalam kursi roda otomotif. Posisi kamera yang digunakan harus berada pada posisi yang dapat menangkap citra manusia dengan jelas. Adapun kamera yang digunakan adalah kamera Logitech C920.

C. Intel NUC

Intel NUC digunakan sebagai komputer pusat dalam kursi roda otomotif untuk memproses citra dari kamera dan menjalankan algoritma deteksi objek. Data yang dihasilkan dari deteksi digunakan untuk mengambil keputusan navigasi. Misalnya, menghindari hambatan dan menentukan arah gerak. Perintah tersebut kemudian dikirim ke ESP32 yang mengontrol mekanisme penggerak kursi roda.

```
pip install opencv-python  
pip install ultralytics  
pip install mediapipe
```

opencv-python digunakan untuk menangkap dan memproses citra dari kamera. Ultralytics digunakan untuk menjalankan deteksi objek menggunakan model YOLOv8. Mediapipe digunakan untuk menampilkan landmark pada manusia

D. ESP32

Untuk dapat menggerakkan kursi roda maka perlu mengirimkan perintah ke kontroler kursi roda. ESP32 menjadi akan menerima input perintah dasar untuk menggerakkan kursi roda, seperti maju, kanan, kiri. Perintah ini kemudian akan digabungkan dengan kecepatan maksimal menjadi satu command atau paket data seperti "Arah". Berikut merupakan tabel kode instruksi kursi roda berdasarkan hasil deteksi [10]

Hasil Klasifikasi	Kode Instruksi
Kiri	A
Maju	B
Stop	C
Mundur	D
Kanan	E

Setelah variabel tersebut dimasukkan maka akan dikirim secara nirkabel. Dalam tugas akhir ini menggunakan wifi dengan ssid Haris-Acess-Point setelah terkoneksi.

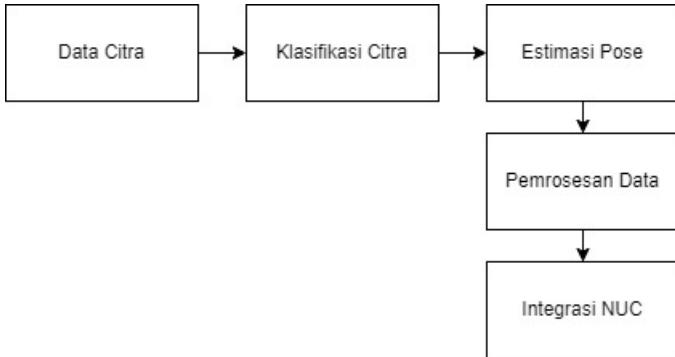
Gambar menampilkan skema detail dari perangkat yang dibahas. Sistem ini menggunakan kamera yang terhubung ke NUC, yang berfungsi sebagai perangkat utama untuk pengambilan gambar objek. Ketika kamera berhasil menangkap gambar objek, data citra tersebut kemudian diproses oleh Laptop atau Jetson Nano. Di dalam sistem, terdapat model klasifikasi yang telah diprogram untuk menganalisis data citra ini secara detail. Hasil analisis ini sangat penting, sebab menjadi dasar dalam pembuatan kode instruksi yang akan dieksekusi oleh sistem.

E. 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 blok diagram alur akan ditampilkan sebagai berikut :

F. Pengumpulan Dataset citra

Dalam pengembangan Tugas Akhir ini akan digunakan dataset citra berupa gambar, Dimana objek yang akan dideteksi pada gambar tersebut ialah Manusia yang fungsinya dalam tugas ini sebagai obstacle yang akan dihindari. citra manusia tersebut diambil melalui setiap frame citra pada video yang didapatkan menggunakan kamera webcam yang terhubung dengan komputer. Kemudian setiap citra yang didapatkan nantinya akan diproses untuk menentukan apakah terdapat manusia atau tidak pada citra tersebut. Dalam pendekstian, nantinya proses ini terjadi secara real-time guna mendapatkan keseluruhan citra yang dibutuhkan untuk mengenali manusia atau tidak pada citra secara terus-menerus



Gambar 4: Diagram Blok Software



Gambar 5: Contoh Hasil data dari Citra.

G. Object Detection YoloV8

Deteksi Objek dilakukan dengan mendeteksi keberadaan Manusia pada citra yang didapatkan, kemudia keberadaan manusia yang terdeteksi pada citra, akan digambar bounding box pada area yang terdeteksi manusia. Dimana Bounding box yang terdeteksi tersebut akan memiliki nilai nilai x dan y dalam posisi pixel pada jendela kamera yang selanjutnya dapat diambil berupa tinggi dan lebar pada pixel yang terdeteksi pada jendela web kamera. Dimana nilai nilai tersebut akan menjadi acuan dalam menentukan jarak objek relatif terhadap kamera.



Gambar 6: Contoh Hasil Deteksi Menggunakan Model YoloV8 yang telah di train.

H. Estimasi Pose MediaPipe

Deteksi pose dilakukan dengan menggunakan Python dengan library OpenCV dan frame-work MediaPipe menggunakan fungsi pose detection saat objek manusia terdeteksi dalam frame citra . Framework MediaPipe digunakan untuk mendapatkan landmark pada tubuh peraga, lalu landmark yang

relevan akan digambarkan garis berbentuk kerangka yang sesuai dengan pose tubuh peraga. Dalam penelitian ini, landmark yang relevan yaitu keypoint siku atas hingga lengan bawah dan bahu kanan dan kiri. Titik keypoint yang digunakan pada estimasi pose dapat dilihat pada Tabel dibawah ini.

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

Dari titik landmark yang didapat akan diambil jarak terhadap pixelnya dimana nilai pixel yang didapatkan akan menjadi acuan dalam perhitungan yang akan digunakan untuk menghindari obstacle. berikut contoh landmark pada data citra.



Gambar 7: Contoh hasil deteksi pose.

I. Perhitungan Rumus

Untuk dapat menentukan jarak, pertama tama model Yolo akan digunakan untuk mendeteksi class Manusia yang akan menjadi obstacle dalam tugas akhir ini. Dari hasil deteksi yang didapat akan digambar Bounding Box penanda kelas telah terdeteksi. Dimana dalam bounding box tersebut juga akan digambar Estimasi Pose dari MediaPipe. Kedua hasil deteksi ini yaitu Bounding Box dan Pose yang dihasilkan akan digunakan untuk mendapat estimasi jarak berdasarkan rumus yang akan dijabarkan, dimana hasil dari jarak yang didapatkan akan dipetakan perpindahannya dalam grid. Dimana grid ini akan menjadi acuan dari keputusan belok yang akan diambil. berikut merupakan contoh gambar pemetaan hasil deteksi dan pengambilan keputusan belok.

Dalam konteks tugas akhir ini Fokus panjang dalam piksel digunakan untuk mengkonversi ukuran halangan dari unit piksel menjadi unit meter. Ini penting karena kursi roda otonom perlu memahami jarak nyata ke halangan untuk mengambil keputusan navigasi yang tepat. Dalam penggunaannya dapat dilihat dalam rumus berikut.

$$focal_length_pixel = \left(\frac{jarak \times tinggi_bounding_box}{tinggi_objek_nyata} \right) \quad (6)$$

Saat menghitung jarak ke objek dengan menggunakan tinggi dari bounding box yang terdeteksi oleh YOLO, dikombinasikan dengan tinggi nyata objek yang diketahui dan fokus panjang dalam piksel. penerapannya dapat dilihat sebagai berikut :

$$jarak = \left(\frac{focal_length_pixel \times tinggi_objek_nyata}{tinggi_bounding_box} \right) \quad (7)$$

2. Perhitungan Jarak Obstacle berdasarkan Pose

Salah satu pendekatan yang baik dalam menentukan jarak dengan pose adalah menggunakan 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.

$$jarak_euclidean = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times scale_factor \quad (8)$$

Dari perhitungan diatas masih didapatkan jarak yang bernilai pixel. Dalam konteks perhitungan jarak perlu menggunakan nilai standar dalam Meter agar perhitungan tersebut sesuai dengan kaidah yang berlaku pada umumnya. Agar nilai pixel tersebut dapat berubah menjadi meter maka perlu dilakukan kalibrasi menggunakan nilai K. nilai K merupakan nilai kalibrasi berdasarkan pengukuran eksperimental. rumusnya dapat dilihat sebagai berikut.

$$jarak_meter = \left(\frac{k}{jarak_pixel} \right) \quad (9)$$

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 bisa berdampak pada keputusan navigasi kursi roda otonom. berikut rumus untuk pengkalibrasian nilai K:

$$k = Jarak_nyata_objek \times Ukuran_objek_dalam_piksel \quad (10)$$

J. Perhitungan Lebar Obstacle Berdasarkan Pose

Dalam perhitungan lebar juga menggunakan perhitungan jarak euclidean namun yang membedakan adalah konteks spesifikasi aplikasi. Dimana kedua pendekatan ini berbeda dalam aplikasi dan konteksnya.

$$width_pixel = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times scale_factor \quad (11)$$

Dimana sesuai dengan dijelaskan sebelumnya nilai output yang didapatkan adalah pixel dari lebar manusia. untuk memetakan lebarnya pixel ke meter maka perlulah membuat sebuah rumus konversi yang menentukan ukuran dalam pixel (yang merupakan ukuran digital dan relatif dikonversikan ke ukuran dunia nyata (meter)). sehingga rumusnya menjadi :

$$width_meters = width_pixel \times scale_factor \quad (12)$$

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:

$$scale_factor = \frac{dimensi_nyata_rata - rata}{ukuran_dalam_piksel_rata - rata} \quad (13)$$

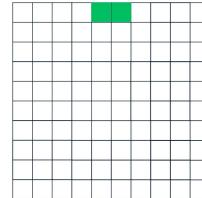
K. Visualisasi hasil deteksi dalam Grid

Setelah perhitungan rumus diatas diimplementasikan dalam kode. Maka akan didapatkan beberapa variabel penting yang akan digunakan dalam memetakan posisi maupun ukuran obstacle dalam grid. sebelum itu berikut tampilan grid yang digunakan.



Gambar 8: Parameter untuk setiap kotak grid.

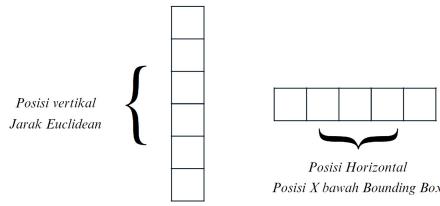
Agar dapat memetakan objek dengan baik dalam grid maka diperlukannya index untuk mengetahui posisi relatif kiri dan kanan objek terhadap kamera. Dimana untuk 10 kotak horizontal akan dibagi dimana Index 1 sampai 5 akan dikategorikan sebagai index kiri, dan index 6 hingga 10 akan dikategorikan dalam index kanan. pengambilan keputusan ini terkait dengan posisi kursi roda statis pada grid yang akan digambarkan sebagai berikut



Gambar 9: Posisi Kursi Pada grid.

Secara horizontal posisi objek hasil deteksi akan diambil berdasarkan Nilai perpindahan titik X bounding box bawah relatif terhadap pixel. Yang mana nilainya akan disesuaikan dengan Resolusi frame yang didapatkan oleh kamera. Dimana akan dibuat patokan untuk posisi tertentu yang melambangkan perpindahan sebesar 0.2 meter secara horizontal. Dengan asumsi bahwa kursi roda bergerak dan objek diam. maka hanya posisi objek yang terdeteksi didepanlah yang akan menjadi patokan penghindaran. Apabila objek terdeteksi namun saat berjalan maju objek menghilang maka kursi roda tidak perlu melakukan penghindaran. Oleh karena itu ditetapkan parameter jarak pada objek yang terdeteksi berdasarkan vertikal.

Namun sebelum membahas lebih lanjut, akan dijelaskan secara rinci mengenai pemetaan hasil deteksi melalui hasil variabel yang didapat.



Gambar 10: Variabel Dalam perpindahan Posisi terhadap Grid.

L. Near Object Condition

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.



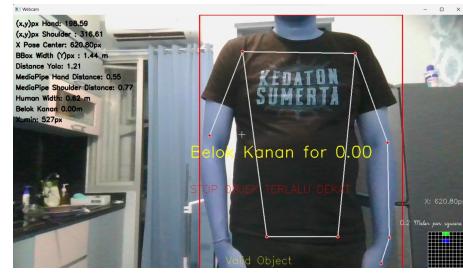
Gambar 11: Contoh kondisi Near Object Index Kiri >Kanan .

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 dibawah



Gambar 12: Contoh kondisi Near Object Index Kanan >kiri

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 13: Contoh kondisi Near Object Linear.

No	FPS
1	13.23
2	13.23
3	13.23
4	13.23
5	13.23

IV. HASIL DAN PEMBAHASAN

A. Pengujian Berdasarkan FPS

Dapat dilihat pada tabel dibawah, didapatkan pada pengujian ini nilai rata - rata FPS sebesar 13.140. Dimana nilai FPS tertinggi adalah sebesar 13.23 dan FPS paling rendah ialah 13.05. Kestabilan nilai pada Pengujian ini menandakan bahwa performa sistem yang dijalankan pada laptop berjalan dengan sangat baik dan efisien.

B. Pengujian pada NUC

TABEL I
TABEL FPS PADA NUC

No	FPS
1	6.37
2	6.43
3	6.43
4	5.57
5	5.54

Dapat dilihat pada tabel dibawah, didapatkan pada pengujian ini nilai rata - rata FPS sebesar 6.111. Dimana nilai FPS tertinggi yang didapatkan ialah 6.47 dan FPS paling rendah ialah 5.54. Performa yang dihasilkan pada NUC tergolong stabil namun hasilnya kurang mendekati performa laptop.

C. Pengujian Berdasarkan Hasil Response Time

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 0.248 dari data yang sudah didapatkan dari pengujian NUC, adapun hasil nya dapat dilihat pada tabel dibawah. Adapun nilai inference rata-rata yang didapatkan adalah

TABEL II
HASIL PENGUJIAN RESPONSE TIME

Inference	Sent	Receive	Motor	Delay	Arah
125.0	19:53:02.763	19:53:02.800	19:53:03.178	0.378	Stop
131.5	19:53:06.543	19:53:06.572	19:53:06.760	0.188	Maju
139.5	19:53:08.800	19:53:08.826	19:53:09.014	0.188	Stop
154.1	19:53:14.058	19:53:14.118	19:53:14.450	0.262	Kanan
131.6	19:53:17.932	19:53:17.974	19:53:18.162	0.188	Stop

D. Performa Keberhasilan

Berdasarkan Hasil yang didapatkan Penghindaran berhasil sebanyak 19 kali dan gagal 1 kali pada percobaan ke 11. Adapun presentasi yang didapatkan dari hasil uji ini sebesar 98%

Percobaan Ke	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	Objek tidak terdeteksi dan Gagal 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

V. KESIMPULAN

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

nolistsep

- Model dengan Metrics tertinggi yang telah di-training dengan berbagai konfigurasi dan arsitektur adalah model dengan skor mAP di IoU 0.5 tertinggi sebesar 94.10% yang menggunakan arsitektur SSD-Mobilenet-V2 FPN-Lite 320, skor tersebut lebih tinggi dibandingkan penelitian sebelumnya yang digunakan sebagai referensi
- Performa NUC dalam pengujian FPS menghasilkan Nilai yang lebih rendah ketimbang Laptop pribadi Penulis dengan selisih 7.029
- Rata - rata delay yang didapatkan pada pengujian adalah sekitar 0.248 dan rata- rata nilai inference yang didapatkan
- Hasil Performa Deteksi menunjukkan hasil yang memuaskan dalam 10 sampel pengujian. Dengan presentasi keberhasilan sebesar 100%

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

nolistsep

- Variasi dataset yang lebih ditingkat untuk meningkatkan performa deteksi
- Menggunakan SBC yang memiliki performa yang lebih baik untuk fps yang lebih tinggi
- Meningkatkan performa grid dengan mengubah ukuran 10x10 menjadi 100x100 atau lebih untuk hasil lebih detail

PUSTAKA

- [1] K. Daring, “Kbbi vi daring,” 2016. [Online]. Available: <https://kbki.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. Uboweja, 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] I. Ekatama, “Perancangan sistem kontrol motor kursi roda secara nirkabel berbasis esp32,” Ph.D. dissertation, Institut Teknologi Sepuluh Nopember, 2024.