Prototype of Duckietown Car Robot Control System on Traffic Light

Ahmad Affandi
Ahmafafandi25@gmail.com
Amani Darma Tarigan
amanidarmatarigan@dosen.pancabudi.ac.id
Yoffi Andinata
Yoffi25@gmail.com

University of Pembangunan Panca Budi

ABSTRACT
This study aims to design and develop a prototype control system for the autonomous Duckiebot robot that can detect and respond to traffic lights in the Duckietown arena. Duckiebot uses Raspberry Pi 3 as a controller and a fisheye camera as a sensor to detect objects around it, including traffic lights. This system implements lane-following mode, where Duckiebot follows a predetermined path until it detects a stop line. After stopping right in front of the stop line, Duckiebot uses the camera to read the color of the traffic light and take appropriate action. If a red light color is detected, Duckiebot will stop, while if a green light color is detected, Duckiebot will continue the journey. Testing was conducted to evaluate Duckiebot's ability to detect traffic lights under various conditions, including detection with different lighting and when the traffic light is only partially visible. The test results showed that Duckiebot had a success rate of 70% in detecting traffic lights with appropriate responses, although the camera's sensitivity to lighting is an important factor affecting detection accuracy. This research is expected to contribute to the development of autonomous robot control systems for smart transportation applications and future robotics research.

Keywords: Duckiebot, Control System, Traffic Light, Autonomous Robot, Lane-Following, Raspberry Pi, Fisheye Camera.

Introduction

Control systems in autonomous robots are becoming an increasingly interesting topic to research along with the rapid development of autonomous vehicle technology. The success of this control system depends not only on the hardware used, but also on software that is able to process data from sensors and make the right decisions in real time. One of the popular platforms that is often used as a test medium in autonomous vehicle research is Duckietown. Duckietown provides a real-world simulation environment with realistic traffic challenges, allowing testing of various autonomous vehicle control scenarios in safe and controlled conditions (Beirigo & Paull, 2018).

Duckietown uses a small robot known as a Duckiebot, which is a simple autonomous vehicle with a camera-based sensor system. The Duckiebot is designed to navigate small roads that resemble urban environments. One of the main challenges in developing the Duckiebot is how the robot can recognize and respond to traffic lights at intersections. This is important because traffic light recognition is a crucial factor in ensuring the safety and smoothness of traffic in environments that involve interactions between multiple Duckiebots (Chen, Lin, & Cheng, 2017).

In Indonesia, the development of Duckietown has become one of the research focuses in several universities. In mid-2017, the Electrical Engineering Study Program of Petra Christian

University collaborated with National Chiao Tung University, Taiwan, to develop the Duckietown project in Indonesia. This collaboration aims to introduce and develop autonomous vehicle technology, especially in higher education environments, as part of an effort to improve student competency and ability in the fields of robotics and artificial intelligence (Lundberg & Olson, 2017).

In the context of traffic light recognition, this research aims to design a prototype of a Duckiebot control system that is able to recognize and react to traffic lights. Accurate traffic light recognition will allow Duckiebot to operate more safely at intersections, so that it can interact with other Duckiebots without causing accidents or traffic disruptions. The development of this system not only supports the overall Duckietown project, but also contributes to broader research in the field of autonomous vehicle technology.

Formulation of the problem:

a. How to design a modification of the Duckietown Car Robot Control System Prototype on Traffic Lights?
b. How to test the reliability and accuracy of the control system on Traffic Lights?

Literature Review

1. Python in Software Development

Python is a very popular programming language in software development, mainly due to its flexibility, ease of learning, and compatibility with various operating systems. Python is freeware software, meaning it is freely available and free to use, copy, or distribute, including full access to the source code that allows for easy modification and further development. Python also provides various tools such as debuggers and profilers that allow developers to track and fix errors in their code (Van Rossum & Drake, 2009).

2. Robot Operating System (ROS)

Robot Operating System (ROS) is an open source platform specifically designed for developing robot software. ROS provides a variety of libraries and tools that allow developers to design, build, and test robot systems without having to write code from scratch (Quigley et al., 2009). The main advantage of ROS lies in its ability as a robotic middleware that can flexibly connect robot hardware with software on a computer system. ROS enables distributed and collaborative development, so developers can share solutions and improve systems through a broad community (Koubaa, 2017).

3. Filesystem Level Concept in ROS

ROS has three main concepts, one of which is the filesystem level. At this level, ROS is organized like an operating system with a clear directory structure. ROS resources contained in this filesystem include packages—basic modules containing program code, metapackages—a collection of related packages, packages manifests—metadata files that describe packages, metapackage manifests, and messages and services used for communication between modules (Mason & Galen, 2013). This level serves as a foundation for developing robotic systems, where developers can organize code and configurations as needed.

4. Computational Graphics Level Concept in ROS

At the computational graph level, ROS implements a peer-to-peer system that allows each node in the network to interact with each other. Nodes in ROS are processes that perform specific tasks, and all nodes can share information with each other through communication mechanisms such as topics, services, and messages (Stöckert et al., 2014). The master in ROS acts as a communication management center, ensuring that each node can connect and exchange information. This allows the computational process to be distributed efficiently, so that the robotic system can operate more reliably and scalably.

5. ROS Implementation in Robotics Development

The use of ROS as middleware in robot software development allows for easier hardware integration, as well as the use of various available tools. ROS provides a framework that supports modularity, where each function can be developed as a separate node. This reduces the overall complexity of the system and makes it easier to maintain and develop new features. In addition, ROS also supports various sensors and actuators commonly used in robotics, such as LiDAR, cameras, and inertial sensors (Quigley et al., 2009).

Research Method

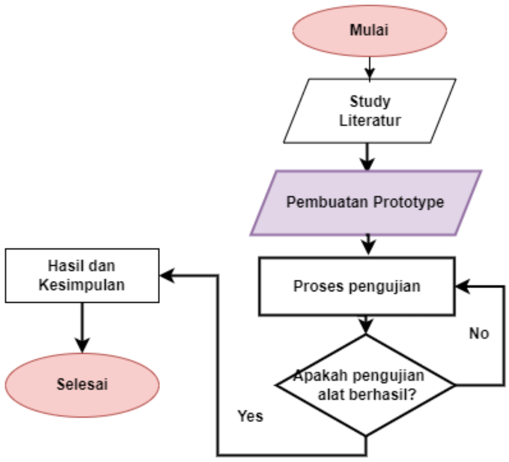

Figure 1. Flowchart

The following is a flowchart description:

1) Start: The starting point of the process.
2) Literature Study: Conducting research and reviewing relevant literature to understand theoretical background or gather information.
3) Prototyping: Designing and developing a prototype based on literature studies and the problems encountered.
4) Process: Testing or analyzing prototypes through experiments or practical applications.
5) Was the Tool Testing Successful?
6) Decision point: If “Yes”, proceed to Results and Conclusions.
7) If “No”, go back to “Prototyping” to refine the design or approach.
8) Results and Conclusions: Based on successful testing, analyze the data and summarize the results.
9) Done: The process ends here.

This flow shows the process of developing and testing the tool, from research to final conclusions.

Localization and Color Detection in Traffic Light

In robotic systems, especially for autonomous vehicles such as Duckiebot, traffic light recognition and response are very important elements. One of the crucial steps in traffic light

recognition is localization or the process of determining the position of the traffic light in an area. Localization allows the robot to recognize the traffic light in the correct position, so that the color of the light read is not considered a signal from the traffic light if it does not match the specified location. This prevents false detection due to similar lights or colors in the surrounding area, so that the system only reacts to the relevant traffic light at the intersection.
This localization process is based on the assumption that each intersection has traffic light placement with similar positions, allowing the system to perform identification more consistently. Thus, localization reduces the possibility of errors in identifying traffic light colors that may occur in complex environments with many light sources or similar colors:

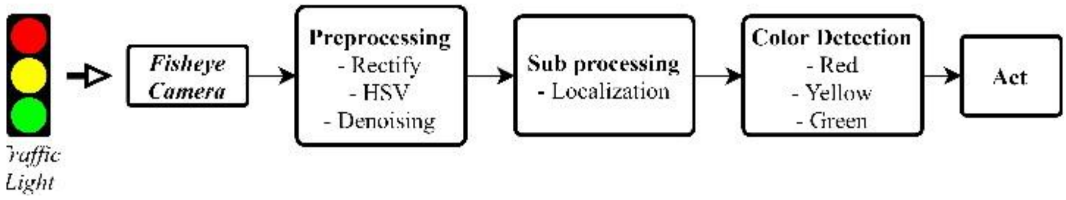

Figure 2. Block diagram of the Duckiebot controller for traffic lights

Once the traffic light position is recognized through localization, the robotic system will use color detection to read the active light color (red, yellow, or green). At this stage, the camera sensor on the robot identifies the color of the traffic light and determines the action to be taken based on the detection results:

- Red Light: The system stops and waits for the light to change.
- Yellow Light: The system is preparing to stop or continue, depending on the traffic context.
- Green Light: The system continues the journey.

Thus, the combination of localization and color detection enables the robotic system to make the right decision at the intersection, according to the signals given by the traffic lights. This process not only increases the operational safety of autonomous vehicles, but also ensures a smoother and more controlled traffic flow.

Results and Discussion

Duckiebot uses Raspberry Pi 3 as the main controller that controls all robot functions. To detect and understand its surroundings, Duckiebot is equipped with a fisheye camera. This camera functions as a visual sensor that captures images around Duckiebot, including lane conditions and signals from traffic lights. The images captured by the fisheye camera are then processed by Raspberry Pi 3, which acts as the brain of the system. Through image processing, Duckiebot can recognize important elements in its environment, such as lane lines, traffic lights, and other objects relevant to navigation.

The results of this image processing are used by the system to make decisions regarding the movements that Duckiebot must make. The decisions taken determine how the drive motors operate to keep Duckiebot on track or stop at the stop line when the traffic light is red. The drive motors on Duckiebot use stepper motor HAT, which allows precise control of wheel movements. With this system, Duckiebot is able to navigate automatically and respond to changing conditions on the test track, according to existing traffic signals and obstacles. Figure 1 shows the overall design of the system used by Duckiebot to carry out its function as an autonomous robot.

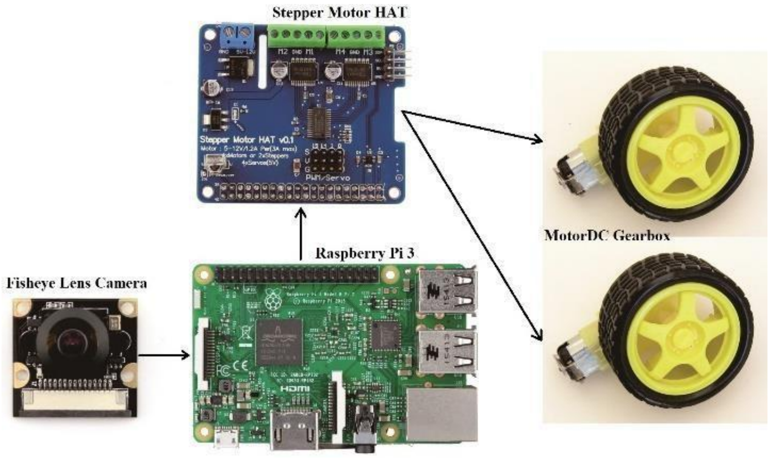

Figure 3. System Design

The arena for testing the Duckietown robot control system is designed to mimic real traffic conditions using intersections equipped with traffic lights. The arena is 240 cm long and 240 cm wide, providing enough space for Duckiebot to simulate maneuvering at intersections. Each lane in the arena is 20.5 cm wide, allowing Duckiebot to move safely in a predetermined lane. Around the arena there is a 5 cm wide white border that serves as a visual boundary for the driving area. To distinguish the lanes, there is a 2.5 cm wide yellow dotted line that acts as a separator between lanes, helping Duckiebot maintain its position in the middle of the lane during navigation.

The traffic light is placed on the right side of the lane and is directly in front of the red stop line. This placement allows the Duckiebot camera to clearly detect the traffic light when it is near the stop line. This strategic placement of the traffic light is designed so that Duckiebot can make the right decision based on the color of the detected traffic signal. The design of this arena emphasizes proportionality of size and the use of contrasting visual elements, such as edges and lane dividers, to make it easier for the Duckiebot camera system to recognize and respond to important elements during the test. All of these aspects aim to improve the detection accuracy and performance of the Duckiebot control system when facing complex intersection scenarios .

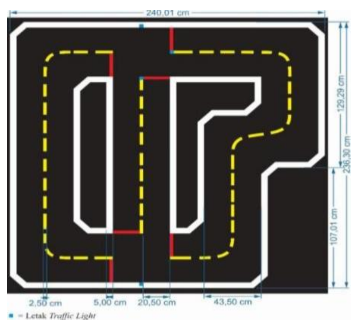

Figure 4. Duckietown Arena Design

In detecting traffic lights, Duckiebot utilizes the stop line provided by the Duckietown developer as a reference. When lane following mode is activated, Duckiebot will move along the specified lane until it detects the presence of a stop line, which is marked with a red line. Once Duckiebot reaches the stop line, it will stop right before the line. At that time, the fisheye camera installed on Duckiebot will read the color of the traffic light at the intersection.

If the camera detects red on the traffic light, Duckiebot will remain stopped, waiting for the signal to change as an indication that the intersection is not yet safe to pass. However, if the camera detects green, Duckiebot will continue the journey by reactivating lane following mode, following the designated lane. This system allows Duckiebot to operate autonomously by obeying the traffic rules in the simulation arena, thus providing a realistic simulation of real traffic conditions.

Figure 5. Traffic Light Detection Test Results

Duckiebot action testing aims to evaluate Duckiebot's response when detecting traffic lights. This test is done to ensure that Duckiebot can make the right decision based on the detected color of the traffic light, despite variations in the position or visibility of the traffic light.

One of the tests conducted was testing a partially covered traffic light. The purpose of this test was to determine whether Duckiebot could still detect the presence of a traffic light even though only part of the traffic light was visible. In this test, the traffic light was blocked by an object or its position was partially covered from the camera's view. The test results showed that even though the traffic light was partially covered, the Duckiebot camera could still detect the traffic light signal as long as the color range detected by the camera was still clearly readable. This shows that Duckiebot is able to adjust color detection even though there is interference with direct vision of the traffic light.

The test positions performed illustrate that as long as the color range read by the camera remains as programmed (red, yellow, or green), Duckiebot can still make decisions based on the detected colors. This test is important to determine the resilience of Duckiebot's detection system to real-world conditions, where objects or obstacles can partially block the camera's view.

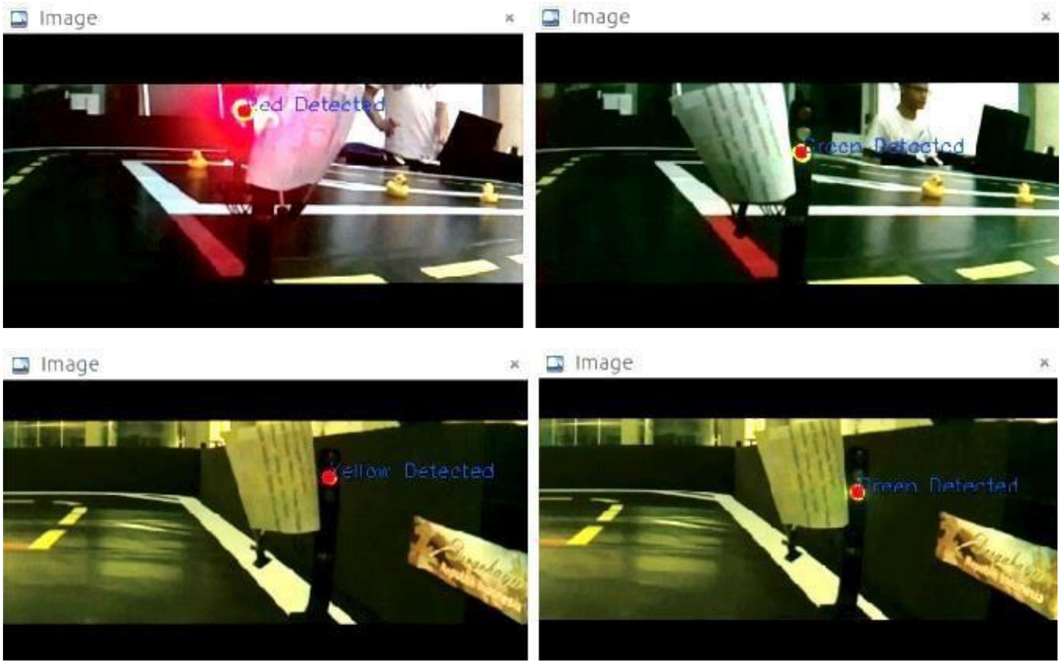

Figure 6. Traffic Light Detection Test Results

Based on the results of the design and testing of Duckiebot in detecting and acting according to traffic lights, there are several findings that can be used as a reference for further improvement and development of this system.

1). Duckiebot's Success in Detecting Traffic Lights

The test results show that Duckiebot successfully detected and responded to traffic lights with a success rate of 70% when walking straight from an intersection. This percentage shows that although Duckiebot can recognize traffic light colors quite well, there is still a possibility of detection errors that need to be corrected. Factors such as the position and distance from the traffic light, as well as the camera's sensitivity to lighting conditions, may play a role in this success rate. Therefore, further optimization of the image processing algorithm is needed so that Duckiebot can be more consistent in detecting traffic lights in various conditions.

2). Camera Resistance to Banner Images Similar to Traffic Lights

One of the key findings in the test was that Duckiebot’s camera was not affected by banner images that resembled traffic lights. This shows that Duckiebot’s detection system is smart enough to distinguish between valid traffic lights and other objects that have similar shapes or colors. The use of more specific color and shape-based detection algorithms for traffic lights is key to avoiding interference from other objects around the intersection.

3). Camera Sensitivity to Lighting

The fisheye camera used by Duckiebot is quite sensitive to changes in lighting. Therefore, even lighting throughout the test arena is very important to ensure accurate detection. When lighting is uneven, the camera can have difficulty distinguishing the color of the traffic light from the background, which leads to errors in decision making. This indicates the need for a consistent lighting system and better camera settings to work optimally in various lighting conditions.

4). Influence of Duckietown Arena Rules and Design

The design of the Duckietown arena, including the material and size of the track, greatly affects Duckiebot's performance when running in lane following mode. The width of the track and the material used to construct the test track can affect Duckiebot's ability to stay on the right lane. In addition, the clear placement of lane dividing lines and stop lines also play an important role in Duckiebot's navigation. A more representative arena with the right size and design can help Duckiebot to work more efficiently and accurately in identifying intersections and responding to traffic signals.

Table 1. Duckiebot test results in detecting traffic lights and factors that influence its performance:

Factor	Test Results	Analysis	Recommendation
Traffic Light Detection Success	70% success rate in detecting traffic lights when driving straight	Duckiebot successfully detected and responded to traffic lights, but there is a possibility of false detection.	Optimize image processing algorithms to improve detection accuracy and decision making.
The Effect of Banner Images Similar to Traffic Lights	The camera is not affected by banners similar to traffic lights.	The camera can distinguish objects that are similar to traffic lights.	Strengthen the algorithm to distinguish objects similar to traffic lights more accurately.
Camera Sensitivity to Lighting	Sensitive to changes in lighting, even lighting is required for accurate detection.	Changes in lighting can affect the camera's ability to read traffic light colors.	Use an even lighting system to avoid detection errors due to changes in light.
Duckietown Arena Design Influences	The arena design, including the width of the lanes and dividing lines, affects the Duckiebot's path.	The width of the track and the design of the arena affect Duckiebot's ability to follow the track accurately.	Adjust the arena design to more optimal specifications so that Duckiebot can follow the path stably.
Stop Line Detection Accuracy	Duckiebot can detect stop lines with good accuracy	The stop line serves as a reference for Duckiebot to stop before detecting a traffic light.	Improve the camera settings to better detect stop lines in various lighting conditions.

The table above summarizes the test results with factors that affect Duckiebot's performance, an analysis of each factor, and recommendations for improving Duckiebot's traffic light detection and navigation systems.

Conclusion

Based on the results of this test, it can be concluded that although Duckiebot has been quite successful in detecting and responding to traffic lights, there is still room for improvement. The development of a more sophisticated image processing system, more even lighting, and improvements to the design of the test arena can improve Duckiebot's performance in detecting and acting according to traffic signals. As a next step, optimization of the detection algorithm and testing under more varied conditions is needed to improve Duckiebot's accuracy and response speed in various situations.

References

[1] Agung.S., Elvan.Y., Edi.S, "Rancang Bangun Home Automation Berbasis Raspberry Pi3 Model B Dengan Interface Aplikasi Media Sosial Selegram Sebagai Sistem Kendali", Jurnal Informatika Polinemat, 1(2), hlm.1-10, 2018.

[2] Dasmen. R. N, "Implementasi Raspberry Pi3 Sebagai Wireless Acces Point Pada STIPER Sriwigama Palembang", Jurnal Informasi Pengembangan IT, 3(3), hlm.387-393, 2018.

[3] Erwin S.L., Handry.K., Resmana.L, "Pengenalan Traffic Light Pada Robot Mobil Duckietown" Jurnal Teknologi Informasi dan Komunikasi, 7(2), hlm. 89-93, 2018.

[4] L. Paull et al., "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," Proc. – IEEE Int. Conf. Robot. Autom., pp. 1497–1504, 2017.

[5] Novrianda.R, "Implementasi SMS Gateway Pada Sistem Pengendali Lampu Ruangan Berbasis Mikrokontroller", Jurnal Maklumatika, 3(2), hlm.130-139,2017.

[6] Rahmat. N. D, Rasmila, "Implementassi Raspberry Pi3 Pada Sistem Pengontrol Lampu Berbasis Raspbian Jessie", Jurnal Edukasi dan Penelitian Informatika, 5(1), hlm.46-53 ,2019.

[7] Riki.R, Sutarti, Anharudin, "Rancang Bangun Kamera Pengawas Menggunakan Raspberry dengan Aplikasi Telegram Berbasis Internet of Things", Jurnal Prosisko, 8(1), hlm 18-32, 2021.

[8] S. K. Dewi, R. D. Nyoto, and E. D. Marindani, "Perancangan Prototipe Sistem Kontrol Suhu dan Kelembaban pada Gedung Walet dengan Mikrokontroler berbasis Mobile", JEPIN (Jurnal Edukasi dan Penelit. Inform, vol. 4, no. 1, pp. 36–42, 2018.

[9] Seday, A. Yuniarti.E, & Sanjaya.E, "Rancang Bangun Home Automation Berbasis Raspberry Pi3 Model B dengan Interface Aplikasi Media Sosial Telegram Sebagai Kendali" Al-Fiziya, Jurnal of Materials Science, Geophysics, Instrumentation and Theoretical Phisics, 1(2), hlm.42-47, 2019.