

PENGENDALIAN *MOBILE ROBOT VISION* MENGGUNAKAN WEBCAM PADA OBJEK ARAH PANAH BERBASIS RASPBERRY PI

Kukuh Darmawan Setyanto

kukuhdarmawan.s@gmail.com
Universitas Jember

Ike Fibriani, S.T, M.T.

ik3fibriani.teknik@gmail.com
Universitas Jember

Sumardi, S.T., M.T.

smardi10@gmail.com
Universitas Jember

Abstrak

Penggunaan Raspberry pi sebagai pengganti perangkat CPU pada teknologi robot vision merupakan solusi alternatif yang dapat dilakukan untuk membuat robot vision menjadi lebih mobile. Pada penelitian ini *mobile robot vision* diintegrasikan dengan perangkat webcam yang berguna sebagai *input* penangkap objek untuk selanjutnya dilakukan proses pengolahan citra. Metode pengolahan citra yang digunakan pada penelitian ini adalah metode *template matching*, dimana metode ini bekerja berdasarkan membandingkan citra masukan dengan citra *template* yang tersimpan pada program robot. Penggunaan metode tersebut pada penelitian ini bertujuan agar robot dapat melakukan kendali gerak berdasarkan informasi gambar yang diperoleh oleh kamera *webcam*. Gambar yang digunakan mewakili 4 kondisi gerak robot yaitu maju, berhenti, belok kanan dan belok kiri. Dari penjelasan diatas didapatkan hasil pengujian program pecocokan gambar dengan nilai *black pixel* ketika *match* berkisar antara 768 hingga 4119. Kemudian Pengujian robot terhadap lintasan memerlukan waktu rata-rata sebesar 93,8 detik serta kemampuan kamera terhadap pengaruh jarak dan intensitas cahaya berkisar antara 20 cm hingga 90 cm dan 0,8 lumens/m² hingga 91500 lumens/m².

Kata Kunci — *mobile robot vision, raspberry pi, template matching, webcam*

Abstract

Use of Raspberry pi as of replacement for CPU on robot vision technology is an choice solution for make the robot vision become more mobile. In this study, a mobile robot vision integrated with a webcam device because mobile robot need an input for catcher image. Image processing method used in this study is a *template matching*, where in the method works by comparing the input image with the image template which stored in the robot program. Use of *template matching* methods robot can perform motion control based on the information of the images obtained by the camera *webcam*. There are four image template as representative conditions of robot motions, there are motion forward motion, stop motion, turn right motion and turn left motion. The results of the above explanation that when the image condition is match, image have a *black pixel* value in ranged from 768 to 4119. And testing the robot on the track requires an average time of 93,8 seconds and the camera's ability to influence the distance and the light intensity ranging from 20 cm to 90 cm and 0.8 lumens/m² up to 91500 lumens/m².

key word — *mobile robot vision, raspberry pi, template matching, webcam*

I. PENDAHULUAN

Robot Vision menggunakan teknologi *image processing* dan *computer vision* dalam tugasnya, baik untuk pengenalan objek maupun kendali gerak robot. Pada implementasinya, robot vision memerlukan sebuah perangkat *input/output* seperti CPU (*Central Processing Unit*) yang dapat memproses objek berupa citra digital menjadi sebuah output berupa kendali gerak robot. Jadi dalam penggunaannya robot masih tergantung pada perangkat CPU sehingga robot tidak bersifat *mobile* dan kemampuan robot akan terbatas. Sehingga mengganti perangkat CPU dengan perangkat *mini computer* raspberry pi adalah solusi alternatif untuk memperbaiki kinerja robot menjadi lebih baik.

Solusi dari permasalahan diatas adalah dengan memanfaatkan suatu *single board computer*. Pada perkembangannya ada beberapa merek *single board computer*, salah satunya adalah Raspberry Pi. Saat ini Raspberry Pi banyak digunakan sebagai pengganti fungsi CPU, dan diterapkan pula pada perkembangan *robot vision* dan *computer vision*. Seperti pada penelitian sebelumnya yang berjudul "Robot Line Follower Berbasis Raspberry Pi Dengan Sensor Kamera" menjelaskan tentang penggunaan Raspberry Pi pada *Robot Line Follower* sebagai pengontrol gerakan dan strategi pada robot line follower.[1] dan juga pada penelitian lainnya yang berjudul "Aplikasi *Tracking Object* pada Sistem Web *Streaming* dengan Protokol TCP/IP sebagai Sistem Navigasi Mobile Robot Berbasis Mini PC" [2].

Metode pengolahan citra yang digunakan pada penelitian ini adalah metode pengenalan pola yang bekerja berdasarkan pencocokan antara citra sampel dan citra streaming atau yang biasa dikenal *template matching* [3]. Dengan metode *template matching* kamera robot dapat membaca informasi yang ditunjukkan oleh gambar arah panah sehingga dari pembacaan tersebut, robot dapat melakukan kendali gerak berdasarkan informasi gambar yang sesuai dengan kondisi dasar gerakan robot yaitu maju, berhenti, belok kanan dan belok kiri.

Sama halnya pada pengutipan penelitian sebelumnya, pada pengutipan penelitian selanjutnya sistem navigasi robot digerakkan berdasarkan teknik *image processing*, yaitu

dengan deteksi pergerakan bola mata menggunakan teknik *canny hough transform* [4].

Dengan menelaah pengutipan sebelumnya, dapat diketahui robot telah memiliki kemampuan visual yang dapat mengatur kondisi gerakan robot yang diinginkan dan robot telah memiliki kemampuan secara kendali otomatis/mobile berdasarkan prinsip *image processing*.

Pada penelitian ini tetap akan mengembangkan penelitian yang mengacu pada pengutipan sebelumnya, dengan merubah metode *image processing* yang digunakan dengan tujuan meningkat akurasi pengolahan citra dan bertujuan membuat *prototype* robot yang dapat bergerak lebih *mobile* layaknya sebuah mobil yang dapat berjalan tanpa kendali manusia.

Untuk penjelasan lebih detail mengenai penelitian ini akan dijelaskan pada bab selanjutnya yaitu tentang alur pengerjaan penelitian pada bab 2, pembahasan dan analisis data pengujian dijelaskan pada bab 3, kemudian pada bab 4 akan dilakukan penjelasan mengenai kesimpulan dari penelitian ini.

II. METODELOGI PENELITIAN

Perancangan alat yang dibangun meliputi susunan perangkat keras yang terbagi kedalam 3 bagian. Pada perangkat *input* digunakan kamera usb *webcam* dengan spesifikasi Logitech HD Webcam C525 Portable HD 720p, dan usb *wireless* dengan spesifikasi TL-WN722N 150 mbps, lalu pada perangkat *processing input/output* terdapat raspberry pi b+ dan pada perangkat *output* terdapat *driver* motor L298N serta 4 buah motor dc. lalu pada sumber tegangan robot menggunakan *power bank* 7200 mAh. Pada gambar 1 menunjukkan *mobile robot vision* yang digunakan dalam penelitian ini.

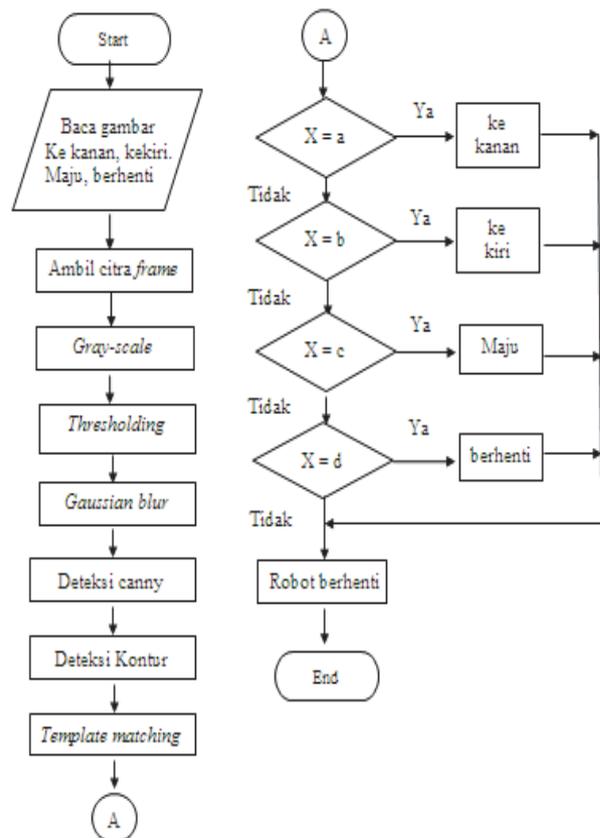
Sistem *image processing* yang digunakan pada penelitian ini adalah pengembangan dari peneliti yang meneliti tentang "*sign reading with opencv*". Sistem tersebut dibangun menggunakan *library* OpenCV [5]. *library* OpenCV berfungsi sebagai perangkat lunak *computer vision* yang bertugas sebagai program pengolah citra pada raspberry pi.



Gbr. 1 Mobile robot Vision

Desain program penelitian ini dimulai dengan mengambil (akuisisi) citra *frame* pada saat proses *streaming* yang didapat dari gambar asli dari kamera saat kondisi *realtime*. citra *frame* tersebut digunakan sebagai *input* untuk kemudian dilakukan proses pengolahan citra yang meliputi, *gray-scale*, *thresholding*, penghalusan citra, deteksi *canny*, deteksi kontur dan metode *template matching*. Pengolahan citra tersebut diolah hingga dilakukan pencocokan *template* antara gambar referensi dengan citra *frame* yang ditangkap kamera, hasil dari proses pencocokan tersebut diperoleh data hingga dilakukan keputusan untuk menentukan kondisi gerak robot. alur atau tahapan *mobile robot vision* dalam bekerja ditunjukkan pada Gambar 2.

Seperti pada gambar 2, proses pengolah citra dimulai dari inialisasi pembacaan gambar referensi yang mewakili empat kondisi yaitu maju, berhenti, ke kanan dan ke kiri. Dan dilanjutkan pada proses pengolah citra akhir yaitu *template matching*. Apabila pada proses *matching* terdeteksi adanya citra yang menyerupai dengan objek a,b,c,d maka program akan menampilkan kondisi *match* pada window dan berlanjut pada pengaktifan pin *gpio* raspberry pi yang menggunakan *library* *bcm2835*. Pin *gpio* yang telah aktif akan memberikan pulsa tegangan sebesar 5V untuk mengaktifkan driver motor untuk berjalan sesuai dengan logika yang telah diatur sebelumnya yaitu sesuai dengan kondisi citra *template* yang dimaksud. Dimana kondisi a = ke kiri, kondisi b = ke kanan, kondisi c = maju dan kondisi d = berhenti.



Gbr. 2 Flowchart program

III. HASIL DAN PEMBAHASAN

Pada pembahasan ini dijelaskan tentang pengujian secara alat maupun program yang telah dibuat sebagai tolak ukur keberhasilan penelitian ini. pengujian alat dan program merupakan pengujian tingkat keberhasilan dari sistem mekanik dan program yang diuji dengan pengaruh perubahan nilai parameter uji seperti *matching*, *lumen*, *thresholding*, dan jarak. *Output* yang didapatkan dari pengaruh parameter tersebut adalah respon robot terhadap citra yang telah diolah menggunakan teknik pengolahan citra. Respon tersebut menghasilkan data berupa kendali gerak *mobile robot vision*, perhitungan *black pixel* dan *delay* pengolahan citra.

A. Pengujian program pengolahan citra

Pada pengujian ini, program akan di eksekusi untuk mengetahui keberhasilan program pengolahan citra yang sudah dibuat. Maka untuk mengetahui keberhasilan program tersebut, akan dilakukan pencocokan gambar dengan gambar referensi sama maupun yang berbeda.

TABEL I
HASIL PENGUJIAN PROGRAM PENGOLAHAN CITRA

No	Gambar referensi yang disimpan dalam bentuk file JPEG.	Output window B menggunakan an deteksi canny	Output window A dan C	Hasil Pencocokan Gambar
1				terdeteksi
2				tidak terdeteksi
3				terdeteksi
4				terdeteksi
5				tidak terdeteksi
6				terdeteksi
7				tidak terdeteksi
8				terdeteksi

Dari 8 percobaan yang dilakukan membuktikan bahwa algoritma program berjalan dengan baik. Hasil tersebut dapat diketahui pada tabel 1, dimana 4 percobaan berhasil dan 4 percobaan gagal dengan pencocokan *template* yang tepat. Proses *matching* gambar yang dilakukan dengan gambar referensi yang tersimpan dan bukan gambar referensi. Sehingga pada percobaan nomor 2 dan 7 terjadi kegagalan sedangkan nomor 5 dan 8 gambar gagal untuk dilakukan pencocokan karena perspektif gambar terlalu berubah.

B. Pengujian jarak citra

Pada tabel 2 menunjukkan bahwa hasil jarak citra mempengaruhi keberhasilan proses pengolahan citra. Perbedaan pengukuran jarak tersebut menghasilkan nilai keberhasilan pendeteksi citra. misalkan Pada jarak 10 cm dan 20 cm citra yang ditampilkan tidak nampak secara sempurna dikarenakan cahaya area citra yang akan di deteksi tidak terdeteksi seluruhnya dan resolusi gambar yang di lakukan saat proses akuisisi citra masih kurang memenuhi.

TABEL II
HASIL PENGUJIAN JARAK CITRA

No	Jarak (cm)	Hasil plot window a dan b	Hasil deteksi
1	10		Tidak terdeteksi
2	20		Sulit terdeteksi
3	30		Terdeteksi
4	40		Terdeteksi
5	50		Terdeteksi
6	60		Terdeteksi
7	70		Terdeteksi
8	80		Terdeteksi
9	90		Terdeteksi
10	100		Tidak terdeteksi

Resolusi minimal sebuah citra harus sesuai dengan gambar template yaitu sebesar 271x195 *pixel* sehingga apabila resolusi kamera yang digunakan sebesar 640x480 *pixel*, maka kamera dengan resolusi tersebut dapat mendeteksi objek gambar dengan range jarak minimal 20 cm hingga maksimal 90 cm. berikut adalah hasil pengukuran jarak citra yang ditunjukkan pada tabel 2.

C. Pengujian *lowthreshold*

Pengujian *low threshold* dipilih karena *low threshold* adalah salah satu indikator keberhasilan dalam algoritma program penelitian ini. Nilai *lowthreshold* yang dirubah adalah *lowthreshold* pada parameter deteksi *canny*. Besar nilai *low threshold* akan menentukan suatu citra dikatakan sebagai tepian atau tidak. Nilai *low threshold* diubah-ubah pada rentang 0 sampai 100 dengan menggeser *trackbar* pada *plot window A*. *Output* yang didapat dari pengujian ini adalah hasil pendeteksian *canny* dan *delay* pendeteksian citra yang dapat diketahui pada tabel 3.

Analisis yang didapatkan pada pengujian ini ialah semakin besar nilai *lowthreshold* maka *edge pixel* akan berkurang dan *delay* yang didapatkan semakin berkurang pula dan sebaliknya. Tetapi apabila semakin besar *low threshold* sensitifitas kamera semakin berkurang hal ini terbukti pada nilai *low threshold* sebesar 90 hingga 100. Pada nilai *low threshold* tersebut potensi kegagalan masih cukup besar, parameter kegagalan yang didapat seperti mulai terdapat sebuah *edge* yang hilang dan *pixel* yang dideteksi sangat rentan terjadi kegagalan sehingga citra sama sekali tidak bisa dideteksi sebagai citra referensi.

D. Pengujian *Lux*

Lux atau intensitas cahaya pada seberkas bidang ditunjukkan dalam satuan *lumens/m2*, *lux* adalah salah satu faktor penting yang mempengaruhi keberhasilan pengolahan citra. Pengujian *lux* ditunjukkan untuk melihat kemampuan kamera serta algoritma program yang digunakan ketika terpengaruh oleh perubahan nilai *lux* yang berbeda-beda.

Dari hasil pengujian *lux* yang dapat diketahui pada tabel 4, didapatkan analisis bahwa keberhasilan kamera dalam mendeteksi objek dapat dipengaruhi oleh besar intensitas cahaya pada area kamera tersebut. Hal ini dikarenakan cahaya dapat merubah warna objek yang terdeteksi, seperti contoh pada pencahayaan gelap atau nilai *lumens* yang terlalu kecil dapat merubah sebuah objek menjadi lebih berwarna hitam karena nilai *pixel* akan mengecil mendekati nilai 0 (hitam) seiring dengan berkurangnya nilai *lumens* hal ini berbanding terbalik apabila pada pencahayaan terang dapat merubah sebuah objek lebih berwarna putih karena nilai *pixel* akan membesar mendekati nilai 255 (putih) seiring dengan bertambahnya nilai *lumens*.

TABEL III
HASIL PENGUJIAN *LOWTHRESHOLD*

No	Thres hold	Delay (ms)	Keberhasilan deteksi	Output deteksi <i>canny</i> (window B)
1	0	2	terdeteksi	
2	10	1	terdeteksi	
3	20	< 1	terdeteksi	
4	30	< 1	terdeteksi	
5	40	< 1	terdeteksi	
6	50	< 1	terdeteksi	
7	60	< 1	terdeteksi	
8	70	< 1	terdeteksi	
9	80	< 1	terdeteksi	
10	90 - 100	-	tidak terdeteksi	

E. Pengujian *Black Pixel*

Citra yang berwarna hitam dan putih, adalah komponen citra dari model citra biner. Pada pengujian *black pixel*, sebuah citra biner akan dihitung nilai citra yang berwarna hitam. Pada tabel 5 adalah hasil perhitungan nilai *black pixel* ketika objek dalam kondisi *match*. Dan diketahui pada pengujian ini menunjukkan nilai *black pixel* kurang dari 12000 *pixel*, hal ini menunjukkan bahwa pendekatan objek yang dicocokkan berhasil dan memiliki nilai yang *black pixel* yang kecil.

TABEL IV
HASIL PENGUJIAN LUX

No	Nilai Lux (lumens /m ²)	Kendali gerak robot			
		Belok Kanan	Belok Kiri	Maju	Berhenti
1	0	Gagal	Gagal	Gagal	Gagal
2	0,8	Berhasil	Berhasil	Berhasil	Berhasil
3	3,4	Berhasil	Berhasil	Berhasil	Berhasil
4	29,1	Berhasil	Berhasil	Berhasil	Berhasil
5	78,8	Berhasil	Berhasil	Berhasil	Berhasil
6	496	Berhasil	Berhasil	Berhasil	Berhasil
7	553	Berhasil	Berhasil	Berhasil	Berhasil
8	794	Berhasil	Berhasil	Berhasil	Berhasil
9	83900	Berhasil	Berhasil	Berhasil	Berhasil
10	91500	Berhasil	Berhasil	Berhasil	Berhasil

TABEL V
HASIL PENGUJIAN BLACK PIXEL

Pengujian ke -	Jumlah pixel hitam (pixel)			
	Belok Kanan	Belok Kiri	Maju	Berhenti
1	1115	1158	3907	1578
2	1390	1201	4081	994
3	1308	1066	3802	1281
4	1087	1143	3955	866
5	1566	1227	3533	1269
6	898	1065	3561	1498
7	1198	1132	2353	1429
8	956	961	3311	1110
9	1249	999	2379	1505
10	1124	768	4119	2175

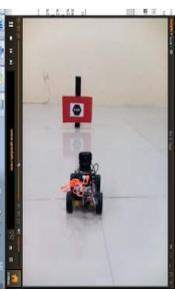
Nilai pada keseluruhan data menunjukkan objek arah maju memiliki nilai *black pixel* tertinggi hingga 4081 piksel. Hal ini dikarenakan untuk mendeteksi bentuk lingkaran cukup rumit dibandingkan dengan bentuk bangun yang memiliki banyak sudut. Bentuk lingkaran terbukti sangat rentan mengalami gangguan *noise* pada proses pengolahan di dalam penelitian ini. Sedangkan untuk objek arah panah berhenti, belok kanan dan belok kiri memiliki nilai *black pixel* yang rendah dengan nilai terkecil sebesar 866 piksel, hal tersebut dikarenakan kontur objek arah panah tidak terlalu rumit sehingga mudah terhindar dari kesalahan pada saat binerisasi citra.

F. Pengujian Program Pengendalian Gerak Robot

Pengolahan citra pada penelitian ini digunakan sebagai program input untuk menentukan kondisi *output* data *matching* yang didapatkan dari perbandingan antara citra kamera dengan citra referensi untuk mendapatkan nilai *error* persen yang terkecil hingga mendekati gambar asli yang dihitung dalam satuan *black pixel*. Setelah melakukan pendekatan *error persen* melalui program, gambar yang cocok disimpan dalam

variabel "match". Kemudian variabel tersebut akan mengaktifkan fungsi program *output* dengan melakukan fungsi perulangan "if-else". hasil pengendalian gerak robot terhadap objek yang telah ditentukan. Dapat diketahui pada tabel 6. Dan untuk hasil waktu yang dibutuhkan robot untuk merespon objek pada tabel 7.

TABEL VI
HASIL PENGUJIAN PENGENDALIAN ROBOT

No	Listing program	Kondisi robot	Keterangan Navigasi
1	<pre>if (match == 0) { //kiri bcm2835_gpio_wri te(INPUT2, HIGH); bcm2835_gpio_wri te(INPUT3, LOW); bcm2835_delay(50);}</pre>		Belok kiri
2	<pre>else if (match == 1) { //kanan bcm2835_gpio_wri te(INPUT2, LOW); bcm2835_gpio_wri te(INPUT3, HIGH); bcm2835_delay(50);}</pre>		Belok kanan
3	<pre>else if (match == 6) { //berhenti bcm2835_gpio_wri te(INPUT2, LOW); bcm2835_gpio_wri te(INPUT3, LOW); bcm2835_delay(50);}</pre>		Berhenti
4	<pre>else if (match == 7) { //maju bcm2835_gpio_wri te(INPUT2, HIGH); bcm2835_gpio_wri te(INPUT3, HIGH); bcm2835_delay(50);}</pre>		maju

TABEL VII
HASIL WAKTU RESPON ROBOT TERHADAP OBJEK

No	Waktu respon robot dalam kondisi diam	Waktu respon robot dalam kondisi bergerak	Waktu respon robot bergerak dalam satu lintasan
1	2 detik	2 detik	107 detik
2	1 detik	3 detik	85 detik
3	1 detik	2 detik	90 detik
4	1 detik	3 detik	97 detik
5	1 detik	2 detik	87 detik
6	1 detik	3 detik	88 detik
7	1 detik	2 detik	88 detik
8	1 detik	3 detik	81 detik
9	1 detik	3 detik	113 detik
10	1 detik	3 detik	102 detik

IV. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan. Pada pengujian deteksi canny dengan mengubah nilai *low threshold* dalam range 20 – 80 didapatkan hasil pendeteksian tepian citra yang dapat mengurangi *noise* dan dapat menghasilkan *delay buffering* yang kecil yaitu sebesar 1 detik, (lihat tabel 3). Hasil Keberhasilan deteksi objek dengan dipengaruhi intensitas lumens/m² yang berbeda-beda tidak terlalu mempengaruhi proses pengolahan citra pada penelitian ini, karena objek yang digunakan bukan citra berwarna melainkan citra biner hitam – putih. Pada pengujian ini didapatkan hasil kemampuan alat untuk mendeteksi objek dalam intensitas nilai lumens/m² minimal yaitu 0,8 lumens/m² dan intensitas nilai lumens/m² maksimal lebih dari 91500 lumens/m², (lihat tabel 4). Respon robot dalam kondisi diam membutuhkan waktu lebih cepat dengan nilai rata-rata sebesar 1,1 detik dibandingkan dengan respon robot dalam kondisi bergerak yaitu sebesar 2,5 detik. Hal ini dikarenakan pada kondisi robot diam, proses akuisisi kamera terhadap sebuah objek lebih cepat. (lihat tabel 7).

V. SARAN

Berdasarkan hasil pengujian yang telah dilakukan pada penelitian ini, peneliti memberikan saran untuk penyempurnaan pada penelitian berikutnya seperti :

1. Penambahan sensor serta motor servo dan pengembangan algoritma pengolahan citra sebagai variasi dalam mengatur navigasi robot.
2. Diperlukan *hardware* yang lebih baik agar mampu memproses citra dalam keadaan *mobile* dan *realtime*.

REFERENSI

- [1] Muhammad Karim *at all*. 2014. *Robot Line Follower Berbasis Raspberry Pi Dengan Sensor Kamera*. Palembang: AMIK MDP, Palembang.
- [2] Awaluddhin Choliq Azis, 2015. “Aplikasi Tracking Object pada Sistem Web Streaming dengan Protokol TCP/IP sebagai Sistem Navigasi Mobile Robot Berbasis Mini PC”, Jember : Universitas Jember.
- [3] Daniel Ricard Andriessen, 2011. “Pengendalian *Mobile Robot* Berbasis Webcam Menggunakan Perintah Isyarat Tangan”, Surabaya : STIKOM Surabaya.
- [4] Reda Anggra Distira, 2012. Desain Sistem Navigasi Robot dengan Isyarat Mata Menggunakan Metode Canny dan Hough Transform”, Jember : Universitas Jember.
- [5] Samuel Matos, 2014. “Signs reading with OpenCV” (<http://roboticssamy.blogspot.pt/>) diakses pada 24 Juni 2015

