

# PERANCANGAN SISTEM PARKIR OTOMATIS SUBSISTEM : DETEKSI GARIS PADA ROBOT MOBIL PENGIKUT GARIS MENGGUNAKAN METODE THRESHOLDING CITRA

Ahmad Fida<sup>1</sup>, Ir. Agus Virgono, M.T.<sup>2</sup>, Randy Erfa Saputra, S.T., M.T.<sup>3</sup>

<sup>1,2</sup>Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

Jln. Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia

[ahmadfida7@gmail.com](mailto:ahmadfida7@gmail.com)<sup>1</sup>, [avirgono@telkomuniversity.ac.id](mailto:avirgono@telkomuniversity.ac.id)<sup>2</sup>, [@telkomuniversity.ac.id](mailto:@telkomuniversity.ac.id)<sup>3</sup>

## ABSTRAK

Sistem parkir otomatis merupakan sistem yang mempunyai prinsip kerja seperti *valet parking*. Sistem ini akan merancang sebuah robot mobil yang dapat melakukan kegiatan parkir secara otomatis. Lahan parkir yang digunakan merupakan lahan parkir yang mempunyai garis hitam dilantai sebagai jalur pergerakan robot dan menggunakan beberapa tanda sebagai informasi. Robot ini dapat bergerak dengan bantuan navigasi *line follower* menggunakan sensor kamera. Pengolahan citra diproses menggunakan sebuah mikrokomputer pada robot mobil. Kamera mendeteksi garis pada arena parkir sehingga dapat menentukan arah jalan robot dan nomor slot parkir yang dituju.

Metode yang digunakan yaitu metode thresholding citra pada citra berwarna dan citra keabuan. Hasil pengujian menunjukkan bahwa metode ini dapat menghasilkan keluaran yang bagus apabila nilai thresholding berada diantara dua puncak pada histogram. Hasil pengujian kedua menunjukkan bahwa pencahayaan pada arena parkir berpengaruh terhadap keluaran citra. Hasil pengujian ketiga menunjukkan bahwa kecepatan robot mobil berpengaruh terhadap tingkat keberhasilan sistem dalam mendeteksi dan memilih percabangan. Tingkat keberhasilan sistem dalam memilih percabangan yaitu 100% pada kecepatan PWM 80 dan 60% pada kecepatan PWM diatas 120.

**Kata Kunci:** deteksi garis, metode threhsolding, pengolahan citra

## ABSTRACT

Automatic parking system is a system that has a valet parking like behaviour. In this system, mobile robot is designed to be able to park the robot automatically. The parking arena that used in this system is a modified parking arena with line and symbols on the floor for robot's navigation and information. This robot used that line for following navigation and using digital image processing.

Image processing is processed in a microcomputer inside the robot. Camera detect lines and symbols in each parking slot and able to navigate through the line until the robot stop at the desired parking slot. The processed image is transferred to microcontroller to move the robot. From the results of the test, we can conclude that the best thresholding value is between the two peaks in histogram. From the second test, the result is illumination affect the detection process. And from the last test, the result is the robot's speed affect the performance of the lines detection. System can work 100% with 80 PWM and 60% with more than 120 PWM.

**Keywords:** line detection, thresholding method, image processing

## I. PENDAHULUAN

### 1.1. Latar Belakang

Memarkir mobil merupakan salah satu kegiatan yang cukup banyak memakan waktu. Berdasarkan penelitian yang dilakukan oleh Donald Shoup dengan judul "*Cruising for Parking*", waktu rata-rata yang dibutuhkan seseorang untuk mendapatkan lahan parkir yang kosong adalah sekitar 3.3 menit.<sup>[1]</sup> Sehingga untuk mengatasi masalah tersebut diperlukan suatu sistem yang dapat melakukan aktifitas parkir secara otomatis. Hal inilah yang mendorong penulis untuk merancang sebuah sistem yang dapat melakukan kegiatan tersebut. Sistem parkir yang akan dirancang adalah sebuah robot mobil yang dapat melakukan parkir secara otomatis. Sistem ini menggunakan bantuan dari sistem parkir cerdas yang telah di implementasikan sebelumnya.<sup>[2]</sup> Sistem parkir cerdas tersebut merupakan sumber informasi bagi robot untuk informasi ketersediaan slot parkir. Robot akan menggunakan kamera yang berfungsi sebagai sensor visual yang dapat mendeteksi garis dan tanda dilahan parkir. Pergerakan robot menyesuaikan garis (*line*) yang terdapat dilantai parkir yang dideteksi oleh kamera robot. Kelebihan dari navigasi menggunakan *line follower* ini yaitu lebih dapat diandalkan karena garis tersebut berfungsi sebagai jalur yang akan dilalui oleh robot mobil.<sup>[3]</sup> Sensor kamera yang digunakan untuk mendeteksi garis mempunyai kelebihan seperti area penglihatan yang lebih luas dan dapat memprediksi jalur yang ada di depan.<sup>[4]</sup>

## II. LANDASAN TEORI

### 2.1 Robot Pengikut Garis

*Line follower robot* adalah robot yang didesain untuk dapat berjalan mengikuti garis yang membentuk sebuah alur *mapping* tertentu (biasanya garis yang dipakai berwarna hitam atau putih). Beberapa hal yang perlu dalam merancang robot pengikut garis ini adalah mekanika, elektronika, dan algoritma (*software*) dari robot (Susilo. 2010).

### 2.2 Raspberry Pi

Raspberry Pi (Raspi) adalah komputer papan tunggal (*singleboard circuit*) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program-program seperti pada komputer umumnya. Raspi mempunyai spesifikasi yang cukup tinggi meskipun ukurannya kecil.

### 2.3 Citra RGB

Citra warna RGB adalah salah satu pemodelan warna citra yang merupakan gabungan dari warna-warna dasar yaitu Red, Green, dan Blue. Setiap warna dasar, diberi rentang nilai dari 0 sampai 255. Pada format citra RGB 24 bit, setiap poin informasi pixel disimpan ke dalam 1 byte data. 8 bit pertama pada kanal warna biru, selanjutnya pada kanal warna hijau dan terakhir merah.

#### 2.5.1 Citra Grayscale

Citra grayscale adalah citra digital yang hanya memiliki 1 nilai kanal pada setiap pixelnya, dimana nilai setiap pixel tersebut menunjukkan tingkat intensitas warnanya. kemungkinan warna yaitu hitam dan putih. Citra grayscale memiliki derajat keabuan 8 bit.

#### 2.5.2 Thresholding

Metode *thresholding* citra adalah salah satu metode paling sederhana dari segmentasi citra. Segmentasi citra merupakan metode yang berfungsi untuk memisahkan antara objek dengan latar belakang dalam suatu citra. Metode *thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih untuk mengetahui wilayah objek dan latar belakang secara jelas dengan menggunakan suatu nilai ambang sebagai parameter pemisahannya.

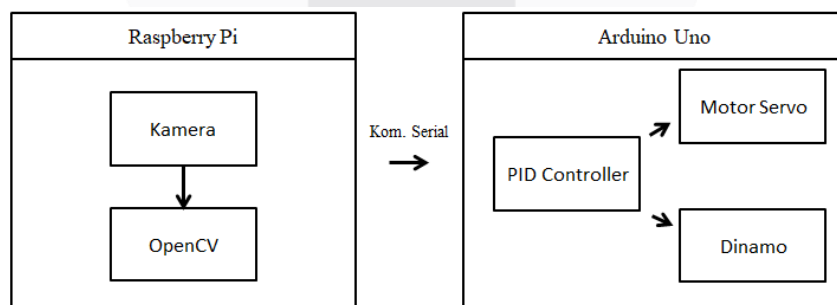
#### 2.5.3 Otsu

Metode Otsu adalah salah satu metode *thresholding* citra yang menggunakan nilai ambang otomatis  $T$  berdasarkan citra masukan. Pendekatan metode ini dengan cara analisis diskriminan yaitu menentukan suatu variable yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimumkan variable tersebut agar dapat memisahkan objek dengan latar belakang. Tujuan utama dari metode ini yaitu memisahkan objek dan latar belakang menggunakan perhitungan nilai minimum *Within Class Variance* atau nilai maksimum *Between Class Variance*. Nilai ini didapat dari perhitungan probabilitas, rata-rata, dan *variance* dari tiap kemungkinan nilai *threshold* yang ada.

## III. PERANCANGAN SISTEM

Pada perancangan sistem ini akan membahas mengenai perancangan sistem yang dibuat meliputi Gambaran Umum Sistem, Perancangan Arena Parkir dan Perancangan Perangkat Lunak.

### 3.1 Gambaran Umum Sistem

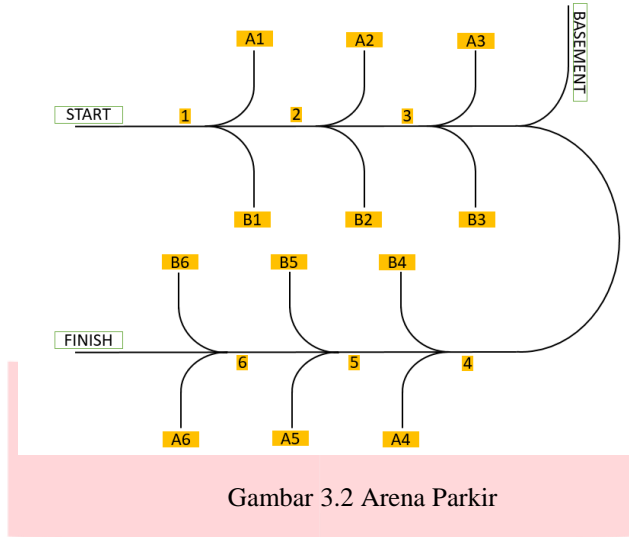


Gambar 3.1 Blok Diagram Sistem Robot Mobil

Kedua kamera dihubungkan dengan Raspberry Pi dengan port yang berbeda. Kamera Raspberry Pi dihubungkan dengan port CSI dengan kabel fleksibel (*ribbon cable*) sedangkan kamera webcam dihubungkan dengan port USB. Data hasil pengolahan citra dari masukan kamera akan dikirimkan ke Arduino melalui komunikasi serial. Komunikasi serial ini dihubungkan dengan kabel jumper. Komunikasi antara kedua perangkat ini bersifat satu arah (dari Raspi ke Arduino). Data kiriman Raspi akan digunakan untuk masukan PID Controller Pada Arduino untuk menggerakkan dinamo dan servo.

### 3.2 Perancangan Arena Parkir

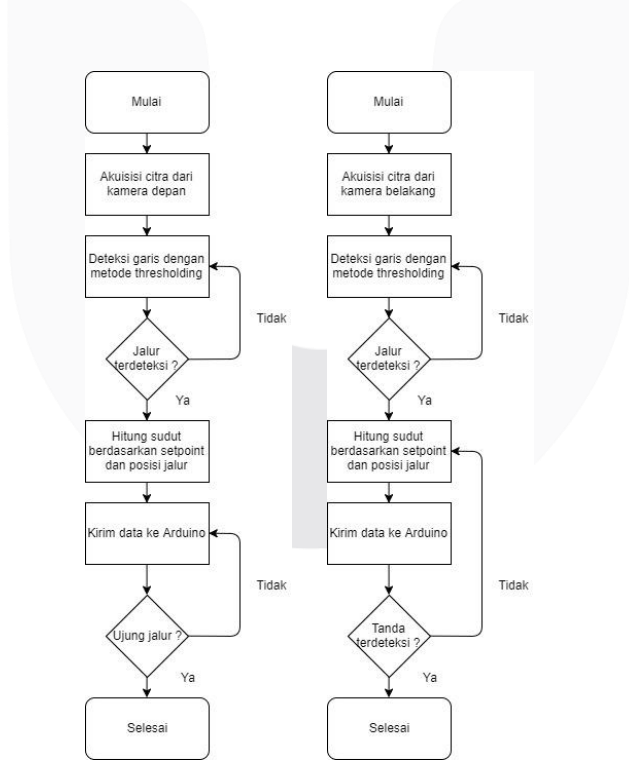
Arena parkir yang akan dirancang pada umumnya mirip dengan arena parkir seperti biasa. Perbedaannya ada pada lantai parkir yang mempunyai garis sebagai jalur dan tanda di setiap slot parkir. Garis mempunyai ciri tertentu seperti berwarna hitam untuk mempermudah pendeteksian menggunakan kamera. Pada setiap slot akan diberi tanda yang berisi informasi nomor masing-masing slot. Pada setiap belokan didesain untuk menyesuaikan mekanika robot mobil yang mempunyai desain *steering* sehingga sudut maksimum yang dihasilkan dari mobil ini tidak lebih besar dari 30°.



Gambar 3.2 Arena Parkir

### 3.3 Perancangan Perangkat Lunak

Perangkat lunak yang dirancang pada sistem ini terletak di mikrokomputer Raspberry Pi dengan OS Raspbian dan aplikasi pengolah citra OpenCV. Sistem akan mengolah citra menggunakan masukan dari perangkat kamera. Sistem akan mendeteksi garis (jalur) menggunakan metode thresholding (RGB, HSV, Otsu). Setelah jalur terdeteksi, maka akan dihitung sudut beloknya dan kemudian dikirimkan ke Arduino. Pada gerakan maju, sistem akan menggunakan masukan dari kamera depan sedangkan pada saat gerakan mundur sistem menggunakan masukan dari kamera belakang. Berikut adalah flowchart sistem secara keseluruhan.



Gambar 3.2 Flowchart Sistem

**IV. PENGUJIAN DAN ANALISA**

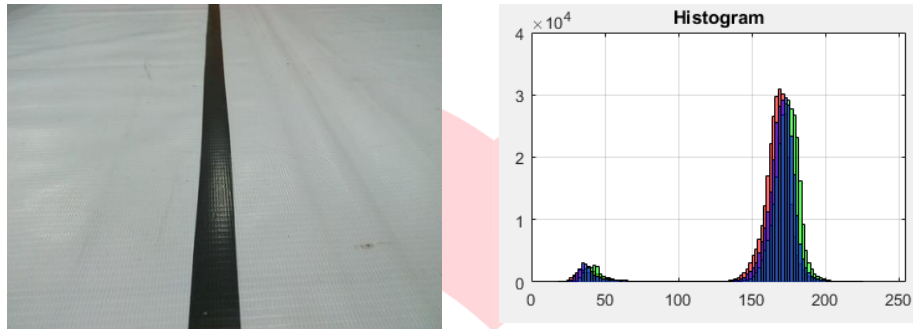
Pada bab ini akan dibahas mengenai pengujian dan analisa dari sistem yang telah dibuat. Pengujian meliputi pengujian *hardware* dan pengujian *software* yang bertujuan untuk mengetahui apakah *hardware* ataupun *software* dalam sistem ini dapat bekerja dengan baik dan berjalan dengan baik sesuai dengan apa yang diinginkan.

**4.1 Pengujian Nilai Thresholding**

Tujuan pengujian ini adalah untuk mengetahui nilai thresholding terbaik dari citra masukan. Dengan pengujian ini, dapat diketahui nilai thresholding yang cocok untuk diimplementasikan dalam sistem. Pengujian ini menggunakan histogram untuk referensi memilih nilai threshold.

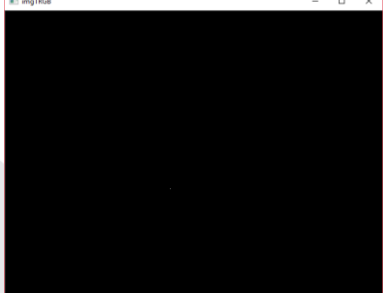

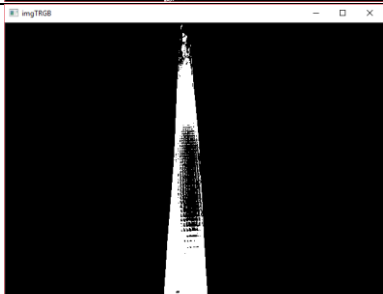
Dari hasil pengujian, didapatkan hasil berupa gambar thresholding citra dengan nilai threshold yang berbeda-beda. Berikut adalah tabel nilai thresholding dan hasil citra yang sudah di threshold.



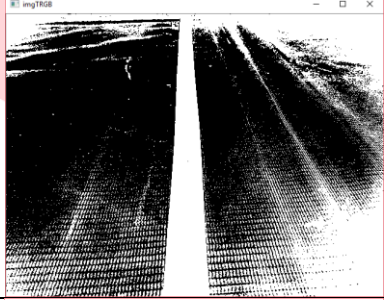
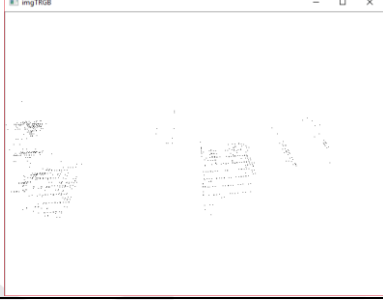
- Sampel citra masukan dengan histgoram



Gambar 4.1 Citra sampel dan histogramnya

Tabel 4.1 Nilai threshold RGB citra ke-1

PengujianKe -	Nilai Threshold	Citra Keluaran	Hasil
1	(0,0,0) sampai (25,25,25)		Sangat Buruk
2	(0,0,0) sampai (35,35,35)		Buruk
3	(0,0,0) sampai (50,50,50)		Kurang Baik

Pengujian Ke -	Nilai Threshold	Citra Keluaran	Hasil
4	(0,0,0) sampai (100,100,100)		Baik
5	(0,0,0) sampai (150,150,150)		Kurang Baik
6	(0,0,0) sampai (170,170,170)		Buruk
7	(0,0,0) sampai (200,200,200)		Sangat Buruk

Histogram citra sampel menunjukkan bahwa keempat citra termasuk dalam kategori *bimodal histogram* yaitu citra yang mempunyai dua kelas (*foreground* dan *background*). Hasil analisis dari pengujian diatas menunjukkan bahwa nilai threshold keempat dapat menghasilkan threshold yang bagus karena titik ini adalah titik yang berada diantara nilai maksimum intensitas objek dan nilai minimum intensitas latar belakang. Hal ini sesuai dengan teori untuk memilih nilai threshold yang menyatakan bahwa nilai T terbaik berada di tengah-tengah (*minimum*) diantara objek (*local maximum*) dan latar belakang (*global maximum*).<sup>[11]</sup>

#### 4.2 Pengujian Percabangan

Tujuan pengujian ini adalah untuk mengetahui tingkat keberhasilan system dalam mendeteksi dan memilih percabangan pada kecepatan yang berbeda. Robot akan diletakkan pada jalur lurus sepanjang 100 cm dan terdapat 3 cabang pada ujung jalur. Untuk pengujian pertama robot akan diatur untuk memilih jalur sebelah kiri, kemudian kanan dan lurus. Pada pengujian ini ada tiga jenis kecepatan PWM robot mobil yaitu 80, 100, dan 120.

Tabel 4.11 Perintah Belok Kiri

Pengujian ke -	Kecepatan (PWM)	Hasil
1	80	Berhasil
2	80	Berhasil
3	80	Berhasil
4	80	Berhasil
5	80	Berhasil
6	100	Gagal
7	100	Berhasil
8	100	Berhasil
9	100	Berhasil
10	100	Berhasil
11	120	Gagal
12	120	Berhasil
13	120	Berhasil
14	120	Gagal
15	120	Berhasil

Tabel 4.12 Perintah Jalan Lurus

Pengujian ke -	Kecepatan (PWM)	Hasil
1	80	Berhasil
2	80	Berhasil
3	80	Berhasil
4	80	Berhasil
5	80	Berhasil
6	100	Berhasil
7	100	Berhasil
8	100	Berhasil
9	100	Berhasil
10	100	Berhasil
11	120	Berhasil
12	120	Berhasil
13	120	Berhasil
14	120	Berhasil
15	120	Berhasil

Tabel 4.13 Perintah Belok Kanan

Pengujian ke -	Kecepatan (PWM)	Hasil
1	80	Berhasil
2	80	Berhasil
3	80	Berhasil
4	80	Berhasil
5	80	Berhasil
6	100	Berhasil
7	100	Berhasil
8	100	Gagal
9	100	Berhasil
10	100	Berhasil
11	120	Gagal
12	120	Berhasil
13	120	Berhasil
14	120	Berhasil
15	120	Gagal

Pada pengujian ini, mobil robot yang diperintahkan belok kiri pada percabangan mempunyai tingkat keberhasilan sebesar 100%, 80%, dan 60% pada kecepatan PWM 80, 100, dan 120. Pada perintah lurus, tingkat keberhasilan sebesar 100% pada semua kecepatan. Dan pada perintah belok kanan, tingkat keberhasilan 100%, 100%, dan 60% pada kecepatan PWM 80, 100, dan 120. Analisis pada kegagalan yang terjadi pada pengujian ini terjadi karena kecepatan jalan mobil lebih cepat daripada proses pengolahan citra yang menyebabkan pendeteksian gagal memilih percabangan. Kegagalan mulai terjadi pada saat kecepatan PWM robot mobil bernilai 100 dan persentase kegagalan bertambah pada saat kecepatan PWM bernilai 120.

## V. KESIMPULAN DAN SARAN

Dari hasil pengujian dan analisis, maka dapat diambil beberapa kesimpulan sebagai yaitu pada thresholding RGB dapat mendeteksi dengan baik pada citra yang mempunyai pencahayaan yang bagus dan kontras warna antara objek dan background yang jelas. Namun thresholding RGB kurang baik apabila lingkungannya mempunyai pencahayaan yang kurang merata karena cahaya dapat mempengaruhi nilai pada intensitas setiap pixelnya. Lingkungan gelap dapat membuat citra masukan dominan berwarna hitam dan lingkungan terang dapat membuat citra masukan dominan berwarna putih.

Thresholding Otsu dirasa lebih cocok untuk diimplementasikan pada sistem ini karena metode ini dapat beradaptasi dengan perubahan pencahayaan pada lingkungan robot. Selain itu, metode ini juga menghasilkan citra biner yang lebih jelas dan lebih tajam dibandingkan dengan metode threshold RGB.

Penulis berharap pada pengembangan selanjutnya untuk dapat disempurnakan dengan cara meratakan pencahayaan pada lingkungan robot. Dengan cara ini, diharapkan tidak terjadi kesalahan pada pendeteksian dikarenakan perubahan intensitas cahaya pada lingkungan robot. Kedua yaitu kecepatan robot disesuaikan dengan kecepatan proses pengolahan citra sehingga tidak terjadi kegagalan dalam proses pendeteksian dikarenakan kecepatan robot mobil yang tinggi.

**Daftar Pustaka**

- [1]. Shoup, Donald. 2006. *Cruising for Parking*. Transport Policy, vol.13.
- [2]. Ulfah , Restu Aulia. 2015. *Implementasi Sistem Parkir Cerdas Di Universitas Telkom. Subsystem : Pengolahan Citra Digital Dengan Deteksi Tepi Canny Dan Embedded System*. Bandung : Telkom University.
- [3]. Pennington, Nicole Marie. 2012. *Line Following Navigation*. University of Tennessee Honor Thesis Projects.
- [4]. Rizal, Muhammad. 2013. *Implementasi Kamera OV7670 Sebagai Pendeteksi Garis Pada Robot Line Follower*. Malang : Universitas Brawijaya.
- [5]. Rusiana, Endang. 2013. *Desain dan Implementasi Sensor Kamera Sebagai Pendeteksi Garis Pada Robot Mobil Pengikut Garis*. Bandung : Telkom University.
- [6]. Sahib, Muhammad. September 2014. *Robust Method for Robotic Guidance Using Line Follower Method*. International Journal of Engineering and Innovative Technology (IJEIT), Vol. 4, Issue 3.
- [7]. E. Elhady, Walaa. 2013. *Implementation And Evaluation of Image Processing Techniques on a Vision Navigation Line Follower Robot*. Journal of Computer Science, Vol.10.
- [8]. Roy, A, Inian. 2015. *Line Following Robot Based on Vision Techniques*. International Journal of Advanced Technology in Engineering and Science, Vol. 03, Issue 02.
- [9]. Dupuis, Jean-Francois. 2013. *Evolving a Vision-Based Line-Following Robot Controller*. Canada : Universite Laval.
- [10]. Dufour, Pascal. 2008. *Intelligent Line Following for Vision Enabled Mobile Robot*. Denmark : Technical University of Denmark.

