

Sistem Orientasi Objek Dengan Metode Stereo Vision Berbasis Raspberry Pi

Rajali Ginting¹, Raditiana Patmasari², Suci Aulia³

Jurusan Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom^{1,2}
Fakultas Ilmu Terapan, Universitas Telkom³

E-mail: rajali@ student.telkomuniversity.ac.id¹, raditiana@telkomuniversity.ac.id²,
sucia@tass.telkomuniversity.ac.id³

Article Info

History :

Dikirim
Direvisi
Diterima

Kata Kunci :

Canny edge detection
Disparitas citra
Harris corner detection
Hough line transform
Stereo vision

Abstrak

Pengukuran estimasi jarak sebuah objek dapat dilakukan dengan berbagai cara, antara lain dengan memanfaatkan sensor pengukur jarak seperti sensor ultrasonik, sensor infra merah dan laser. Tetapi cara tersebut memiliki kekurangan seperti memiliki batasan-batasan karakteristik bahan yang digunakan dan memiliki kesulitan tersendiri jika dikaitkan dengan orientasi dimana objek tersebut berada.

Berdasarkan kekurangan tersebut maka pada penelitian ini akan dirancang suatu sistem pengukuran orientasi objek menggunakan metode *stereo vision* yang berbasis Raspberry pi. Sistem orientasi objek ini mengacu pada estimasi jarak dan dimensi objek. Metode *stereo vision* melakukan pengukuran jarak dan dimensi secara visual dengan menggunakan dua buah kamera *webcam*. Akuisisi citra dilakukan dalam interval waktu yang bersamaan, sehingga akan terdapat dua citra yang memiliki objek yang sama tetapi dari sudut pandang yang berbeda. Kedua citra akan digunakan untuk menghitung nilai disparitas yang akan digunakan sebagai salah satu parameter dalam melakukan pengukuran jarak.

Menghitung nilai disparitas citra dilakukan menggunakan beberapa metode *image processing* yaitu menggunakan deteksi tepi *Canny edge*, deteksi garis *Hough line transform* dan deteksi sudut *Harris corner*. Penelitian ini dapat digunakan sebagai alat pengukuran jarak dan dimensi objek yang bergerak secara *real time* dengan tingkat akurasi diatas 95% dengan tingkat presisi 100%.

© This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License..

Koresponden:

Rajali Ginting,
Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro
Universitas Telkom,
Jl.Telemunikasi No.1, Bandung, Indonesia, 40287
Email : Rajaliginting@gmail.com

1. PENDAHULUAN

Pengukuran jarak dan dimensi suatu objek menjadi hal yang penting dalam berbagai aplikasi, terutama dalam sistem otonom bergerak [1]. Informasi tentang jarak dan ukuran objek yang berada disekeliling berguna untuk navigasi dan identifikasi sekitar. Terdapat dua metode yang sering digunakan untuk mengukur jarak, yaitu dengan menggunakan metode aktif dan metode pasif. Metode aktif bekerja dengan cara mengirimkan sinyal kepada objek yang akan diukur, seperti penggunaan sensor ultrasonik [2], sinar laser dan sensor infra merah [3], sedangkan metode pasif bekerja tanpa

mengirimkan sinyal ke objek seperti menggunakan citra kamera. *Stereo vision* merupakan sistem yang memiliki konfigurasi menggunakan dua buah kamera yang memiliki karakteristik yang sama dan dipasang sejajar dengan jarak tertentu [4] [5].

Sebagian besar sistem otonom bergerak saat ini dilengkapi dengan sensor jarak menggunakan kamera. Beberapa karya penelitian sebelumnya menggunakan konfigurasi kamera stereo dengan tujuan pengukuran jarak objek. Sebagai contoh, Tsung-Shiang Hsu dan Ta-Chung Wang dengan judul penelitian “*An Improvement Stereo Vision Images Processing for Object Distance Measurement*” [6]. Penelitian tersebut menggunakan metode *Speed up Robust feature* (SURF) untuk melakukan pengukuran jarak objek dengan tingkat ketelitian diatas 95%. Irfan Marzuqi dan Agus Khumaidi dengan judul penelitian “*Segmentasi dan Estimasi Jarak Bola dengan Robot Menggunakan Stereo Vision*” [7]. menggunakan metode *principal point* dari bidang proyeksi dengan tingkat ketelitian diatas 95%. Yasir M Mustafah, dkk dengan judul penelitian “*Stereo Vision Images Processing for Real-Time Object Distance and Size Measurements*” [8]. Menggunakan metode *Morphology and Connected Component Analysis* (CCA) dan *blob tracking* dengan tingkat ketelitian yang baik.

Berdasarkan kelebihan tersebut maka pada penelitian ini akan dirancang suatu sistem pengukuran orientasi objek menggunakan metode *stereo vision* yang berbasis Raspberry pi. Sistem orientasi objek ini mengacu pada estimasi jarak dan dimensi objek. Akusisi citra objek dilakukan dengan menggunakan dua kamera *webcam* dalam interval waktu yang bersamaan. Masalah yang akan ditemui pada saat menggunakan metode *stereo vision* adalah mencari nilai disparitas dari kedua citra [9]. Nilai disparitas sangat bergantung pada hasil pencarian piksel yang sama dari kedua citra yang diambil oleh kamera. Pada penelitian ini proses pencarian piksel tersebut dilakukan menggunakan beberapa metode *image processing* yaitu menggunakan deteksi tepi *Canny edge*, deteksi garis *Hough line transform* dan deteksi sudut *Harris corner*. Sehingga penelitian ini dapat digunakan sebagai alat pengukur estimasi jarak antara objek dengan kamera dan pengukuran dimensi objek yang bekerja secara *real time*.

2. Dasar Teori

2.1. Citra

Secara umum citra merupakan suatu foto, gambar atau berbagai visualiasi objek yang ditampilkan secara dua dimensi. Citra dapat direpresentasikan dalam bentuk digital maupun tercetak. Citra digital merupakan kumpulan angka – angka dalam dua dimensional. Angka – angka pada citra merupakan hasil kuantifikasi dari intensitas tingkat kecerahan dari masing – masing piksel penyusun citra. Piksel merupakan bagian elemen terkecil penyusun citra, jumlah piksel per unit panjang dalam citra dikenal sebagai resolusi citra. Semakin tinggi resolusi suatu citra maka jumlah piksel penyusunnya akan semakin banyak.

2.2. Raspberry pi

Raspberry pi merupakan sebuah *Personal Computer* (PC) yang memiliki ukuran yang kecil, tetapi berbeda dengan laptop maupun PC karena Raspberry pi dapat digunakan untuk proyek elektronika, namun Raspberry Pi juga dapat melakukan apa yang dapat dilakukan PC. Raspberry pi hanya memiliki RAM sebesar 512 MB hingga 1 GB dan memiliki prosesor ARM dan pengolah grafik.

Raspberry pi juga memiliki sistem operasi seperti PC pada umumnya, bernama *Raspbian* namun dapat juga di instalasi sistem operasi lainnya seperti *Ubuntu Mate*, *Windows*, dan berbagai macam sistem operasi lainnya. Raspberry pi memiliki beberapa *General Pin Input/Output* (GPIO). GPIO dapat digunakan untuk berinteraksi dengan rangkaian elektronika, seperti untuk mengaktifkan sensor maupun menjalankan rangkaian lainnya. Setiap pin GPIO memiliki fungsinya masing-masing dimana terdapat pin yang dapat diprogram dan ada yang tidak dapat diprogram [11].

2.3. Webcam

Webcam merupakan suatu perangkat yang berupa kamera yang digunakan sebagai pengambil citra/gambar dan memiliki mikropon (*optional*) digunakan sebagai pengambil suara/audio yang dapat dikendalikan oleh sebuah komputer atau suatu jaringan komputer. Citra yang diambil oleh

webcam akan ditampilkan ke layar monitor. Karena *webcam* dikendalikan oleh komputer maka terdapat interface atau port yang digunakan untuk menghubungkan *webcam* dengan komputer.

2.4. Stereo vision

Stereo vision merupakan suatu cara yang digunakan untuk mendapatkan citra *stereo* dari suatu objek menggunakan dua posisi kamera yang berbeda. Citra *stereo* didapatkan menggunakan dua kamera yang diletakkan pada bidang sejajar dengan jarak tertentu. Citra *stereo* yang digunakan untuk mencari nilai disparitas, nilai tersebut dapat digunakan untuk menghitung jarak dan dimensi suatu objek dengan memprosesnya menggunakan beberapa metode *image processing*. Semakin besar nilai disparitas menunjukkan bahwa objek semakin dekat, begitupun dengan sebaliknya.

2.5. Deteksi Tepi Canny edge

Canny edge detection dikemukakan pada tahun 1986 oleh John Canny [12]. *Canny edge* merupakan salah satu teknik deteksi tepi yang cukup populer digunakan dalam pengolahan citra, dikarenakan metode deteksi tepi ini cukup optimal dengan tingkat kesalahan yang rendah. *Canny edge* melokalisasi posisi titik – titik *edge* pada citra setebal satu piksel sehingga didapat *edge* yang sepresisi mungkin.

2.6. Hough Line Transform

Hough line transform merupakan transformasi citra yang digunakan untuk memperoleh fitur dari sebuah gambar. *Hough line Transform* adalah salah satu teknik yang umum digunakan dalam sistem pengolahan citra untuk mendeteksi objek yang memiliki bentuk seperti garis. Keuntungan utama dari penggunaan metode transformasi *Hough* adalah dapat mendeteksi sebuah tepian atau *edge* secara relatif tidak terpengaruh oleh derau. Transformasi *Hough* bekerja menggunakan *voting* suara terbanyak untuk menentukan nilai parameter yang paling tepat. Apabila didalam sebuah citra yang telah ditransformasikan ke dalam ruang parameter $m - c$ terdapat beberapa garis yang saling berpotongan pada sebuah titik. Untuk mengetahui titik tersebut adalah sebuah garis lurus atau bukan didalam ruang parameter $x - y$ dilakukan *voting* menggunakan persamaan 3.

$$y = mx + c \quad (3)$$

Apabila di dalam citra terdapat sebuah garis lurus yang memiliki persamaan yang sama dengan persamaan (1.3), kemudian garis tersebut ditransformasikan ke bentuk ruang parameter $m - c$ maka akan diperoleh beberapa garis yang saling berpotongan pada suatu titik. Transformasi tersebut diperoleh dengan cara melakukan perhitungan dari setiap titik pada citra kedalam nilai $m - c$ menggunakan persamaan 4.

$$c = -mx + y \quad (4)$$

Pada kenyataannya, apabila terdapat garis yang vertikal akan menjadi masalah didalam melakukan perhitungan, dikarenakan garis vertikal memiliki nilai gradien kemiringan m yang nilainya tak terhingga. Untuk mengatasi masalah tersebut dapat menggunakan persamaan 5.

$$\rho = x \cos \theta + y \sin \theta \quad (5)$$

Dari persamaan 5 dilakukan transformasi garis lurus kedalam ruang parameter ρ (*rho*) - θ (*theta*) yang akan mengakibatkan penumpukan antar kurva sinusoida di suatu titik [13].

2.7. Harris Corner Detection

Sudut merupakan suatu titik yang memiliki lingkungan lokal terdiri dari dua arah tepi atau *edge* yang dominan dan berbeda. Sudut juga dapat diartikan sebagai persimpangan yang memiliki dua sisi, dimana sisi memiliki intensitas kecerahan yang tiba-tiba berubah dalam suatu citra. *Harris corner detection* merupakan operator yang dapat digunakan untuk mendeteksi suatu sudut karena dapat membedakan antara tepi dengan sudut secara akurat, karena akan menghasilkan nilai yang konsisten dari suatu citra walaupun telah mengalami rotasi, penskalaan, variasi pencahayaan ataupun memiliki derau didalamnya [14]. *Harris corner* akan mendeteksi sudut dengan cara memperhitungkan variasi nilai intensitas dari suatu citra menggunakan suatu jendela biner yang digeser ke arah sumbu x dan sumbu y . Untuk mencari variasi nilai intensitasnya dapat menggunakan persamaan 6.

$$\begin{aligned}
E(u, v) &= \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \\
&= \sum u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2 \\
&= \sum [u \ v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\
&= [u \ v] \left(\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}
\end{aligned} \tag{6}$$

dimana :

E = perbedaan antara jendela asli dengan jendela yang digeser.

u = pergeseran jendela ke arah sumbu x.

v = pergeseran jendela ke arah sumbu y.

w(x,y) = fungsi jendela pada posisi x dan y.

I(x+u,y+v) = intensitas pergeseran jendela

I(x,y) = intensitas asli.

2.8. Parameter Pengukuran Jarak dan Dimensi

Langkah awal dalam menghitung jarak dan dimensi objek adalah melakukan konversi piksel kedalam sentimeter. Konversi piksel dilakukan menggunakan persamaan 7.

$$\text{Ukuran piksel} = \frac{(Focal\ Length) \times (\text{Ukuran Objek sebenarnya})}{(\text{Jarak kamera dengan objek}) \times (\text{Jumlah Piksel Objek Sumbu x})} \tag{7}$$

Setelah ukuran piksel didapatkan, selanjutnya melakukan pengukuran jarak antara objek dengan kamera menggunakan persamaan 8.

$$\text{Jarak} = \frac{(Focal\ Length) \times (\text{Baseline Kamera})}{(\text{Disparitas}) \times (\text{Ukuran Piksel})} \tag{8}$$

Untuk dapat melakukan pengukuran dimensi panjang dan tinggi objek dengan jarak yang berubah – ubah digunakan persamaan 9 dan 10.

$$\text{Panjang Objek} = \frac{(\text{Jarak}) \times (\text{Ukuran Piksel}) \times (\text{Jumlah Piksel Objek Sumbu x})}{(Focal\ Length)} \tag{9}$$

$$\text{Tinggi Objek} = \frac{(\text{Jarak}) \times (\text{Ukuran Piksel}) \times (\text{Jumlah Piksel Objek Sumbu y})}{(Focal\ Length)} \tag{10}$$

Dimana :

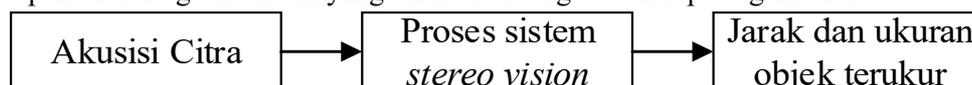
Focal length = 4.4 mm (berdasarkan spesifikasi kamera).

Baseline = 260 mm (jarak antar kamera)

3. Model Sistem

3.1. Perancangan Sistem

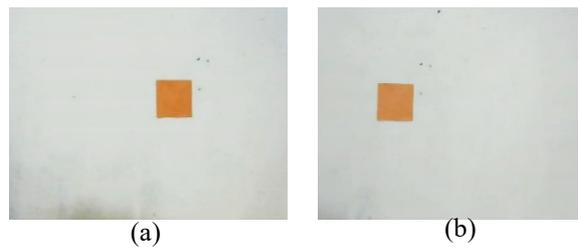
Adapun blok diagram sistem yang akan dirancang adalah seperti gambar 1.



Gambar 1. Blok diagram perancangan sistem

3.2. Akusisi Citra

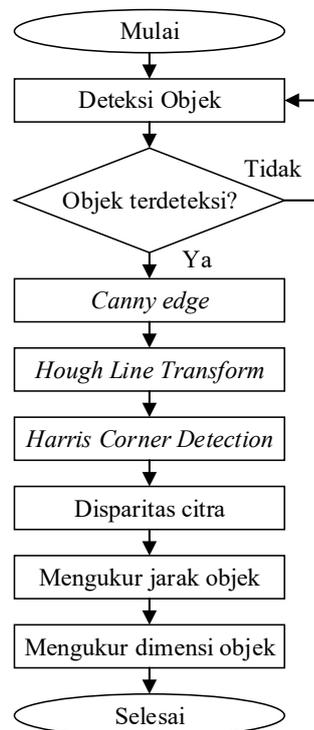
Proses akusisi citra ini dilakukan dengan menggunakan dua buah kamera *webcam* yang dipasang sejajar dengan jarak *baseline* 26 cm. Pengambilan citra dilakukan dalam selang waktu yang bersamaan. Sehingga akan menghasilkan dua buah citra dengan objek yang sama, tetapi dengan sudut pandang berbeda. Gambar 2 merupakan hasil akusisi citra menggunakan kamera *stereo* dengan jarak kamera dengan objek sejauh 1 meter.



Gambar 2. (a) Citra kanan. (b) Citra kiri

3.3. Proses Sistem Stereo vision

Adapun tahap dan metode yang dilakukan untuk memproses citra dapat dilihat pada diagram alir gambar 3.



Gambar 3. Diagram alir proses stereo vision

Tahap – tahap tersebut meliputi akuisisi citra, deteksi objek, deteksi tepi menggunakan operator *Canny edge*, deteksi garis menggunakan metode *Hough line transform*, deteksi sudut menggunakan metode *Harris corner* dan menghitung perbedaan nilai disparitas antar kedua citra. Penjabaran berikut ini hanya menjelaskan proses satu buah citra, dikarenakan proses kedua citra yang sama.

3.4. Mendeteksi Objek

Deteksi objek dilakukan dengan segmentasi warna. Segmentasi warna bertujuan untuk memisahkan antara warna objek dengan warna latar belakang pada citra. Format citra RGB yang dihasilkan pada saat melakukan akuisisi citra kemudian dikonversi kedalam format YCbCr. Untuk mendeteksi warna dilakukan menggunakan dua nilai *threshold* yaitu minimum dan maksimum intensitas nilai piksel pada objek. Untuk nilai yang memiliki intensitas diluar nilai *threshold* akan diubah menjadi piksel 0 atau warna hitam, karena akan dianggap sebagai warna latar belakang seperti pada gambar 4.

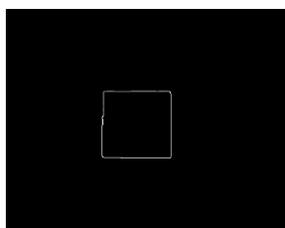


Gambar 4. (a) Citra asli. (b) Hasil Deteksi warna

3.4.1. Deteksi Tepi *Canny Edge*

Secara ringkas metode *Canny edge* akan dijelaskan sebagai berikut :

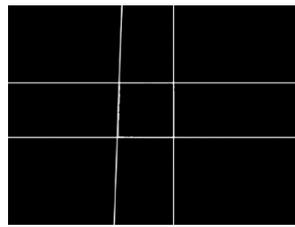
1. Proses pertama dalam melakukan deteksi tepi adalah mengubah citra kedalam bentuk *grayscale*.
2. Selanjutnya citra dihaluskan menggunakan filter *Gaussian*.
3. Hitung nilai gradien horizontal dan vertikal pada citra dengan melakukan konvolusi menggunakan operator *Sobel*.
4. Hitung *magnitude* dan arah *magnitude* citra.
5. Setelah arah tepi didapatkan, selanjutnya melakukan proses *Non-maximum suppression* untuk menekan atau menghilangkan piksel – piksel yang tidak memiliki nilai maksimum, disepanjang garis tepi dan searah dengan garis tepi. Sehingga menghasilkan garis tepi yang presisi setebal satu piksel.
6. Pada proses *Non-maximum Suppression* yang belum maksimal mengakibatkan beberapa tepi yang sebenarnya bukan tepi akan terdeteksi yang diakibatkan oleh noise pada citra. *Double Thresholding* digunakan untuk menangani masalah tersebut dikarenakan memiliki dua *thresholding* yaitu *low thresholding* dan *high thresholding*. *Low thresholding* berfungsi untuk mengubah semua nilai piksel dibawah *thresholding* tersebut bernilai 0 sedangkan *High thresholding* berfungsi untuk membuat semua nilai piksel diatas nilai tersebut menjadi bernilai 1. Sehingga dengan menggunakan *Double Thresholding* deteksi tepi dapat dilakukan dengan lebih maksimal.
7. Langkah terakhir melakukan proses untuk memutuskan semua tepi yang terdeteksi benar – benar tepi dan mana yang bukan tepi. Oleh karena itu, proses ini mengacu pada nilai *double thresholding*. Untuk setiap tepi dengan intensitas nilai lebih tinggi dari nilai *high thresholding* dipastikan sebagai tepi sebenarnya, sedangkan intensitas dibawah nilai *low thresholding* dipastikan bukan tepi sehingga akan dihilangkan. Sedangkan untuk intensitas nilai tepi yang berada diantara kedua nilai *thresholding* akan diklasifikasikan menjadi tepi atau bukan tepi berdasarkan konektivitas tepi tersebut. Apabila tepi tersebut terhubung dengan piksel yang tepi sebenarnya, maka tepi tersebut akan dianggap sebagai bagian dari tepi sebenarnya, apabila tidak terhubung dengan tepi sebenarnya maka akan dihilangkan. Gambar 5 dibawah merupakan hasil deteksi menggunakan metode *Canny edge*.



Gambar 5. Hasil deteksi *Canny edge*

3.4.2. *Hough Line Transform*

Metode *Hough line transform* digunakan untuk memperbaiki garis tepi yang tidak sempurna dari hasil keluaran deteksi tepi *Canny edge*. Perbaikan garis tepi dilakukan dengan cara mengganti garis tepi dengan garis tepi yang baru dari hasil deteksi *Hough line transform*. Sehingga akan dihasilkan objek dengan sudut yang sempurna, untuk memaksimalkan perhitungan untuk mencari nilai disparitas menggunakan deteksi sudut *Harris corner*. Hasil deteksi garis dari keluaran *Canny edge* menggunakan *hough line transform* seperti pada gambar 6.



Gambar 6. Hasil deteksi *Hough line transform*

Dari hasil garis yang telah terdeteksi, selanjutnya dilakukan perbaikan tepi keluaran *Canny edge* menggunakan garis yang baru. Sehingga didapatkan sudut yang maksimal pada objek.

3.4.3. *Harris Corner Detection*

Metode *Harris corner* digunakan untuk mendeteksi sudut pada citra, yang bertujuan untuk mencari nilai disparitas antar kedua citra. Nilai disparitas digunakan sebagai salah satu parameter untuk menghitung jarak dan dimensi objek. Langkah pertama dalam mendeteksi sudut adalah mengubah citra hasil keluaran *Hough line transform* kedalam bentuk *grayscale* yang memiliki rentang nilai piksel 0 sampai 255. Selanjutnya menghitung nilai gradien vertikal dan horizontal citra dengan melakukan konvolusi menggunakan kernel *prewitt*.

$$Prewitt G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad Prewitt G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Setelah mendapatkan nilai gradien, kemudian hitung nilai turunan setiap piksel menggunakan persamaan 11.

$$I_x^2 = I_x * I_x \quad I_y^2 = I_y * I_y \quad I_{xy} = I_x * I_y \quad (11)$$

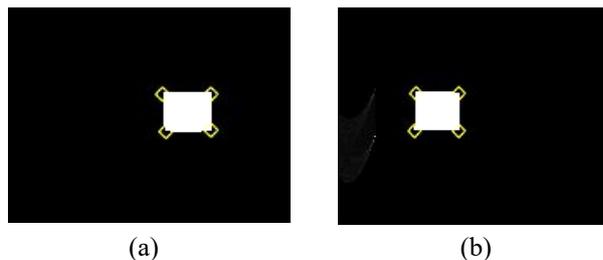
Konvolusi dengan filter *Gaussian* (G_σ) untuk mencari jumlah nilai turunan produk tiap pikselnya dengan persamaan berikut ini.

$$S_x^2 = I_x^2 \otimes G_\sigma \quad S_y^2 = I_y^2 \otimes G_\sigma \quad S_{xy} = I_{xy} \otimes G_\sigma \quad (12)$$

Hasil konvolusi tersebut digunakan untuk membentuk matriks baru untuk menghitung nilai respon *Harris*. Nilai respon *Harris* didapatkan dengan menggunakan persamaan berikut.

$$\text{Matriks } H = \begin{bmatrix} S_x^2 & S_{xy} \\ S_{xy} & S_y^2 \end{bmatrix} \\ \text{Respon } Harris = ((S_x^2) \cdot (S_y^2) - (S_{xy})^2) - k((S_x^2) + (S_y^2))^2 \quad (13)$$

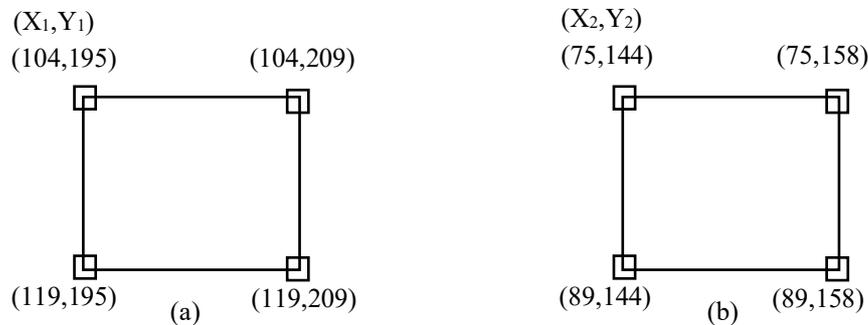
Nilai respon *Harris* digunakan untuk menentukan piksel tersebut suatu sudut atau bukan. Apabila nilai respon lebih besar dari nilai *threshold* maka piksel tersebut akan dianggap sebagai suatu sudut. Gambar 7 merupakan hasil deteksi sudut menggunakan metode *Harris corner*.



Gambar 7. (a) Deteksi sudut citra kanan. (b) Deteksi sudut citra kiri.

3.4.4. Menghitung Disparitas Citra

Untuk menentukan nilai disparitas antar kedua citra, digunakan pengurangan posisi kolom pada titik piksel sudut yang dideteksi, kolom piksel pada sudut citra kanan dikurangi dengan kolom pada sudut citra kiri, sehingga didapatkan hasil perbedaan jarak piksel diantara kedua citra. Posisi piksel dari hasil deteksi sudut menggunakan metode *Harris corner* ditunjukkan pada gambar 8.

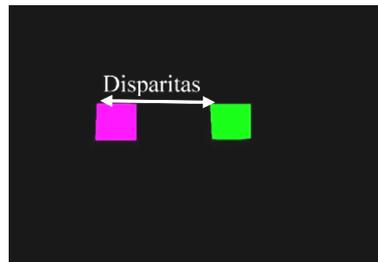


Gambar 8. (a) Posisi sudut citra kanan. (b) Posisi sudut citra kiri

Pada gambar 8 diatas (a) merepresentasikan citra yang diambil menggunakan webcam kanan dan (b) merepresentasikan citra yang diambil menggunakan webcam kiri. Hasil deteksi sudut digunakan untuk mencari nilai disparitas (D) citra dari kedua citra, dengan cara menghitung selisih posisi piksel antara kedua citra tersebut.

$$|D| = (Y_1 - Y_2) = (195 - 144) = 51$$

Dari perhitungan diatas didapatkan nilai disparitas antara kedua citra adalah 51.



Gambar 9. Disparitas Citra

3.5. Pengukuran Jarak dan Dimensi

Dalam melakukan pengukuran jarak dan dimensi objek, langkah pertama adalah mengkonversi ukuran piksel ke dalam sentimeter menggunakan persamaan 7. Pada contoh kasus berikut, objek yang diukur memiliki dimensi panjang dan lebar 15x10cm dengan jarak pengambilan citra sejauh 1 meter.

$$\text{Konversi ukuran piksel ke cm} = \frac{(4.4 \text{ mm}) \times (100 \text{ mm})}{(1900 \text{ mm}) \times (14)} = 0.16532 \text{ mm}$$

Setelah melakukan konversi ukuran piksel, didapatkan ukuran piksel pada citra sebesar 0.16352 mm. Selanjutnya melakukan pengukuran jarak antara objek dengan kamera menggunakan persamaan 8.

$$\text{Jarak} = \frac{(4.4 \text{ mm}) \times (260 \text{ mm})}{(51) \times (0.118 \text{ mm})} = 190 \text{ cm}$$

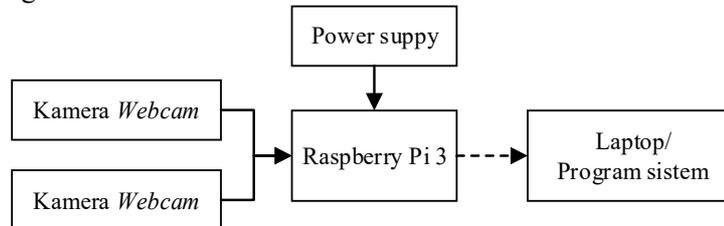
Pengukuran dimensi panjang dan lebar objek menggunakan persamaan 9 dan 10.

$$\text{Panjang Objek} = \frac{(1900 \text{ mm}) \times (0.16532 \text{ mm}) \times (14)}{(4.4 \text{ mm})} = 10 \text{ cm}$$

$$\text{Tinggi Objek} = \frac{(1900 \text{ mm}) \times (0.16532 \text{ mm}) \times (14)}{(4.4 \text{ mm})} = 10 \text{ cm}$$

4. Pengujian dan Analisis

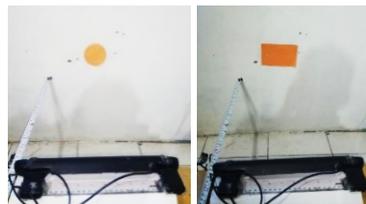
Pengujian sistem dilakukan menggunakan dua kamera *webcam* yang terhubung dengan Raspberry pi 3. Program dijalankan menggunakan laptop dan dikirim ke Raspberry pi menggunakan sambungan *wireless*. Laptop yang digunakan memiliki spesifikasi *dual-core* 1.8 Ghz. Skema pengujian seperti pada gambar 10.



Gambar 10. Diagram perancangan perangkat keras

4.1. Skema Pengujian

Skema pengujian menggunakan objek yang memiliki bentuk sederhana seperti persegi, persegi panjang dan lingkaran. Objek yang akan diukur berada didepan kamera. Skema pengukuran objek seperti gambar 11.



Gambar 11. Skema pengujian

Nilai yang akan diambil untuk menjadi data hasil pengukuran menggunakan skema pengulangan sebanyak 5 kali, dengan nilai yang paling banyak muncul yang akan dijadikan sampel nilai hasil. Nilai pengujian yang diambil seperti jarak, dimensi, nilai disparitas dan lama waktu komputasi. Hasil pengukuran jarak dibulatkan menggunakan *ceil* dan *floor*

4.2. Pengujian Akurasi pengukuran

Akurasi merupakan suatu sistem pengukuran dengan tingkat kedekatan hasil pengukuran terhadap nilai yang sebenarnya. Pengujian akurasi dilakukan untuk mengetahui tingkat keakuratan sistem dalam melakukan pengukuran.

4.2.1. Pengujian Menggunakan Metode *Canny Edge*, *Hough Line Transform* dan *Harris Corner*.

Pengujian ini bertujuan untuk mengetahui tingkat keakuratan sistem dan lama waktu komputasi apabila menggunakan ketiga metode *Canny edge*, *Hough line transform* dan *Harris corner* dalam *stereo vision*. Waktu komputasi merupakan hasil dari akumulasi setelah melakukan perulangan. Objek yang diukur memiliki bentuk segi empat dengan dimensi panjang dan lebar 10x10cm yang berada didepan kamera.

Tabel 1. Pengujian akurasi pengukuran segi empat

Jarak (cm)		Keakuratan Jarak (%)	Waktu Komputasi (detik)	Pengukuran (cm)		Disparitas
Sebenarnya	Terukur			A	B	
50	48	96	23.6	10.2	10	206
60	58	96.67	23.5	10.2	10	168
70	69	98.57	23.2	10.1	10	143
80	79	98.75	23.6	10	10	124
90	89	98.89	23.7	9.9	10.1	110

100	100	100	23.1	10	10	98
110	111	99.09	23.5	9.9	9.9	88
120	122	98.33	23	10.2	9.9	80
130	135	96.15	23.5	9.9	9.9	72
140	147	95	23.9	10	10	66
150	158	94.67	22	9.8	9.8	62
Rata - rata	97.4	23.3	10	9.9		

A = Panjang Objek B = Lebar Objek

Hasil pengujian menggunakan ketiga metode yaitu *Canny edge*, *Hough line transform* dan *Harris corner* didapatkan pengukuran yang stabil. Dikarenakan setiap melakukan perulangan hasil pengukuran tidak berubah, tetapi memiliki kekurangan waktu komputasi yang besar dikarenakan penggunaan metode *Hough line transform*. Tabel 1 merupakan hasil pengukuran akurasi objek segiempat, dari pengujian yang dilakukan sebanyak 11 kali didapatkan tingkat keakuratan jarak sebesar 97.4% dengan rata – rata waktu komputasi 23.3 detik.

4.2.2. Pengujian Menggunakan Metode *Canny Edge* dan *Harris Corner*.

Pengujian ini bertujuan untuk mengetahui tingkat keakuratan sistem dan lama waktu komputasi apabila menggunakan dua metode yaitu *Canny edge* dan *Harris Corner*. Waktu komputasi merupakan hasil akumulasi selama melakukan perulangan. Objek yang diukur dengan bentuk segi empat memiliki dimensi panjang dan lebar 10x10cm yang berada didepan kamera.

Tabel 2. Pengujian akurasi pengukuran segi empat

Jarak (cm)		Keakuratan Jarak (%)	Waktu Komputasi (detik)	Pengukuran (cm)		Disparitas
Sebenarnya	Terukur			A	B	
50	48	96	13.8	9.8	9.7	209
60	59	98.33	13.2	10	9.8	171
70	69	98.57	13.5	9.8	9.8	146
80	79	98.75	13	9.8	9.8	129
90	89	98.89	12.8	10	10	111
100	100	100	13.4	10	10	102
110	111	99.09	14.1	10.2	10.1	91
120	123	97.5	12.8	10	10.3	82
130	135	96.15	14	10.1	10.1	76
140	144	97.14	14.7	10.3	10.3	70
150	158	94.67	14.2	10.4	10.4	64
Rata - rata		97.7	13.5	10	10	

A = Panjang Objek B = Lebar Objek

Tabel 2 merupakan hasil pengukuran akurasi objek segi empat. Hasil pengujian dengan menggunakan dua metode yaitu *Canny edge* dan *Harris corner* didapatkan pengukuran yang kurang stabil. Dikarenakan setiap melakukan perulangan hasil pengukuran mengalami perubahan, tetapi memiliki kelebihan waktu komputasi yang kecil. Perubahan tersebut dikarenakan dari hasil keluaran *Canny edge* yang tidak sempurna, mengakibatkan sudut yang dideteksi oleh *Harris corner* berubah – ubah. Dari pengujian yang dilakukan sebanyak 11 kali didapatkan tingkat keakuratan jarak sebesar 97.7% dengan rata – rata waktu komputasi 13.5 detik.

4.2.3. Pengujian Menggunakan Metode *Canny Edge*

Pengujian ini bertujuan untuk mengetahui tingkat keakuratan sistem dan lama waktu komputasi apabila hanya menggunakan metode *canny edge* dalam *stereo vision*. waktu komputasi merupakan hasil dari akumulasi selama melakukan perulangan sebanyak lima kali. Objek yang diukur memiliki bentuk segi empat dengan dimensi panjang dan lebar 10x10cm yang berada didepan kamera.

Tabel 3. Pengujian akurasi pengukuran segi empat

Jarak (cm)		Keakuratan Jarak (%)	Waktu Komputasi	Pengukuran (cm)		Disparitas
Sebenarnya	Terukur			A	B	
50	48	96	9.7	9.7	9.9	212
60	58	96.67	9.1	9.7	10	176
70	68	97.14	9.9	9.7	10.1	150
80	77	96.25	9.2	9.9	9.9	132
90	88	97.78	8.9	10	10	116
100	100	100	9.5	10	10.4	102
110	109	99.09	9.6	10.3	10.3	94
120	119	99.17	9.1	10.1	10.1	86
130	131	99.23	8.5	10.5	10.5	78
140	138	98.57	9.4	10.4	10.4	74
150	150	100	9.2	10.6	10.6	68
Rata - rata		98.17	9.28	10.08	10.2	

A = Panjang Objek B = Lebar Objek

Hasil pengujian hanya menggunakan satu metode *Canny edge* didapatkan hasil memiliki kelebihan waktu komputasi yang kecil. Dikarenakan hanya menggunakan satu metode sehingga meringankan beban komputasi. Pada tabel 3 hasil pengukuran akurasi objek segi empat. Dari pengujian yang dilakukan sebanyak 11 kali didapatkan tingkat keakuratan jarak sebesar 98.17% dengan rata – rata waktu komputasi 9.28 detik.

4.3. Pengujian Presisi Pengukuran

Presisi merupakan kedekatan perbedaan nilai saat melakukan pengulangan pengukuran [12]. Pengujian tingkat presisi dalam pengukuran dimensi objek dengan bentuk segi empat, persegi panjang dan lingkaran. Pengulangan pengukuran dilakukan sebanyak 5 kali. Setiap pengujian menggunakan metode yang berbeda – beda.

Tabel 4 merupakan pengujian presisi menggunakan tiga metode yaitu *Canny edge*, *Hough line transform* dan *Harris corner*. Objek yang diukur memiliki bentuk persegi panjang dengan dimensi 15x10cm.

Tabel 4. Pengujian presisi objek persegi panjang

Jenis Objek	Jarak Pengukuran (cm)	Dimensi Objek (cm)	Panjang Objek Terukur (cm)	Lebar Objek Terukur (cm)
			15.02	9.58
			15.02	9.58
Persegi Panjang	100	15x10	15.02	9.58
			15.02	9.58
			15.02	9.58
Presisi			100%	100%

Tabel 5 merupakan pengujian presisi menggunakan dua metode yaitu *Canny edge* dan *Harris corner*. Objek yang diukur memiliki bentuk lingkaran dengan diameter 10cm.

Tabel 5. Pengujian presisi objek lingkaran

Jenis Objek	Jarak Pengukuran (cm)	Diameter Objek (cm)	Panjang Objek Terukur (cm)
			9.76
			9.76
Lingkaran	100	10	9.76
			9.76
			9.76
Presisi			100%

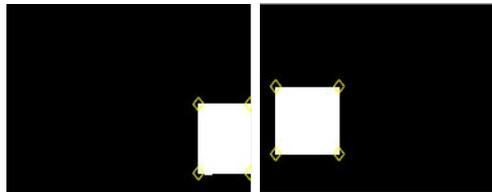
Tabel 6 merupakan pengujian presisi yang hanya menggunakan metode *Canny edge*. Objek yang diukur memiliki bentuk segi empat dengan dimensi 10x10cm.

Tabel 6. Pengujian presisi objek segi empat

Jenis Objek	Jarak Pengukuran (cm)	Dimensi Objek (cm)	Panjang Objek Terukur (cm)	Lebar Objek Terukur (cm)
			9.9	9.7
			9.9	9.7
Segi empat	80	10x10cm	9.9	9.7
			9.9	9.7
			9.9	9.7
	Presisi		100%	100%

Dari hasil pengujian presisi yang dilakukan menggunakan berbagai macam metode didapatkan hasil pengujian sebesar 100% atau hasil maksimal. Hasil tersebut dapat maksimal dikarenakan tidak ada perubahan cahaya atau pergerakan kamera dan objek yang diukur. Sehingga hasil pengukuran tidak mengalami perubahan.

Secara keseluruhan hasil pengujian menggunakan kamera dengan jarak *baseline* 26cm, didapatkan jarak minimum pengukuran 50cm untuk objek yang memiliki dimensi 10x10cm. Jarak minimum pengukuran dipengaruhi oleh jarak *baseline* kamera. Semakin jauh jarak *baseline* kamera mengakibatkan penurunan kemampuan kamera untuk mengambil citra objek dari jarak dekat. Sehingga citra yang diambil oleh kamera menjadi *overlap*, yang berakibat pada kesalahan perhitungan nilai disparitas, kesalahan tersebut akan berdampak pada hasil pengukuran jarak dan dimensi objek. Gambar 9 merupakan objek yang mengalami *overlap* dikarenakan jarak pengukuran yang terlalu dekat.



Gambar 9 Objek *overlap*

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan dari hasil pengujian dan analisis sistem orientasi objek menggunakan metode *stereo vision* maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan pengujian yang dilakukan dalam pengukuran jarak menggunakan metode *stereo vision* didapatkan hasil akurasi diatas 97%.
2. Metode Harris corner dapat digunakan untuk mencari nilai disparitas citra dalam melakukan pengukuran estimasi jarak dan dimensi objek.
3. Dari hasil pengujian yang dilakukan dalam pengukuran dimensi objek didapatkan tingkat presisi 100%.
4. Penggunaan metode *Hough line transform* mengakibatkan waktu komputasi semakin besar dikarenakan metode tersebut melakukan dua kali transformasi untuk mendeteksi garis.

5.2. Saran

Adapun saran sebagai pengembangan berikutnya agar bisa didapatkan hasil yang lebih baik lagi adalah sebagai berikut :

1. Disarankan untuk mencari metode yang dapat digunakan untuk mengukur jarak dan dimensi objek dengan jumlah objek lebih dari satu.
2. Disarankan untuk menggunakan kamera yang stereo yang arah pandang yang benar – benar lurus kedepan atau menggunakan kamera kinect.

Daftar Pustaka

- [1] P. Tripicchio, "Autonomous navigation of mobile robots: from basic sensing to problem solving," *Research on Electric and Electronic Measurement for the Economic Upturn*, pp. 1-6, September 2014.
- [2] R. Vairavan, S. Kumar dan L. Ashiff, "Obstacle Avoidance Robotic Vehicle Using Ultrasonic Sensor, Arduino Controller," *International Research Journal of Engineering and Technology*, vol. II, no. 5, pp. 2140-2143, Februari 2018.
- [3] R. Ismail, Z. Omar dan S.Suaibun, "Obstacle-avoiding robot with IR and PIR motion sensors," *Innovation in Aerospace Engineering and technology*, no. 152, pp. 1-6, 2016.
- [4] C. Suharlim, E. Putra, T. Budiman dan I. H. Kartowisastro, "Estimasi Posisi Objek Berdasarkan Stereo Vision," *Jurnal Teknik Komputer*, vol. I, no. 21, pp. 36-47, Februari 2013.
- [5] Y. D. Salman, K. R. Ku-Mahamud dan E. Kamioka, "Distance Measurement for Self-Driving Cars Using Stereo Camera," *International Conference on Computing and Informatics*, vol. I, no. 105, pp. 235-242, 2017.
- [6] H. Tsung-Shiang dan T.-C. Wang, "An Improvement Stereo Vision Images Processing for Object Distance Measurement," *International Journal of Automation and Smart Technology*, vol. II, no. 5, pp. 85-90, 2015.
- [7] M. Irfan, G. P. Arinata dan Z. Putra, "Segmentasi dan Estimasi Jarak Bola dengan Robot Menggunakan Stereo Vision," *Symposium on Robotic Systems and Control*, pp. 140-144, 2017.
- [8] Y. M. Mustafah, R. Noor, H. Hasbi dan A. W. Azma, "Stereo Vision Images Processing for Real-time Object Distance and Size Measurements," *International Conference on Computer and Communication Engineering*, pp. 659-663, 2012.
- [9] N. Rajajan, M.Ramkumar dan B.Monisha, "Disparity Estimation from Stereo Image," *Science Direct*, pp. 462-472, 2012.
- [10] D. A. Prabowo, D. Abdullah dan A. Manik, "Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking," *Jurnal Pseudocode*, vol. V, no. 2, pp. 85-91, September 2018.
- [11] J. Marot dan S. Bourennane, "Raspberry Pi for Image Processing Education," *European Signal Processing Conference*, pp. 2428-2432, 2017.
- [12] C. John, "A Computational Approach to Edge Detection," *Ieee Transactions On Pattern Analysis And Machine Intelligence*, vol. 6, no. 8, pp. 679-698, 1986.
- [13] H. Sa'diyah, R. Isnanto dan A. Hidayatno, "Aplikasi Transformasi Hough Untuk Deteksi Garis Lurus," *Jurnal Undip*, pp. 1-5, 2015.
- [14] S. Javier, N. Monz'on dan A. Salgado, "An Analysis and Implementation of the Harris Corner Detector," *Image Processing On Line*, pp. 305-328, 2018.