

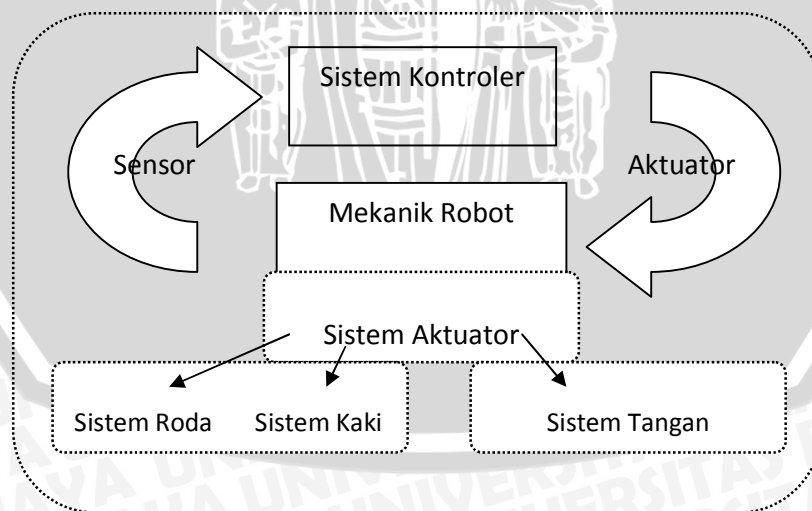
## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Sebelumnya

Penelitian sebelumnya mengenai analisa manipulator telah dilakukan oleh Syafruddin, RM (2012). Penelitian ini menitik beratkan pada jenis kontrol kinematika dengan lima *DOF* (*Degree of freedom*) memanfaatkan program mikrokontroler, dan sistem kontrol gerak *holomic*. Gerak *holomic* merupakan gerak yang serupa dengan ujung pensil atau bulpoin kearah permukaan kertas sesuai dengan keinginan. Penelitian ini memiliki dimensi disesuaikan dengan ukuran lengan manusia yang sebenarnya, yaitu panjang lengan antara bahu dan siku ( $l_1$ ) = 20 cm, panjang lengan antara sikudengan pergelangan tangan ( $l_2$ ) = 25 cm, tinggi tiang penyanggah ( $l_3$ ) = 80 cm. Mikrokontroler yang digunakan adalah *ATMega 8535* sebagai pemroses masukan dari sensor *potensiometer* dan mikrokontroler *ATTiny 2313* sebagai pemroses keluaran yang akan menggerakkan motor servo pada lengan robot. Kesimpulan yang dihasilkan dari penelitian ini mampu menghasilkan kemiripan mengikuti gerak lengan manusia dengan presentase keberhasilan 86% (Syafruddin, RM, 2012).

### 2.2 Robot Manipulator

Robot industri pada awalnya diilustrasikan sebagai robot manipulator yaitu robot tangan yang memiliki dua lengan (dilihat dari persendian), dan pergelangan. Di ujung pergelangan dapat diinstal berbagai tool sesuai dengan fungsi yang diharapkan (Endra, 2006).



Gambar 2.1 Sistem Robot Dan Orientasi Fungsi

Sumber :Endra Pituwarno, (2006)



Pada Gambar diatas memperlihatkan anatomi robot industri yang Komponen utamanya terdiri dari empat bagian, yaitu:

1. Manipulator

Manipulator adalah bagian mekanik yang dapat difungsikan untuk memindah, mengangkat dan memanipulasi benda kerja.

2. Sensor

Sensor adalah komponen berbasis instrumentasi (pengukuran) yang berfungsi sebagai pemberi informasi tentang berbagai keadaan atau kedudukan dari bagian-bagian manipulator.

3. Aktuator

Aktuator adalah komponen penggerak yang jika dilihat dari prinsip penghasil gerakanya dapat dibagi menjadi 3 bagian, yaitu penggerak berbasis motor listrik (motor DC dan motor AC),

4. Kontroler

Kontroler adalah rangkaian elektronik berbasis mikroprosesor yang berfungsi sebagai pengatur seluruh komponen dalam membentuk fungsi kerja. (Endra, 2006)

### 2.2.1 Klasifikasi Robot Manipulator

Secara umum struktur robot dapat dibedakan menurut sumbu koordinat yang digunakan.

Tabel 2.1 Struktur Umum Robot

| No. | Jenis Robot        | Sumbu 1 | Sumbu 2 | Sumbu 3 | Total Rotasi |
|-----|--------------------|---------|---------|---------|--------------|
| 1   | <i>Cartesian</i>   | P       | P       | P       | 0            |
| 2   | <i>Cylindrical</i> | R       | P       | P       | 1            |
| 3   | <i>Spherical</i>   | R       | R       | P       | 2            |
| 4   | <i>SCARA</i>       | R       | R       | P       | 2            |
| 5   | <i>Articulated</i> | R       | R       | R       | 3            |

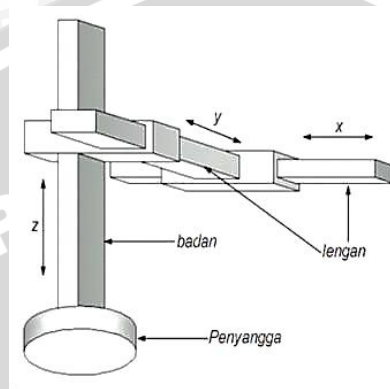
Sumber: Endra Pituwarno, (2006)

Catatan :  $P$ =Prismatic joint yaitu pergeseran sepanjang sumbu translasi.

$R$ = Revolute joint yaitu perputaran pada sumbu rotasi.

a. *Robot Cartesian*

Struktur Robot ini terdiri dari tiga sumbu linier (*prismatic*). Masing-masing sumbu dapat bergerak ke arah sumbu x-y-z. Keuntungan robot ini adalah pengontrolan posisi yang mudah dan mempunyai struktur yang lebih kokoh.



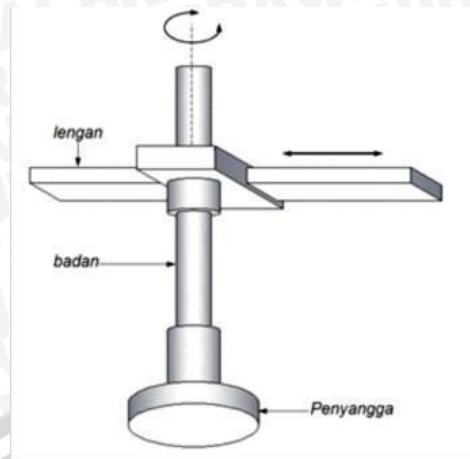
Gambar 2.2 Konfigurasi cartesian

Sumber :Endra Pituwarno, (2006)

Pada gambar diatas memperlihatkan manipulator berkonfigurasi *cartesian* dimana secara relatif adalah yang paling kokoh untuk tugas mengangkat beban yang berat. Struktur ini banyak dipakai secara permanen pada instalasi pabrik baik untuk mengangkat dan memindah barang-barang produksi maupun untuk mengangkat peralatan-peralatan berat pabrik ketika melakukan kegiatan instalasi.

b. *Robot Silindris*

Struktur dasar dari robot silindris adalah terdiri dari *Horisontal Arm* dan *Vertical Arm* yang dapat berputar pada *basel* landasannya. Jika dibandingkan dengan robot kartesian, robot silindris mempunyai kecepatan gerak lebih tinggi dari *end effectornya*, tapi kecepatan tersebut tergantung momen inersia dari beban yang dibawahnya.



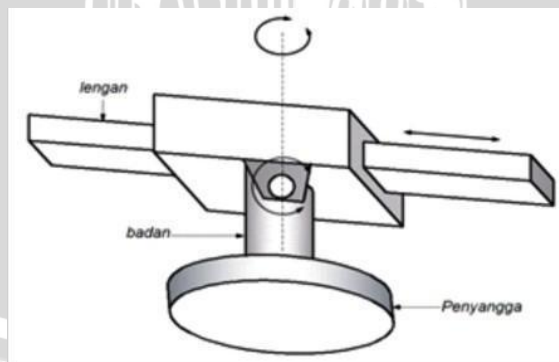
Gambar 2.3. Konfigurasi Silinder

Sumber :Endra Pituwarno, (2006)

Konfigurasi silinder mempunyai kemampuan jangkauan berbentuk ruang silinder yang lebih baik, meskipun sudut ujung lengan terhadap garis penyangga tetap. Konfigurasi ini banyak diadopsi untuk sistem *gantry* atau *crane* karena strukturnya yang kokoh untuk tugas mengangkat beban.

c. *Robot Spheris/Polar*

Konfigurasi struktur robot ini mirip dengan sebuah tank dimana terdiri atas *Rotary Base*, *Elevated Pivot*, dan *Telescopic Arm* (lihat gambar 4). Keuntungan dari robot jenis ini adalah fleksibilitas mekanik yang lebih baik.



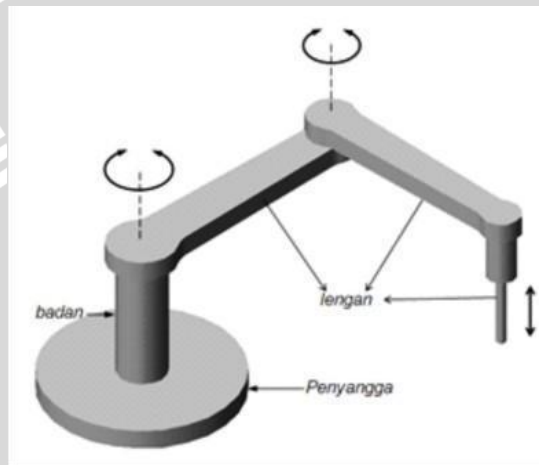
Gambar 2.4. Konfigurasi Polar

Sumber: Endra Pituwarno, (2006)

Pada gambar diatas terlihat konfigurasi polar dimana badan dapat berputar ke kiri atau kanan. Sendi pada badan dapat mengangkat atau menurunkan pangkal lengan secara polar. Lengan ujung dapat digerakkan maju-mundur secara translasi

d. *Robot SCARA (Selective Compliance Assembly Robot Arm)*

Robot *Assembly* bisa didesain menurut koordinat kartesian, silindris maupun spheris. Pada beberapa aplikasi hanya membutuhkan sumbu gerak vertikal, misalnya robot *assembly* yang memasang komponen pada PCB. Robot ini mempunyai lengan dengan dua artikulasi, sedangkan *wrist* mempunyai gerakan *linier* dan *rolling*.

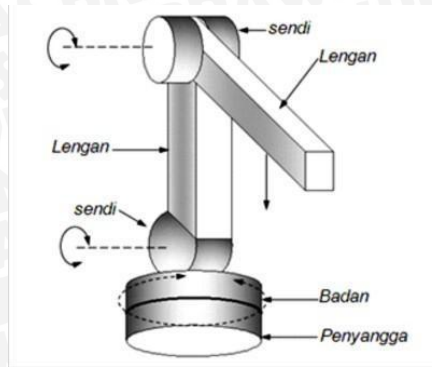


Gambar 2.5. Struktur robot SCARA Smb

Sumber: Endra Pituwarno, (2006)

e. *Robot Artikulasi / Konfigurasi Sendi Lengan*

Robot ini terdiri dari tiga lengan yang dihubungkan dengan dua *Revolute Joint*. *Elbow Joint* menghubungkan *Force Arm* dengan *Upper Arm*. *Shoulder Joint* menghubungkan *Upper Arm* dengan *Base*.

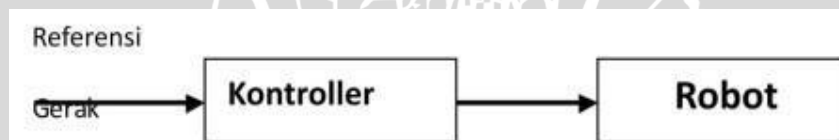


Gambar 2.6. Konfigurasi Sendi-lengan

Sumber: *Endra Pituwarno, (2006)*

Konfigurasi ini yang paling populer untuk melaksanakan fungsi layaknya pekerja pabrik seperti mengangkat barang, mengelas, memasang komponen mur, baut, dan sebagainya. Struktur lengan-sendi cocok digunakan untuk menjangkau daerah kerja yang sempit dengan sudut jangkauan yang beragam.

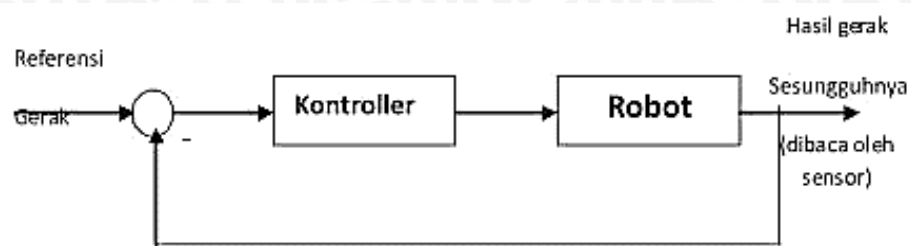
Sistem kontrol robotik pada dasarnya terbagi menjadi dua kelompok, yaitu sistem *control loop* terbuka (*open loop*) dan *loop* tertutup (*close loop*). Diagram *loop* terbuka atau umpan maju (*feed forward control*) dapat dinyatakan dalam gambar berikut ini.



Gambar 2.7. Kontrol Robot *Loop* Terbuka

Sumber :*Endra Pituwarno, (2006)*

Kontrol *loop* terbuka atau umpan maju (*feed forward control*) dapat dinyatakan sebagai sistem kontrol yang outputnya tidak diperhitungkan ulang oleh kontroller. Keadaan apakah robot telah benar-benar mencapai target seperti yang dikehendaki sesuai referensi, adalah tidak mempengaruhi kerja kontroller. Sedangkan kontrol robot *loop* tertutup dapat dinyatakan seperti gambar di bawah:



Gambar 2.8. Kontrol Robot *Loop* Tertutup

Sumber :Endra Pituwarno, (2006)

Pada gambar di atas, jika hasil gerak aktual telah sama dengan referensi maka input kontroler akan sama dengan nol. Artinya kontroler tidak lagi memberikan sinyal aktuasi kepada robot karena target akhir perintah gerak telah diperoleh. Makin kecil error terhitung maka makin kecil pula sinyal pengemudian kontroler terhadap robot, sampai pada akhirnya mencapai kondisi tenang (*steady state*).

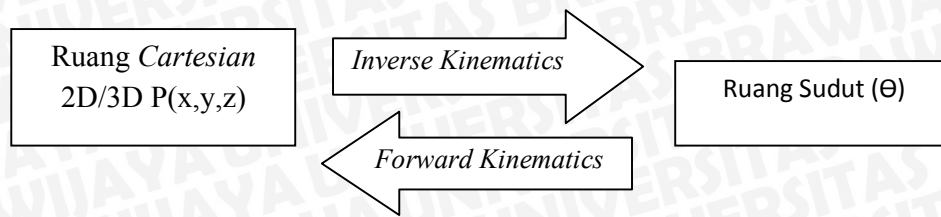
## 2.3 Analisis Kinematika Robot Manipulator

### 2.3.1 Kinematika dan Dinamika Robot Manipulator

Robot manipulator dapat dianalisis dalam dua kajian yaitu analisis kinematika dan dinamika. Analisis kinematika berkaitan dengan pemodelan gerakan robot tanpa menggunakan efek inersial/kelembaman yang terjadi ketika robot melakukan gerakan, sedangkan analisis dinamika berhubungan dengan efek inersia dari struktur robot secara fisik hasil dari gerakan yang ditimbulkan oleh torsi aktuator ketika robot sedang melakukan pergerakan.

Kontrol dinamika tidak bisa bekerja sendirian dalam kontrol robotik tanpa bantuan kontrol kinematika. Sedangkan kontrol kinematika dapat diterapkan langsung tanpa memasukkan unsur kontrol dinamika. Pada dasarnya pengertian umum tentang kontrol robot adalah bagaimana cara mengontrol gerakan suatu robot. Ketika berbicara masalah kontrol gerak, maka yang lebih utama adalah bagaimana membuat kontrol kinematik yang tepat yang mampu menyelesaikan permasalahan-permasalahan posisi yang diinginkan dari pada mendahulukan kontrol dinamikanya. Sebab pemodelan dinamika biasanya lebih rumit dan jika hasilnya tidak tepat maka fungsi pemodelan kinematikanya menjadi tidak akurat dan bahkan tidak berguna dalam memperbaiki kualitas kontrol keseluruhan.

### 2.3.2 Konsep Kinematika



Gambar2.9 Model Kontrol Kinematik Robot

Sumber :Endra Pituwarno, (2006)

Kontroler dinyatakan sebagai kontroler kinematika karena mengandung komponen transformasi ruang kartesian ke ruang sudut sendi maupun sebaliknya.

#### 1. *Forward Kinematics*

*Forward Kinematics* adalah metode untuk menentukan orientasi dan posisi *end-effector* dari besarnya sudut sendi dan panjang link dengan robot. Persamaan *Forward Kinematics* didapatkan berdasarkan jumlah *DOF* dan jenis *kinematic chain* robot manipulator.

#### 2. *Inverse Kinematics*

*Inverse kinematics* merupakan kebalikan dari *Forward Kinematics*. Metode ini diperlukan untuk mengetahui nilai sudut pada sendi-sendi yang diperlukan agar *end-effector* dapat mencapai posisi yang dikehendaki.

*Inverse Kinematics* lebih banyak diaplikasikan, namun perhitungannya jauh lebih rumit dikarenakan beberapa hal diantaranya :

1. Melibatkan persamaan non-linier.
2. Solusi yang dihasilkan lebih banyak, dan kadang tidak terhingga.
3. Kemungkinan tidak mendapat solusi pun ada, jika posisi berada di luar *workspace* atau di luar *configuration space*.

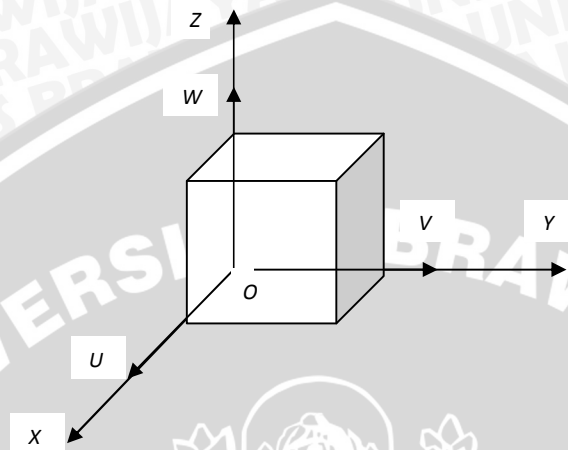
- Penggunaan matriks rotasi dan translasi

Dalam analisa dalam daerah kerja 2D dapat menggunakan persamaan trigonometri biasa. Namun tidak demikian dengan halnya dengan ruang 3D. penggunaan persamaan trigonometri kurang dapat memberikan solusi yang tepat untuk menyatakan pergerakan dari titik ke titik dalam ruang 3D.



### 1. Matriks Rotasi

Matriks rotasi adalah suatu matriks (3x3) yang didefinisikan sebagai matriks transformasi yang beroperasi pada suatu vector posisi di ruang 3D untuk memetakan sistem koordinat OUVW (*body-attached frame*) vector ke dalam sistem koordinat referensi (OXYZ).



Gambar 2.10 Sistem OUVW vs OXYZ

Sumber :Endra Pituwarno, (2006)

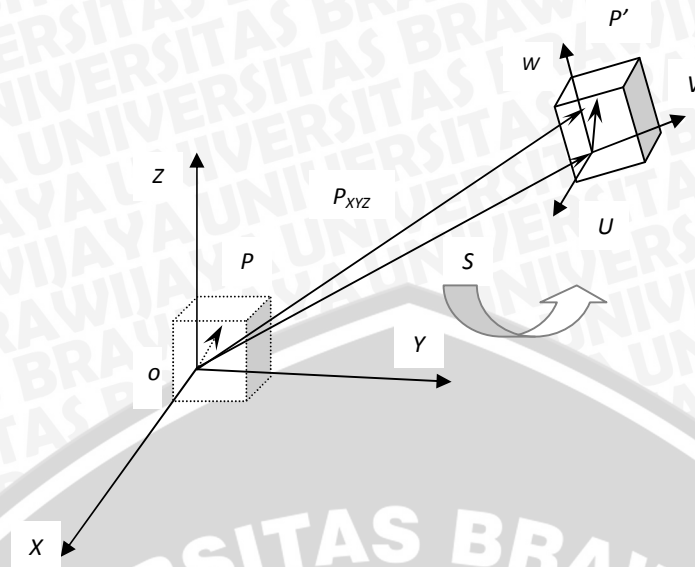
Perpindahan koordinat  $p$  pada sumbu  $O$  secara rotasi dapat dinyatakan sebagai,

$$\mathbf{P}_{xyz} = \mathbf{R} \mathbf{P}_{uvw} \quad (\text{Endra, 2006})$$

$\mathbf{R}$  disebut matriks rotasi.

### 2. Matriks Translasi

Transformasi koordinat dalam ruang 3D pada dasarnya terdiri dari dua bagian yang dapat beroperasi secara bersamaan. Misalnya bentuk kubus mengalami perpindahan tempat secara translasi sekaligus rotasi, seperti pada gambar dibawah,



Gambar 2.11 Sistem  $OUVW$  vs  $OXYZ$

Sumber :Endra Pituwarno, (2006)

Titik  $p'$  dapat dinyatakan dalam  $p_{xyz}$  maupun  $p_{uvw}$ , karena  $p$  melakukan gerakan translasi sekaligus rotasi maka dapat ditulis,

$$\begin{bmatrix} P_{xyz} \\ 1 \end{bmatrix} = \begin{bmatrix} R & s \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{uvw} \\ 1 \end{bmatrix} \quad (\text{Endra, 2006})$$

Dimana  $R$  merupakan matriks rotasi dan  $s$  adalah matriks translasi.

### 2.3.3 Sistem Kendali Lintasan Gerak Robot Manipulator

*Forward* dan *inverse kinematics* merupakan solusi dari control kinematika robot manipulator, yang hanya memperhatikan aspek geometri dari robot. Solusi ini tidak memperhatikan batasan-batasan lain yang diberlakukan oleh *workspace* di mana robot beroperasi. Secara khusus, *Forward* dan *inverse kinematics* tidak memperhitungkan kemungkinan akan tabrakan antara robot manipulator dengan benda di sekitar *workspace*. Oleh karena itu dibutuhkan sebuah sistem perencanaan lintasan gerak robot (*path planning*), sehingga robot akan mampu bergerak pada titik-titik lintasan (*path points*) yang meliputi titik awal, titik-titik singgah hingga titik tujuan.

Ada berbagai cara dalam melakukan *path planning*, dimana semuanya memiliki tujuan yang sama yaitu memberikan serangkaian titik yang disebut *via points* pada sepanjang lintasan. Cara yang paling sederhana dalam melakukan *path planning* adalah

dengan memberikan serangkaian urutan posisi *end-effector*. Pada cara ini *inverse kinematics* dibutuhkan dalam mengubah posisi *end-effector* menjadi konfigurasi tiap sendi.

Agar robot dapat bergerak dengan tepat ke setiap *path points* hasil *path planning* tadi, maka diperlukan perencanaan gerak. Perencanaan gerak ini dilakukan dengan membuat fungsi posisi berdasarkan waktu. Fungsi waktu tersebut biasa dikenal dengan sebutan *trajectory*. Karena *trajectory* merupakan fungsi waktu, maka dapat dihitung juga besar kecepatan dan percepatan robot sepanjang jalur lintasan. Perencanaan *trajectory* atau *trajectory planning* dapat dilakukan dengan *joint space* ataupun *cartesian space*.

### 2.3.4 Sistem Kendali Aktuator

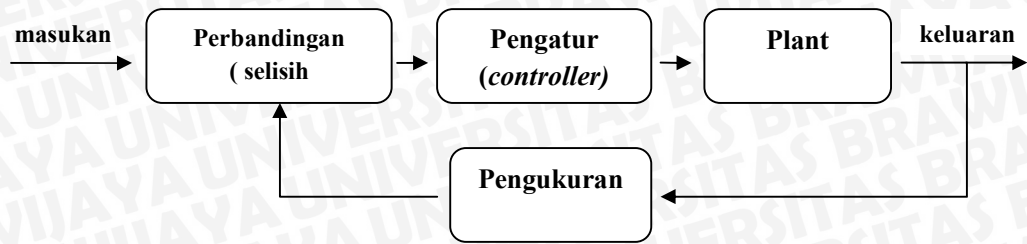
Sistem kendali dapat dikatakan sebagai hubungan antara komponen yang membentuk sebuah konfigurasi sistem, yang akan menghasilkan respon dari sistem yang dihasilkan. Pada sistem kendali dikenal sistem *loop* terbuka (*open loop system*) dan sistem *loop* tertutup (*closed loop system*). Sistem kendali *open loop* umumnya menggunakan pengendali (*controller*) serta aktuator kendali (*control actuator*) yang berguna untuk memperoleh respon sistem yang baik. Sistem kendali ini keluarannya tidak diperhitungkan ulang oleh *controller*. Suatu keadaan apakah suatu perencanaan benar-benar telah mencapai target seperti yang dikehendaki masukan atau referensi, tidak dapat mempengaruhi kinerja kontroler.



Gambar 2.12 Blok Diagram Sistem Kendali *Open Loop*

Sumber :Hardiansyah R.N, 2010

Berbeda dengan sistem kendali *open loop*, pada sistem kendali *closed loop* dimanfaatkan variable yang sebanding dengan selisih respon yang terjadi terhadap respon yang diinginkan. Sistem seperti ini juga sering dikenal dengan sistem kendali umpan balik (*feedback system*).



Gambar 2.13 Blok Diagram Sistem Kendali *Closed Loop*

Sumber : Hardiansyah R.N, 2010

## 2.4 Aktuator Robot Manipulator

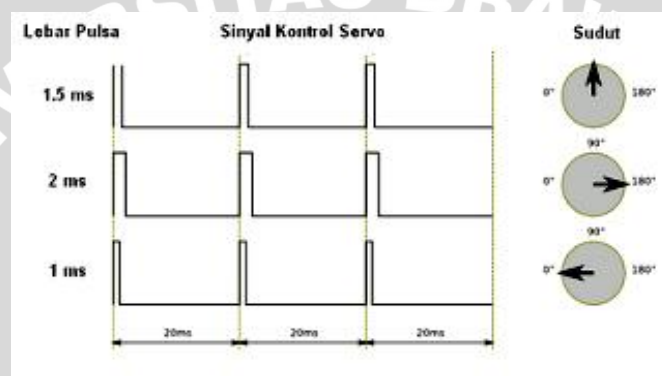
### 2.4.1 Motor DC

Motor DC *Brush* merupakan salah satu jenis aktuator yang paling banyak digunakan dalam industri ataupun sistem robot manipulator. Motor ini menggunakan prinsip elektromagnetika untuk menghasilkan kerja yaitu putaran. Motor DC terdiri dari rotor yang berputar dan bagian magnet sebagai stator (bagian yang diam). Arus yang datang melalui sikat/*brush* dan medan magnet stator akan menyebabkan rotor berputar. Bagian magnet pada stator biasa menggunakan magnet permanen. Dengan demikian motor menerima sumber arus searah yang kemudian diubah menjadi energy mekanik berupa putaran, yang nantinya dipakai pada peralatan lain. Arah arus yang datang menuju rotor berpengaruh pada arah putar motor DC.

Motor DC sering diaplikasikan terutama pada sistem robot yang memerlukan torsi cukup tinggi karena masih bisa menambahkan reduksi gear untuk meningkatkan torsi. Selain itu merubah kecepatan motor DC cukup mudah, yaitu dengan menggunakan sinyal PWM (*Pulse Width Modulation*). Pengaturan sinyal PWM (*Pulse Width Modulation*) dapat dilakukan dengan cara mengatur lebar pulsa (waktu ON) dari tegangan sumbernya (tegangan DC). Perbandingan antara waktu ON dan waktu OFF disebut *duty cycle* (siklus kerja). Semakin besar siklus kerja maka akan semakin besar keluaran yang dihasilkan sehingga kecepatan motor akan semakin baik. Pembangkitan sinyal PWM dengan mikrokontroler memiliki beberapa keuntungan, seperti teknik pemrogramannya yang lebih sederhana dan rangkaian listriknya yang lebih sederhana.

## 2.4.2 Motor Servo

Motor RC servo atau sering disingkat motor servo saja, merupakan sebuah motor dengan sistem *closed loop* dimana posisi dari motor akan diinformasikan kembali pada rangkaian kontrol yang ada di dalam motor servo (Iswanto, 2009). Motor servo terdiri dari sebuah motor DC *brush*, reduksi gear untuk meningkatkan torsi, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo, kebanyakan motor servo berputar sekitar 90-180 derajat. Namun beberapa diantaranya ada yang dapat berputar 360 derajat atau lebih. Hanya saja motor ini tidak dapat berputar terus menerus sehingga tidak dapat digunakan untuk mengendalikan roda (kecuali dimodifikasi).



Gambar 2.14 Variasi Lebar Pulsa Terhadap Sudut Servo

Sumber : Endra Pituwarno, (2006)

Motor servo mampu bekerja dua arah, dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya dengan jangkauan gerakan poros outputnya berkisar antara (-120 sampai 120) derajat.

## 2.5 Sensor Posisi Robot Manipulator

### 2.5.1 Limit Switch

*Limit switch* adalah salah satu jenis kontak sensor yaitu sensor yang bekerja dengan sentuhan. Pada penggunaan gripper kadang diperlukan informasi bahwa benda sudah berada dalam genggaman, sehingga akan mengaktifkan / menonaktifkan sensor. Sensor akan mengirimkan sinyal elektrik menuju kontroler dan akan direspon dengan mengirimkan perintah pada robot untuk melakukan kerja tertentu.

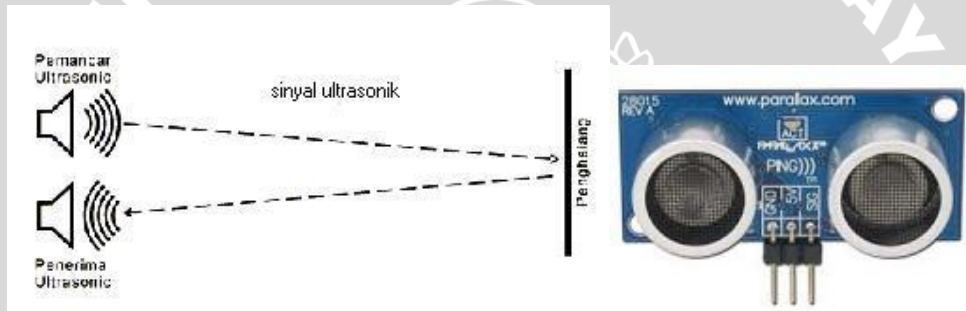


Gambar 2.15 *Limit Switch*

Sumber : Anonymous b

### 2.5.2 Sensor Jarak

Sensor jarak yang digunakan yaitu *HSCR04* dengan media suara (*ultrasonic*) sebagai pengukur jarak. Sensor ultrasonik merupakan sensor yang bekerja dengan cara memancarkan suatu gelombang dan kemudian menghitung waktu pantulan gelombang tersebut.



Gambar 2.16 Sensor Jarak

Sumber : Anonymous c

Gelombang ultrasonik bekerja pada frekuensi mulai 20 kHz hingga sekitar 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan rendah pada fasa gas, cair hingga padat. Secara umum sensor ultrasonik digunakan untuk menghitung jarak dari suatu objek yang berada di depan sensor tersebut. Sehingga dengan fungsinya tersebut, sensor ultrasonik biasa digunakan pada perangkat yang membutuhkan perhitungan jarak. Contoh : smart robot, Kapal laut, kapal selam, dll.

### 2.6 Board Arduino

Board yang digunakan sebagai *microcontroller* yaitu *Arduino Mega2560*. *Arduino Mega2560* adalah sebuah *Boards* yang ditanamkan *Microcontroller* ATmega2560, dengan 54 digital *input/output* pins (yang mana 14 port dapat digunakan

sebagai *output* PWM), 16 analog input, 4 UARTs (*hardware serial ports*), 16 MHz *crystal oscillator*, koneksi USB, sebuah *jack power*, dan sebuah tombol reset.

*Arduino Mega2560* berbeda dari *Board* sebelumnya, karena versi terbaru sudah tidak menggunakan chip driver FTDI USB-to-serial. Tapi, menggunakan chip *ATmega16U2* (*ATmega8U2* pada *Board* Revisi 1 dan Revisi 2) yang diprogram sebagai konverter USB-to-serial. *Arduino Mega2560* Revisi 2 memiliki resistor penarik jalur HWB 8U2 ke Ground, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU.

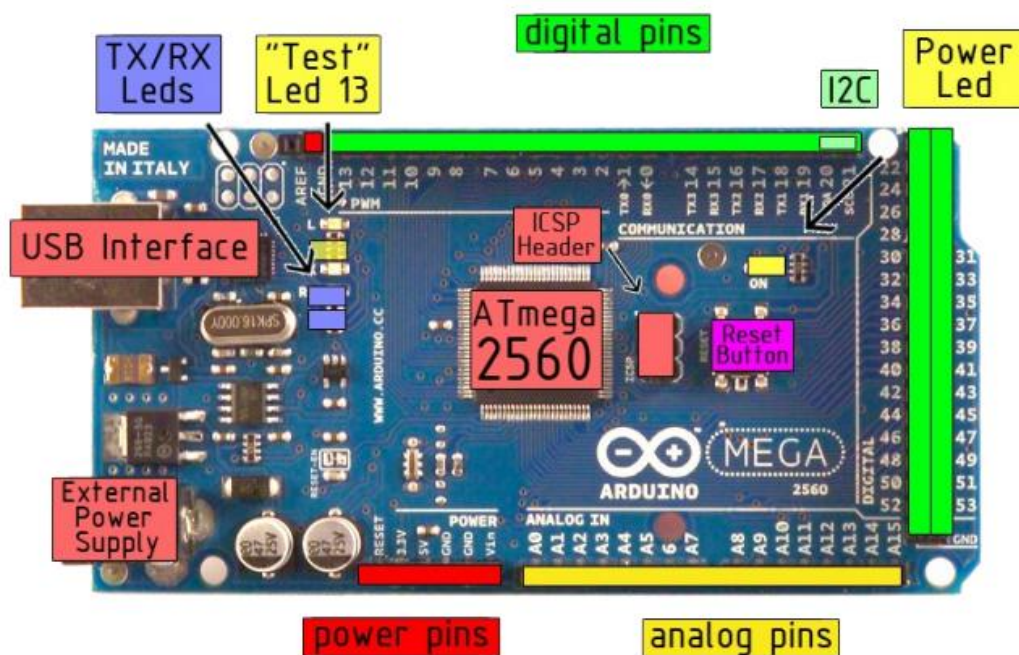
*Arduino Mega2560* Revisi 3 memiliki fitur-fitur baru sebagai berikut:

- Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan shield untuk beradaptasi dengan tegangan yang tersedia pada *Boards*.
- Sirkuit *RESET*.
- Chip *ATmega16U2* menggantikan chip *ATmega8U2*.

#### Spesifikasi *Arduino Mega2560*

|                                    |   |
|------------------------------------|---|
| <i>Microcontroller</i>             | <i>ATmega2560</i>                       |
| <i>Operating Voltage</i>           | 5V                                      |
| <i>Input Voltage (recommended)</i> | 7-12V                                   |
| <i>Input Voltage (limits)</i>      | 6-20V                                   |
| <i>Digital I/O Pins</i>            | 54 (of which provide PWM output)        |
| <i>Analog Input Pins</i>           | 16                                      |
| <i>DC Current for I/O Pin</i>      | 40 mA                                   |
| <i>DC Current for 3.3V Pin</i>     | 50 mA                                   |
| <i>Flash Memory</i>                | 256 KB of which 8 KB used by bootloader |
| <i>SRAM</i>                        | 8 KB                                    |
| <i>EEPROM</i>                      | 4 KB                                    |
| <i>Clock Speed</i>                 | 16 MHz                                  |

(Sumber : *Datasheet Arduino Mega2560*)



Gambar 2.17 Board Arduino Mega2560.

Sumber : Datasheet Arduino Mega2560.

Tabel 2.2 Pemetaan Pin ATmega2560 pada Arduino Mega2560.

| Nomor Pin | Nama Pin             | Peta Nama Pin        |
|-----------|----------------------|----------------------|
| 1         | PG5 (OC0B)           | Digital pin 4 (PWM)  |
| 2         | PE0 (RXD0/PCINT8)    | Digital pin 0 (RX0)  |
| 3         | PE1 (TXD0)           | Digital pin 1 (TX0)  |
| 4         | PE2 (XCK0/AIN0)      |                      |
| 5         | PE3 (OC3A/AIN1)      | Digital pin 5 (PWM)  |
| 6         | PE4 (OC3B/INT4)      | Digital pin 2 (PWM)  |
| 7         | PE5 (OC3C/INT5)      | Digital pin 3 (PWM)  |
| 8         | PE6 (T3/INT6)        |                      |
| 9         | PE7 (CLKO/ICP3/INT7) |                      |
| 10        | VCC                  | VCC                  |
| 11        | GND                  | GND                  |
| 12        | PH0 (RXD2)           | Digital pin 17 (RX2) |
| 13        | PH1 (TXD2)           | Digital pin 16 (TX2) |
| 14        | PH2 (XCK2)           |                      |



|    |                        |                       |
|----|------------------------|-----------------------|
| 15 | PH3 (OC4A)             | Digital pin 6 (PWM)   |
| 16 | PH4 (OC4B)             | Digital pin 7 (PWM)   |
| 17 | PH5 (OC4C)             | Digital pin 8 (PWM)   |
| 18 | PH6 (OC2B)             | Digital pin 9 (PWM)   |
| 19 | PB0 (SS/PCINT0)        | Digital pin 53 (SS)   |
| 20 | PB1 (SCK/PCINT1)       | Digital pin 52 (SCK)  |
| 21 | PB2 (MOSI/PCINT2)      | Digital pin 51 (MOSI) |
| 22 | PB3 (MISO/PCINT3)      | Digital pin 50 (MISO) |
| 23 | PB4 (OC2A/PCINT4)      | Digital pin 10 (PWM)  |
| 24 | PB5 (OC1A/PCINT5)      | Digital pin 11 (PWM)  |
| 25 | PB6 (OC1B/PCINT6)      | Digital pin 12 (PWM)  |
| 26 | PB7 (OC0A/OC1C/PCINT7) | Digital pin 13 (PWM)  |
| 27 | PH7 (T4 )              |                       |
| 28 | PG3 (TOSC2)            |                       |
| 29 | PG4 (TOSC1)            |                       |
| 30 | RESET                  | RESET                 |
| 31 | VCC                    | VCC                   |
| 32 | GND                    | GND                   |
| 33 | XTAL2                  | XTAL2                 |
| 34 | XTAL1                  | XTAL1                 |
| 35 | PL0 (ICP4)             | Digital pin 49        |
| 36 | PL1 (ICP5)             | Digital pin 48        |
| 37 | PL2 (T5 )              | Digital pin 47        |
| 38 | PL3 (OC5A)             | Digital pin 46 (PWM)  |
| 39 | PL4 (OC5B)             | Digital pin 45 (PWM)  |
| 40 | PL5 (OC5C)             | Digital pin 44 (PWM)  |
| 41 | PL6                    | Digital pin 43        |
| 42 | PL7                    | Digital pin 42        |
| 43 | PD0 (SCL/INT0)         | Digital pin 21 (SCL)  |
| 44 | PD1 (SDA/INT1)         | Digital pin 20 (SDA)  |
| 45 | PD2 (RXDI/INT2)        | Digital pin 19 (RX1)  |
| 46 | PD3 (TXD1/INT3)        | Digital pin 18 (TX1)  |
| 47 | PD4 (ICP1)             |                       |

|    |                    |                      |
|----|--------------------|----------------------|
| 48 | PD5 (XCK1)         |                      |
| 49 | PD6 (T1)           |                      |
| 50 | PD7 (T0)           | Digital pin 38       |
| 51 | PG0 (WR)           | Digital pin 41       |
| 52 | PG1 (RD)           | Digital pin 40       |
| 53 | PC0 (A8)           | Digital pin 37       |
| 54 | PC1 (A9)           | Digital pin 36       |
| 55 | PC2 (A10)          | Digital pin 35       |
| 56 | PC3 (A11)          | Digital pin 34       |
| 57 | PC4 (A12)          | Digital pin 33       |
| 58 | PC5 (A13)          | Digital pin 32       |
| 59 | PC6 (A14)          | Digital pin 31       |
| 60 | PC7 (A15)          | Digital pin 30       |
| 61 | VCC                | VCC                  |
| 62 | GND                | GND                  |
| 63 | PJ0 (RXD3/PCINT9)  | Digital pin 15 (RX3) |
| 64 | PJ1 (TXD3/PCINT10) | Digital pin 14 (TX3) |
| 65 | PJ2 (XCK3/PCINT11) |                      |
| 66 | PJ3 (PCINT12)      |                      |
| 67 | PJ4 (PCINT13)      |                      |
| 68 | PJ5 (PCINT14)      |                      |
| 69 | PJ6 (PCINT 15)     |                      |
| 70 | PG2 (ALE)          | Digital pin 39       |
| 71 | PA7 (AD7)          | Digital pin 29       |
| 72 | PA6 (AD6)          | Digital pin 28       |
| 73 | PA5 (AD5)          | Digital pin 27       |
| 74 | PA4 (AD4)          | Digital pin 26       |
| 75 | PA3 (AD3)          | Digital pin 25       |
| 76 | PA2 (AD2)          | Digital pin 24       |
| 77 | PA1 (AD1)          | Digital pin 23       |
| 78 | PA0 (AD0)          | Digital pin 22       |
| 79 | PJ7                |                      |
| 80 | VCC                | VCC                  |

|     |                     |                  |
|-----|---------------------|------------------|
| 81  | GND                 | GND              |
| 82  | PK7 (ADC15/PCINT23) | Analog pin 15    |
| 83  | PK6 (ADC14/PCINT22) | Analog pin 14    |
| 84  | PK5 (ADC13/PCINT21) | Analog pin 13    |
| 85  | PK4 (ADC12/PCINT20) | Analog pin 12    |
| 86  | PK3 (ADC11/PCINT19) | Analog pin 11    |
| 87  | PK2 (ADC10/PCINT18) | Analog pin 10    |
| 88  | PK1 (ADC9/PCINT17)  | Analog pin 9     |
| 89  | PK0 (ADC8/PCINT16)  | Analog pin 8     |
| 90  | PF7 (ADC7)          | Analog pin 7     |
| 91  | PF6 (ADC6)          | Analog pin 6     |
| 92  | PF5 (ADC5/TMS)      | Analog pin 5     |
| 93  | PF4 (ADC4/TMK)      | Analog pin 4     |
| 94  | PF3 (ADC3)          | Analog pin 3     |
| 95  | PF2 (ADC2)          | Analog pin 2     |
| 96  | PF1 (ADC1)          | Analog pin 1     |
| 97  | PF0 (ADC0)          | Analog pin 0     |
| 98  | AREF                | Analog Reference |
| 99  | GND                 | GND              |
| 100 | AVCC                | VCC              |

Sumber : *Datasheet Arduino Mega2560*

### Sumber Daya

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm, yang bagian tengahnya terminal positif ke *jack* sumber tegangan pada *Board*. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin Gnd dan pin Vin dari konektor *POWER*.

*Board* Arduino ATmega2560 dapat beroperasi dengan catu daya eksternal 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat *Board* menjadi

tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak *Board*. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt.

Pin tegangan yang tersedia pada *Board* Arduino adalah sebagai berikut :

- **VIN** : Adalah input tegangan untuk *Board* Arduino ketika menggunakan sumber daya eksternal. Tegangan dapat diberikan melalui pin Vin, atau jika memasok tegangan untuk *Board* melalui *jack power*, kita bisa mengakses tegangan melalui pin Vin.
- **5V** : Sebuah pin yang mengeluarkan tegangan hasil dari regulator 5 Volt, dari pin ini tegangan sudah diatur regulator yang tersedia (*built-in*) pada *Board*. Arduino dapat diaktifkan dengan sumber daya baik berasal dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada *Board* (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator dapat merusak *Board* Arduino.
- **3.3V** : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada *Board* (*on-Board*). Arus maksimum yang dihasilkan adalah 50 mA.
- **GND** : Pin Ground.
- **IOREF** : Pin ini pada *Board* Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah *shield* dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan *voltage translator* pada output untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

### Memori

Arduino ATmega2560 memiliki 256 KB flash memory untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan *library* EEPROM).

### Input Output

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()` , `digitalWrite()` , dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau

menerima arus maksimum 40 mA dan memiliki resistor pull-up internal ( yang terputus secara *default* ) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- **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 (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.
- **Eksternal Interupsi** : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat, menurun, atau berubah nilai.
- **SPI** : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan *library* SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.
- **LED** : Pin 13. Tersedia secara *built-in* pada *Board* Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED mati (OFF).
- **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan *library* Wire.

*Arduino Mega2560* memiliki 16 pin sebagai analog input, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur atau diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi `analogReference()`. Ada beberapa pin lainnya yang tersedia, antara lain:

- **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
- **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada *shield* yang menghalangi *Board* utama Arduino.

## Komunikasi

*Arduino Mega2560* memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan *Arduino* lain, atau dengan mikrokontroler lainnya. *Arduino ATmega328* menyediakan 4 hardware komunikasi serial UART TTL (5 Volt). Sebuah chip *ATmega16U2* (*ATmega8U2* pada *Board* Revisi 1 dan Revisi 2) yang terdapat pada *Board* digunakan sebagai media komunikasi serial melalui USB dan sebagai *COM Port Virtual* (pada *Device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi *Windows* masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali *Board* sebagai *port* COM secara otomatis. Perangkat lunak *Arduino* termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke, dan, dari *Board* *Arduino*. LED RX dan TX yang tersedia pada *Board* akan berkedip ketika data sedang dikirim atau diterima melalui chip USB ke serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

Sebuah *library* *SoftwareSerial* memungkinkan untuk komunikasi serial pada salah satu pin digital *Mega2560*. *ATmega2560* juga mendukung komunikasi TWI dan SPI. Perangkat lunak *Arduino* termasuk *library* *Wire* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan *library* SPI.

## Pemrograman

*Arduino Mega* dapat diprogram dengan software *Arduino*. *ATmega2560* pada *Arduino Mega* sudah tersedia *preburned* dengan *bootloader* yang memungkinkan untuk meng-upload kode baru tanpa menggunakan *programmer hardware* eksternal. Hal ini karena komunikasi yang terjadi menggunakan protokol asli STK500. Juga dapat melewati *bootloader* dan program mikrokontroler melalui pin *header* ICSP (*In-Circuit Serial Programming*).

## Reset (software) Otomatis

Daripada menekan tombol *reset* sebelum *upload*, *Arduino Mega2560* didesain dengan cara yang memungkinkan untuk me-reset melalui perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu jalur kontrol hardware (DTR) mengalir dari *ATmega8U2/16U2* dan terhubung ke jalur *reset* dari *ATmega2560* melalui kapasitor

100 *nanofarad*. Bila jalur ini di-set rendah/low, jalur *reset drop* cukup lama untuk me-reset chip. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan meng-*upload* kode dengan hanya menekan tombol *upload* pada perangkat lunak Arduino. Ini berarti bahwa *bootloader* memiliki rentang waktu yang lebih pendek, seperti menurunkan DTR dapat terkoordinasi (berjalan beriringan) dengan dimulainya *upload*.

Pengaturan ini juga memiliki implikasi lain. Ketika *Mega2560* terhubung dengan komputer yang menggunakan sistem operasi *Mac OS X* atau *Linux*, *Board Arduino* akan di-*reset* setiap kali dihubungkan dengan software komputer (melalui USB). Dan setengah detik kemudian atau lebih, *bootloader* berjalan pada *Board Mega2560*. Proses *reset* melalui program ini digunakan untuk mengabaikan data yang cacat (yaitu apapun selain meng-*upload* kode baru), kemudian akan memotong dan membuang beberapa byte pertama dari data yang dikirim ke *Board* setelah sambungan dibuka. Jika sebuah sketsa dijalankan pada *Board* untuk menerima satu kali konfigurasi atau menerima data lain ketika pertama kali dijalankan, pastikan bahwa perangkat lunak diberikan waktu untuk berkomunikasi dengan menunggu satu detik setelah terkoneksi dan sebelum mengirim data.

*Mega2560* memiliki trek jalur yang dapat dipotong untuk menonaktifkan fungsi *auto-reset*. Pad di kedua sisi jalur dapat dihubungkan dengan disolder untuk mengaktifkan kembali fungsi *auto-reset*. Pada berlabel “*RESET-EN*”, juga dapat menonaktifkan *auto-reset* dengan menghubungkan resistor 110 ohm dari 5V ke jalur *reset*.

### **Perlindungan Beban Berlebih pada USB**

*Arduino Mega2560* memiliki *polyfuse reset* yang melindungi port USB komputer dari hubungan singkat dan arus berlebih. Meskipun pada dasarnya komputer telah memiliki perlindungan internal pada port USB mereka sendiri, sekering memberikan lapisan perlindungan tambahan. Jika arus lebih dari 500 mA dihubungkan ke port USB, sekering secara otomatis akan memutuskan sambungan sampai hubungan singkat atau overload dihapus/dibuang.

### **Karakteristik Fisik dan Kompatibilitas Shield**

Maksimum panjang dan lebar PCB *Mega2560* adalah 4 x 2.1 inch (10,16 x 5,3 cm), dengan konektor USB dan jack power menonjol melampaui batas dimensi. Empat

lubang sekrup memungkinkan *Board* terpasang pada suatu permukaan atau wadah. *Arduino Mega2560* dirancang agar kompatibel dengan sebagian *shield* yang dirancang untuk *Arduino Uno*, *Arduino Diecimila* atau *Arduino Duemilanove*. Pin Digital 0-13 (pin AREF berdekatan dan pin GND), input analog 0 sampai 5, *header power*, dan *header ICSP* berada di lokasi yang ekuivalen. Selanjutnya UART utama (port serial) terletak di pin yang sama (0 dan 1), seperti pin interupsi eksternal 0 dan 1 (masing-masing pada pin 2 dan 3). SPI di kedua *header ICSP* yaitu *Mega2560* dan *Duemilanove/Diecimila*.

(Sumber : *Datasheet Arduino Mega2560*)

