

## BAB II TINJAUAN PUSTAKA

### 2.1 *Webcam Logitech C270 HD*

*Webcam* atau *web camera* adalah sebuah kamera video digital kecil yang dihubungkan ke computer melalui port USB ataupun *port* COM. Pada umumnya *webcam* tidak membutuhkan kaset atau tempat penyimpanan data, data hasil perekaman yang didapat langsung akan ditransfer ke komputer. Sehingga banyak digunakan untuk mengolah citra yang kemudian akan diolah dengan perangkat lunak untuk pemrosesan berbasis pixel, RGB, dan lain lain. Gambar *webcam* logitech C270 HD dapat dilihat pada **Gambar 2.1**.



**Gambar 2. 1** *Webcam* Logitech C270 HD  
(Sumber: Logitech.com)

Spesifikasi *webcam* logitech C270 HD [3]:

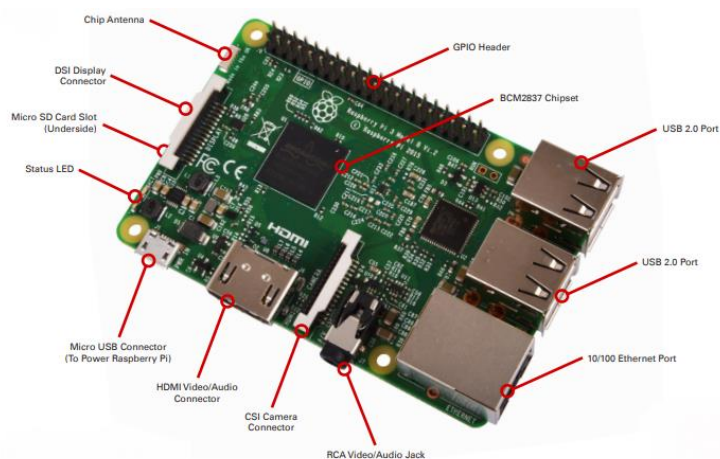
- a. *Full HD 720p/30fps video capture (up to 1080 pixels)*
- b. *HD video calling (1080 x 720 pixels) with recommended system*
- c. *Field of View (FOV) 60°*
- d. *Logitech Fluid Crystal™ Technology*
- e. *Autofocus*
- f. *Photos up to 8 megapixels (software enhanced)*
- g. *Built-in mics mono with automatic noise reduction*

- h. Hi-speed USB 2.0 *certified (recommended) and Universal clip fits laptops, LCD or CRT monitors.*

## 2.2 Raspberry Pi 3

Raspberry Pi 3 (juga dikenal sebagai RasPi) adalah sebuah SBC (Single Board Computer) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK)[1]. Raspberry Pi 3 sebuah *open source* yang berbasis Linux. Raspberry Pi 3 tidak menggunakan *hard disk*, namun menggunakan SD Card untuk proses *booting* dan penyimpanan data jangka-panjang. Sistem operasinya ditanam pada sebuah *secure digital flash card*, yang menjadikannya sangat mudah untuk diganti dan ditukar[1].

Raspberry Pi 3 adalah generasi ketiga dari Raspberry Pi, menggantikan Raspberry Pi 2 Model B pada Februari 2016. Raspberry Pi3 memiliki bentuk yang identik dengan Raspberry Pi 2 sebelumnya (dan Pi 1 Model B +) dan memiliki kompatibilitas lengkap dengan Raspberry Pi 1 dan 2. Pada perangkat terbarunya ini Raspberry menambahkan fitur *built-in wireless* dan *processor* yang lebih bertenaga yang belum pernah dimiliki pada versi sebelumnya[1]. Gambar dan spesifikasi Raspberry Pi 3 dapat dilihat pada **Gambar 2.2** dan **Tabel 2.1**.



**Gambar 2.2** Raspberry Pi 3

(Sumber: Alliedelec.com)

**Tabel 2.1** Spesifikasi Raspberry Pi 3

<b>Parameter</b>	<b>Spesifikasi</b>
SoC ( <i>System on Chip</i> )	BCM2837
<i>Procesor</i>	1.2GHz 64-bit quad-core ARMv8 CPU
<i>Memory/RAM</i>	1 GB SDRAM 400MHz
GPU	VideoCore IV 3D graphics core
<i>Wireless Adapter/LAN</i>	802.11n Wireless LAN
<i>Bluetooth</i>	Bluetooth 4.1 (built in), Bluetooth Low Energy(BLE)
GPIO	40 Pin
<i>Port USB</i>	4 USB <i>Ports</i>
<i>Card Storage</i>	Micro SD <i>card slot (now push-pull rather than pushpush)</i>
Jaringan	Ethernet Port
External Audio and Video	Full HDMI port, Camera interface (CSI), Display interface (DSI), Combined 3.5mm audio jack and composite video
Sistem Operasi	Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS
Dimensi	3,37" x 2,21" x 0,83"
Berat	45 gr

### 2.2.1 Arsitektur Raspberry Pi 3

Raspberry pi 3 memiliki arsitektur yang dijelaskan sebagai berikut[4]:

1. *USB port*

Jumlah *USB port* pada tipe-tipe Raspberry pi 3 berbeda, Tipe A memiliki buah *USB port* dan Tipe B memiliki 4 buah *USB port*. *USB port* yang difungsikan merupakan USB 2.0 memiliki arus maksimum sebesar 100mA jangka pendek.

2. *LAN Port*

Sebuah Raspberry pi 3 terdidapat 1 *LAN port* yang difungsikan mengakses jaringan.

3. *CSI Header*

Bagian belakang *LAN port* terdidapat sebuah *CSI header* yang terdiri dari 15 pin *flat flex* dan terhubung pada GPU. *CSI header* sebagai standar *interface* serial yang didapat dihubungkan pada suatu kamera CSI-COMPLIANT.

4. *DSI Header*

*DSI (Display Serial Interface)* secara luas didapat difungsikan modul LCD. Seperti halnya *CSI*, *DSI* juga terhubung dengan GPU. Apabila *DSI* difungsikan dengan tambahan *Inter-Integrated Circuit Bus (I2C bus)* maka akan memberikan kemampuan *touch-interface*.

5. *SD Card Slot*

Media penyimpanan pada sebuah Raspberry pi 3 merupakan *card*. Dimana *card* tersebut sebagai media penyimpan OS yang akan difungsikan Raspberry pi 3. Jenis-jenis *card* yang didapat difungsikan *SD/MMC/SDIO*.

6. *GPIO Headers*

*GPIO (General-Purpose Input/Output)* merupakan pin generik berjumlah 26 pin. Konektor *GPIO* difungsikan inisial "P1-XX" agar tidak merancukan penggunaannya, "XX" tersebut menandakan letak posisi pin tersebut. Jika melihat bagian bawah PCB pada *GPIO header* didapat sebuah label bertuliskan "Pi", itu menandakan pin 1 *GPIO* atau inisialnya Pi-01. Berhadapan dengan itu merupakan P1-02. Pin di akhir kebalikan P1-01

merupakan P1-25 dan pin di akhir kebalikan P1-02 merupakan P1-26. Beberapa pin diberi label seperti “NC” atau “DNC”. Itu berarti “No Connect” atau “DoNot Connect” dan pin yang berlabel seperti itu tidak dihubungkan dengan apapun.

#### 7. LED

LED berfungsi sebagai indikator dan menyatakan status atau kondisi sebuah Raspberry pi 3. Indikator LED menyatakan status Raspberry Pi 3 penjelasannya dapat dilihat pada **Tabel 2.2**.

**Tabel 2.2** Indikator LED Raspberry Pi 3

<b>No. Pin</b>	<b>LED Color</b>	<b>Description</b>
D5	<i>Green</i>	<i>System okay/SD card access</i>
D6	<i>Red</i>	Power 3.3V
D7	<i>Green</i>	<i>Full duplex; half duplex if the LED is off</i>
D8	<i>Green</i>	<i>Link activity for the LAN</i>

#### 8. Pi Brains

Pi Brains Raspberry pi 3 menggunakan ARM1176JZF-S 700MHz, versi ARM yang dipakai merupakan arsitektur ARMv6.

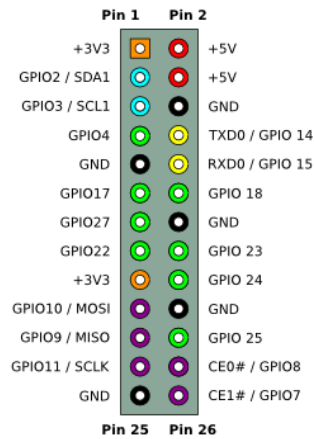
#### 9. Memory Chips

Raspberry pi 3 Tipe A sebesar 256MB dan Tipe B memiliki 512MB. *Memory chip* ini tidak utuh difungsikan sebagai RAM, tetapi penggunaan memorinya dibagi dengan penggunaan memori GPU.

### 2.2.2 Konfigurasi Pin GPIO Raspberry Pi 3

Raspberry Pi 3 memiliki 40 Pin[4]. Pin GPIO merupakan pin yang digunakan untuk control melalui bahasa pemrograman Python. Python dapat diatur pin aktif atau mati maupun nyala berdasarkan kondisi tertentu dengan program yang dibuat. Pin 3.3V dan 5V merupakan pin yang berfungsi untuk memberikan tegangan ke komponen. Pin GND (*Ground*) pin ini menghubungkan ke pin *ground* atau negatif

(-) pada led, sensor, motor maupun relay[5]. Berikut konfigurasi pin GPIO raspberry pi 3 dapat dilihat pada **Gambar 2.3**.



**Gambar 2.3** Konfigurasi Pin GPIO Raspberry Pi 3

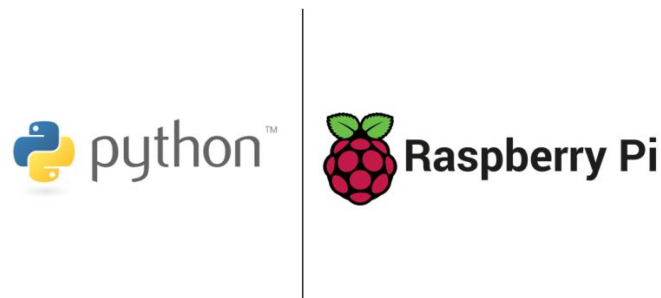
(Sumber: raspberrypitgs.blogspot.com)

### 2.2.3 Python IDE

Python IDE (*Integrated Development Environment*) adalah *software* yang membantu developer untuk mengembangkan aplikasi dengan menggunakan bahasa pemrograman Python yang dilengkapi GUI. Python adalah sebuah bahasa pemrograman yang bisa digunakan pada beberapa platform (multiplatform), dan bersifat *open-source*, pertama kali dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Belanda. Bahasa ini dikategorikan sebagai bahasa tingkat tinggi (*very-high-level language*) dan merupakan bahasa berorientasi objek yang dinamis (*object-oriented-dynamic language*)[6].

Hal utama yang membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program. Python memiliki aturan yang berbeda dengan bahasa lain, seperti indentifikasi, tipe data, *tuple*, dan *dictionary*. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi seperti Linux, Windows, Unix, Symbian dan masih banyak lagi[6].

Python telah digunakan pada berbagai aplikasi saat ini, contohnya adalah BitTorrent, Yum, Civilization 4, bahkan saat ini Python merupakan bahasa resmi dari Raspberry Pi. Kata “Pi” dalam Raspberry Pi merujuk pada kata Python. Python mendukung beberapa modul khusus untuk Raspberry Pi seperti modul picamera, dan modul GPIO[4]. Gambar logo Python dapat dilihat pada **Gambar 2.4**.



**Gambar 2.4** Logo Python

(Sumber: medium.com)

### 2.3 Pengolahan Citra

Pengolahan citra atau *image processing* merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu[8]. Pengolahan ini umumnya dilakukan pada nilai intensitas dari sinyal citra. Pengolahan citra digital dapat dilakukan pada domain asalnya yakni domain spatial, maupun domain frekuensinya[2].

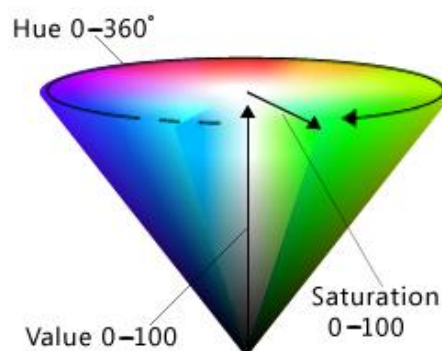
Pada awalnya image processing ini dilakukan untuk memperbaiki kualitas gambar, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari

suatu citra maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision*[12].

HSV (*Hue*, Model warna HSV merupakan model warna yang diturunkan dari model warna RGB, sehingga untuk mendapatkan warna HSV harus melakukan proses konversi warna dari RGB ke HSV. HSV merupakan salah satu cara untuk mendefinisikan warna yang didasarkan pada roda warna. HSV merupakan kepanjangan dari *Hue Saturation* dan *Value*. Model warna HSV ini dipandang lebih dekat dari model warna RGB dalam mendeskripsikan sensasi warna oleh mata manusia[7].

### 2.3.1 HSV (*Hue, Saturation, Value*)

Hue merupakan suatu ukuran panjang gelombang dari warna utama, hue mempunyai ukuran berkisar antara 0-255. 0 mewakili warna merah hingga melalui suatu spektrum kembali bernilai 256 atau kembali menjadi warna merah kembali. Saturation merupakan suatu proses untuk meningkatkan kecerahan warna yang di dasari dari jumlah hue murni pada warna akhir. *Value* merupakan sebuah ukuran seberapa besar kecerahan dari suatu warna. Apabila warna itu memiliki ukuran 100% maka akan terlihat sangat cerah, dan ketika 0% maka akan terlihat gelap[7].



**Gambar 2.5** Model Warna HSV

(Sumber: commons.wikimedia.org)

Penjelasan hexacone pada **Gambar 2.5** adalah sebagai berikut[9]:



- a. *Hue* (H) : Jenis-jenis pada warna (misalnya merah, hijau, atau kuning). Ini direpresentasikan juga sebagai tingkat sudut yang nilainya berkisar dari 0° sampai 360° (meskipun untuk beberapa aplikasi dinormalisasi dari 0% sampai 100%)
- b. *Saturation* (S) : saturasi (saturation) adalah ukuran kemurnian warna. Misalnya warna pada spectrum warna yang semua warnanya merah tanpa warna putih, maka itu merupakan saturasi penuh. Apabila ada penambahan warna putih kedalam warna tersebut, maka akan menghasilkan warna-warna lain dan warna bergeser dari warna merah menjadi merah muda. Nilainya berkisar dari 0% sampai 100%.
- c. *Value* (V) : direpresentasikan sebagai tinggi pada poros hitam putih. Kemungkinan jarak nilai berkisar dari 0% sampai 100%. Nilai 0 selalu hitam. Berdasarkan pada saturation, 100 bisa menjadi putih atau tingkat saturation yang lebih bahkan kurang.

Suatu warna dengan nilai *value* 100% akan tampak secerah mungkin dan suatu warna dengan nilai *value* 0 akan tampak segelap mungkin. Sebagai contoh jika *hue* adalah merah dan *value* bernilai tinggi maka warna kelihatan cerah tetapi ketika nilai *value* rendah maka warna tersebut akan kelihatan gelap.

Proses untuk mendapatkan nilai dari setiap warna yang ingin ditampilkan adalah dengan melakukan konversi ruang warna R-G-B (*Red, Green, Blue*) ke ruang warna H-S-V (*Hue, Saturation, Value*). Nilai R, G, B menggunakan skala 0..255, sedangkan H, S, dan V menggunakan skala 0..1. Diasumsikan koordinat-koordinat R, G, B [0,1] adalah berurutan *red, green, blue* dalam ruang warna RGB, dengan *max* adalah nilai maksimum dari nilai *red, green, blue*, dan *min* adalah nilai minimum dari nilai *red, green, blue*. Diperlukan normalisasi terlebih dahulu dengan rumus berikut:

$$r = \frac{R}{R+G+B} \dots\dots\dots (3)$$

$$g = \frac{G}{R+G+B} \dots\dots\dots (4)$$

$$b = \frac{B}{R+G+B} \dots\dots\dots (5)$$

Setelah nilai r, g, b sudah di normalisasi, maka rumus transformasi RGB ke HSV sebagai berikut:

$$h(\text{hue}) = \begin{cases} 0, & \text{jika } \max = \min \\ 60^\circ \times \left( \frac{G - B}{\max - \min} \bmod 6 \right), & \text{jika } \max = R \\ 60^\circ \times \left( \frac{B - R}{\max - \min} + 2 \right), & \text{jika } \max = G \\ 60^\circ \times \left( \frac{R - G}{\max - \min} + 4 \right), & \text{jika } \max = B \end{cases} \dots\dots\dots (6)$$

$$s(\text{saturation}) = \begin{cases} 0, & \text{jika } \max = \min \\ \frac{\max - \min}{V}, & \text{otherwise} \end{cases} \dots\dots\dots (7)$$

$$V(\text{value}) = \max \dots\dots\dots (8)$$

### 2.3.2 Canny Edge Detection

*Canny edge detection* dikemukakan oleh Jhon Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal. Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek), dan hanya memberikan satu tanggapan untuk satu tepi.

Metode *Canny* memaksimalkan deteksi titik tepi yang benar dengan cara memaksimalkan signal to noise dan tepi yang terdeteksi harus sedekat mungkin dengan tepi yang nyata. Dilakukannya pendeteksian tepi untuk meningkatkan penampakan garis batas suatu daerah yang terdapat pada objek di dalam suatu citra[10]. Operator *Canny Edge detection* menggunakan *Gaussian Derivative Kernel* untuk menyaring kegaduhan dari citra awal untuk mendapatkan hasil deteksi tepi yang halus. Metode yang paling bagus untuk deteksi tepi adalah *Canny Edge detection* karena mendeteksi tepian lebih jelas dan noisinya lebih sedikit.

Langkah-langkah dalam melakukan deteksi tepi *Canny Edge Detection*:

- a. Menghilangkan *noise* yang terdapat pada objek dengan menggunakan *averaging* dan *Gaussian Blur*. *Averaging* berfungsi untuk mencari nilai rata-rata dari semua *pixel*. *Pixel* yang digunakan adalah 3x3

$$K = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Pada *Filter Gaussian* digunakan sebuah matriks 5x5 yang digunakan untuk menghilangkan *noise* yang berada pada tepian dengan nilai  $\sigma = 1.4$

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

- b. Mencari nilai *intensitas gradient* untuk memperoleh *gradient* vertical dan horizontal pada setiap tepi akan menghasilkan persamaan seperti pada persamaan 9.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

$$G = \sqrt{G_x^2 + G_y^2} \dots\dots\dots (9)$$

- c. Mencari direksi untuk mendapatkan *angle* pada tepian. Dapat dilihat pada persamaan 10.

$$\theta = \arctan \frac{G_y}{G_x} \dots\dots\dots (10)$$

### 2.3.3 Shape Detection

*Shape detection* merupakan metode untuk mengetahui bentuk objek yang telah dideteksi dengan menggunakan perhitungan titik puncak (*vertice*) dari setiap tepi yang telah terdeteksi oleh *canny edge detection*. Metode pengenalan bentuk adalah gabungan dari fungsi approximation, luas kontur, dan aspek rasio. Fungsi approximation adalah fungsi untuk menemukan jumlah titik puncak atau vertice pada sebuah bangun datar. Kedua adalah luas kontur yang berasal dari jumlah pixel pada objek. Dan terakhir aspek rasio adalah perbandingan panjang dan lebar dari suatu gambar dapat dilihat pada persamaan 11 sampai 14

- a. Mengidentifikasi kategori *square*,  $ar = \frac{width}{height} \dots\dots\dots (11)$

b. Mengenali bentuk bulat,  $radius = \frac{width}{2}$  ..... (12)

c. Mencari nilai aspek rasio,  $spect\ ratio = 1 - \frac{width}{height}$  ..... (13)

d. Menemukan perbandingan luas dengan *bounding box*,  
 $skala = 1 - \frac{luas\ kontur}{\pi\ radius^2}$  ..... (14)