

Sistem Pengikut Manusia pada Robot Servis Menggunakan Model YOLO dan Kamera Stereo

Muhammad Ilham Ramadhan, Djoko Purwanto, Hendra Kusuma
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: djoko@ee.its.ac.id

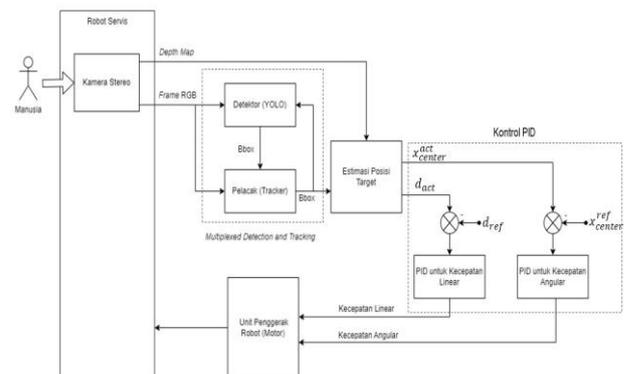
Abstrak—Kemampuan mengikuti seseorang merupakan fitur penting bagi robot servis yang bekerja berdampingan dengan manusia. Untuk merancang sistem pengikut manusia pada robot servis, diperlukan akurasi yang tinggi tapi juga tanpa mengorbankan kecepatan komputasi agar sistem berjalan secara *real-time*. Penelitian ini bertujuan untuk merancang sebuah sistem pengikut manusia untuk robot servis dengan memanfaatkan model pendeteksi objek *You Only Look Once* (YOLO) dan kamera stereo. Sistem ini dirancang agar robot dapat menjaga jarak yang tetap dari target yang diikuti dan menjaga orientasinya sehingga target tetap berada di tengah pandangan robot. Perancangan sistem ini juga memanfaatkan algoritma pelacak dari OpenCV yang dikoreksi dengan model YOLOv7 setiap 20 *frame* untuk menghasilkan proses yang lebih cepat. Pengontrol PID digunakan untuk menghasilkan kecepatan linear dan angular robot berdasarkan jarak relatif orang yang dijadikan target dari robot dan posisinya pada *frame*. *Robot Operating System* (ROS) digunakan untuk mempublish kecepatan pada *node* yang sesuai. Berdasarkan hasil pengujian algoritma pelacak, pelacak *Boosting* memiliki hasil terbaik untuk digunakan. Selanjutnya, sistem ini diuji untuk mengontrol robot servis di dalam ruangan dengan berbagai variasi kondisi. Dari pengujian-pengujian tersebut, robot berhasil untuk mengikuti seseorang dengan error RMS sebesar 41,88 mm dan standar deviasi sebesar 35,59 mm saat robot berhenti di jarak 1 m dari target. Nilai error terbesar yang didapat bernilai 320,369 mm yang terjadi ketika sistem dijalankan pada ruangan gelap. Sistem ini berjalan dengan *frame rate* rata-rata sebesar 17,18 FPS.

Kata Kunci—Kamera Stereo, Pelacakan, Pengikut Manusia, ROS, YOLO.

I. PENDAHULUAN

KEMAMPUAN sebuah robot untuk mengikuti manusia merupakan salah satu aspek penting yang diperlukan untuk robot yang bekerja di sekitar manusia seperti robot servis. Dengan kemampuan mengikuti manusia, robot servis akan dapat memberikan banyak bantuan dan layanan [1]. Beberapa contoh robot pengikut manusia adalah robot perawatan lansia asisten belanja pengikut bidan dan lain sebagainya [2]–[5]. Beberapa manfaat dari kemampuan mengikuti manusia antara lain untuk membantu seseorang, memberikan kemudahan dalam pekerjaan, dan dapat juga dimanfaatkan untuk alasan keamanan [6].

Salah satu permasalahan terkait robot pengikut manusia adalah cara mendeteksi dan mendapatkan jarak dari seseorang. Penglihatan stereo merupakan salah satu metode yang digunakan oleh beberapa peneliti untuk mencapai hal tersebut. Dalam pengembangan robot pengikut seseorang, kamera stereo digunakan untuk mengontrol robot secara nyata dengan memanfaatkan *depth template* dari bentuk manusia dalam pendeteksiannya [7], [8]. Mereka melakukan pengembangan sistem *robotic follower* dengan cara menggabungkan pendeteksian manusia berbasis stereo vision



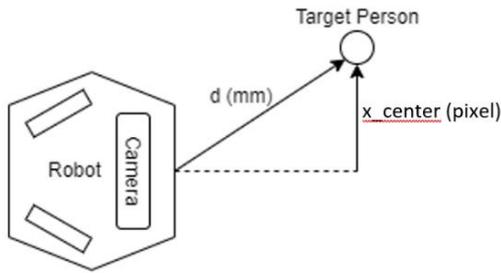
Gambar 1. Diagram Blok Sistem.

dengan pelacakan manusia menggunakan *modified Kalman filter*.

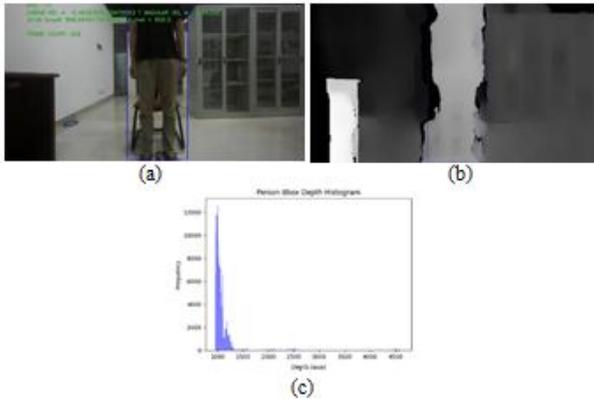
Penelitian terdahulu umumnya menggunakan pendekatan konvensional untuk mengolah citra visual atau mendapatkan posisi objek pada frame gambar. Saat ini, metode pengolahan citra visual sudah mengalami banyak perkembangan, terlebih lagi dengan meningkatnya jumlah penelitian dalam model berbasis jaringan saraf tiruan. Salah satu contoh penggunaan algoritma berbasis jaringan saraf tiruan adalah penelitian yang memanfaatkan pelacak *Convolutional Neural Network* (CNN) berbasis kamera stereo untuk mengikuti target pada suasana yang kompleks seperti adanya halangan, perubahan penampilan, perubahan pose, perbedaan pencahayaan, dan sebagainya [9]. Penelitian yang memanfaatkan *Deep Convolutional Neural Network* sebagai detektor untuk menghasilkan sistem robot *omnidirectional* yang dapat mengikuti target objek dinamis [10].

Salah satu model berbasis jaringan saraf tiruan yang menjadi topik menarik belakangan ini adalah detektor objek *You Only Look Once* (YOLO) [11]. YOLO menganggap masalah pendeteksian suatu objek sebagai permasalahan terkait regresi, dimana *bounding box* dan probabilitas kelas suatu objek didapatkan langsung dari pixel-pixel suatu gambar, sehingga menghasilkan model deteksi dengan kecepatan yang tinggi. YOLO terus dikembangkan oleh berbagai kalangan dan salah satu versi terbarunya adalah YOLOv7 [12]. YOLOv7 memperkenalkan strategi baru untuk menambah akurasi model tanpa kehilangan kecepatannya. YOLOv7 berhasil melampaui performa model pendeteksi objek lainnya baik dari segi akurasi maupun kecepatan [12].

Penggunaan model detektor berbasis *neural network* menghasilkan kecepatan yang rendah pada sistem pengikut manusia karena beban komputasinya yang berat [13]. Mereka mengusulkan penggunaan *multiplexed detection and tracking*, yaitu metode dimana detektor berbasis *neural network* digunakan secara bergantian dengan algoritma



Gambar 2. Sistem Koordinat Robot.



Gambar 3. Pengambilan Data *Depth*/Jarak Seseorang: (a) Gambar RGB, (b) *Depth Image*, (c) Histogram Jarak yang Didapat dari *Bounding Box* Seseorang.

pelacak. Karena beban komputasi pelacak lebih ringan dibandingkan dengan detektor berbasis *neural network*, penelitian ini menghasilkan hasil yang seimbang antara kecepatan yang ada pada pelacak dan akurasi yang ada pada detektor. Mereka menggunakan model You Only Look Once (YOLO) YOLOv3 dan kamera RealSense.

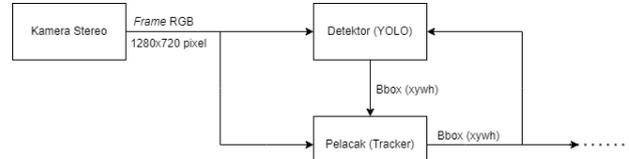
Berdasarkan alasan-alasan di atas, penulis mengusulkan Sistem Pengikut Manusia pada Robot Servis Menggunakan Kamera Stereo. Kamera yang akan digunakan adalah ZED2i Stereo Camera yang dibuat oleh Stereolabs. Kamera tersebut berhasil melampaui performa kamera dengan *depth sensing* lainnya seperti kamera seri D400 dari Intel RealSense dalam berbagai kondisi [14]. Model YOLOv7 akan digunakan untuk memproses gambar RGB dan menghasilkan *bounding box* pada tiap orang yang dideteksi. Pendekatan *multiplexed detection and tracking* akan digunakan untuk menghasilkan sistem dengan *frame rate* yang tinggi dan akurat. Robot Operating System (ROS) akan digunakan untuk mengatur komunikasi dari node yang diperlukan.

II. PENELITIAN TERDAHULU

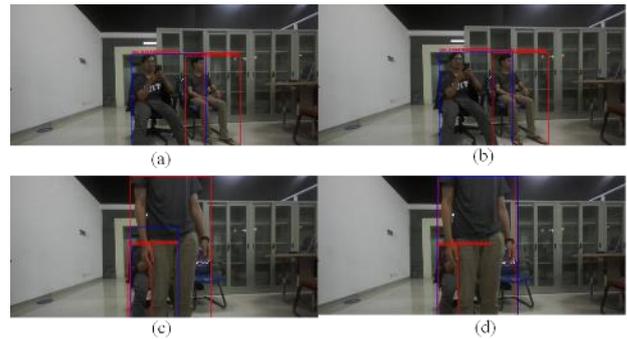
Beberapa penelitian telah dilakukan untuk membuat sistem pengikut manusia. Penelitian yang membuat metode pelacakan manusia berbasis penglihatan stereo yang digunakan untuk mengontrol robot untuk mengikuti seseorang [8]. Pada penelitiannya, deteksi berbasis penglihatan stereo digabungkan dengan pelacak berupa *modified Kalman filter*. Penelitian ini menggunakan *Backpropagation Neural Network* yang dilatih menggunakan *feature vectors* dari wilayah pada gambar yang tersegmentasi sebagai manusia. Penelitian ini berhasil untuk menghasilkan sistem pengikut manusia. Namun, nilai *frame rate* yang dihasilkan cukup rendah, yaitu sebesar 4Hz atau 4 FPS (*frame per second*). Penelitian yang dilakukan distribusi komputasi



Gambar 4. Proses Penentuan Target Di Awal Sistem Dijalankan: (a) *Frame Pertama*, (b) *Frame Selanjutnya*.



Gambar 5. Diagram Blok Penggabungan Deteksi dan Pelacakan.



Gambar 6. Koreksi Pelacak Menggunakan Detektor: Gambar (a) dan (b) Kondisi Normal. Gambar (c) dan (d) Terdapat Orang Lain yang Menghalangi Target.

menggunakan dua buah PC untuk mengatasi lambatnya *frame rate* tersebut sehingga sistem yang mereka buat mencapai *rate* 12Hz atau 12 FPS [8].

Penelitian menggunakan algoritma berbasis *deep learning* dan *color feature* untuk mengikuti manusia di dalam ruangan [15]. Algoritma *deep learning* digunakan untuk mendeteksi manusia sedangkan *color feature* (menggunakan histogram *hue-saturation value*) digunakan untuk mengidentifikasi orang yang dijadikan target. Sistem yang mereka buat sudah diintegrasikan dengan *obstacle avoidance* dan juga lokalisasi. Selain itu, pada penelitian ini juga dibuat *state-machine* dimana robot memiliki *state* yang berbeda-beda tergantung pada kondisi ketika sistem dijalankan. Sama seperti penelitian sebelumnya, pada penelitian ini pengujian dilakukan dengan membagi beban komputasi ke dua komputer, dimana satu komputer digunakan untuk menjalankan modul *deep learning* dan satu lagi adalah komputer pada robot yang digunakan untuk kontrol, *path planning*, dan menjalankan algoritma *obstacle avoidance*. Hasil pengujian dari penelitian ini memberikan nilai *frame rate* rata-rata sebesar 23.34 FPS. Beberapa kekurangan dari penelitian ini meliputi lemahnya efektivitas sistem pada perubahan pencahayaan yang jauh dan juga kegagalan pelacakan jika terdapat orang lain yang menggunakan baju yang sama dengan orang yang dijadikan target.

Dalam mengembangkan robot pengikut seseorang menggunakan metode *multiplexed detection and tracking* [13]. Metode ini dikembangkan untuk mengatasi beratnya beban komputasi yang dihasilkan oleh detektor berbasis jaringan saraf tiruan seperti YOLO. Mereka menggunakan model YOLOv3 sebagai detektor dan pelacak *Kernelized Correlation Filter* (KCF) yang disediakan oleh OpenCV.

Tabel 1.
Hasil Pengujian Pelacak dan Detektor.

| Pelacak | Frame Rate (fps) | Rata-rata IoU | Success Rate | Kelebihan | Kekurangan |
|--|------------------|---------------|--------------|--|--|
| Tanpa Pelacak (Detektor YOLOv7 digunakan setiap frame) | 17 | - | - | <i>Bounding box</i> yang dihasilkan memiliki akurasi yang tinggi | <i>Frame rate</i> tidak begitu tinggi |
| BOOSTING | 23 | 0.3722 | 94.79% | <i>Frame rate</i> stabil, dapat melacak orang yang sedang berjalan | Tidak dapat menangani perubahan ukuran pada target |
| MIL | 16 | 0.3463 | 91.07% | <i>Frame rate</i> stabil | Terkadang <i>bounding box</i> yang dihasilkan tertinggal dari target |
| KCF | 24 | 0.2681 | 40.05% | <i>Frame rate</i> stabil | Gagal ketika target terhalang |
| TLD | 10 | 0.0627 | 87.26% | Tidak ada | <i>Frame rate</i> terlalu rendah |
| MEDIANFLOW | 41 | 0.0999 | 35.06% | <i>Frame rate</i> tinggi | Tidak dapat melacak target yang bergerak dengan cepat |
| MOSSE | 36 | 0 | 0 | <i>Frame rate</i> tinggi | Sering terjadi kegagalan secara acak |
| CSRT | 9 | 0.4222 | 89.94% | Dapat melacak target yang bergerak dengan cepat | <i>Frame rate</i> tidak stabil dan terlalu rendah |

Pada metodenya, detektor akan berjalan hanya setiap beberapa frame saja. Sebagian besar proses akan dilakukan oleh algoritma pelacak KCF. Pelacak tersebut akan dikoreksi oleh detektor YOLOv3 setiap 100 frame, atau ketika pelacak mengalami kegagalan, untuk mengatasi error yang terjadi. Proses yang memanfaatkan pelacak dan detektor secara bergantian tersebut menghasilkan sistem pengikut seseorang yang cepat dan juga akurat. Dari penelitian ini, mereka melakukan pengujian robot untuk mengikuti seseorang dengan kondisi yang bervariasi dan mengambil kesimpulan bahwa robot bekerja dengan baik dan siap digunakan dalam kegiatan sehari-hari [13]. Keberhasilan metode tersebut dan kemudahan dalam pengaplikasiannya dapat dijadikan acuan untuk pengembangan sistem robot pengikut manusia

III. METODOLOGI

A. Metode yang Digunakan

Metode ini didasari oleh pendekatan *multiplexed tracking and detection* dengan memanfaatkan algoritma pelacak yang proses komputasinya lebih ringan dibandingkan dengan algoritma detektor berbasis *neural network* seperti YOLO [13]. Pelacak akan berjalan selama jumlah frame yang ditentukan. Setelah mencapai jumlah tersebut, atau jika pelacak mengalami kegagalan, detektor akan dijalankan untuk melakukan koreksi pelacak. Proses tersebut akan berjalan terus menerus sehingga menghasilkan sistem di mana pelacak dan detektor yang berjalan secara bergantian. Algoritma pelacak dengan beban komputasi yang lebih ringan namun rentan dengan eror akan dikoreksi dengan detektor berbasis *neural network* dengan akurasi yang tinggi namun juga beban komputasi yang berat, sehingga akan memberikan hasil yang lebih seimbang antara keduanya.

Sistem ini memanfaatkan kamera stereo ZED2i Stereo Camera untuk mendapatkan *frame* RGB yang akan diproses oleh detektor dan *depth map* yang akan digunakan untuk estimasi jarak seseorang. Detektor akan dibangun menggunakan model YOLOv7 yang telah dilatih menggunakan dataset MS COCO. Algoritma pelacak yang akan digunakan adalah algoritma yang disediakan oleh OpenCV. Pengontrol PID digunakan untuk menghasilkan kecepatan linear dan angular robot dengan melihat eror antara

posisi target yang diamati dengan posisi seharusnya. Diagram blok sistem dapat dilihat pada Gambar 1. Penjelasan dari masing-masing bagian dan alur sistem akan dijelaskan lebih lanjut pada bagian selanjutnya.

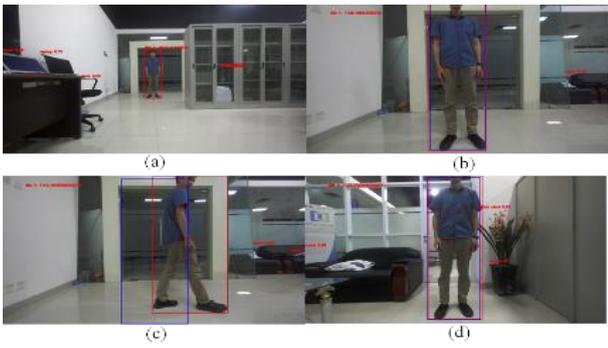
Terdapat beberapa jenis pelacak yang disediakan oleh OpenCV. Untuk memilih jenis pelacak yang akan digunakan, diperlukan percobaan pada robot untuk melihat kelebihan dan kekurangan dari masing-masing pelacak. Metode pengujian akan dijelaskan lebih lanjut pada Sub-bab 3.3.

Terdapat beberapa jenis model YOLOv7 yang telah dilatih menggunakan dataset MS COCO yang dapat digunakan, mulai dari YOLOv7-tiny, YOLOv7, YOLOv7-X, hingga YOLOv7-E6E. Semakin besar akurasi dari suatu model umumnya berbanding terbalik pada waktu yang dibutuhkan untuk memproses suatu gambar. Pada sistem yang diharapkan untuk bekerja secara *real-time*, model yang cepat seperti YOLOv7-tiny tentu menjadi bahan pertimbangan untuk digunakan. Sayangnya, nilai *average precision* dari model YOLOv7-tiny jatuh terlalu jauh dengan nilai 38.7 jika dibandingkan dengan model di atasnya yaitu YOLOv7 dengan nilai 51.4 [12]. Dengan *frame rate* yang sudah cukup tinggi, model YOLOv7 dijadikan model yang digunakan sebagai detektor untuk pengembangan sistem ini.

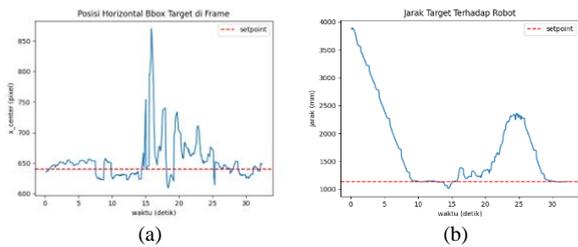
1) Estimasi Posisi Target

Pada saat sistem ini dijalankan, robot akan menjaga jarak (d) sejauh 1-meter (1000 mm) dari orang yang dijadikan target. Di saat yang bersamaan, robot juga akan mengubah orientasinya sehingga posisi seseorang yang dijadikan target selalu berada di tengah frame gambar. Hal tersebut dilakukan karena robot hanya bisa melihat ke depan, sesuai dengan arah lensa kamera. Dengan menjaga target tetap berada di tengah frame, robot diharapkan untuk dapat terus melacak target dan target tidak keluar frame.

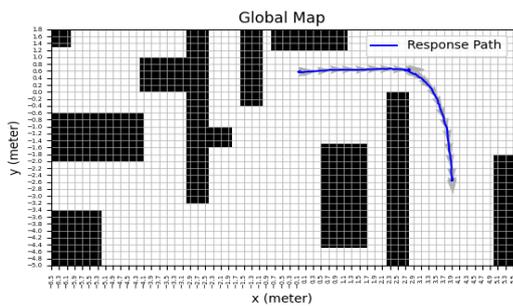
Menjaga posisi seorang target untuk tetap berada di dalam frame dapat dilakukan dengan mengambil jarak horizontal titik tengah *bounding box* nilai x pada Gambar 2. Orang tersebut dari titik tengah frame gambar. Posisi vertical *bounding box* target dalam frame gambar dapat diabaikan karena robot hanya bisa melakukan putaran terhadap sumbu- y (putaran *yaw*). Resolusi gambar yang digunakan adalah HD720 atau 1080×720 pixel, sehingga titik tengah horizontal *frame* berada di nilai 640 pixel.



Gambar 7. Posisi Target Selama Pengujian Pertama: (a) Posisi Awal, (b) Robot Mencapai Jarak 1 Meter dari Target, (c) Target Bergerak ke Arah Kanan, (d) Posisi Akhir.



Gambar 8. (a) Posisi Titik Tengah *Bounding box* Target pada *Frame* Selama Pengujian Pertama, (b) Jarak Target Terhadap Robot Selama Pengujian Pertama.



Gambar 9. Jalur Robot pada Pengujian Pertama.

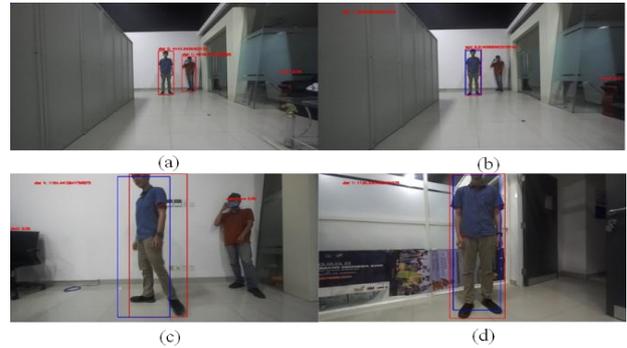
Estimasi jarak target nilai d pada Gambar 2 dapat dilakukan dengan mengambil data jarak dari seluruh titik yang ada di *bounding box* seseorang. Selanjutnya, rata-rata dari 5% data jarak terkecil akan digunakan untuk merepresentasikan jarak orang tersebut. Penggunaan rata-rata dilakukan untuk mengantisipasi adanya data eror agar robot tidak keluar kendali.

Contoh estimasi posisi seseorang dapat dilihat pada Gambar 3. Pada Gambar 3, terlihat bahwa ada seseorang yang terdeteksi oleh robot relatif di tengah *frame* gambar. Posisi horizontal titik tengah *bounding box* orang tersebut diperkirakan bernilai 650.5 pixel yang tidak begitu jauh dari posisi horizontal titik tengah *frame* yang berada pada nilai 640 pixel.

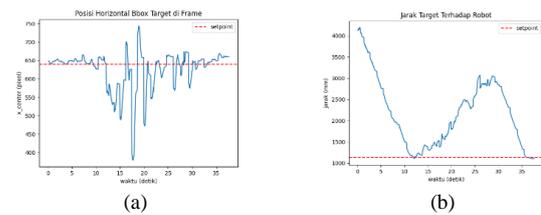
Pada Gambar 3, dapat dilihat visualisasi *depth map* dan juga data jarak yang didapat pada seluruh titik di *bounding box* orang yang terdeteksi. Jarak orang tersebut diperkirakan sejauh 999.66 mm dari kamera. Perkiraan tersebut tidak begitu jauh dari nilai jarak orang sebenarnya yang berada pada jarak 1 meter di depan kamera.

2) Deteksi Awal Menggunakan YOLOv7

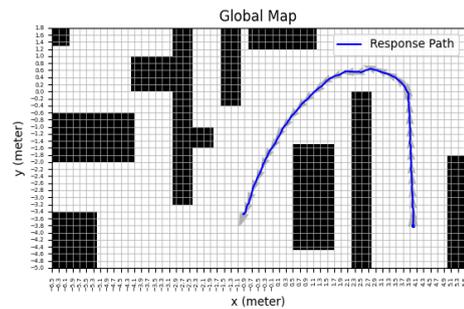
Pada saat pertama kali sistem dijalankan, kamera stereo akan mengambil *frame* RGB pertama dan *depth map*-nya. *Frame* RGB akan diproses oleh detektor YOLOv7 untuk mendeteksi manusia yang ada pada gambar. Dari seluruh



Gambar 10. Tampilan *Frame* Selama Pengujian Ketujuh: (a) Pada *Frame* Pertama Terdeteksi Dua Orang, (b) Orang Terdekat Dijadikan Target, (c) Robot Berhasil Mendekati Orang yang Sama, (d) Posisi Akhir Orang yang Diikuti.



Gambar 11. (a) Posisi Titik Tengah *Bounding box* Target pada *Frame* Selama Pengujian Ketujuh, (b) Jarak Target Terhadap Robot Selama Pengujian Ketujuh.



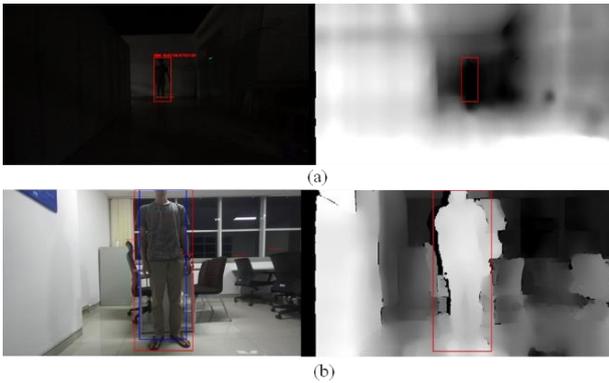
Gambar 12. Jalur Robot pada Pengujian Ketujuh.

manusia yang terdeteksi, orang yang terdekat dengan robot akan dijadikan target pelacakan. *Bounding box* orang tersebut akan digunakan untuk inisialisasi pelacak. Selanjutnya, sistem akan masuk ke *main loop* dimana proses pelacakan akan dilakukan. Perlu diketahui juga bahwa sebelum diproses menggunakan model YOLOv7, gambar yang didapat dari kamera stereo akan diubah ukurannya menjadi 640x640 pixel tanpa mengubah *aspect ratio*-nya. Jika gambar yang didapat tidak memiliki rasio dimensi 1:1, maka pada gambar yang dihasilkan akan ditambahkan wilayah berwarna hitam untuk mengisi wilayah yang kosong.

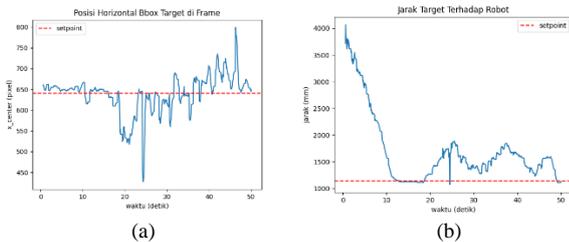
Contoh proses penentuan target di awal sistem dijalankan dapat dilihat pada Gambar 4. Terlihat bahwa pada saat sistem pertama kali dijalankan terdapat dua orang yang terdeteksi oleh model YOLOv7 (ditandai oleh *bounding box* berwarna merah). Jarak orang sebelah kiri dan sebelah kanan secara berurutan bernilai sekitar 4147 mm dan 4919 mm dari kamera. Karena orang sebelah kiri memiliki jarak yang lebih kecil, *bounding box* orang tersebut digunakan untuk inisiasi pelacak dan dapat dilihat pada *frame* selanjutnya (ditandai oleh *bounding box* berwarna merah).

3) Menggabungkan Proses Deteksi dan Pelacakan

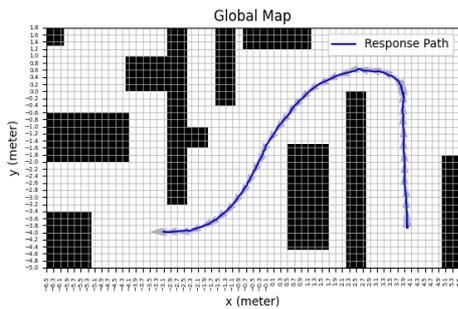
Di dalam *main loop*, pelacak OpenCV akan digunakan untuk melacak target dalam *frame* gambar sebanyak jumlah *frame* yang sudah ditentukan yang ditunjukkan pada Gambar



Gambar 13. Tampilan *Frame* Selama Pengujian Ketujuh: (a) Pada *Frame* Pertama Terdeteksi Dua Orang, (b) Orang Terdekat Dijadikan Target, (c) Robot Berhasil Mendekati Orang yang Sama, (d) Posisi Akhir Orang yang Diikuti.



Gambar 14. (a) Posisi Titik Tengah *Bounding box* Target pada *Frame* Selama Pengujian Ketujuh, (b) Jarak Target Terhadap Robot Selama Pengujian Ketujuh.



Gambar 15. Jalur Robot pada Pengujian Ketujuh.

5. Proses pelacakan bekerja dengan cara meng-*update* pelacak (yang telah diinisiasi menggunakan *bounding box* seseorang pada *frame* pertama) dengan *frame* selanjutnya untuk mendapatkan estimasi posisi orang tersebut dalam bentuk *bounding box* dengan format $(x_{min}, y_{min}, width, height)$. Proses pelacakan tersebut akan berjalan selama 20 *frame*. Setelah jumlah tersebut tercapai, atau ketika pelacak mengalami kegagalan, detektor akan digunakan untuk melakukan perbaikan. Detektor akan mendeteksi seluruh manusia yang ada pada *frame* gambar dan menghasilkan *bounding box* dengan format $(x_{min}, y_{min}, width, height)$. Nilai *intersection over union* (IoU) akan digunakan untuk mencari *bounding box* mana yang dihasilkan oleh detektor yang merupakan target pelacak sebelumnya. *Bounding box* dari detektor yang memiliki nilai IoU terbesar dengan *bounding box* dari pelacak akan digunakan untuk inisiasi ulang pelacak. Pada penelitian ini, ukuran *frame* yang diproses oleh pelacak adalah 1280×720 pixel.

Gambar 6 adalah contoh dari koreksi yang dilakukan oleh detektor pada pelacak. Pada Gambar 6, sebelah kiri adalah *frame* tepat sebelum koreksi pelacak dan kanan adalah *frame* tepat setelah koreksi. Terlihat bahwa penggunaan IoU untuk

menentukan orang yang dilacak dapat berfungsi dengan baik selama orang tersebut tidak terhalang dengan objek atau orang lain. Pada Gambar 6 terlihat bahwa ketika ada orang yang menghalangi orang yang dilacak sebelumnya oleh pelacak, proses koreksi akan menyebabkan orang yang menghalangi tersebut untuk dilacak. Hal tersebut disebabkan oleh besarnya kemungkinan bahwa nilai IoU dari *bounding box* orang yang menghalangi dengan *bounding box* pelacak lebih besar daripada IoU yang dimiliki oleh *bounding box* target yang terhalang.

4) Kontrol Gerakan Robot

Kontrol robot dilakukan menggunakan dua persamaan kontrol PID, dimana persamaan pertama digunakan untuk menghasilkan kecepatan angular dan yang kedua digunakan untuk menghasilkan kecepatan linear. Kecepatan angular robot akan dihitung menggunakan persamaan PID berdasarkan posisi horizontal *bounding box* target pada persamaan 1.

$$v_{\theta}(n) = Kp_{\theta} \cdot e_{\theta}(n) + Ki_{\theta} \sum_{i=0}^n e_{\theta}(i) + Kd_{\theta} (e_{\theta}(n) - e_{\theta}(n-1)) \quad (1)$$

dimana v_{θ} adalah kecepatan translasi robot terhadap sumbu- θ robot, e_{θ} adalah eror antara posisi horizontal titik tengah *bounding box* target terhadap nilai referensi, n menandakan iterasi ke- n , dan Kp_{θ} , Ki_{θ} , dan Kd_{θ} adalah konstanta PID untuk persamaan kecepatan angular robot. Error pada kontrol kecepatan angular (e_{θ}) pada persamaa 2.

$$e_{\theta}(n) = x_{center}^{act}(n) - x_{center}^{ref} \quad (2)$$

dimana e_{θ} adalah error untuk perhitungan kecepatan angular, $x_{center}^{act}(n)$ adalah posisi horizontal sebenarnya dari titik tengah *bounding box* target pada *frame* pada iterasi ke- n , dan x_{center}^{ref} adalah nilai referensi dari posisi horizontal titik tengah *bounding box* target pada *frame*. Ukuran *frame* yang digunakan adalah HD720 atau sebesar 1280×720 pixel. Titik tengah pada *frame* tersebut berada di nilai 640×360 pixel. Sehingga, nilai x_{center}^{ref} adalah 640 pixel. Hasil kecepatan angular (v_{θ}) akan digunakan untuk menghasilkan gerakan *yaw* yang menyebabkan robot untuk mengubah orientasinya ke arah target. Pada pengujian, nilai Kp_{θ} , Ki_{θ} , dan Kd_{θ} yang digunakan secara berurutan adalah 0.09, 0.0001, 0.01.

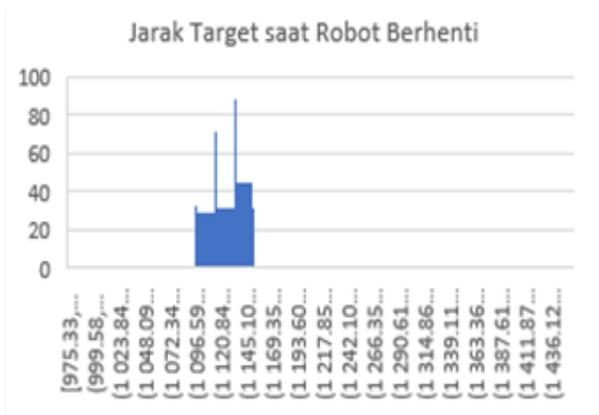
Kecepatan linear robot akan dihitung berdasarkan jarak target. Persamaan untuk mencari kecepatan linear robot pada persamaan 3.

$$v_l(n) = Kp_l \cdot e_l(n) + Ki_l \sum_{i=0}^n e_l(i) + Kd_l (e_l(n) - e_l(n-1)) \quad (3)$$

dimana v_l adalah kecepatan linear robot, e_l adalah eror antara posisi target dengan nilai referensi, n menandakan iterasi ke n , dan Kp_l , Ki_l , dan Kd_l adalah konstanta PID untuk persamaan kecepatan linear robot. Error e_l dapat dituliskan pada persamaan 4.

$$e_l(n) = d_{act}(n) - d_{ref} \quad (4)$$

dimana e_l adalah eror untuk perhitungan kecepatan linear robot, $d_{act}(n)$ adalah jarak seorang target yang didapat dari kamera stereo dalam milimeter pada iterasi ke- n , dan d_{ref} adalah nilai referensi dari jarak antara robot dengan target yang diinginkan. Perlu diketahui bahwa jarak robot dengan target yang dibahas dalam penelitian ini adalah jarak dari ujung depan *body* robot. Sehingga, jarak yang dihasilkan dari kamera stereo harus dijumlahkan dengan jarak antara kamera



Gambar 16. Data Jarak Target saat Robot Berhenti.



Gambar 17. Data Waktu Komputasi untuk Tiap Iterasi.

dengan ujung depan *body* robot. Jarak tersebut adalah 14 cm atau 140 mm. Sehingga, nilai d_{ref} adalah 1000 mm ditambah dengan 140 mm, yaitu 1140 mm. Kecepatan linear yang telah dihitung akan digunakan untuk menghasilkan gerakan maju atau mundur pada robot. Pada pengujian di penelitian ini, nilai Kp_l , Ki_l , dan Kd_l yang digunakan secara berurutan adalah 0.15, 0.00001, dan 0.1.

Kecepatan linear robot dibatasi pada nilai 40 cm/s atau 0.4 m/s untuk alasan keamanan. Batasan tersebut diambil berdasarkan kondisi ruangan tempat robot beraksi yang cukup sempit dan untuk mengantisipasi adanya eror yang tidak terduga. Setelah nilai kecepatan didapatkan, Robot Operating System (ROS) akan digunakan untuk mengirim (*publish*) kedua kecepatan tersebut dalam bentuk *array* untuk menghasilkan manuver robot.

5) Menangani Target yang Tidak Terlacak dengan Baik

Terdapat beberapa kondisi yang dapat menyebabkan pelacak untuk gagal, seperti ketika melacak target yang terlalu cepat, target yang terhalang objek lain, atau ketika target keluar dari *frame*. Kondisi tersebut dapat menyebabkan kegagalan pada program sehingga sistem langsung berhenti. Salah satu solusi yang dapat digunakan ketika hal tersebut terjadi adalah dengan melakukan deteksi kembali dengan asumsi bahwa orang yang terdekat dengan robot adalah orang yang sebelumnya diikuti. Lalu, jika orang yang diikuti keluar dari *frame* gambar, robot dapat berputar sesuai dengan posisi terakhir *bounding box* orang tersebut pada *frame* gambar. Dengan kata lain, jika orang tersebut terlihat terakhir di sebelah kiri *frame* maka robot akan berputar ke arah kiri dan sebaliknya.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Pelacak dan Detektor

Pada pengujian ini, proses kontrol seperti perhitungan kecepatan, pengiriman kecepatan melalui ROS, dan juga proses lainnya seperti penulisan info pada gambar tidak dilakukan. Pada Tabel 1, terlihat bahwa pengujian pertama memanfaatkan detektor saja tanpa pelacak untuk memproses setiap *frame* dan menghasilkan *frame rate* sebesar 17 FPS. Hasil tersebut tidak begitu buruk, namun belum bisa digunakan untuk melakukan pelacakan seseorang.

Pengujian untuk mencari *frame rate* pelacak OpenCV dilakukan dengan mengkoreksi pelacak setiap 20 *frame*, atau ketika pelacak mengalami kegagalan, menggunakan detektor dan tanpa melakukan kontrol gerakan robot. Sementara itu, pada pengujian untuk mencari rata-rata IoU dan *success rate*, pelacak tidak dikoreksi sehingga terlihat performa pelacak saja. *Intersection over Union* didapatkan dengan mencari area tumpang tindih dibagi dengan area gabungan antara *bounding box* pelacak dan *bounding box* sebenarnya. *Success rate* didapatkan dengan menghitung perbandingan antara jumlah iterasi dengan nilai IoU di atas nol dengan jumlah iterasi seluruhnya.

Terlihat pada Tabel 1, nilai rata-rata IoU terbesar dihasilkan oleh pelacak CSRT yang menandakan bahwa pelacak ini memiliki akurasi yang lebih tinggi dibandingkan dengan pelacak lainnya. Namun, *frame rate* yang didapat menggunakan pelacak ini berada pada nilai 9 FPS. Nilai tersebut terlalu rendah, bahkan lebih rendah dari saat penggunaan detektor saja. Sementara itu, beberapa pelacak lainnya mengalami kegagalan ketika melacak target yang berjalan seperti pelacak *MedianFlow* dan *MOSSE*. Kegagalan tersebut menyebabkan pelacak menjadi terlalu bergantung pada detektor sehingga *frame rate* yang dihasilkan dapat turun secara drastis. Beberapa pelacak lainnya seperti *KCF* dan *MIL* tidak mengalami kegagalan, namun *bounding box* yang dihasilkan sering kali tertinggal dari objek yang dilacak. Hal tersebut menyebabkan proses pelacakan menjadi tidak relevan terhadap orang yang menjadi target sebelumnya.

Selama pengujian ini, pelacak *Boosting* memberikan hasil yang paling stabil dengan nilai *frame rate* sekitar 23 FPS. Pelacak ini juga berhasil untuk melacak orang yang sedang berjalan dan jarang mengalami kegagalan, ditandai oleh tingginya nilai IoU dan *success rate* yang dihasilkan. Hasil tersebut menjadikan pelacak *Boosting* sebagai pelacak yang digunakan pada penelitian ini.

Pada sistem ini, pelacakan dan pendeteksian dilakukan secara bergantian. Pelacak akan dijalankan dalam jumlah *frame* yang lebih banyak dibandingkan dengan detektor untuk menghasilkan *frame rate* yang tinggi. Sebagai perbandingan, jika sistem ini melakukan pelacakan (dengan pelacak *Boosting*) dan deteksi pada setiap *frame*, nilai *frame rate* yang dihasilkan berada di sekitar nilai 3 FPS. Oleh karena itu, pada sistem ini pelacak akan dilakukan secara terus menerus dan hanya akan dikoreksi oleh detektor setiap 20 *frame* atau ketika pelacak mengalami kegagalan. Jumlah 20 *frame* tersebut digunakan untuk menghindari error yang terlalu besar dari pelacak jika dijalankan dalam waktu yang lebih lama lagi. Selain itu, proses koreksi ini juga menghasilkan solusi dari kekurangan yang ada pada pelacak *Boosting*

karena *bounding box* yang dihasilkan oleh YOLO dapat beradaptasi dengan perubahan ukuran dari orang yang dideteksi.

B. Hasil Pengujian Mengikuti Manusia

Pengujian dilakukan dalam berbagai kondisi mulai dari perubahan posisi awal dan akhir target di map, perubahan kondisi ada dan tidak adanya orang lain yang terdeteksi, hingga perubahan cahaya. Perlu diperhatikan bahwa robot menjaga jarak 1 meter dari target dihitung dari ujung depan body robot, bukan dari kamera. Jarak kamera dengan ujung depan body robot adalah 14 cm, sehingga untuk menjaga jarak 1 meter dari target, setpoint yang digunakan adalah 1140 mm atau 1.14 m. Titik tengah frame gambar berada di nilai 640 pixel.

Pengujian untuk mengikuti manusia dilakukan sebanyak sepuluh kali. Berikut ini adalah beberapa hasil dari pengujian tersebut:

1) Pengujian Kedua: Pengujian dengan Satu Orang di Ruang

Posisi target selama pengujian pertama dapat dilihat pada Gambar 7. *Bounding box* berwarna biru adalah *bounding box* yang dihasilkan oleh pelacak *Boosting Tracker*, sementara *bounding box* berwarna merah adalah milik detektor YOLOv7. Terlihat bahwa sistem berhasil untuk melacak target selama pengujian seperti data yang ditunjukkan pada Gambar 7, Gambar 8, dan Gambar 9.

Respon gerakan robot terhadap posisi target di *frame* dapat dilihat pada Gambar 8. Terlihat pada Gambar 8 bahwa robot selalu berusaha untuk menjaga posisi horizontal target untuk berada di tengah *frame* (di posisi horizontal 640 pixel). Terdapat eror sekitar 20 pixel ketika target sudah diam di tengah *frame* (detik 0 hingga detik 10). Jika dibandingkan dengan lebar *frame* yang digunakan, eror tersebut adalah sekitar 1.8% dari keseluruhan lebar *frame*. Nilai tersebut cukup kecil sehingga dapat disimpulkan bahwa *bounding box* target relatif berada di tengah *frame*. Eror posisi *bounding box* pada *frame* dapat disebabkan oleh *human error*, variasi gambar RGB yang didapat oleh kamera, dan juga akumulasi eror dari *bounding box* yang diprediksi oleh pelacak.

Pada Gambar 8 terlihat bahwa robot selalu berusaha untuk menjaga jarak 1-meter dari target. Dari grafik tersebut, terlihat bahwa robot memiliki kecepatan konstan 0.4 m/s (kecepatan maksimal robot) hingga jarak antara target dan robot sudah mendekati *setpoint* (detik 1 hingga detik 10). Selanjutnya pada detik 10 hingga detik 14, robot relatif diam menjaga targetnya tetap berada di jarak 1 meter. Nilai error yang didapatkan pada saat robot diam sangat kecil, dengan error rata-rata sebesar -2.92 mm dan error terbesar berada di nilai 9.92 mm dari nilai seharusnya (1140 mm).

2) Pengujian Ketujuh: Pengujian dengan Adanya Orang Lain di Ruang

Pengujian selanjutnya yang akan ditampilkan adalah pengujian dengan adanya orang lain yang terdeteksi oleh robot selain dari orang yang dijadikan target. Pengujian ini adalah pengujian ketujuh dari sepuluh pengujian yang dilakukan. Data dapat dilihat pada Gambar 10, Gambar 11, dan Gambar 12.

Terlihat pada Gambar 10, pada *frame* pertama terdeteksi dua orang. Selanjutnya, orang yang terdekat dari robot

dijadikan target yang diikuti. Dari pengujian ini didapat hasil bahwa robot berhasil mengikuti target yang sama dari awal hingga akhir.

3) Pengujian Kesembilan: Hasil Pengujian dengan Variasi Cahaya pada Ruang

Pengujian ini dilakukan dengan meredupkan cahaya di ruangan KATK ITS. Pada pengujian ini *depth image* yang didapat dari ZED2i *Stereo Camera* juga diperlihatkan. Pengujian ini adalah pengujian kesembilan dari seluruh sepuluh pengujian yang dilakukan. Data dapat dilihat di Gambar 13, Gambar 14, dan Gambar 15.

Terlihat pada Gambar 13, Gambar 14, dan Gambar 15, robot berhasil mengikuti target walaupun variasi pencahayaan yang diberikan sangat jauh dari awal hingga akhir. Hal tersebut dimungkinkan karena pelacak *Boosting* selalu dikoreksi setiap 20 *frame*, atau ketika gagal, oleh detektor YOLOv7.

C. Evaluasi Performa dan Perbandingan dengan Penelitian Terdahulu

Pada penelitian ini, sistem yang dibuat ditujukan untuk menjaga jarak 1 meter dari seseorang yang dijadikan target. Sehingga, untuk mendapatkan akurasi gerakan robot, nilai yang dibandingkan adalah saat robot berhenti pada jarak tersebut dari target, bukan saat robot sedang bergerak. Data tersebut dapat dilihat di Gambar 16. Pada data tersebut, rata-rata jarak robot dari target saat diam adalah 1137.98 mm, sementara jarak yang diinginkan adalah 1140 mm. Eror RMS dari data tersebut adalah 41.88 mm (3.67% dari jarak yang diinginkan) dengan standar deviasi sebesar 35.59 mm. Eror maksimal dari data tersebut adalah 320.369 mm yang didapat pada saat kondisi ruangan gelap (pengujian sistem kesepuluh).

Selain akurasi, diperlukan perhitungan performa berupa kecepatan komputasi sistem selama pengujian. Waktu komputasi tiap iterasi dapat dilihat pada Gambar 17. Dapat dilihat persebaran data waktu komputasi untuk memproses tiap *frame* selama pengujian. Rata-rata waktu komputasi yang dibutuhkan pada keseluruhan pengujian berada adalah 0.0582 detik. Dengan kata lain, rata-rata *frame rate* pada sistem ini adalah senilai 17.18 FPS. Hasil ini lebih besar dari hasil penelitian yang berada pada 12 FPS (rata-rata waktu komputasi 0.083 detik) [8]. Namun, *frame rate* dari penelitian ini lebih kecil dari penelitian yang dilakukan yang berada pada 23.81 FPS (rata-rata waktu komputasi 0.042 detik) [15].

Waktu rata-rata yang dibutuhkan untuk mendapatkan gambar dan *depth map* dari kamera stereo, melakukan pelacakan, koreksi pelacak menggunakan detektor YOLO, perhitungan kecepatan dan publish, dan proses lainnya secara berurutan adalah sebesar 11.03 ms, 20.54 ms, 280.63 ms, 6.34 ms, dan 4.812 ms. Sehingga, total waktu yang dibutuhkan dapat dihitung dengan persamaan 5.

$$T_{tot} = T_1 + T_2 + T_3 + T_4 + T_5 \quad (5)$$

Dimana T_1 , T_2 , T_3 , T_4 , dan T_5 adalah waktu yang diperlukan untuk menjalankan masing-masing proses. Berdasarkan persamaan 5, jika seluruh proses tersebut berjalan secara berurutan pada setiap *frame*, rata-rata waktu total yang dibutuhkan adalah sebesar 323.36 ms. Dengan kata lain, *frame rate* yang didapat akan bernilai 3.09 FPS. Nilai tersebut berbeda dengan rata-rata *frame rate* sesungguhnya

yang berada pada nilai 17.18 FPS. Perbedaan tersebut disebabkan oleh penggunaan detektor yang hanya dilakukan setiap 20 *frame* saja, sehingga efek dari penggunaan YOLO tidak terlihat pada setiap *frame*.

Penelitian yang dilakukan menuliskan bahwa sistem yang mereka buat mampu untuk bekerja pada variasi cahaya yang tidak begitu signifikan [15]. Tetapi, pada saat terjadi perubahan pencahayaan yang signifikan, sistem yang mereka buat tidak dapat bekerja dengan baik karena limitasi dari kamera yang digunakan. Namun, variasi cahaya yang signifikan tersebut tidak dijelaskan lebih lanjut lagi pada penelitian [15]. Sistem yang dibuat pada penelitian ini berhasil ketika sistem dijalankan dengan variasi pencahayaan yang cukup signifikan seperti yang ditunjukkan Gambar 13. Walaupun begitu, tidak adanya metode identifikasi target pada penelitian ini menyebabkan robot tidak dapat melacak kembali orang yang dijadikan target ketika orang tersebut keluar dari pandangan robot.

V. KESIMPULAN

Pada penelitian ini, robot telah diuji secara ekstensif dalam berbagai kondisi mulai dari posisi awal dan akhir target pada map, adanya banyak orang yang terdeteksi oleh detektor, dan juga variasi cahaya pada ruangan. Sistem yang telah dikembangkan berhasil untuk mengikuti manusia di dalam ruangan dengan baik. Pada saat robot berhenti pada jarak 1 meter di depan target, terdapat eror RMS sebesar 41.88 mm dan standar deviasi sebesar 35.59 mm. Nilai eror terbesar yang didapat bernilai 320.369 mm yang terjadi ketika sistem dijalankan pada ruangan gelap. Selain itu, dengan melakukan pelacakan yang dikoreksi oleh detektor setiap 20 *frame*, robot dapat mengikuti manusia dengan nilai *frame rate* rata-rata 17.18 FPS. Sistem ini juga berhasil untuk mengikuti seseorang walaupun diberikan variasi cahaya yang signifikan.

Terdapat beberapa kekurangan dari sistem yang dibuat. Pertama, penggunaan IoU terbesar antara *bounding box* pelacak dan detektor menandakan bahwa robot tidak benar-benar mengetahui perbedaan target dari orang lainnya. Jika target yang diikuti keluar dari *frame* atau terhalang oleh orang lain, robot akan kehilangan target tersebut. Kekurangan lainnya adalah robot tidak dapat mengikuti manusia dengan baik pada ruangan yang sangat gelap.

DAFTAR PUSTAKA

- [1] D. Han and Y. Peng, "Human-Following of Mobile Robots Based on Object Tracking and Depth Vision," *3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*, Shanghai, China, 16-18 October 2020, pp. 105–109.
- [2] A. Tomoya, S. Nakayama, A. Hoshina, and M. Sugaya, "A mobile robot for following, watching and detecting falls for elderly care," *Procedia Comput. Sci.*, vol. 112, pp. 1994–2003, 2017.
- [3] M. M. Islam, A. Lam, H. Fukuda, Y. Kobayashi, and Y. Kuno, "A Person-Following Shopping Support Robot Based on Human Pose Skeleton Data and Lidar Sensor," in *Intelligent Computing Methodologies: 15th International Conference, ICIC 2019, Nanchang, China, August 3-6, 2019*, pp. 9–19.
- [4] S. Nishimura, H. Takemura, and H. Mizoguchi, "Development of Attachable Modules for Robotizing Daily Items-Person Following Shopping Cart Robot," *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, China, 15-18 December 2007, pp. 1506–1511.
- [5] B. Ilias, S. A. Shukor, S. Yaacob, A. H. Adom, and M. H. M. Razali, "A nurse following robot with high speed Kinect sensor," *ARPN J. Eng. Appl. Sci.*, vol. 9, no. 12, pp. 2454–2459, 2014.
- [6] M. S. Hassan, M. W. Khan, and A. F. Khan, "Design and development of human following robot," *Student Research Paper Conference*, vol. 2, no. 15, 2015.
- [7] J. Satake and J. Miura, "Robust Stereo-Based Person Detection and Tracking for a Person Following Robot," in *ICRA Workshop on People Detection and Tracking*, Kobe, Japan, May 2009, pp. 1–10.
- [8] E. Petrović, A. Leu, D. Ristić-Durrant, and V. Nikolić, "Stereo vision-based human tracking for robotic follower," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 5, p. 230, 2013.
- [9] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot," in *Computer Vision Systems: 11th International Conference, ICVS 2017, Shenzhen, China, July 10-13, 2017*, pp. 300–313.
- [10] N. G. Shakev, S. A. Ahmed, V. L. Popov, and A. V. Topalov, "Recognition and Following of Dynamic Targets by an Omnidirectional Mobile Robot using a Deep Convolutional Neural Network," *International Conference on Intelligent Systems (IS)*, Funchal, Portugal, 25-27 September 2018, pp. 589–594.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, pp. 779–788.
- [12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," *Conference on Computer Vision and Pattern Recognition (CVPR)*, Canada, 18-22 June 2023, pp. 7464–7475.
- [13] K. Agrawal and R. Lal, "Person Following Mobile Robot using Multiplexed Detection and Tracking," *Advances in Mechanical Engineering (ICAME)*, Singapore, 2020, pp. 815–822.
- [14] V. Tadic *et al.*, "Perspectives of Realsense and ZED depth sensors for robotic vision applications," *Machines Journal*, vol. 10, no. 3, p. 183, 2022.
- [15] R. Algabri and M.-T. Choi, "Deep-learning-based indoor human following of mobile robot using color feature," *Sensors Journal*, vol. 20, no. 9, p. 2699, 2020.