

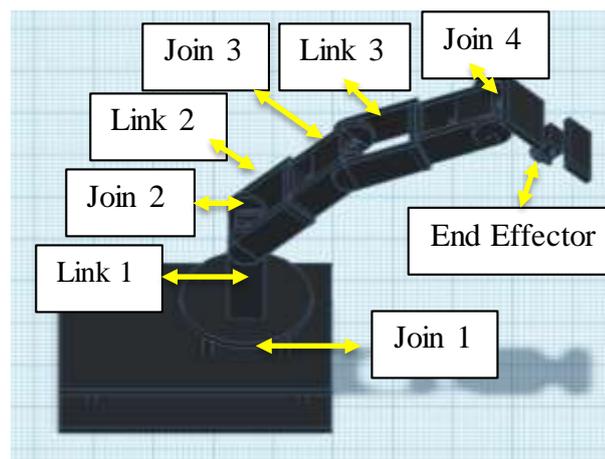
BAB II

TINJAUAN PUSTAKA

2.1 *Arm Robot*

Arm robot (robot lengan) adalah mekanik dan gerakan robot yang dibuat menyerupai anatomi lengan manusia, *Arm robot* minimal memiliki sendi (*joint*) dan pencengkram (*gripper*) yang disesuaikan dengan kebutuhan alat pemilah tomat. Didalam *arm robot* memiliki komponen-komponen lain sebagai pembentukannya, seperti actuator, sensor dan kontroler. *Arm robot* pula mengenal derajat kebebasan (DOF) yang menentukan banyaknya gerakan pada *arm* itu sendiri, seperti pada **Gambar 2.1**

Arm robot berfungsi untuk mengambil buah tomat (objek), kemudian dapat meletakkan objek tersebut pada tempat lain yang bisa dilakukan secara manual ataupun otomatis sesuai dengan kondisi yang diinginkan, pada *arm robot* juga memungkinkan diletakkannya sensor-sensor seperti sensor warna, sensor jarak maupun sensor lainnya yang bisa mendukung kinerja alat pemilah tomat secara otomatis.



Gambar 2.1 *Arm Robot*
(Dokumen Pribadi, 2022)

Arm robot memiliki beberapa bagian yaitu:

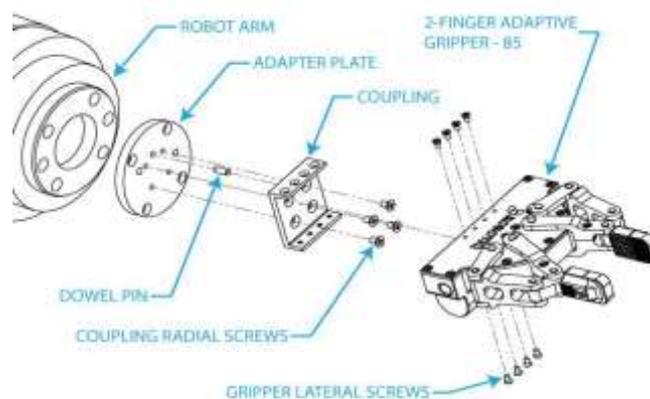
1. *Link*: bagian robot yang bentuknya tetap dan dapat bergerak. *Link* biasanya dihubungkan dengan *joint*.

2. *Joint*: penghubung *link* dengan *link* atau *base* yang dapat bergerak aktif (biasanya terdapat aktuator). Pada robot lengan, *joint* ini adalah motor servonya.
3. *End Effector* (ujung): titik akhir yang menghubungkan *arm robot* dengan objek. *End effector* pada *arm robot* terletak dibagian ujung dari *arm robot* yaitu *gripper* yang menyentuh langsung dengan objek kerja *arm robot*.
4. *DoF (Degree of Freedom)*: jumlah gerakan *independent* yang dapat dilakukan oleh suatu robot.

2.1.1 Gripper

Gripper adalah sebuah pendekatan terhadap pola gerakan dari salah satu bagian tubuh manusia yaitu jari tangan. Dimana dari segi desain bentuk, gerak dan sistem kendali disesuaikan dengan bentuk jari tangan, pola gerakan, serta otot sebagai pengendali gerakan.

Pada proyek tugas akhir ini, gripper didesain untuk dapat melakukan gerakan pemilahan buah tomat dengan optimal, tampilan *gripper* dapat terlihat pada **Gambar 2.2**

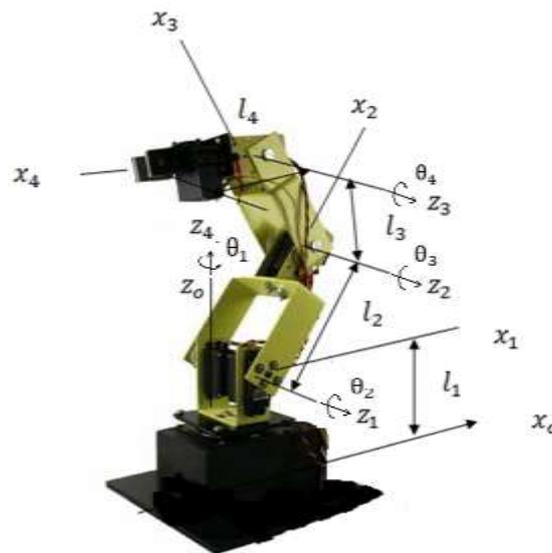


Gambar 2.2 *Gripper*

Effector robot gripper dilakukan dengan menentukan koordinat obyek dan sudut masing-masing *joint* robot untuk mencapai koordinat obyek menggunakan *forward kinematic* pada setiap *joint*, dari simulasi *software* pengendali robot yang kemudian akan dibandingkan secara implementasi pada gerakan *gripper* sehingga didapatkan nilai *error* dari koordinat yang dicapai *gripper*.

2.1.2 Konvensi DH

Konvensi Denavit-Hartenberg (DH) umumnya digunakan dalam analisis kinematika *manipulator robot*. Yang berdasarkan lampiran kerangka koordinat pada setiap sendi, Seperti pada **Gambar 2.3** dan menentukan empat parameter yang dikenal sebagai parameter DH untuk setiap tautan, serta memanfaatkan parameter ini untuk membuat tabel DH. dengan melakukan perhitungan Matriks transformasi antara yang berbeda kerangka koordinat diperoleh.



Gambar 2.3 Robot Arm dan DH Parameter

Tujuan utamanya adalah untuk mengontrol posisi dan orientasi EE atau *gripper* di ruang kerjanya. yang akan menurunkan hubungan antara variabel bersama dan posisi, dan orientasi *gripper*, menggunakan metode DH. [7]

Tabel 2.1 DH Parameter

Joint	θ_i (°)	d_i (m)	a_i (m)	α_i (°)
1	θ_1	0.19	0	90
2	θ_2	0	0.105	0
3	θ_3	0	0.105	0
4	θ_4	0	0.18	0

Keterangan:

d_i = link offset

a_i = panjang link

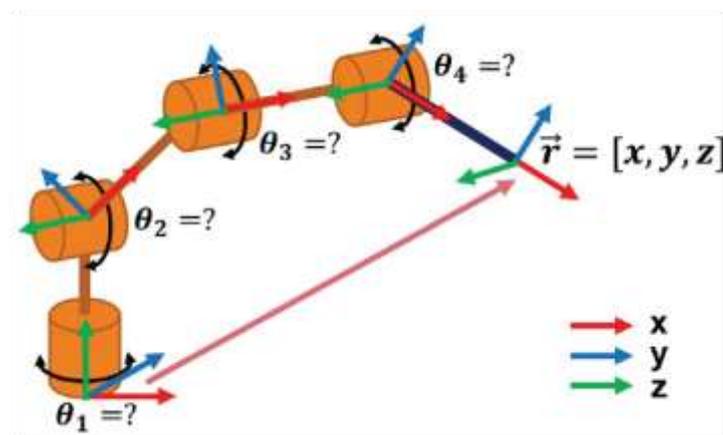
θ_i = sudut joint

α = sudut putaran link

Dengan menerapkan notasi Denavit-Hartenberg (DH) untuk koordinat sambungan, tabel DH dapat dibuat sebagai tercantum di atas dalam **Tabel 2.1**.

2.1.3 Kinematik

Kinematik adalah metode pemodelan gerak robot tanpa mempertimbangkan gaya yang diberikan untuk menggerakkan robot. Analisis kinematika dibagi menjadi kinematika maju (*forward kinematic*) dan kinematika terbalik (*invers kinematic*).



Gambar 2.4 Struktur *Arm Robot* 4DOF

Dengan menerapkan pemodelan kinematika, desain robot dapat menjadi efisien dengan mempertimbangkan bagaimana robot bergerak pada sumbu x , y , dan z . Robot lengan yang dipertimbangkan dalam penelitian tugas akhir ini ditunjukkan pada **Gambar 2.4** adalah sudut antara sudut *joint* pertama dan alas, 2 dan 3 adalah sudut kedua dan ketiga, dan 4 adalah sudut *end-effector* atau *grippers*. [8]

2.1.4 Forward Kinematic

Analisis kinematika dari *arm robot manipulator* dimulai dari kinematika maju (*forward kinematic*). Yang mana *Forward kinematics* menggunakan parameter Denavit Hartenberg untuk menentukan posisi akhir dari *end effector*. Berikut ini adalah perhitungan untuk menentukan posisi *end effector* berdasarkan sudut yang telah ditentukan.

$${}^0_4T = {}^0_1T \times {}^1_2T \times {}^2_3T \times {}^3_4T$$

$${}^0_1T = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2 \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2 \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3 \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3 \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & a_4 \cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & a_4 \sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_2T = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2 \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2 \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_2T = \begin{bmatrix} \cos\theta_1 \cdot \cos\theta_2 & -\cos\theta_1 \cdot \sin\theta_2 & \sin\theta_1 & \cos\theta_1 \cdot a_2 \cdot \cos\theta_2 \\ \sin\theta_1 \cdot \cos\theta_2 & -\sin\theta_1 \cdot \sin\theta_2 & -\cos\theta_1 & \sin\theta_1 \cdot a_2 \cdot \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2 \cdot \sin\theta_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_4T = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & a_3 \cos\theta_3 \\ \sin\theta_3 & 0 & -\cos\theta_3 & a_3 \sin\theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & a_4 \cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & a_4 \sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_3 \cdot \cos\theta_4 - \sin\theta_3 \cdot \sin\theta_4 & -\cos\theta_3 \cdot \sin\theta_4 - \sin\theta_3 \cdot \cos\theta_4 & 0 & \cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \cos\theta_4 \\ \sin\theta_3 \cdot \cos\theta_4 + \cos\theta_3 \cdot \sin\theta_4 & -\sin\theta_3 \cdot \sin\theta_4 + \cos\theta_3 \cdot \cos\theta_4 & 0 & \sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \cos\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_4 = {}^0T_2 \cdot x \cdot {}^2T_4$$

$${}^0T_4 =$$

$$\begin{bmatrix} - & - & - & \cos\theta_1 \cdot \cos\theta_2 (\cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) - \cos\theta_1 \cdot \sin\theta_2 (\sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) + \cos\theta_1 \cdot a_2 \cdot \cos\theta_2 \\ - & - & - & \sin\theta_1 \cdot \cos\theta_2 (\cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) - \sin\theta_1 \cdot \sin\theta_2 (\sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) + \sin\theta_1 \cdot a_2 \cdot \cos\theta_2 \\ - & - & - & \sin\theta_2 (\cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) + \cos\theta_2 (\sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) + a_2 \cdot \sin\theta_2 \cdot d_1 \\ & & & 1 \end{bmatrix}$$

Berdasarkan perhitungan diatas, maka didapatkan rumus untuk menentukan posisi x, y dan z dari *end effector*.

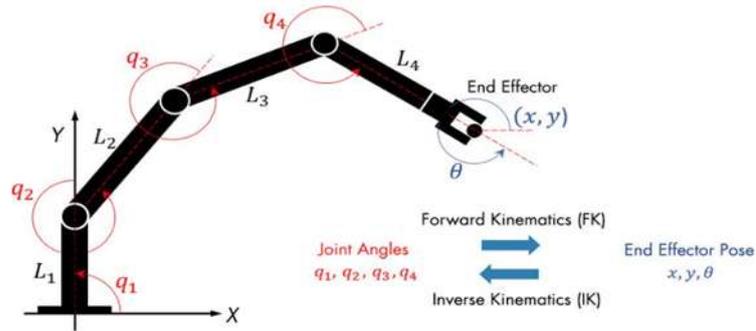
$$\begin{aligned} p_x &= \cos\theta_1 \cdot \cos\theta_2 (\cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) \\ &\quad - \cos\theta_1 \cdot \sin\theta_2 (\sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) \\ &\quad + \cos\theta_1 \cdot a_2 \cdot \cos\theta_2 \end{aligned}$$

$$\begin{aligned} p_y &= \sin\theta_1 \cdot \cos\theta_2 (\cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) \\ &\quad - \sin\theta_1 \cdot \sin\theta_2 (\sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) \\ &\quad + \sin\theta_1 \cdot a_2 \cdot \cos\theta_2 \end{aligned}$$

$$\begin{aligned} p_z &= \sin\theta_2 (\cos\theta_3 \cdot a_4 \cdot \cos\theta_4 - \sin\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) \\ &\quad + \cos\theta_2 (\sin\theta_3 \cdot a_4 \cdot \cos\theta_4 + \cos\theta_3 \cdot a_4 \cdot \sin\theta_4 + a_3 \cdot \cos\theta_3) \\ &\quad + a_2 \cdot \sin\theta_2 \cdot d_1 \end{aligned}$$

2.1.5 Invers Kinematic

Invers Kinematic adalah perhitungan matematika dengan sebuah *input variable* yaitu titik kordinat yang dituju, sehingga menghasilkan *output* sudut untuk menggerakkan setiap *joint* pada *arm robot*, adapun tampilan posisi kerangka *invers kinematic* terlihat pada **Gambar 2.5**,



Gambar 2.5 *Forward kinematic* dan *Invers Kinematic*

(<https://ww2.mathworks.cn/discovery/inverse-kinematics.html>)

Invers kinematic mengharuskan kita untuk mengetahui variable gabungan $\theta_1, \theta_2, \theta_3$ dan θ_4 untuk EE posisi tertentu [dy, dx dan dz] dan orientasi Φ , yang didapatkan pada persamaan 1 s/d 7, dengan cara membagi, mengkuadratkan, menjumlahkan dan menggunakan beberapa rumus trigonometri.

$$\theta_1 = \tan^{-1} \left(\frac{dy}{dx} \right) \dots\dots\dots (1)$$

$$\theta_2 = \tan^{-1} \left(c, \pm \sqrt{r^2 - c^2} \right) - \tan^{-1} (a, b) \dots\dots\dots (2)$$

$$\theta_3 = \cos^{-1} \left(\frac{A^2 + B^2 + C^2 - l_2^2 - l_3^2}{2 l_2 l_3} \right) \dots\dots\dots (3)$$

Dimana

$$a = l_3 \sin \theta_3, b = l_2 + l_3 \cos \theta_3, c = dz - l_1 - l_4 \sin \phi, \text{ and } r = \sqrt{a^2 + b^2} \dots\dots\dots (4)$$

Selain itu

$$A = (dx - l_4 \cos \theta_3 \cos \phi), B = (dy - l_4 \sin \theta_3 \cos \phi), \text{ and } C = (dz - l_1 - l_4 \sin \phi) \dots\dots\dots (5)$$

Setelah ditentukan θ_1 , θ_2 dan θ_3 maka kita dapat mencari θ_4 dari orientasi EE berikut;

$$\theta_4 = \phi - \theta_2 - \theta_3 \dots\dots\dots (6)$$

Dalam perancangan alat pemilah tomat ini, *invers kinematic* hanya untuk menemukan 1 hingga 3. Parameter 4 terkait dengan *end-effector*, dan FLC memutuskan gerakan *end-effector* berdasarkan *input* dari sensor kamera *Pixy Cmcum5* dan sensor Ultrasonik yang terpasang pada sistem robot.

Forward Kinematic dalam tugas akhir ini dihitung dengan metode aljabar untuk menemukan *invers* dari (3) dan mencapai 1 hingga 3. Perhitungan 1 sampai 3 diberikan oleh rumus dibawah ini.[8]

$$C\theta_2 = \frac{px^2+py^2-a_1^2-a_2^2}{2a_1a_2} \dots\dots\dots (7)$$

$$\theta_2 = \text{atan2} \left(\pm \sqrt{1 - \left(\frac{px^2+py^2-a_1^2-a_2^2}{2a_1a_2} \right)^2}, \frac{px^2+py^2-a_1^2-a_2^2}{2a_1a_2} \right) \dots\dots\dots (8)$$

$$\theta_1 = \text{atan2}(p_x^2 + p_y^2 - (a_2 \cos\theta_2 + a_1)^2 a_2 \cos\theta_2 + a_1) \dots\dots\dots (9)$$

$$\theta_3 = \pm \text{atan2}(p_x^2 + p_y^2 - (a_2 \cos\theta_{12} + a_1)^2 a_2 \cos\theta_{12} + a_1) \dots\dots\dots (10)$$

$$\theta_4 = \Phi - (\theta_2 - \theta_3) \dots\dots\dots (11)$$

2.2 Sensor Pixy Cmcum5

Pixy Cmcum5 dirancang untuk menangkap citra dan digunakan sebagai *co-processor* untuk Arduino atau mikrokontroler sejenisnya. Berbasiskan *prosesor* NXP LPC4330 yang mampu mengakomodasi pengolahan gambar *digital* dengan cepat dan data tersebut dapat dibaca dengan mudah oleh *mikrokontroler*. Selain itu, *Cmcum5* juga dibekali dengan sensor kamera Omnivision OV9715 yang mempunyai resolusi 1.3 MegaPixel. Gambar *Pixy Cmcum5* dapat dilihat pada **Gambar 2.6** [5]

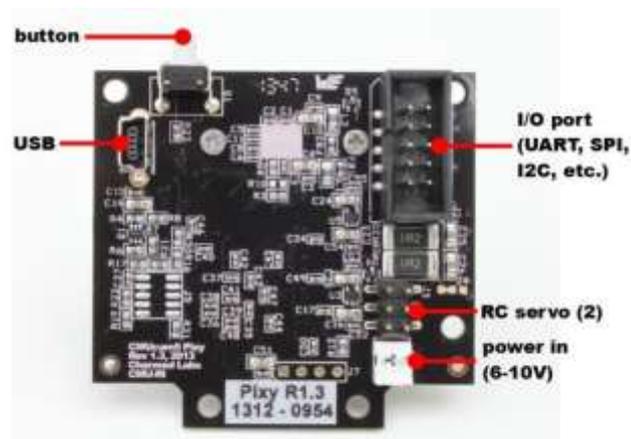


Gambar 2.6 *Pixy Cmucam5*[5]

Pixy Cmucam5 diprogram untuk mengirimkan informasi berupa data gambar, sehingga mikrokontroler tidak terbebani dengan proses pembacaan data. Proses pengiriman data pada *Pixy Cmucam5* dapat melakukan berbagai jalur komunikasi data yang meliputi: UART serial, SPI, I2C, *digital out* dan *analog out*, yang terlihat pada **Gambar 2.7**

Pixy Cmucam5 juga menggunakan warna dan saturasi sebagai sasaran utama pada pendeteksian gambar. Hal ini berarti bahwa pencahayaan atau *exposure* tidak akan mempengaruhi deteksi sensor pada objek.

Sensor ini juga mampu mengingat 7 warna yang berbeda dan menemukan ratusan benda pada saat bersamaan dengan kecepatan 50fps. *Pixy Cmucam5* memiliki aplikasi *open source* yang disebut *PixyMon*. Terlihat pada **Gambar 3.4** [6]



Gambar 2.7 Bagian-Bagian *Pixy Cmucam5*

(<http://cmucam.org/projects/cmucam5>, 2017)

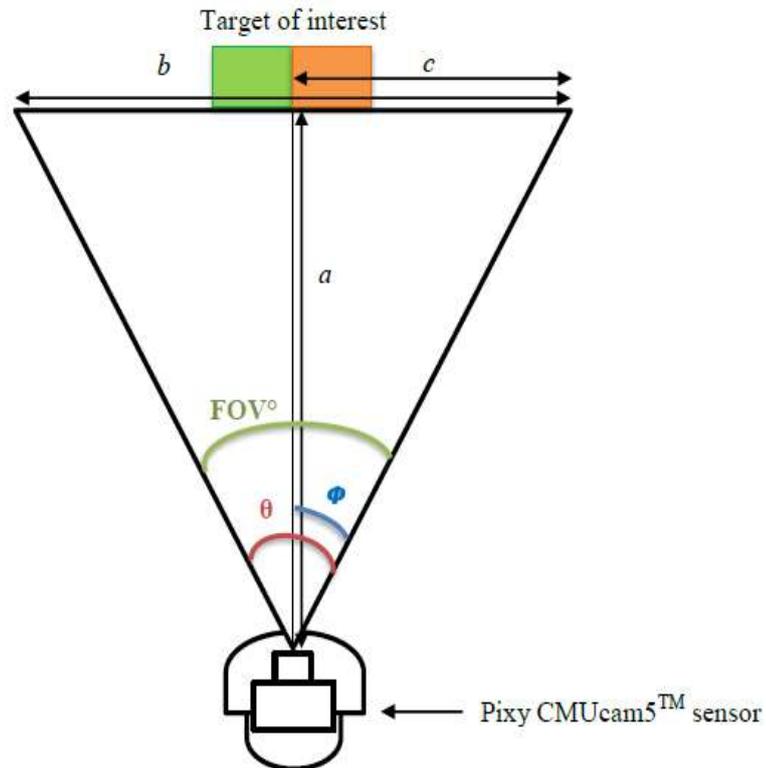
Spesifikasi *Pixy Cmcam5*

1. *Processor* : NXP LPC4330, 204 MHz, *dual core*.
2. *Image sensor* : Omnivision, OV9715, ¼", 1280x800.
3. *Lens field-of-view* : 750 horizontal dan 470 vertikal.
4. Tipe Lensa : standard M12 (*several different types available*).
5. Konsumsi daya : 140mA.
6. *Power input* : USB input (5V) atau *unregulate input* (6V sampai 10V).
7. RAM : 264 Kb.
8. *Flash* : 1 Mb.
9. *Available data output* : UART *serial*, SPI, I2C, USB, *digital*, *analog*.
10. Dimensi : 2,1" x 2,0" x 1,4".
11. Berat : 27 gram.

Pixy Cmcam5 menggunakan algoritma berbasis penyaringan warna untuk mendeteksi benda-benda. Metode penyaringan berdasarkan warna yang populer karena cepat, efisien, dan relatif kuat. Penyaringan warna yang biasa dikenal menggunakan RGB (merah, hijau, dan biru) untuk mewakili warna. *Pixy* menghitung warna (*hue*) dan saturasi dari setiap piksel RGB dari sensor gambar dan menggunakan ini sebagai parameter penyaringan utama. Perubahan pencahayaan dan paparan dapat memiliki *frustating effect* pada algoritma penyaringan warna.

Adapun penerapan sensor kamera *Pixy Cmcam5* pada alat pemilah buah tomat ialah untuk mendeteksi objek berdasarkan warna yang juga terhubung pada *robot arm* sehingga *gripper* pada *arm* dapat mendeteksi objek tepat sesuai perintah.

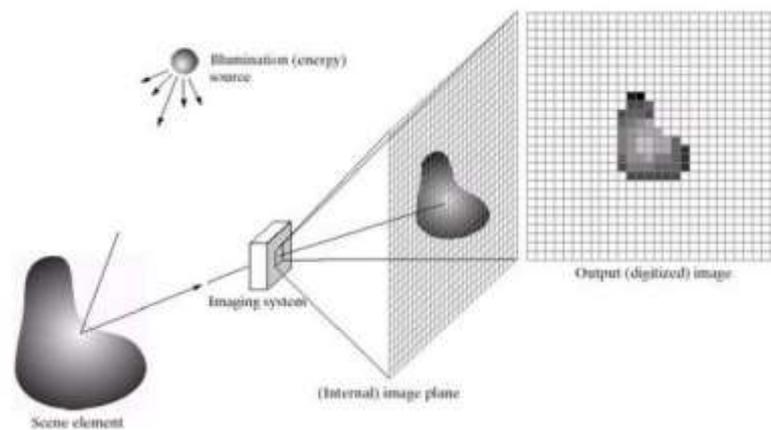
Berikut visualisasi cara kerja sensor kamera *Pixy CmuCam5*, Terlihat pada **Gambar 2.8**



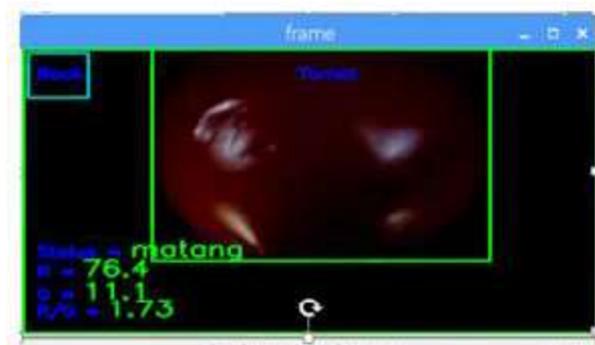
Gambar 2.8 Cara Kerja *Pixy CmuCam5* [1]

2.2.1 Pengolahan Citra

Kata citra atau yang lebih dikenal secara umum dengan kata “gambar” dapat didefinisikan sebagai fungsi dua dimensi (2D), $f(x,y)$, dimana x dan y adalah koordinat spasial, dan amplituda dari f pada setiap kombinasi atau pasangan koordinat (x,y) merupakan tingkat kecerahan dari sebuah citra pada titik tersebut. Suatu citra diperoleh dari penangkapan pantulan sinar yang dipantulkan oleh suatu objek dan cahaya yang ditangkap disajikan dalam bentuk citra. Alat-alat optik juga mempunyai proses yang sama dalam mendapatkan suatu citra atau gambar seperti kamera. Dalam proses akuisisi citra mengalami proses citra biner dapat dilihat seperti pada **Gambar 2.9**



Gambar 2.9 Contoh Proses Akuisisi Citra Biner



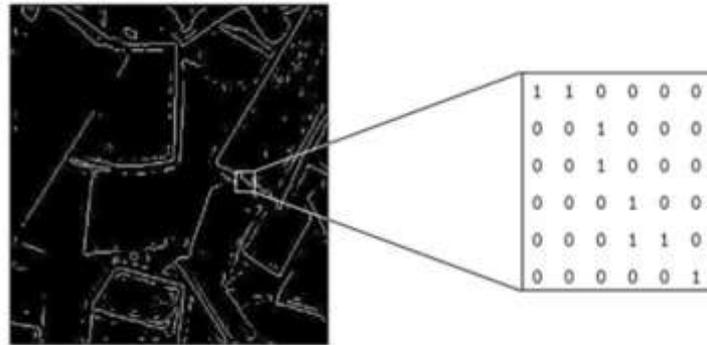
Gambar 2.10 Proses Pendeteksian Citra pada Tomat

Citra *digital* adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu. Format data citra *digital* berhubungan erat dengan warna. Pada kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data *digital* merepresentasikan warna dari citra yang diolah seperti pada **Gambar 2.10**. Format citra *digital* yang banyak dipakai adalah Citra Biner (*monokrom*), Citra Skala Keabuan (*gray scale*), Citra Warna (*true color*), dan Citra Warna Berindeks.

2.2.2 Citra Digital Bertipe Biner

Citra biner adalah citra *digital* yang terdiri dari piksel-piksel, memiliki 2 warna yaitu hitam dan putih, citra ini menyimpan tiap piksel dalam bit tunggal 0 dan 1. yang mana nilai 0 menjadi representasi dari warna hitam sedangkan nilai 1 merupakan representasi dari warna putih. Karena itu setiap piksel hanya berukuran 1 bit yang membuat citra tipe biner sangat efisien dalam proses penyimpanan.

Gambar 2.11 adalah contoh tipe citra biner, yang mana warna putih mewakili piksel tepi dan warna hitam mewakili latar belakang.



Gambar 2.11 Citra Digital Tipe *Biner*

Berikut hubungan antara bit dan rentang nilai *digital number* (nomor digitalnya) atau DN pada suatu citra;

- 8bit = 2 pangkat 8 = 256 informasi dimana nilai nomor digitalnya 0 - 255
- 7bit = 2 pangkat 7 = 128 informasi dimana nilai nomor digitalnya 0 - 127
- 6bit = 2 pangkat 6 = 64 informasi dimana nilai nomor digitalnya 0 - 63
- 5bit = 2 pangkat 5 = 32 informasi dimana nilai nomor digitalnya 0 - 31
- 4bit = 2 pangkat 4 = 16 informasi dimana nilai nomor digitalnya 0 - 15
- 3bit = 2 pangkat 3 = 8 informasi dimana nilai nomor digitalnya 0 - 7
- 2bit = 2 pangkat 2 = 4 informasi dimana nilai nomor digitalnya 0 - 3
- 1bit = 2 pangkat 1 = 2 informasi dimana nilai nomor digitalnya 0 & 1

Nilai *digital number* jika dilihat dari bilangan biner seperti ini pada derajat keabuan atau *grayscale*, contoh :

- 1bit = 2 informasi dimana nilai nomor digitalnya 0 & 1
- 0 = mati (hitam)
- 1 = hidup (putih)
- 2bit = 4 informasi dimana nilai nomor digitalnya 0 - 3
- 0 0 = 0 (hitam)
- 0 1 = 1 (abu-abu gelap)
- 1 0 = 2 (abu-abu cerah)
- 1 1 = 4 (putih)

- 8bit = 256 informasi dimana nilai nomor *digital* nya 0 - 255
- 00000000 = 0 (hitam)
- 00010000 = 16 (abu-abu gelap)
- 11001000 = 200 (abu-abu cerah)
- 11111111 = 255 (putih)

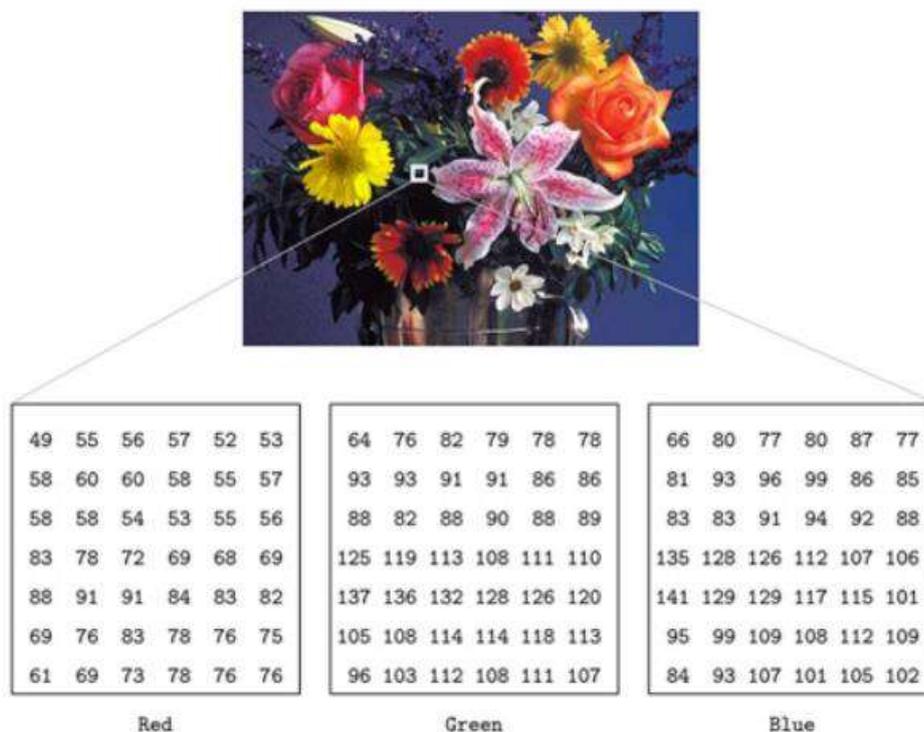
2.2.3 Citra *Digital* Bertipe RGB atau *True Colour*

Citra RGB adalah citra yang nilai intensitas pikselnya tersusun oleh tiga komponen warna, yaitu merah(R), hijau(G) dan biru(B). Setiap komponen warna mempunyai jangkauan nilai sebesar 256 yaitu antara 0 sampai 255 (8 bit).

Artinya, setiap piksel berukuran sebesar 24 bit. Sehingga total kemungkinan warna sebanyak 255^3 .

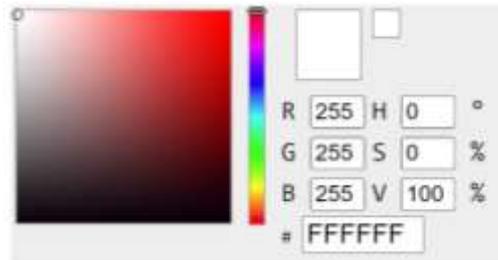
Hal ini yang membuat citra warna RGB disebut dengan citra warna 24 bit.

Gambar 2.12 merupakan contoh dari citra RGB.

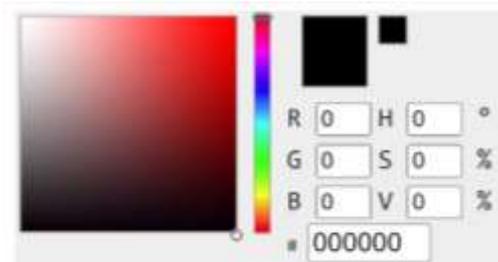


Gambar 2.12 Citra *Digital* Tiper RGB

Berikut tampilan RGB *color picker* dari nilai tertinggi dan nilai terendah RGB.

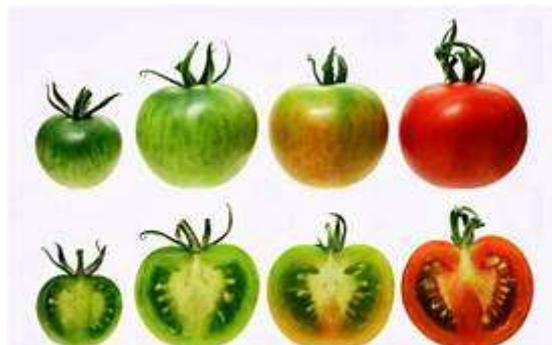


Gambar 2.13 Nilai RGB tertinggi



Gambar 2.14 Nilai RGB terendah

Adapun berikut citra warna RGB sesuai tingkatan warna merah pada buah tomat dan ukurannya.



Gambar 2.15 Tingkatan Warna Buah Tomat



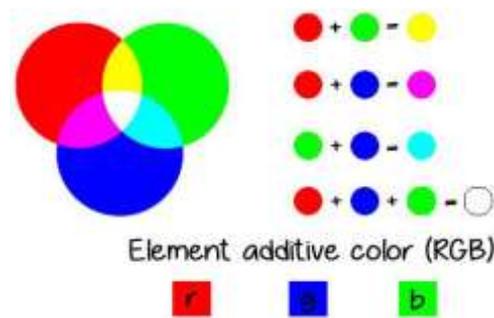
Gambar 2.16 Proses Segmentasi Buah Tomat

2.2.4 Warna

Warna adalah *spektrum* tertentu yang terdapat di dalam suatu cahaya sempurna (warna putih) yang merupakan pantulan tertentu dari cahaya yang dipengaruhi oleh *pigmen* yang terdapat di permukaan benda.

Warna dapat didefinisikan secara obyektif/fisik sebagai sifat cahaya yang dipancarkan, atau secara subyektif/psikologis sebagai bagian dari pengalaman indera pengelihatan. Secara obyektif atau fisik, warna dapat diberikan oleh panjang gelombang. Dilihat dari panjang gelombang, cahaya yang tampak oleh mata merupakan salah satu bentuk pancaran energi yang merupakan bagian yang sempit dari gelombang elektromagnetik.

Warna pada buah tomat pula memiliki warna merah yang ciri khas yang ditentukan oleh tingkatan RGB, Model warna RGB yang terdiri atas 3 buah warna : merah (*Red*), hijau (*Green*) dan biru (*Blue*), yang ditambahkan dengan berbagai cara untuk menghasilkan bermacam-macam warna seperti ditunjukkan pada **Gambar 2.17**



Gambar 2.17 Model Warna RGB

(<http://fotografidesain.com/>)

Tabel 2.2 Daftar Warna RGB

Warna	R (Red)	G (Green)	B (Blue)
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Kuning	255	255	0
Magenta	255	0	255
Cyan	0	255	255
Putih	255	255	255
Hitam	0	0	0
Abu-abu	128	128	128

Kegunaan utama model warna RGB adalah untuk menampilkan citra atau gambar dalam perangkat elektronik, seperti televisi dan komputer, walaupun juga telah digunakan dalam fotografi biasa. Sebelum era elektronik, model warna RGB telah memiliki landasan yang kuat berdasarkan pemahaman manusia terhadap teori trikromatik. Jika masing-masing warna memiliki *range* 0 - 255, maka totalnya adalah $255^3 = 16.581.375$ (16 K) variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap piksel, gambar tersebut juga disebut gambar-bit warna.

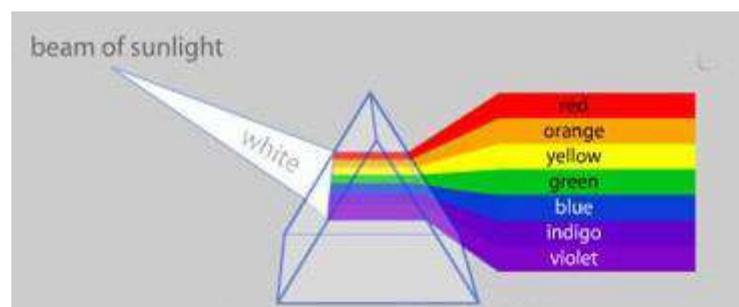
Warna kuning memiliki kadar RGB dengan nilai $R = 255$, $G = 255$, dan $B = 0$. Kadar warna RGB inilah yang nantinya akan diolah oleh sensor kamera *Pixy Cmucam5* untuk mengirimkan data ke arduino.

2.2.5 Ruang Warna (*Colour Space*)

Warna adalah *spectrum* tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Intensitas suatu warna ditentukan oleh panjang gelombang cahaya. Panjang gelombang warna yang masih bisa ditangkap mata manusia berkisar antara 380-780 nanometer. Sebagai contoh warna biru memiliki panjang gelombang 460 nanometer.

Cahaya yang dapat dilihat adalah bukan cahaya dengan satu panjang gelombang melainkan kumpulan panjang gelombang tertentu. Warna terbentuk dari kumpulan gelombang dengan panjang gelombang yang berbeda-beda.

Penyajian campuran elemen dasar untuk menghasilkan warna dinamakan dengan *Color Space* atau Ruang Warna. Warna terbentuk dari kumpulan gelombang dengan panjang gelombang yang berbeda-beda seperti pada **Gambar 2.18** untuk pembentukan *spectrum* warna.



Gambar 2.18 Pembentukan *spectrum* warna

2.2.6 Metode HSV

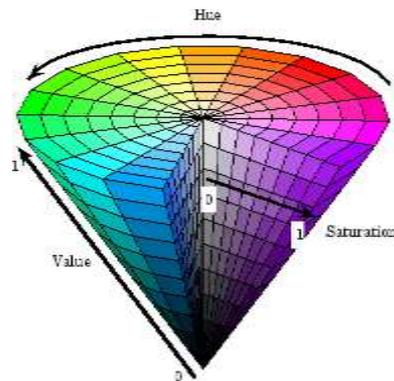
Ruang warna adalah model matematis abstrak yang menggambarkan cara agar suatu warna dapat dipresentasikan sebagai baris angka biasanya dengan nilai-nilai dari tiga atau empat buah warna.

HSV mendefinisikan warna pada terminologi *Hue*, *Saturation*, dan *Value*. HSV merupakan warnawarna yang sama dengan ditangkap oleh indera manusia. Sedangkan warna RGB merupakan campuran dari warna primer.

Model HSV ditunjukkan pada **Gambar 2.19**. Melalui model gambar diatas, HSV memiliki 3 karakteristik pokok, yaitu *Hue*, *Saturation* dan *Value*.

Hue menyatakan *Hue* memiliki sudut 360° yang menunjukkan jenis warna yang dapat ditemukan pada spektrum warna. *Saturation* adalah tingkat kemurnian suatu warna yang biasanya memiliki nilai 0 sampai 1.

Sedangkan *Value* adalah intensitas kecerahan dari suatu warna yang dapat bernilai dari 0 sampai 100%. [2]



Gambar 2.19 Model HSV

Perhitungan konversi RGB menjadi HSV dapat dirumuskan sebagai berikut;

$$H = \tan \left(\frac{3 (G-B)}{(R-G)+(R-B)} \right) \dots\dots\dots (12)$$

$$S = 1 - \frac{\min(R,G,B)}{v} \dots\dots\dots (13)$$

$$V = \frac{R+G+B}{3} \dots\dots\dots (14)$$

Apabila nilai S = 0, maka H tidak dapat ditentukan. Oleh karena itu diperlukan normalisasi RGB terlebih dahulu dengan menggunakan rumus sebagai berikut

$$r = \frac{R}{(R+G+B)} \dots\dots\dots (15)$$

$$g = \frac{G}{(R+G+B)} \dots\dots\dots (16)$$

$$b = \frac{B}{(R+G+B)} \dots\dots\dots (17)$$

Setelah nilai r, g, dan b dinormalisasi, rumus transformasi RGB ke HSV menjadi sebagai berikut

$$v = \max(r,g,b) \dots\dots\dots (18)$$

$$s = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(R,G,B)}{v}, & V > 0 \end{cases} \dots\dots\dots (19)$$

2.3 Motor Servo

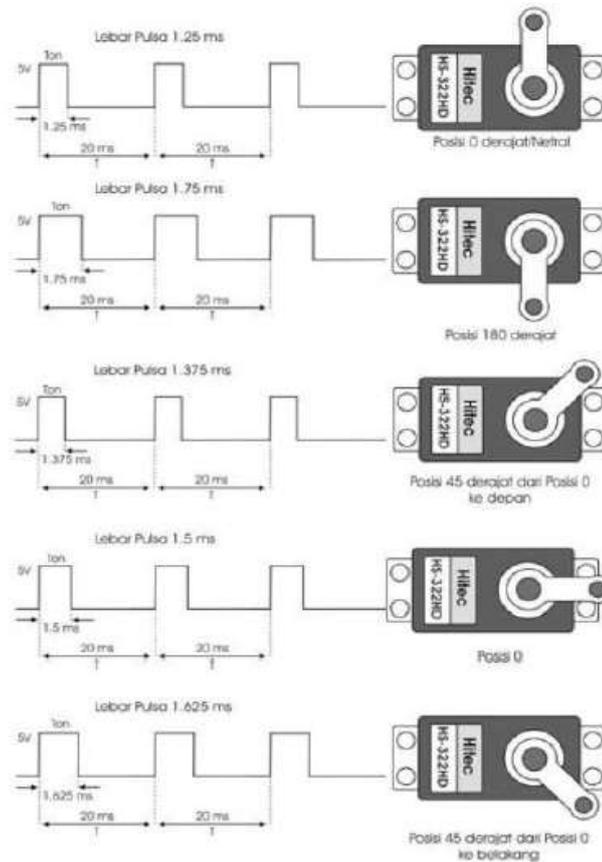
Motor servo adalah sebuah motor dengan *system* umpan balik tertutup, posisi dari motor akan diinformasikan kembali kerangkaian kontrol yang ada di dalam motor servo.

Motor ini terdiri dari sebuah motor DC, serangkaian roda gigi (*gear*), potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan putaran sudut dari sumbu motor servo diatur (dengan sinyal PWM) berdasarkan lebar pulsa (berkisar antara 0.5ms s.d. 2ms) yang dikirim melalui kaki sinyal dari motor servo. Secara umum terdapat 2 jenis motor servo, yaitu motor servo *standard* (dapat berputar 180 derajat) dan servo *icontinuous* (dapat berputar sebesar 360 derajat).[11]



Gambar 2.20 Motor Servo

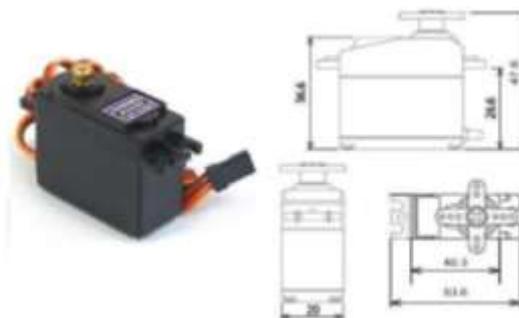
Motor servo pada dasarnya mempunyai sistem *close loop* sehingga dapat mempertahankan *horn* pada posisinya. Motor servo terdiri dari sebuah motor DC kecil, sistem kombinasi *gear* yang berfungsi mengatur kecepatan motor DC, sebuah potensiometer, dan sebuah *controller*. [10]



Gambar 2.21 Pulsa Kendali Motor *servo*

Servo TowerPro MG996R

TowerPro MG996R merupakan *servo digital* dengan *gear* metal memiliki ukuran medium dan ringan dengan daya *output* tinggi . *servo* dapat memutar sekitar 120 derajat (60o ke arah kanan dan 60o ke arah kiri).

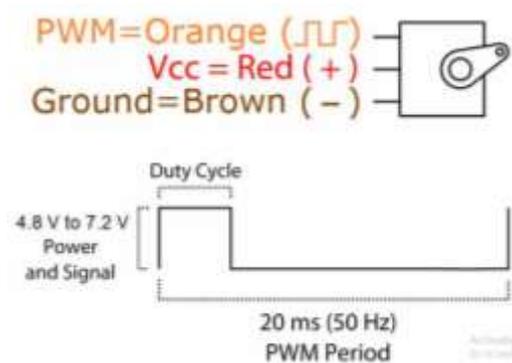


Gambar 2.22 *MG996R Tower-Pro* [8]

(Sumber : [http:// electronicscaldas/datasheet/ MG996R_Tower-Pro.pdf](http://electronicscaldas/datasheet/MG996R_Tower-Pro.pdf))

Spesifikasi motor

- Berat : 55 g
- Dimensi : 40.7 x 19.7 x 42.9 mm approx.
- Torsi Maksimal : 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Kecepatan Operasi : 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
- Tegangan Kerja : 4.8 V a 7.2 V
- Kebutuhan Arus : 500 mA



Gambar 2.23 Wiring MG996R Tower-Pro [8].

(Sumber : [http:// electronicscaldas/datasheet/ MG996R_Tower-Pro.pdf](http://electronicscaldas/datasheet/MG996R_Tower-Pro.pdf))

2.4 Driver Motor tipe PCA9685

PCA9685 adalah Driver Servo PWM 16-Channel dan 12-Bit yang dapat banyak membantu ketika tidak cukup pin output PWM untuk MCU. Satu-satunya syarat adalah bahwa *chip control* utama mendukung komunikasi I2C dan *integrated clock*, yang memungkinkan komunikasi antara *chip* dan *Driver Servo* PCA9685.

Driver Motor PCA9685 berperan untuk mengendalikan 4 buah motor servo yang digunakan pada Alat pemilah buah tomat, sehingga dapat mengontrol beberapa servo secara bersamaan.



Gambar 2.24 *Driver Motor PCA9685*

Sebagaimana perangkat i2c lainnya, modul *driver motor* ini memiliki alamat perangkat, untuk bisa diakses menggunakan protokol i2c oleh arduino. Alamat *default* nya adalah 0x40, namun untuk keperluan tingkat lanjut seperti saat dibutuhkan lebih dari 1 modul, maka anda perlu mengeset alamat modul anda (agar tidak terjadi konflik dengan *device* pertama) dengan men-*jumper* pin alamat yang tersedia pada modul, seperti tampak pada gambar.

Berikut pengaturan pengalamatannya;

- *Board 0: Address = 0x40 Offset = binary 00000 (default tidak diperlukan jumper)*
- *Board 1: Address = 0x41 Offset = binary 00001 (jumper A0, seperti gambar)*
- *Board 2: Address = 0x42 Offset = binary 00010 (jumper A1)*
- *Board 3: Address = 0x43 Offset = binary 00011 (jumper A0 & A1)*
- *Board 4: Address = 0x44 Offset = binary 00100 (jumper A2)*

Pengkabelan modul PCA9685 dengan Arduino sangat sederhana, berikut konfigurasinya

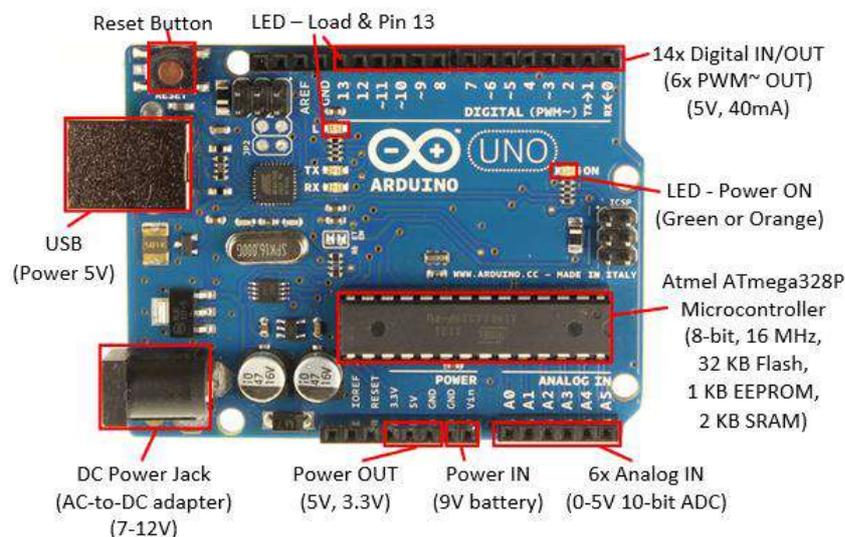
Arduino	PCA9685
• +5v	VCC
• GND	GND
• SDA	SDA
• SCL	SCL

2.5 Arduino

Arduino Uno R3 adalah keluaran pabrik "Arduino Uno", sudah menggunakan mikrokontroler ATmega328. Arduino Uno R3 mempunyai 14 *digital pin input/output* (6 dapat digunakan sebagai *output* PWM), 6 *analog input*, *resonator* keramik 16 MHz, koneksi USB, *jack* listrik, *header* ICSP, dan tombol *reset* di papan. Uno dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan *power* USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai. Arduino Uno berbeda dari semua papan sebelumnya dalam hal tidak menggunakan *FTDI chip driver USB-to-serial*.

Sebaliknya, fitur Atmega16U2 (Atmega8U2 sampai versi R2) diprogram sebagai *converter* USB-to-serial. Revisi 2 dari Uno memiliki resistor *pulling* 8U2 HWB yang terhubung ke tanah, sehingga lebih mudah untuk menggunakan mode DFU.

Board Arduino dapat dilihat pada **Gambar 2.25** di bawah ini. [10]

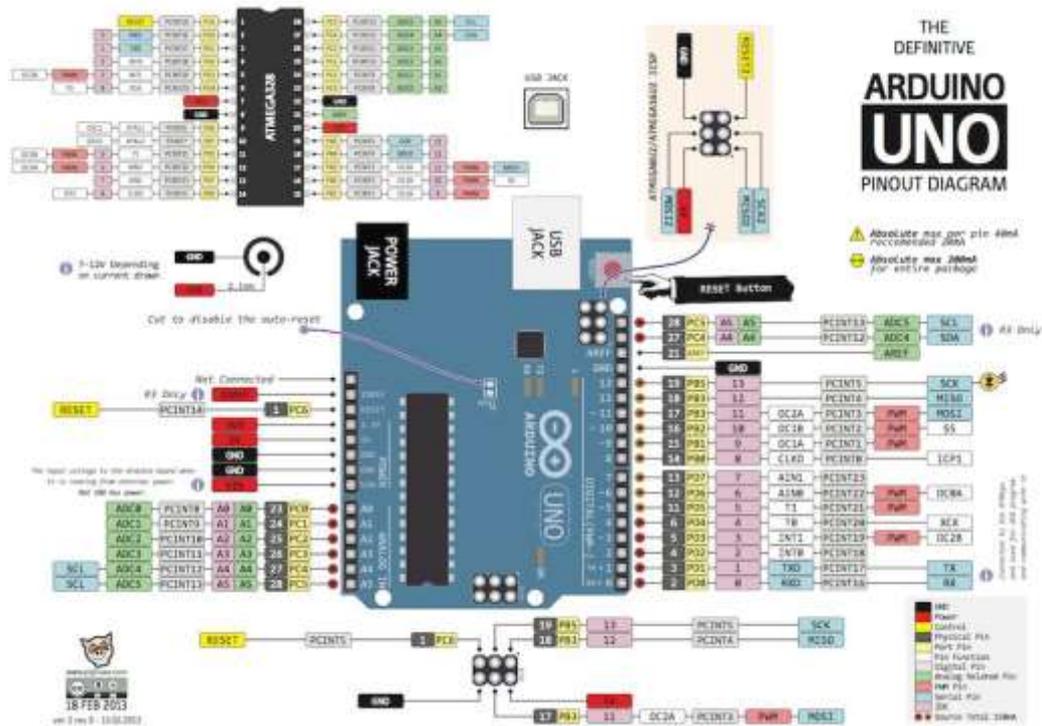


Gambar 2.25 Arduino Uno R3

(Sumber : <https://www.arduino.cc/en/Main/arduinoBoardUno>)

2.5.1 Arsitektur Arduino Uno

Konfigurasi pin-pin *board* arduino uno dapat dilihat pada **Gambar 2.26** dibawah ini.



Gambar 2.26 Konfigurasi Pin – Pin pada Arduino UNO R3

(Sumber : <https://www.arduino.cc/en/Main/arduinoBoardUno>)

Tabel 2.3 Spesifikasi Arduino Uno R3

Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Panjang	68.6mm
Lebar	53.4mm
Berat	25gram
Microcontroller	ATmega328
Tegangan Jalan	5V
Tegangan Masukan	7-12V
Batas Tegangan	6-20V
Digital I/O Pins	14
Analog Input Pins	6
DC Current per I/O Pins	40 m/A
DC Current for 3.3V Pins	50 m/A

2.5.2 Power Arduino UNO R3

Arduino Uno R3 dapat dioperasikan menggunakan 3 cara. pertama dengan menggunakan koneksi USB dengan komputer, Adaptor AC-DC , dan menggunakan batu baterai. Pin listrik pada *board* Arduino Uno R3 :

- Pin VIN atau *Voltage In* yaitu pin untuk tegangan masukan papan.
- Pin 5V atau pin 5 Volts adalah pin keluaran dari papan arduino yang nantinya digunakan untuk menyuplai perangkat serial yang bekerja dengan arduino.
- Pin 3.3 V adalah pin yang memasok 3,3 volt yang dihasilkan oleh *regulator on-board*. Menarik arus maksimum adalah 50 mA.
- Pin GND yaitu Pin *Ground*.
- Tegangan yang disarankan Untuk pengoprasian Arduino Uno R3 adalah 7 -12 V. tegangan dibawah atau diatasnya akan merusak papan Arduino. [10]

2.6 *Power Supply Switching*

Power Supply Switching adalah sebuah sistem *power supply* atau catu daya yang menggunakan teknologi *switching*. *Power supply* jenis ini menggunakan sebuah perangkat *switching* (sakelar) elektronik, dan biasanya *power supply switching* terdapat pada rangkaian sumber daya utama sebuah peralatan elektronik. Nama lain dari *power supply switching* adalah SMPS (*Switched Mode Power Supply*).



Gambar 2.27 *Power Supply Switching*

(Sumber : <https://www.belajaronline.net/2020/07/pengertian-power-supply-switching-dan-cara-kerjanya.htm>)

Power supply switching atau SMPS biasanya menggunakan transistor seri *on* atau *off* dan mempunyai frekuensi yang konstan untuk men *switching* transistor seri tersebut untuk menghasilkan tegangan regulasi, Tampilan *Power supply switching* terlihat pada **Gambar 2.27**, Adapun besarnya frekuensi *switching* adalah diatas 20 Kilo Hertz sehingga tidak dapat didengar oleh manusia.

Power supply dengan teknologi *switching* ini memiliki beberapa keuntungan dibandingkan dengan *power supply* konvensional yang menggunakan trafo besi berukuran besar. Selain memiliki ukuran yang jauh lebih kecil, *power supply switching* juga memiliki efisiensi daya listrik hingga 83% jika dibandingkan dengan *power supply* konvensional. *Power supply* konvensional memiliki efisiensi yang rendah karena tegangan input yang jauh lebih tinggi dari tegangan *output* akan berubah menjadi panas sehingga sebagian besar daya listrik *input* akan hilang karena berubah menjadi panas tersebut.

2.6.1 Komponen Utama *Power Supply Switching*

Power Supply Switching atau SMPS terdiri dari beberapa komponen utama diantaranya :

- 1) Pada blok *Unregulated* menggunakan 4 buah Dioda (*Half Wave*) tipe 1N5406 dan menggunakan elco dengan ukuran 400 volt 220 MF.
- 2) Pada blok *switching* menggunakan Mosfet dengan tipe K 2141.
- 3) Pada blok trafo *Inverter* menggunakan trafo tipe 1621-0074-00. TVE 9614
- 4) Pada blok *Regulated* menggunakan bermacam-macam ukuran dioda dan elco sesuai dengan tegangan yang dihasilkan dari trafo *inverter*.
- 5) Pada blok *Komparator* menggunakan IC tipe UC 3842.

2.6.2 Cara Kerja *Power Supply Switching*

Power Supply Switching atau SMPS secara garis besar meliputi kerja :

- 1) Penyearahan, merubah tegangan masukan AC menjadi tegangan keluaran DC.
- 2) *Converter*, merubah tegangan DC menjadi tegangan keluaran DC sesuai kebutuhan.
- 3) *Filtering*, menghilangkan denyut atau *ripple* pada tegangan keluaran.
- 4) Regulasi, membuat tegangan keluaran agar tetap stabil terhadap perubahan tegangan masukan dan perubahan beban.
- 5) Isolasi, membatasi bagian primer dan bagian sekunder dengan tujuan agar chasis jika dipegang tidak menimbulkan bahaya akibat sengatan listrik.
- 6) Proteksi, mampu melindungi peralatan elektronik dari tegangan keluaran yang *over* serta melindungi *power supply* dari kerusakan jika terdapat suatu kesalahan.

Sumber tegangan dari PLN akan mengalirkan arus masuk ke blok *Unregulated* (Dioda Kiprox dan elco), di blok ini tegangan AC diubah menjadi tegangan DC yang kemudian tegangan masuk ke blok *Switching* (yang dimaksud *switching* adalah dapat berupa IC, transistor, Mosfet). Setelah itu tegangan masuk ke blok *Inverter* (Trafo inti Ferrit).

Di blok ini akan kembali terjadi perubahan tegangan yang tadinya sudah tegangan DC dirubah lagi menjadi tegangan AC, tetapi tegangan AC yang dihasilkan pada blok ini memiliki sifat yang berbeda dari tegangan AC sumber.

Tegangan AC yang dihasilkan dari blok *Inverter* kemudian masuk ke blok *Regulated* (Dioda *Half Wave*, Penyearah dan elco), di blok ini akan kembali terjadi perubahan tegangan yang tadinya tegangan AC akan berubah menjadi tegangan DC. Tegangan ini kemudian baru masuk ke beban dan ada juga yang masuk ke blok Komparator (blok ini berupa IC dimana IC ini masih terletak satu blok dengan blok *power supply switching*).