

REPUBLIC INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202167905, 23 November 2021

Pencipta

Nama : **Dr. Florentinus Budi Setiawan, ST. MT, Rosita Herawati, ST, MIT dkk**

Alamat : Jl. Sinar Pelangi 491, Perum Sinar Waluyo, RT06 RW01 Kelurahan Kedungmudu, Kecamatan Tembalang, Semarang, JAWA TENGAH, 50273

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **LPPM Universitas Katolik Soegijapranata**

Alamat : Jl. Pawiyatan Luhur IV/1, Benda Dhuwur, Semarang, JAWA TENGAH, 50234

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Program Navigasi Autonomous Car**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 17 November 2021, di Semarang

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000289538

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Dr. Syarifuddin, S.T., M.H.
NIP.197112182002121001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Dr. Florentinus Budi Setiawan, ST. MT	Jl. Sinar Pelangi 491, Perum Sinar Waluyo. RT06 RW01 Kelurahan Kedungmundu, Kecamatan Tembalang
2	Rosita Herawati, ST, MIT	Jl. Sanggung Raya II Kav 9, RT06 RW06 Kelurahan Jatingalch, Kecamatan Candisari
3	Yonathan Purbo Santosa, S.S, M.Sc	Jl. Sekip III/20, RT02 RW06 Kelurahan Tembalang, Kecamatan Tembalang



Perancangan Prototipe Mobil Otonom untuk Kendaraan Shuttle pada Industri

Nama, Peran	Perguruan Tinggi/ Institusi	Program Studi/ Bagian	Bidang Tugas	ID Sinta	H-Index
FLORENTINUS BUDI SETIAWAN Ketua Pengusul	Universitas Katolik Soegijapranata	Teknik Elektro		6004303	1
ROSITA HERAWATI S.T, MIT Anggota Pengusul 1	Universitas Katolik Soegijapranata	Teknik Informatika	Teknik Pengolahan Citra	6011074	0

Sampai saat ini terdapat banyak penelitian yang terjadi di bidang kendaraan otomatis dan jalan raya cerdas, sejumlah penelitian kendali jelajah canggih telah diterapkan baik pada kendaraan skala penuh dan model. Penelitian ini bervariasi dalam jenis sensor yang digunakan untuk penggerak otomatis seperti inframerah, ultrasonik, magnetik atau gambar video. Jenis pengontrol yang digunakan diantaranya pengontrol klasik, logika fuzzy, jaringan saraf dan H_{∞} .

Manfaat bagi ilmu pengetahuan adalah pengembangan mobil otomatis yang selama ini belum banyak dilakukan di Indonesia. Hasil penelitian ini dapat bermanfaat bagi pengembangan metode untuk membangun sebuah sistem mobil otomatis. Manfaat bagi industri elektronik berupa tersedia desain peralatan kontrol untuk dapat mengendalikan mobil secara otomatis. Efek bagi masyarakat berupa didapatkan kendaraan masa depan yang dapat bergerak otomatis tanpa pengemudi.

Berdasarkan hasil uji sistem computer vision untuk navigasi mobil berbasis Raspberry Pi, maka dapat disimpulkan bahwa mobil bekerja dengan baik dan memiliki fisik yang kecil, sehingga terlihat rapih dan ringkas. Perpindahan mobil dari titik koordinat (x,y) sampai dengan titik koordinat (x',y') sangat tepat dan presisi. Dengan memasang modul driver motor L298N dapat memudahkan dalam men-setting program dalam bahasa python, sehingga hasil sinyal yang dikirimkan ke motor DC dapat langsung digunakan. Menggunakan motor DC yang memiliki kinerja sangat baik dalam torka maupun kecepatan yang dihasilkan. Roda yang digunakan merupakan mecanum wheels, sehingga mampu manuver dengan lincah dan cepat. Pada saat di realisasi pada roda mobil seperti kondisi riil, perlu penelitian lanjutan pada tahun kedua.

Luaran kegiatan tahun pertama penelitian ini adalah mendapatkan sistem kontrol mobil otonom yang dapat dikembangkan lebih lanjut. Hak Kekayaan intelektual sedang diusulkan berdasarkan hasil penelitian ini. Publikasi di prosiding dan jurnal internasional mulai dilaksanakan pada akhir waktu penelitian di tahun pertama. Pada tahun kedua

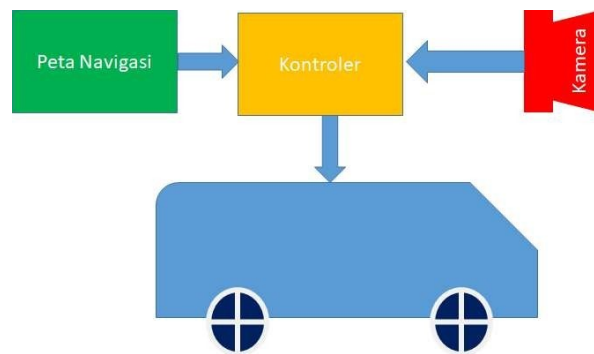
diharapkan dapat diperoleh sistem yang baik dan handal.

Kata kunci : Otonom, mobil, jalan, transportasi, image

Metode

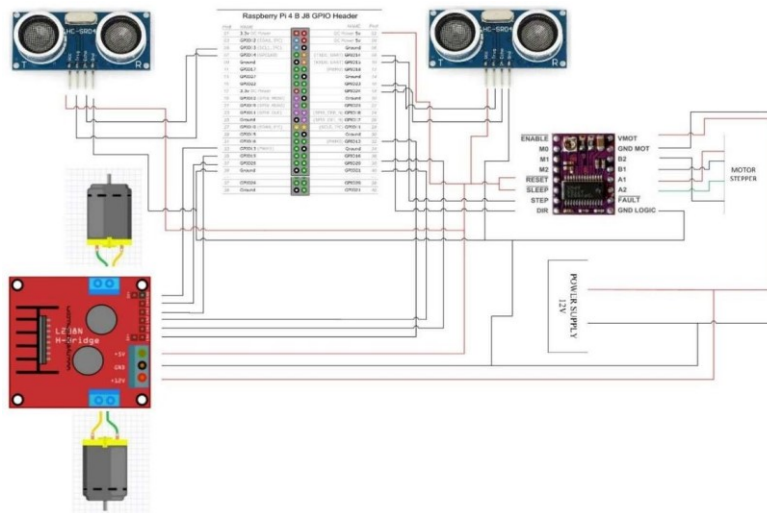
Prototype automus car menggunakan image processing, sehingga sistem dapat berjalan dengan baik. Prototipe mencakup perangkat lunak dan perangkat keras.

- Perangkat keras Python yang dipakai adalah Raspberry Pi 4. Perangkat ini mengendalikan sistem motor listrik pada prototype mobil, untuk dapat bergerak sesuai dengan image yang ditangkap oleh kamera.
- Perangkat lunak yang digunakan adalah Python pada sistem operasi Open CV Backpropagation sedang dicoba digunakan untuk sistem yang berkerja pada sistem yang sering melakukan perulangan. Penggunaan bobot untuk setiap neuron juga sangat penting untuk mempercepat input mencapai target yang ditentukan. Perangkat keras



yang lebih tinggi kemampuan komputasinya akan dapat memproses lebih banyak data yang akan digunakan.

Gambar 1. Desain Awal Mobil Otonom



Gambar 2. Realisasi Prototip Kontroler

Dalam mode otomatis, perintah mengemudi dari pengemudi manusia digantikan oleh pengontrol yang menghasilkan perintah ini dari informasi yang didapat sebagai inputnya, dalam hal ini informasi visual dari kamera yang dipasang pada kendaraan bergerak. Di sini, unit stasioner terpisah tidak lagi diperlukan, karena pengontrol dapat ditempatkan di kendaraan itu sendiri.. Kontrol lateral otomatis berdasarkan pemrosesan gambar sebenarnya merupakan proses dua langkah. Tujuan langkah pertama, pemrosesan gambar aktual, adalah untuk mendapatkan posisi kendaraan terhadap jalan dari gambar video. Pada langkah kedua, informasi ini digunakan sebagai masukan untuk algoritme kontrol. Keluaran dari algoritma ini adalah sudut kemudi yang akan menjaga kendaraan pada posisi yang diinginkan di jalan raya.

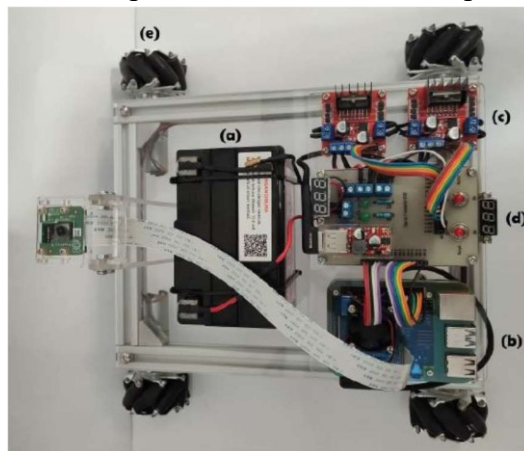
Kerangka dasar yang digunakan untuk kendaraan ini adalah model mobil kendali jarak jauh segala medan dalam skala tertentu. Oleh karena itu, mobil model mainan ini menyediakan sasis yang murah untuk penambahan yang diperlukan agar dapat digunakan untuk percobaan. Dengan menggunakan dua sinyal PWM yang berbeda, maka dimungkinkan untuk mengontrol perilaku longitudinal dan lateral kendaraan model. Salah satu modifikasi yang langsung diterapkan pada semua model kendaraan yang digunakan adalah mengganti ban plastik off- road dengan ban karet low profile yang lebih cocok untuk penggunaan di dalam ruangan.

Modifikasi perangkat keras yang sebenarnya ke unit pemancar dilakukan dengan memutuskan kedua potensiometer dan mengarahkan koneksinya ke luar unit. Unit pemancar kemudian memiliki dua port yang menerima level tegangan variabel sebagai ukuran untuk kecepatan atau sudut kemudi yang diinginkan.

Meskipun desain ini sangat sederhana dan berguna untuk pengujian cepat perangkat keras, ada sejumlah kelemahan. Yang paling serius adalah kontrol atas kecepatan dan sudut kemudi sangat tidak akurat karena cara unit pemancar / penerima dirancang: perintah kemudi dan kecepatan ditransmisikan sebagai nilai analog dalam bentuk frekuensi yang berbeda-beda. Desain yang buruk ini berperan terutama pada kecepatan rendah yang karenanya sangat sulit dicapai.

Alat Penelitian

Dalam penelitian ini, dihasilkan bentuk rancangan mobil prototipe seperti pada berikut ini yang terdiri atas beberapa bagian : komponen accu, komponen Raspberry Pi, komponen L298N dual half-bridge motor driver, komponen modul pembagi tegangan, komponen mecanum wheel, komponen motor DC dan komponen modul step-down.



Gambar 3. Bagian-bagian dari prototype : accu (a), Raspberry Pi (b), L298N (c),

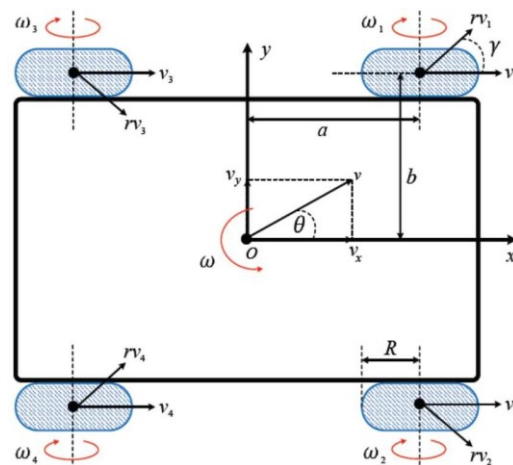
pembagi daya (d), mecanum wheel, motor DC (e) dan modul step-down.



Gambar 3. Rute Mobil Prototipe

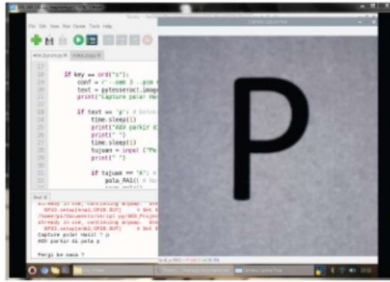
Perpindahan Mobil Prototipe dengan Computer Vision

Pengujian perpindahan pada mobil prototipe dilakukan dengan menggunakan pola berupa huruf. Pola tersebut digunakan sebagai letak titik koordinat untuk navigasi dari mobil terkini dengan menggunakan computer vision.



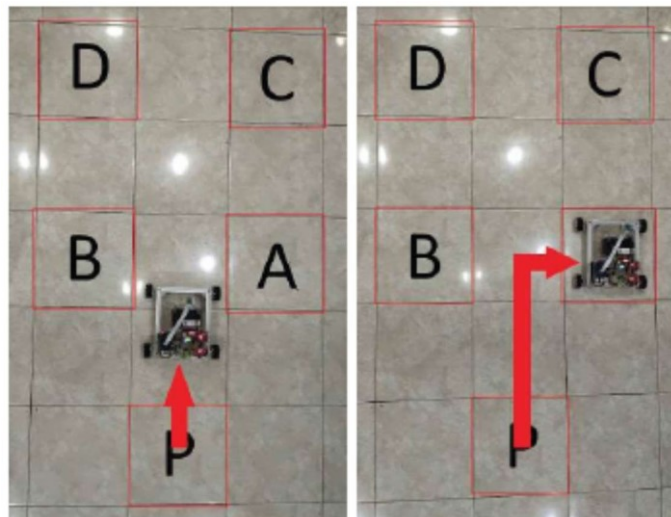
Gambar 5. Pergerakan roda

Pertama-tama, mobil berada pada titik koordinat “P. Kamera mampu mendeteksi titik koordinat tersebut. Kemudian, input tujuan untuk perpindahan mobil. Contoh menunjukkan input tujuan perpindahan ke titik koordinat “A”. Maka, mobil akan bergerak dari titik koordinat “P” menuju titik koordinat “A”. Lalu, kamera kembali mendeteksi posisi dari mobil. Bilamana posisi tersebut sudah sesuai dengan titik koordinat “A”, maka mobil akan berhenti dan menunggu untuk input tujuan perpindahan selanjutnya. Mobil dapat berhenti tepat pada titik koordinat “A”.

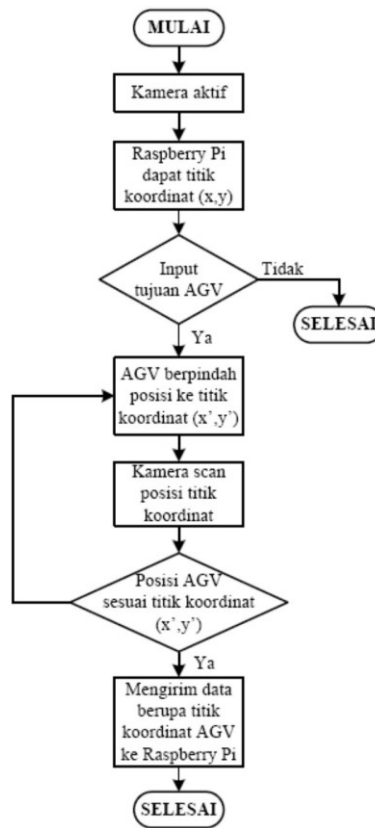


Gambar 6. Deteksi Posisi Parkir P

Selanjutnya, kembali dilakukan pengujian perpindahan posisi mobil dari titik koordinat “A” menuju titik koordinat “D”. Input perpindahan titik koordinat “D”, pergerakan mobil dari titik koordinat “A” menuju titik koordinat “D”. Lalu, kamera kembali mendeteksi posisi dari mobil. Bilamana posisi tersebut sudah sesuai dengan titik koordinat “D”, maka mobil akan berhenti seperti sebelumnya. Dapat dilihat pada Gambar 17 menunjukkan bahwa mobil berhenti tepat pada titik koordinat “D”.



Gambar 7. Pergerakan dari Posisi Parkir P ke titik A



Gambar 8. Diagram alir pergerakan prototype mobil /AGV

Hasil Pengukuran

Di bawah ini dilakukan beberapa pengukuran pada beberapa arah jalan perpindahan oleh mobil. Dengan parameter pengukuran Pulse Width Modulation (PWM) menggunakan osiloskop, kecepatan dalam RPM menggunakan tachometer digital, dan tegangan keluaran dari duty cycle yang berfungsi untuk mengatur putaran roda dengan menggunakan voltmeter.

a. Pengukuran pada Arah Jalan Maju

Pengukuran dilakukan dengan mengatur duty cycle menjadi 70% pada arah yang sama menunjukkan kecepatan yang dihasilkan yaitu 207.8 RPM. Dan tegangan keluaran sebesar 7.65 Volt DC .

b. Pengukuran pada Arah Jalan Putar

Kemudian dilakukan pengukuran pada arah jalan putar kanan dan kiri. Dengan diaturnya besaran duty cycle menjadi 80%, perubahan bentuk PWM, pada tegangan keluaran sebesar 7.89 Volt DC, dihasilkan kecepatan 215 RPM.

c. Pengukuran pada Arah Jalan Geser

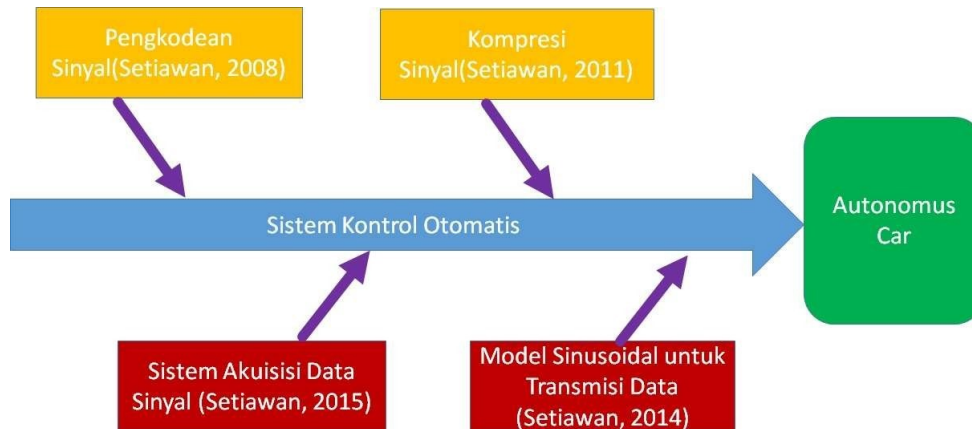
Duty cycle diatur menjadi 90% untuk pengukuran arah jalan geser kanan dan kiri, menghasilkan kecepatan 220.3 RPM tegangan keluaran sebesar 8.13 Volt DC.

Kesimpulan

Berdasarkan hasil uji sistem computer vision untuk navigasi mobil berbasis Raspberry Pi, maka dapat disimpulkan bahwa mobil bekerja dengan baik dan memiliki fisik yang kecil, sehingga terlihat rapih dan ringkas. Perpindahan mobil dari titik koordinat (x,y) sampai dengan titik koordinat (x',y') sangat tepat dan presisi. Dengan memasang modul driver motor L298N dapat memudahkan dalam men-setting program dalam bahasa python, sehingga hasil sinyal yang dikirimkan ke motor DC dapat langsung digunakan. Menggunakan motor DC yang memiliki kinerja sangat baik dalam torka maupun kecepatan yang dihasilkan. Roda yang digunakan merupakan mecanum wheels, sehingga mampu manuver dengan lincah dan cepat. Pada saat di realisasi pada roda mobil seperti kondisi riil, perlu penelitian lanjutan pada tahun kedua

Sepanjang 2020, Mitra PT Djarum Tbk aktif melakukan kunjungan ke Semarang untuk berdiskusi lebih dalam mengenai perangkat keras dan perangkat lunak. Komunikasi juga dilakukan secara online, karena kondisi Pandemi, sehingga diperoleh pemahaman yang sama. Validitas penelitian dapat ditingkatkan dan akurasi pengukuran pada tahun 2021 lebih meningkat. Mitra memberikan konsultasi mengenai perangkat lunak dan perangkat keras.

Berikut ini adalah *roadmap* penelitian secara keseluruhan. Pada tahun kedua akan direalisasikan sistem monitoring dan pengendalian pergerakan mobil yang dapat menjadi sumbangan bagi riset autonomus car yang lebih terintegrasi. Untuk mencapai luaran yang diharapkan, beberapa perbaikan metode akan dilakukan,



diantaranya adalah memperbaiki posisi kamera, algoritma dan konstruksi mobil. Perbaikan sistem penggerak juga akan dilakukan. Di sisi lain, akurasi posisi terkini juga akan ditingkatkan.

DAFTAR PUSTAKA

1. Chantara, W., Mun, JH., Shin, DW., Ho, YS, (2015), *Object Tracking using Adaptive Template Matching*, IEIE Transactions on Smart Processing and Computing, vol. 4, no. 1, February 2015 <http://dx.doi.org/10.5573/IEIESPC.2015.4.1.001>
2. Tayara, H., Soo, KG., Chong, KT., (2018) "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," IEEE Access, Vol. 6, hal. 2220–2230
3. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., (2016), You Only Look Once: Unified, Real-Time Object Detection , IEEE Conference on Computer Vision and

Pattern Recognition (CVPR)

4. Pushkin Kachroo, *Setup for advanced vehicle control systems experiments in the flexible low-cost automated scaled highway (FLASH) laboratory*, SPIE's Photonics East Symposium, Mobile Robots X, Philadelphia, PA, 1995
5. P.Kachroo, K.Ozbay, R.G.Leonard, C. " Unsal, " *Flexible Low-cost Automated Scaled Highway (FLASH) Laboratory for Studies on Automated Highway Systems*, IEEE Intl. Conf. Systems, Man, & Cybernetics, Vancouver, Canada, 1995
6. Brett Benham, *FLASH Notes*, Center for Transportation Research, 1995
7. Steven E. Shladover et al., *Automatic Vehicle Control Developments in the PATH Program*, IEEE Transactions on Vehicular Technology, Vol 40, P. 114-130, IEEE, Piscataway, NJ, 1991
8. Jesus Mena, *Finding the PATH to Automated Highways*, Berkeley Engineering - Forefront, UC Berkeley, Berkeley, CA 1990
9. D. Pomerleau, *Neural Network Based Autonomous Navigation*, In: Vision and Navigation: The Carnegie Mellon Navlab, Kluwer, Norwall MA, 1990
10. K. Liu, F.L. Lewis, *Fuzzy-Logic Based Navigation Controller for an Autonomous Mobile Robot*, Proceedings IEEE Intl. Conf. Systems, Man, & Cybernetics, P.1782-1789, Piscataway, NJ, 1994
11. Nashman, M., Schneidman, H., (1993) *Real-Time Visual Processing for Autonomous Driving*, Intelligent Vehicles Symposium Proceedings 1993, P.373-378, IEEE, Piscataway.
12. Manigel, J., Leonhard, W., (1992) *Vehicle Control by Computer Vision*, IEEE Transactions on Industrial Electronics, Vol 39, P.181-188, IEEE, Piscataway, NJ, 1992
13. Guldner, J., Utkin, V., Ackermann, J., (1994) *A Sliding Mode Control Approach to Automatic Car Steering*, Proceedings of the American Control Conference, P.1969-1973, Baltimore, MY.
14. Byrne, RH., Abdallah, C., (1994) *Robust Lateral Control of Highway Vehicles*, Intelligent Vehicles Symposium Proceedings 1994, P.375-379, IEEE, Piscataway, NJ.
15. A. Sekhon, (2015) "Face Recognition using Back Propagation Neural Network Technique," International Conference on Advances in Computer Engineering and Applications, IMS Engineering College, Ghaziabad, India, , pp. 226–230.
16. U. Andayani, E. B. Nababan, B. Siregar, M. A. Muchtar, T. Hamonangan, and I. Siregar, (2017) "Optimization Backpropagation Algorithm Based on Nguyen-Widrom Adaptive Weight and Adaptive Learning Rate," International Conference on Industrial Engineering and Applications, Nagoya, Japan, pp. 363 - 367.
17. S. S. Panda, M. S. R. S. Prasad, M. N. M. Prasad, and C. S. Naidu (2012), "Image Compression Using Back Propagation Neural Network," International journal of engineering science and advance technology, pp. 74– 78.
18. P. Viola(2001), "Rapid Object Detection using a Boosted Cascade of Simple Features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, USA, pp. 1-511
19. N. T. Jayanti, A. Rusdinar, and A. S. Wibowo, "Perancangan sistem

- pengontrolan pergerakan automated guided vehicle (agv) untuk menarik troli menggunakan sensor lidar,” e-Proceedings of Engineering, vol. 4, no. 2, 2017, pp. 1596–1603.
20. H. Wicaksono and I. Nilkhamhang, “Glocal controller-based formation control strategy for flexible material handling,” IEEE International Conference of the Society of Instrument and Control Engineers, 2017, pp. 787-792.
 21. I. Draganjac, D. Miklić, Z. Kovačić, G. Vasiljević and S. Bogdan, “Decentralized control of multi-agv systems in autonomous warehousing applications,” IEEE Transactions on

- Automation Science and Engineering, vol. 13, no. 4, pp. 1433-1447, 2016.
22. R. K. Piyare and R. Singh, "Wireless control of an automated guided vehicle," Proceedings of the International Multi Conference of Engineers and Computer Scientist, vol. II, 2011, pp. 828 – 833.
 23. Raspberry Pi, "Datasheet raspberry pi 3 model b technical specification," [Online], <http://www.farnell.com/datasheets/2027912.pdf>, tanggal akses: 16-Jun-2020.
 24. Eames, The Official Raspberry Pi Projects Book, Edition I, London: Liz Upton, 2015.
 25. Raspberry Pi, "Datasheet raspberry pi 4 model b", [Online] <http://www.farnell.com/datasheets/2819352.pdf>, tanggal akses: 16-Jun-2020.
 26. A. A. Abed and S. A. Rahman, "Computer vision for object recognition and tracking based on raspberry pi," International Conference on Change, Innovation, Informatics and Disruptive Technology ICCIIDT, 2016, pp. 177-189.
 27. W. T. Freeman et al., "Computer vision for interactive computer graphics," IEEE Computer Graphics and Applications, vol. 18, no. 3, pp. 42-53, 1998.
 28. X. Zhou, T. Chen, and Y. Zhang, "Research on intelligent agv control system," Proc. 2018 Chinese Autom. Congr. CAC, pp. 58–61, 2018.
 29. X. Lin, Y. Liu, and Y. Wang, "Design and research of dc motor speed control system based on improved bas," Proc. Chinese Automation Congress (CAC), 2018, pp. 3701– 3705.
 30. M. Sahana, S. Angadi, and A. B. Raju, "Speed control of separately excited dc motor using class a chopper," International Conference on Circuits, Controls, Communications and Computing (I4C), no. 1, 2016, pp. 1–6.
 31. R. Wen and M. Tong, "Mecanum wheels with astar algorithm and fuzzy PID algorithm based on genetic algorithm," International Conference on Robotics and Automation Sciences (ICRAS), 2017, pp. 114-118.
 32. M. Y. Naing, A. S. Oo, I. Nilkhamhang and T. Than, "Development of computer vision- based movement controlling in mecanum wheel robotic car," First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA- SYMP), 2019, pp. 45-48.
 33. Handson Technology, "User manual v1.2 esp8266 nodemcu wifi devkit," [Online] <https://pdf4pro.com/fullscreen/handson-technology-5a77a7.html>, tanggal akses: 07-Jul- 2020.

Dokumen pendukung

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

N a m a : Dr. Florentinus Budi Setiawan, ST. MT
Alamat : Jl. Sinar Pelangi 491, Perum Sinar Waluyo. RT06
RW01 Kelurahan Kedungmundu, Kecamatan
Tembalang, Semarang, Jawa Tengah

N a m a : Rosita Herawati, ST, MIT
Alamat :
, Semarang, Jawa Tengah

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya

kepada : N a m a: Lembaga Penelitian dan Pengabdian kepada Masyarakat

(LPPM)

Universitas Katolik Soegijapranata Semarang
Alamat : Jl. Pawiyatan Luhur IV/1, Bendan Dhuwur-Semarang

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer (*Program Navigasi Autonomuos Car*) untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta
Semarang, 17 November 2021
Pencipta

Dr. Berta Bekti
Retnawati, MSi,
NPP. 058.1.1998.219

Dr.Florentinus Budi S,
ST, MT
NPP. 058.1.1994.150

Rosita Herawati, ST, MIT

Syarat-syarat yang harus dilengkapi untuk pengurusan HKI (hak cipta):

1. Jenis ciptaan

Program Komputer

2. Judul

“Program Navigasi Autonomous Car”

3. Uraian singkat ciptaan

Program Komputer ini berisi informasi mengenai cara untuk mengontrol navigasi prototype autonomus car. Program mendeteksi huruf yang diletakkan pada tempat-tempat yang telah ditentukan. Selanjutnya program akan menggerakkan motor penggerak roda agar mengarah pada tujuan yang telah ditentukan. Program dapat diaplikasikan di mikrokontroler maupun sistem digital terprogram yang sejenis.

4. Tanggal pertama kali diumumkan

17 November 2021

5. Negara pertama kali diumumkan

Indonesia

6. Kota pertama kali diumumkan

Semarang

7. Contoh ciptaan (dalam bentuk File)

File

8. Scan KTP pencipta

File

9. Surat Pernyataan Karya Cipta + meterai 6 rb

Dilampirkan

10. Surat Kuasa + Meterai 6 rb

Dilampirkan

11. Scan NPWP pencipta

File

Contoh Ciptaan :

Jenis Ciptaan	File Contoh Ciptaan	Bentuk	Kapasitas Unduh
Buku	Softcopy buku	pdf	s.d. 5 MB
Program Komputer	Cover, program, dan manual penggunaan program	pdf	s.d. 5 MB
Ceramah, kuliah, pidato, dan Ciptaan lain yang sejenis dengan itu;	Suara/softcopy buku	mp4/pdf	s.d. 5 MB
Alat peraga yang dibuat untuk kepentingan pendidikan dan ilmu pengetahuan;	Foto	jpg	s.d. 1 MB
Lagu atau musik dengan atau tanpa teks;	Suara/tulisan	mp4/pdf	s.d. 5 MB
Drama atau drama musikal, tari, koreografi, pewayangan, dan pantomim;	Video/rekaman	mp4	s.d. 5 MB
Seni rupa dalam segala bentuk seperti seni lukis, gambar, seni ukir, seni kaligrafi, seni pahat, seni patung, kolase, dan seni terapan;	Foto/gambar	jpg	s.d. 1 MB
Arsitektur	Foto/gambar	jpg/pdf	s.d. 1 MB
Peta	Foto/Gambar/program	jpg/pdf	s.d. 5 MB
Seni batik	Foto/Gambar	jpg	s.d. 1 MB
Fotografi	Foto/Gambar	jpg	s.d. 1 MB
Sinematografi	Video/rekaman	mp4	s.d. 5 MB
Terjemahan, tafsir, saduran, bunga rampai, dan karya lain dari hasil pengalih wujudan.	Dokumen	pdf	s.d. 5 MB
Database	Meta data, kompilasiciptaan	pdf	s.d. 5 MB
Rekaman suara dan/atau gambar atas suatu pertunjukan	Video/rekaman	mp4	s.d. 5 MB
Rekaman Suara atau rekaman bunyi yang dihasilkan oleh Produser rekaman	Video/rekaman	mp4	s.d. 5 MB
Karya Siaran yang dihasilkan oleh Lembaga Penyiaran	Video/rekaman	mp4	s.d. 5 MB

Keterangan

- ▮ Terhadap contoh ciptaan yang filenya lebih dari 5 MB wajib dikirimkan ke Direktorat Hak Cipta, Desain Industri, DTLST dan Rahasia Dagang, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan HAM R.I Jalan Rasuna Said Kav.8-9 Jakarta Selatan dengan menyebutkan nomor agenda permohonan yang didapat ketika mengajukan permohonan pendaftaran secara elektronik.
- ▮ File contoh ciptaan dikirimkan dalam jangka waktu tiga hari sejak tanggal pengajuan permohonan pendaftaran ciptaan secara elektronik. Apabila file contoh ciptaan yang

melebihi dari 5MB tersebut tidak dikirimkan maka dianggap permohonan pendaftaran ciptaan tersebut ditarik kembali.

SURAT KUASA

Yang bertandatangan di bawah ini, selanjutnya disebut sebagai Pihak Pertama selaku pemilik Ciptaan berupa Program Komputer dengan judul ciptaan “*Program Navigasi Autonomus Car*” :

1. N a m a : Dr. Florentinus Budi Setiawan,
ST. MT Kewarganegaraan : Indonesia
Alamat : Jl. Sinar Pelangi 491, Perum Sinar Waluyo. RT06
RW01 Kelurahan Kedungmundu, Kecamatan
Tembalang, Semarang, Jawa Tengah.
Kodepos : 50273

2. N a m a : Rosita Herawati, ST,
MIT Kewarganegaraan : Indonesia
Alamat :

, Semarang, Jawa Tengah

Memberikan kuasa kepada Pihak Kedua:

Nama : Dr. Berta Berti Retnawati, MSi
Kewarganegaraan : Indonesia
Jabatan : Kepala LPPM Universitas Katolik Soegijapranata
Alamat : Jl. Pawiyatan Luhur IV/1, Bendan Dhuwur

untuk mendaftarkan HKI dengan judul ciptaan “*Program Navigasi Autonomus Car*” secara online. Demikian surat pernyataan ini dibuat untuk dimanfaatkan sebagaimana mestinya.

Pihak Kedua,
Kepala LPPM
Universitas
Katolik
Soegijapranata

Semarang, 17 November
2021 Pihak Pertama,

Dr. Berta Berti
Retnawati, MSi,
NPP. 058.1.1998.219

Dr.Florentinus
Budi S, ST, MT
NPP.
058.1.1994.150

Rosita Herawati, ST,
MIT

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini, pemegang hak cipta

N a m a : LPPM Universitas Katolik

Soegijapranata Kewarganegaraan :

Indonesia

Alamat : Jl. Pawiyatan Luhur IV/1.

BendanDhuwur Semarang

50234

Dengan ini menyatakan bahwa:

1. Karya Cipta yang kami mohonkan:

Berupa : Program Komputer.

Berjudul : *“Program Navigasi Autonomuos Car”*

- ▮ Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
- ▮ Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
- ▮ Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
- ▮ Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
- ▮ Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
- ▮ Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:

- a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
- b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta,

Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.

- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam perkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Semarang, 17 November
2021

Materai 6.000, -

(Dr. Berta Bekt
Retnawati, MSi)

Program komputer navigasi autonomous car

```
import RPi.GPIO as GPIO # importing GPIO
library import time # importing time library for
delay

#Roda Belakang
in1 = 27      # Set pin 27 as output
in2 = 22      # Set pin 22 as output
in3 = 23      # Set pin 23 as output
in4 = 24      # Set pin 24 as output
ena1 = 17     # Set pin 17 as output
enb1 = 25     # Set pin 25 as output

#Roda Depan
in6 = 13      # Set pin 13 as output
in7 = 26      # Set pin 26 as output
in8 = 16      # Set pin 16 as output
in9 = 20      # Set pin 20 as output
ena2 = 6      # Set pin 6 as output
enb2 = 21     # Set pin 21 as output

#Void Setup
GPIO.setmode(GPIO.BCM) # Enable BCM as GPIO

GPIO.setup(in1,GPIO.OUT) # HIGH belakang kiri
maju GPIO.setup(in2,GPIO.OUT) #HIGH belakang kiri
mundur GPIO.setup(in3,GPIO.OUT) # HIGH belakang
kanan maju GPIO.setup(in4,GPIO.OUT) #HIGH
belakang kanan mundur GPIO.setup(ena1,GPIO.OUT)
# Set Enable PWM P1
GPIO.setup(enb1,GPIO.OUT) # Set Enable PWM P2
GPIO.setup(in6,GPIO.OUT) # HIGH depan kiri maju
GPIO.setup(in7,GPIO.OUT) # HIGH depan kiri mundur
GPIO.setup(in8,GPIO.OUT) # HIGH depan kanan maju
GPIO.setup(in9,GPIO.OUT) #HIGH depan kanan mundur
GPIO.setup(ena2,GPIO.OUT) # Set Enable PWM P3
GPIO.setup(enb2,GPIO.OUT) # Set Enable PWM P4

GPIO.output(in1,GPIO.LOW) # HIGH belakang kiri release
GPIO.output(in2,GPIO.LOW)
GPIO.output(in3,GPIO.LOW) # HIGH belakang kanan release
GPIO.output(in4,GPIO.LOW)
GPIO.output(in6,GPIO.LOW) # HIGH depan kiri release
GPIO.output(in7,GPIO.LOW)
GPIO.output(in8,GPIO.LOW) # HIGH depan kanan release
GPIO.output(in9,GPIO.LOW)
```



```
p1 =GPIO.PWM(ena1,1000)
p2 =
GPIO.PWM(enb1,1000) p3
= GPIO.PWM(ena2,1000) p4
= GPIO.PWM(enb2,1000)
```

```
st_1 = 0 # maju & mundur
dt_1 = 54 # maju &
mundur
st_2=0 # geser kanan & geser kiri
dt_2=98 # geser kanan & geser kiri
st_3=0 # putar kanan & putar kiri
dt_3=75 # putar kanan & putar kiri
st_4 = 0 # balik kanan & balik kiri
dt_4=72 # balik kanan & balik kiri
st_5 = 50 # berhenti
dt_5 = 70 # berhenti
```

```
def maju(): # Gerak maju
    print("AGV berjalan
    maju") print(" ")
    GPIO.output(in1,GPIO.HIGH) # HIGH belakang kiri maju
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.HIGH) # HIGH belakang kananmaju
    GPIO.output(in4,GPIO.LOW)
    GPIO.output(in6,GPIO.HIGH) # HIGH depan kiri maju
    GPIO.output(in7,GPIO.LOW)
    GPIO.output(in8,GPIO.HIGH) # HIGH depan kananmaju
    GPIO.output(in9,GPIO.LOW)
```

```
p1.start(st_1)
p2.start(st_1)
p3.start(st_1)
p4.start(st_1)
time.sleep(0)
```

```
p1.ChangeDutyCycle(dt_1)
p2.ChangeDutyCycle(dt_1)
p3.ChangeDutyCycle(dt_1)
p4.ChangeDutyCycle(dt_1)
time.sleep(1)
```

```
def mundur(): #Gerak mundur
    print("AGV berjalan mundur")
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH) # HIGH belakang kiri mundur
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.HIGH) # HIGH belakang kanan mundur
```

```

GPIO.output(in6,GPIO.LOW)
GPIO.output(in7,GPIO.HIGH) # HIGH depan kiri mundur
GPIO.output(in8,GPIO.LOW)
GPIO.output(in9,GPIO.HIGH) # HIGH depan kanan mundur

p1.start(st_1)
p2.start(st_1)
p3.start(st_1)
p4.start(st_1)
time.sleep(0)

p1.ChangeDutyCycle(dt_1)
p2.ChangeDutyCycle(dt_1)
p3.ChangeDutyCycle(dt_1)
p4.ChangeDutyCycle(dt_1)
time.sleep(1)

def geser_kanan(): # Geserkanan
    print("AGV geser ke
    kanan") print(" ")
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH) # HIGH belakang kiri
    mundur GPIO.output(in3,GPIO.HIGH) # HIGH belakang
    kananmaju GPIO.output(in4,GPIO.LOW)
    GPIO.output(in6,GPIO.HIGH) # HIGH depan kiri maju
    GPIO.output(in7,GPIO.LOW)
    GPIO.output(in8,GPIO.LOW)
    GPIO.output(in9,GPIO.HIGH) # HIGH depan kanan mundur

    p1.start(st_2)
    p2.start(st_2)
    p3.start(st_2)
    p4.start(st_2)
    time.sleep(0)

    p1.ChangeDutyCycle(dt_2)
    p2.ChangeDutyCycle(dt_2)
    p3.ChangeDutyCycle(dt_2)
    p4.ChangeDutyCycle(dt_2)
    time.sleep(1)

def geser_kiri(): # Geser
    kiri print("AGV geser ke
    kiri") print(" ")
    GPIO.output(in1,GPIO.HIGH) # HIGH belakang kiri maju
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.HIGH) # HIGH belakang kanan mundur

```

```

GPIO.output(in6,GPIO.LOW)
GPIO.output(in7,GPIO.HIGH) #HIGHdepankirimundur
GPIO.output(in8,GPIO.HIGH) # HIGH depan kanan maju
GPIO.output(in9,GPIO.LOW)

p1.start(st_2)
p2.start(st_2)
p3.start(st_2)
p4.start(st_2)
time.sleep(0)

p1.ChangeDutyCycle(dt_2)
p2.ChangeDutyCycle(dt_2)
p3.ChangeDutyCycle(dt_2)
p4.ChangeDutyCycle(dt_2)
time.sleep(1)

def putar_kanan(): #Putar kanan
    print("AGV putar ke
    kanan") print(" ")
    GPIO.output(in1,GPIO.HIGH) # HIGH belakang kiri maju
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.HIGH) # HIGH belakang kanan mundur
    GPIO.output(in6,GPIO.HIGH) # HIGH depan kiri maju
    GPIO.output(in7,GPIO.LOW)
    GPIO.output(in8,GPIO.LOW)
    GPIO.output(in9,GPIO.HIGH) # HIGH depan kanan mundur

    p1.start(st_3)
    p2.start(st_3)
    p3.start(st_3)
    p4.start(st_3)
    time.sleep(0)

    p1.ChangeDutyCycle(dt_3)
    p2.ChangeDutyCycle(dt_3)
    p3.ChangeDutyCycle(dt_3)
    p4.ChangeDutyCycle(dt_3)
    time.sleep(1)

def putar_kiri(): # Putar kiri
    print("AGV putar ke kiri")
    print(" ")
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH) #HIGHbelakangkirimundur
    GPIO.output(in3,GPIO.HIGH) # HIGH belakang kanan Maju
    GPIO.output(in4,GPIO.LOW)

```

```
GPIO.output(in6,GPIO.LOW)
GPIO.output(in7,GPIO.HIGH) #HIGH depan kiri mundur
GPIO.output(in8,GPIO.HIGH) # HIGH depan kanan maju
GPIO.output(in9,GPIO.LOW)
```

```
p1.start(st_3)
p2.start(st_3)
p3.start(st_3)
p4.start(st_3)
time.sleep(0)
```

```
p1.ChangeDutyCycle(dt_3)
p2.ChangeDutyCycle(dt_3)
p3.ChangeDutyCycle(dt_3)
p4.ChangeDutyCycle(dt_3)
time.sleep(1)
```

```
def balik_kanan(): # Berbalik kanan
```

```
    print("AGV berbalik ke
    kanan") print(" ")
    GPIO.output(in1,GPIO.HIGH) # HIGH belakang kiri maju
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.HIGH) # HIGH belakang kanan mundur
    GPIO.output(in6,GPIO.HIGH) # HIGH depan kiri maju
    GPIO.output(in7,GPIO.LOW)
    GPIO.output(in8,GPIO.LOW)
    GPIO.output(in9,GPIO.HIGH) # HIGH belakang kiri mundur
```

```
p1.start(st_4)
p2.start(st_4)
p3.start(st_4)
p4.start(st_4)
time.sleep(0)
```

```
p1.ChangeDutyCycle(dt_4)
p2.ChangeDutyCycle(dt_4)
p3.ChangeDutyCycle(dt_4)
p4.ChangeDutyCycle(dt_4)
time.sleep(2)
```

```
def balik_kiri(): # Berbalik kiri
```

```
    print("AGV berbalik ke kiri")
    print(" ")
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH) # HIGH belakang kiri
    mundur GPIO.output(in3,GPIO.HIGH) # HIGH belakang
    kanan maju GPIO.output(in4,GPIO.LOW)
```



```
GPIO.output(in6,GPIO.LOW)
```

```
GPIO.output(in7,GPIO.HIGH) #HIGHdepankirimundur
```

```
GPIO.output(in8,GPIO.HIGH) # HIGH depan kananmaju
```

```

GPIO.output(in9,GPIO.LOW)

p1.start(st_4)
p2.start(st_4)
p3.start(st_4)
p4.start(st_4)
time.sleep(0)

p1.ChangeDutyCycle(dt_4)
p2.ChangeDutyCycle(dt_4)
p3.ChangeDutyCycle(dt_4)
p4.ChangeDutyCycle(dt_4)
time.sleep(2)

defberhenti():#Berhenti
    print("AGV
berhenti") print(" ")
    GPIO.output(in1,GPIO.LOW) # HIGH belakang kiri release
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.LOW) # HIGH belakang kanan release
    GPIO.output(in4,GPIO.LOW)
    GPIO.output(in6,GPIO.LOW) # HIGH depan kiri release
    GPIO.output(in7,GPIO.LOW)
    GPIO.output(in8,GPIO.LOW) # HIGH depan kanan release
    GPIO.output(in9,GPIO,LOW)

    p1.start(st_5)
    p2.start(st_5)
    p3.start(st_5)
    p4.start(st_5)
    time.sleep(1)

    p1.ChangeDutyCycle(dt_5)
    p2.ChangeDutyCycle(dt_5)
    p3.ChangeDutyCycle(dt_5)
    p4.ChangeDutyCycle(dt_5)
    time.sleep(1)
#####
def scan_pola():
    print("Sampai tujuan, Cek Camera. . . !\nTekan s untuk lanjut Capture pola. . . !")
    print("\n")

defpola_PA1():#NavigasiPolaPkeA1
    balik_kiri()
    maju()
    maju()
    geser_kanan()
    balik_kanan()

```

berhenti()

defpola_PB1():#NavigasiPolaPkeB1

 balik_kanan()

 maju()

 maju()

 geser_kiri()

 balik_kiri(

)

 berhenti()

defpola_PC1():# Navigasi Pola P ke C1

 balik_kiri()

 maju()

 maju()

 geser_kanan()

 maju()

 maju()

 balik_kanan()

 berhenti()

defpola_PD1():#NavigasiPolaPkeD1

 balik_kanan()

 maju()

 maju()

 geser_kiri()

 maju()

 maju()

 balik_kiri()

 berhenti()

defpola_PA2():#NavigasiPolaPkeA2

 balik_kanan()

 maju()

 maju()

 geser_kiri()

 maju()

 maju()

 geser_kanan()

 putar_kanan()

 maju()

 geser_kanan()

 putar_kanan()

 maju()

 berhenti()

defpola_PB2():#NavigasiPolaPkeB2

 balik_kiri()

maju()

```
maju()
geser_kanan()
maju()
maju()
geser_kiri()
putar_kiri(
) maju()
geser_kiri()
putar_kiri(
) maju()
berhenti()
```

```
def pola_PC2(): # Navigasi Pola P ke C2
```

```
    balik_kanan()
    maju()
    maju()
    geser_kiri()
    maju()
    maju()
    geser_kanan()
    putar_kanan()
    maju()
    putar_kanan()
    berhenti()
```

```
def pola_PD2(): # Navigasi Pola P ke D2
```

```
    balik_kiri()
    maju()
    maju()
    geser_kanan()
    maju()
    maju()
    geser_kiri()
    putar_kiri(
) maju()
    putar_kiri(
)
    berhenti()
```

```
#####
```

```
def pola_AB1(): # Navigasi Pola A ke
```

```
    B1 geser_kanan()
    putar_kanan()
    maju()
    putar_kiri()
    berhenti()
```

```
def pola_AC1(): # Navigasi Pola A ke
```

```
    C1 putar_kiri()
```

geser_kiri()

```
putar_kiri(  
) maju()  
balik_kiri(  
)  
berhenti()
```

```
defpola_AD1():#NavigasiPolaAke  
D1 geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
balik_kanan()  
berhenti()
```

```
defpola_AP1():#NavigasiPolaAkeP1  
geser_kanan()  
maju()  
maju()  
berhenti()
```

```
defpola_AB2():#NavigasiPolaAke  
B2 putar_kiri()  
geser_kiri()  
putar_kiri(  
) maju()  
geser_kiri()  
putar_kiri(  
) maju()  
geser_kiri()  
putar_kiri(  
) maju()  
berhenti()
```

```
defpola_AC2():#NavigasiPolaAke  
C2 geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
putar_kanan()  
berhenti()
```

```
def pola_AD2(): # Navigasi Pola A ke D2
```



```
    putar_kiri(  
    )  
    geser_kiri()  
    putar_kiri(  
    ) maju()  
    geser_kiri()  
    putar_kiri(  
    ) maju()  
    putar_kiri(  
    )  
    berhenti()
```

```
defpola_AP2():#NavigasiPolaAkeP2
```

```
    putar_kiri()  
    geser_kiri()  
    putar_kiri(  
    ) maju()  
    geser_kiri()  
    putar_kiri(  
    ) maju()  
    geser_kiri()  
    putar_kiri(  
    ) maju()  
    geser_kiri()  
    maju()  
    maju()  
    berhenti()
```

```
#####
```

```
defpola_BC1(): # Navigasi Pola B ke C1
```

```
    geser_kanan()  
    putar_kiri(  
    ) maju()  
    geser_kiri()  
    putar_kiri(  
    ) maju()  
    balik_kiri(  
    )  
    berhenti()
```

```
defpola_BD1():#NavigasiPolaBke
```

```
    D1 putar_kanan()  
    geser_kanan()  
    putar_kanan()  
    maju()  
    balik_kanan()  
    berhenti()
```

```
defpola_BA1():#NavigasiPolaBke
```

A1 geser_kiri()
putar_kiri(
) maju()

```
putar_kanan()  
berhenti()
```

```
defpola_BP1():#NavigasiPolaBkeP1  
geser_kiri()  
maju()  
maju()  
berhenti()
```

```
defpola_BC2():#NavigasiPolaBkeC2  
putar_kanan()  
geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
putar_kanan()  
berhenti()
```

```
defpola_BD2():#NavigasiPolaBke  
D2 geser_kiri()  
putar_kiri(  
 ) maju()  
geser_kiri()  
putar_kiri(  
 ) maju()  
geser_kiri()  
putar_kiri(  
 ) maju()  
putar_kiri(  
 )  
berhenti()
```

```
defpola_BA2():#NavigasiPolaBke  
A2 putar_kanan()  
geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
berhenti()
```

```
defpola_BP2():#NavigasiPolaBkeP2
```

putar_kanan()

```
geser_kanan()
putar_kanan()
maju()
geser_kanan()
putar_kanan()
maju()
geser_kanan()
putar_kanan()
maju()
geser_kanan()
maju()
maju()
berhenti()
#####
def pola_CD1(): # Navigasi Pola C ke D1
    geser_kanan()
    putar_kanan()
    maju()
    putar_kiri()
    berhenti()

defpola_CA1():#NavigasiPolaCke
    A1 maju()
    maju()
    berhenti()

def pola_CB1(): # Navigasi Pola C ke B1
    maju()
    maju()
    geser_kanan()
    putar_kanan()
    maju()
    putar_kiri()
    berhenti()

defpola_CP1(): # Navigasi Pola C ke P1
    maju()
    maju()
    geser_kanan()
    maju()
    maju()
    berhenti()

defpola_CD2(): # Navigasi Pola C ke D2
    maju()
    maju()
    geser_kanan()
    putar_kanan()
```

```
maju()
geser_kanan()
putar_kanan()
maju()
balik_kanan()
berhenti()
```

```
defpola_CA2():#NavigasiPolaCke
A2 geser_kanan()
putar_kanan()
maju()
geser_kiri()
putar_kiri()
maju()
geser_kiri()
putar_kiri()
maju()
putar_kanan()
berhenti()
```

```
defpola_CB2():#NavigasiPolaCkeB2
geser_kanan()
putar_kanan()
maju()
geser_kiri()
putar_kiri()
maju()
berhenti()
```

```
defpola_CP2():#NavigasiPolaCkeP2
geser_kanan()
putar_kanan()
maju()
geser_kiri()
putar_kiri()
maju()
geser_kiri()
maju()
maju()
berhenti()
```

```
#####
```

```
defpola_DA1():#NavigasiPolaDke
A1 maju()
maju()
geser_kiri()
putar_kiri()
maju()
putar_kanan()
```

berhenti()

defpola_DB1():#NavigasiPolaDke

B1 maju()

maju()

berhenti()

defpola_DC1():#NavigasiPolaDkeC1

geser_kiri()

putar_kiri()

maju()

putar_kanan()

berhenti()

defpola_DP1():#NavigasiPolaDkeP1

maju()

maju()

geser_kiri()

maju()

maju()

berhenti()

defpola_DA2():#NavigasiPolaDke

A1 geser_kiri()

putar_kiri()

maju()

geser_kanan()

putar_kanan()

maju()

berhenti()

defpola_DB2():#NavigasiPolaDke

B2 geser_kiri()

putar_kiri()

maju()

geser_kanan()

putar_kanan()

maju()

geser_kanan()

putar_kanan()

maju()

putar_kiri()

berhenti()

defpola_DC2():#NavigasiPolaDkeC2

maju()

maju()

geser_kiri()

```
putar_kiri(  
) maju()  
geser_kiri()  
putar_kiri(  
) maju()  
balik_kiri(  
)  
berhenti()
```

```
defpola_DP2():#NavigasiPolaDkeP2
```

```
geser_kiri()  
putar_kiri()  
maju()  
geser_kanan()  
putar_kanan()  
maju()  
geser_kanan()  
maju()  
maju()
```

```
#####
```


Dokumen pendukung luaran Tambahan #1

Luaran dijanjikan: Artikel pada Conference/Seminar Internasional di Pengindeks Bereputasi

Target: Terbit dalam Prosiding

Dicapai: Submitted

Dokumen wajib diunggah:

1. Naskah artikel
2. Bukti submit

Dokumen sudah diunggah:

1. Naskah artikel
2. Bukti submit

Dokumen belum diunggah:

-

Peran penulis: first author

Nama Konferensi/Seminar: The 8th International Conference on Engineering Technology and Industrial Application

Lembaga penyelenggara: Universitas Muhamadiyah Surakarta

Tempat penyelenggara: Surakarta

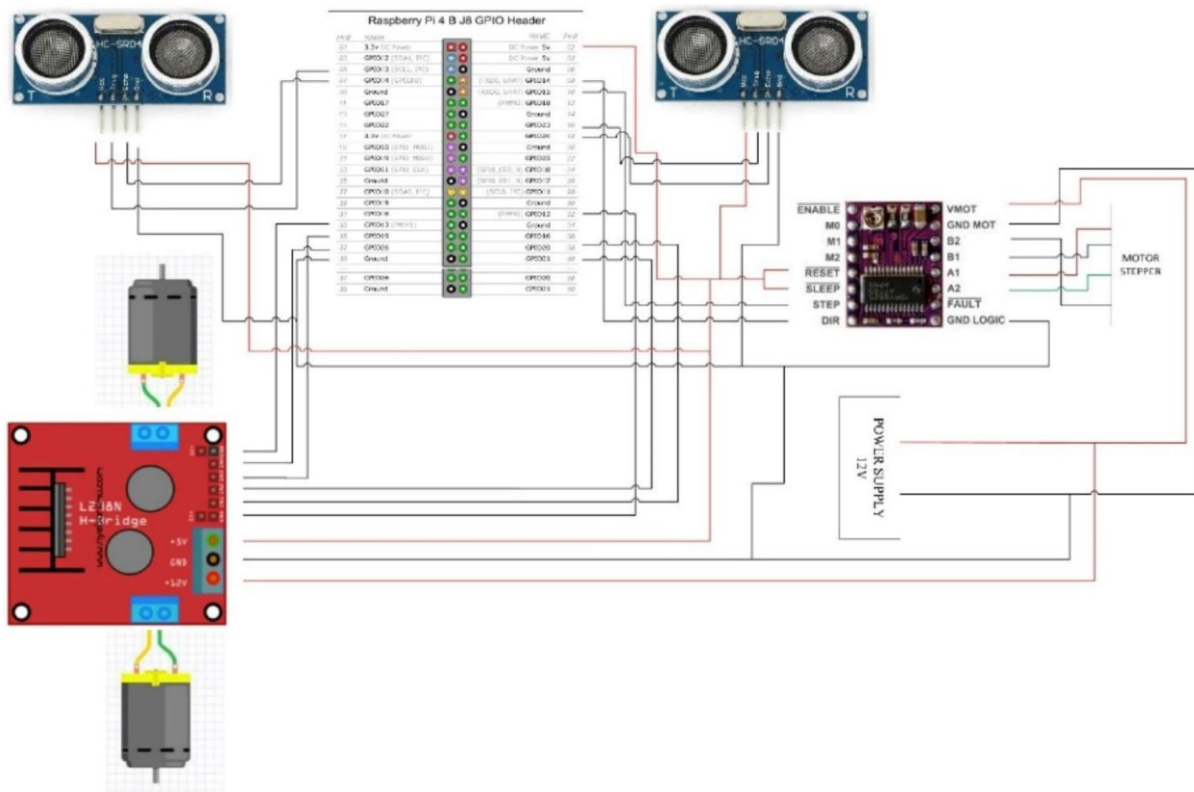
Tgl penyelenggaraan mulai: 15 Desember 2021 | Tgl selesai: 16 Desember 2021

Lembaga pengindeks: Scopus

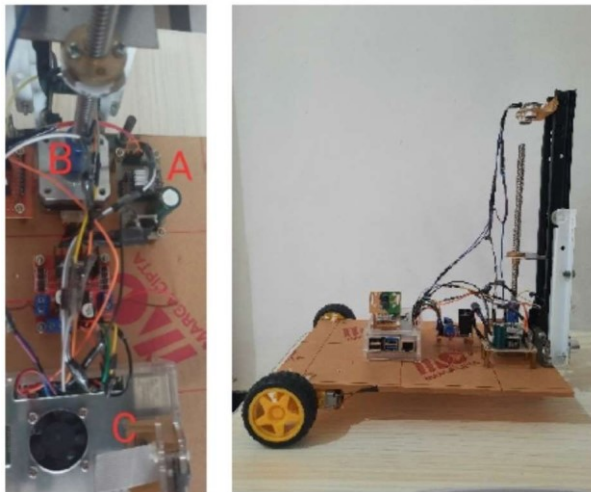
URL website: <https://icetia.ums.ac.id/2021>

Judul artikel: Autonomus Car Prototype Navigation Using Simultaneous Localization And Mapping System

1. Prototipe AGV dari PT. Djarum Tbk. untuk dikembangkan



2. Prototipe Forklift untuk Industri yang dikembangkan oleh PT Djarum Tbk.



3. Replikasi Prototipe Autonomous Car dari PT. Darum Tbk



LISTING PROGRAM NAVIGASI

```
import cv2          # importing OpenCV library
import pytesseract # importing OCR library
from picamera.array import PiRGBArray
from picamera import PiCamera
from motor_dc import *

camera = PiCamera()
camera.resolution = (640, 640)
camera.framerate = 30
rawCapture = PiRGBArray(camera, size=(640, 640))

for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    cv2.imshow("Camera Capture Pola", image)
    key = cv2.waitKey(1) & 0xFF
    rawCapture.truncate(0)

    if key == ord("s"):
        conf = r'--oem 3 --psm 6 outputbase digits'
        text = pytesseract.image_to_string(image, config=conf)
        print("Capture pola! Hasil =", text)

    if text == 'P': # Deteksi Pola P
        time.sleep(1)
        print("AGV parkir di pola", text)
        print(" ")
        time.sleep(1)
```

```
tujuan = input ("Pergi ke mana ? ")  
print(" ")
```

```
if tujuan == "A1": # Tujuan A1  
    pola_PA1() # Navigasi ke Pola A1  
    scan_pola()
```

```
if tujuan == "B1": # Tujuan B1  
    pola_PB1() # Navigasi ke Pola B1  
    scan_pola()
```

```
if tujuan == "C1": # Tujuan C1  
    pola_PC1() # Navigasi ke Pola C1  
    scan_pola()
```

```
if tujuan == "D1": # Tujuan D1  
    pola_PD1() # Navigasi ke Pola D1  
    scan_pola()
```

```
if tujuan == "A2": # Tujuan A2  
    pola_PA2() # Navigasi ke Pola A2  
    scan_pola()
```

```
if tujuan == "B2": # Tujuan B2  
    pola_PB2() # Navigasi ke Pola B2  
    scan_pola()
```

```
if tujuan == "C2": # Tujuan C2  
    pola_PC2() # Navigasi ke Pola C2  
    scan_pola()
```

```
if tujuan == "D2": # Tujuan D2  
    pola_PD2() # Navigasi ke Pola D2  
    scan_pola()
```

```
if text == 'A': # Deteksi Pola A
```

```
time.sleep(1)
print("AGV parkir di pola", text)
print(" ")
time.sleep(1)
tujuan = input ("Pergi ke mana ? ")
print(" ")
```

```
if tujuan == "B1": # Tujuan B1
    pola_AB1() # Navigasi ke Pola B1
    scan_pola()
```

```
if tujuan == "C1": # Tujuan C1
    pola_AC1() # Navigasi ke Pola C1
    scan_pola()
```

```
if tujuan == "D1": # Tujuan D1
    pola_AD1() # Navigasi ke Pola D1
    scan_pola()
```

```
if tujuan == "P1": # Tujuan P1
    pola_AP1() # Navigasi ke Pola P1
    scan_pola()
```

```
if tujuan == "B2": # Tujuan B2
    pola_AB2() # Navigasi ke Pola B2
    scan_pola()
```

```
if tujuan == "C2": # Tujuan C2
    pola_AC2() # Navigasi ke Pola C2
    scan_pola()
```

```
if tujuan == "D2": # Tujuan D2
    pola_AD2() # Navigasi ke Pola D2
    scan_pola()
```

```
if tujuan == "P2": # Tujuan P2
```

```
pola_AP2() # Navigasi ke Pola P2  
scan_pola()
```

```
if text == 'B': # Deteksi Pola B
```

```
time.sleep(1)
```

```
print("AGV di pola", text)
```

```
print(" ")
```

```
time.sleep(1)
```

```
tujuan = input ("Pergi ke mana ? ")
```

```
print(" ")
```

```
if tujuan == "C1": # Tujuan C1
```

```
    pola_BC1() # Navigasi ke Pola B1
```

```
    scan_pola()
```

```
if tujuan == "D1": # Tujuan D1
```

```
    pola_BD1() # Navigasi ke Pola D1
```

```
    scan_pola()
```

```
if tujuan == "A1": # Tujuan A1
```

```
    pola_BA1() # Navigasi ke Pola P1
```

```
    scan_pola()
```

```
if tujuan == "P1": # Tujuan P1
```

```
    pola_BP1() # Navigasi ke Pola P1
```

```
    scan_pola()
```

```
if tujuan == "C2": # Tujuan C2
```

```
    pola_BC2() # Navigasi ke Pola C2
```

```
    scan_pola()
```

```
if tujuan == "D2": # Tujuan D2
```

```
    pola_BD2() # Navigasi ke Pola D2
```

```
    scan_pola()
```

```
if tujuan == "A2": # Tujuan A2
    pola_BA2() # Navigasi ke Pola A2
    scan_pola()
if tujuan == "P2": # Tujuan P2
    pola_BP2() # Navigasi ke Pola P2
    scan_pola()
```

```
if text == 'C': # Deteksi Pola C
    time.sleep(1)
    print("AGV di pola", text)
    print(" ")
    time.sleep(1)
    tujuan = input ("Pergi ke mana ? ")
    print(" ")
```

```
if tujuan == "D1": # Tujuan D1
    pola_CD1() # Navigasi ke Pola D1
    scan_pola()
if tujuan == "A1": # Tujuan A1
    pola_CA1() # Navigasi ke Pola A1
    scan_pola()
if tujuan == "B1": # Tujuan B1
    pola_CB1() # Navigasi ke Pola B1
    scan_pola()
if tujuan == "P1": # Tujuan P1
    pola_CP1() # Navigasi ke Pola P1
    scan_pola()
if tujuan == "D2": # Tujuan D2
    pola_CD2() # Navigasi ke Pola D2
```



```
    scan_pola()
if tujuan == "A2": # Tujuan A2
    pola_CA2() # Navigasi ke Pola A2
    scan_pola()
if tujuan == "B2": # Tujuan B2
    pola_CB2() # Navigasi ke Pola B2
    scan_pola()
if tujuan == "P2": # Tujuan P2
    pola_CP2() # Navigasi ke Pola P2
    scan_pola()

if text == 'D': # Deteksi Pola D
    time.sleep(1)
    print("AGV di pola", text)
    print(" ")
    time.sleep(1)
    tujuan = input ("Pergi ke mana ? ")
    print(" ")

if tujuan == "A1": # Tujuan A1
    pola_DA1() # Navigasi ke Pola A1
    scan_pola()
if tujuan == "B1": # Tujuan B1
    pola_DB1() # Navigasi ke Pola B1
    scan_pola()
if tujuan == "C1": # Tujuan C1
    pola_DC1() # Navigasi ke Pola C1
    scan_pola()
if tujuan == "P1": # Tujuan P1
```

```
pola_DP1() # Navigasi ke Pola P1
scan_pola()

if tujuan == "A2": # Tujuan A2
    pola_DA2() # Navigasi ke Pola A2
    scan_pola()

if tujuan == "B2": # Tujuan B2
    pola_DB2() # Navigasi ke Pola B2
    scan_pola()

if tujuan == "C2": # Tujuan C2
    pola_DC2() # Navigasi ke Pola C2
    scan_pola()

if tujuan == "P2": # Tujuan P2s
    pola_DP2() # Navigasi ke Pola P2
    scan_pola()

cv2.imshow("Camera Capture Pola", image)
cv2.waitKey(0)

GPIO.cleanup()
PiCamera.release()
cv2.destroyAllWindows()
```