

MODEL DETEKSI RAMBU UNTUK SISTEM NAVIGASI *PROTOTYPE* AGV

Damar Yusuf Sutrisno^{a*}, Munadi^b, Joga Dharma Setiawan^c

^aProgram Studi Sarjana, Departemen Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

^{b,c}Departemen Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

Jalan Prof. Sudarto No. 13, Tembalang, Kec. Tembalang, Kota Semarang, Jawa Tengah 50275

*e-mail: damar.yusuf1@gmail.com

Abstrak

Automated guided vehicle atau AGV adalah salah satu jenis *material handling equipment* yang banyak digunakan karena kelebihanannya dalam hal fleksibilitas. Industri *garment* merupakan salah satu yang mulai mengembangkan aplikasi AGV untuk mengangkut material yang sedang dikerjakan sepanjang plan produksi. Dikarenakan *layout* plan produksi pada beberapa industri *garment* sering diubah sesuai dengan produk yang sedang dikerjakan, diperlukan sistem navigasi untuk AGV yang mampu menangani berbagai konfigurasi lintasan pada plan produksi tersebut. Di antara beberapa komponen sistem navigasi, prosedur deteksi objek merupakan salah satu yang banyak digunakan untuk mendeteksi perintah gerak. Sebelumnya, sudah dikembangkan model deteksi objek sebagai salah satu komponen sistem navigasi oleh Al-Hafiez [8] dengan kemampuan untuk mendeteksi objek seperti miniatur orang dan rambu sederhana sebagai perintah navigasi. Penelitian ini bertujuan untuk membuat sebuah model deteksi rambu yang dikembangkan dari penelitian tersebut agar mampu mendeteksi objek berupa enam buah rambu yang modelnya dibuat berdasarkan rambu sesungguhnya sebagai perintah gerak AGV. Model deteksi rambu dibuat dengan proses *transfer learning* menggunakan *pre-trained model* yang tersedia pada TensorFlow Object Detection API yang kompatibel dengan perangkat USB Accelerator. Pengujian model deteksi rambu dilakukan setelah model selesai dibuat.

Kata kunci: AGV; deteksi objek; navigasi

Abstract

Automated guided vehicle or AGV is one of the widely used material handling equipment for its superiority in flexibility. Industries such as garments start to develop the implementation of AGV to transfer materials between nodes within the production plant. Since the production plant layout in some garment industry is frequently rearranged depending on what product is processed at the time, AGV needs to be able to handle various track configurations possible. Among many of navigation system components, object detection procedure is one of the most commonly used procedure to detect navigation commands. An object detection model had been developed by Al-Hafiez [8] as a navigation system component, which can detect objects such as human miniature and simply modeled signs as navigation commands. This work aims to build a sign detection model based on the work by Al-Hafiez which should be able to detect six different types of signs which the model is based on the commonly used road signs. The model is built by utilizing transfer learning method using a pre-trained model available within TensorFlow Object Detection API which also compatible with USB Accelerator device. A test is performed after the model has been built.

Keywords: AGV; object detection; navigation

1. Pendahuluan

Tahun 2017 merupakan tahun pertumbuhan ekonomi global terbesar yang terjadi sejak tahun 2011 [1]. Setiap industri mulai melakukan percepatan dan pertumbuhan produktivitas melalui pemberdayaan peran integrasi digital. *Additive manufacturing, autonomous robot, internet of things*, dan teknologi lainnya yang terkait dengan industri 4.0 menjadi akselerator tercapainya fleksibilitas dan efisiensi biaya [2]. Salah satu aspek yang menjadi fokus perkembangan integrasi digital yang terkait dengan sistem automasi adalah dalam hal *material handling equipment* (MHE). MHE yang dipakai juga akan memengaruhi biaya produksi, biaya operasional, dan laju produksi [3].

AGV atau *automated guided vehicle* merupakan salah satu MHE yang umum digunakan dalam industri karena fleksibilitasnya yang baik, di mana konsep dari AGV adalah kendaraan angkut bertenaga baterai dan beroperasi secara *driverless*, dengan kemampuan program untuk bisa memilih rute dan posisi [4]. AGV banyak dikembangkan untuk aplikasi proses angkut material seperti menyuplai *part* ke area *assembly* [5], proses transfer barang pada *warehouse* [6-8], maupun untuk aplikasi *unmanned warehouse system* [9-11]. Industri seperti *garment* juga sudah mulai

mengembangkan AGV untuk nantinya bisa mengangkat material yang sedang dikerjakan sepanjang *line* produksi. Di beberapa kasus plan produksi garmen, *layout* mesin sering diubah menyesuaikan produk yang sedang dikerjakan untuk menghemat waktu transfer antar proses dengan mesin yang berbeda. Dengan demikian, diperlukan sebuah sistem navigasi AGV yang mampu beradaptasi dengan berbagai kemungkinan konfigurasi lintasan tanpa banyak mengubah pengaturan robot agar tetap bisa beroperasi di plan tersebut.

Sebelumnya, Al-Hafiez [12] dalam penelitiannya sudah membuat model deteksi objek sebagai bagian dari sistem navigasi untuk *prototype* AGV. Batasan dari sistem navigasi ini adalah objek yang dideteksi masih berupa objek-objek sederhana seperti miniatur manusia serta rambu berupa tulisan atau bentuk-bentuk sederhana yang dinilai masih belum bisa digunakan pada plan industri yang sesungguhnya. Artikel ini menjelaskan proses pembuatan sistem deteksi rambu yang mampu mendeteksi enam jenis rambu sebagai perintah gerak yang modelnya dibuat berdasarkan model rambu lalu lintas yang dipakai di jalan raya. Keenam rambu tersebut adalah rambu *faster* dan *slower* untuk mengatur mode kecepatan robot, rambu *stop* untuk berhenti, dan rambu *turn back*, *turn left*, dan *turn right* untuk mengatur arah hadap robot ketika berada di persimpangan. Sistem deteksi rambu ini diharapkan bisa menambah fleksibilitas sistem navigasi untuk *prototype* AGV dan mampu menjawab kebutuhan industri.

2. Bahan dan Metode Penelitian

2.1. Hardware

Sistem deteksi rambu ini nantinya akan dikombinasikan dengan sistem deteksi jalur ketika dioperasikan pada *prototype* AGV sebagai satu sistem navigasi utuh. Sistem navigasi ini dijalankan pada SBC Raspberry Pi 4 Model B dengan dilengkapi perangkat USB Accelerator. USB Accelerator merupakan perangkat yang berfungsi untuk menambah performa komputer dengan kemampuan perhitungan rendah agar bisa melakukan prosedur deteksi objek menggunakan algoritma *computer vision* dengan lebih baik. Kamera digunakan untuk menangkap video sebagai masukan untuk sistem deteksi rambu.

2.2. Desain rambu

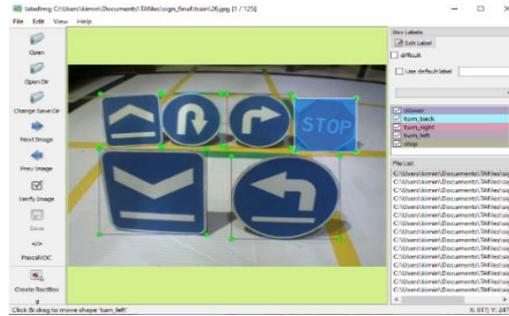
Rambu perintah berjumlah enam jenis berbeda dengan tiap rambu mewakili satu perintah gerak seperti yang telah dijelaskan sebelumnya. Berdasarkan modelnya, keenam rambu ini digolongkan ke dalam dua jenis, yaitu rambu larangan yang meliputi rambu *stop*, serta rambu perintah yang meliputi rambu lain selain rambu *stop*. Pemodelan rambu ini berdasarkan ketentuan rambu yang ada pada Peraturan Menteri Perhubungan Republik Indonesia No.13 Tahun 2014 tentang Rambu Lalu Lintas [13] dengan disertai beberapa modifikasi, diantaranya penambahan garis horizontal di bagian bawah rambu belok kiri sebagai fitur pembeda terkait dengan kesulitan sistem deteksi rambu untuk membedakannya dengan rambu belok kanan, serta rambu perintah mode kecepatan *faster* dan *slower* yang murni dibuat secara *custom*. Gambar 1 menunjukkan model rambu perintah gerak yang dipakai. Berdasarkan mode eksekusinya, keenam rambu ini juga digolongkan ke dalam dua jenis, yaitu rambu perintah langsung yang meliputi rambu *faster*, *slower*, dan *stop*, di mana perintah ini dieksekusi secara langsung ketika rambu sudah terdeteksi secara valid, serta rambu perintah tidak langsung yang meliputi rambu *turn back*, *turn left*, dan *turn right*, di mana rambu ini akan tersimpan pada *buffer* ketika terdeteksi dan hanya akan dieksekusi ketika AGV sudah memenuhi kondisi tertentu, yaitu ketika AGV berada di tengah persimpangan ketika sudah beroperasi di lintasan nantinya.



Gambar 1. Rambu perintah. Dari kiri: rambu *stop*, rambu *faster*, rambu *slower*, rambu *turn back*, rambu *turn right*, rambu *turn left*.

2.3. Pembuatan model deteksi rambu

Sebelum masuk ke pembuatan model, pengambilan gambar sebagai *training data* perlu dilakukan. Pengambilan gambar dilakukan menggunakan kamera Logitech C270 dan diambil sebanyak 150 gambar dengan masing-masing gambar mengandung 3-6 rambu dengan susunan dan posisi yang acak. Selanjutnya dilakukan proses *labelling* menggunakan *software* labelImg. Gambar 2 menunjukkan proses *labelling* salah satu gambar dalam *dataset*. Setelah *dataset* siap, proses selanjutnya adalah *training model*. Proses ini dilakukan dengan *script* yang dijalankan pada Google Colaboratory. Tahap-tahap pada proses ini meliputi pembuatan *.tfrecord file*, pemilihan *pre-trained model*, menjalankan *training* tanpa melakukan *setting hyperparameter*, *compiling model* deteksi, dan melakukan *inference test* awal. *Pre-trained model* yang digunakan adalah Quantized SSD MobileNet V2 yang di-*training* menggunakan COCO dataset. Model ini merupakan salah satu model yang kompatibel dengan perangkat USB Accelerator dengan performa yang baik dibandingkan dengan model lainnya. Terakhir, model deteksi rambu hasil *training* diproses lebih lanjut sehingga menjadi model akhir yang kompatibel dan bisa dijalankan pada perangkat USB Accelerator.



Gambar 2. Proses labelling gambar.

2.4. Metode deteksi rambu

Selanjutnya, dilakukan proses pembuatan program deteksi rambu agar model yang telah dibuat bisa diaplikasikan sebagai komponen sistem navigasi. Program ini mencakup tiga fungsi, pertama, fungsi *inference* untuk mendeteksi presensi rambu di dalam *frame*, di mana luaran fungsi ini masih belum bisa dijadikan panduan navigasi karena luarannya masih bersifat acak. Acak yang dimaksud adalah kemungkinan kondisi di mana pada suatu *frame* rambu terdeteksi, kemudian di *frame* selanjutnya rambu kembali tidak terdeteksi, lalu di *frame* selanjutnya rambu kembali terdeteksi. Fungsi kedua adalah fungsi deteksi, di mana untuk mengatasi pembacaan acak tersebut, dibuat algoritma tambahan agar pembacaan rambu menjadi valid, yaitu nilai *state* rambu hanya akan berubah apabila sistem deteksi sudah mendeteksi rambu sebanyak 10 *frame*, serta kondisi perubahan *state* rambu terjadi pada saat rambu terdeteksi di *frame* ke-10. Fungsi ketiga adalah fungsi prioritas, yaitu kondisi eksekusi rambu disusun berdasarkan nilai signifikansi tertentu, di mana *stop* memiliki berada di tingkat tertinggi, *faster* dan *slower* ada pada tingkat yang sama, serta *turn back*, *turn left*, dan *turn right* juga berada di tingkat yang sama.

2.5. Pelaksanaan Pengujian

Pengujian performa sistem deteksi rambu dilakukan dengan meletakkan rambu perintah di depan kamera pada posisi dan arah hadap yang berbeda sembari dilakukan proses deteksi rambu. Perangkat yang digunakan adalah Raspberry Pi, USB Accelerator, dan kamera yang sudah terpasang pada *prototype*. Kamera yang dipakai adalah Logitech C930e yang dilengkapi dengan *wide lens* tambahan. Pengujian ini dilakukan di dalam ruang Laboratorium RAC, lantai 3 gedung A Departemen Teknik Mesin, Universitas Diponegoro, dengan dikondisikan agar cahaya dari luar ruangan tidak masuk dan hanya menggunakan pencahayaan buatan dari lampu.

3. Hasil dan Pembahasan

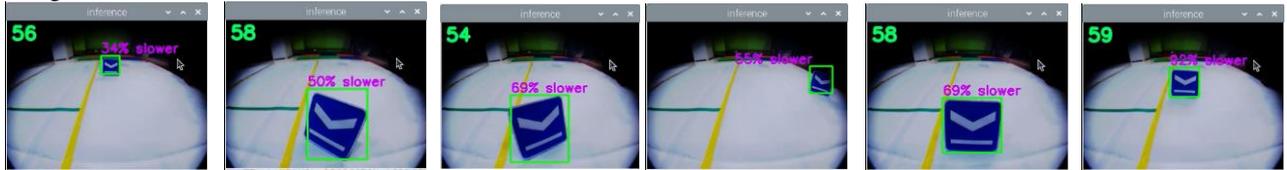
Gambar 3 menunjukkan hasil pengujian dengan meletakkan beberapa objek dalam satu *frame*. Gambar di kiri berisi rambu *stop* dengan *confidence level* sebesar 78%, rambu *faster* dengan *confidence level* sebesar 65%, dan rambu *slower* dengan *confidence level* sebesar 44%. Gambar di kanan berisi rambu *turn back* dengan *confidence level* sebesar 81%, rambu *turn left* dengan *confidence level* sebesar 69%, dan rambu *turn right* dengan *confidence level* sebesar 55%. Nilai *confidence level* tersebut berfluktuasi seiring bergantinya *frame* video pada *range* yang relatif tidak terlalu jauh dari nilai-nilai yang tertera pada gambar. Nilai pada gambar hanyalah nilai sesaat pada saat *screen capture* dilakukan.



Gambar 3. Hasil *inference* dengan beberapa objek dalam satu *frame*.

Pada kenyataannya, ketika AGV sudah dioperasikan, posisi rambu seolah bergerak relatif terhadap kamera sehingga pasti akan menimbulkan perbedaan nilai *confidence level* untuk posisi yang berbeda. Gambar 4 menunjukkan ilustrasi rambu *slower* yang diletakkan pada posisi yang berbeda ditinjau dari kamera. Gambar 4 (a) berisi rambu yang diletakkan seperti posisi ketika AGV beroperasi di lintasannya, yaitu sejauh 30 cm ke arah depan dan 30 cm ke arah kanan dari kamera. Gambar 4 (b) berisi rambu yang diletakkan sejauh 10 cm dari kamera ke arah depan. Gambar 4 (c) berisi rambu yang diletakkan sejauh 30 cm dari kamera ke arah lurus. Gambar 4 (d) berisi rambu yang diletakkan sejauh 50 cm ke arah lurus dari kamera. Nilai *confidence level* untuk keempat posisi tersebut secara berurutan adalah 55%, 69%, 92%, dan 34%. Selain posisinya terhadap kamera, arah hadap juga memengaruhi nilai *confidence level*. Gambar 4 (e) dan (f) menunjukkan ilustrasi rambu *slower* yang diletakkan pada jarak yang sama terhadap kamera, tetapi dihadapkan ke arah yang berlawanan. Gambar 4 (e) berisi rambu *slower* dengan dihadapkan sedikit ke arah kiri, sedangkan pada Gambar 4 (f) rambu *slower* dihadapkan sedikit ke arah kanan. Nilai *confidence level* untuk keduanya

secara berurutan adalah sebesar 50% dan 69%. Tren ini berlaku untuk semua rambu. Pada akhirnya, perbedaan nilai *confidence level* terhadap posisi dan arah hadap dari kamera ini tidak menjadi masalah karena *range* nilai tersebut masih berada di atas nilai *threshold* yaitu sebesar 30%, dan model deteksi ini bisa diaplikasikan sebagai komponen sistem navigasi.



Gambar 4. Deteksi rambu *slower* untuk beberapa posisi. Dari kiri: di sebelah kanan jalur, 10 cm di depan kamera, 30 cm di depan kamera, 50 cm di depan kamera, rambu menghadap sedikit ke kiri, rambu menghadap sedikit ke kanan.

4. Kesimpulan

Sebuah sistem deteksi rambu sebagai bagian dari sistem navigasi untuk *prototype* AGV telah berhasil dibuat, di mana mampu mendeteksi enam jenis rambu yang berbeda sebagai perintah gerak navigasi. Dari pengujian performa yang dilakukan, diketahui bahwa nilai *confidence level* untuk tiap rambu berbeda sesuai dengan posisi dan arah hadapnya terhadap kamera. Hal tersebut tidak menjadi masalah karena fluktuasi nilai *confidence level* tetap berada di atas nilai *threshold*, dan model deteksi ini bisa diaplikasikan untuk komponen sistem navigasi.

Saran yang bisa diberikan adalah pertama, sebaiknya, kamera yang dipakai untuk mengambil gambar untuk *dataset* disamakan dengan kamera yang akan diaplikasikan pada *prototype*, atau setidaknya kamera memiliki karakteristik gambar yang sama. Kedua, kamera yang dipilih juga sebaiknya yang memiliki sudut pandang yang lebar agar tidak perlu diberi *wide lens* tambahan yang menyebabkan distorsi di area dekat dengan sisi *frame* dan adanya area hitam di sudut-sudut *frame*. Kamera pada *prototype* memiliki luas pandangan 90°, maka kamera lain yang sudut pandangnya lebih dari itu bisa dipilih sebagai alternatif dengan tetap mempertimbangkan aspek-aspek lainnya. Hal tersebut diharapkan mampu menambah performa sistem deteksi rambu.

5. Daftar Pustaka

- [1] Suharman dan Hari W. M. 2019. Kajian Industri 4.0 untuk Penerapannya di Indonesia. *Jurnal Manajemen Industri dan Logistik*, Vol. 3, No. 1, hal 1-3.
- [2] Saturno, Maicon dkk. 2017. Proposal of an Automation Solutions Architecture for Industry 4.0. *24th International Conference on Production Research*.
- [3] Matthew, Manoj dan Sagar Sahu. 2018. Comparison of New Multi-criteria Decision Making Methods for Material Handling Equipment Selection. *Management Science Letter*, Vol. 8, hal 139-150.
- [4] Kaliappan, S. dkk. 2018. Mechanical Design and Analysis of AGV for Cost Reduction of Material Handling in Automobile Industries. *International Research Journal of Automotive Technology*, Vol. 1, No. 1, hal 1-7.
- [5] Čech, Martin dkk. 2020. Autonomous Mobile Robot Technology for Supplying Assembly Lines in The Automotive Industry. *Acta Logistica – International Scientific Journal about Logistics*, Vol. 7, No. 2, hal 103-109.
- [6] Correia, Nuno dkk. 2020. Implementing an AGV System to Transport Finishes Goods to the Warehouse. *Advances in Science, Technology, and Engineering Systems Journal*, Vol. 5, No. 2, hal. 241-247.
- [7] Dharmasiri, Pasan dkk. 2020. Novel Implementation of Multiple Automated Ground Vehicles Traffic Real Time Control Algorithm for Warehouse Operations: Dijkstra Approach. *OPERATIONS AND SUPPLY CHAIN MANAGEMENT*, Vol. 13, No. 4, hal. 396 – 405.
- [8] Yang, Qifan dkk. 2022. Multi-AGV Tracking System Based on Global Vision and AprilTag in Smart Warehouse. *Journal of Intelligent & Robotic Systems*, Vol. 104, No. 4.
- [9] Tang, Hongtao dkk. 2021. Research on Equipment Configuration Optimization of AGV Unmanned Warehouse. *IEEE Access*, Vol. 9.
- [10] Mingxing, Deng dkk. 2018. Development of Automated Warehouse Management System. *MATEC Web of Conferences*, Vol. 232.
- [11] Liu, Yubang dkk. 2019. Multi-Objective AGV Scheduling in An Automatic Sorting System of An Unmanned (Intelligent) Warehouse by Using Two Adaptive Genetic Algorithms and A Multi-Adaptive Genetic Algorithm. *PLoS ONE*, Vol. 14, No. 12.
- [12] Al-Hafiez, Muhammad. 2020. Analisis Performa Model Automated Guided Vehicle Terhadap Variasi Kecepatan Berbasis Computer Vision dan Deep Learning. Tugas Akhir. Teknik Mesin Universitas Diponegoro. Semarang.
- [13] Republik Indonesia. 2014. Peraturan Menteri Perhubungan Republik Indonesia No. 13 Tahun 2014 tentang Rambu Lalu Lintas. Kementerian Perhubungan: Jakarta.