SinarFe7
Seminar Nasional Fortei Regional 7

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

PENGENALAN RAMBU BATAS KECEPATAN PADA DUCKIEBOT

Steven Jaya Nugraha, Petrus Santoso, Handry Khoswanto.
Program Studi Teknik Elektro, Universitas Kristen Petra, Surabaya
steven90jaya@gmail.com, petrus@petra.ac.id, handry@petra.ac.id

***Abstrak* — Proyek ini menambahkan fitur pada *duckiebot* untuk mengenali rambu batas rambu kecepatan. Pengenalan rambu batas kecepatan mengunakan metode haarcascade. Hasilnya *duckiebot* dapat mengenali rambu ketika berjalan dengan tingkat keberhasilan 90%. Gangguan cahaya diberikan pada proses pembacaan dengan tingkat keberhasilan 95%. Gangguan lain juga diberikan dengan memberikan objek yang identik dengan rambu batas kecepatan dengan tingkat keberhasilan 100%.**

Kata kunci: Rambu Batas Kecepatan, Duckiebot, Duckietown, Robot Operating System (ROS) dan Raspberry Pi 3.

***Abstrak* — This project adds a feature on duckiebot to recognize speed limit signs. The recognition system of speed limit signs using the haarcascade method. The result duckiebot can recognize signs when walking with a 90% success rate. Light disturbances are given in the reading process with a recognition rate of 95%. Other shows the system can differentiate the object and actual speed limit signs with 100% recognition rate.**

Kata kunci: Rambu Batas Kecepatan, Duckiebot, Duckietown, Robot Operating System (ROS) dan Raspberry Pi 3.

I. PENDAHULUAN

Duckiebot adalah *platform* yang digunakan untuk edukasi *Autonomy* dan *research. Platform* terdiri dari *duckiebot* sebagai *autonomous vehicle* dan *Duckietown* sebagai kotanya. *Duckietown* lengkap dengan jalan, lampu lalu lintas, rintangan, serta penduduk yang dibutuhkan dalam system transportasi. *Duckiebot* medeteksi sekitarnya hanya mengunakan *camera* mono dan semua diproses dengan *Raspberry Pi*. Banyak hal yang dapat dilakukan oleh *duckietown* mulai dari mengikuti garis, hingga membaca rambu rambu lalu lintas. Semua materi yang dibutuhkan merupakan *open source,* sehingga diharapkan komunitas komunitas dapat mengambil *platform* tersebut sebagai edukasi dan *research*[1].

Di Universitas Kristen Petra, terdapat mahasiswa yang telah membuat proyek tentang *duckietown.* Pertama, Glenn Ryan Purwanto melakukan penelitian pengenalan rambu rambu. *Duckiebot* tersebut dapat berjalan sendiri serta dapat beraksi sesuai dengan perintah atau peraturan rambu- rambu tersebut. Rambu-rambu yang digunakan oleh peneliti ialah rambu lurus, rambu belok kiri, rambu belok kanan, rambu putar balik [2]. Kedua, Michael Emmanuel Victorious melakukan penelitian pada menjaga jarak aman antar *duckiebot*. Peneliti mengunakan metode *circles grid,* dimana terdapat pola *circles grid* yang terletak pada bagian belakang *duckiebot.* Program yang dijalankan akan mendeteksi pola *circles grid* untuk mengetahui koordinat dari setiap lingkaran dalam satuan pixel. Sehingga dari *circles grid* tersebut, *duckiebot* dapat menjaga jarak aman dan tidak menabrak *duckiebot* didepannya.[3]

Kemampuan mengenali rambu rambu lalu lintas adalah sesuatu yang penting dalam dunia *autonomous car*. Sistem ini dapat membantu supir jika mereka tidak sengaja melewatkan rambu rambu terutama rambu batas kecepatan. Sistem akan memberikan peringatan pada driver. Untuk kedepannya, *autonomous car* seharusnya sudah memiliki fitur tersebut untuk mengkontrol *autonomous car* agar tidak melebihi batas kecepatan pada jalan tersebut [4]. Fitur ini juga berfungsi sebagai *safety* pada kendaraan di masa depan.

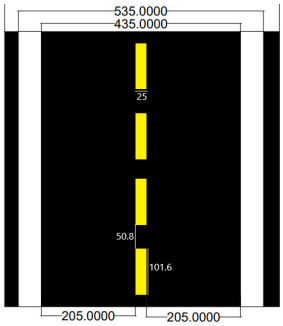
Pada proyek yang telah dilakukan oleh Glenn dan Michael belum terdapat fitur yang memiliki kemampuan untuk pembacaan rambu batas kecepatan. Maka dari itu pada proyek ini, akan berfokus pada *platform duckietown* yang dapat mengenali rambu batas kecepatan. Dilengkapi dengan mikrokontroller Raspberry pi, yang memiliki kemampuan membaca rambu batas kecepatan pada suatu jalan. Sensor yang digunakan adalah kamera *fisheye* dari *Raspberry Pi*. *Duckiebot* dapat berjalan sendiri dan menaati rambu batas kecepatan yang ada tanpa perlu dikendalikan mengunakan *remote control*. Ketika kecepatan melebihi batas maksimal maka *duckiebot* akan menyesuikan kecepatan dibawah batas maximal. Begitu juga jika kecepatan *duckiebot* dibawah kecepatan minimal maka *duckiebot* akan mempercepat sesuai dengan rambu yang sudah ditentukan.

II. PERANCANGAN SISTEM DAN IMPLEMENTASI

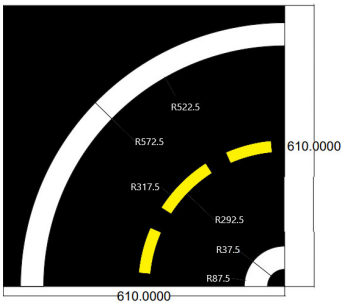
A. *Desain Arena Duckietown*

Arena *duckietown* terdiri dari 2 buah seperti yang ditunjukan pada **Gambar 1** yang merupakan desain jalur lurus dan **Gambar 2** merupakan desain jalur belok. Desain arena tersebut dirancang memiliki masing masing panjang 610 mm dan lebar 610 mm. lebar garis putih yaitu 50 mm, lebar garis kuning 25 mm, panjang garis kuning 101.6 mm dan jarak antara garis kuning dengan garis kuning yang lain 50.8 mm. Arena ini memiliki 2 jalur agar terhidar dari tabrakan dari arah berlawanan. Jarak antara jalur kiri dan kanan adalah 205 mm. Ukuran ukuran tersebut sudah sesuai *standart* yang sudah ditetapkan oleh pengembang dari *duckietown.*

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

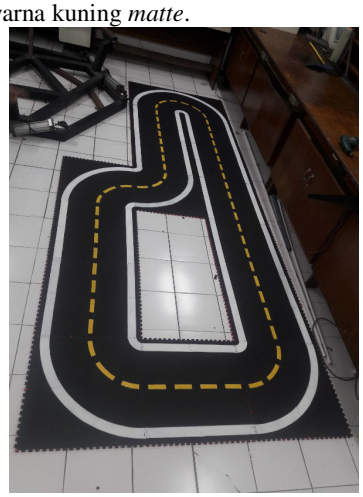


Gambar 1 Jalur Lurus



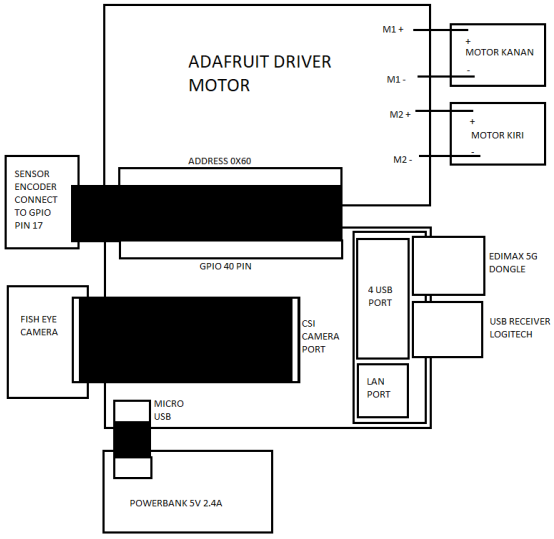
Gambar 2 Jalur Belok

Arena *duckietown* pada **Gambar 1** dan **Gambar 2** dapat digabung dan hasilnya dapat dilihat pada **Gambar 3** Arena *duckietown* terbuat dari bahan dasar *evamat* agar arena *duckietown* dapat dibentuk sesuai dengan kebutuhan. Bahan *evamat* memiliki Panjang 610 mm lebar 610 mm dan ketebalan 10 mm. *Evamat* dilapisi warna dasar hitam *matte*, pengunaan warna tersebut berfungsi untuk meminimalisir pantulan cahaya yang berasal dari lampu ruangan. Garis putih digunakan cat semprot berwarna putih *matte* dan garis kuning mengunakan warna kuning *matte*.


Gambar 3 Perakitan Arena

B. Platform Duckiebot

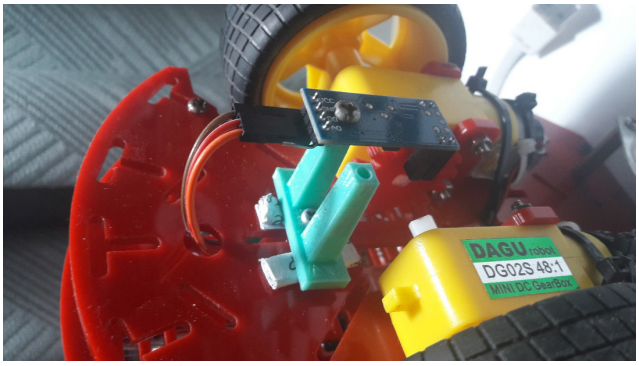
Duckiebot terdiri dari beberapa komponen yang dirangkai menjadi satu kesatuan. Komponen tersebut ialah mikroprosessor sebagai pengelola data atau bias dikatakan sebagai komputernya mengunakan *Raspberry pi 3B* atau yang lebih tipe baru. Kamera digunakan sebagai sensor, Mengunakan kamera *Rapberry pi* dengan lensa *Fisheye* agar cakupan visualnya lebih lebar. *Driver* motor yang digunakan untuk memberikan output PWM ke motor sesuai intuksi yang diberikan mikroprosessor. 2 buah motor yang sudah tergabung dengan *gearbox* dan 2 buah roda yang masing masing memiliki diameter 60 mm. *Roller Ball* yang memiliki diameter 28 mm. *Edimax 5GHz dongle* digunakan untuk memancarkan wifi yang terhubung ke Laptop atau *personal computer* untuk melakukan pemrogramman. Pada proyek ini, digunakan sensor tambahan berupa encoder yang berfungsi membaca pergerakan roda. Sensor tersebut ditambahkan dengan desain system seperti **Gambar 4**



Gambar 4 Desain Duckiebot

Pada proyek ini ditambahkan sensor encoder pada bagian roda yang memanfaatkan cahaya sebagai media pembacaan RPM pada motor. Dudukan sensor dibuat mengunakan bantuan dari aplikasi *Solidworks* kemudian direalisasikan mengunakan bantuan dari printer 3D. Sensor yang encoder yang telah terpasang pada body *duckiebot* kemudian akan dihubungan ke GPIO *Raspberry Pi 3B.* Sensor diberi aliran listrik 3.3V. *Socket VCC* pada sensor dihubungkan ke Pin 3.3V pada GPIO, *socket GND* dihubungkan ke Pin Groung GPIO, *Socket D0* dihubungkan ke Pin 17 pada GPIO. Seperti pada **Gambar 5**

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

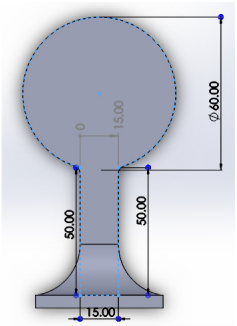


Gambar 5 Pemasangan Dudukan Sensor

C. Rambu Batas Kecepatan

Dalam proyek ini menggunakan rambu batas kecepatan yang sudah didesain oleh pengembang menyesuaikan dengan kondisi *duckiebot* dan factor ukuranya. Untuk lebih detil berikut **Gambar 6** dan **Gambar 7** merupakan desain rambu batas kecepatan yang digunakan dalam proyek ini berikut dengan ukuran yang disesuaikan dan desain warna yang menyesuaikan dengan rambu batas keceptan yang ada di dunia nyata. Rambu batas kecepatan ini memiliki diameter 60 mm sesuai dengan **Gambar 6 Desain Rambu Batas Kecepatan** Tiang rambu miliki tinggi keseluruhan 115 mm. Diameter lingkaran 60 mm, kemudian tinggi tiang 50mm dan ketebalan alasnya 5 mm.

Gambar 6 Desain Rambu Batas Kecepatan



Gambar 7 Desain Tiang Rambu

Hasil desain tiang yang telah dilakukan mengunakan aplikasi *Solidworks* dikonversikan menjadi nyata. Pengaturan control meliputi kepadatan bahan, kecepatan print, hingga estimasi berat dan lama proses *printingnya.* Lalu hasilnya ditempel pada tiang rambu yang telah dibuat sebelumnya seperti pada **Gambar 8**

Gambar 8 Rambu Batas Kecepatan

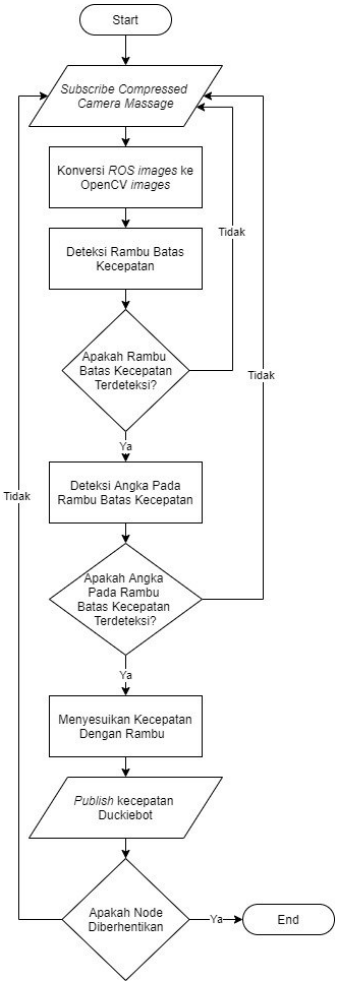
D. Program Pengenalan Rambu Batas Kecepatan

Pembuatan *file training* dimulai dari menyediakan gambar *positive* dan *negative. Positive* adalah gambar yang terdapat rambu dan negative adalah gambar yang tidak terdapat rambu. *Sample* gambar yang digunakan sebaiknya 100 gambar *positive* dan 100 gambar *negative*. Mempunyai banyak gambar *positive* dan *negative* akan menghasilkan *classifier* yang lebih akurat. Gambar *positive* dalam format .bmp dan gambar *negative* dalam format .jpg.

Setelah melakukan training pada bentuk rambu akan dihasilkan file dengan format ".XML". File ini yang akan dimasukan kedalam *duckiebot.* Mengirim file dari laptop ke *Duckiebot* penulis mengunakan aplikasi yang bernama *WinSCP*. Aplikasi ini dapat memindahkan file dari laptop ke *duckiebot* melalui transmisi local. Pertama dengan memasukan *ip address duckiebot* kemudian *username* dan *password* yang telah dibuat untuk *duckiebot.* Setelah itu akan keluar tampilan file yang terdapat di *duckiebot* Pilih file pada laptop yang akan dikirim lalu pilih folder pada *duckiebot* untuk tempat file tersebut.

Pada **Gambar 9** akan menjelaskan cara kerja dari node yang akan digunakan pada proyek ini. Proses kerja dari program tersebut dimulai dengan node tersebut melakukan *subscribe* pada *compressed camera message*. Setelah node menerima *compressed camera message* yang berisi *images* dari camera_node, maka node akan melakukan konversi *images*. Konversi *images* yang dilakukan adalah mengubah jenis *images* yang awalnya adalah ROS *images* menjadi OpenCV *images*. Dari *images* tersebut akan dideteksi apakah terdapat rambu pada gambar tersebut. Jika tidak terdapat rambu maka program akan melakukan *publish* yang dimana menyatakan bahwa tidak terdapat rambu pada gambar tersebut, kemudian node akan melakukan *subcribe* pada *compressed camera massage* kembali

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

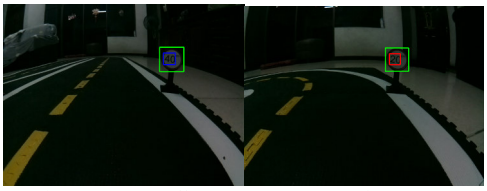


Gambar 9 Flowchart Program *Sign Detector*

III. PENGUJIAN SISTEM

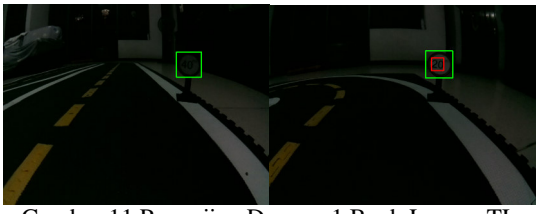
A. *Pengujian Pengaruh Intensitas Cahaya*

Pada pengujian ini adalah menguji pengenalan rambu dengan gangguan cahaya. Tujuan dari pengujian adalah mengetahui akurasi pendeteksian rambu ketika minim cahaya. Ruangan tersebut terdapat lampu TL sebanyak 3 biji yang masing masing memiliki kapasitas 16W dan temperatur cahaya 6500K. Metode Pengujian yang akan dilakukan dengan mematikan lampu sebanyak 1 buah, kemudian penulis akan tidak menjalankan *duckiebot* dengan mode *lane following* dan akan dilakukan 2 pengujian yaitu pengujian dengan mematikan 1 lampu TL dan pengujian kedua mematikan 2 lampu TL. Representasi pengujian pengenalan rambu batas kecepatan 40 km/jam pada arena *duckietown* dapat dilihat pada Gambar 10



Gambar 10 Pengujian Dengan 2 Buah Lampu TL

Pada pengujian kedua lampu TL akan dimatikan sebanyak 2 buah. Representasi pengujian pengenalan rambu batas kecepatan 40 Km/jam dan 20 Km/jam pada arena *duckietown* dapat dilihat Pada Gambar 11



Gambar 11 Pengujian Dengan 1 Buah Lampu TL

Dari hasil pengujian pengaruh intensitas cahaya terhadap pendeteksian rambu batas kecepatan pada *duckiebot* diatas yaitu ditunjukan pada Gambar 10 Gambar 11. Didapatkan hasil bahwa intensitas cahaya berpengaruh terhadap pendeteksian rambu. Pada Gambar 11 yang merupakan pengujian kedua dimana hanya 1 buah lampu yang menyala, pada rambu 40 Km/jam angka pada rambu tidak dapat terdeteksi dan pada rambu 20 Km/jam angka masih dapat terdeteksi dengan baik. Bedasarkan percobaan diatas maka pendeteksian rambu membutuhkan pencahayaan yang cukup agar rambu dapat terbaca dengan baik.

B. *Pengujian Pendeteksian Rambu Ketika Berjalan*

Pengujian pengenalan terhadap rambu batas kecepatan 40 Km/jam oleh *duckiebot.* Tujuan dilakukan pengujian adalah untuk mengetahui apakah Duckiebot dapat mendeteksi ketika sedang berjalan dengan kecepatan lebih rendah dan kecepatan lebih tinggi dari rambu 40 Km/jam yaitu setara dengan 8 Cm/s. Metode pengujian dilakukan dengan 2 cara yaitu pertama *Duckiebot* akan dijalankan dengan kecepatan sekitar 6.5 Cm/det dan kedua *Duckiebot* akan dijalankan dengan kecepatan 10 Cm/det. Hasil dari pengujian akan dipresentasikan pada Tabel 1 dibawah ini

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

Tabel 1 Percobaan 6.5 Cm/det ke 8 Cm.det

Percobaan ke	Point
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	1
Total Point	19

Keterangan:
0 = Tidak Mendeteksi Rambu
1 = Hanya Mendeteksi Rambu
2 = Mendeteksi Rambu dan Angka

Penjelasan Tabel 1 hasil pengujian pendeteksian rambu batas kecepatan 40 km/jam mendapatkan point 19 dari 20 sehingga didapat bahwa 0 kali atau *duckiebot* tidak sama sekali tidak menegenali rambu. Kedua 1 kali *duckiebot* hanya mendeteksi rambu dan tidak mendeteksi angka pada rambu. Ketiga yang paling dominan 9 kali *duckiebot* mendeteksi rambu dan angka pada rambu.

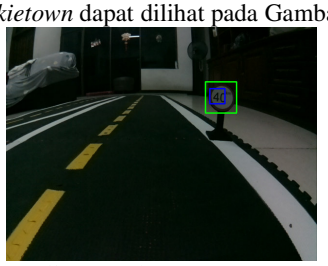
Tabel 2 Percobaan Kecepatan 10 Cm/det ke 8 Cm/det

Percobaan ke	Point
1	2
2	2
3	2
4	0
5	1
6	2
7	0
8	0
9	1
10	0
Total Point	10

Keterangan:
0 = Tidak Mendeteksi Rambu
1 = Hanya Mendeteksi Rambu
2 = Mendeteksi Rambu dan Angka

Penjelasan Tabel 2 hasil pengujian pendeteksian rambu batas kecepatan 40 km/jam mendapatkan point 10 dari 20 sehingga didapat bahwa 4 kali atau *duckiebot* tidak sama sekali tidak menegenali rambu. Kedua 2 kali *duckiebot* hanya mendeteksi rambu dan tidak mendeteksi angka pada rambu. Ketiga yang paling dominan 4 kali *duckiebot* mendeteksi rambu dan angka

Dengan demikian dapat dikatakan bahwa pergujian tingkat keberhasilan 90%, kemudian pengujian kecepatan tinggi ke kecepatan rendah memiliki tingkat keberhasilan 40%. Representasi pengujian pengenalan rambu batas kecepatan pada arena *duckietown* dapat dilihat pada Gambar 12.



Gambar 12 Pengujian Rambu Batas Kecepatan 40 Km/jam

Pengujian pengenalan terhadap rambu batas kecepatan 20 Km/jam oleh *duckiebot.* Tujuan dilakukan pengujian adalah untuk mengetahui apakah *duckiebot* dapat mendeteksi ketika berjalan dengan kecepatan yang lebih tinggi. Pada pengujian kali ini tidak menguji dengan kecepatan dibawah rambu karena kecepatan 6.5 Cm/det yang setara dengan 20 Km/jam merupakan kecepatan terendah pada *duckiebot* tersebut. Metode pengujian dilakukan dengan *duckiebot* akan dijalankan dengan kecepatan sekitar 8 Cm/det dan Kecepatan 11 Cm/det. Hasil dari pengujian akan dipresentasikan pada table dibawah ini

Tabel 3 Percobaan Kecepatan 8 Cm/det ke 6.5 Cm/det

Percobaan ke	Point
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	1
9	2
10	2
Total Point	19

Keterangan:
0 = Tidak Mendeteksi Rambu
1 = Hanya Mendeteksi Rambu

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

2 = Mendeteksi Rambu dan Angka

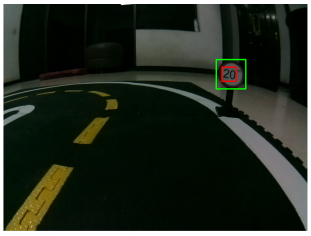
hasil pengujian pendeteksian rambu batas kecepatan 20 km/jam dengan kecepatan 8 Cm/det mendapatkan point 19 dari 20 sehingga didapat bahwa 0 kali atau *duckiebot* tidak sama sekali tidak menegenali rambu. Kedua 1 kali *duckiebot* hanya mendeteksi rambu dan tidak mendeteksi angka pada rambu. Ketiga yang paling dominan 9 kali *duckiebot* mendeteksi rambu dan angka pada rambu

Tabel 4 Percobaan kecepatan 11 Cm/det ke 6.5 Cm/det

Percobaan ke	Point
1	2
2	2
3	2
4	1
5	2
6	1
7	2
8	1
9	1
10	1
Total Point	15

hasil pengujian pendeteksian rambu batas kecepatan 20 km/jam dengan kecepatan 11 Cm/det mendapatkan point 15 dari 20 sehingga didapat bahwa 0 kali atau *duckiebot* tidak sama sekali tidak menegenali rambu. Kedua 5 kali *duckiebot* hanya mendeteksi rambu dan tidak mendeteksi angka pada rambu. Ketiga yang paling dominan 5 kali *duckiebot* mendeteksi rambu dan angka pada rambu.

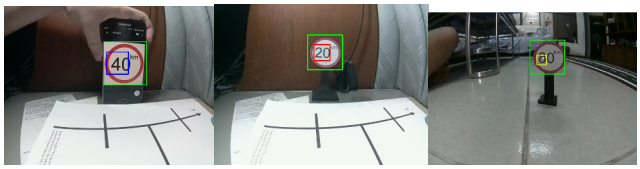
Dengan demikian dapat dikatakan bahwa tingkat keberhasilan 90% untuk kecepatan 8 Cm/det dan 50% untuk kecepatan 11 Cm/det.



Gambar 13 Pengujian Rambu Batas Kecepatan 20 Km/jam

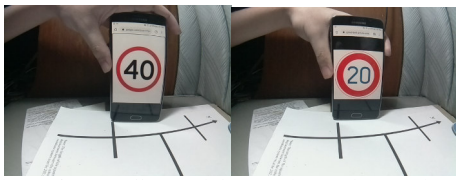
C. Pengujian Pengaruh Angka Selain Angka Pada Rambu

Pada pengujian proyek ini *duckiebot* akan ditunjukan beberapa gambar yang identic dengan rambu batas kecepatan yang telah di training sebelumnya. Pada pengujian ini pertama *duckiebot* akan diberikan rambu batas kecepatan sesuai dengan standart Indonesia seperti pada Gambar 14



Gambar 14 Pengujian Rambu Batas Kecepatan

Pengujian kedua adalah *duckiebot* diberikan rambu yang indentik dengan rambu batas kecepatan. Rambu yang diambil adalah rambu dengan warna yang sama dan terdapat angka sebagai gangguannya. Seperti yang ditunjukan pada gambar dibawah ini Gambar 15



Gambar 15 Pengujian Selain Angka Pada Rambu Batas Kecepatan

Dari hasil pengujian angka selain rambu terhadap pendeteksian rambu batas kecepatan maka, dapat disimpukan bahwa *duckiebot* dapat mendeteksi rambu batas kecepatan yang berlaku di Indonesia, dan *duckiebot* tidak mendeteksi selain rambu batas kecepatan yang sudah di training sebelumnya. Jadi *duckiebot* sudah dapat dikatakan berhasil dalam mendeteksi rambu batas kecepatan yang berlaku Indonesia.

IV. KESIMPULAN

1. Dari hasil pengujian mendeteksi rambu 40 Km/jam dan 20 Km/jam didapatkan hasil bahwa *duckiebot* memiliki tingkat keberhasilan 90% jika *duckiebot* melaju dengan kecepatan dibawah 8 Cm/det dan tingkat keberhasilan akan semakin menurun ketika kecepatan *duckiebot* dinaikkan
2. Dari hasil pengujian intensitas cahaya dalam mengenali rambu, diketahui bahwa *duckiebot* memiliki kamera sebagai sensor utama dan tentunya sensitif terhadap intensitas cahaya. Sehinga kamera membutuhkan cahaya yang cukup dan merata dalam proses pembacaanya
3. Dari hasil pengujian angka selain angka rambu batas kecepatan didapatkan hasil bahwa *duckiebot* tidak mendeteksi angka selain rambu batas kecepatan dengan tingkat keberhasilan 100%

V. DAFTAR PUSTAKA

[1] L. Paull *et al.*, “Duckietown: An open, inexpensive and flexible platform for autonomy education and

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

research," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1497–1504, 2017, doi: 10.1109/ICRA.2017.7989179.

[2] G. R. Purwanto, "Pngembangan Duckiebot Untuk Dapat Mengenal Rambu-Rambu," 2019.

[3] M. E. Victorious, "Sistem Kendali Platform Duckietown Untuk Menjaga Jarak Aman 2 Autonomus Car," 2019.

[4] J. Torresen, J. W. Bakke, and L. Sekanina, "Efficient recognition of speed limit signs," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 90, pp. 652–656, 2004.