

# Rancang Bangun *Robot Vision* Untuk *Object Finding* Menggunakan *Color Tracking* pada perangkat RaspberryPi

Institut Teknologi Sepuluh Noverember

Oleh:  
Achmadi 2410100085

Pembimbing:  
Ir. Apriani K. MSc NIP: 195304041979012001

# Latar Belakang

## Robot Membantu Manusia



Asimo melayani Pelanggan



Roomba870 membersihkan karpet



Darpa membawa barang bawaan



AirDrone untuk survey udara

# Latar Belakang

## Sensor Robot



Sound Sensor



Ph-Diode Array untuk Line Reading



SRF07 untuk Proximity



Camera untuk vision sensor

# Latar Belakang

## Robot Vision

[Brett, 2010]

Dengan ketersediaannya yang luas, konten informasi tinggi, dan kesesuaian untuk lingkungan manusia serta harga yang murah dari kamera warna, sistem vision menjadi cocok untuk banyak platform robot.

# Latar Belakang

Contoh Vision: Asimo



Pengolahan Citra oleh Asimo

[Honda, 2007]

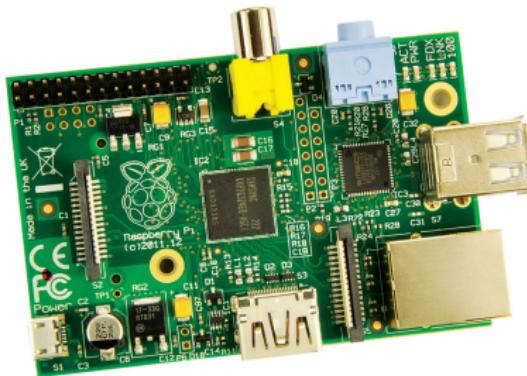
Jenis Kamera : CCD

Processor : 20 Unit Honda Proprietary CPU

Harga : \$2.500.000

# Latar Belakang

## Single Board Computer Raspberry Pi



### Single-Board Computer RaspberryPi B (\$60)

CPU : 700 MHz

RAM : 512 MB

USB : 2x USB 2.0

HMI : HDMI

# Rumusan Masalah

Bagaimana merancang bangun robot vision berbasis color tracking menggunakan OpenCV pada perangkat RaspberryPi.

# Batasan Masalah

Posisi objek terhadap robot telah ditentukan.

Objek yang akan diolah telah ditentukan ukuran dan warnanya.

Robot bekerja pada tingkat pencahayaan yang telah diatur.

# Robot Vision

[Budiharto,2012]

Robot Vision adalah mesin automatis atau semi-automatis yang mampu memperoleh informasi melalui proses akuisisi dan pemrosesan citra.

## Menemukan suatu objek (Object Finding)

Robot mampu mendapatkan posisi objek terhadap dirinya berdasarkan pengolahan citra yang didapatkan oleh kamera yang ada pada robot.

Robot mampu mendekati objek berdasarkan informasi yang diperoleh dari proses penentuan posisi sebelumnya.

# Mode Warna HSV

Mode Warna yang mengkarakteristikkan warna dalam 3 variabel, yaitu Hue, Saturation, dan Value

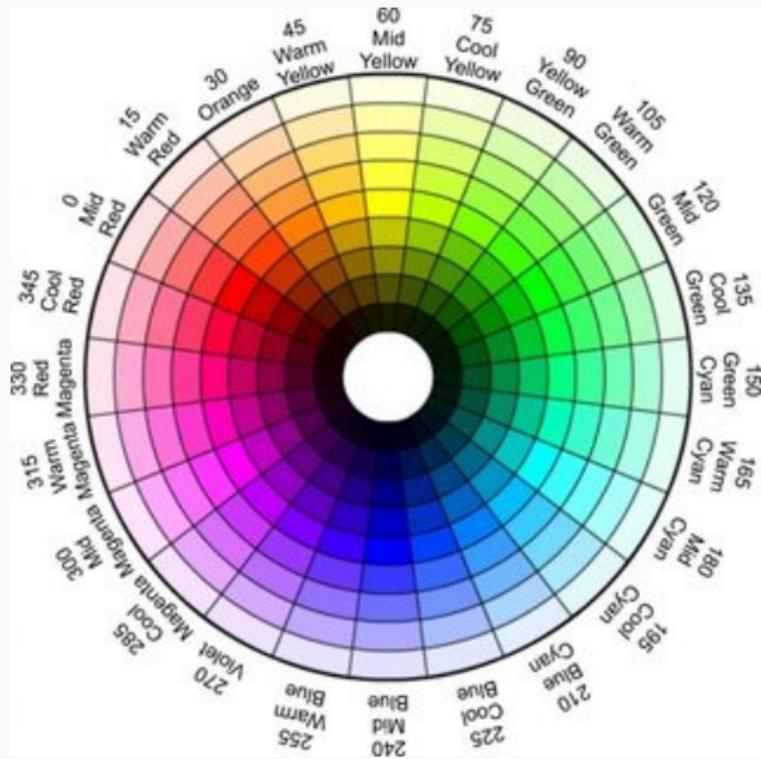
Hue adalah nilai warna dalam satuan sudut. Hue di plot dalam diagram berbentuk lingkaran dengan warna Red, Green, dan Blue terpisah dalam sudut 120 derajat. Warna lainnya merupakan tingkat campuran dari ketiga warna tersebut.

Saturation adalah nilai tingkat suatu warna terhadap warna Hitam. Tingkat Pencahayaan dapat mempengaruhi nilai Saturation.

Value adalah nilai tingkat suatu warna terhadap warna Putih. Tingkat Pencahayaan dapat mempengaruhi nilai Saturation.

# Mode Warna HSV

## Diagram HSV



# Mode Warna HSV

## Konversi RGB ke HSV

Kamera menghasilkan gambar dalam mode RGB maka perlu dikonversi ke HSV sebelum proses lebih lanjut.

$$V = \max(R, G, B) \quad (1)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V}, & \text{jika } V \neq 0 \\ 0, & \text{jika } V = 0 \end{cases} \quad (2)$$

$$H = \begin{cases} \frac{60(G-B)}{(V-\min(R,G,B))}, & \text{jika } V = R \\ 120 + \frac{60(B-R)}{(V-\min(R,G,B))}, & \text{jika } V = G \\ 240 + \frac{60(R-G)}{(V-\min(R,G,B))}, & \text{jika } V = B \end{cases} \quad (3)$$

# Thresholding

Thresholding adalah proses menyeleksi pixel berdasarkan range nilai HSV yang telah ditentukan.

Untuk pixel (8bit) yang sesuai jangkauan maka nilainya dikonversi ke 255 (putih). Jika tidak sebaliknya maka nilainya dikonversi ke 0 (hitam).

$$P(i,j) = \begin{cases} 255, & H_{min} \leq H \leq H_{max} \wedge S_{min} \leq S \leq S_{max} \wedge V_{min} \leq V \leq V_{max} \\ 0, & \text{selainnya} \end{cases} \quad (4)$$

Konversi ke binary image.

$$B(i,j) = \begin{cases} 1, & P(i,j) = 255 \\ 0, & P(i,j) = 0 \end{cases} \quad (5)$$

# Jangkauan Thresholding

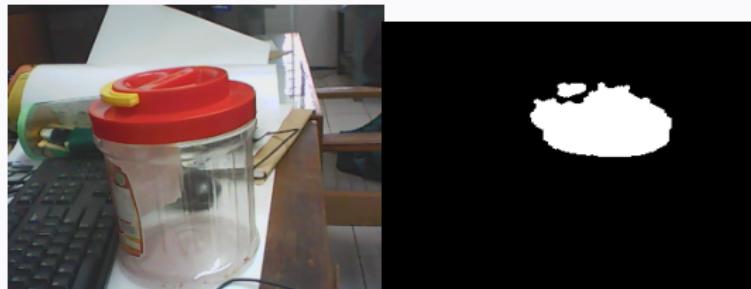
## Jangkauan HSV untuk warna Merah

$$160 \leq H \leq 179 \quad (6)$$

$$171 \leq S \leq 255 \quad (7)$$

$$120 \leq V \leq 255 \quad (8)$$

## Contoh hasil Thresholding



## Titik Pusat

Titik Pusat (Centroid) adalah titik yang menjadi pusat luasan dari total luas pixel. Titik ini identik dengan pusat massa untuk benda 2D.

$$\bar{x} = \frac{x_1 * m_1 + x_2 * m_2 + x_3 * m_3 + \dots}{m_1 + m_2 + m_3 + \dots} \quad (9)$$

$$\bar{y} = \frac{y_1 * n_1 + y_2 * n_2 + y_3 * n_3 + \dots}{n_1 + n_2 + n_3 + \dots} \quad (10)$$

# OpenCV

OpenCV (Open Computer Vision) adalah pustaka pemrograman untuk pengolahan citra yang bersifat opensource.

OpenCV mendukung bahasa pemrograman C, C++, dan Python.

OpenCV menyediakan fungsi-fungsi pengolahan citra yang lengkap mulai dari perhitungan histogram hingga algoritma Haar-Cascade untuk deteksi wajah.

OpenCV dapat digunakan sebagai pengganti Matlab untuk proses pengolahan citra yang mengharuskan program ditulis dalam C atau C++.

# Raspbian

Raspbian adalah Operating System berbasis Debian yang dijalankan di platform RaspberryPi.

Raspbian menggunakan kernel Linux untuk arsitektur armhf.

Melalui Raspbian ini nantinya menjalankan OpenCV untuk pengolahan citra dan komunikasi serial untuk mengontrol robot.

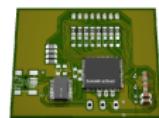
Lumbung Software (Repository) dari Raspbian telah menyediakan binary siap pakai sehingga memudahkan instalasi.

# Rancangan Bagian Robot

5 volt



WebCam Logitech C170

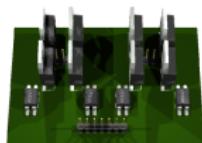


Motor Controller



Raspberry Pi

12 volt

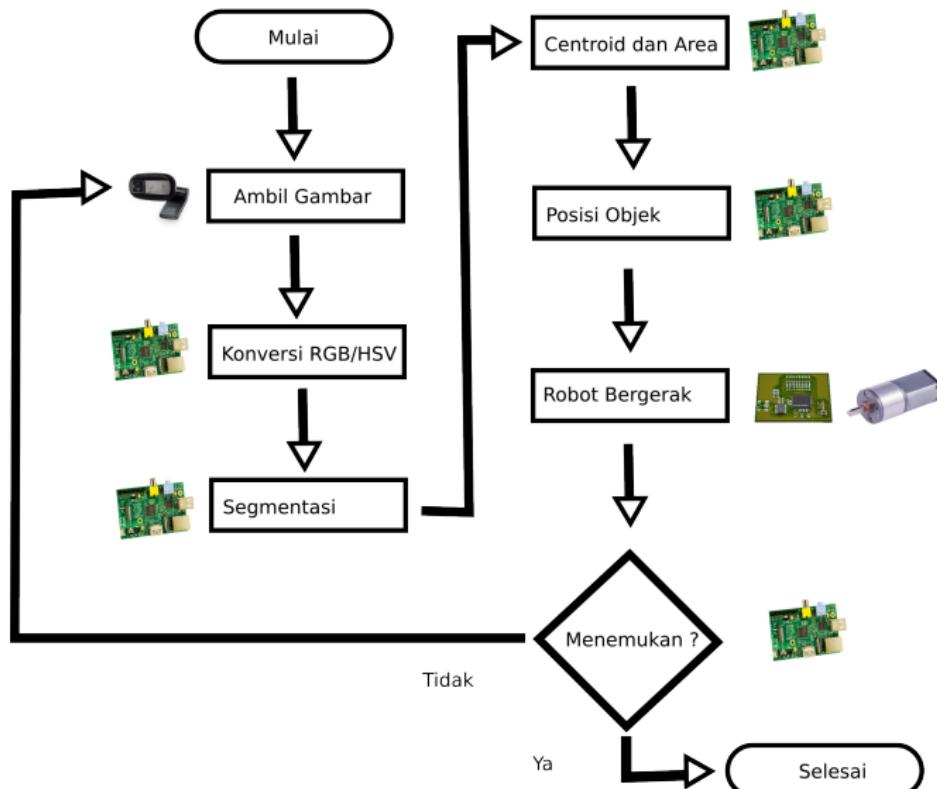


Motor Driver

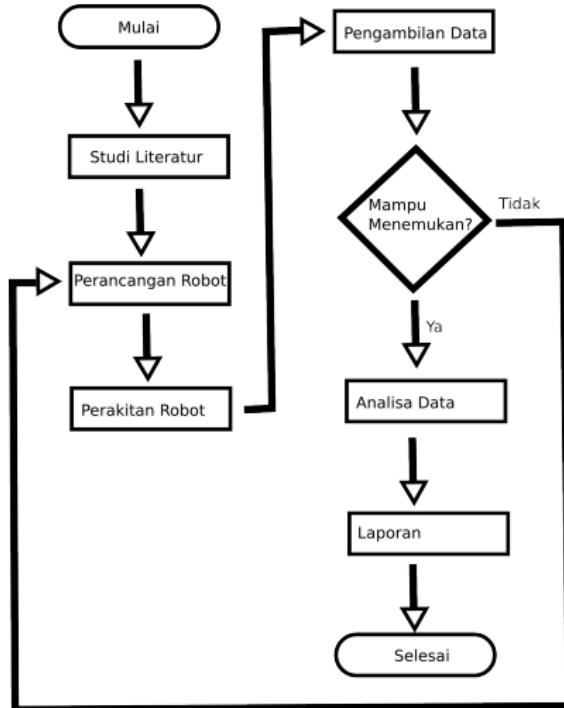


DC Geared Motor

# Rancangan Flow Chart Robot



# Flow Chart Penggerjaan



# Daftar Pustaka

-  Widodo Budiharto, Djoko Purwanto *Robot Vision* 2012
-  Brett Browning, Manuela Veloso *Real-Time, Adaptive Color-based Robot Vision* 2010
-  Honda Public Division *Asimo Technical Information* 2007
-  Ale Ude *Robot Vision* 2010
-  Andor Team *Digital Camera Fundamentals* 2012
-  Chao, Fei *A developmental approach to robotic pointing via humanrobot interaction* 2014
-  Upton, Eben *RaspberryPi User Guide* 2012
-  OpenCV Team *The OpenCV Reference Manual* 2014
-  Phillips, Dwyne *Image Processing in C* 2000
-  Zhou, Huiyu *Digital Image Processing Part I* 2010

# Mator Sakalangkong

# Catatan