

# Perancangan Robot Pengikut berbasis Visual dan Kontrol Logika Samar

**ALI SADIYOKO, AVICENNA FAHRUNASA, TRIANA MUGIA RAHAYU**

Jurusan Teknik Elektro, Universitas Katolik Parahyangan, Indonesia  
Email: [alfa51@unpar.ac.id](mailto:alfa51@unpar.ac.id)

*Received* 24 Agustus 2023 | *Revised* 27 September 2023 | *Accepted* 20 Oktober 2023

## ABSTRAK

*Pada penelitian ini dilakukan replikasi teknologi leader-follower (konvoi) pada sebuah purwarupa mobile robot. Purwarupa yang dibuat terdiri dari komponen perangkat keras, program pengolahan citra, dan sistem kontrol logika samar untuk sistem leader dan follower. Robot pengikut yang dirancang pada penelitian ini menggunakan mekanisme kemudi Ackermann, sehingga meniru perilaku pengemudi (steering) sebuah mobil roda 4 pada umumnya. Percobaan akan direkam dengan kamera dan dianalisis menggunakan program Python untuk mendapatkan nilai koordinat robot leader dan follower. Hasil percobaan menunjukkan bahwa purwarupa robot follower berhasil mengikuti robot di depannya dengan rata-rata jarak 11,79 cm pada jalur lurus dan 19,82 cm pada jalur berbelok.*

**Kata kunci:** robot bergerak, robot pemimpin-pengikut, kemudi Ackermann, pengolahan citra, logika samar

## ABSTRACT

*In this research, the leader-follower (convoy) technology is replicated in mobile robot aimed to show the implementation of the technology to follow the target in front of it. This article described the designs and performance of a follower robot designs. The design contained hardware components, image processing program, and a fuzzy logic control algorithm for the leader and follower system. The follower robot designed in this research utilizes the Ackermann steering mechanism, which imitates the steering behavior of a typical 4-wheeled car. The experiment will be recorded with a camera and analyzed using Python program to obtain the coordinates of the leader and follower robots. The experimental results indicate that the prototype of the follower robot successfully follows the robot in front of it with an average distance of 11,79 cm on straight paths and 19,82 cm on curved paths.*

**Keywords:** mobile robot, leader-follower robot, Ackermann steering, image processing, fuzzy logic control

## 1. PENDAHULUAN

Saat ini beberapa produsen mobil menambahkan teknologi *car-following* pada beberapa produknya. Teknologi tersebut merupakan salah satu bentuk dari *Advanced Driver Assistance System* (ADAS) yang mulai banyak diimplementasikan pada mobil produksi terbaru. ADAS merupakan sistem pendukung canggih dalam berkendara yang digunakan pada bidang otomotif untuk menangkap situasi sekitar kendaraan menggunakan berbagai macam sensor. ADAS digunakan untuk memberikan sinyal bahaya kepada pengemudi, memberikan kenyamanan dan keamanan bagi pengemudi dan pengguna jalan yang lain. ADAS menyediakan segala fungsi yang berhubungan dengan pengaturan kemudi dan rem kendaraan serta sistem pendukung pengemudi lainnya termasuk *Adaptive Cruise Control* (ACC) dan *Lane Keeping Assist System* (LKAS). ACC digunakan untuk mengontrol kecepatan kendaraan sedangkan LKAS digunakan untuk dapat mempertahankan posisi kendaraan agar tetap berada di jalurnya. Pada level kendaraan otonom, ADAS ditempatkan pada level 2 yaitu *autonomous driving* (Bengler, dkk, 2014).

Kedua teknologi tersebut menggunakan kamera yang digunakan untuk mendeteksi target serta aktuator yang bisa digunakan untuk menggerakkan kendaraan menuju target. Namun, teknologi tersebut tidak untuk digunakan pada situasi (iring-iringan) konvoi atau mengikuti kendaraan di depannya. Jika akan digunakan pada situasi konvoi, maka sistem kamera pada kendaraan tersebut perlu ditambahkan sistem kontrol yang dengan algoritma *leader-follower* (L-F). Dengan algoritma L-F ini, jarak antar kendaraan dapat terjaga terus menerus sehingga situasi konvoi tersebut dapat dicapai. Saat ini, teknologi untuk dapat melakukan konvoi secara otomatis sudah mulai diterapkan di beberapa merek truk Eropa, dengan harapan bahwa teknologi ini mampu menurunkan biaya transportasi, mengurangi konsumsi bahan bakar serta menurunkan resiko terjadinya kecelakaan di jalan raya (Aarts & Feddes, 2016) (Bergenheim, dkk, 2012) (Janssen, dkk, 2015).

Namun, jika kondisi konvoi terjadi pada waktu yang lama, hal ini akan menurunkan tingkat kewaspadaan pengemudi, yang pada akhirnya akan meningkatkan potensi kecelakaan (Axelsson, 2017). Hal ini terkadang sulit dilakukan manusia. Respon pengemudi tidak dapat selalu konsisten, terutama jika kondisi fisik pengemudi tersebut tidak fit, misalnya karena sakit, lelah, mengantuk atau tidak fokus. Sistem yang dirancang pada penelitian ini diharapkan dapat mengurangi resiko terjadinya kecelakaan akibat adanya kondisi pengemudi yang tidak fit pada saat dalam keadaan konvoi. Mathesson (Mathesson, 2019) dalam penelitiannya menyebutkan bahwa manusia membutuhkan waktu respon 1,5 detik untuk bereaksi setelah mendeteksi sebuah bahaya yang ada pada saat mengemudi. Sedangkan pada penelitian lain (Johansson & Schröder, 2016) disebutkan bahwa dengan sistem kemudi konvoi otomatis, kendaraan yang terpasang dengan sistem tersebut mampu bereaksi dengan waktu respon antara 20-60 ms. Penelitian lebih baru (Castritius, dkk, 2020) juga menyatakan bahwa para pengemudi truk di Jerman dapat menerima penggunaan teknologi *automated driving* karena memang dirasakan lebih aman.

Pada penelitian sebelumnya yang dilakukan oleh Dewi (Dewi, dkk, 2018), konsep L-F disimulasikan dengan model robot dua roda dengan prinsip kemudi diferensial (*differential steering*). Pada penelitian ini, kontrol logika samar akan diimplementasikan pada robot nyata dengan sistem kemudi Ackermann. Bentuk fungsi keanggotaan pada penelitian ini akan menggunakan bentuk segitiga, sementara pada penelitian sebelumnya menggunakan bentuk *sigmoid*.

## 2. METODE

### 2.1 *Advanced Driving Assist System (ADAS)*

Rancangan sistem kontrol untuk mengikuti target didasarkan dari teknologi ADAS yang menyediakan teknologi *Advanced Cruise Control (ACC)* dan *Line Keeping Assist System (LKAS)*. Teknologi ACC menggunakan radar untuk mengetahui jarak kendaraan di depannya agar bisa mengubah kecepatannya tergantung halangan yang ada di depannya. Teknologi LKAS melakukan pengolahan citra dengan mendeteksi garis jalan yang ada menggunakan kamera depan kendaraan. Informasi garis jalan yang ada selanjutnya digunakan untuk mengatur pergerakan kemudi agar tetap berada di antara dua garis jalan. ADAS sendiri mulai populer di Amerika Serikat pada awal tahun 2000 sedangkan teknologi ACC muncul ditahun 2000 pada pabrikan Toyota dengan nama *Dynamic Laser Cruise Control* dan teknologi LKAS muncul ditahun 2016 pada produk LEXUS LS dengan nama *Lane Keeping Assist (Eichelberger & McCartt, 2016)*.

### 2.2 Pengolahan Citra

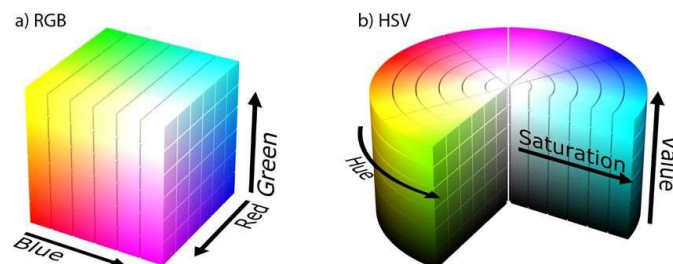
Pada penelitian ini, sebuah kamera dipasang di depan robot untuk menangkap citra robot *leader* di depannya. Terdapat tiga langkah utama dalam pengolahan citra yang dilakukan dalam penelitian ini, yaitu :

- Proses pengambilan citra menggunakan kamera untuk mengakuisisi gambar.
- Proses analisis dan manipulasi citra.
- Proses menghasilkan keluaran berupa hasil olahan citra atau parameter-parameter citra yang telah diolah.

Untuk melakukan pengolahan citra, dalam penelitian ini digunakan *tool* OpenCV berbasis *Python*. OpenCV digunakan karena *tool* ini memiliki banyak kemudahan *syntax* dan *command* untuk pengolahan citra serta memiliki kecepatan pemrosesan yang baik. Citra objek pada penelitian ini menggunakan jenis ruang warna *Hue, Saturation, Value (HSV)*. Hal ini dipilih karena ruang warna HSV memiliki pembagian warna yang lebih baik dibanding ruang warna *Red, Green, Blue (RGB)* (Crneković, dkk, 2013). Ruang warna HSV juga lebih mudah untuk diolah oleh komputer karena pembagian informasi warna yang lebih jelas dibandingkan ruang warna RGB. Pada Tabel 1 berikut, diperlihatkan data nilai jangkuan parameter ruang warna RGB, HSV, dan HSV yang digunakan pada OpenCV. Pada Gambar 1 juga diperlihatkan ruang warna yang menggunakan parameter RGB dan HSV (Popov, dkk, 2018).

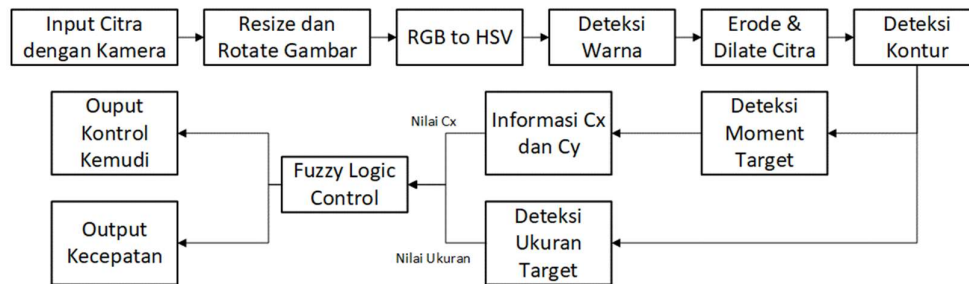
**Tabel 1. Nilai Pada Ruang Warna RGB, HSV, dan HSV OpenCV**

No	Jenis	Nilai			Keterangan
1	RGB	0-255	0-255	0-255	Range RGB
2	HSV	0-360	0-100	0-100	Range HSV
3	HSV OpenCV	0-180	0-255	0-255	Range HSV OpenCV



**Gambar 1. Ruang Warna RGB dan HSV**

Sedangkan untuk proses pengolahan citra, mulai dari pengambilan citra hingga menghasilkan parameter nilai kontrol untuk kemudi dan kecepatan robot pengikut, dapat dilihat pada Gambar 2 di halaman berikutnya.



**Gambar 2. Diagram Blok Proses Pengolahan Citra pada Robot *Follower***

Teknik pengolahan citra yang dilakukan pada penelitian ini antara lain :

- Mengubah ukuran dan posisi *frame* menggunakan *library imutils*.
- Mengubah ruang warna gambar yang umumnya RGB menjadi HSV.
- Mendeteksi dan mengubah warna target yang diinginkan agar berubah menjadi putih dan warna lainnya menjadi hitam. Proses tersebut membutuhkan batas warna bawah dan atas dalam bentuk parameter HSV.
- Menghilangkan anomali kecil yang ada pada hasil deteksi warna.
- Mengisi kekosongan ruang warna pada area deteksi warna yang rusak.
- Mendeteksi kontur berbentuk kotak dilanjutkan dengan mencari momen dari kotak yang ada untuk mendapatkan nilai titik tengah dari target.
- Mendeteksi ukuran dari target kotak dengan mendeteksi jari-jari dari target.

Deteksi titik tengah target dilakukan dengan cara mencari *image moment*. *Image moment* sendiri dapat digunakan untuk menghitung pusat massa objek, area objek, dan lain-lain. Berikut persamaan untuk mencari *centroid* sebuah objek citra yang dikembangkan oleh Mordvintsev dan Abid (**Mordvintsev & Abid, 2014**).

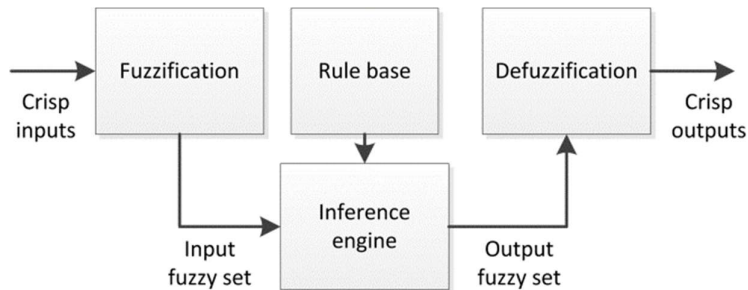
$$C_x = \text{int}(M['m10']/M['m00']) \quad (1)$$

$$C_y = \text{int}(M['m01']/M['m00'])$$

Di mana  $C_x$  dan  $C_y$  adalah nilai *centroid* sebuah objek di sumbu-x dan sumbu-y, serta  $\text{int}(M)$  adalah momen dari sebuah citra yang diperoleh dengan menggunakan algoritma yang dikembangkan oleh Mordvintsev dan Abid. Deteksi ukuran jari-jari target diperoleh dengan menggunakan fungsi *Minimum Enclosing Circle*. Fungsi tersebut digunakan untuk memperoleh jari-jari minimal dari target karena memperhitungkan rotasi dari objek dibanding fungsi *Straight Bounding Rectangle* yang bisa memperoleh jari-jari minimal target tetapi tidak memperhitungkan rotasi dari objek. Karena pada percobaan ini target berbentuk kotak dan tidak selalu berbentuk lurus, maka cara ini menjadi lebih sesuai. *Minimum Enclosing Circle* hanya memberikan keluaran berupa panjang jari-jari dari sebuah target, sehingga lebih sederhana bila dibandingkan *Bounding Rectangle* yang memberikan keluaran nilai panjang dan lebar dari sebuah objek.

### 2.3 Kontrol Logika Samar

Pada penelitian ini, kontrol logika samar pada robot ini diilustrasikan seperti pada Gambar 3 berikut (Panduru, dkk, 2014).



**Gambar 3. Diagram Blok Pengendali dengan Logika Samar**

Dari Gambar 3 di atas, dapat diuraikan tiga proses utama pada kontrol logika samar, yaitu:

- Proses fusifikasi (*fuzzyfication*), adalah proses pengubahan bentuk nilai masukan *crisp* ke bentuk nilai dalam himpunan samar, yang kemudian menjadi masukan bagi *inference engine* (mesin inferensi/penalaran). Nilai masukan dikonversi dari nilai *crisp* menjadi nilai samar menggunakan fungsi keanggotaan yang sesuai.
- Mesin inferensi (*inference engine*), adalah proses pembentukan hubungan (pemetaan) antara masukan ke keluaran dengan menggunakan logika samar. Nilai masukan dan keluaran pada mesin inferensi ini berbentuk himpunan samar. Pemetaan ini memberikan dasar tentang bagaimana sebuah keputusan dapat dibuat, atau pola yang dibentuk. Aturan-aturan samar dalam bentuk "*if-then*" digunakan untuk mengevaluasi sebuah nilai masukan samar dan menggabungkannya dalam tingkat kesimpulan yang tepat. Ada dua metode inferensi yang dikenal, yaitu metode Sugeno dan metode Mamdani, serta variasi turunannya. Sedangkan pada penelitian, ini digunakan metode inferensi Takagi-Sugeno-Kang (TSK). Metode inferensi TSK dipilih karena metode ini mengekspresikan dinamika lokal dari setiap aturan samarnya dengan model sistem linier dan memiliki efisiensi komputasi yang lebih baik dibanding metode Mamdani (Kamali, dkk, 2017). Metode TSK akan dijelaskan melalui Persamaan (2) berikut ini (Li, dkk, 2018):

$$\begin{aligned}
 R_1: & \text{IF } x_1 \text{ is } A_{11} \text{ and } \dots \text{ and } x_m \text{ is } A_{m1} \\
 & \text{THEN } y = f_1(x_1, \dots, x_m) \\
 & \quad \vdots \\
 R_n: & \text{IF } x_1 \text{ is } A_{1n} \text{ and } \dots \text{ and } x_m \text{ is } A_{mn} \\
 & \text{THEN } y = f_n(x_1, \dots, x_m)
 \end{aligned} \tag{2}$$

Di mana pada Persamaan (2),  $R_1, \dots, R_n$  adalah aturan samar,  $x_1, \dots, x_m$  adalah masukan bernilai samar dan  $y$  adalah keluaran bernilai *crisp* dengan fungsi pemetaan  $f_1(\cdot), \dots, f_n(\cdot)$ .

- Proses defusifikasi (*defuzzification*), adalah proses pengubahan bentuk nilai keluaran proses inferensi, yang masih berbentuk himpunan samar, menjadi sebuah nilai bilangan *crisp* (real). Merujuk pada Persamaan (2), setelah aturan dievaluasi, maka bobot atau tingkat kebenaran dari setiap aturan ( $R_i: i \in \{1, \dots, n\}$ ). Bobot ini ( $\alpha_r$ ) mencerminkan seberapa kuat setiap aturan berkontribusi terhadap hasil akhir. Proses defusifikasi pada metode TSK menggunakan metode rata-rata terbobot (*weighted average method*), seperti yang ditunjukkan pada Persamaan (3). Hal inilah membuat

metode TSK lebih efisien dalam proses komputasinya dibanding dengan metode Mamdani.

$$y = \frac{\sum_{r=1}^n \alpha_r f_r(x_1^*, \dots, x_m^*)}{\sum_{r=1}^n \alpha_r} \quad (3)$$

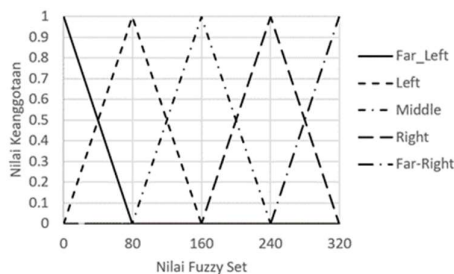
Di mana pada Persamaan (3) ini,  $\alpha_r$  adalah bobot dari masing-masing aturan samar dan  $x_1^*, \dots, x_m^*$  adalah nilai representasi (*defuzzified value*) dari himpunan samar  $A$ .

Untuk fungsi keanggotaan samar untuk masukan posisi dan ukuran objek target digunakan fungsi keanggotaan berbentuk segitiga. Hal ini karena bentuk segitiga merupakan bentuk yang sederhana, sehingga akan mempersingkat waktu komputasi. **(Kosheleva & Kreinovich, 2018) (Sadollah, 2018)** Pada Gambar 4 di bawah, diilustrasikan himpunan keanggotaan masukan untuk posisi dan ukuran objek.

Pada tabel sebelah kiri di Gambar 4, nilai 0 (minimum) dan 320 (maksimum) diperoleh dari ukuran resolusi gambar yang digunakan pada program dan kamera. Resolusi dipilih pada nilai 640x360 agar tidak membebani mikroprosesor saat mengolah data. Pada tabel sebelah kanan, sumbu X merupakan nilai himpunan samar dan sumbu Y merupakan nilai keanggotaan samar. Nilai 30 (maksimum) pada tabel merupakan nilai referensi posisi dari objek, yang menunjukkan bahwa objek (robot pemimpin) berada 30 cm di depan robot pengikut. Ketika nilai referensi bernilai 30, nilai piksel yang diperoleh adalah 45, sehingga perubahan jarak berbanding ukuran target menjadi 1,5 *pixel/cm*.

Himpunan Keanggotaan Masukan Posisi Target

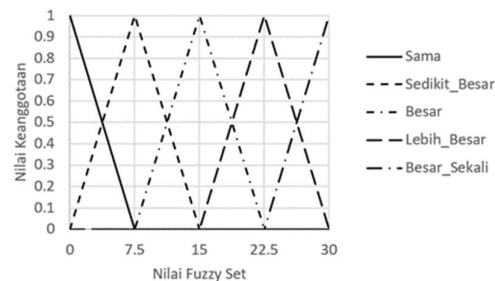
No	Label	Min	Max
1	Far_Left (FL)	0	80
2	Left (L)	0	160
3	Middle (M)	80	240
4	Right (R)	160	320
5	Far_Right (FR)	240	320



(a)

Himpunan Keanggotaan Masukan Ukuran Objek

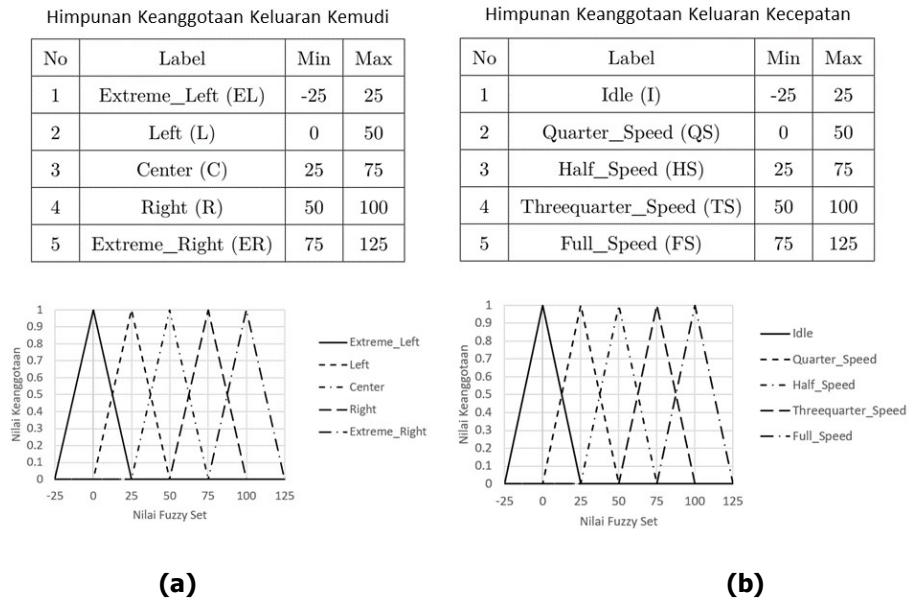
No	Label	Min	Max
1	Sama (S)	0	7.5
2	Sedikit_Besar (SB)	0	15
3	Besar (B)	7.5	22.5
4	Lebih_Besar (LB)	15	30
5	Besar_Sekali (BS)	22.5	30



(b)

**Gambar 4. Himpunan Keanggotaan Masukan (a) Posisi Target dan (b) Ukuran Objek**

Untuk himpunan keanggotaan keluaran posisi kemudi dan kecepatan robot pengikut, dijelaskan dengan menggunakan ilustrasi pada Gambar 5 berikut ini.



**Gambar 5. Himpunan Keanggotaan Keluaran (a) Kemudi dan (b) Kecepatan Robot**

Gambar 5 menjelaskan bahwa nilai minimum dan maksimum untuk kemudi dan kecepatan adalah -25 dan 125. Nilai ini kemudian disesuaikan dengan sudut kemudi yang dapat dicapai oleh konstruksi robot ( $-30^\circ$  hingga  $30^\circ$ ). Berdasarkan himpunan keanggotaan masukan dan keluaran logika samar tersebut, disusun lima peraturan untuk kemudi dan lima peraturan untuk kecepatan robot. Berikut lima peraturan untuk kemudi robot pengikut :

1. Jika Posisi = `Far\_Left` maka Kemudi = `Extreme\_Left`,
2. Jika Posisi = `Left`, maka Kemudi = `Left`,
3. Jika Posisi = `Center`, maka Kemudi = `Center`,
4. Jika Posisi = `Right`, maka Kemudi = `Right`,
5. Jika Posisi = `Far\_Right`, maka Kemudi = `Extreme\_Right`.

Sedangkan lima peraturan untuk kecepatan robot pengikut, adalah sebagai berikut :

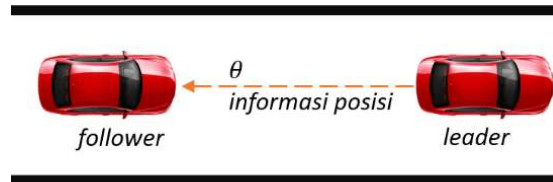
1. Jika Ukuran = `Sama`, maka Kecepatan = `Full\_Speed`,
2. Jika Ukuran = `Sedikit\_Besar`, maka Kecepatan = `Threequarter\_Speed`,
3. Jika Ukuran = `Besar`, maka Kecepatan = `Half\_Speed`,
4. Jika Ukuran = `Lebih\_Besar`, maka Kecepatan = `Quarter\_Speed`,
5. Jika Ukuran = `Besar\_Sekali`, maka Kecepatan = `Idle`.

## 2.4 Kondisi Leader-Follower pada Sistem Multi-robot

Kondisi *leader-follower* dalam penelitian ini didefinisikan sebagai sebuah keadaan dalam sistem multi-robot, di mana sebuah robot menjadi pemimpin (*leader*) dan robot lainnya menjadi pengikutnya (*follower*). Pada sistem ini, robot pemimpin harus memiliki variabel yang dapat dikomunikasikan kepada robot lainnya, misalkan kecepatan atau posisi. Dengan istilah lain, robot pemimpin menjadi acuan/referensi bagi robot lain dalam kelompok itu untuk melakukan sebuah tindakan/tugas. Pada penelitian ini, kondisi *leader-follower* menirukan kondisi konvoi dua mobil, di mana robot pemimpin terletak di depan robot pengikut. Robot pengikut harus mampu mengikuti posisi dan menghitung jarak aman antara dirinya dengan robot pemimpin.



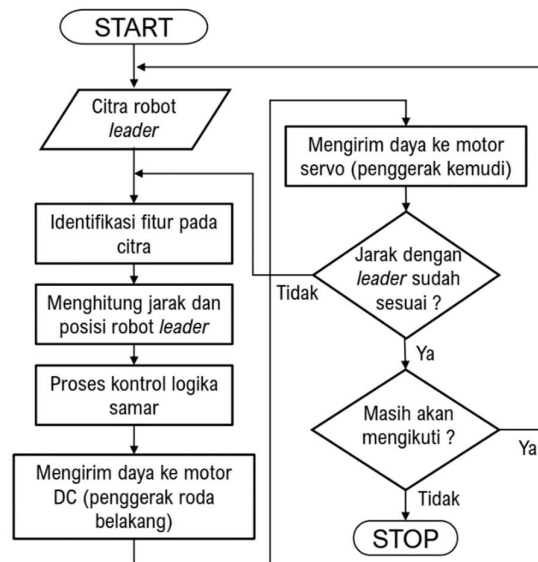
Konsep *leader-follower* terbagi menjadi tiga yaitu, (1) *static leader* di mana pemimpin selalu dalam keadaan diam (*statis*), (2) *dynamic leader* di mana pemimpin dapat bergerak (*dinamis*), dan (3) *virtual leader* di mana pemimpin yang ditetapkan adalah hanya berupa variabel status (*state variabel*) hasil perhitungan perangkat lunak (**Soni & Hu, 2018**). Pada penelitian ini akan digunakan konsep *leader-follower* dengan *dynamic leader*, seperti yang diilustrasikan dengan Gambar 6 di bawah ini.



**Gambar 6. Ilustrasi Kondisi *Leader-Follower***

## 2.5 Teknik Identifikasi Target

Pada penelitian ini, proses identifikasi target di depan robot akan menggunakan kamera, sesuai penelitian yang dilakukan Johansson (**Johansson & Schröder, 2016**) dan Wang (**Wang, dkk, 2019**). Sebuah target objek di depan robot dapat diidentifikasi dengan menggunakan teknik pengolahan sederhana, seperti mendeteksi bentuk tertentu, warna, sudut serta ukuran target. Setelah objek dapat diidentifikasi, maka langkah berikutnya adalah mencari cara untuk mencapai posisi objek tersebut. Langkah-langkah proses identifikasi objek hingga gerakan robot diilustrasikan dengan diagram alir berikut ini:



**Gambar 7. Diagram Alir Proses Identifikasi dan Gerak di Robot Pengikut**

Penjelasan proses pada Gambar 7 adalah sebagai berikut:

1. Tentukan nilai konversi jarak ke lebar objek di citra kamera. Pada penelitian ini, objek berbentuk bujursangkar berwarna merah yang diletakkan 30 cm di depan robot pengikut menghasilkan nilai 45 piksel pada citra yang ditangkap kamera.



2. Hitung jarak objek dengan menggunakan kamera dan bergerak mendekati target berdasarkan informasi kedalaman yang langsung diperoleh oleh kamera tersebut **(Luo, dkk, 2019)**.
3. Hitung perubahan rasio data saat ini dan sebelumnya dengan kamera monokular untuk menentukan apakah robot harus maju, mundur, belok kiri atau kanan **(Hendi & Mohammed, 2016)**.

Seperti dapat dilihat di Gambar 7, proses identifikasi objek di robot pengikut dilakukan oleh *Pi Camera*. Citra yang dihasilkan oleh kamera ini diolah untuk diambil data nilai titik tengah dan jari-jari dari objek. Titik tengah digunakan untuk mendapatkan perubahan kemudi dan ukuran target digunakan untuk menentukan kecepatan motor. Jarak antar pemimpin dan pengikut yang relatif dekat menyebabkan area yang terambil oleh kamera menjadi lebih kecil sehingga benda-benda di sekitar objek harus ditapis agar menjadi lebih sedikit dan tidak harus mendeteksi keseluruhan objek (robot pemimpin). Pada proses pengolahan citra, robot pemimpin diidentifikasi dari objek kotak merah yang ditempel di bagian belakang robot pemimpin, seperti diperlihatkan pada Gambar 8(b) di bawah ini. Untuk robot pemimpin (*leader robot*), digunakan model robot dari Lego Mindstorm seperti dapat dilihat pada Gambar 8 berikut ini.



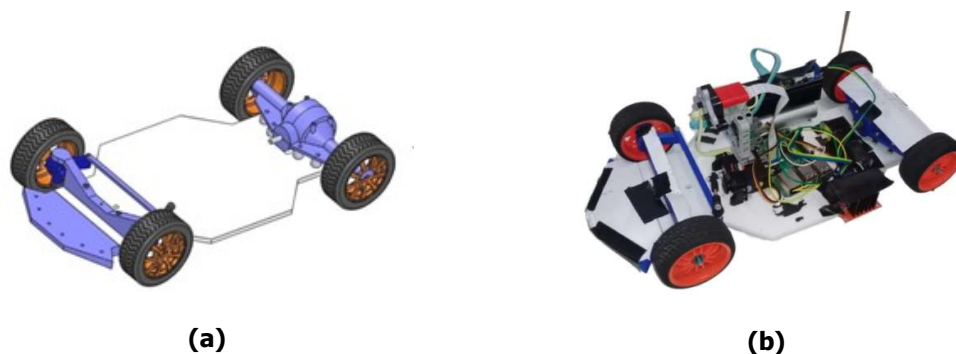
**Gambar 8. (a) Bagian Atas dan (b) Bagian Belakang Robot Pemimpin (*Leader Robot*)**

Pada saat proses pengolahan, citra dari kamera yang memiliki format RGB dikonversi terlebih dahulu ke format HSV. Setelah itu, dilakukan proses *smoothing* pada citra agar piksel-piksel yang menjadi *blob* pada citra dapat terlihat untuk dapat dipilah dan dihapus. Langkah berikutnya adalah melakukan identifikasi bentuk dan ukuran fitur (bujur sangkar merah). Langkah berikutnya adalah menentukan nilai titik tengah dari fitur objek dan ukuran jari-jari target menggunakan fungsi *minimum enclosing circle*.

Titik tengah objek ini digunakan untuk mendapatkan *duty cycle* dari motor servo yang berfungsi untuk mengatur pergerakan kemudi. Untuk mendapatkan nilai kecepatan putar motor DC, digunakan *library* dari program *Python* yang telah ada untuk mengubah nilai keluaran kecepatan yang ada menjadi nilai PWM (*Pulse Width Modulation*). Nilai kecepatan diperoleh dengan membandingkan ukuran objek dengan ukuran referensi. Hal ini dilakukan agar dapat ditentukan apakah nilai kecepatan harus naik atau turun. Berdasarkan data perubahan ukuran ini juga, dapat diambil perubahan kecepatan berbanding perubahan ukuran referensi.

## 2.6 Rancangan Robot Pengikut

Robot yang dirancang pada penelitian ini menggunakan purwarupa basis robot bergerak empat roda dengan sistem kemudi Ackermann, dari hasil penelitian sebelumnya (**Pangestu, 2020**), seperti diperlihatkan pada Gambar 9. Pada basis robot ini kemudian ditambahkan kamera RGB (Pi Camera), *driver* motor L298N IC dan mikrokontroler Raspberry Pi 3 model B.

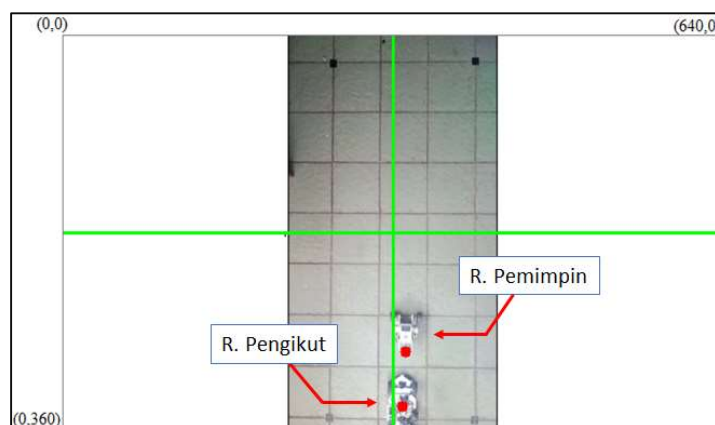


**Gambar 9. Robot Pengikut: (a) Basis dan (b) Terpasang**

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengujian Sistem

Pengujian sistem akan dilakukan menggunakan dua buah jalur yaitu jalur lurus dan berbelok. Robot akan diletakkan seperti pada Gambar 4. Robot pemimpin akan diatur pada interval tertentu untuk mempercepat/memperlambat kecepatannya pada pengujian jalur lurus. Pada pengujian jalur berbelok, robot pemimpin akan diatur sudut beloknya agar membentuk jalur bergelombang seperti sinyal sinus. Percobaan direkam dengan menggunakan kamera lain di atas daerah percobaan. Hasil rekaman ini selanjutnya dianalisis dengan menggunakan bantuan program *Python* untuk mendapatkan data koordinat, perubahan jarak, dan waktu dari kedua robot. Masing-masing robot akan diberi penanda warna yang jelas, sehingga dapat dengan mudah dibedakan, mana robot pemimpin dan mana robot pengikut. Pada Gambar 10 ditunjukkan hasil citra yang diambil dari kamera di atas *platform* percobaan.



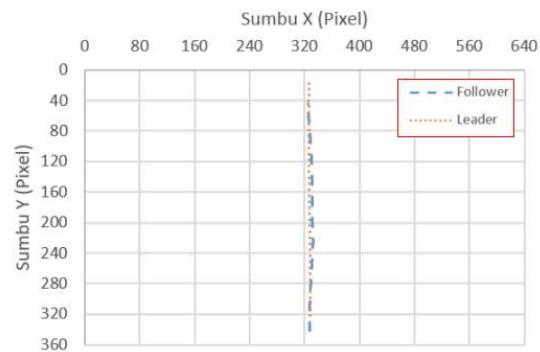
**Gambar 10. Pengambilan Data Percobaan Lurus dan Belok**

Garis hijau yang terdapat pada Gambar 10 merupakan garis tengah pada sumbu X dan Y sesuai resolusi video yang digunakan yaitu 640x360 piksel. Garis tengah yang digunakan

adalah garis horizontal di titik 180 piksel pada sumbu Y dan garis vertikal di titik 320 piksel pada sumbu X.

### 3.2 Hasil Pengujian

Hasil pengujian pada jalur lurus yang direkam menggunakan video, kemudian diolah menggunakan program *Phyton*, untuk ditampilkan di layar PC. Hasil pengolahan data video tersebut diperlihatkan pada Gambar 11 berikut ini.



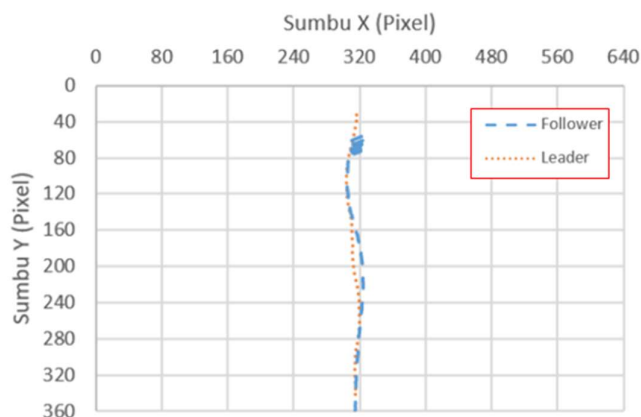
**Gambar 11. Data Percobaan pada Jalur Lurus**

Dari pengolahan video, diperoleh data jarak antar robot, seperti ditampilkan pada Tabel 2. Dari Tabel 2, dapat dihitung jarak rata-rata antara robot pemimpin dan robot pengikut. Di mana hasilnya adalah 17,70 piksel, atau setara dengan 11,79 cm, serta nilai simpangan baku sebesar 0,90 cm. Nilai ini menunjukkan robot pengikut berhasil mengikuti robot pemimpin di depannya sesuai dengan sistem kontrol yang diaplikasikan, serta masih di dalam batasan yang diinginkan. Pada penelitian ini ditetapkan bahwa robot pengikut dianggap gagal mengikuti robot pemimpin jika jarak di antara mereka lebih dari 20 cm.

**Tabel 2. Hasil Percobaan Pada Jalur Lurus**

Data ke-	Jarak X (piksel)	Jarak X (cm)	Jarak Y (piksel)	Jarak Y (cm)	Jarak Antar Robot	
					(piksel)	(cm)
1	4,59	3,06	17,81	11,87	18,39	12,25
2	2,70	1,80	18,75	12,49	18,94	12,61
3	3,74	2,49	18,17	12,11	18,55	12,35
4	3,92	2,61	16,28	10,85	16,74	11,15
5	4,13	2,75	18,80	12,54	19,25	12,82
6	4,93	3,29	17,37	11,58	18,05	12,02
7	4,06	2,70	18,21	12,14	18,66	12,43
8	3,58	2,39	14,66	9,77	15,09	10,05
9	1,76	1,17	16,20	10,79	16,29	10,85
10	3,37	2,25	16,67	11,11	17,01	11,33
Rerata =					17,70	<b>11,79</b>
Std. Dev =					1,35	<b>0,90</b>

Untuk pengujian pada jalur berkelok, hasil yang diperoleh dari pengolahan citra video dapat dilihat pada Gambar 12 di bawah ini. Data yang lebih detail dari pengujian ini dapat dilihat pada Tabel 3.



**Gambar 12. Data Percobaan Pada Jalur Berkelok**

Dari Tabel 3 di bawah ini, dapat dihitung jarak rata-rata antara robot pemimpin dan robot pengikut pada saat robot pemimpin melakukan gerakan berkelok. Hasil perhitungan menunjukkan bahwa jarak rata-rata antara robot pemimpin dan robot pengikut adalah 29,77 piksel, atau setara dengan 19,82 cm, serta nilai simpangan baku sebesar 0,23 cm.

**Tabel 3. Hasil Percobaan Pada Jalur Berkelok**

Data ke-	Jarak X (piksel)	Jarak X (cm)	Jarak Y (piksel)	Jarak Y (cm)	Jarak antar robot	
					(piksel)	(cm)
1	3,02	2,01	29,01	19,34	29,17	19,42
2	0,66	0,44	29,58	19,72	29,59	19,71
3	2,91	1,94	30,09	20,06	30,23	20,13
4	0,32	0,22	29,32	19,54	29,32	19,53
5	0,42	0,28	29,88	19,92	29,88	19,90
6	1,95	1,29	29,89	19,93	29,96	19,95
7	1,19	0,79	29,90	19,94	29,93	19,93
8	2,00	1,34	30,05	20,04	30,12	20,06
9	1,75	1,17	29,81	19,87	29,86	19,89
10	1,35	0,89	29,58	19,72	29,61	19,72
Rerata =					29,77	<b>19,82</b>
Std. Dev =					0,34	<b>0,23</b>

Nilai ini menunjukkan robot pengikut berhasil mengikuti robot pemimpin di depannya sesuai dengan sistem kontrol yang diaplikasikan. Namun, jika diperhatikan lebih teliti pada Tabel 3 di atas, ada dua keadaan yang memperlihatkan bahwa robot pengikut keluar dari batas jarak

yang diinginkan, yaitu pada data ke-3 dan ke-8. Di mana jarak kedua robot bernilai 20,13 cm dan 20,06 cm. Hal ini tidak serta merta menunjukkan bahwa robot gagal mengikuti target. Karena, data berikutnya menunjukkan nilai kurang dari 20 cm. Hal ini justru menjelaskan bahwa robot pengikut mampu mengejar kecepatan robot pemimpin di depannya dan berusaha memper-tahankan jaraknya agar tidak lebih dari 20 cm.

Jika dibandingkan hasil antara pengujian jalur lurus dengan jalur berkelok, tampak perbedaan jarak yang cukup signifikan, antara 11,79 cm dengan 19,82 cm. Hal ini menunjukkan bahwa robot pengikut mampu mengikuti target namun tidak cukup responsif untuk tetap berada sedekat-dekatnya dengan target jika robot pemimpin yang diikutinya bergerak berkelok-kelok.

#### **4. KESIMPULAN**

Penelitian ini berhasil dirancang sebuah purwarupa robot pengikut berbasis visual dengan sistem kendali logika samar. Purwarupa robot yang dibangun ini, berhasil mengikuti robot pemimpin dengan jarak rata-rata 11,79 cm dan simpangan baku sebesar 0,9 cm. Angka ini diperoleh jika robot pemimpin berjalan dalam jalur yang lurus. Nilai ini cukup baik mengingat batas jarak yang diberikan untuk robot ini adalah 20 cm. Sementara itu, jika robot pengikut ini mengikuti robot pemimpin yang berjalan berkelok-kelok, maka jarak rata-rata yang bisa diperoleh adalah 19,82 cm dengan simpangan baku 0,23 cm. Hal ini membuktikan bahwa purwarupa ini memiliki unjuk kinerja yang baik.

#### **UCAPAN TERIMA KASIH**

Bersama ini tim penulis mengucapkan terimakasih kepada tim Laboratorium Sistem Kontrol dan Robotika Jurusan Teknik Elektro (Konsentrasi Mekatronika), Fakultas Teknologi Industri, Universitas Katolik Parahyangan yang telah memberikan bantuan dan fasilitas eksperimen sehingga terlaksananya penelitian ini.

#### **DAFTAR RUJUKAN**

- Aarts, L., & Feddes, G. (2016). European Truck Platooning Challenge. In *Proceedings of the HVTT14: International Symposium on Heavy Vehicle Transport Technology*, (pp. 15-18).
- Axelsson, J. (2017). Safety in Vehicle Platooning: A Systematic Literature Review. *IEEE Transactions on Intelligent Transportation Systems*, 18(5), 1033-1045.
- Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., & Winner, H. (2014). Three Decades of Driver Assistance Systems: Review and Future Perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4), 6-22.
- Bergenheim, C., Shladover, S., Coelingh, E., Englund, C., & Tsugawa, S. (2012). Overview of Platooning Systems. *Proceedings of the 19th ITS World Congress*, (pp. 1-7).
- Castritius, S. M., Hecht, H., Möller, J., Dietz, C. J., Schubert, P., Bernhard, C., Morvilius, S., Haas, C. T. & Hammer, S. (2020). Acceptance of Truck Platooning by Professional

- Drivers on German Highways - A Mixed Methods Approach. *Applied Ergonomics*, 85, 103042.
- Crneković, M., Zorc, D., & Kunica, Z. (2013). Mobile Robot Vision System for Object Color Tracking. *CIM 2013: Computer Integrated Manufacturing and High Speed Machining Proceedings*, (pp. 93 - 98).
- Dewi, T., Wijanarko, Y., Risma, P., & Oktarina, Y. (2018). Fuzzy Logic Controller Design for Leader-Follower Robot Navigation. *Proceedings of 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics*, (pp. 298-303).
- Eichelberger, A. H., & McCartt, A. T. (2016). Toyota Drivers' Experiences with Dynamic Radar Cruise Control, Pre-Collision System, and Lane-Keeping Assist. *Journal of Safety Research*, 56, 67 - 73.
- Hendi, S. H., & Mohammed, F. G. (2016). Robotic System for Tracking Moving Objects Based On Their Color. *Iraqi Journal of Science*, 1833 - 1845.
- Janssen, G. R., Zwijnenberg, J., Blankers, I. J., & De Kruijff, J. S. (2015). *Truck Platooning: Driving the future of transportation*. Retrieved from <https://repository.tno.nl/islandora/object/uuid%3A778397eb-59d3-4d23-9185-511385b91509>.
- Johansson, J., & Schröder, J. (2016). *Automated Vehicle Follower System Based on a Monocular Camera*. (Dissertation KTH Royal Institute of Technology, Stockholm, Swedia). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-192305>
- Kamali, M. Z. M., Kumaresan, N., & Ratnavelu, K. (2017). Takagi–Sugeno Fuzzy Modelling of Some Nonlinear Problems Using Ant Colony Programming. *Applied Mathematical Modelling*, 48, 635-654.
- Kosheleva, O., & Kreinovich, V. (2018). Why Triangular Membership Functions Are Often Efficient in F-Transform Applications: Relation to Probabilistic and Interval Uncertainty and to Haar Wavelets. *Proceedings of 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. (pp. 127-138).
- Li, J., Yang, L., Qu, Y., & Sexton, G. (2018). An Extended Takagi–Sugeno–Kang Inference System (Tsk+) with Fuzzy Interpolation and Its Rule Base Generation. *Soft Computing*, 22, 3155 - 3170.
- Luo, W., Xiao, Z., Ebel, H., & Eberhard, P. (2019). Stereo Vision-based Autonomous Target Detection and Tracking on an Omnidirectional Mobile Robot. *ICINCO*, (pp. 268 -275).
- Matheson, R. (2019). Study Measures How Fast Humans React to Road Hazards. *MIT News*.

- Mordvintsev, A., & Abid, K. (2014). Opencv-python tutorials documentation. Retrieved from [https://opencv24-python-tutorials.readthedocs.io/\\_/downloads/en/stable/pdf/](https://opencv24-python-tutorials.readthedocs.io/_/downloads/en/stable/pdf/).
- Panduru, K. K., Riordan, D., & Walsh, J. (2014). Fuzzy Logic Based Intelligent Energy Monitoring and Control for Renewable Energy. *ISSC 2014/CIICT 2014*, (pp. 96 - 101)
- Pangestu, I.E.A. (2020). Rancangan Bangun Robot Bergerak dengan Sistem Kemudi Ackermann dan Penggerak Roda Belakang Differensial. [Tugas Akhir. Jurusan Teknik Elektro] Universitas Katolik Parahyangan.
- Popov, V., Ostarek, M., & Tenison, C. (2018). Practices and Pitfalls in Inferring Neural Representations. *NeuroImage*, *174*, 340 - 351.
- Sadollah, A. (2018). Introductory Chapter: Which Membership Function is Appropriate in Fuzzy System?. In A. Sadollah (Ed.), *Fuzzy Logic Based in Optimization Methods and Control Systems and Its Applications*. IntechOpen. Retrieved from <https://www.intechopen.com/chapters/62600>.
- Soni, A., & Hu, H. (2018). Formation Control for a Fleet of Autonomous Ground Vehicles: A Survey. *Robotics*, *7*(4), 67.
- Wang, H., Li, R., Gao, Y., Cao, C., Ge, L., & Xie, X. (2019). Target Recognition and Localization of Mobile Robot with Monocular PTZ Camera. *Journal of Robotics*, *2019*, 1 - 12.