TEKNIK

Tersedia online di: http://ejournal.undip.ac.id/index.php/teknik

TEKNIK, 43 (3), 2022, 254-263

Analisis Performa Mekanik *Autonomous Car* Dengan Metode *Region of Interest* Menggunakan Raspberry Pi 4 dan Arduino Nano

Florentinus Budi Setiawan*, Martinus Hendra Dewantara, Leonardus Heru Pratomo, Slamet Riyadi

Departemen Teknik Elektro, Fakultas Teknik, Universitas Katolik Soegijapranta,
Jl. Pawiyatan Luhur Sel. IV No.1, Bendan Duwur, Kec. Gajahmungkur, Kota Semarang, Indonesia 50234

Abstrak

Perkembangan ilmu pengetahuan dan teknologi kecerdasan buatan kini sudah mulai merambah pada bidang otomotif. Autonomous car ialah salah satunya. Teknologi ini mengambil peran termasuk diantaranya kemampuan sensor lidar, sistem Global Positioning System (GPS), dan pembacaan citra melalui kamera. Peneltian ini menganalisis autonomous car menggunakan sistem pembacaan citra dengan menggunakan kamera sebagai sensor optik. Deteksi objek diperlukam secara realtime karena autonomous car terus bergerak mengikuti lintasan. Peneliti menggunakan klasifikasi warna Hue, Saturation, Value (HSV) dengan metode Region of Interest (ROI), yang memiliki kemampuan untuk menandai area tertentu sehingga dapat digunakan untuk mengoptimalkan kinerja sistem untuk mendeteksi dan mengklasifikasi lintasan secara cepat dan tepat. Autonomous car ini menggunakan sistem 4 Wheels Drive (4WD) dengan motor Direct Current (DC) sebagai penggerak utama. Raspberry Pi 4 dan Arduino nano sebagai mainboard untuk pengoperasiannya. Metode pengujian pada penelitian ini meliputi pengujian image processing menggunakan Region of Interest, pengujian ini meliputi deteksi jalan yang sudah dirancang dan pengujian mekanikal penggerak yang digunakan pada autonomous car ini. Pada penelitian ini telah dilakukan uji coba dan prototype ini berhasil bekerja sesuai dengan algoritma yang sudah dibuat. Pada uji coba ini, Autonomous Car menggunakan metode ROI memiliki akurasi baca dan pergerakan yang sangat akurat. Pada Uji coba dan implementasi perangkat keras yang dilakukan di laboratorium autonomous car ini dengan kecerdasan buatan dapat bekerja sesuai dengan algoritma yang dibuat dengan tingkat keberhasilan sebesar 90%.

Kata kunci: *Raspberry Pi 4; Arduino Nano; sistem HSV; Region of Interest; autonomous car*

Abstract

[Title: Autonomous Car Mechanical Performance Analysis with Region of Interest Method Using Raspberry Pi 4 and Arduino Nano] *The development of artificial intelligence science and technology has now begun to penetrate the automotive sector. Autonomous car is one of them that is now starting to take on the role with several capabilities such as lidar sensors, GPS systems, and image reading through the camera. In this paper, an autonomous car uses an image reading system using a camera as an optical sensor. Object detection is needed in real time because the autonomous car continues to move along the track. Researchers use HSV color classification with the Region of Interest method, which has the ability to mark certain areas so that it can be used to optimize system performance to detect and classify trajectories quickly and precisely. This autonomous car uses a 4WD system with a DC motor as the main driver, Raspberry Pi 4 and Arduino nano as the mainboard for operation. The test method in this study includes testing image processing using the Region of Interest, this test includes road detection that has been designed and mechanical testing of the propulsion used in this autonomous car. In this study, trials have been carried out and this prototype successfully works according to the algorithm that has been made. In this trial, the AGV using the ROI method has very accurate reading and movement accuracy. In trials and hardware implementations carried out in this autonomous car laboratory with artificial intelligence, it can work according to the algorithm created with a success rate of 90%.*

*) Penulis Korespondensi.
E-mail: f.budi.s@unika.ac.id

Keywords: *Raspberry Pi 4; Arduino Nano; HSV System; Region of Interest; Autonomous Car*

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919

1. Pendahuluan

Perkembangan otomotif di dunia sangat pesat. Inovasi ini muncul pertama kali pada tahun 1885 oleh Karl Benz. Perusahaan Benz Patent Motorwagen membuat mobil bertenaga mesin bensin pertama di dunia. Mobil ini dirancang menggunakan mesin sentrifugal dan berkapasitas 954 cc. Pada tahun 1913 mobil hybrid pertama ditemukan oleh Ferdinand Porsche (Gandhi & Salvi, 2019).

Mobil *hybrid* menggunakan lebih dari satu sumber energi. Mobil jenis ini menggabungkan mesin bensin dengan motor listrik, dimana kedua sistem bekerja satu sama lain untuk menggerakkan kendaraan. Hal ini membuat mobil membakar lebih sedikit bensin sehingga mencapai efisiensi yang lebih baik daripada mesin mobil konvensional yang hanya menggunakan bahan bakar bensin (Dewangan & Sahu, 2021).

Pada perkembangan berikutnya, muncul inovasi *autonomous car.* Titik awal dimulainya penelitian mengenai *autonomous car* terjadi pada tahun 1964 oleh Stanford University yang menciptakan *autonomous car* sederhana yang diberi nama Stanford Artificial Intelligence Laboratory Cart. Teknologi yang digunakan pada *autonomous car* ini adalah deteksi visual (Koike & Sueda, 2019). Pengembangan *autonomous car* gencar dilakukan pada beberapa tahun terakhir (Choi dkk., 2010).

Kemampuan untuk dapat bergerak secara otomatis mejadi poin penting dari perkembangan *autonomous car*. Kemampuan ini didapatkan dari hasil pemetaan sebelumnya, yaitu dengan merekam lintasan terlebih dahulu. Hasil rekaman tersebut kemudian diinputkan pada program, sehingga *autonomous car* dapat mengenali *track*. *Autonomous car* akan bergerak sesuai dengan rekaman yang telah diprogram (Widodo & Prasetyaningrum, 2018). Metode lain dengan menggunakan sistem *realtime*. Metode ini memungkinkan pemetaan secara otomatis, misalnya perubahan lintasan maupun halangan yang terdapat pada lintasan sehingga dapat dihindari tanpa melakukan proses rekaman terlebih dahulu (Okuyama, Gonsalves, & Upadhay, 2018).

Region of Interest (ROI) merupakan salah satu metode segmentasi *realtime* dengan tingkat presisi dan akurasi yang cukup tinggi. ROI berfungsi untuk memilah suatu objek untuk memaksimalkan kinerja sistem (Ikhlayel dkk., 2020). Tanpa adanya ROI, pemrosesan pada citra akan dilakukan terhadap seluruh piksel citra tanpa terkecuali. Implementasi ROI pada pengolahan citra proses segmentasi dilakukan pada area yang telah ditentukan saja tanpa memotong gambar yang telah diperoleh sebelumnya (Kojima & Nose, 2018).

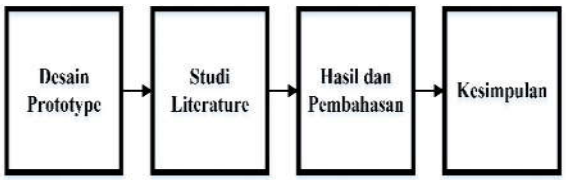
Penelitian terkait implementasi ROI telah dilakukan, termasuk untuk mendeteksi rute pelayaran. ROI digunakan untuk membatasi wilayah yang akan ditandai untuk menghemat proses komputasi, penyimpanan dan pengambilan keputusan secara tepat untuk kepentingan jangka panjang (Li, Li, & Randhawa, 2017). Implementasi ROI dapat dilakukan secara manual, namun ini menyebabkan berkurangnya tingkat akurasi. Pembuatan ROI dapat dilakukan secara otomatis dengan mendeteksi bagian terluar pada bagian atas dan bawah objek, sehingga dapat meningkatkan akurasi (Pratomo, Kaswidjanti, & Mu'arifah, 2020).

Penelitian *autonomous car* telah dilakukan dengan menggunakan *convolutional neural network* (Kafadar, 2021). Penelitian ini menerapkan sistem *convolutional neural network* yaitu dengan mendeteksi dan mengenali sebuah objek pada sebuah *image*, memanfaatkan proses konvolusi dengan menggerakan sebuah kernel konvolusi (*filter*) berukuran tertentu ke sebuah gambar, serta *local binary pattern* dengan melakukan perbandingan nilai tengah piksel dan nilai pinggir piksel terdekat dengan citra *gray scale* (Amat, Sari, & Ningrum, 2017), serta metode *learning vector quantization* (Mik & Bouchner, 2020).

Penelitian ini berfokus pada pengembangan *autonomous car* yang dikemudian hari dapat diimplementasikan untuk kendaraan transportasi ataupun digunakan sebagai alat distribusi barang otomatis yang ada di industri. Peneliti menggunakan sistem deteksi lintasan dengan menambahkan dan memodifikasi metode ROI. Raspberry Kamera V2 akan berfungsi sebagai sensor optik yeng berfungsi untuk menangkap gambar lintasan. Raspberry Pi 4 akan berperan dalam prosesi pengolahan gambar untuk deteksi jalur dan Arduino nano akan berfungsi sebagai kontrol *steering* dari *autonomous car* ini sendiri.

2. Metode Penelitian

Penelitian ini didesain dengan pendekatan *Research and Development* (R&D) berdasarkan metode *Forward Engineering*. Penelitian ini memiliki beberapa tahapan utama yaitu proses *prototype* desain yang menyangkut *prototype autonomous car*, sistem skematik pada *autonomous car,* dan komponen yang digunakan Studi literature mengenai sistem HSV dengan metode ROI*,* dan performa mekanik pada *autonomous car*. Hasil dan pmbahasan akan menyangkut mengenai hasil pengujian yang menjadi topik masalah pada *autonomous car.* Kesimpulan akan membahas mengenai hasil analisi yang terhadap *autonomous car.* Diagram blok penelitian terdapat pada Gambar 1.



Gambar 1. Diagram Blok Penelitian

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919

2.1. Desain Prototype

Autonomous car yang dikembangkan pada penelitian ini berupa *prototype* a*utonomous car* 4WD dengan skala 1:10 seperti pada Gambar 2. Pada pengujian ini *power supply* yang digunakan yaitu *accu* dengan tegangan 12V dengan kapasitas 12AH. *Accu* digunakan sebagai sumber tegangan untuk menyalakan seluruh komponen pada perangkat *autonomous car*. Untuk dapat menurunkan tegangan 12V menjadi 5V diperlukan modul *step down* untuk mensuplai pada Raspberry Pi 4 yang terhubung melalui kabel USB *type-C*. Raspberry Pi 4 dan Arduino nano berfungsi sebagai *mainboard* untuk menerima sinyal yang telah dikirim oleh kamera Raspberry V2 setelah mendeteksi lintasan, lalu data yang telah diproses oleh kamera akan diproses oleh raspberry dan Arduino untuk kontrol *steering* motor dan motor penggerak. Gambar skematik dapat dilihat pada Gambar 3. Konfigurasi perangkat keras dalam sistem ini, terdiri dari beberapa komponen pendukung utama yaitu: Raspberry Pi 4, Arduino Nano, driver motor, dan Motor DC.

2.1.1. Raspberry Pi 4

Raspberry Pi *mini-computer* yang sering digunakan untuk development beberapa *project* seperti *image processing, data analyst, data scientist,*dan masih banyak yang dapat dilakukan raspberry ini. Raspberry pi 4 memiliki port USB, LAN, *jack audio*, menggunakan *power input* tipe-C, dan HDMI untuk input dan output, hal ini dikarenakan raspberry pi 4 adalah *Single-Board Computer* yang dapat dikonfigurasikan pada program *python* sehingga menjadikan raspberry pi 4 dapat dijadikan layaknya komputer pada umumnya. Raspberry Pi 4 menggunakan SoC (System-on-a-chip) dari Broadcom BCM2835, prosesor ARM1176JZF-S 700 MHz, GPU VideoCore IV yang dikemas dan diintegrasikan diatas PCB. Pin GPIO pada raspberry pi digunakan sebagai konektor antara raspberry pi dan perangkat lain yang diintegrasikan sehingga memudahkan pada saat *wiring*. Dari 40 pin, 26 pin GPIO, berupa *input* dengan penomeran ganjil, dan *output* dengan penomeran genap. Pin GPIO yang memiliki dua sistem penomeran yaitu *Broadcom* (BCM) sebagai sistem penomeran di *software* dan GPIO board sebagai sistem penomeran pada board PCB (Jo et al. 2019).

Sistem Raspbian merupakan sistem operasi resmi yang ditunjukan langsung bagi Raspberry pi 4, yang merupakan turunan dari Linux distro debian 32-bit. Sistem operasi ini digunakan pada Open CV yang menggunakan bahasa *python*, dikarenakan lebih berfokus pada *image processing, data science, dan Internet of Things* (IOT). Sistem Raspbian memiliki lebih dari 350.000 *library precompiled* yang disajikan dalam format yang mudah di*instal* pada raspberry pi (Padmaja et al. 2019).

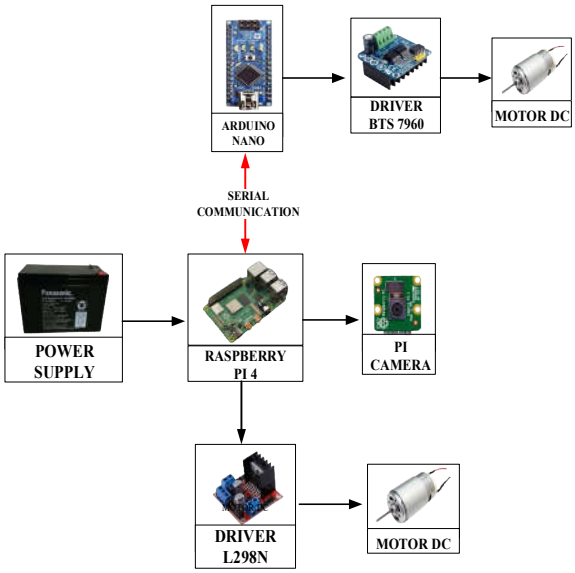
2.1.2. Arduino Nano

Arduino Nano ini adalah salah satu dari sekian banyak mikrokontroller. Arduino nano menggunakan atmega328P, dengan *serial communication* CH340, tipe *port* USB menggunakan mini USB. Pin GPIO pada Arduino nano memiliki 22 digital pin, 6 pin bagi pwm, 8 *input* analog pin. Arduino nano juga dilengkapi dengan regulator AMS1117 5V yang outputnya tersambung pada pin *output* outputnya tersambung pada pin *output* 5V dengan arus maksimal 1A. Selain itu, dimensi Arduino nano memiliki keunggulan pada sisi ukuran yang lebih kecil dari Arduino uno dan Arduino mega. Sehingga, Arduino nano hanya sedikit menggunakan ruang pada *project board* yang digunakan.

2.1.3. Driver Motor

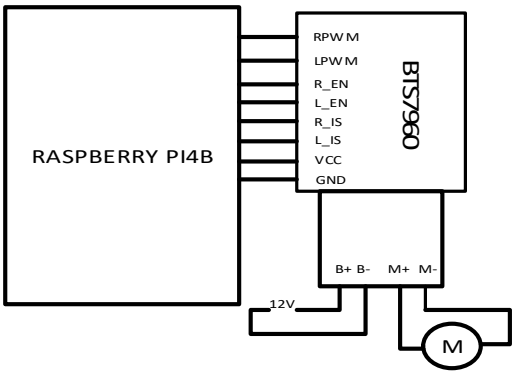
Penelitian ini menggunakan 2 tipe driver yang memiliki fungsi yang berbeda yaitu driver BTS7960 dan L298N. BTS7960 merupakan driver dengan rangkaian *full H-bridge* dengan IC BTS7960. Driver BTS7960 berfungsi untuk menggerakkan motor dc pada *autonomous car.*

Gambar 2. Prototype *Autonomous Car*

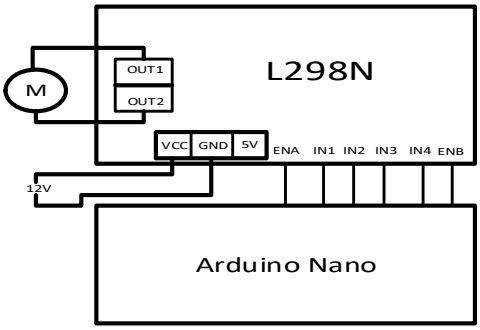


Gambar 3. Skematik *Autonomous Car*

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919



Gambar 4. Konfigurasi Motor penggerak



Gambar 5. Konfigurasi Motor Steering

Pemilihan driver motor BTS7960 karena ia mampu bekerja hingga tegangan 27V, mampu mengalirkan arus lebih dari 43A sehingga dapat dengan mudah untuk menggerakkan motor DC dengan ukuran yang jauh lebih besar (Yapriono & Dewanto, 2016). Keunggulan lain yang dimiliki yaitu mudah didapat, mudah dalam instalasi, dan memiliki kemasan atau bentuk yang kecil sehingga mudah dalam pengaturan ruang pada *project board*. L298N berfungsi sebagai pengatur *steering* roda depan, dikarenakan kemampuan motor driver L298N sebagai driver motor stepper bipolar. L298N memiliki kemampuan menggerakkan motor DC hingga 2A, dan tegangan maksimum 40V DC pada 1 kanalnya. Rangkaian driver *steering* dan penggerak dapat dilihat pada Gambar 4 dan Gambar 5.

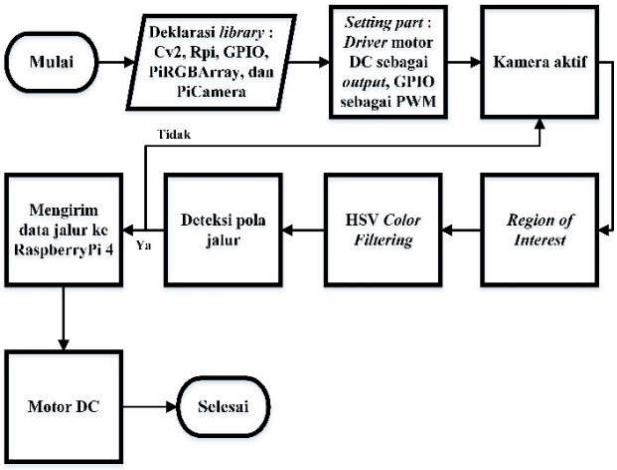
2.1.4. Motor DC

Motor DC adalah komponen penggerak yang berfungsi untuk mengubah energi listrik menjadi energi kinetik. Motor DC dikendalikan dengan mengatur kecepatan putarnya. Pengendalian kecepatan motor DC dilakukan dengan metode *Pulse Width Modulation* (PWM), yaitu dengan menaikkan *duty cyle* 30% hingga 90% dengan periode 1ms (Nakamoto dan Kobayashi, 2019). Pengendalian kecepatan putar motor DC juga dapat dikombinasikan dengan menggunakan driver BTS7960 yang dikontrol dengan Raspberry pi 4.

2.2. Sistem HSV dengan Metode ROI

Metode pemrosesan citra pada penelitian ini menggunakan metode ROI, yang merupakan metode deteksi dengan cara memilih area pada *frame* video (Rodriguez et al. 2018). *Flowchart* sistem deteksi lintasan dapat terlihat pada Gambar 6. Metode ROI ini digunakan untuk membatasi atau memperkecil area deteksi seperti yang terlihat pada Gambar 7. Pembatasan area dilakukan dengan cara menentukan *street mark* agar objek yang tidak diklasifikasikan sebagai *street mark* tidak menjadi penambah *noise* (Deshpande, Renu Madhavi, & Bhatt, 2021).

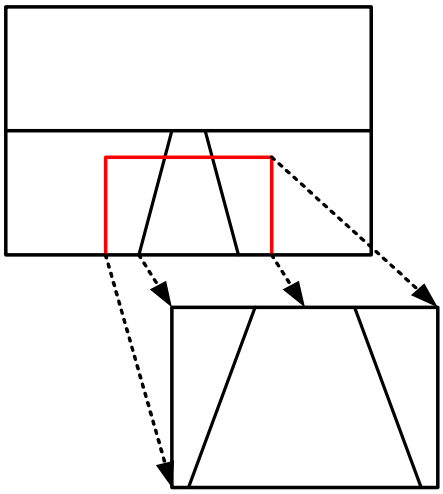
Metode ROI digunakan untuk meningkatkan waktu deteksi sehingga dapat diimplementasikan pada Raspberry Pi 4. Pengambilan metode ROI ini dilakukan dengan menggunakan kamera Raspberry V2, yang pengolahan data citranya dilakukan dengan bahasa Python dan



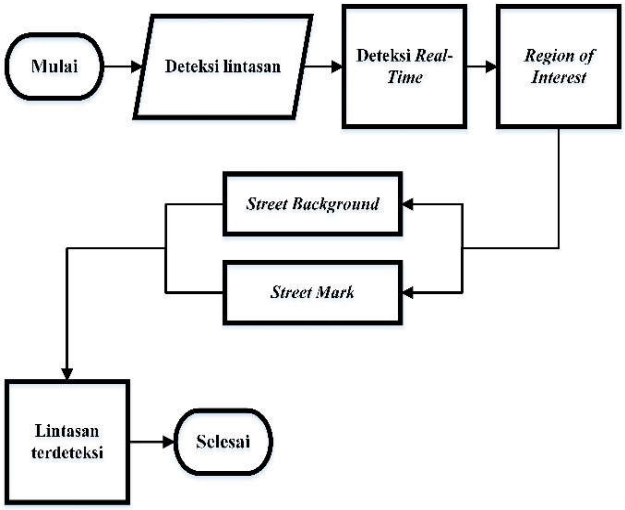
Gambar 6. Flowchart Sistem HSV ROI

Gambar 7. Hasil deteksi awal lintasan

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919



Gambar 8. Proses Penggambaran ROI



Gambar 9. Flowchart Deteksi Lintasan

menggunakan OpenCV untuk *library* pengolahan data citra. Selanjutnya citra di*blur* dengan *gaussian blur*, agar garis-garis tepi menjadi lebih halus. Deteksi tepi dilakukan dengan Fungsi Canny diatur agar maksimal, maka warna pada deteksi diubah jadi putih layaknya garis marka jalan. (Luu, Lupu, & Chirita 2019).

Ruang warna HSV terdiri atas *Hue*, *Saturation*, dan *Value*. Nilai *Hue* dapat dinyatakan dalam bilangan dinyatakan sebagai nilai persentase warna abu-abu dalam bilangan decimal sampai 1, dan nilai *Value* adalah nilai kecerahan dari warna dalam bilangan decimal 0 sampai 1 (Ab Wahab et al. 2021). Untuk dapat menentukan nilai HSV diperlukan, untuk mencari nilai (r,g,b) terlebih dahulu dengan persamaan 1.

$$r = \frac{R}{(R+G+B)}, \mathrm{g} = \frac{G}{(R+G+B)}. b = \frac{B}{(R+G+B)} \quad (1)$$

Setelah nilai (r,g,b) telah didapatkan langkah selanjutnya dalam mencari nilai HSV, dimulai dengan mencari nilai V terlebih dahulu dengan Persamaan 2.

$$V = \max(r,g,b) \quad (2)$$

Nilai V terdiri dari nilai maksimal pada (r,g,b). Nilai max didapat dari nilai maksimal warna *red, green,* dan *blue.* Setelah mendapatkan nilai V, berikutnya ialah mencari nilai S dengan Persamaan 3.

$$S = \begin{cases} 0, jika\ V = 0 \\ \frac{\max(r,g,b)-\min(r,g,b)}{\max(r,g,b)}, V > 0 \end{cases} \quad (3)$$

Nilai S didapatkan dari nilai V yang telah kita dapatkan sebelumnya dikurangi dengan nilai minimum pada (r,g,b) dibagi nilai V. Nilai minimal didapat dari nilai minimal warna *red, green* dan *blue*. persamaan dapat terjadi apabila nilai V lebih besar dari 0. Langkah terakhir dalam menentukan nilai HSV adalah dengan mencari nilai H dengan Persamaan 4.

$$H = \begin{cases} 0, jika\ S = 0 \\ 60\ \left(\frac{(g-b)}{\max(r,g,b-\min(r,g,b))}\right), jika\ V = r \\ 60\ \left(\frac{(b-r)}{max(r,g,b-min(r,g,b))}\right), jika\ V = g \\ 60\ \left(\frac{(r-g)}{max(r,g,b-min(r,g,b))}\right), jika\ V = b \end{cases} \quad (4)$$

Setelah mendapatkan nilai H, maka langkah selanjutnya ialah menyempurnakan nilai H dikarenakan nilai H, harus berada pada *range* nilai 0 sampai 1, dengan Persamaan 5.

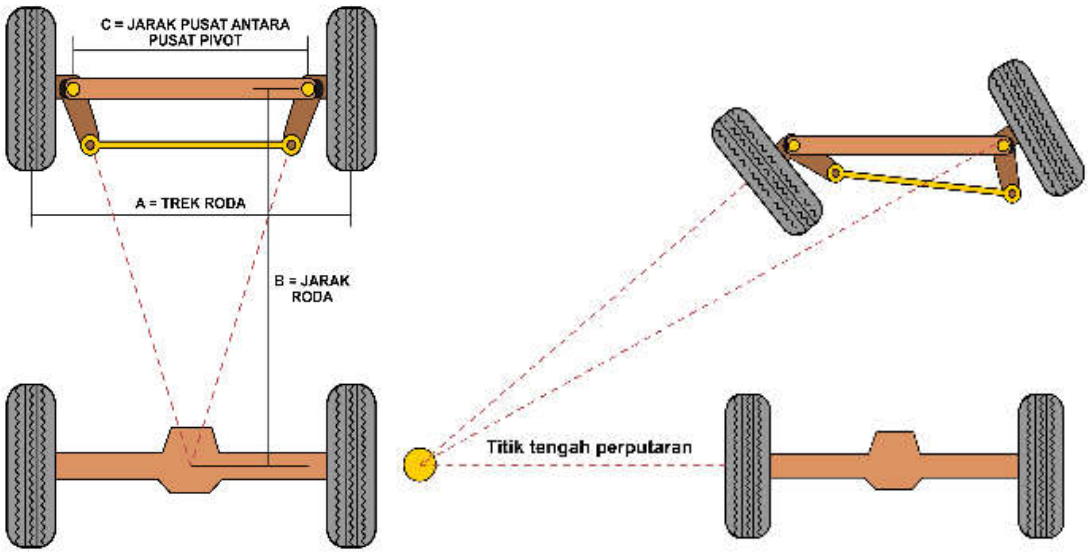
$$H = \frac{H}{6} \quad (5)$$

Tahapan untuk penentuan *region* atau wilayah pada deteksi jalur yang digunakan *autonomous car* adalah

Tabel 1. Pengujian Kalibrasi dan nilai *Threshold*

Kalibrasi	Nilai Min	Nilai Max	Objek Putih	Objek Hitam	Keterangan
Minimal ke Maksimal	155	179	Terdeteksi	Terdeteksi	Nilai *Hue*
Maksimal ke Minimal	179	179			
Minimal ke Maksimal	79	255	Sebagian	Terdeteksi	Nilai *Saturation*
Maksimal ke Minimal	255	255			
Minimal ke Maksimal	44	255	Terdeteksi	Sebagian	Nilai *Value*
Maksimal ke Minimal	94	255			

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919



Gambar 10. Sistem Ackerman pada *Autonomous Car*

dengan mengambil bagian luar frame dan gunakan titik paling luar kanan dan kiri pada tepi jalan kemudian meletakkan piksel pada sebelah kiri *frame* dikurangi dengan konstanta yang telah diatur. Pada bagian *frame* sebelah kanan letakkan piksel titik terluar ditambahkan dengan konstanta yang ditentukan. Tarik garis lurus sejajar dengan keempat titik yang sudah ditentukan sehingga terbentuk wilayah ROI. Penentuan ROI ini dimulai dari titik kiri atas hingga ke titik kanan bagian bawah. Proses penggambaran ROI dapat dilihat pada Gambar 8.

2.3. Sistem Deteksi Lintasan

Sistem deteksi lintasan akan dapat bekerja, setelah Raspberry Kamera V2 diaktifkan. Kamera akan menangkap semua lintasan secara *real time*. Setelah melalui pembacaan metode *Region of Interest,* maka jalur lintasan akan terbagi menjadi 2, yaitu *street mark* yang berfungsi sebagai jalur lintasan utama yang akan diikuti oleh *autonomous car,* sedangkan *street background* akan berfungsi sebagai nilai *error. Flowchart* sistem deteksi lintasan dapat terlihat pada Gambar 9.

2.4. Sistem Ackerman pada *Autonomous Car*

Sistem *steering* yang digunakan pada *autonomous car* ini menggunakan model sistem *Ackermann*, yang diasumsikan agar tidak ada slip yang terjadi antara tanah dan roda, untuk kendaraan yang bergerak pada kecepatan rendah. Sistem Ackerman mengatur agar semua roda yang diatur berada pada jari-jari lingkaran dengan titik tengah adalah sama. Jadi ketika roda belakang (sudut)-nya diperbaiki, titik tengah ini harus berada pada garis yang diperpanjang dari poros belakang. Perpotongan sumbu roda depan pada garis ini juga mensyaratkan bahwa roda depan bagian dalam diputar, saat kemudi melalui sudut yang lebih besar daripada roda luar. Sistem *steering* model *Ackermann* dapat terlihat pada Gambar 10.

Sistem kendali *Ackermann* atau sudut *Ackermann* terdapat pada Persamaan 6, dimana, a merupakan jalur trek roda, b adalah jarak roda depan dan roda belakang, sedangkan c adalah jarak pusat roda depan dengan pusat pivot.

$$\alpha = \tan^{-1}\frac{\frac{c}{2}}{b} \quad (6)$$

Nilai sudut roda dalam dijelaskan pada Persamaan 7, dimana T_{IF} merupakan radius putar roda pada bagian dalam, yang digunakan pada saat *autonomous car* berbelok ke kiri atau berlawanan arah jarum jam (*counter clock wise*). Dan nilai θ merupakan sudut putar roda bagian dalam.

$$T_{IF} = \frac{b}{\sin\theta} \quad \frac{a-c}{2} \quad (7)$$

Nilai sudut roda luar dijelaskan pada Persamaan 8, dimana T_{OF} merupakan radius putar roda bagian luar yang digunakan pada saat *autonomous car* berbelok ke kanan atau searah jarum jam (*clock wise*). Nilai merupakan sudut putar roda bagian luar.

$$T_{OF} = \frac{b}{\sin} + \frac{a-c}{2} \quad (8)$$

Persamaan 9 digunakan untuk menghitung radius putar rata-rata.

$$T_R = \frac{T_{IF}+T_{OF}}{2} \quad (9)$$

Setelah pengaturan sistem *steering* pada *autonomous car,* yang selanjutnya ialah pengaturan kecepatan motor DC dilakukan dengan menentukan frekuensi terlebih dahulu dari hasil rpm yang didapat menggunakan tachometer, dengan Persamaan 10.

$$rpm = \frac{v\ putaran}{60\ s} \quad (10)$$

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919

Setelah mendapatkan nilai frekuensi kecepatan putar sudut dapat dicari dengan persamaan 11.

$$\omega = 2\pi f \quad (11)$$

Sedangkan untuk menentukan nilai kecepatan linear diperlukan data jari-jari roda dan berat *prototype*

$$v = rw \quad (12)$$

3. Hasil dan Analisa

Pengujian dilakukan untuk mengetahui kemampuan dari *prototype autonomous car* yang dirancang dengan melakukan ujicoba kemampuan deteksi jalur, keakurasian, dan kecepatan putaran roda yang dihasilkan oleh *prototype autonomous car ini.*

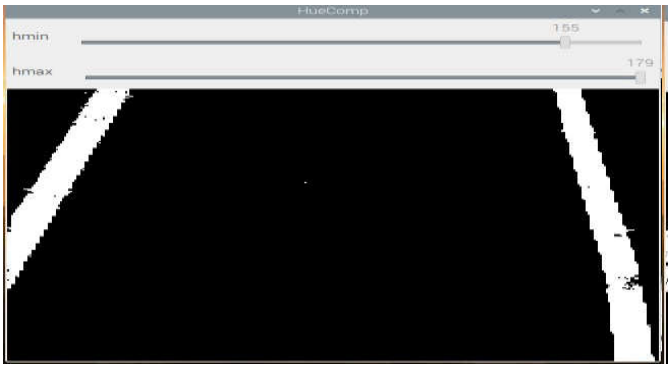
3.1. Pengujian *image processing*

Pada tahap ini akan dilakukan pengujian deteksi jalur dengan menggunakan metode *Region of Interest* untuk membedakan antara *street mark* dengan *sreet background*, dan juga menggunakan metode HSV *color filtering* dengan cara mengubah citra RGB menjadi citra *grayscale,* selanjutnya gambar *grayscale* dirubah menjadi gambar *threshold* menjadi warna hitam (0), dan juga putih (255) tanpa adanya gradasi. Pengujian terhadap penentuan nilai *threshold hue, saturation,* dan *value* dilakukan untuk membedakan *street mark* dan *street background* dilakukan sesuai dengan kondisi pencahayaan ruangan.

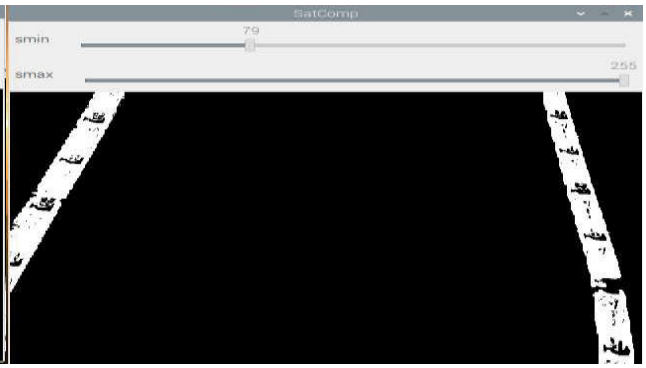
Berdasarkan data yang telah diperoleh pada Tabel 1, dapat diihat bahwa pengujian nilai *Hue* mampu mendeteksi objek hitam dan putih secara maksimal. Setelah nilai hue didapatkan langkah selanjutnya ialah menentukan nilai *saturation*. Pada pengujian nilai *saturation* menghasilkan bahwa pengujian nilai *saturation* mampu mendeteksi objek hitam pada nilai maksimal dan objek putih pada nilai minimal. Setelah nilai *saturation* didapatkan langkah terakhir ialah menentukan nilai value. Pada pengujian nilai vue didapati hasil bahwa pengujian nilai value mampu mendeteksi objek hitam pada nilai minimal dan objek putih pada nilai maksimal. Hasil deteksi nilai Hue, nilai *Saturation*, dan Value dapat dilihat pada Gambar 11, Gambar 12, dan Gambar 13.

3.2. Pengujian Keakurasian *Autonomous Car*

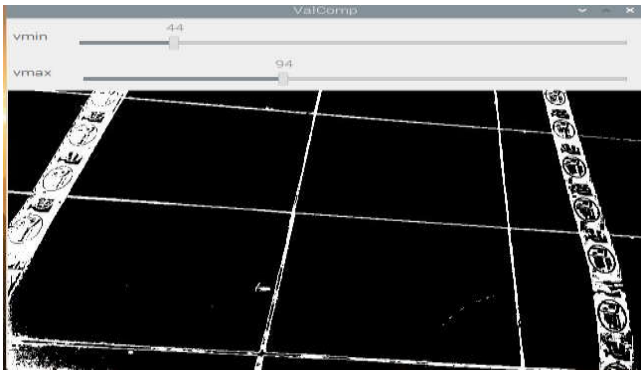
Pengujian keakurasian dilakukan untuk mengetahui kondisi prototype *autonomous car* yang diuji pada jalur lintasan yang telah ditentukan. Melalui nilai error yang dihasilkan dari selisih antara nilai lebar street mark dan jarak tangkap pixel kamera pada objek. Berdasarkan hasil pengukuran jarak maksimal yang dapat terdeteksi oleh prototype *autonomous car* yaitu 20 cm Selain itu, hasil pengukuran nilai maksimal pixel yang adalah 88 pixel. Nilai akurasi dan nilai error pada pengujian keakurasian dapat dilihat pada Tabel 2.



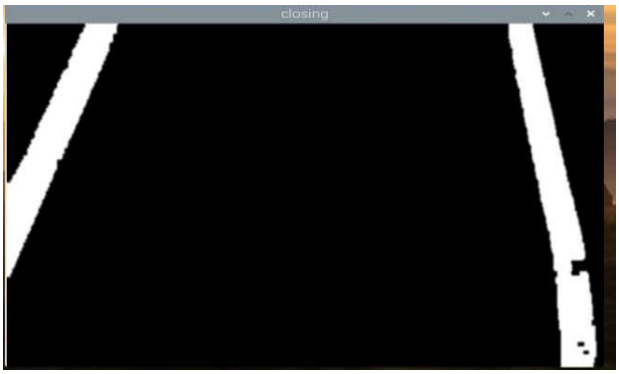
Gambar 11. Hasil Deteksi *Hue*



Gambar 12. Hasil Deteksi *Saturation*

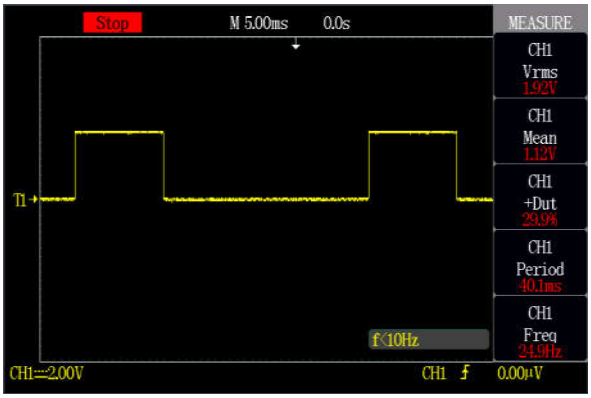


Gambar 13. Hasil Deteksi *Value*

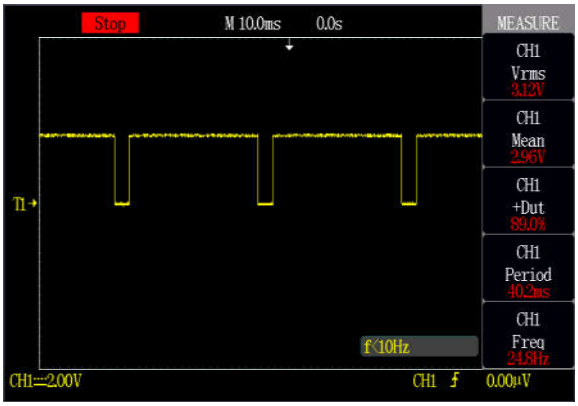


Gambar 14. Hasil deteksi akhir lintasan

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919



Gambar 15. Hasil pwm *dutycyle* 30%



Gambar 16. Hasil pwm *dutycyle* 90%

Berdasarkan data pada Tabel 4, diperoleh hasil bahwa kinerja *autonomous car* sangat efektif pada jalur lurus hingga mencapai tingkat keakurasian hingga 95,38%. Hal ini dikarenakan nilai PWM dan error akan terlihat rata (nilai negatif dan positif). Akan tetapi pada saat *autonomous car* berbelok ke kanan, nilai keakurasiannya menurun hingga 91,66% hal ini dikarenakan nilai error akan banyak bernilai positif karena deteksi jalan yang berbelok kanan, namun tidak signifikan dikarena jalur yang dibuat tidak memiliki belokan yang tajam ke kanan. Dan pada saat berbelok ke arah kiri, nilai keakurasiannya juga menurun yaitu pada angka 91,15% hal ini disebabkan nilai errornya banyak yang bernilai negatif, dan untuk menggerakkan ke arah kiri, nilai PWM motor DC sebelah kanan akan bernilai lebih tinggi dibandingkan nilai PWM motor DC sebelah kiri. Untuk hasil setelah kita mengatur filter warna HSV hingga kamera dapat mendeteksi jalan yang dibaca hingga sangat jelas maka robot AGV bisa langsung dioperasikan mengikuti jalur. Untuk jalur yang sudah dideteksi dapat dilihat pada Gambar 14. Setelah hasil pembacaan diperoleh maka *autonomous car* dapat beroperasi apabila *autonomous car* ini dibandingkan dengan *automated guided vehicle* yang menggunakan sensor LIDAR atau Infrared untuk mendeteksi jalur Autonomous cara yang dibuat oleh peneliti ini meminimasilir biaya produksi karena untuk deteksi jalur hanya menggunakan sebuah kamera dan penggunaan kamera ini dapat dikalibrasi dengan sangat mudah apabila dioperasikan dalam kondisi gelap ataupun terang.

Tabel 2. Hasil pengujian tingkat akurasi

Pengujian	Nilai Akurasi (%)	Nilai Error (%)
Jalur Lurus	95,38	4,62
Jalur belok kanan	91,66	8,34
Jalur belok kiri	91,15	8,85
Rata-rata	92,73	7,27

Pembacaan yang dihasilkan *autonomous car* yang dibuat oleh peneliti ini memiliki tingkat akurasi baca jalur yang cukup tinggi.

3.3. Pengujian kecepatan putar roda berdasarkan input PWM

Pada pengujian ini bertujuan untuk mengatahui kecepatan *autonomous car* secara linear, sekaligus untuk mengetahui kemampuan kecepatan pada *autonomous car* baik pada saat jalur lurus dan jalur berbelok. Kecepatan putaran, dan kecepatan linear dipengaruhi oleh nilai PWM yang ditentukan, motor DC yang digunakan dan jumlah yang digunakan, diameter roda serta berat pada *autonomous car* itu sendiri. Pengujian ini dilakukan menggunakan tachometer, yang kecepatan putaran (RPM) rodanya telah menggunakan tachometer. Pada pengujian ini, *autonomous car* memiliki berat 4,8 kg, dan memiliki jari-jari roda 14 cm. Hasil pengukuran kecepatan putar roda dan kecepatan linear dapat dilihat pada Tabel 3.

Berdasarkan data yang diperoleh dari Tabel 3. Pengujian kecepatan putaran roda dilakukan dengan mengatur kecepatan *dutycycle* yang bervariasi. Hasil gelombang yang dihasilkan dapat dilihat pada Gambar 15. Pada nilai pwm *dutycyle* 30% didapati hasil kecepatan putarannya ialah 16,2 Rad/s dan kecepatan linearnya yaitu

Tabel 3. Hasil kecepatan putar roda dan kecepatan linear

PWM Dutycycle (%)	Kecepatan Putaran (RPM)	Kecepatan Putaran (Rad/s)	Kecepatan Linear (m/s)
30	155	16,2	2,26
50	255,7	26,75	3,74
60	355,4	37,17	5,2
80	443,8	46,4	6,49
90	512,2	53,56	7,49
Rata-rata	344,42	36,01	5,03

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919

2,26 m/s. Dan pada nilai *dutycyle* 60% didapati hasil kecepatan putarannya adalah 37,17 Rad/s dan kecepatan linearnya adalah 5,2 m/s. Pada pengujian yang terakhir nilai pwm *dutycyle* dinaikkan menajadi 90% yang menyebabkan kecepatan putarnnya menjadi jauh lebih besar yaitu 53,56 Rad/s dan kecepatan linearnya sebesar 7,49 m/s. Hasil gelombang yang dihasilkan dapat dilihat pada Gambar 16. Sehingga dari data yang diperoleh pada Tabel 3 didapatkan nilai rata-rata 36,01 Rad/s dan kecepatan linearnya sebesar 5,03 m/s.

4. Kesimpulan

Dari pengujian ini dapat disimpulkan bahwa *autonomous car* telah berhasil dirancang sesuai dengan metode *region of interest* dan memiliki tingkat keakurasian hingga 95,38% pada jalur lurus, 91,66% pada jalur belok kanan, dan 91,15% pada jalur berbelok kiri. Rancangan *autonomous car* ini memiliki tingkat akurasi rata-rata sebesar 92,73%. Hasil pengujian *color filtering* didapatkan nilai *hue min* sebesar 155, *hue max* sebesar 179, nilai *saturation min* sebesar 79, nilai *saturation max* sebesar 255, nilai *value min* sebesar 44, nilai *value max* sebesar 94, sehingga didaptai hasil *street mark* secara jelas baik pada jalur lurus, maupun pada jalur berbelok. Pergerakan menggunakan metode Ackermann menggunakan penggerak motor DC dan *steering* motor DC sehingga *autonomous car* dapat bergerak dengan baik mengikuti garis jalur yang dibaca oleh kamera dengan kecepatan putar roda yang bervariasi. Algoritma dengan metode ROI menggunakan klasifikasi warna yang sudah dibuat oleh peneliti ini dapat bekerja sesuai algoritma yang dibuat dan memiliki tingkat kepresisian mencapai 90% dalam operasinya apabila dibandingkan AGV dengan menggunakan metode infrared ataupun Lidar.

Ucapan Terima Kasih

Terima kasih ditujukan kepada Jurusan Teknik Elektro Universitas Katolik Soegijapranta atas fasilitas yang memadai

Daftar Pustaka

Ab Wahab, M. N., Nazir, A., Ren, A. T. Z., Noor, M. H. M., Akbar, M. F., & Mohamed, A. S. A. (2021). Efficientnet-lite and hybrid CNN-KNN implementation for facial expression recognition on raspberry pi. IEEE Access, 9, 134065-134080.

Amat, R., Sari, J. Y., & Ningrum, I. P. (2017). Implementasi metode local binary patterns untuk pengenalan pola huruf hiragana dan katakana pada smartphone. JUTI J. Ilm. Teknol. Inf, 15(2), 152.

Choi, J. H., Chun, Y. D., Han, P. W., Kim, M. J., Koo, D. H., Lee, J., & Chun, J. S. (2010). Design of high power permanent magnet motor with segment rectangular copper wire and closed slot opening on electric vehicles. IEEE Transactions on Magnetics, 46(6), 2070-2073.

Deshpande, R. R., Madhavi, C. R., & Bhatt, M. R. (2021). 3d image generation from single image using color filtered aperture and 2.1 d sketch-a computational 3d imaging system and qualitative analysis. IEEE Access, 9, 93580-93592.

Dewangan, D. K., & Sahu, S. P. (2020). Deep learning-based speed bump detection model for intelligent vehicle system using raspberry Pi. IEEE sensors journal, 21(3), 3570-3578.

Gandhi, G. M. (2019, March). Artificial intelligence integrated blockchain for training autonomous cars. In 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (Vol. 1, pp. 157-161). IEEE.

Ikhlayel, M., Iswara, A. J., Kurniawan, A., Zaini, A., & Yuniarno, E. M. (2020, November). Traffic Sign Detection for Navigation of Autonomous Car Prototype using Convolutional Neural Network. In 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM) (pp. 205-210). IEEE.

Jo, K., Lee, M., Lim, W., & Sunwoo, M. (2019). Hybrid local route generation combining perception and a precise map for autonomous cars. IEEE Access, 7, 120128-120140.

Kafadar, Ö. (2020). RaspMI: Raspberry Pi assisted embedded system for monitoring and recording of seismic ambient noise. IEEE Sensors Journal, 21(5), 6306-6313..

Koike, A., & Sueda, Y. (2019, September). Contents delivery for autonomous driving cars in conjunction with car navigation system. In 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 1-4). IEEE.

Kojima, A., & Nose, Y. (2018, December). Development of an autonomous driving robot car using FPGA. In 2018 International Conference on Field-Programmable Technology (FPT) (pp. 411-414). IEEE.

Li, N., Li, J. S. J., & Randhawa, S. (2017). Color filter array demosaicking based on the distribution of directional color differences. IEEE Signal Processing Letters, 24(5), 604-608..

Luu, D. L., Lupu, C., & Chirita, D. (2019, June). Design and development of smart cars model for autonomous vehicles in a platooning. In 2019 15th International Conference on Engineering of Modern Electric Systems (EMES) (pp. 21-24). IEEE.

Mík, A. J., & Bouchner, B. P. (2020, June). Safety of crews of autonomous cars. In 2020 Smart City Symposium Prague (SCSP) (pp. 1-5). IEEE..

Nakamoto, N., & Kobayashi, H. (2019, October).

 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919

Development of an Open-source Educational and Research Platform for Autonomous Cars. In IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society (Vol. 1, pp. 6871-6876). IEEE.

Okuyama, T., Gonsalves, T., & Upadhay, J. (2018, March). Autonomous driving system based on deep q learnig. In 2018 International conference on intelligent autonomous systems (ICoIAS) (pp. 201-205). IEEE.

Padmaja, B., Rao, P. N., Bala, M. M., & Patro, E. K. R. (2018, August). A novel design of autonomous cars using IoT and visual features. In 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on (pp. 18-21). IEEE.

Pratomo, A. H., Kaswidjanti, W., & Mu'arifah, S. (2020). Implementasi Algoritma Region of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan. J. Teknol. Inf. dan Ilmu Komput, 7(1), 155-162..

Rodríguez, R. A., Cammarano, P., Giulianelli, D. A., Vera, P. M., Trigueros, A., & Albornoz, L. J. (2018). Using Raspberry Pi to create a solution for accessing educative questionnaires from mobile devices. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 13(4), 144-151..

Widodo, A. S., & Prasetyaningrum, P. T. (2018, October). Perancangan Aplikasi Internet of Thing (IoT) Autonomous Pada Mobil. In Seminar Multimedia & Artificial Intelligence (Vol. 1, pp. 35-38)..

Yapriono, D. H., & Dewanto, J. (2015). Perancangan Spion Elektrik Tipe Tanduk Pada Bus Pariwisata Berukuran Besar. *Jurnal Teknik Mesin Universitas Kristen Petra* 16(1):1–8. doi: 10.9744/jtm.16.1.9-16

doi: 10.14710/teknik.v43i3.47582 Copyright © 2022, TEKNIK, p-ISSN: 0852-1697, e-ISSN: 240-9919