

PERANCANGAN MOBILE ROBOT PEMINDAH BARANG DENGAN SINGLE VISION SEBAGAI SENSOR BENTUK OBJEK BERBASIS RASPBERRY PI

*Designing Of Thing Mover Mobile Robot With A Single Vision
As Object Shapes Sensor Based On Raspberry Pi*

I Putu Andika Priyatna¹, I Made Budi Suksmadana², Giri Wahyu Wiriasto³

ABSTRAK

Suatu *mobile* robot dengan sensor penginderaan (*vision*) berupa kamera biasanya ditugaskan untuk menjelajah tempat. Pemanfaatan robot tersebut pada pabrik besar akan sangat tepat, khususnya untuk memindahkan paket dari suatu tempat ke tempat lainnya. Pada perancangan ini, Robot yang dirancang ditugaskan untuk memindahkan benda sesuai bentuknya (*common-shape*) ke tempat tujuan dan setelahnya kembali lagi ke tempat semula.

Pemrosesan gambar, pengambilan keputusan sistem, dan kontrol roda robot dilakukan oleh raspberry pi (tipe 2 B), dan pengontrolan aktuator lainnya serta sensor-sensor dilakukan oleh mikrokontroler (atmega8L) melalui perintah raspberry pi. Pendeteksian bentuk benda menggunakan aturan Pythagoras dan *cosinus* yang diterapkan kedalam algoritma yang telah dibuat.

Hasil dari penelitian ini adalah robot berhasil mengenali benda dengan ciri yang telah dibuat, serta kamera berhasil mengikuti arah pergerakan benda yang dikenali. Selain itu, robot juga berhasil melakukan tugasnya dengan baik. Waktu yang dibutuhkan robot dalam menyelesaikan tugasnya untuk bentuk benda persegi selama 40,49 detik, bentuk benda segitiga sembarang selama 64,65 detik, dan bentuk benda segitiga siku-siku selama 65,05 detik.

Kata kunci: *mobile* robot, kamera, *vision*, Raspberry pi, mikrokontroler, memindahkan benda.

ABSTRACT

A mobile robot equipped with a vision sensor is usually used to explore an open environment. The use of the robot in a big-spaced factory will be goodly compatible, especially to move any packet from a place to another. In this design, a goal is that the robot must be able to move a thing according to its shape (common-shape) to a destination and after that turning back to the first place (home).

Images processing, system decisions making, and wheels controlling are handled by raspberry pi (2 B type), and other actuators control together with sensors are handled by a microcontroller (atmega8L) system via raspberry pi commands. Shapes detections use Pythagoras and cosine rules which are applied into an algorithm having made.

The results show that the robot is able to detect some shapes by features having made, and robot's camera is able to track a moving object known by its system. In addition, the goal of this project has been achieved well. Time needed for the robot to do its task or achieve the goal is for 40,5 seconds for a square-shaped thing, 64,6 seconds for a non-right angle triangle-shaped thing, and 65,0 seconds for a right angle triangle-shaped thing.

Keywords: *mobile* robot, camera, *vision*, raspberry pi, microcontroller, moving a shaped thing.

PENDAHULUAN

Mobile robot biasanya digunakan pada medan terbuka untuk melakukan tugas tertentu, seperti Robot pada pabrik yang biasanya menggunakan sensor untuk mengidentifikasi keberadaan benda. Namun untuk mengidentifikasi keberadaan dan juga bentuk suatu benda, penggunaan sensor akan sangat sulit diterapkan apalagi pada robot yang sifatnya *mobile*. Oleh karena itu penggunaan sebuah kamera (*Single Vision*) merupakan solusi dari permasalahan

tersebut.

Dalam mengidentifikasi gambar atau citra, ada banyak metode yang dapat diterapkan, seperti SURF (*Speeded Up Robust Features*), SIFT (*Scale Invariant Features Transform*), BRIEF (*Binary Robust Independent Elementary Features*), atau metode yang dapat digeneralisasi dengan persamaan matematis sederhana yang dikombinasi dengan fungsi-fungsi tertentu yang telah disediakan oleh perangkat lunak pengolah gambar; salah satunya adalah OpenCV (*Open-source Computer Vision*). Penggunaan

¹Jurusan Teknik Elektro, Fakultas Teknik Universitas Mataram, Jln Majapahit 62 Mataram - Nusa Tenggara Barat

persamaan matematis sederhana akan sangat mendukung untuk suatu perangkat pemrosesan yang terbilang memiliki tingkat kecepatan eksekusi yang cukup rendah, seperti raspberry pi. Penggunaan metode tersebut merupakan hal yang tepat untuk mengidentifikasi benda dengan bentuk dasar, seperti persegi, empat persegi panjang, segitiga siku-siku, dan segitiga sembarang. Oleh karena itu, pada perancangan ini, pendeteksian bentuk-bentuk benda tersebut menggunakan aturan Pythagoras dan cosinus untuk menetapkan ciri objek yang akan dideteksi oleh robot.

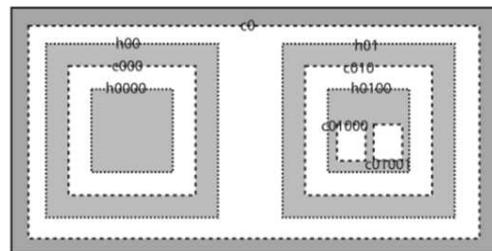
Terdapat banyak penelitian yang telah dilakukan mengenai robot dengan kamera sebagai sensor (*vision*), beberapa di antaranya adalah Astua (2012), yang menggunakan teknik berbasis ciri dengan mengaplikasikan algoritma SURF (*Speeded Up Robust Features*), Herceg (2011), yang melakukan penelitian mengenai pendeteksian objek sekitar dengan menerapkan bidang alir optik citra, Damaryam (2015), yang di dalam penelitiannya menerapkan transformasi *Hough* dalam mengidentifikasi garis, dan Borse (2012), dalam penelitiannya menggunakan kombinasi antara kamera dan sensor jarak berupa infrared. Dalam penelitian mereka tersebut, pengklasifikasian suatu ciri citra didasarkan pada pendeteksi tepi dan juga kontur citra yang kemudian diolah oleh algoritma tertentu untuk mendapatkan informasi dari suatu citra yang diolah. Dari penelitian-penelitian tersebut dapat ditarik kesimpulan bahwa pada robot yang sifatnya *mobile*, kecepatan eksekusi citra (*frame-rate*) akan mempengaruhi kestabilan robot dalam bernavigasi / bermanuver.

Raspberry pi merupakan komputer berukuran mini yang dilengkapi dengan beberapa peripheral khusus seperti USB, HDMI, slot kamera paralel, serta GPIO yang dapat digunakan untuk antarmuka dengan perangkat keras lain, seperti mikrokontroler, atau sejenisnya.

Mikrokontroler merupakan perangkat keras elektronika yang digunakan untuk pengontrolan alat-alat yang tidak memerlukan tingkat pemrosesan tinggi, seperti sensor cahaya, pengontrolan motor servo, dan motor stepper. Ia telah dilengkapi dengan *peripheral* berupa ADC (*Analog-to-Digital Converter*), PWM (*Pulse Width Modulation*), USART (*Universal Synchronous and Asynchronous serial Receiver Transmitter*), dan lainnya. (Atmel, 2013)

kontur merupakan sekumpulan titik yang

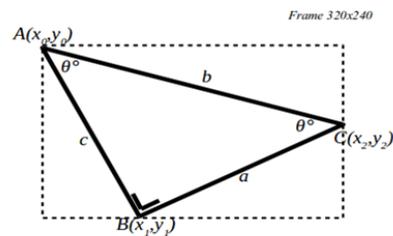
terhubung satu sama lain, dan terletak pada bagan objek. Citra bagian bawah gambar 1 menggambarkan citra yang sama dengan kontur teralokasi. Kontur-kontur tersebut berlabel cX atau hX , dimana "c" merupakan singkatan dari "contour", "h" merupakan singkatan dari "hole", dan "X" merupakan angka. Beberapa dari kontur-kontur tersebut berupa *dash line* (garis hubung panjang); garis ini merepresentasikan batasan dari area putih (yaitu, daerah bukan nol). OpenCV dan *findContours()* membedakan antara batasan eksterior tersebut dan garis dari titik-titik yang terhubung (*dotted lines*), yang dapat dianggap sebagai batasan interior atau batasan eksterior dari *holes* (yaitu daerah nol).



Gambar 1. Citra uji kontur (Bradski, 2008)

Bentuk Dasar di dalam *Frame* citra.

Penentuan bentuk citra dalam satu *frame* didasarkan atas parameter yang digunakan. Misalkan pendeteksian bentuk segitiga siku-siku, dapat dilihat pada gambar 2.



Gambar 2. Segitiga siku-siku di dalam *frame*

Pada gambar 2, misalkan diambil dua titik koordinat yaitu $A(x_0, y_0)$ dan $B(x_1, y_1)$ yang merupakan panjang c maka garis tebal yang menghubungkan kedua titik tersebut merupakan *hypotenuse* dari segitiga dengan garis putus-putus sebelah kiri tanda c . Panjang *hypotenuse* ini ditentukan dengan persamaan 5, begitu pula dengan panjang kedua *hypotenuse* a dan b . Untuk menentukan besar sudut A , B , dan C , digunakan persamaan 2.6.

Hubungan antara radian dan derajat dapat dituliskan dalam persamaan berikut: anggaps adalah satu putaran penuh, maka:

$$s = 2\pi r \dots\dots\dots(1)$$

dengan r adalah panjang jari-jari yang dinyatakan dalam bentuk persamaan:

$$r = \frac{s}{\theta \text{ radians}} \dots\dots\dots(2)$$

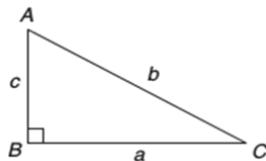
maka:

$$\theta = \frac{s}{r} = \frac{2\pi r}{r} = 2\pi \dots\dots\dots(3)$$

dalam satu rotasi penuh, $\theta = 360^\circ$, maka hubungan antara derajat dan radian:

$$360^\circ = 2\pi \text{ radians atau } 180^\circ = \pi \text{ rad}$$

$$1 \text{ rad} = \frac{180^\circ}{\pi} \approx 57.30^\circ$$



Gambar 3. Segitiga siku-siku (Bird, 2010)

Pada gambar 3, panjang a disebut *adjacent*, b disebut *hypotenuse*, dan c disebut *opposite*. Untuk mencari masing-masing panjang a , begitu juga b , dan c dapat digunakan persamaan pythagoras:

$$a = \sqrt{b^2 - c^2} \dots\dots\dots(4)$$

dan besar sudut A , begitu juga B , dan C dapat dicari dengan menggunakan persamaan kosinus:

$$A = \cos^{-1} \left(\frac{b^2 + c^2 - a^2}{2 \cdot bc} \right) \dots\dots\dots(5)$$

sedangkan untuk luasan segitiga dapat dicari dengan persamaan berikut:

$$\text{Area} = \sqrt{[s(s - a)(s - b)(s - c)]} \dots\dots\dots(6)$$

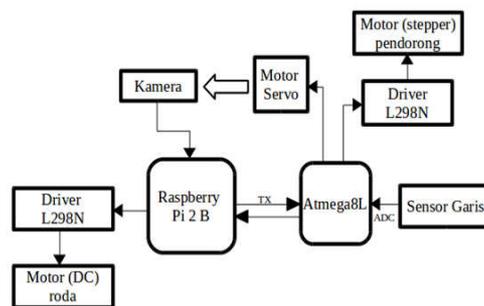
$$s = \frac{a+b+c}{2}$$

METODE PERANCANGAN

Perancangan ini berfokus pada *mobile* robot dengan kamera sebagai alat pengindraan berupa pendeteksian suatu objek dengan bentuk dasar (*common shape*). Metode yang digunakan adalah deteksi kontur pada citra dan membuatnya tertutup (*closed contours*). Dari kontur tertutup tersebut dapat dicari jumlah vertise, pajang garis, dan besar sudutnya. Pada bagian kamera robot terdapat dua motor servo dengan Atmega8L sebagai media kendali rotasi dari raspberry pi yang digunakan untuk menggerakkan kamera, satu servo difungsikan untuk rotasi vertikal dan yang satunya lagi untuk rotasi horizontal sehingga memungkinkan robot mencari titik pusat objek secara lebih efisien. Oleh karena

robot yang akan dirancang bersifat *mobile*, maka diperlukan alat gerak berupa dua buah roda yang masing-masing terkopel dengan motor DC gearbox, dan satu buah roda sebagai penyeimbang, *driver* motor yang digunakan adalah IC L298N yang dapat mengatur arah putaran, dan kecepatan rotasi motor melalui sinyal PWM yang didapat dari raspberry pi. Agar dapat meletakkan benda yang telah diberikan pada bagian depan robot, digunakan motor stepper dengan kendali driver IC L298N dari mikrokontroler. Motor ini terkopel dengan *worm-gear* yang menghasilkan gerakan translasi untuk meletakkan benda dengan cara mendorongnya.

Pada blok diagram gambar 3, Raspberry pi 2 B digunakan sebagai sistem utama yang memegang kendali atas semua sistem yang berjalan, dan atmega8L merupakan sistem tambahan yang mengontrol motor servo, motor stepper, dan menerima masukan sensor garis yang dikendalikan oleh sistem utama. Tujuan dari penggunaan sistem tambahan ini adalah pertama, untuk kesetabilan motor servo melalui pulsa PWM yang dihasilkannya, Raspberry pi 2 B, sekalipun merupakan sistem *multitasking*, namun jika pengontrolan motor servo dilakukan secara bersamaan dengan perangkat lunak pengolah gambar, maka pulsa PWM yang dihasilkan tidak akan stabil, dikarenakan pemrosesan gambar memerlukan tingkat eksekusi yang cukup tinggi. Kedua, sistem raspberry pi 2 B tidak mendukung ADC (*Analog to Digital Converter*) sehingga masalah ini dapat ditangani oleh sistem atmega8L yang memiliki fitur tersebut.

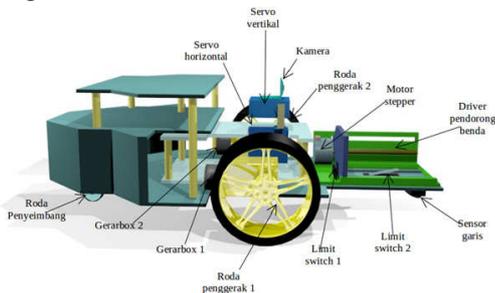


Gambar 4. Blok diagram sistem

Masukan sistem utama berupa citra inputan yang diterima dari modul kamera RaspICam, dan keluaran sistem berupa kontrol motor DC untuk roda robot serta sinyal PWM untuk kontrol kecepatan motor DC tersebut melalui

driver motor. Selain itu, keluaran sistem utama berupa antar-muka komunikasi serial satu arah dengan protokol *USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter)*. untuk mengontrol sistem tambahan; atmega8L. Sedangkan masukan sistem tambahan atmega8L berupa data sensor garis, dan keluaran sistem berupa sinyal PWM untuk mengontrol rotasi servo, mengontrol putaran motor stepper melalui driver motor tersebut, serta antar-muka komunikasi melalui bit-bit I/O port yang dimilikinya ke I/O port raspberry pi 2 B atau dikenal dengan istilah *GPIO (General Input-Output)*. Terdapat tanda panah besar di antara motor servo dan kamera yang menunjukkan keterkaitan antara peran servo dan peran kamera, yaitu ketika kamera mendeteksi benda yang secara sengaja poisisi benda diubah-ubah maka motor servo sebagai aktuator kamera akan mengtrek – mengarahkan posisi kamera – ke objek tersebut.

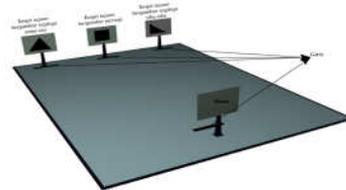
Mekanika Robot. Mekanika robot terlihat pada gambar 5. Robot yang dirancang miliki 3 pasang roda; satu roda untuk penyeimbang dan dua roda untuk menggerakkan robot serta mengarahkan robot.



Gambar 5. Desain bodi robot

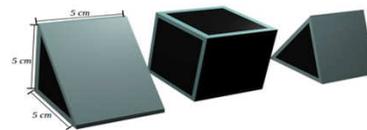
Robot juga dilengkapi dengan kamera dan satu sensor garis. Kamera berfungsi sebagai alat indra robot, dan sensor garis berfungsi untuk mendeteksi keberadaan garis didepannya. Pada kamera terdapat dua motor servo yang digunakan sebagai aktuator dari kamera, satu motor servo mewakili gerak vertikal, dan yang satunya lagi mewakili arah horizontal, jadi memungkinkan kamera untuk bergerak mengikuti objek dengan lebih fleksibel. Gambar 5 juga memperlihatkan adanya aktuator untuk pendorong benda. Aktuator tersebut tepat berada ditengah-tengah bagian depan robot. Motor pada aktuator yang digunakan berupa motor stepper yang telah dilengkapi dengan *worm-gear*.

Perancangan Arena Robot. Arena robot yang akan dirancang terlihat pada gambar 6. Pada arena tersebut terdapat empat buah gambar yang terpasang, yaitu gambar segitiga siku-siku, segitiga sembarang, persegi, dan empat persegi panjang. Khusus untuk empat persegi panjang akan digunakan sebagai tempat awal robot bernavigasi (*home*), dan bentuk lainnya dijadikan sebagai tempat tujuan robot menaruh benda yang diberikan.



Gambar 6. Skema arena robot

Perancangan Benda (Objek yang Dipindahkan). Gambar 7 memperlihatkan tiga macam bentuk benda yang dikenali oleh sistem robot.



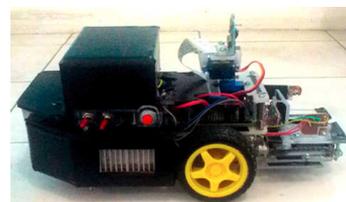
Gambar 7. Tiga bentuk benda yang dikenal sistem

Ketiga benda ini secara sengaja diberi warna gelap (hitam) pada bagian tengahnya (yang menggambarkan bentuk benda) agar proses pendeteksian tepi dapat lebih cepat dan akurat diproses.

HASIL DAN PEMBAHASAN

Mobile robot yang dirancang tersusun atas perangkat keras elektronika maupun mekatronika. Keseluruhan tampak luar robot terlihat pada gambar 8.

Komponen penyusun robot gambar 8 dapat dilihat secara lebih detail pada desain robot gambar 5.



Gambar 8. Tampilan tampak luar robot

Pengujian Nilai Sensor Garis. pengujian nilai sensor garis bertujuan untuk mencari nilai batasan yang ditentukan sebagai indikasi adanya garis yang terdeteksi. Pengujian yang dilakukan menggunakan fitur ADC pada

mikrokontroler atmega8L dengan resolusi 10-bit yang artinya rentang nilai tegangan referensi sebesar 0 V hingga 5 V adalah 0 – 1023 desimal 10-bit ADC. Penentuan nilai batasan sensor garis dilakukan dengan meletakkan robot pada lantai hitam (garis) dan putih, lalu mengamati nilai yang terbaca oleh sensor. Pengambilan sampel dilakukan sebanyak 20 kali untuk kedua warna lantai (hitam dan putih).

Tabel 1. Data pengujian nilai batasan sensor garis

Warna Lantai	Pengambilan Data ke-	Nilai ADC Sensor garis	Warna Lantai	Pengambilan Data ke-	Nilai ADC Sensor garis
putih	1	835	hitam (garis)	1	529
	2	825		2	531
	3	828		3	531
	4	833		4	531
	5	824		5	531
	6	825		6	531
	7	824		7	531
	8	814		8	531
	9	822		9	531
	10	820		10	531
	11	819		11	531
	12	825		12	531
	13	824		13	531
	14	824		14	531
	15	824		15	531
	16	824		16	529
	17	827		17	531
	18	830		18	529
	19	831		19	529
	20	831		20	531
Rata-rata		825.45	Rata-rata		530.6

Setelah ke-duapuluh sampel data didapat dari kedua warna lantai, kemudian dicari nilai rata-ratanya. Dua nilai rata-rata dari warna lantai putih dan hitam (garis) ditentukan titik tengahnya dengan cara:

$$Titiktengah = \frac{825,45+530,6}{2} = 678,025 \dots\dots\dots(7)$$

nilai 678,025 inilah yang kemudian dijadikan batasan nilai (titik tengah) untuk warna hitam (garis), yang mana jika nilai ADC terbaca lebih besar dari titik tengah, maka hal tersebut merupakan indikasi adanya garis.

Pengujian Sudut Motor Servo terhadap nilai PWM. Pengujian data motor servo dilakukan dengan alat bantu busur derajat. Berdasarkan data hasil tersebut didapat nilai OCRnx. Pengujian sudut dilakukan dengan melihat nilai OCRnx yang didapat pada setiap 10° data sampel. Data-data sampel ini kemudian digunakan untuk mencari persamaan perhitungan besaran sudut terhadap nilai OCRnx PFCPWM dengan menggunakan regresi linier.

Motor servo TowerPro SG90 memiliki spesifikasi rotasi putaran dari 0° hingga 180°. Dua motor servo pada robot dikontrol oleh atmega8L melalui raspberry pi. Mode PWM yang digunakan adalah PFCPWM 16-bit. Jadi dari hasil pengujian sudut motor servo didapat rentang nilai OCR dari 623 hingga 2670 untuk rotasi dari 0° hingga 180° berturut-turut. Untuk mendapati besaran sudut terhadap nilai register OCR menggunakan metode regresi linier.

$$var_1 = \frac{\sum y_i(x_i - \bar{x})}{\sum x_i(x_i - \bar{x})}$$

$$var_2 = \bar{y} - \bar{x} \cdot var_1$$

dengan:

y_i = sampel OCRnx

x_i = sampel sudut

\bar{x} = rata-rata sampel sudut

\bar{y} = rata-rata sampel OCRnx

tabel 2 memperlihatkan banyaknya sampel nilai sudut dan nilai OCRnx. Nilai-nilai tersebut kemudian disubstitusikan ke dalam persamaan 9a dan 9b yang kemudian menghasilkan persamaan 10 yang digunakan sebagai persamaan rotasi motor servo terhadap besarnya nilai OCRnx.

$$OCRnx = var_1 * SdtPutar + var_2$$

$$OCRnx = 11.17474 * SdtPutar + 580.01$$

Tabel 2. Sampel besar sudut terhadap OCRnx

Sampe ke-	Besar Sudut (°)	nilai register OCRnx
1	0	623
2	10	723
3	20	812
4	30	924
5	40	1046
6	50	1135
7	60	1246
8	70	1335
9	80	1458
10	90	1547
11	100	1658
12	110	1769
13	120	1891
14	130	2003
15	140	2125
16	150	2247
17	160	2403
18	170	2514
19	180	2670

Bentuk Benda Terdeteksi. Pengujian ini dilakukan untuk mengetahui perubahan frame-rate terhadap masing-masing bentuk benda. Masing-masing pengujian bentuk dilakukan secara realtime dengan jarak objek dari kamera berkisar 30 cm. Pengujian

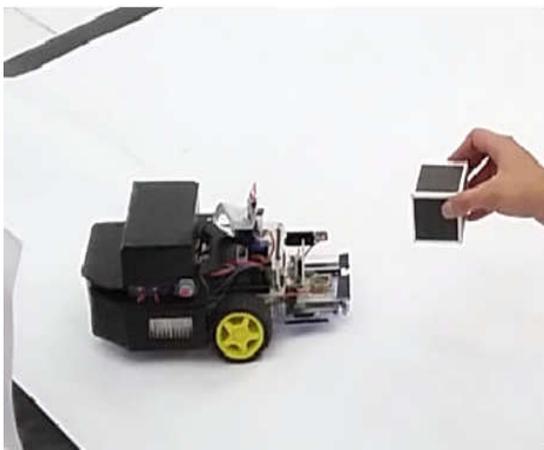
dilakukan sebanyak lima kali percobaan pada masing-masing bentuk objek (segitiga siku-siku, segitiga sembarang, persegi, dan empat persegi panjang). Tabel 3 memperlihatkan hasil pengambilan data *frame-rate* terhadap bentuk benda.

Tabel 3. Besar *frame-rate* terhadap bentuk benda

Pengujian ke-1	<i>Frame-rate</i> (fps)			
	seg. Siku-siku	seg. Sembarang	persegi	4 persegi pjg
1	29.9	27.3	20.6	38.3
2	21.4	38.7	29.1	30.7
3	27.4	26.2	28.5	37.7
4	33.7	35.6	20.8	33.8
5	28.1	23.6	37.7	20.34
Rata-rata	28.1	30.28	27.34	32.048

Masing-masing *frame-rate* yang dihasilkan dari masing-masing objek berbeda setiap perputaran proses yang terjadi. *Frame-rate* yang dihasilkan pada keempat bentuk objek cukup bervariasi yang dapat diakibatkan karena setiap bentuk yang didapat dari inputan kamera tidak sama persis posisi piksel dari objek, dan intensitasi cahaya yang dipantulkan objek.

Waktu yang Dibutuhkan Robot dalam Memindahkan Benda ke Tempat Tujuan hingga Kembali ke Posisi Semula (Home). Pengujian dilakukan sebanyak sepuluh kali percobaan. Pada pengujian yang telah dilakukan robot dapat memindahkan benda dengan baik ke tempat tujuan dengan waktu tempuh robot ke tempat tujuan yang bervariasi. Tujuan dari pengujian ini adalah untuk mengetahui waktu yang dibutuhkan robot dalam bermanuver ke tempat tujuan untuk meletakkan benda dan waktu dari tempat tujuan menuju posisi semula (*home*).



Gambar 9. Robot diperlihatkan objek berlambang berlambang persegi (berbentuk kubus)



Gambar 10. Objek diletakan pada bagian depan robot



Gambar 11. Robot bernavigasi ke-tempat tujuan



Gambar 12. Robot berbalik arah untuk mendeteksi *home*



Gambar 13. Robot kembali ke *home* dengan posisi seperti semula

Berikut adalah langkah-langkah yang dilakukan:

- a) **Robot diperlihatkan salah satu bentuk benda yang telah dikenali sistem:** Pada posisi *home*, robot diperlihatkan salah satu bentuk benda, sebagai contoh bentuk persegi yang terlihat pada gambar 9. Pada tahap ini kamera akan menyesuaikan posisinya dengan posisi objek yang diperlihatkan.
- b) **Objek diletakan di bagian depan robot (tempat khusus peletakan benda):** Setelah objek diletakan maka kamera akan mengarah ke tempat di mana terdapat gambar yang sama dengan bentuk objek yang diberikan (tempat tujuan). Setelah itu robot akan menyesuaikan posisinya dengan posisi kamera sehingga arah depan robot dengan kamera sama.
- c) **Robot bernavigasi ke tempat tujuan:** Saat posisi kamera robot dan arah depan robot sama, robot menuju tempat tujuan bersama dengan benda yang diberikan tadi seperti yang terlihat pada gambar 10.
- d) **Robot berbalik arah untuk mendeteksi *home* (gambar empat persegi panjang):** Setelah robot meletakan benda, kemudian robot berbalik arah untuk mendeteksi *home* dengan lambang gambar empat persegi panjang seperti yang diperlihatkan pada gambar 11. Robot akan berhenti ketika bagian depan robot mencapai garis yang berada di bawah gambar empat persegi panjang (*home*), kemudian robot mundur sesaat dan memutar arah ke kiri dengan tujuan untuk mendapati posisi semula, terlihat pada gambar 12.
- e) **Robot kembali ke *home* dengan posisi seperti semula:** Tahap ini merupakan tahap terakhir, dimana robot berhenti dan menyesuaikan posisi seperti semula. Robot akan tetap diam seperti yang terlihat pada gambar 12 sampai satu bentuk benda (salah satu bentuk benda yang dikenali sistem) diperlihatkan kembali. Sebagai contoh ketika benda berbentuk segitiga siku-siku diperlihatkan dan diberikan kepadanya, maka ia akan meletakan benda tersebut pada tempat yang memiliki gambar segitiga siku-siku, dan kembali ke *home* dengan metode yang sama seperti objek persegi tadi.

Pencatatan Waktu

- 1) Waktu pertama (menuju tempat tujuan) dimulai ketika benda diletakan pada

bagian depan robot hingga robot mencapai tempat tujuan dan meletakan benda di tempat tersebut.

- 2) Waktu kedua (menuju *home*) dimulai ketika robot telah meletakan benda hingga robot mencapai *home* dan memutar arah agar sesuai dengan posisi semula.

Waktu yang dibutuhkan robot dalam sepuluh kali pengujian terangkum dalam tabel 4.

Table 4. Pengujian robot dalam bermanuver terhadap waktu yang dibutuhkan

Pengujian ke-	Waktu yang dibutuhkan robot (detik)					
	A1	A2	B1	B2	C1	C2
1	65.10	25.30	17.40	19.30	57.40	25.40
2	40.49	23.36	16.24	22.24	42.45	19.73
3	37.08	17.72	22.77	21.81	43.45	54.53
4	31.10	21.77	24.37	15.45	29.24	27.40
5	68.06	21.37	16.48	19.80	32.62	30.62
6	35.89	22.46	17.67	26.24	30.31	25.50
7	26.58	21.45	29.39	21.02	38.20	19.14
8	34.92	32.72	14.36	18.61	23.36	31.65
9	29.85	45.78	23.56	24.01	39.70	19.84
10	25.99	19.20	14.37	19.80	26.89	33.02
Rata-rata	39.51	25.11	19.66	20.83	36.36	28.68
Total	64.62		40.49		65.05	

Keterangan nama-nama kolom pada tabel 4:

A1: Robot memindahkan barang dari posisi awal (*home*) menuju posisi tujuan yang terdapat gambar segitiga sembarang.

A2 : Robot kembali menuju posisi awal (*home*) dari posisi tujuan yang terdapat gambar segitiga sembarang.

B1: Robot memindahkan barang dari posisi awal (*home*) menuju posisi tujuan yang terdapat gambar persegi.

B2 : Robot kembali menuju posisi awal (*home*) dari posisi tujuan yang terdapat gambar persegi.

C1: Robot memindahkan barang dari posisi awal (*home*) menuju posisi tujuan yang terdapat gambar segitiga siku-siku.

C2 : Robot kembali menuju posisi awal (*home*) dari posisi tujuan yang terdapat gambar segitiga siku-siku.

Waktu tercepat dalam memindahkan benda dari posisi *home* ke tempat tujuan adalah bentuk persegi, dengan waktu rata-rata 19,66 detik. Hal ini dikarenakan bentuk persegi berhadapan langsung dengan posisi awal robot (*home*) sehingga pada saat bernavigasi menuju bentuk persegi, robot hanya perlu sedikit menyesuaikan arahnya. Selain itu, benda dengan bentuk segitiga siku-siku dan segitiga sembarang memiliki kemiringan

tertentu terhadap posisi *home*, sehingga kalkulasi oleh sistem robot dalam mendeteksi bentuk agak sedikit lebih lama dilakukan.

KESIMPULAN

Dari proses pembuatan *mobile* robot pemindah barang yang dilengkapi dengan *single vision*, mulai dari proses perancangan perangkat keras, perangkat lunak, hingga pengambilan data, dapat ditarik beberapa kesimpulan sebagai berikut:

- 1) Sistem robot yang dibuat telah dapat mengenali benda berbentuk segitiga siku-siku, segitiga sembarang, dan persegi. Selain itu, robot juga telah dapat bernavigasi memindahkan masing-masing benda dengan salah satu dari ketiga bentuk tersebut menuju tempat tujuan (tempat dengan gambar yang sama dengan bentuk benda yang diberikan) dan kembali lagi ke posisi awal (*home*) dengan indikasi gambar empat persegi panjang.
- 2) Kamera pada robot telah dapat mengikuti arah benda yang dikenali secara horizontal maupun vertikal dengan mengimplementasi aturan Pythagoras dan *cosinus* dalam suatu algoritma untuk menentukan ciri dari masing-masing bentuk benda. Pergerakan kamera dilakukan oleh dua buah motor servo yang dikendalikan oleh sistem tambahan berupa sistem minimum atmega8L melalui perintah dari raspberry pi.
- 3) Waktu yang dibutuhkan robot dalam bernavigasi memindahkan benda berbentuk persegi dari posisi awal (*home*) menuju tempat tujuan, dan dari tempat tujuan kembali ke posisi awal adalah selama 40,49 detik, benda dengan bentuk segitiga sembarang selama 64,65 detik, dan benda dengan bentuk segitiga siku-siku selama 65,05 detik.
- 4) Setiap bentuk benda yang dikenali robot mempengaruhi kecepatan *frame-rate*, yang pada perancangan ini *frame-rate* tertinggi adalah pendeteksian bentuk persegi dengan kecepatan 27,34 fps. Hal ini dipengaruhi oleh jumlah ciri serta kalkulasi yang dilakukan oleh sistem untuk mendapati ciri tersebut.

SARAN

Saran untuk pengembangan berikutnya adalah:

1. Bentuk benda dasar (*Common shape*) yang dikenali sistem dapat ditambahkan sehingga robot dapat mendeteksi lebih

banyak bentuk dasar.

2. Menggantikan *single vision* dengan *stereo vision* (menggunakan dua kamera) sehingga robot selain dapat mengenali bentuk benda yang terdeteksi, ia juga dapat mengestimasi jarak benda dari kamera.

DAFTAR PUSTAKA

- Astua, Carlos., Barber, Ramon., Crespo, Jonathan., and Jardon, Alberto, 2014, *Object Detection Technique Applied on Mobile Robot Semantic Navigation*, p. 6734-6757.
- Atmel., 2013, *8-bit Atmel with 8KBytes In-System Programmable Flash ATmega8L*, USA.
- BCS., 2012, *The Raspberry Pi Education Manual Version 1.0*.
- Bird, John., 2010, *Basic Engineering Mathematics Fifth Edition*, Elsevier Ltd., USA.
- Bradski, Gary., Kaehler, Andrian., 2008, *Learning OpenCV*, O'Reilly Media, Inc., Sebastopol.
- Borse, Himanshu., Dumbare, Amol., Gaikwad, Rohit., and Lende, Nikhi, 2012, *Mobile Robot for Object Detection Using Image Processing*, Global Journals Inc. Vol. 12.
- Damaryam, GK., 2015, *A Hough Transform Implementation for Line Detection for a Mobile Robot Self-Navigation System*, Journal of Computer Engineering, IOSR-JCE, Vol. 17, p. 33-34.
- Herceg, Domagoj., Markovic, Ivan., and Petrovic, Ivan, 2011, *Real-Time Detection of Moving Objects by a Mobile Robot with an Omnidirectional Camera*.



I Putu Andika Priyatna, lahir di Golong tanggal 11 April 1994, menempuh pendidikan program Strata 1 (S1) di Fakultas Teknik Universitas Mataram sejak Tahun 2012.