

BAB II

TINJAUAN PUSTAKA

2.1. Robot Berkaki

Robot berkaki merupakan sebuah robot yang dapat bergerak dengan leluasa karena memiliki kemampuan bergerak untuk berpindah posisi yang didukung oleh bentuk kaki yang dirancang sebagai alat penggerakannya. Penggunaan kaki dan bentuk tubuh, ini semua akan disesuaikan dengan medan yang akan dihadapi oleh robot dan juga harus sesuai dengan tugas yang akan dilaksanakan oleh robot nantinya.

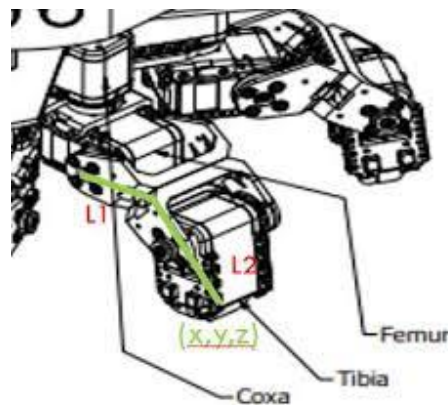


Gambar 2.1 (a) Robot Quadruped

Salah satu jenis robot berkaki yang sering digunakan adalah robot *quadruped* (berkaki empat), robot *quadruped* memiliki beberapa keunggulan jika dibandingkan dengan *hexapod* (berkaki enam), diantaranya adalah memiliki manuver yang sangat baik dalam melewati rintangan. Bahkan dengan metode yang tepat dapat melebihi kecepatan dari hexapod. Tetapi memiliki kestabilan yang kurang baik jika beban robot tidak seimbang^[5]. Robot *Quaruped* ini memiliki perumusan pergerakan kaki atau kinematika gerak dimana kinematika pada robot ini terbagi menjadi dua yaitu *forward kinematics* dan *inverse kinematics*, selain itu juga terdapat trayektori ujung kaki, pola langkah (*gait*) dan manuver gerakan sehingga memungkinkan robot dapat bergerak dengan halus dan presisi. Pada permasalahan kali ini, Robot *quadruped* akan difokuskan pada misi SAR (*search and rescue*) dimana robot akan melewati rintangan yang terjal seperti tanjakan, jalan bebatuan hingga jalan berlubang.

2.1.1 Kinematika Gerak

Kinematika adalah suatu ilmu yang mempelajari mengenai gerakan (meliputi lintasan, kecepatan dan percepatan) dari mesin atau mekanisme tanpa memperhitungkan penyebabnya. Kinematika pada robot dapat didefinisikan sebagai studi pergerakan robot (*motion*) tanpa memperhatikan gaya (*force*) ataupun faktor lain yang mempengaruhi pergerakan robot tersebut. Pada sebuah analisis kinematik, posisi, kecepatan dan akselerasi dan seluruh link dihitung tanpa memperhatikan gaya yang menyebabkan pergerakan tersebut. Pada struktur kaki robot berkaki empat menggunakan struktur kaki serangga yang terdiri dari 3 persendian dan 3 bagian tulang yaitu *coxa*, *femur* dan *tibia* seperti gambar 2.1 (b) berikut.



Gambar 2.1 (b) Struktur Kaki Robot Quadruped
Persamaan Kinematik terbagi 2 bagian yaitu *forward* dan *inverse*.

- Forward kinematik** adalah suatu fungsi yang digunakan untuk mencari posisi *end-effector* ($P_{x,y}$) berdasarkan sudut-sudut pada joint.
- Inverse kinematik** adalah sebaliknya; diberikan posisi *end-effector*, maka yang akan dicari adalah berapa besar nilai sudut yang harus diubah untuk tiap joint *coxa*, *femur*, *tibia* ($teta1$, $teta2$, $teta3$) atau (x,y,z) untuk dapat mencapai posisi *end-effector*.

2.1.2 Body Kinematic

Body kinematics mengolah data posisi relatif setiap ujung kaki terhadap titik pusat bodi robot, kemudian digunakan untuk digunakannya setiap ujung kaki robot sehingga bodi robot dapat bergerak

2.1.3 Trayektori Langkah

Trayektori pada kaki robot merupakan lintasan gerak dari suatu titik *end-point* ke titik *end-point* lainnya. Trayektori ini diperlukan agar perubahan dari suatu titik ke titik lainnya berlangsung sama dan gerakan dari lengan robot menjadi lebih halus. Pada robot berkaki, trayektori ini digunakan untuk mengatur langkah pada kaki robot.

2.1.4 Pola Langkah (Gait)

Pola langkah (*gait*) digunakan untuk menjaga keseimbangan robot saat bergerak. Pola ini merupakan urutan sekuensial tiap kaki robot agar robot dapat bergerak secara dinamis. Jenis pola *gait* yang sering digunakan diantaranya *tripod gait* dan *wave gait*. Pada pola *tripod gait*, pola langkah dibuat dengan menggunakan tiga buah kaki untuk menapak dan tiga buah kaki untuk melangkah. Kemudian pada pola *wave gait*, satu buah kaki robot digunakan bergantian untuk melangkah

2.2. Fuzzy Logic

Metode *fuzzy logic* merupakan salah satu metode kecerdasan buatan (*artificial intelligent*) yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan dengan baik seperti manusia. *Fuzzy logic* merupakan logika yang mempresentasikan nilai samar, ketidakpastian, kebenaran sebagian atau *degree of truth*. Logika fuzzy merupakan pengembangan dari Logika Boolean yang hanya bernilai 0 dan 1 yang memungkinkan nilai keanggotaan antara 0 dan 1 dalam bentuk linguistik, sehingga memungkinkan keberadaan konsep tidak pasti seperti “agak”, “sedikit”, “sedang”, “sangat”, dan sebagainya.

2.2.1. Teori Dasar *Fuzzy Logic*

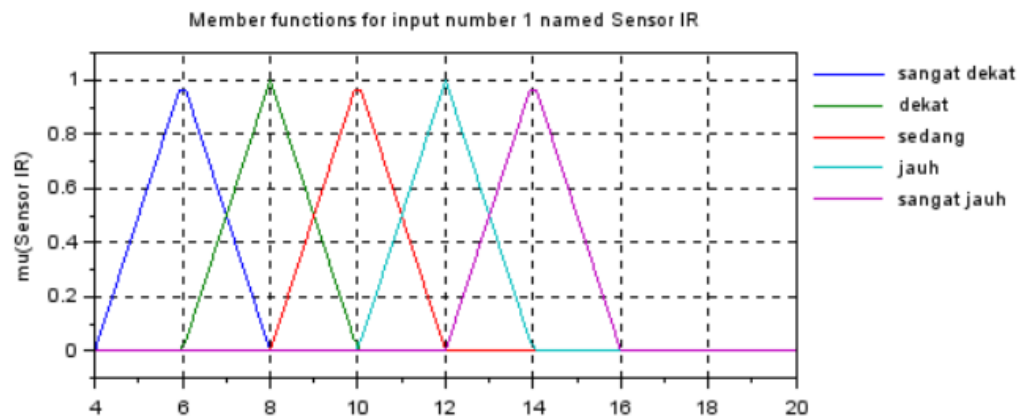
Ada beberapa hal yang harus diketahui dalam memahami sistem *fuzzy logic*, yaitu :

a. Variabel Fuzzy,

Variabel *fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *fuzzy*. Contoh : umur, kecepatan, jarak, suhu, dan lain-lain.

b. Himpunan Fuzzy,

Himpunan *fuzzy* merupakan kelompok yang mewakili suatu keadaan tertentu dalam variabel *fuzzy*. Contoh : 5 himpunan variabel input jarak (sangat dekat, dekat, sedang, jauh dan sangat jauh), dan 5 himpunan variabel kecepatan (sangat lambat, lambat, sedang, cepat dan sangat cepat). Gambar 2.2 dibawah merupakan contoh himpunan *fuzzy* pada variabel kecepatan.



Gambar 2.2 Himpunan *fuzzy* pada variabel kecepatan

c. Semesta Pembicaraan,

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dengan variabel *fuzzy*. Contoh : semesta pembicaraan untuk input jarak [4, 40], dan output kecepatan adalah [0, 100]

d. Domain Himpunan Fuzzy,

Domain himpunan *fuzzy* merupakan seluruh nilai yang izinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Contoh : domain himpunan input jarak (sangat dekat [4, 10], dekat [10, 16], sedang [16, 22], jauh [22, 28], dan sangat jauh [28, 40]). Dan domain himpunan output kecepatan (sangat lambat [0, 20], lambat [20, 40], sedang [40, 60], cepat [60, 80], dan sangat cepat [80, 100]).

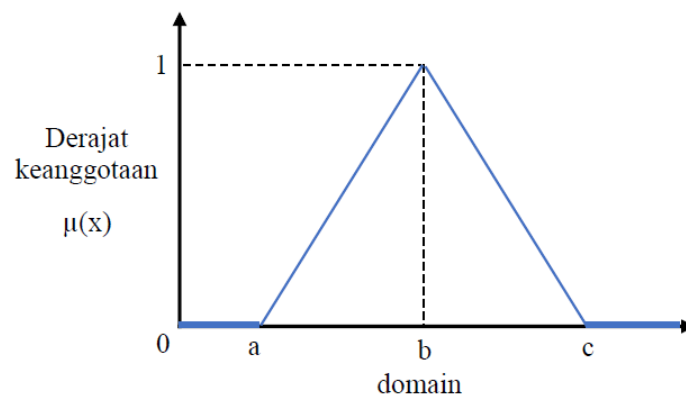
2.2.2. Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) merupakan grafik yang mewakili besar dari derajat keanggotaan masing-masing variabel input yang berada dalam interval antara 0 dan 1. Derajat keanggotaan adalah sebuah variabel x yang

dilambangkan $\mu[x]$. Aturan-aturan (*Rules*) menggunakan nilai keanggotaan sebagai faktor bobot untuk menentukan pengaruhnya pada saat melakukan inferensi dalam menarik kesimpulan. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan dengan melalui grafik pendekatan fungsi kurva linear, segitiga, dan trapesium.

1. Representasi Segitiga

Representasi kurva segitiga adalah gabungan antara dua representasi linear (linear naik dan linear turun)



Gambar 2.3 (a) Representasi Segitiga

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \\ (x - a) / (b - a); & a \leq x \leq b \\ (c - x) / (c - b); & b \leq x \leq c \\ 0; & x \geq c \end{cases}$$

Keterangan :

a = nilai domain yang mempunyai derajat keanggotaan nol

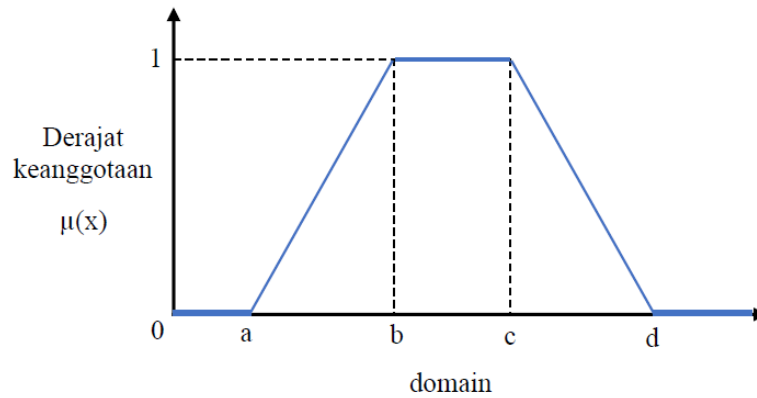
b = nilai domain yang mempunyai derajat keanggotaan satu

c = nilai domain yang mempunyai derajat keanggotaan nol

x = nilai input yang akan diubah kedalam bilangan Fuzzy

2. Representasi Trapesium

Representasi kurva trapesium terdiri dari bentuk kurva segitga dan memiliki beberapa titik yang mempunyai nilai keanggotaan 1 (satu).



Gambar 2.3 (b) Representasi Trapesium

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ dan } x \geq d \\ (x - a) / (b - a); & a \leq x \leq b \\ 1; & b < x \leq c \\ (d - x) / (d - c); & c \leq x \leq d \end{cases}$$

Keterangan :

a = nilai domain yang mempunyai derajat keanggotaan nol

b = nilai domain yang mempunyai derajat keanggotaan satu

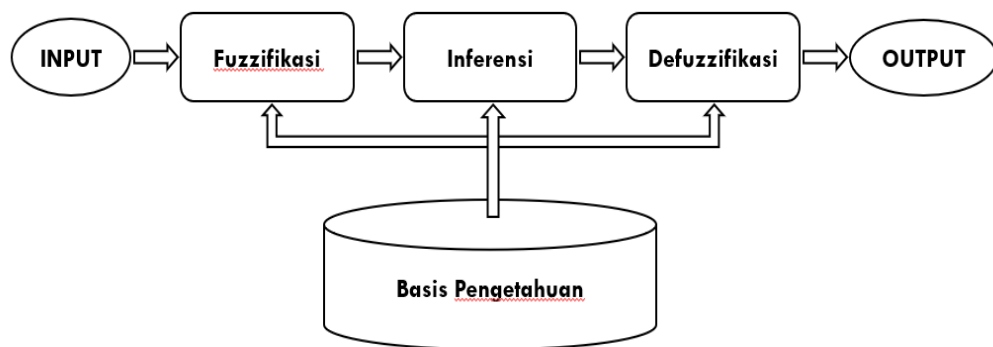
c = nilai domain yang mempunyai derajat keanggotaan satu

d = nilai domain yang mempunyai derajat keanggotaan nol

x = nilai input yang akan diubah kedalam bilangan Fuzzy

2.2.3. Sistem Inferensi Fuzzy

Sistem inferensi fuzzy atau *Fuzzy Inference System* (FIS) adalah cara memetakan ruang input menuju ruang output menggunakan fuzzy logic.



Gambar 2.4 Sistem Inferensi *Fuzzy Logic*

Berdasarkan gambar 2.4 dapat diterangkan bahwa :

- a Fuzzifikasi, merupakan proses mengubah input sistem yang mempunyai nilai tegas (*crisp*) menjadi variabel linguistik (*fuzzy*) menggunakan fungsi keanggotaan yang disimpan pada basis pengetahuan
- b Basis Pengetahuan, adalah kumpulan aturan (*rule*) dalam bentuk pernyataan *if-then* yang akan dibuat
- c Inferensi, merupakan proses mengubah input *fuzzy* menjadi output *fuzzy* dengan cara mengikuti aturan-aturan (*if-then*) yang telah ditetapkan pada basis pengetahuan *fuzzy*.
- d Defuzzifikasi, merupakan proses mengubah hasil dari tahap inferensi menjadi output yang bernilai tegas (*crisp*) menggunakan fungsi keanggotaan yang telah ditetapkan.

2.2.4. Metode Fuzzy Sugeno

Secara umum logika fuzzy sugeno adalah suatu logika yang digunakan untuk menghasilkan keputusan tunggal/*crisp* saat defuzzifikasi, penggunaannya tergantung dari domain masalah yang terjadi. Dimana urutan prosesnya dimulai dari fuzzifikasi, penerapan rule, defuzzifikasi dan output. Fuzzy sugeno pertama kali diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985^[4]. Sehingga metode ini sering dinamakan dengan metode TSK (Takagi-Sugeno Kang). Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output (konsekuen) sistem tidak berupa himpunan Fuzzy, melainkan berupa konstanta atau persamaan linear. Michio Sugeno mengusulkan penggunaan *singleton* sebagai fungsi keanggotaan dari konsekuen. Singleton adalah sebuah himpunan Fuzzy dengan fungsi keanggotaan yang pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut. Metode TSK ini terdiri dari 2 jenis yaitu:

1. Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model fuzzy ini adalah :

IF (x₁ is A₁) ◦ (x₂ is A₂) ◦ (x₃ is A₃) ◦ ... ◦ (x_N is A_N) THEN z = k

dengan A_i adalah himpunan Fuzzy ke-i sebagai antesenden, dan ***k*** adalah suatu konstanta sebagai konsekuen.

2. Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model fuzzy sugeno orde-satu adalah :

$$\mathbf{IF} (x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N) \mathbf{THEN} z = p_1 * x_1 + \dots + p_N * x_N + q$$

dengan A_i adalah himpunan Fuzzy ke- i sebagai antesenden, dan p_i adalah suatu konstanta ke- i dan q juga merupakan konstanta dalam konsekuen.

Berdasarkan model Fuzzy tersebut, ada tahapan-tahapan yang harus dilakukan dalam implementasi metode Sugeno yaitu sebagai berikut:

1) Pembentukan himpunan Fuzzy

Pada tahapan ini variabel input dari system Fuzzy ditransfer ke dalam himpunan Fuzzy untuk dapat digunakan dalam perhitungan nilai kebenaran dari premis pada setiap aturan dalam basis pengetahuan. Dengan demikian tahap ini mengambil nilai-nilai tegas dan menentukan derajat di mana nilai-nilai tersebut menjadi anggota dari setiap himpunan Fuzzy yang sesuai.

2) Aplikasi fungsi implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan Fuzzy akan berhubungan dengan suatu relasi fuzzy. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah sebagai berikut: **IF x is A THEN y is B** dengan x dan y adalah skalar, dan A dan B adalah himpunan Fuzzy. Proposisi yang mengikuti IF disebut sebagai antesenden sedangkan proposisi yang mengikuti THEN disebut konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator Fuzzy seperti, **IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN y is B** dengan \circ adalah operator (misal: OR atau AND). Pada metode Sugeno ini, fungsi implikasi yang digunakan adalah fungsi Minimum (min) yaitu fungsi yang akan memotong output himpunan Fuzzy.

3) Defuzzifikasi (*defuzzification*)

Input dari proses defuzzifikasi adalah himpunan Fuzzy yang dihasilkan dari proses komposisi dan output adalah sebuah nilai. Untuk

aturan IF-THEN Fuzzy dalam persamaan $RU(k) = \text{IF } x_1 \text{ is } A_{1k} \text{ and } \dots \text{ and } x_n \text{ is } A_{nk} \text{ THEN } y \text{ is } B_k$, dimana A_{1k} dan B_k berturut-turut adalah himpunan Fuzzy dalam $U_i R$ (U dan V adalah domain fisik), $i = 1, 2, \dots, n$ dan $x = (x_1, x_2, \dots, x_n)$ U dan $y \in V$ berturut-turut adalah variabel input dan output (linguistik) dari sistem Fuzzy. Defuzzifier pada persamaan di atas didefinisikan sebagai suatu pemetaan dari himpunan Fuzzy B ke dalam $V R$ (yang merupakan output dari inferensi Fuzzy) ke titik tegas $y \in V$. Pada metode Sugeno, *defuzzification* dilakukan dengan perhitungan *Weight Average* (WA):

$$WA = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \dots + \alpha_n z_n}{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n}$$

Keterangan:

WA : Nilai rata-rata,

α_n : Nilai predikat aturan ke- n ,

z_n : Indeks nilai output (konstanta) ke- n

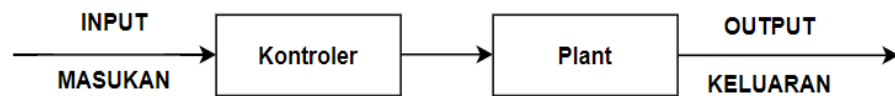
2.3. Sistem Kendali

Sistem kendali (*control system*) merupakan bagian yang terintegrasi dari sistem kehidupan modern saat ini. Sebagai contoh, kendali suhu ruang, mesin cuci, robot, pesawat, dan lain sebagainya. Dengan sistem kendali memungkinkan variabel yang ingin dikendalikan dapat mencapai nilai yang diinginkan dengan mekanisme umpan balik dan pengendalian yang stabil, akurat, dan tepat waktu. Sistem kendali dapat dirancang untuk melakukan pengendalian secara otomatis. Sistem kendali dapat dikatakan sebagai hubungan antara komponen yang membentuk sebuah konfigurasi sistem, yang akan menghasilkan tanggapan sistem yang diharapkan. Jadi harus ada yang dikendalikan, yang merupakan suatu sistem fisis, yang biasa disebut dengan kendalian (*plant*)^[12]. Secara umum, sistem kendali dapat dibedakan menjadi 2 sebagai berikut :

1. Sistem Kendali Terbuka (*open loop*)

Sistem kendali dengan jaringan terbuka (*open loop*) adalah sistem pengendalian dimana keluaran tidak memberikan efek terhadap besaran masukan sehingga variabel yang dikendalikan tidak dapat dibandingkan

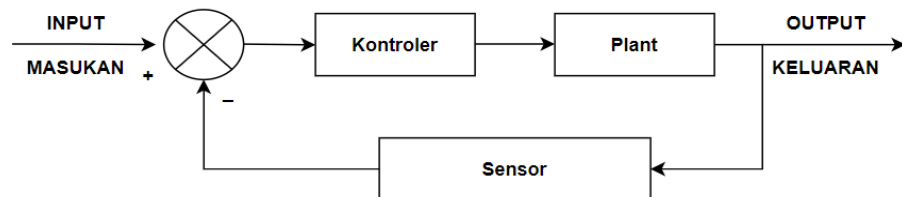
terhadap harga yang diinginkan. Untuk penggambaran lebih jelasnya, adapun penggambaran blok diagram sistem kendali *open loop* seperti pada gambar 2.5(a) berikut.



Gambar 2.5 (a) Sistem *open loop*

2. Sistem Kendali Tertutup (*close loop*)

Sistem kendali dengan jaringan tertutup (*close loop*) adalah sistem pengendalian dimana besaran keluaran memberikan efek terhadap besaran masukan sehingga besaran yang dikendalikan dapat dibandingkan terhadap nilai yang diinginkan. Selanjutnya perbedaan harga yang terjadi antara besaran yang dikendalikan dan penunjukkan alat pencatat digunakan sebagai koreksi pada gilirannya akan merupakan sasaran pengendalian. Istilah “loop tertutup” berarti menggunakan aksi umpan balik untuk memperkecil kesalahan sistem, adapun penggambaran blok diagram sistem kendali *open loop* seperti pada gambar 2.5(b) berikut.



Gambar 2.5 (b) Sistem *close loop*

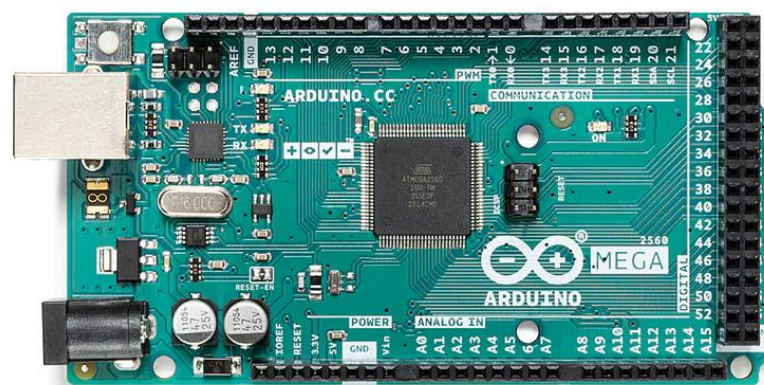
Berikut ini adalah komponen pada sistem kendali tertutup:

- a. **Input** (masukan), merupakan rangsangan yang diberikan pada sistem kontrol, merupakan harga yang diinginkan bagi variabel yang dikontrol selama pengontrolan. Harga ini tidak tergantung pada keluaran sistem
- b. **Output** (keluaran, respons), merupakan tanggapan pada sistem kontrol, merupakan harga yang akan dipertahankan bagi variabel yang dikontrol, dan merupakan harga yang ditunjukkan oleh alat pencatat

- c. **Plant** (kendalian beban), merupakan sistem fisis yang akan dikontrol (misalnya mekanis, elektris, hidraulik ataupun *pneumatic*)
- d. **Controller** (alat kendali), merupakan peralatan atau rangkaian untuk mengontrol beban (sistem). Alat ini bisa digabung dengan penguat
- e. **Feedback Path** (elemen umpan balik), menunjukkan atau mengembalikan hasil pencatatan ke *detector* sehingga bisa dibandingkan terhadap harga yang diinginkan
- f. **Error Detector** (alat deteksi kesalahan), merupakan alat pendeteksi kesalahan yang menunjukkan selisih antara *input* (masukan) dan respons melalui umpan balik (*feedback path*)
- g. **Error** (gangguan), merupakan sinyal-sinyal tambahan yang tidak diinginkan. Gangguan ini cenderung mengakibatkan harga keluaran berbeda dengan harga masukannya, gangguan ini biasanya disebabkan oleh perubahan beban sistem, misalnya adanya perubahan kondisi lingkungan, getaran ataupun yang lain.

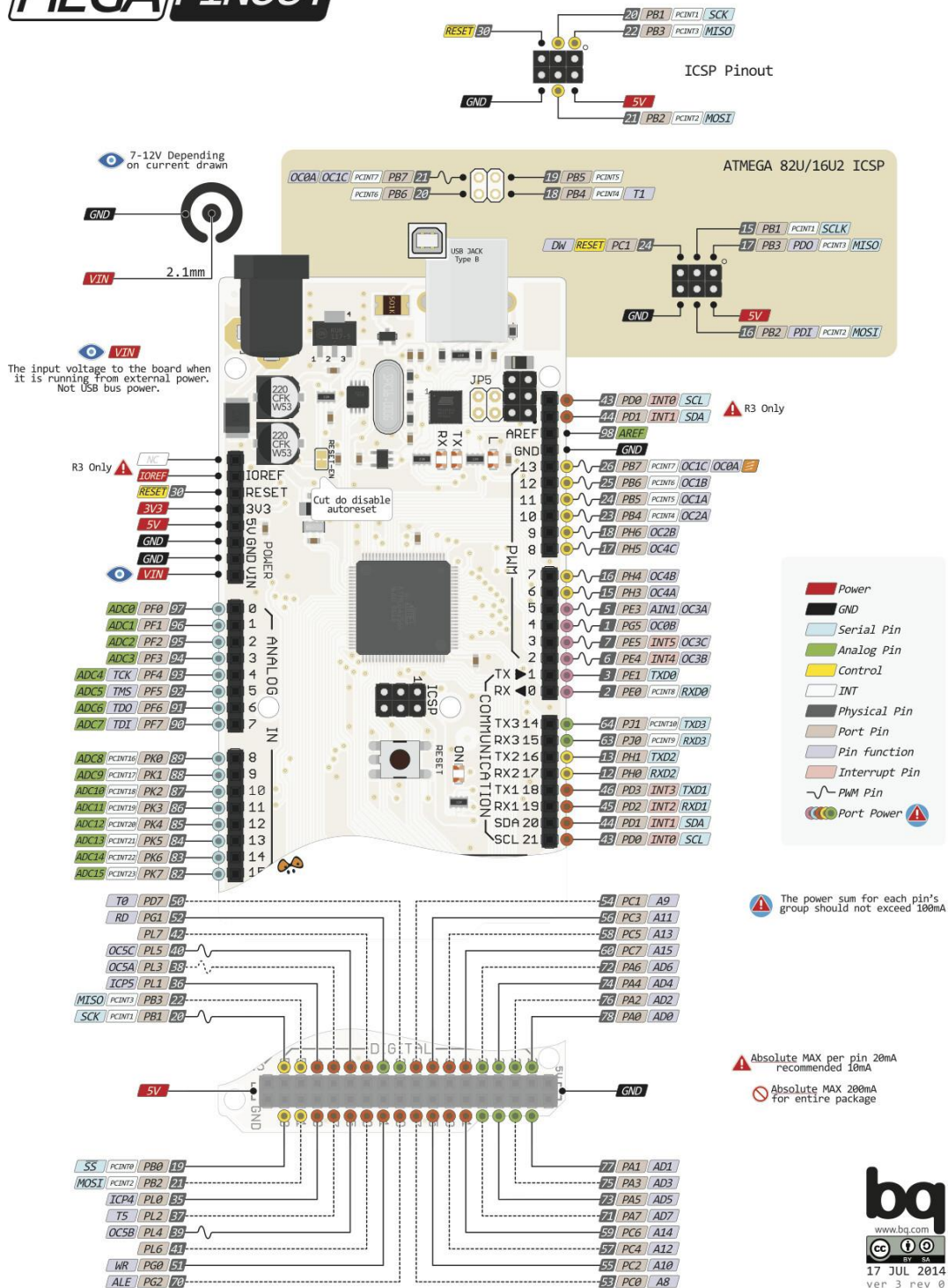
2.3.1 Arduino Mega2560

Arduino adalah papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah chip atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut, dan kemudian menghasilkan output sesuai yang diinginkan^[13].



Gambar 2.6 (a) Arduino Mega2560 Rev3

MEGA PINOUT



Gambar 2.6 (b) Pin Out Arduino Mega2560 Rev3

Pada gambar 2.6 diatas Arduino ATmega2560 ini memiliki 54 pin, masing-masing dari 54 pin digital pada Mega dapat digunakan sebagai input atau output,

menggunakan *pinMode ()*, *digitalWrite ()*, dan *digitalRead ()* fungsi. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up internal* yang (terputus secara *default*) dari 20- 50 KOhms. Selain itu, beberapa pin memiliki fungsi khusus: Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan data serial (TX) TTL. Pin 0 dan 1 juga terhubung ke pin dari ATmega8U2 USB-to-TTL *Chip Serial*.

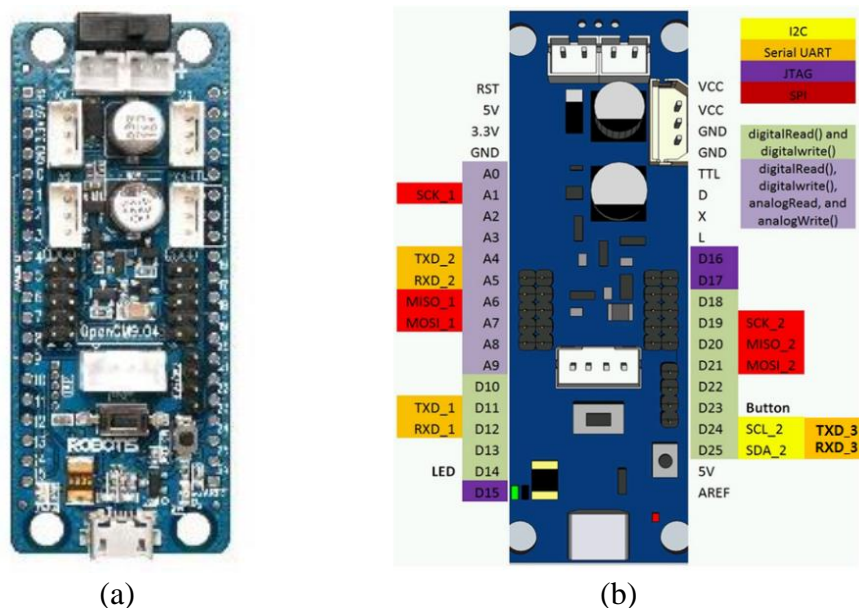
1. Interupsi Eksternal: 2 (*interrupt 0*), 3 (*interrupt 1*), 18 (*interrupt 5*), 19 (*interrupt 4*), 20 (*interrupt 3*), dan 21 (*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat *attachInterrupt ()* fungsi untuk rincian.
2. PWM: 0 13. Memberikan output PWM 8-bit dengan fungsi *analog Write ()*.
3. SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga pecah pada *header ICSP*, yang secara fisik kompatibel dengan Uno, *Duemilanove* dan *Diecimila*.
4. LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin tinggi nilai, LED menyala, ketika pin rendah, itu *off*.
5. I2C: 20 (SDA) dan 21 (SCL). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan kawat (dokumentasi di website *Wiring*). Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin I2C pada *Duemilanove* atau *Diecimila*.

Arduino Mega 2560 memiliki 16 input analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* mereka mengukur dari tanah ke 5 volt, meskipun adalah mungkin untuk mengubah batas atas dari kisaran mereka menggunakan pin AREF dan fungsi *analogReference*. Ada beberapa pin lainnya di papan:

1. AREF. tegangan referensi untuk input analog. Digunakan dengan *analogReference()*.
2. *Reset*. Bawa garis LOW ini untuk me-reset mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset untuk perisai yang menghalangi satu di papan tulis.

2.3.2 Open CM9.04

OpenCM 9.04 merupakan serangkaian papan mikrokontroler yang dikeluarkan oleh perusahaan ROBOTIS yang bersifat Open Source. Kontroler ini berbasis STM32F013CB dengan menggunakan chip mikrokontroler ARM CortexM3 CPU dengan memori flash sebesar 128 kB dan SRAM sebesar 20kB. Kontroler ini didesain dengan skematik catu daya yang mampu memberikan suplai tegangan 5 volt dan 3,3 volt DC. Tegangan 5 volt terdapat pada jalur TTL sedangkan tegangan 3,3 volt digunakan untuk suplay mikrokontroler, port sensor 5 pin, dan port komunikasi 4 pin. Mikrokontroler ini dilengkapi dengan konektor 3 pin TTL untuk mengoperasikan smart servo dynamixel jenis AX, MX, XL-series. Selain itu terdapat pula pin GPIO sebanyak 26 pin yang dapat digunakan untuk input analog, USB, CAN, SPI, I2C, dan serial port. Penggunaan pin GPIO hanya dapat diakses apabila openCM 9.04 diprogram menggunakan software Robotis OpenCM IDE^[16].



Gambar 2.7 (a) Open CM9.04, (b) Konfigurasi Pin Data

Pada gambar 2.7 merupakan bagian *hardware* Open CM9.04 yang dengan spesifikasi sebagai berikut:

1. Tombol power sebagai Saklar catu daya
2. Soket baterai Soket penghubung baterai
3. 3-Pin dynamixel TTL, sebagai konektor dynamixel berbasis TTL
4. Port Komunikasi, sebagai penghubung antar komponen dengan komunikasi UART BT-110A, BT-210, ZIG-110A, LN-101. 4 Pin port komunikasi pada OpenCM 9.04 menggunakan Serial2 (USART2)
5. Tombol user, digunakan untuk memasuki mode pemulihan (*recovery*) firmware pada OpenCM9.04
6. Ext. ADC Ref Jumper, sebagai pengubah tegangan referensi analog
7. Mikro USB-B Jalur komunikasi, mengunduh program, dan sebagai catu daya 5V
8. Status LED, sebagai indikator percobaan program LED OpenCM 9.04
9. Tombol reset, digunakan untuk reset CPU
10. 4-Pin JTAG/SWD digunakan untuk menghubungkan ST-LINK dan sejenisnya
11. 5-Pin sensor eksternal, digunakan untuk mengakses produk sensor ROBOTIS
12. Pin GPIO 2,5mm, digunakan untuk mengakses perangkat eksternal ke CPU OpenCM 9.04

2.3.3 Komunikasi Data Serial

Dalam komunikasi data serial, data dikirim dengan bentuk pulsa listrik kontinyu yang disebut bit. Data dikirim satu bit demi satu bit secara berurutan melalui kanal komunikasi yang telah ditentukan. Penerima juga menerima data dalam bentuk bit - bit pulsa listrik yang kontinyu. Ada tiga metode yang sering dijumpai pada komunikasi data serial yaitu

Simpleks, half dupleks, dan full duplex.

1. Transmisi data *simplex*, data hanya dapat dikirim dalam satu arah saja.
2. Transmisi data *half duplex*, data dapat ditransmisikan dalam dua arah secara bergantian.

3. Transmisi data *full duplex*, merupakan transmisi data dua arah dimana data dapat diterima oleh system dan sekaligus system tersebut dapat mengirimkan data secara bersamaan. Transmisi data full duplex dapat dijumpai pada system telepon.

Dalam komunikasi data serial pada Arduino, biasanya digunakan beberapa komunikasi sebagai berikut :

- a. **USB** koneksi ke komputer (tapi tidak untuk komunikasi serial pada pin 0/1)
- b. **SPI** (*serial peripheral interface*), adalah protokol *master-slave* yang sinkron dan *full duplex*; data yang dikirim oleh *master* atau *slave* disinkronkan oleh *clock* yang dibangkitkan oleh master. *SPI interface* dapat menggunakan 3 atau 4 kabel
- c. **I2C**, I2C berasal dari kata IIC, kepanjangannya *Inter-Integrated Circuit*. Sedangkan TWI kepanjangannya adalah *Two Wire Interface*. Ketiga istilah tersebut adalah sama, merujuk pada protokol komunikasi serial sinkron dengan dua kabel yakni SDA (*Serial Data*) dan SCL (*Serial Clock*). SDA merupakan data serialnya, sedangkan SCL adalah jalur *clock* sinkronisasinya.
- d. **UART** (*Universal Asynchronous receiver/ Transmitter*), fungsinya adalah untuk mengirim atau menerima data serial UART hanya menggunakan dua kabel untuk menghubungkan satu perangkat dengan perangkat lainnya. Antara lain dengan melakukan komunikasi pin *transmitter* dan pin *receiver*, sehingga untuk bisa dua perangkat berkomunikasi dengan serial maka kita harus menghubungkan :
TX dengan RX dan RX – TX

2.4. Definisi Sensor

Sensor merupakan bagian dari suatu piranti pengukuran maupun sistem pengendalian yang langsung berhubungan baik secara kontak langsung maupun tak langsung dengan lingkungan di luar piranti atau sistem. Sensor merupakan pintu masuk signal atau besaran dari luar yang selanjutnya akan diproses didalam piranti pengukuran atau sistem pengendalian. Sensor berada pada titik terdepan dalam

keseluruhan proses untuk mendapatkan besaran fisis, kimiawi atau biologis dari obyek menjadi suatu informasi.

Metode deteksi yang dipergunakan meliputi dua macam, yaitu kontak dan non-kontak. Untuk sebuah sensor yang menggunakan metode deteksi kontak, maka sensor secara fisik akan bersinggungan dengan obyek ukur. Sedangkan metode non-kontak menempatkan sensor dengan obyek terpisah secara fisik. Untuk metode non-kontak maka gelombang suara, radiasi elektromagnetik dan radiasi lain merupakan media penyampai besaran dari obyek ke sensor. Secara teknis, sensor dibedakan menjadi 2 macam, yaitu :

1. Sensor Digital

Sensor Digital adalah sensor yang menghasilkan sinyal keluaran diskrit. Sinyal diskrit akan non-kontinu dengan waktu dan dapat direpresentasikan dalam “bit”. Sebuah sensor digital biasanya terdiri dari sensor, kabel dan pemancar. Sinyal yang diukur akan diwakili dalam format digital. Output digital dapat dalam bentuk Logika 1 atau logika 0 (ON atau OFF).

2. Sensor Analog

Sensor Analog adalah sensor yang menghasilkan sinyal output yang kontinu atau berkelanjutan. Sinyal keluaran kontinu yang dihasilkan oleh sensor analog ini sebanding dengan pengukuran. Berbagai parameter analog ini diantaranya adalah suhu, tegangan, tekanan, pergerakan dan lain-lainnya.

ADC (*Analog to Digital Converter*) adalah perangkat elektronika yang berfungsi mengubah sinyal analog menjadi sinyal digital. ADC memiliki 2 karakter prinsip, yaitu:

1). Kecepatan sampling

Kecepatan sampling ADC adalah menyatakan “seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu”. Kecepatan sampling biasanya dinyatakan dalam *sample per second* (SPS).

2). Resolusi

Resolusi ADC adalah menentukan “ketelitian nilai hasil konversi ADC”. Sebagai contoh ADC 8 Bit akan memiliki output 8 Bit data digital. Ini berarti sinyal input dapat dinyatakan dalam 255 atau (-1) nilai diskrit. ADC 12 Bit akan

memiliki output 12 Bit data digital. Ini berarti sinyal input dapat dinyatakan dalam 4096 data diskrit. Dari contoh di atas dapat disimpulkan bahwa ADC 12 bit akan memberikan resolusi yang lebih baik daripada 8 Bit.

Prinsip Kerja ADC adalah mengonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 Bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar $60\% \times 255 = 153$ (bentuk desimal) atau 10011001 (bentuk biner).

$$ADC\ Signal = \frac{Sample}{Max\ value} \times reference\ voltage$$

Bentuk komunikasi yang paling mendasar antara wujud digital dan analog adalah piranti yang disebut komparator. Secara sederhana, komparator bertujuan untuk membandingkan dua tegangan pada kedua terminal inputnya. Bergantung pada tegangan mana yang lebih besar, outputnya akan berupa sinyal digital 1 (*high*) atau 0 (*low*). Komparator ini digunakan secara luas untuk sinyal alarm ke komputer atau sistem pemrosesan digital. Elemen ini juga merupakan satu bagian dengan konverter analog ke digital dan digital ke analog. Sebuah komparator dapat tersusun dari sebuah *op-amp* yang memberikan output terpotong untuk menghasilkan level yang diinginkan untuk kondisi logika (+5 dan 0 untuk TTL 1 dan 0). Komparator komersil didesain untuk memiliki level logika yang diperlukan pada bagian outputnya.

2.4.1 Sensor Jarak

Sensor jarak atau *proximity* adalah sensor yang digunakan untuk menentukan kedekatan suatu objek tanpa adanya kontak fisik. Karakteristik dari sensor ini adalah mendeteksi objek benda dengan jarak yang cukup dekat. Ada 3 jenis sensor jarak (*proximity*) yaitu:

1). *Inductive Proximity*

Inductive Proximity berfungsi untuk mendeteksi objek logam. Prinsip kerja dari *proximity inductive* adalah apabila ada tegangan sumber maka osilator yang ada pada *proximity* akan membangkitkan medan magnet dengan frekuensi

tinggi. Jika sebuah benda logam di dekatkan pada permukaan sensor maka medan magnet akan berubah. Perubahan pada osilator ini akan dideteksi sensor sebagai sinyal adanya objek. Contoh *Inductive Proximity* ini biasanya digunakan pada metal detector di bandara. Sensor proximity ini akan mendeteksi adanya objek logam walaupun tidak terlihat.

2). *Capacitive Proximity*

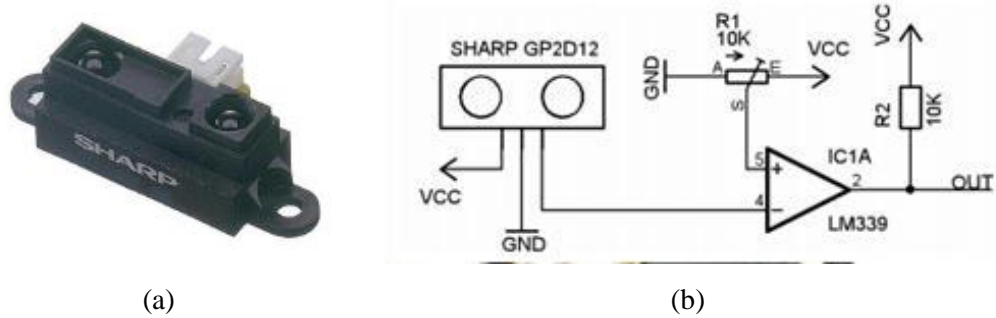
Sensor *Capacitive Proximity* mampu mendeteksi objek logam maupun non logam. Prinsip kerja dari *proximity capacitive* adalah dengan cara mengukur perubahan kapasitansi mean listrik sebuah kapasitor yang disebabkan oleh sebuah objek yang mendekatinya. *Capacitive proximity* ini biasanya digunakan pada bumper mobil atau bagian mobil yang lainnya. Manfaat sederhananya adalah untuk memudahkan mobil parkir, karena sensor ini akan bekerja apabila mendeteksi benda-benda pada jarak tertentu sehingga mobil tidak akan menabrak benda tersebut.

3). Sensor *Proximity* Optik

Sensor ini mendeteksi adanya objek dengan cahaya biasanya adalah infra red. *Proximity* optik ini terdiri dari sebuah cahaya dan penerima (*receptor*) yang mendeteksi sebuah benda dengan refleksi. Jika benda dalam jarak yang sensitif atau benda mengenai cahaya dari sensor, maka cahaya akan memantul kembali ke penerima dan mengindikasikan bahwa terdapat sebuah benda yang tertangkap sensor.

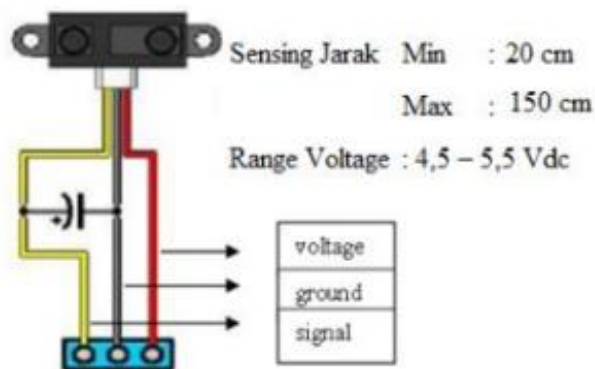
2.4.2 Sensor Sharp GP2Y0A41SK0F

Sharp GP2Y041SK0F adalah sensor pengukur jarak yang terdiri dari kombinasi terintegrasi PSD (*position sensitive detector*), IR-ED (*infrared emitting diode*) dan rangkaian pemrosesan sinyal. Variasi reflektifitas objek, suhu lingkungan dan durasi operasi tidak mudah terpengaruh oleh deteksi jarak karena mengadopsi metode triangulasi. Sensor ini menggunakan prinsip pantulan sinar infra merah. Dalam aplikasi ini nilai tegangan keluran dari sensor yang berbanding terbalik dengan hasil pembacaan jarak dikomparasi dengan tegangan referensi komparator^[18].



Gambar 2.8 (a) Sharp GP2Y041SK0F, (b) Rangkaian Sistem Komparator Sensor

Prinsip kerja dari rangkaian komparator sensor sharp GP2Y041SK0F pada gambar 2.8(c) diatas adalah jika sensor mengeluarkan tegangan melebihi tegangan referensi, maka keluaran dari komparator akan berlogika rendah. Jika tegangan referensi lebih besar dari tegangan sensor maka keluaran dari komparator akan berlogika tinggi. Selain menggunakan komparator, untuk mengakases sensor jarak sharp GP2Y041SK0F dapat dengan menggunakan prinsip ADC, atau dengan kata lain mengolah sinyal analog dari pembacaan sensor sharp GP2Y041SK0F ke bentuk digital dengan bantuan pemrograman



Gambar 2.8 (c) Pin *Out* pada Sensor Sharp GP2Y041SK0F
Perangkat ini mengeluarkan tegangan yang sesuai dengan jarak deteksi dengan spesifikasi sebagai berikut :

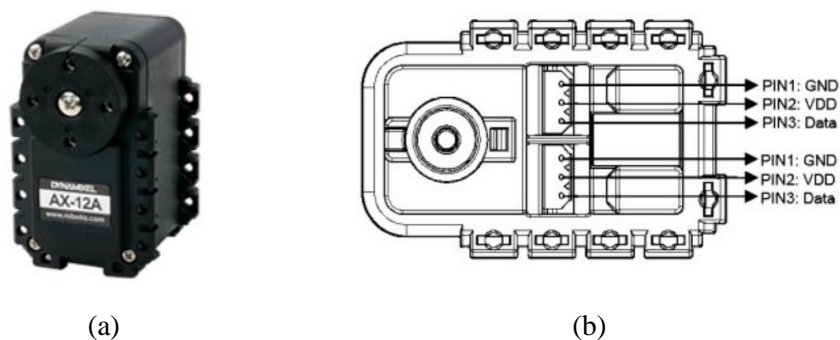
- a. Pengukuran jarak : 4 cm ~ 30 cm
- b. Jenis keluaran (output) : analog
- c. Ukuran komponen : (29,5 × 13 × 21,6) mm
- d. Arus konsumsi : Tip. 33 mA
- e. Tegangan suplai : 4,5V - 5,5V

2.5. Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (servo), sehingga dapat *diset-up* atau diatur untuk menentukan dan memastikan posisi sudut dari poros output motor. Motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo. Penggunaan sistem kontrol *close loop* pada motor servo berguna untuk mengontrol gerakan dan posisi akhir dari poros motor servo.

2.5.1. Motor Servo Dynamixel AX-12

Motor servo dynamixel AX-12 adalah aktuator dan modular yang terdiri atas *gear reducer*, motor DC (*Direct Current*), dan kontroler dengan kemampuan komunikasi UART TTL *half duplex* dengan kontroler utama. Setiap motor servo AX-12A memiliki dua buah konektor yang masing-masing konektor tersebut terdapat 3 pin.



Gambar 2.9 (a) Motor Servo Dynamixel AX-12A, (b) Pin Data Servo

Seperti yang terlihat pada gambar 2.9, pin ketiga diantaranya adalah Data, Vdd (tegangan suplai), dan Gnd (*ground*). Pin data yang digunakan untuk komunikasi dengan kontroler.

Spesifikasi dari Dynamixel AX-12A adalah sebagai berikut:

- a. Berat : 54.6 gram
- b. Dimensi : (32 x 50 x 40) mm

- c. Resolusi : 0.29°
- d. Rasio roda gigi : 254 : 1
- e. Torsi : 15 kgf.cm
- f. Kecepatan : 59 rpm (ketika tegangan 12 volt dan tanpa beban)
- g. Derajat tempuh : 0 ~ 300°
- h. Suhu kerja : -5 ~ 70°
- i. Tegangan kerja : 9 ~ 12 volt (direkomendasikan 11.1V)
- j. Tipe protokol : serial asinkron half duplex (8bit, 1stop, no parity)
- k. ID : 0 ~ 253 (254)
- l. Komunikasi : 7343bps ~ 1Mbps

Kecepatan Motor Servo AX-12 ini adalah kecepatan bergerak ke suatu posisi tujuan, rentang dan satuan nilai dapat bervariasi tergantung pada mode operasi.

a. *Joint mode* (mode bersama)

0 ~ 1,023 (0x3FF) dapat digunakan, dan unit ini sekitar 0,111rpm.

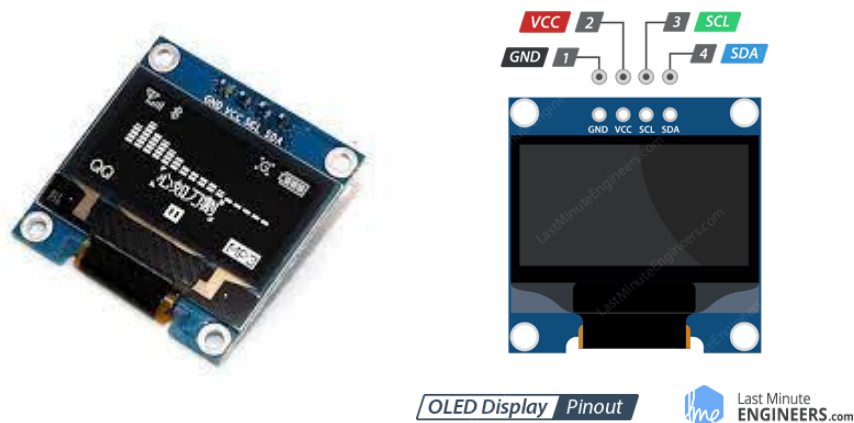
Jika disetel ke 0, berarti rpm maksimum motor digunakan tanpa mengontrol kecepatan. Jika 1023, itu sekitar 114rpm. Misalnya, jika disetel ke 300, itu sekitar 33,3 rpm. Motor tidak dapat melebihi rpm maksimum dengan nilai kecepatan gerak yang lebih tinggi.

b. *Wheel mode* (mode roda)

0 ~ 2.047 (0x7FF) dapat digunakan, unitnya sekitar 0,1%. Jika nilai dalam kisaran 0 ~ 1,023 digunakan, itu dihentikan dengan mengatur ke 0 sambil berputar ke arah CCW. Jika nilai dalam kisaran 1.024 ~ 2.047 digunakan, itu dihentikan dengan mengatur ke 1.024 sambil berputar ke arah CW. Artinya, bit ke-10 menjadi bit arah untuk mengontrol arah. Dalam Mode Roda, hanya kontrol output yang dimungkinkan, bukan kecepatan. Misalnya, jika diatur ke 512, itu berarti output dikendalikan oleh 50% dari output maksimum.

2.6. OLED (*Organic Light Emitting Diode*)

OLED adalah sebuah komponen semikonduktor yang solid seperti halnya komponen LED (*Light Emitting Diode*) yang dibuat dengan menyisipkan beberapa lembar lapisan tipis organik diantara dua konduktor. OLED 128×64 bertipe SSD1306, layar ini berjenis OLED monokrom yang memiliki resolusi 128 piksel (lebar) dan 64 piksel (tinggi) dengan ukuran layar 0,96 inci. Layar ini dapat menghasilkan cahaya sendiri tanpa perlu lampu latar seperti LCD 16×2 biasa.



Gambar 2.10 OLED 128×64

Display OLED berfungsi untuk menampilkan hasil data dari Arduino dengan menggunakan komunikasi serial i2c^[14]. Pada OLED terdapat 4 pin yaitu SDA, SCL, Vcc, dan Gnd. Dengan spesifikasi sebagai berikut :

- a. Resolusi tinggi : 128 * 64
- b. Sudut pandang : > 160°
- c. Konsumsi daya : 0,08W (layar penuh)
- d. Tegangan : 3V ~ 5V DC
- e. Suhu Kerja : -30 c ~ 70 c
- f. Ukuran Modul : 27.0MM * 27.0MM * 4.1MM
- g. Antarmuka : I2C/IIC, hanya perlu 2 IO.
- h. IC Driver : SSD1306

2.7. Catu Daya

Pencatu Daya adalah sebuah peranti elektronika yang berguna sebagai sumber daya untuk peranti lain, terutama daya listrik. Pada dasarnya pencatu daya

bukanlah sebuah alat yang menghasilkan energi listrik saja, namun ada beberapa pencatu daya yang menghasilkan energi mekanik, dan energi yang lain. Catu-daya dengan tegangan rendah dengan tegangan keluaran AC dan DC untuk penggunaan umum. *DC power supply* merupakan penyuplai tegangan yang berfungsi untuk memberikan arus DC (*Dirrect Current*). Jenis catu daya ini memiliki polaritas yang tetap, yakni untuk bebannya terdapat positif dan negatif.

2.7.1. Baterai Lithium Polymer (LiPo)

Di dalam baterai ini, menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada *charging* dan *discharging rate*. Masalah ini sebenarnya bisa diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini dianggap tidak dapat untuk diaplikasikan pada keadaan sehari-hari.



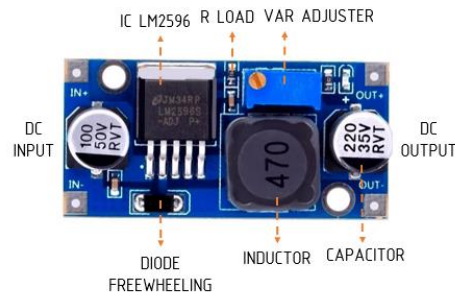
Gambar 2.11 Baterai LiPo

2.7.2. Regulator *Stepdown*

DC Stepdown merupakan sebuah regulator yang dapat digunakan sebagai penyetabil tegangan. LM2596 berfungsi sebagai *Step-Down DC converter* dengan *current rating* 3A. Regulator seri LM ini memiliki beberapa variasi tegangan output tetap 3,3 Volt, 5 Volt, 9 Volt, 12 Volt dan versi output yang dapat disesuaikan dengan kebutuhan. Seri LM2596 beroperasi pada frekuensi

switching 150 KHz, sehingga memungkinkan komponen berukuran lebih kecil dari apa yang diperlukan dengan regulator beralih ke frekuensi lebih rendah.

MODUL REGULATOR LM2596



Gambar 2.12 *Step Down Regulator*

Pada perancangan rangkaian elektronik digunakan modul konverter DC to DC penurun tegangan LM 2596. Spesifikasi yang dimiliki antara lain :

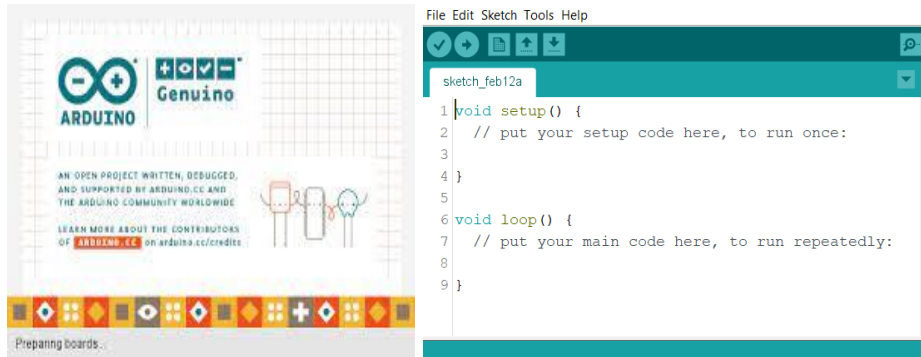
- Tegangan Input : 3,2 V – 40 V
- Tegangan Output : 1,25 V – 37 V
- Arus Output : 3 A
- Ukuran : 50 x 23 x 14mm

2.8. Software Aplikasi

Perangkat Lunak atau *Software* adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya, dimana data elektronik yang disimpan dapat berupa program atau instruksi yang akan dijalankan oleh perintah, maupun catatan-catatan yang diperlukan oleh komputer untuk menjalankan perintah yang dijalankannya. Untuk mencapai keinginannya tersebut dirancanglah suatu susunan logika, logika yang disusun ini diolah melalui perangkat lunak, yang disebut juga dengan program beserta data-data yang diolahnya. Pengolahan pada *software* ini melibatkan beberapa hal, diantaranya sistem operasi, program, dan data.

2.8.1 Arduino IDE

Gambar 2.13 merupakan *icon* dan tampilan dari *software* ArduinoIDE. *Arduino Integrated Development Environment (IDE)* adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory microcontroller*.



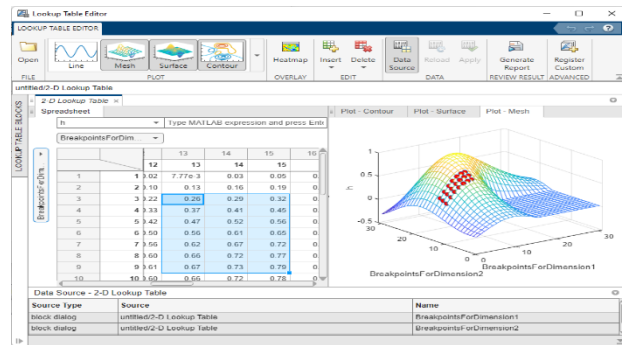
Gambar 2.13 Tampilan *Software* Arduino IDE

Secara umum, struktur program pada Arduino dibagi menjadi dua bagian yaitu *setup* dan *loop*. Bagian *setup* adalah bagian yang menempatkan kode-kode inialisasi sistem sebelum masuk ke dalam bagian *loop (body)*. Secara prinsip, *setup* merupakan bagian yang dieksekusi hanya sekali yaitu pada program dimulai (*start*). Sedangkan bagian *loop* adalah bagian yang merupakan inti utama dari program Arduino dan bagian ini yang dieksekusi secara terus menerus.

2.8.2 Matlab

Matlab adalah singkatan dari *Matrices Laboratory* yang dikembangkan oleh MathWork, dan termasuk bahasa pemrograman tingkat tinggi. Matlab dikembangkan sebagai Bahasa pemrograman sekaligus sebagai alat visualisasi yang menawarkan banyak kemampuan untuk menyelesaikan berbagai kasus yang berhubungan langsung dengan disiplin keilmuan Matematika, seperti bidang rekayasa teknik, fisika, statistika, komputasi dan modeling. Dengan memanfaatkan Matlab, pengguna dapat melakukan analisis data, mengembangkan algoritma, dan membuat model maupun aplikasi. Bahasa, tools, dan fungsi-fungsi *built-in* akan memudahkan pengguna untuk mengeksplorasi berbagai pendekatan dan memperoleh solusi dengan lebih cepat dibandingkan apabila menggunakan *spreadsheets* atau Bahasa pemrograman

tradisional, seperti C/C++ atau Java™. Gambar 2.14 merupakan *icon* dan tampilan dari *software* Matlab.



Gambar 2.14 Tampilan *Software* Matlab

