

BAB II

TINJAUAN PUSTAKA

2.1 Pengolahan Citra Digital

Pengolahan citra (*image processing*) adalah pemrosesan citra yang memiliki input citra dan output citra hasil pemrosesan yang khususnya dilakukan menggunakan komputer untuk menjadikan kualitas citra yang lebih baik. 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.

(Putra, Darma.2010.Pengolahan Citra Digital.Yogyakarta: Andi Offset.)

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. Format citra digital yang banyak dipakai adalah Citra Biner (monokrom), Citra Skala Keabuan (*grayscale*), Citra Warna (*true color*), dan Citra Warna Berindeks.

Pengolahan citra memiliki beberapa fungsi, diantaranya adalah:

1. Digunakan sebagai proses untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau komputer.
2. Digunakan untuk teknik pengolahan citra dengan mentransformasikan citra menjadi citra lain. Contoh : pemampatan citra (*image compression*) Sebagai proses awal (*preprocessing*) dari komputer.

Pengolahan citra dapat dibagi kedalam tiga kategori yaitu :

1. Kategori rendah melibatkan operasi-operasi sederhana seperti pra-pengolahan citra untuk mengurangi derau, pengaturan kontras, dan pengaturan ketajaman citra. Pengolahan kategori rendah ini memiliki input dan output berupa citra.
2. Pengolahan kategori menengah melibatkan operasi-operasi seperti segmentasi dan klasifikasi citra. Proses pengolahan citra menengah ini

melibatkan input berupa citra dan output berupa atribut (fitur) citra yang dipisahkan dari citra input. Pengolahan citra kategori melibatkan proses pengenalan dan deskripsi citra.

3. Pengolahan kategori tinggi ini termasuk menjadikan objek-objek yang sudah dikenali menjadi lebih berguna, berkaitan dengan aplikasi, serta melakukan fungsi-fungsi kognitif yang diasosiasikan dengan vision.

2.2 Metode *Edge Detection*

Edge Detection (Deteksi tepi) adalah metode yang dapat mendeteksi garis tepi, yaitu garis yang memisahkan antara objek dengan latar belakang (*background*). *Edge Detection* merupakan pengolahan citra tingkat dasar yang diperlukan untuk melakukan pengolahan citra pada tingkat yang lebih tinggi. *Edge Detection* banyak digunakan dalam analisa pengolahan citra untuk berbagai macam tujuan (Bin & Yeganeh, 2012).

Tujuan pendeteksian ini adalah bagaimana agar objek di dalam gambar dapat dikenali dan disederhanakan bentuknya dari bentuk sebelumnya. Ada banyak algoritma yang digunakan untuk mendeteksi tepi, salah satunya *canny edge detection*.

2.2.1 Canny *Edge Detection*

Canny edge detection merupakan pengembangan dari metode dasar *edge detection*. *Canny edge detector* dikembangkan oleh John F. Canny pada tahun 1986 dan menggunakan proses multi-tahap untuk mendeteksi berbagai tepi dalam gambar. (Danil, C., 2011) Algoritma *canny* memiliki langkah-langkah berbeda untuk mencapai tujuan, yaitu:

A. *Smoothing*

Smoothing membuat gambar menjadi kabur untuk menghilangkan *noise*. Tidak dapat dipungkiri bahwa semua gambar yang diambil dari kamera akan berisi sejumlah *noise*. Untuk mencegah *noise* salah dideteksi sebagai tepian, maka *noise* harus dikurangi. Oleh karena itu pada langkah pertama gambar harus diperhalus

dengan menggunakan *Gaussian filter*. Inti dari *Gaussian filter* adalah standar deviasi dengan $\sigma = 1.4$ ditunjukkan pada persamaan 2.1 di bawah ini.

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \dots\dots\dots(2.1)$$

B. *Finding Gradients*

Pada dasarnya algoritma canny adalah untuk menemukan titik tepi pada gambar *grayscale* dengan perubahan nilai intensitas yang paling besar, area inilah yang ditemukan dengan menentukan gradien gambar. Gradien pada setuao piksel gambar yang telah *dismoohiting* ditentukan dengan menerapkan operator Sobel. Langkah kedua adalah memperikarakan gradien pada arah x dan y. Hal tersebut terdapat pada persamaan 2.2 berikut.

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \dots\dots\dots(2.2)$$

Magnitudo gradien (juga dikenal sebagai kekuatan tepi) dapat ditentukan sebagai jarak Euclidean yang diukur mengukur dengan menerapkan hukum Pythagoras seperti yang ditunjukkan dalam Persamaan 2.3 yang terkadang disederhanakan dengan menerapkan ukuran jarak Manhattan seperti yang ditunjukkan dalam persamaan berikut untuk mengurangi kompleksitas komputasi.

$$|G| = \sqrt{G_x^2 + G_y^2}$$

$$|G| = |G_x| + |G_y| \dots\dots\dots(2.3)$$

Dimana: G_x dan G_y adalah gradien pada masing-masing arah x dan y. gambar dengan gradien yang besar sering menunjukkan tepian yang cukup jelas. Namun, tepian biasanya luas dan dengan demikian tidak dapat menunjukkan persis di mana tepian yang sebenarnya. Untuk menentukan tepian yang sebenarnya ini,

arah tepian harus ditentukan dan disimpan seperti ditunjukkan dalam Persamaan 2.4 berikut.

$$\theta = \arctan \left(\frac{|Gy|}{|Gx|} \right) \dots\dots\dots(2.4)$$

C. Non – Maximum Suppression

Setelah melakukan “*blur*” pada bagian *smoothing*, langkah ini bertujuan untuk menajamkan gambar yang *blur*. Pada dasarnya hal ini dilakukan dengan mempertahankan semua maxima local dalam gambar gradien dan menghapus segala sesuatu yang lain-lain.

D. Double Thresholding

Setelah *non-maximum suppression* dilakukan, Piksel tepian yang tersisa, ditandai dengan nilai piksel per piksel yang kuat. Kebanyakan dari titik ini adalah tepian yang nyata pada gambar, tetapi beberapa kemungkinan disebabkan oleh *noise* atau variasi warna karena permukaan yang kasar. Cara sederhana untuk membedakannya adalah menggunakan nilai *threshold* (ambang batas) sehingga hanya tepian dengan nilai yang kuat yang akan dipertahankan. Canny menggunakan sistem *double thresholding*. Nilai piksel tepi yang lebih nesar dari nilai *threshold* tertinggi ditandai sebagai titik yang “kuat”, tepian dengan nilai yang lebih kecil dari *threshold* bawah akan dihapus, dan tepian dengan nilai piksel antara *threshold* atas dengan *threshold* bawah akan ditandai sebagai tepian yang lemah.

E. Edge Tracking by Hysteresis

Tepian yang kuat diinterpretasikan sebagai "tepiian pasti" dan dapat segera dimasukkan sebagai tepian pada hasil akhir citra. Tepi yang lemah akan termasuk jika hanya tepi tersebut terhubung dengan tepi yang kuat, dengan logika bahwa *noise* dan variasi warna tidak mungkin untuk menghasilkan tepi yang kuat (dengan penyesuaian yang tepat dari *thresholding*). Dengan demikian tepian yang kuatlah yang akan menghasilkan tepian yang asli pada gambar. Tepian yang lemah dapat terjadi karena memang merupakan tepian yang nyata atau *noise* / variasi warna.

2.3 Kamera FPV

Kamera FPV (*First-Person View*) adalah kamera yang biasa dipakai sebagai CCTV dan sebagai *vision* dari robot UAV (*Unmanned Aerial Vehicle*) dan ROV (*Remotely Operated Vehicle*) yang berguna untuk melihat apa yang terjadi di lapangan. Kamera yang dipakai pada robot ini adalah kamera FPV Sony 700TVL dapat dilihat pada gambar 2.1.



Gambar 2.1 Kamera Sony 700TVL

(Sumber: <http://www.vacron.com/download/manual/camera/SONYMANUAL-V1.0.pdf>)

Kamera Sony 700TVL memiliki ukuran 3.8 x 3.8 x 0.2 cm dan jarak lensa yang bersikar pada 3.6mm / 6mm / 8mm (3.6mm *default*). Kamera ini memiliki resolusi 1020 x 508 pixel dimana gambar yang dihasilkan pada kualitas yang baik. Tegangan yang dibutuhkan kamera ini adalah 12VDC.

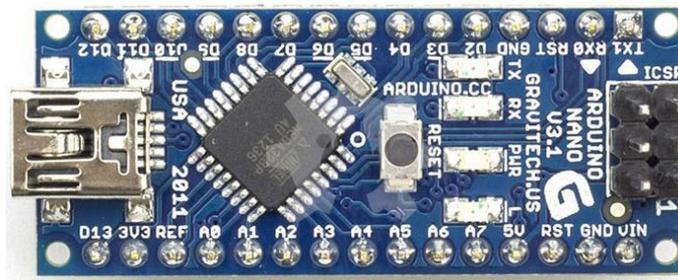
(<http://www.vacron.com/download/manual/camera/SONYMANUAL-V1.0.pdf>.)

2.4 Mikrokontroler

Mikrokontroler adalah sebuah otak dari suatu sistem elektronika seperti halnya mikroprosesor sebagai otak komputer. Nilai plus bagi mikrokontroler adalah terdapatnya memori dan Port Input/Output dalam suatu IC. Kemampuannya yang programmable, fitur yang lengkap (ADC internal, EEPROM internal, Port I/O, Komunikasi Serial, dll), dan juga harga yang terjangkau memungkinkan mikrokontroler digunakan pada berbagai sistem elektronis, seperti pada robot, automasi industri, sistem alarm, peralatan telekomunikasi, hingga peralatan rumah tangga.. Salah satu contoh dari mikrokontroler adalah Arduino Nano (Wardhana, Lingga.2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 : Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Andi.)

2.4.1 Arduino Nano

Mikrokontroler Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan Integrated Development Environment (IDE) yang canggih. IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory mikrokontroler. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B (<http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>). Arduino Nano dapat dilihat pada gambar 2.2 berikut.



Gambar 2.2 Arduino Nano

(sumber: <http://www.gravitech.us/arna30wiatp.html>)

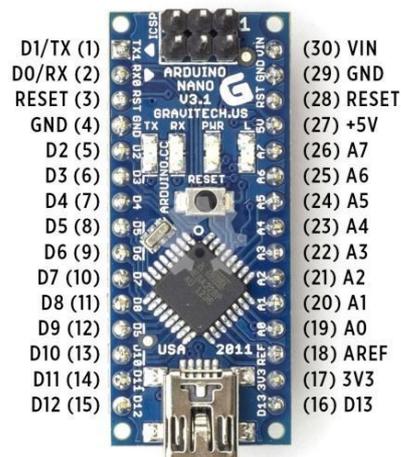
2.4.2 Konfigurasi Pin Arduino Nano

Konfigurasi pin Arduino Nano. Arduino Nano memiliki 30 Pin. Berikut Konfigurasi pin Arduino Nano.

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya digital.
 2. GND merupakan pin ground untuk catu daya digital.
 3. AREF merupakan Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
 4. RESET merupakan Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino
-

5. Serial RX (0) merupakan pin yang berfungsi sebagai penerima TTL data serial.
6. Serial TX (1) merupakan pin yang berfungsi sebagai pengirim TT data serial.
7. External Interrupt (Interupsi Eksternal) merupakan pin yang dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
8. Output PWM 8-Bit merupakan pin yang berfungsi untuk analogWrite().
9. SPI merupakan pin yang berfungsi sebagai pendukung komunikasi.
10. LED merupakan pin yang berfungsi sebagai pin yang diset bernilai HIGH, maka LED akan menyala, ketika pin diset bernilai LOW maka LED padam. LED Tersedia secara built-in pada papan Arduino Nano.
11. Input Analog (A0-A7) merupakan pin yang berfungsi sebagai pin yang dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi analogReference().

Konfigurasi pin Arduino Nano dapat dilihat pada gambar 2.3.



Gambar 2.3 Konfigurasi Pin Arduino Nano

(sumber: <http://www.gravitech.us/arna30wiatp.html>)

2.4.3 Sumber Daya Arduino Nano

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan

tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano diberikan daya dari luar (Non-USB) maka Chip FTDI tidak aktif dan pin 3.3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkan LED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi HIGH (<http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>).

2.4.4 Komunikasi Arduino Nano

Arduino Nano memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. ATmega168 dan ATmega328 menyediakan komunikasi serial UART TTL (5 Volt), yang tersedia pada pin digital 0 (RX) dan pin 1 (TX). Sebuah chip FTDI FT232RL yang terdapat pada papan Arduino Nano digunakan sebagai media komunikasi serial melalui USB dan driver FTDI (tersedia pada software Arduino IDE) yang akan menyediakan COM Port Virtual (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip FTDI dan koneksi USB yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1) (<http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>).

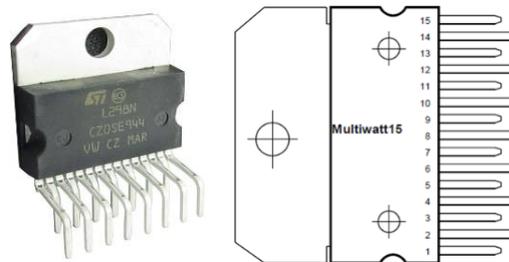
2.5 Driver Motor

Driver motor adalah rangkaian yang digunakan untuk mengatur arah putaran dari motor DC. Rangkaian terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang memudahkan dalam menentukan arah putaran suatu motor DC. Tetapi dipasaran telah disediakan IC L298N sebagai driver motor DC yang dapat mengatur arah putar dan disediakan pin untuk input yang berasal dari PWM untuk mengatur kecepatan motor DC. Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada motor DC sedangkan arus keluaran pada mikro sangat kecil. *Driver* motor

merupakan pilihan alternatif yang harus digunakan untuk mengendalikan motor DC pada robot beroda.

(<http://kedairobot.com/components/35-l298-motor-driver.html>).

L298N adalah *driver* motor yang memiliki dua buah rangkaian *H-Bridge* didalamnya, sehingga dapat digunakan untuk *men-drive* dua buah motor DC. *H-Bridge driver* motor DC L298N masing-masing dapat menghantarkan arus hingga 2A. IC L298N memiliki 15 kaki yang memiliki fungsi tersendiri. Konfigurasi kaki-kaki IC L298N dapat kita lihat pada gambar 2.4 berikut ini, sedangkan keterangan fungsi untuk setiap kakinya dapat dilihat pada tabel 2.3.



Gambar 2.4 IC Driver motor L298N

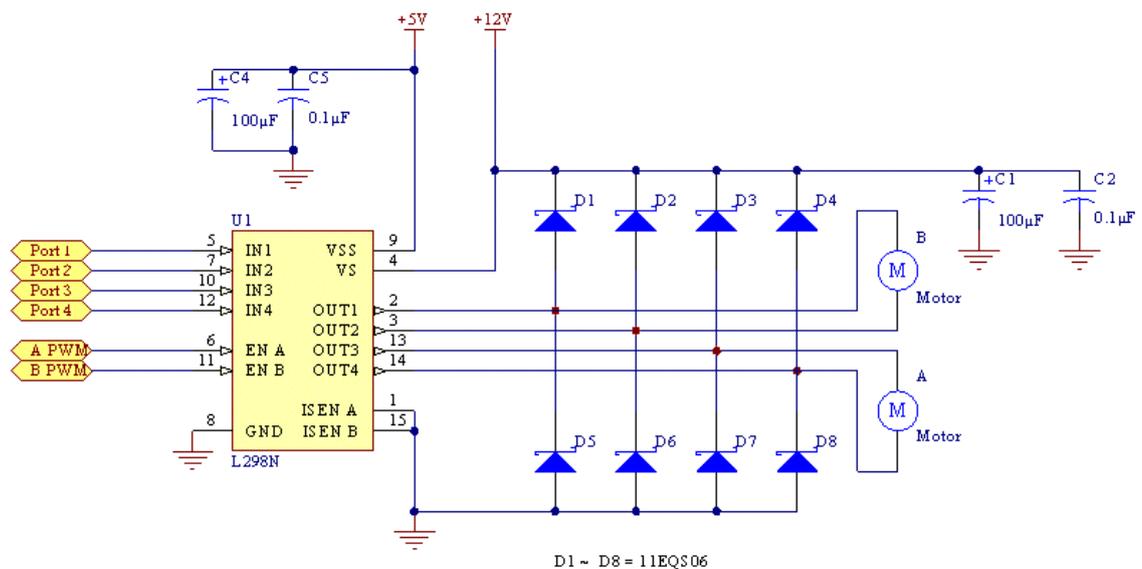
(<http://www.robotics-university.com/2015/01/driver-motor-dcmp-menggunakan-ic-l298.html>, 2017)

Tabel 2.1 Keterangan fungsi kaki/pin IC L298

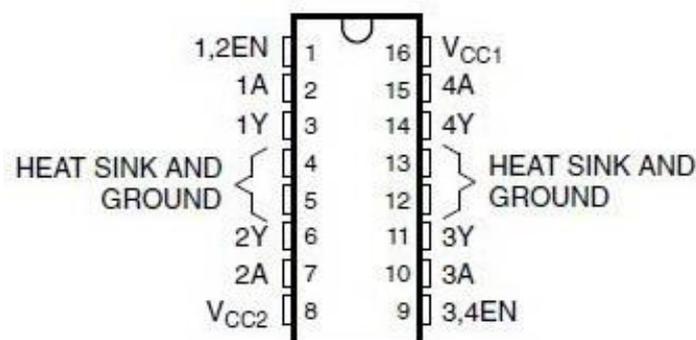
Pin	Nama pin	Pin	Nama pin
1	Current Sensing A	9	Vss (tegangan supply IC)
2	Output 1	10	Input 3
3	Output 2	11	Enable B
4	Vs (tegangan supply motor)	12	Input 4
5	Input 1	13	Output 3
6	Enable A	14	Output 4
7	Input 2	15	Current Sensing B
8	Ground		

2.5.1 Prinsip Kerja *Driver* Motor DC dengan IC L298N

Untuk mengendalikan putaran motor DC menggunakan *driver* motor L298N perlu melibatkan 3 buah pin/kaki IC L298, yaitu: Kaki In1 (Input 1) dan kaki In2 (Input 2) untuk menentukan arah putaran motor DC yang dikendalikan, apakah berputar searah jarum jam atau berputar berlawanan jarum jam. Kaki EnA (Enable A) yang berfungsi untuk mengaktifkan atau menonaktifkan *driver* motor L298N. Aktif ketika kaki EnA diberi logika high dan nonaktif ketika kaki EnA diberi logika low) atau memasukkan nilai PWM pada kaki EnA untuk mengatur kecepatan motor.



Gambar 2.5 Skematik IC L298 sebagai driver motor DC
(<http://fritzing.org/projects/working-with-l298n-dc-motor-driver>, 2017)



Gambar 2.6 Pin IC *driver* motor L293D
(Sumber: <https://electrosome.com/l293d-quadruple-half-h-dc-motor-driver/>)

2.5.2 Fungsi Pin Driver Motor DC IC L298D

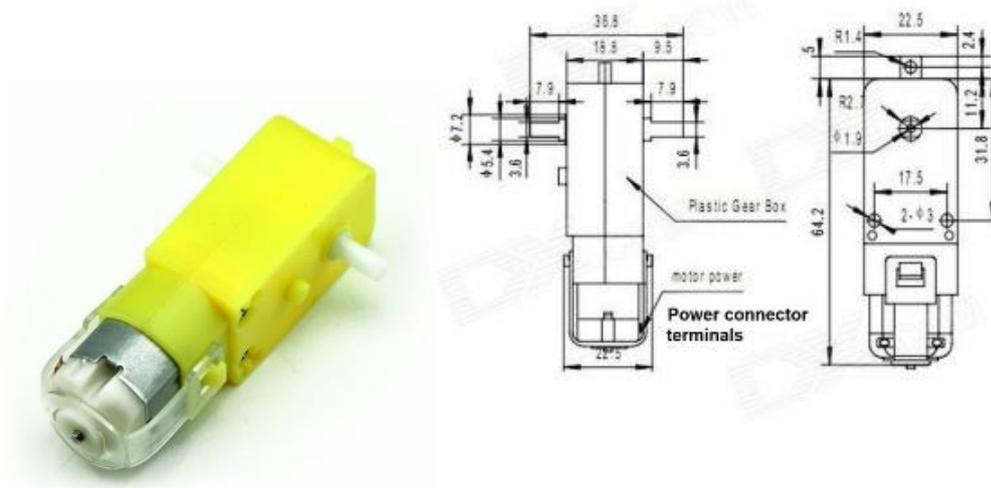
- Pin EN (Enable, EN1.2, EN3.4) berfungsi untuk mengizinkan driver menerima perintah untuk menggerakkan motor DC.
- Pin In (Input, 1A, 2A, 3A, 4A) adalah pin input sinyal kendali motor DC
- Pin Out (Output, 1Y, 2Y, 3Y, 4Y) adalah jalur output masing-masing driver yang dihubungkan ke motor DC
- Pin VCC (VCC1, VCC2) adalah jalur input tegangan sumber driver motor DC, dimana VCC1 adalah jalur input sumber tegangan rangkaian kontrol driver dan VCC2 adalah jalur input sumber tegangan untuk motor DC yang dikendalikan.
- Pin GND (Ground) adalah jalur yang harus dihubungkan ke ground, pin GND ini ada 4 buah yang berdekatan dan dapat dihubungkan ke sebuah pendingin kecil.

Tabel 2.2 Kondisi Gerak Motor dengan Menggunakan Driver Motor L298

Input M1		Input M2		Enable		Motor 1	Motor 2	Ket
In1	In2	In3	In4	EnA	EnB			
1	0	1	0	1	1	Maju	Maju	Maju
0	1	0	1	1	1	Mundur	Mundur	Mundur
1	0	0	0	1	1	Maju	Diam	Belok kiri depan
0	0	1	0	1	1	Diam	Maju	Belok kanan depan
0	1	0	0	1	1	Mundur	Diam	Belok kiri belakang
0	0	0	1	1	1	Diam	Mundur	Belok kanan belakang
1	0	0	1	1	1	Maju	Mundur	Berputar ke kiri
0	1	1	0	1	1	Mundur	Maju	Berputar ke kanan
1	0	1	0	0	0	Diam	Diam	Diam

2.6 Motor DC

Motor DC adalah suatu piranti elektronik yang berfungsi untuk mengubah energi listrik menjadi energi mekanik. Pada aplikasi robotika pergerakan atau actuator robot beroda umumnya menggunakan motor DC yang menggunakan gear box dan roda sebagai komponen pendukungnya, karena jenis motor ini lebih mudah untuk dikendalikan. Kecepatan yang dihasilkan oleh motor DC berbanding lurus dengan potensial yang diberikan.



Gambar 2.7 Motor DC

(<http://www.jogjarobotika.com/motor-dc/76-motor-dc-gearbox-gearbox-6v-roda.html>, 2017)

Kecepatan motor DC dapat diatur dengan beberapa cara, yaitu dengan mengatur fluks medan, dengan mengatur tahanan jangkar, dan dengan mengatur tegangan sumber. Cara yang ketiga ini merupakan pengaturan yang sering digunakan karena penggunaannya yang relatif mudah. Pengaturan tegangan sumber biasanya menggunakan metode PWM (*Pulse Width Modulation*).

2.6.1 Pengaturan Kecepatan Motor DC

Kontrol kecepatan berarti perubahan pada kecepatan motor DC akibat adanya perubahan nilai tegangan. Salah satu fitur penting motor DC adalah kecepatannya dapat dikontrol dengan relatif mudah. Kecepatan motor akan berbanding lurus dengan tegangan yang masuk, jika tegangan naik maka rpm motor DC juga akan naik.

(<https://www.electrical4u.com/speed-control-of-dc-motor/>)

$$N = \frac{V - I_a R_a}{k \phi} \dots\dots\dots(2.5)$$

Keterangan :

- N = Kecepatan rotasi (rpm)
- V = Tegangan (volt)
- I = Arus Jangkar
- R = Resistansi Jangkar
- \emptyset = Flux Magnet per kutub

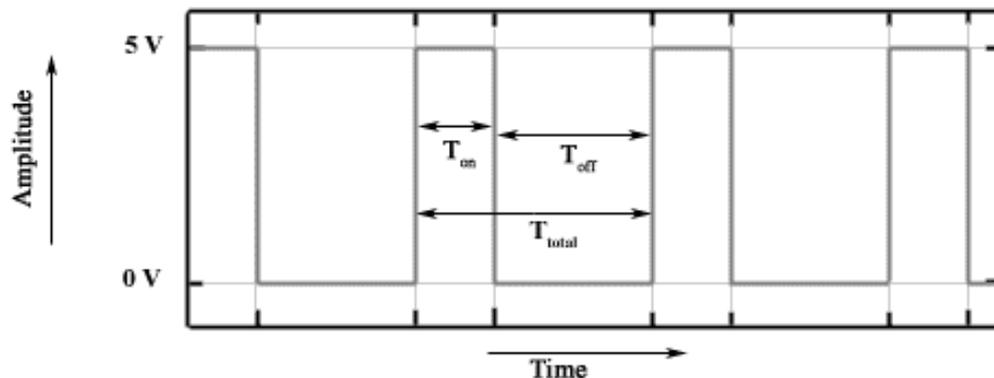
2.6.2 Pengaturan Motor DC Dengan Modulasi Lebar Pulsa (PWM)

Salah satu cara untuk mengatur kecepatan putar motor dc adalah dengan cara modulasi lebar pulsa (PWM). Modulasi lebar pulsa (PWM) yaitu sebuah cara memanipulasi lebar pulsa dalam satu. Sinyal PWM pada umumnya memiliki amplitude dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun duty cycle bervariasi (antara 0% hingga 100%).

(<https://www.codepolitan.com/tutorial/analog-output-arduino-menggunakan-pwm-pulse-width-modulation#sthash.LeBJt9Na.dpuf>)

Duty cycle adalah persentasi panjang pulsa *high* dalam satu periode sinyal. Ketika *duty cycle*-nya 0% atau sinyal *low* penuh, maka nilai tegangan yang dikeluarkan adalah 0V. Ketika *duty cycle*-nya 100% atau sinyal *high* penuh maka tegangan yang dikeluarkan adalah 5V. Contoh penggunaan PWM pada pengaturan kecepatan motor dc semakin besar nilai *duty cycle* yang diberikan maka akan berpengaruh terhadap cepatnya putaran motor. Apabila nilai *duty cycle*-nya kecil maka motor akan bergerak lambat.

PWM yang dapat dihasilkan oleh arduino memiliki alokasi data 8bit, atau memiliki variasi perubahan nilai parameter mulai dari 0 – 255, perubahan nilai yang mewakili *duty cycle* 0 – 100% dari keluaran PWM tersebut. Untuk mengatur nilai *duty cycle* pada arduino yaitu menggunakan fungsi `analogWrite([nomorPin], [nilai])`. Jika mengeset *duty cycle* ke 0%, maka set nilai parameter ke 0, dan untuk *duty cycle* 100%, maka set nilai parameter ke 255. Jadi *duty cycle* ke 50%, berarti nilai parameter yang harus diset adalah 127.



Gambar 2.8 Sinyal PWM dan rumus perhitungannya

(sumber: <http://www.arisulistiono.com/2010/02/pulse-width-modulation-pwm-pengenalan.>)

T_{on} adalah waktu dimana tegangan keluaran berada pada posisi tinggi (*high* atau 1) dan T_{off} adalah waktu dimana tegangan keluaran berada pada posisi rendah (*low* atau 0).

Anggap T_{total} adalah waktu satu siklus atau penjumlahan antara T_{on} dengan T_{off} , biasa dikenal dengan istilah “periode satu gelombang”.

$$T_{total} = T_{on} + T_{off} \quad \dots\dots\dots(2.6)$$

Sinyal PWM adalah periode sinyal dan lebar pulsanya. Lebar pulsa dinyatakan dengan *duty cycle* yang merupakan presentase dari perbandingan pulsa on (T_{on}) dengan T_{total} , sehingga :

$$D = \frac{T_{on}}{(T_{on} + T_{off})} \times 100\% \text{ atau } D = \frac{T_{on}}{T_{total}} \times 100\% \quad \dots\dots\dots(2.7)$$

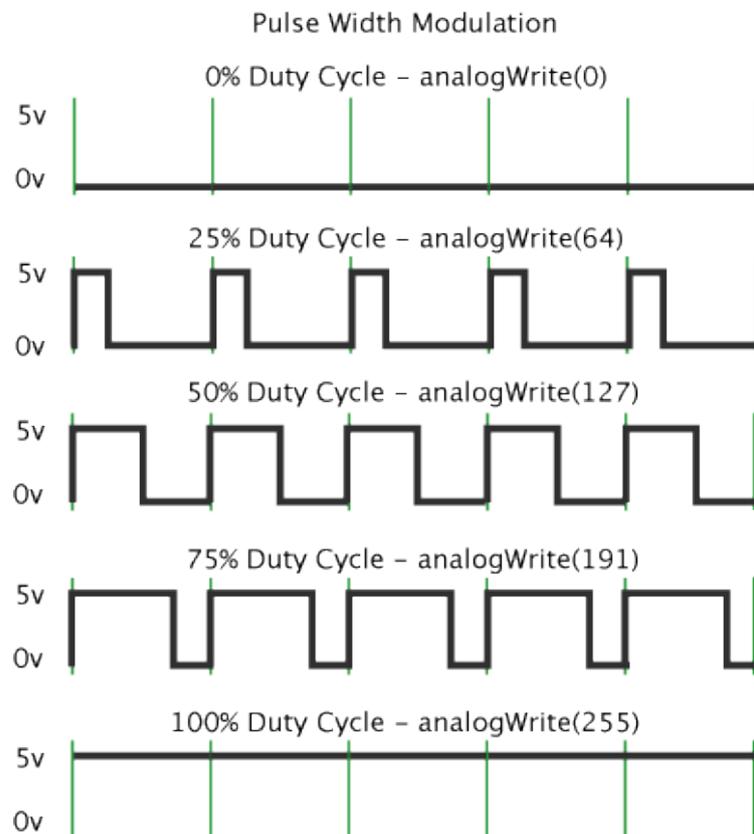
Hubungan antara *duty cycle* dengan PWM yaitu untuk setiap presentase nilai *duty cycle* berbanding lurus dengan nilai PWM. Untuk lebih jelasnya hubungan *duty cycle* dengan PWM dapat dilihat pada rumus berikut :

$$PWM = PWM_{max} \times Duty\ Cycle \quad \dots\dots\dots(2.8)$$

Tegangan keluaran dapat bervariasi sesuai dengan *duty cycle* dan dapat dirumuskan sebagai berikut,

$$V_{out} = D \times V_{in} \text{ atau } V_{out} = \frac{T_{on}}{T_{total}} \times V_{in} \quad \dots\dots\dots(2.9)$$

Dari rumus diatas bahwa tegangan keluaran dapat diubah-ubah secara langsung dengan mengubah nilai Ton. Apabila Ton adalah 0, Vout juga akan 0. Apabila Ton adalah Ttotal maka Vout adalah Vin atau katakanlah nilai maksimumnya.



Gambar 2.9 Grafik PWM

(<https://www.codepolitan.com/tutorial/analog-output-arduinomenggunakan-pwm-pulse-width-modulation#sthash.LeBJt9Na.dpuf>, 2017)

Tabel 2.3 Linearitas *Duty Cycle*

Nilai Analog Arduino	Nilai Duty Cycle	Nilai Tegangan
0	0%	0 volt
64	25%	1.25 volt
172	50%	2.5 volt
191	75%	3.7 volt
255	100%	5 volt

2.7 Motor Servo

Motor *servo* pada robot pendeteksi kecacatan pipa ini berfungsi sebagai aktuator yang menggerakkan arah penglihatan dari kamera. Motor *servo* dikemas dalam bentuk segi empat dengan sebuah *output shaft* motor dan konektor dengan 3 kabel yaitu *ground*, *power* dan *control*. Jenis motor *servo* berdasarkan sudut operasi motor *servo* dibagi menjadi 2 yaitu:

a. Motor *Servo Standart*

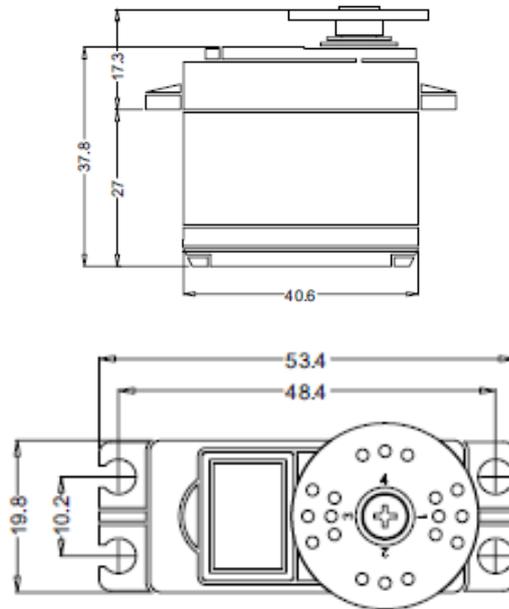
Motor *servo standart* merupakan motor *servo* yang mampu bergerak CW (berputar searah jarum jam) dan CCW (berputar berlawanan arah jam) dengan sudut operasi tertentu, misal 60°, 90° atau 180°. sudut maksimal yang diperbolehkan untuk motor *servo* standart adalah 180°. Motor *servo* ini sering dipakai pada sistem robotika yang menggunakan lengan atau kaki.

b. Motor *Servo Continous*

Motor *servo continous* adalah motor *servo* yang mampu bergerak CW dan CCW tanpa batasan sudut operasi (berputar secara kontinyu). Motor *servo* ini sering digunakan sebagai aktuator pada *mobile* robot. Motor *servo* beroperasi pada tegangan *supply* 4,8 volt hingga 7,2 volt. Gambar 2.0 dibawah merupakan gambar motor servo yang digunakan dan Gambar 2.11 merupakan skematik motor servo.

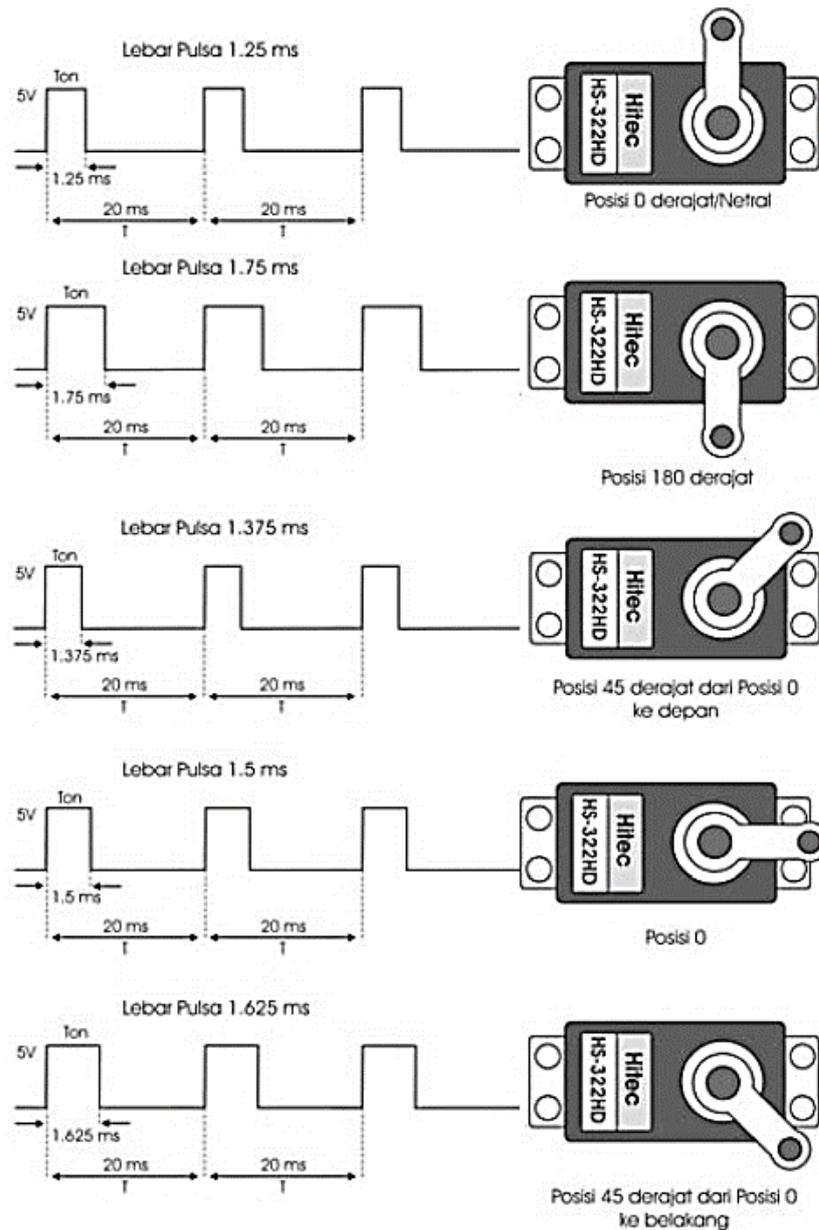


Gambar 2.10 Motor *Servo Standart*
(Sumber : www.sparkfun.com/products/11884)



Gambar 2.11 Skematik Motor Servo
(Sumber : Datasheet Motor Servo.pdf)

Motor servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan pulsa PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *Ton Duty Cycle* 1.5 ms, maka rotor dari motor akan berhenti tepat pada tengah-tengah (sudut 90^0). Semakin lebar pulsa *high (Ton)* maka akan semakin besar gerakan sumbu ke arah yang searah dengan jarum jam. Pada gambar 2.12 menunjukkan sudut yang dibentuk oleh motor servo berdasarkan lebar pulsa yang diberikan.

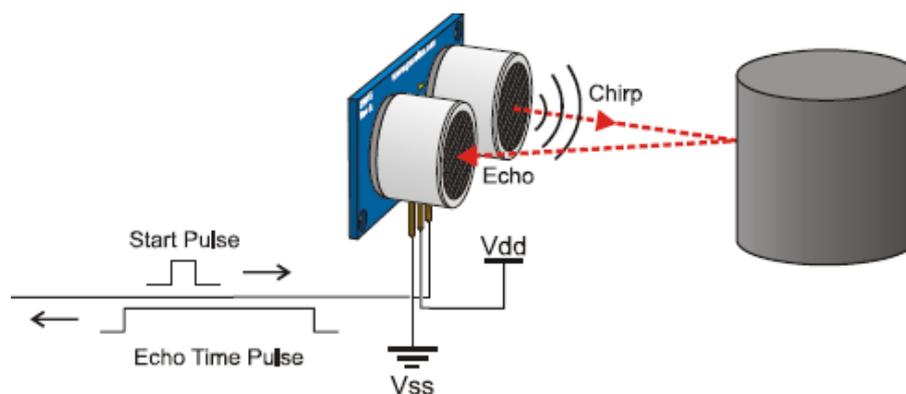


Gambar 2.12 Pengaruh Pemberian Pulsa terhadap posisi Motor Servo
(Sumber : <http://elektronika-dasar.web.id/motor-servo>)

2.8 Sensor Ultrasonik

Sensor Ultrasonik pada robot pendeteksi kecacatan pipa digunakan sebagai pendeteksi nilai jarak pada dinding pipa untuk menyesuaikan posisi robot. Sensor ultrasonik merupakan alat elektronika yang kemampuannya bisa mengubah energi listrik menjadi energi mekanik dalam bentuk gelombang suara dengan frekuensi di atas 20 kHz. Sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang

disebut transmitter dan rangkaian penerima ultrasonik yang disebut receiver. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari transmitter ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh receiver ultrasonik. Sinyal yang diterima oleh rangkaian receiver dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul). Prinsip kerja dari sensor ultrasonik dapat dilihat pada Gambar 2.13 dibawah ini.



Gambar 2.13 Prinsip Kerja Sensor Ultrasonik

(Sumber : Parallax-28015-datasheet.pdf)

Dari gambar 2.13 dapat dijelaskan mengenai prinsip kerja dari sensor ultrasonik adalah sebagai berikut:

- Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20 kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40 kHz. Sinyal tersebut dibangkitkan oleh rangkaian pemancar ultrasonik.
- Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal/gelombang bunyi dengan kecepatan bunyi yang berkisar 340m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima ultrasonik.
- Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya.