

**KINEMATIKA DAN ANTARMUKA ROBOT SCARA BERBASIS  
PROCESSING IDE**

**LAPORAN KERJA PRAKTIK**



**Disusun oleh:**  
**IVAN SYAHRONI HERMAWAN**  
**17/415746/SV/13611**

**PROGRAM STUDI TEKNOLOGI LISTRIK  
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA  
SEKOLAH VOKASI UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2019**

# **HALAMAN PENGESAHAN**

## **KINEMATIKA DAN ANTARMUKA ROBOT SCARA BERBASIS PROCESSING IDE**

### **LAPORAN KERJA PRAKTIK**

Diajukan Sebagai Salah Satu Syarat Kelulusan Program Kerja Praktik Laboratorium  
Instrumentasi dan Kendali  
Pada Program Studi Teknologi Listrik Departemen Teknik Elektro dan Informatika  
Sekolah Vokasi Universitas Gadjah Mada

**Disusun oleh:**

**IVAN SYAHRONI HERMAWAN**  
**17/415746/SV/13611**

Telah disetujui dan disahkan  
pada tanggal 2 Agustus 2019

Ketua Program Studi Teknologi Instrumentasi

Dosen Pembimbing

**Ma'un Budiyanto, S.T., M.T.**  
**NIP 197007071999031002**

**Fahmizal, S.T., M.Sc.**  
**NIP 111198807201609101**

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena hanya dengan rahmat dan hidayah-Nya, laporan kerja praktik ini dapat diselesaikan tanpa halangan yang berarti. Keberhasilan dalam menyusun laporan kerja praktik ini tidak lepas dari bantuan berbagai pihak yang mana dengan tulus dan ikhlas memberikan masukan guna sempurnanya laporan kerja praktik ini. Oleh karena itu dalam kesempatan ini, dengan kerendahan hati penulis mengucapkan terimakasih kepada:

1. Bapak Ma'un Budiyanto, S.T., M.T. selaku Ketua Program Studi Teknologi Listrik Universitas Gadjah Mada,
2. Bapak Fahmizal, S.T., M.Sc selaku dosen pembimbing pertama yang telah memberikan banyak bantuan, bimbingan, serta arahan dalam kerja praktik,
3. Seluruh Dosen di Teknologi Listrik Sekolah Vokasi Universitas Gadjah Mada, yang tidak bisa disebutkan satu-satu, atas ilmu dan bimbingannya,
4. Ibu dan Bapak yang selama ini telah sabar membimbing, mengarahkan, dan mendoakan penulis tanpa kenal lelah untuk selama-lamanya, dan

Penulis menyadari bahwa penyusunan laporan kerja praktik ini jauh dari sempurna. Kritik dan saran dapat ditujukan langsung pada e-mail saya. Akhir kata penulis mohon maaf yang sebesar-besarnya apabila terdapat kekeliruan di dalam penulisan kerja praktik ini.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 2 Agustus 2019

**Penulis**

## DAFTAR ISI

|  |             |
|--|-------------|
| <b>HALAMAN PENGESAHAN</b>                            | <b>iii</b>  |
| <b>KATA PENGANTAR</b>                                | <b>iii</b>  |
| <b>DAFTAR ISI</b>                                    | <b>v</b>    |
| <b>DAFTAR TABEL</b>                                  | <b>vi</b>   |
| <b>DAFTAR GAMBAR</b>                                 | <b>vii</b>  |
| <b>Intisari</b>                                      | <b>viii</b> |
| <i>Abstract</i>                                      | <b>ix</b>   |
| <b>I PENDAHULUAN</b>                                 | <b>1</b>    |
| 1.1 Latar Belakang Masalah . . . . .                 | 1           |
| 1.2 Tujuan Penelitian . . . . .                      | 2           |
| 1.2.1 Secara Umum . . . . .                          | 2           |
| 1.2.2 Tujuan Khusus . . . . .                        | 2           |
| 1.3 Batasan Penelitian . . . . .                     | 2           |
| 1.4 Metode Kerja Praktik . . . . .                   | 3           |
| 1.5 Sistematika Penulisan . . . . .                  | 5           |
| <b>II LANDASAN TEORI</b>                             | <b>6</b>    |
| 2.1 Gambaran Umum Robot Lengan . . . . .             | 6           |
| 2.1.1 <i>Degress of Freedom</i> . . . . .            | 7           |
| 2.1.2 Konfigurasi Robot Lengan . . . . .             | 7           |
| 2.1.3 <i>Wrist</i> dan <i>End-effector</i> . . . . . | 10          |
| 2.2 Kinematika . . . . .                             | 11          |
| 2.2.1 Kinematika Maju . . . . .                      | 12          |
| 2.2.2 Kinematika Balik . . . . .                     | 13          |
| 2.3 Motor DC . . . . .                               | 13          |
| 2.4 Regulator . . . . .                              | 15          |
| 2.5 Arduino . . . . .                                | 15          |
| 2.6 Driver Motor H . . . . .                         | 17          |

|                               |   |           |
|-------------------------------|---|-----------|
| 2.7                           | Processing Integrated Development Environment (IDE) . . . . . | 18        |
| 2.7.1                         | Syntax dalam Processing IDE . . . . .                         | 18        |
| 2.7.2                         | <i>Library</i> untuk Processing IDE . . . . .                 | 22        |
| <b>III PERANCANGAN SISTEM</b> |   | <b>24</b> |
| 3.1                           | Diagram Blok Sistem . . . . .                                 | 24        |
| 3.2                           | Perancangan Perangkat Keras . . . . .                         | 25        |
| 3.2.1                         | Sistem Mekanik . . . . .                                      | 25        |
| <b>IV PENUTUP</b>             |   | <b>31</b> |
| <b>DAFTAR PUSTAKA</b>         |   | <b>32</b> |

## DAFTAR TABEL

|           |   |    |
|-----------|---|----|
| Tabel 2.1 | Spesifikasi Arduino Mega 2560 . . . . .         | 16 |
| Tabel 2.2 | Pin pada Driver EMS 30A H-Bridge . . . . .      | 17 |
| Tabel 2.3 | <i>Syntax</i> Transformasi . . . . .            | 19 |
| Tabel 2.4 | <i>Syntax</i> Shape . . . . .                   | 20 |
| Tabel 2.5 | <i>Syntax</i> Koordinat Mouse . . . . .         | 21 |
| Tabel 2.6 | Fungsi <i>Tools</i> ControlP5 . . . . .         | 23 |
| Tabel 3.1 | Spesifikasi Robot SCARA . . . . .               | 27 |
| Tabel 3.2 | Spesifikasi Motor DC pada Robot SCARA . . . . . | 28 |

## DAFTAR GAMBAR

|             |  |    |
|-------------|--|----|
| Gambar 1.1  | Flowchart Prosedur Penelitian . . . . .                | 4  |
| Gambar 2.1  | Jenis-Jenis <i>Joint</i> . . . . .                     | 6  |
| Gambar 2.2  | Struktur dari Konfigurasi <i>Articulated</i> . . . . . | 8  |
| Gambar 2.3  | Struktur dari Konfigurasi <i>Spherical</i> . . . . .   | 8  |
| Gambar 2.4  | Struktur dari Konfigurasi SCARA . . . . .              | 9  |
| Gambar 2.5  | Struktur dari Konfigurasi <i>Cylindrical</i> . . . . . | 9  |
| Gambar 2.6  | Struktur dari Konfigurasi <i>Cartesian</i> . . . . .   | 10 |
| Gambar 2.7  | Struktur dari <i>Joint Spherical Wrist</i> . . . . .   | 11 |
| Gambar 2.8  | Struktur <i>End-Effector</i> . . . . .                 | 11 |
| Gambar 2.9  | Blok Diagram Kinematika . . . . .                      | 12 |
| Gambar 2.10 | Kinematika Maju . . . . .                              | 12 |
| Gambar 2.11 | Kinematika Balik . . . . .                             | 13 |
| Gambar 2.12 | Jenis-Jenis <i>Joint</i> . . . . .                     | 14 |
| Gambar 2.13 | Prinsip Kerja Motor DC . . . . .                       | 14 |
| Gambar 2.14 | Regulator <i>Buck</i> LM2596 . . . . .                 | 15 |
| Gambar 2.15 | Arduino Mega 2560 . . . . .                            | 16 |
| Gambar 2.16 | Driver Motor EMS 30A H-Bridge . . . . .                | 17 |
| Gambar 2.17 | Processing IDE . . . . .                               | 18 |
| Gambar 2.18 | Toolbar Sketch . . . . .                               | 21 |
| Gambar 2.19 | ControlP5 . . . . .                                    | 22 |
| Gambar 3.1  | Diagram Blok Sistem . . . . .                          | 24 |
| Gambar 3.2  | <i>Free Body</i> Robot SCARA . . . . .                 | 26 |
| Gambar 3.3  | Bentuk Fisik Robot SCARA . . . . .                     | 26 |
| Gambar 3.4  | Box Panel Robot SCARA . . . . .                        | 27 |
| Gambar 3.5  | Motor DC dengan Potensiometer . . . . .                | 28 |
| Gambar 3.6  | End-Effector Robot SCARA . . . . .                     | 29 |
| Gambar 3.7  | Bentuk Fisik Pneumatic . . . . .                       | 30 |
| Gambar 3.8  | Bentuk Fisik Kompresor . . . . .                       | 30 |

## **Intisari**

Penelitian ini bertujuan untuk melakukan pengoperasian terhadap robot SCARA. SCARA merupakan akronim untuk Selective Compliance Assembly Robot Arm dimana robot ini dapat bergerak dalam dua aksis, yaitu horisontal dan vertikal. Pergerakan Robot ini menggunakan kedua lengan untuk pergerakan horisontal dan satu lengan untuk pergerakan vertikal. Robot SCARA ini dioperasikan dengan bantuan antarmuka yang dibuat dari Processing yang dibuat menggunakan program bahasa c. Antarmuka yang ditampilkan menunjukkan pengoperasian robot SCARA mulai dari kinematika maju dan kinematika balik.

Kata kunci : SCARA, Processing, Kendali, inverse kinematic, Forward Kinematics.



## *Abstract*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi serta ilmu pengetahuan pada masa ke masa semakin berkembang. Perkembangan ini berjalan seiring dengan penelitian-penelitian di berbagai disiplin ilmu khususnya dalam bidang instrumentasi dan kendali. Hal ini dapat dilihat dari banyaknya penggunaan sistem instrumentasi dan kendali dalam dunia industri seperti penggunaan robot dalam menyelesaikan pekerjaan manusia. Untuk itu perancangan robot merupakan salah satu solusi untuk memenuhi tuntutan dalam membantu kebutuhan manusia.

Pemilihan robot untuk menggantikan pekerjaan manusia tidak terlepas dengan berbagai kelebihanannya. Salah satu kelebihanannya, sebuah robot dapat melakukan suatu pekerjaan yang sama dan berulang tanpa merasakan lelah seperti halnya manusia. Pekerjaan ini lah yang biasa ditemukan dalam bidang industri khususnya pada bagian produksi. Robot dengan sistem lengan robot (*robot arm sistem*) merupakan salah satu jenis robot yang dominan berada dalam bidang industri.

Robot lengan memiliki berbagai jenis salah satunya adalah robot SCARA (*Selective Compliance Assembly Robot Arm*). Robot SCARA dapat bergerak secara optimal dan efisien karena sebuah persamaan kinematika. Persamaan kinematika yang digunakan adalah *inverse kinematic* dengan masukan berupa titik koordinat kartesius  $(x_1, y_2)$  dan keluaran berupa nilai sudut untuk mengendalikan motor DC pada *shoulder* dan *elbow*.

Dalam mengendalikan sebuah robot dibutuhkan *platform* antarmuka sebagai jembatan antara *user* dengan *hardware*. Dalam penelitian ini program antarmuka dirancang menggunakan *software* Processing IDE. *Software* Processing IDE memiliki beberapa keunggulan yang membuatnya lebih efektif dan cukup mudah untuk digunakan sebagai *platform* antarmuka. Keunggulan tersebut salah satunya mudahnya sarana komunikasi terhadap *hardware* yang digunakan. Oleh karena itu, pada program kerja praktik ini dilakukan analisis kinematika robot SCARA dengan perancangan antarmuka berbasis Processing IDE.

## 1.2 Tujuan Penelitian

Adapun tujuan dalam melaksanakan penelitian "Kinematika dan Antarmuka Robot SCARA Berbasis Processing IDE" adalah sebagai berikut:

### 1.2.1 Secara Umum

1. Merancang *arm manipulator robot* SCARA berbasis Arduino Mega 2560.
2. Memahami dan mengimplementasikan antarmuka Processing *Integrated Development Environment* (IDE).
3. Mengimplementasikan kinematika pada *arm manipulator robot* SCARA.

### 1.2.2 Tujuan Khusus

Untuk memenuhi salah satu syarat kelulusan dalam menempuh pendidikan Program Diploma III Teknologi Listrik, Sekolah Vokasi, Universitas Gadjah Mada.

## 1.3 Batasan Penelitian

Pembatasan masalah diperlukan untuk mempermudah pelaksanaan penulisan laporan kerja praktik sehingga tidak menyimpang dari judul laporan. Lingkup pembatasan masalah dalam Laporan kerja praktik ini dibatasi pada:

1. Akurasi dari robot lengan dipengaruhi oleh spesifikasi dan torsi dari masing-masing motor DC pada *joint*.
2. Rancangan mekanik yang sudah tersusun dari awal sehingga tidak dapat diubah lagi.
3. Motor DC pada setiap sendi dari *joint* robot memiliki kebutuhan arus yang berbeda – beda.
4. Komunikasi antara Processing IDE dan perangkat menggunakan komunikasi serial.

## 1.4 Metode Kerja Praktik

Metodologi adalah suatu cara yang digunakan untuk memperoleh data yang akurat, baik melalui observasi lapangan maupun dari *datasheet* setiap alat yang digunakan. Observasi juga dilakukan dengan meninjau jurnal-jurnal dan konsultasi mengenai penelitian yang dilakukan. Pada bagian ini dijelaskan meliputi waktu dan tempat penelitian, alat dan bahan penelitian, rancangan alat, metode penelitian dan prosedur penelitian. Penjelasan lebih rinci tentang metodologi penelitian akan dijelaskan sebagai berikut:

### 1. Waktu dan Tempat Penelitian

Penelitian dilakukan di Labolatorium Instrumentasi dan Kendali Diploma Teknik Elektro Sekolah Vokasi Universitas Gadjah Mada pada bulan Juli sampai bulan Agustus 20189.

### 2. Alat dan Bahan

Peralatan yang digunakan dalam Kerja Praktik adalah personal komputer, Arduino Mega 2560, AC-to-DC Converter 12V, modul DC-to-DC Converter LM2596, IC TIP 31, multimeter, Relay Pneumatik, Driver Motor EMS 30A dan catu daya. Sedangkan bahan yang digunakan adalah Motor DC yang terpasang di setiap *joint* robot lengan.

### 3. Pengumpulan Data

Studi pustaka dilakukan dengan cara mengumpulkan buku-buku, dokumen, serta jurnal-jurnal berbentuk *e-book* yang berkaitan dengan robot lengan. Selain itu, *datasheet* dari setiap komponen juga ditinjau. Data-data tersebut menjadi referensi untuk merancang, membuat dan menguji alat.

Konsultasi dilakukan untuk mengumpulkan data melalui tanya jawab atau berdiskusi dengan pihak yang mengetahui dan menguasai segala permasalahan yang dihadapi dalam merancang, membuat, dan menguji robot lengan SCARA. Dalam metode ini penulis berdiskusi dengan dosen pembimbing Kerja Praktik.

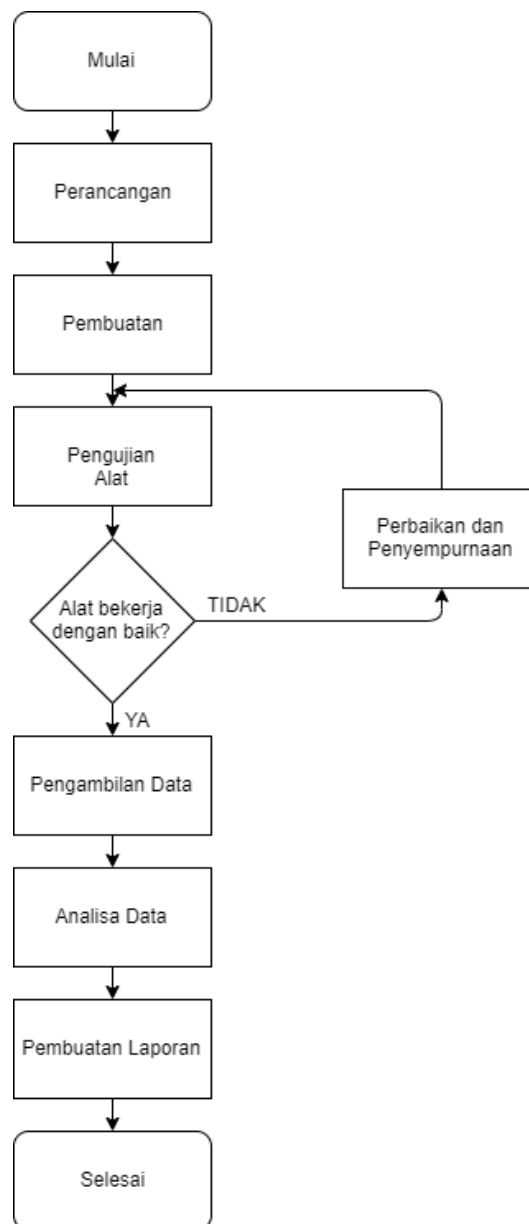
### 4. Metode Penelitian

Metode penelitian yang dilakukan meliputi: Pengujian keluaran tegangan catu daya, pengujian kinerja motor DC, pengujian konfigurasi driver motor H – Bridge, pengujian *Graphical User Interface* (GUI) pada Processing IDE, pe-

ngujian perhitungan kinematika balik, pengujian kinematika maju, serta pengambilan data dan analisa data.

## 5. Prosedur Penelitian

Prosedur penelitian adalah langkah-langkah dalam menyelesaikan kerja praktik yang akan disajikan dalam bentuk diagram alir. Berikut adalah gambar diagram alir prosedur penelitian yang disajikan pada Gambar 1.1.



**Gambar 1.1:** Flowchart Prosedur Penelitian

## 1.5 Sistematika Penulisan

Penulisan laporan kerja praktik ini dilakukan dengan mengikuti sistematika sebagai berikut:

### **BAB I : PENDAHULUAN**

Merupakan pendahuluan dari laporan kerja praktik yang menjelaskan latar belakang, tujuan, batasan masalah dan metodologi penyusunan laporan proyek akhir.

### **BAB II : LANDASAN TEORI**

Memuat gambaran umum robot lengan, *Degree of Freedom*, konfigurasi robot lengan, *wrist* dan end-effector, kinematika, pengertian dan prinsip kerja Processing IDE sebagai antarmuka robot lengan, dan prinsip kerja setiap piranti yang digunakan dalam pembuatan robot lengan.

### **BAB III : PERANCANGAN SISTEM**

Memuat perancangan sistem secara umum, perancangan perangkat keras berupa elektronis dari robot lengan, perancangan perangkat lunak berupa pengolahan *Graphical User Interface* (GUI) menggunakan Processing IDE, sistem kinematika balik untuk robot lengan, dan integrasi keseluruhan program.

### **BAB IV : PENGUJIAN DAN ANALISA KERJA SISTEM**

Memuat pengujian dan analisis kerja motor DC yang berfungsi sebagai *joint* dari robot lengan, pengujian umpan balik melalui GUI Processing IDE serta pengujian sistem secara menyeluruh.

### **BAB V : PENUTUP**

Memuat tentang kesimpulan dari perancangan, pengujian dan analisis dari sistem kerja robot, serta berisi saran – saran untuk mengembangkan penelitian *arm manipulator robot SCARA* lebih lanjut.

## BAB II

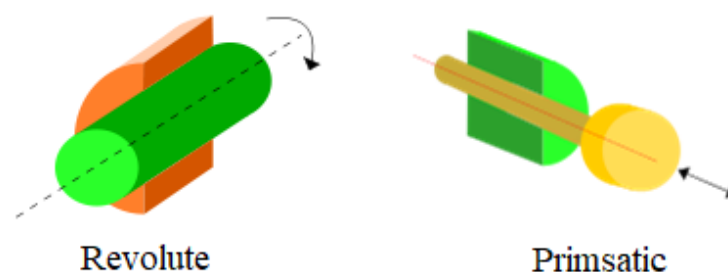
### LANDASAN TEORI

#### 2.1 Gambaran Umum Robot Lengan

Robot adalah sebuah alat yang terdiri dari gabungan mekanik dan elektronik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kendali manusia maupun secara otomatis. Robot dapat melakukan suatu tugas secara berulang tanpa merasa lelah sehingga robot banyak digunakan dalam dunia industri khususnya pada bidang produksi. Salah satu jenis robot yang sering digunakan dalam bidang produksi adalah sistem lengan robot.

Robot lengan adalah robot yang memiliki bentuk fisik seperti halnya lengan pada manusia dan memiliki derajat kebebasan (*Degree of Freedom*) tertentu bergantung pada jumlah sendi yang digunakan. Robot lengan pada bidang industri biasa digunakan sebagai aktuator untuk mengambil dan meletakkan suatu objek secara terus menerus.

Pada umumnya struktur robot lengan terdiri dari beberapa bagian. Bagian utama dari robot lengan adalah struktur mekanik (*Manipulator*) yang merupakan susunan kerangka yang tidak dapat digerakkan (*Rigid*) dan lengan (*Link*) yang satu sama lain terhubung oleh sendi (*joint*). Dengan adanya *joint* yang menghubungkan dua *link* menjadi satu kesatuan sehingga *joint* membentuk satu derajat kebebasan. Jika diibaratkan dengan tubuh manusia, *link* adalah tulang sedangkan *joint* adalah sendi-sendinya. *Joint* memiliki dua pergerakan, yaitu pergerakan *revolute joint* (gerak berputar) dan *prismatic joint* (gerak bergeser). Jenis-jenis dari pergerakan *joint* ditunjukkan pada Gambar 2.1.



**Gambar 2.1:** Jenis-Jenis *Joint*

Pada ujung pangkal lengan, robot lengan umumnya menggunakan *gripper* yang dapat dipakai untuk memindahkan suatu objek. Robot lengan dalam menjalankan tugasnya dikontrol menggunakan sensor serta aktuator yang telah dirancang untuk melakukan tugas sesuai dari yang diperintahkan. Perpaduan antara sensor dan aktuator ini yang menyebabkan robot lengan dapat bekerja secara optimal dan presisi.

### 2.1.1 *Degree of Freedom*

*Degree of Freedom* (DOF) merupakan sebuah konfigurasi yang dapat meminimalkan spesifikasi dengan menggunakan parameter yang dapat menyatakan posisi suatu sistem pada setiap saat. Umumnya robot lengan mempunyai paling sedikit enam independen derajat kebebasan, tiga derajat kebebasan untuk translasi dan tiga derajat kebebasan untuk rotasi. Robot lengan paling tidak memiliki tiga derajat kebebasan untuk dapat memiliki *workspace* yang cukup. *Workspace* dari sebuah robot lengan merupakan total volume yang dapat dijangkau oleh *end-effector* dari pergerakan semua *joint*-nya dari titik minimum hingga maksimum.

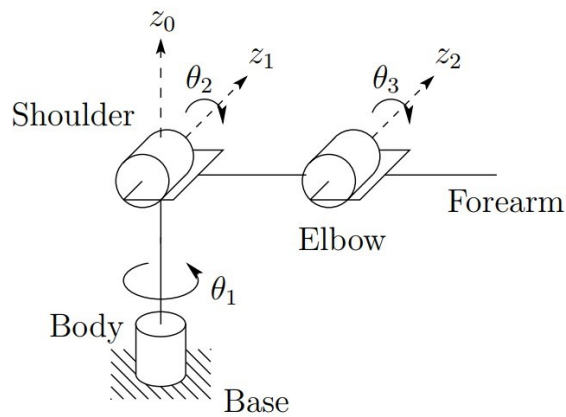
### 2.1.2 Konfigurasi Robot Lengan

Pada dasarnya berbagai jenis dari robot lengan dapat dibedakan dari konfigurasinya. Konfigurasi robot lengan merupakan perpaduan antara pergerakan *joint* yang dimiliki oleh robot lengan. Konfigurasi ini memiliki tipe yang berbeda-beda sehingga *workspace* yang dimiliki pada tiap robot lengan pasti berbeda.

#### 2.1.2.1 Konfigurasi *Articulated (Revolute - Revolute - Revolute)*

*Articulated* manipulator ini pada dasarnya mempunyai jenis *revolute joint* pada ketiga *joint* robot lengan (*base, shoulder, elbow*). Dengan konfigurasi ini, robot lengan dengan konfigurasi *Articulated* dapat memiliki variasi DOF yang banyak. DOF yang dapat dihasilkan dengan robot lengan dengan konfigurasi seperti ini adalah tiga DOF hingga sampai dengan enam DOF tergantung dari kebutuhan dan fungsi yang akan dilakukan oleh robot lengan. Konfigurasi dari *joint revolute* ini menjadikan robot lengan jenis ini mempunyai kebebasan yang besar dari pergerakannya dalam ruang yang kecil sehingga menjadikan jenis konfigurasi *Articulated* manipulator ini banyak dipakai dan memiliki desain yang populer. Konfigurasi *Articulated* ini ditunjukkan pada Gambar 2. 2.

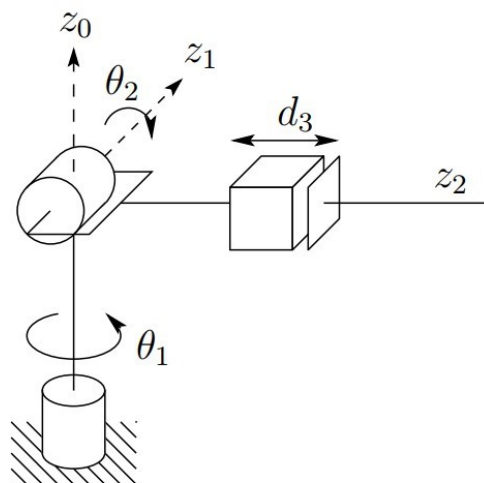




**Gambar 2.2:** Struktur dari Konfigurasi *Articulated*

#### 2.1.2.2 Konfigurasi *Spherical (Revolute – Revolute – Prismatic)*

Konfigurasi *Spherical* merupakan konfigurasi yang mempunyai dua buah *joint revolute* dan satu buah *joint prismatic*. *Joint prismatic* berada ini *joint* ketiga atau pada bagian *elbow*. Sementara dua *joint* lainnya berada di *shoulder* dan *wrist*. Struktur dari konfigurasi *Spherical* ditunjukkan pada Gambar 2.3.

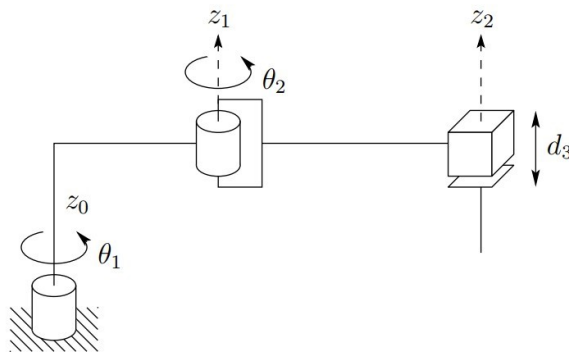


**Gambar 2.3:** Struktur dari Konfigurasi *Spherical*

#### 2.1.2.3 Konfigurasi SCARA (*Revolute – Revolute – Prismatic*)

Konfigurasi *Selective Compliant Articulated Robot for Assembly (SCARA)* merupakan konfigurasi yang mempunyai dua buah *joint revolute* dan satu buah *joint*

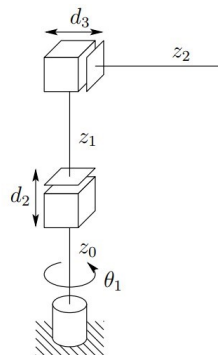
*prismatic* sama seperti konfigurasi *Spherical*. Meskipun SCARA memiliki struktur *joint revolute – revolute – prismatic* (RRP) sama seperti konfigurasi yang dimiliki *Spherical*, struktur ini sedikit berbeda dengan konfigurasi *Spherical* dari tampilannya maupun dari jarak *workspace* nya. Tidak seperti konfigurasi *Spherical*, dimana  $z_0$  tegak lurus terhadap 1, dan  $z_1$  tegak lurus dengan  $z_2$ , konfigurasi SCARA memiliki struktur  $z_0$ ,  $z_1$ , dan  $z_2$  yang paralel. Struktur dari konfigurasi SCARA ditunjukkan pada Gambar 2.4.



**Gambar 2.4:** Struktur dari Konfigurasi SCARA

#### 2.1.2.4 Konfigurasi *Cylindrical* (*Revolute – Prismatic – Prismatic*)

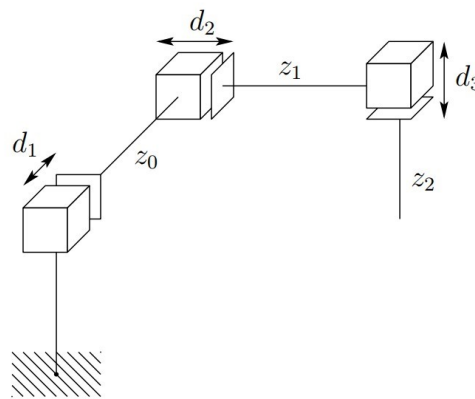
Konfigurasi *Cylindrical* merupakan konfigurasi yang mempunyai satu buah *joint revolute* dan dua buah *joint prismatic*. *Joint revolute* menghasilkan pergerakan rotasi di *base*, sementara *joint prismatic* berada di bagian *shoulder* dan *elbow*. Struktur dari konfigurasi *Cylindrical* ditunjukkan pada Gambar 2.5



**Gambar 2.5:** Struktur dari Konfigurasi *Cylindrical*

### 2.1.2.5 Konfigurasi *Cartesian* (*Prismatic – Prismatic – Prismatic*)

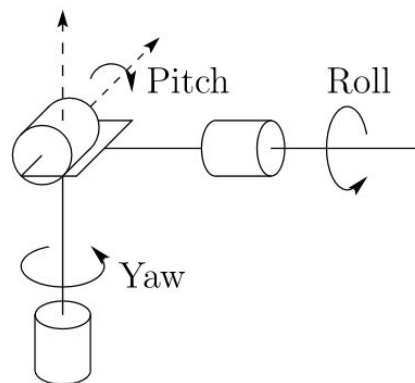
Konfigurasi *Cartesian* mempunyai tiga buah *joint prismatic*. Variabel *joint* dari konfigurasi *prismatic* adalah koordinat *Cartesian* dari *end-effector* dengan memperhatikan letak *base* dari robot lengan. Kinematika dari jenis konfigurasi ini adalah yang paling sederhana dari semua konfigurasi robot lengan. Konfigurasi *Cartesian* sangat berguna untuk penyusunan suatu barang di bidang datar seperti mesin laser, kargo atau memindahkan barang. Struktur dari konfigurasi *Cartesian* ditunjukkan pada Gambar 2.6.



**Gambar 2.6:** Struktur dari Konfigurasi *Cartesian*

### 2.1.3 Wrist dan *End-effector*

*Wrist* atau pergelangan tangan merupakan *joint* diantara lengan dan *end-effector*. *Joint wrist* ini pada umumnya terdapat *joint revolute*. Hal ini umum digunakan pada desain manipulator lengan dengan konfigurasi *Spherical*. Konfigurasi *Spherical* mempunyai *joint revolute* yang saling berpotongan diantara ketiganya, maksudnya setiap *joint* berputar sesuai koordinat x, y dan z. Rotasi atau perputaran dengan *axis* sumbu x adalah *roll*, perputaran dengan *axis* sumbu y adalah *pitch* dan perputaran dengan *axis* sumbu z adalah *yaw*. *Joint spherical wrist* ini dijelaskan pada Gambar 2.7.



**Gambar 2.7:** Struktur dari *Joint Spherical Wrist*

*End-effector* merupakan perangkat atau alat yang terhubung dengan ujung lengan robot. *End-effector* adalah bagian robot yang berhubungan langsung dengan objek. Struktur, pergerakan, material dari *end-effector* bergantung pada tugas yang akan dilakukan robot tersebut. Struktur dan bentuk dari *end-effector* ditunjukkan pada Gambar 2.8.



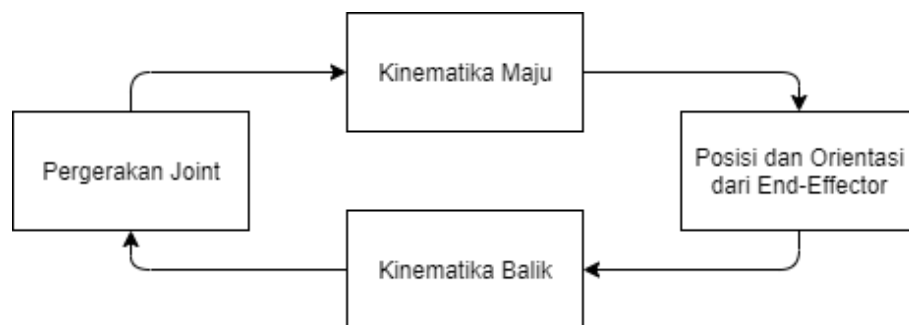
**Gambar 2.8:** Struktur *End-Effector*

## 2.2 Kinematika

Kinematika merupakan pembelajaran pergerakan tubuh robot tanpa memperhitungkan gaya, torsi maupun momen tertentu yang menyebabkan pergerakan. Terdapat berbagai jenis pergerakan dari kinematika tergantung dari tujuan dari setiap robot.

Kinematika yang dijelaskan disini adalah kinematika yang khusus mempelajari dan menganalisa pergerakan robot lengan.

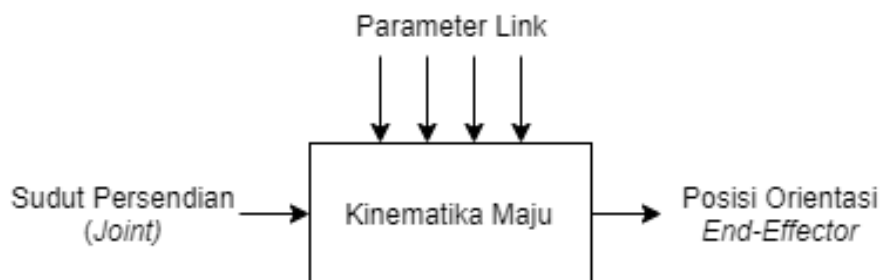
Pada kinematika robot, terdapat dua buah pembahasan kinematika. Pembahasan pertama adalah kinematika maju yang merupakan proses menghitung orientasi dan posisi dari *end-effector* berdasarkan sudut-sudut dari *joint*. Sedangkan kinematika balik sebaliknya dari kinematika maju, diberikan posisi *end-effector*, dimana yang akan dicari adalah besaran sudut yang harus diubah untuk tiap *joint* dalam mencapai posisi *end-effector* tersebut. Diagram blok sederhana dari pemodelan kinematika ditunjukkan pada Gambar 2.9.



**Gambar 2.9:** Blok Diagram Kinematika

### 2.2.1 Kinematika Maju

Kinematika maju atau biasa disebut *forward kinematics* merupakan kinematika untuk mendapatkan hasil akhir berupa koordinat posisi (x, y, z) dengan diketahuinya variabel sudut pada setiap *joint* dari lengan robot. Variabel sudut tersebut kemudian dilakukan perhitungan satu sama lain hingga pada akhirnya akan mendapatkan koordinat x, koordinat y, dan koordinat z. Proses dari kinematika maju ditunjukkan pada Gambar 2.10.

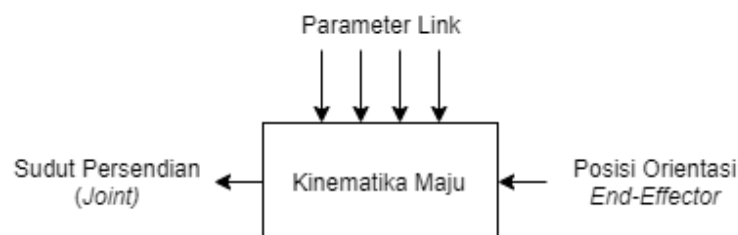


**Gambar 2.10:** Kinematika Maju

### 2.2.2 Kinematika Balik

Kinematika Balik Kinematika balik (*inverse kinematics*) digunakan untuk mencari variabel sudut (*joint*) robot dalam menentukan posisi dan orientasi dari *end-effector*. Dalam menentukan koordinat *end-effector*, kinematika balik mengacu pada penggunaan persamaan kinematika robot untuk menentukan parameter bersama yang memberikan posisi yang diinginkan pada posisi akhir atau *end-effector*. Kinematika balik mengubah rencana gerak menjadi nilai yang harus diberikan bagi aktuator atau penggerak dalam pergerakan robot. Dalam pergerakannya, robot dimodelkan dalam bentuk persamaan kinematika. Persamaan ini menentukan konfigurasi robot dalam hal parameter untuk setiap aktuator. Kinematika maju menggunakan parameter untuk menghitung konfigurasi robot, dan kinematika balik membalikkan perhitungan ini untuk menentukan parameter bersama dalam mencapai konfigurasi yang diinginkan.

Secara garis besar metode kinematika balik akan mencari nilai-nilai parameter yang harus diberikan kepada setiap aktuator untuk mencapai tujuan akhir. Untuk mendapatkan nilai-nilai parameter tersebut, robot harus mengetahui terlebih dahulu manipulator yang dimilikinya, baik ukuran maupun jumlah aktuator serta derajat kebebasan yang ada. Kemudian robot harus ditanamkan rumus-rumus yang didapat dari berbagai model perhitungan, baik dari segi analisa grafik langsung maupun menggunakan metode-metode dari berbagai penelitian. Proses dari kinematika balik ditunjukkan pada Gambar 2.11.



**Gambar 2.11:** Kinematika Balik

## 2.3 Motor DC

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Motor DC mempunyai dua bagian utama, yaitu stator dan rotor. Stator merupakan bagian yang tidak berputar dan rotor merupakan bagian yang berputar dan merupakan kumparan jangkar. Motor DC menghasilkan jumlah putaran dalam setiap satuan waktu yang

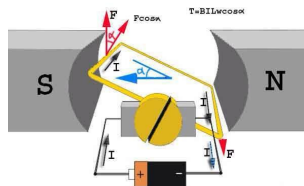
biasanya dihitung setiap satuan menit (*rotations per minute*) dan dapat diatur arah putaranya searah jarum jam (*clock wise*) atau berkebalikan dengan arah jarum jam (*counter clock wise*) bergantung dengan kutub atau polaritas dari catu daya yang diberikan pada motor DC. Bentuk dari motor DC ditunjukkan pada Gambar 2.12.



**Gambar 2.12:** Jenis-Jenis Joint

Motor DC dapat bergerak karena adanya elektromagnet. Saat kumparan diberi arus listrik, permukaan kumparan yang bersifat utara akan bergerak menghadap ke magnet yang berkutub selatan dan kumparan yang bersifat selatan akan bergerak menghadap ke utara magnet. Pada saat ini, karena kedua kutub saling menyebabkan pergerakan kumparan berhenti. Untuk menggerakkannya lagi, tepat pada saat kutub kumparan berhadapan dengan kutub magnet, arah arus pada kumparan dibalik. Dengan demikian, kutub utara kumparan akan berubah menjadi kutub selatan dan kutub selatannya akan berubah menjadi kutub utara.

Pada saat perubahan kutub tersebut terjadi, kutub selatan kumparan akan berhadapan dengan kutub selatan magnet dan kutub utara kumparan akan berhadapan dengan kutub utara magnet. Karena kutubnya sama, maka akan terjadi tolak menolak sehingga kumparan bergerak memutar hingga utara kumparan berhadapan dengan selatan magnet dan selatan kumparan berhadapan dengan utara magnet. Siklus ini akan berulang-ulang hingga arus listrik pada kumparan diputuskan. Prinsip kerja motor DC dijelaskan pada Gambar 2.13.



**Gambar 2.13:** Prinsip Kerja Motor DC

## 2.4 Regulator

Dalam suatu rangkaian elektronika dibutuhkan suatu sumber tegangan yang stabil dan sesuai dengan nilai yang dibutuhkan oleh komponen. Untuk memenuhi kebutuhan tersebut digunakanlah sebuah rangkaian regulator. Rangkaian regulator berfungsi untuk mengatur atau menghasilkan nilai tegangan pada nilai tertentu dari suatu tegangan masukan. Regulator dapat mempertahankan nilai tegangan yang keluar tanpa dipengaruhi besar arus yang dikeluarkannya. Regulator tegangan mempunyai banyak jenisnya, salah satunya adalah regulator *switching*.

Regulator *switching* mengatur besarnya nilai tegangan keluaran dengan men-saklar (ON/OFF) tegangan masukan dengan frekuensi berbeda – beda. Kelebihan dari regulator *switching* adalah mempunyai disipasi daya yang terjadi lebih kecil dibandingkan dengan regulator linear. Sedangkan kekurangan yaitu tegangan keluarannya akan berbentuk gelombang akibat adanya proses *switching*. Oleh karena itu, regulator jenis ini umumnya membutuhkan induktor, kapasitor , dan dioda untuk memperhalus tegangan keluaran. Regulator *switching* ada dua jenis yaitu regulator *Buck* dan regulator *Boost*. Regulator *Buck* untuk menghasilkan nilai tegangan keluaran yang lebih kecil dari tegangan masukannya. Sedangkan Regulator *Boost* untuk menghasilkan nilai tegangan yang lebih besar dari tegangan masukannya. Salah satu jenis dari regulator *Buck* adalah LM2596 . Bentuk fisik dari regulator *Buck* ditunjukkan pada Gambar 2.14.



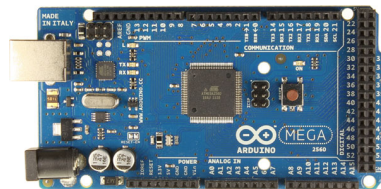
**Gambar 2.14:** Regulator *Buck* LM2596

## 2.5 Arduino

Arduino merupakan papan pengembangan dari mikrokontroler yang menggunakan basis Arduino serta menggunakan microchip At 2560. Arduino memiliki pin input dan output dengan jumlah yang cukup banyak, yaitu sejumlah 54 buah pin input dan output yang 15 diantaranya merupakan pin pulse with modulation (PWM),



16 diantaranya merupakan pin analog input, dan terdapat empat pin yang digunakan sebagai UART (*Serial port hardware*). Arduino ini sudah dilengkapi dengan oscillator sebesar 16Mhz, sebuah port usb, power jack DC, ICSP header serta tombol reset. Gambar 2.15 Merupakan bentuk fisik Arduino.



**Gambar 2.15:** Arduino Mega 2560

Arduino Mega 2560 memuat semua yang dibutuhkan untuk mendukung kerja dari sebuah mikrokontroler. Arduino Mega 2560 dapat dengan mudah dioperasikan untuk pengaplikasian ke sebuah sistem kerja karena dapat dihubungkan dengan kabel USB sebagai komunikasinya. Untuk dayanya, Arduino Mega 2560 dapat dihidupkan melalui konektor DC yang dipunyainya dengan diberi tegangan DC, dapat dengan adapter DC atau baterai dengan nilai tegangan sesuai dengan spesifikasi pada Arduino. Spesifikasi Arduino Mega 2560 ditunjukkan pada Tabel 2.1

**Tabel 2.1:** Spesifikasi Arduino Mega 2560

|                         |                                |
|-------------------------|--------------------------------|
| Mikrokontroler          | Atmega2560                     |
| Tegangan operasional    | 5 V                            |
| Tegangan masukan        | 5-12 V                         |
| Pin digital I/O         | 54 (15 pin untuk keluaran PWM) |
| Pin analog masukan      | 16                             |
| Arus DC per Pin I/O     | 20 mA                          |
| Arus DC untuk Pin 3.3 V | 50 mA                          |
| Memori flash            | 256 KB                         |
| Kecepatan clock         | 16 MHz                         |
| Dimensi                 | 101.52 x 53.3 mm               |

## 2.6 Driver Motor H

Driver Motor berfungsi sebagai pengontrol dari setiap pergerakan dari motor DC. Pergerakan seperti kecepatan, arah putar serta lamanya pergerakan motor DC dapat dikontrol oleh sebuah driver motor. Driver Motor memiliki beberapa jenis tergantung dari spesifikasi dari motor DC yang digunakan. Salah satu jenis driver motor adalah driver motor EMS 30A H-Bridge. Bentuk fisik dari driver motor EMS 30A H-Bridge ditunjukkan pada Gambar 2.16.



**Gambar 2.16:** Driver Motor EMS 30A H-Bridge

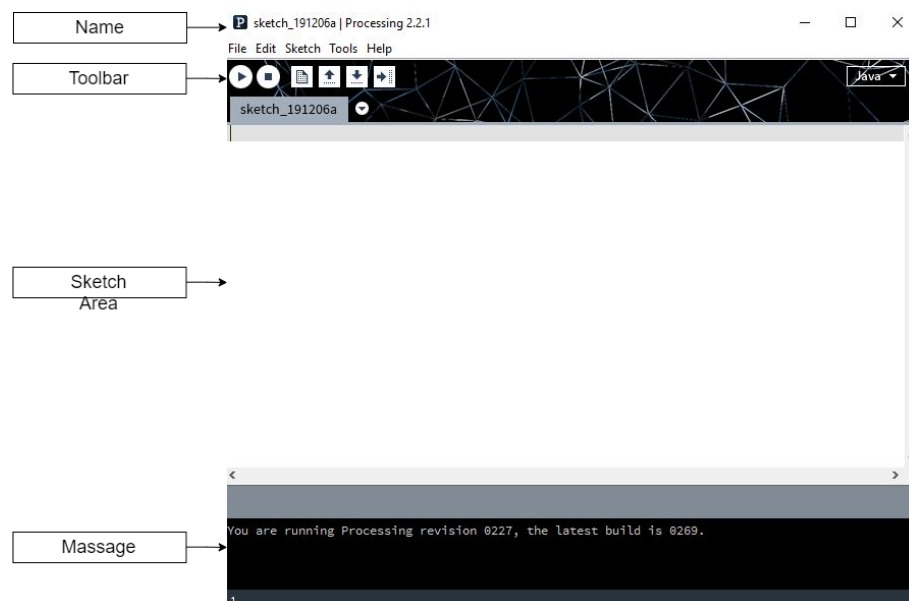
Driver Motor EMS 30A H-Bridge merupakan driver motor yang dapat mengoperasikan sebuah Motor DC dengan batasan arus hingga 30 Ampere. Driver ini memiliki 10 buah pin data yang dapat dihubungkan dengan sebuah mikrokontroler. Alokasi pin Driver Motor EMS 30A H-Bridge ini ditunjukkan pada Tabel 2.2.

**Tabel 2.2:** Pin pada Driver EMS 30A H-Bridge

| Pin  | Nama | I/O | Fungsi  |
|------|------|-----|---|
| 1    | MIN1 | I   | Pin input untuk menentukan output MOUT1   |
| 2    | MIN2 | I   | Pin input untuk menentukan output MOUT2   |
| 3    | MEN1 | I/O | Pin enable untuk output MOUT1<br>diberi logika HIGH untuk mengaktifkan half H-Bridge 1,<br>diberi logika LOW untuk menonaktifkannya |
| 4    | MEN2 | I/O | Pin enable untuk output MOUT2<br>diberi logika HIGH untuk mengaktifkan half H-Bridge 2,<br>diberi logika LOW untuk menonaktifkannya |
| 5    | MSC  | O   | Output tegangan analog yang berbanding lurus dengan arus beban  |
| 6    | MPWM | I   | Pin input untuk mengatur kerja modul H-Bridge secara PWM  |
| 7,9  | VCC  | -   | Terhubung ke catu daya untuk input (5 Volt)   |
| 8,10 | PGND | 0   | Titik referensi untuk catu daya input   |

## 2.7 Processing Integrated Development Environment (IDE)

Processing IDE adalah pemrograman sederhana yang diciptakan untuk pengembangan aplikasi yang berorientasi visual atau pencitraan dengan penekanan pada animasi dan memberi pengguna umpan balik melalui interaksi antarmuka. Para developer Processing IDE ini menginginkan sebuah cara untuk "membuat sketsa" gagasan dalam bentuk kode. Karena pemrograman telah berkembang selama beberapa dekade terakhir, Processing IDE telah mulai digunakan untuk pekerjaan tingkat produksi yang lebih maju. Awalnya dibangun sebagai ekstensi khusus domain Java yang ditargetkan untuk seniman dan desainer, Processing IDE telah berkembang menjadi alat desain dan prototipe lengkap dan penuh yang digunakan untuk pekerjaan instalasi skala besar, grafis gerak, dan visualisasi data yang rumit. Tampilan pada Processing IDE ditunjukkan pada gambar 2.17.



**Gambar 2.17:** Processing IDE

### 2.7.1 Syntax dalam Processing IDE

#### 2.7.1.1 Mengatur ukuran

Ukuran sebuah objek pada processing IDE diatur menggunakan sebuah syntax `size()`. Fungsi `size()` digunakan untuk menetapkan variabel global lebar dan tinggi dari suatu program. Ukuran untuk panjang dan tinggi tersebut menggunakan ukuran piksel. Lebar diwakili dengan variabel "*width*" dan tinggi diwakili dengan variabel.

Untuk objek yang ukurannya tergantung pada layar, selalu gunakan variabel lebar dan tinggi, bukan angka. Ini mencegah masalah saat parameter fungsi `size()` diubah. Berikut penulisan dari fungsi `size()` untuk menetapkan lebar dan tinggi suatu program: `size(1000, 1000);`

### 2.7.1.2 Transformasi Bentuk Pada Processing IDE

Transformasi pada Processing IDE digunakan untuk memindahkan, memutar atau, mengecilkan atau membesarkan suatu objek dan perpindahannya dapat diatur dengan parameter-parameter tertentu di dalam Processing IDE. Transformasi adalah dasar dari pemrograman Processing IDE. Tabel 2.4 adalah contoh penulisan *syntax* dari Transformasi.






**Tabel 2.3:** *Syntax* Transformasi

| No | <i>Syntax</i>               | Keterangan   |
|----|-----------------------------|--|
| 1  | <code>Translate(x,y)</code> | x = transisi kiri/kanan<br>y = transisi atas/bawah |
| 2  | <code>Rotate(angle)</code>  | angle=besar sudut rotasi(radian)                   |
| 3  | <code>RotateX(angle)</code> | angle=besar sudut rotasi(radian)                   |
| 4  | <code>RotateY(angle)</code> | angle=besar sudut rotasi(radian)                   |
| 5  | <code>Rotatez(angle)</code> | angle=besar sudut rotasi(radian)                   |
| 6  | <code>Scale(S)</code>       | s = besar pengecilan pembesaran                    |

### 2.7.1.3 Shape

Shape adalah *syntax* pada Processing IDE yang berfungsi untuk membuat berbagai macam bentuk, seperti persegi panjang, lingkaran, garis, dan bentuk lainnya yang dapat diatur ukurannya dengan parameter-parameter tertentu. Tabel 2.3 merupakan contoh dari beberapa *syntax* shape.

**Tabel 2.4:** *Syntax Shape*

| No | Syntax                       | Bentuk  | Keterangan  |
|----|------------------------------|---|---|
| 1  | ellipse(a,b,c,d)             |    | a = koordinatX<br>b = koordinatY<br>c= lebar diameter<br>d= tinggi diameter   |
| 2  | arc(a,b,c,d,start,stop,mode) |    | a = koordinatX<br>b = koordinatY<br>c = lebar diameter<br>d = tinggi diameter<br>start = sudut mulai<br>stop = sudut akhir<br>mode = PIE, CHORD, OPEN |
| 3  | line(x1,y1,x2,y2)            |  | x1 = koordinatX1<br>y1 = koordinatY1<br>x2 = koordinatX2<br>y2 = koordinatY2  |
| 4  | point(x,y)                   |  | x = koordinatX<br>y = koordinatY  |
| 5  | rect(a,b,c,d,r)              |  | a = koordinatX1<br>b = koordinatY1<br>c = koordinatX2<br>d= koordinatY1   |

#### 2.7.1.4 Akses Koordinat Mouse

Processing IDE mempunyai fungsi khusus yaitu dapat melacak posisi mouse baik secara vertical maupun horizontal. Akan tetapi Processing IDE hanya dapat melacak posisi mouse saat sketch dijalankan saja. Nilai default mouseX dan mouseY adalah 0, jadi 0 akan dikembalikan sampai mouse bergerak di depan jendela sketsa. (ini biasanya terjadi saat sketsa dijalankan pertama kali). Setelah mouse bergerak menjauh dari jendela, mouseX akan terus melaporkan posisi terakhirnya. Tabel 2.6

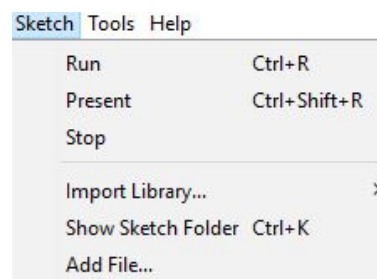
adalah contoh penulisan *syntax* dari mouse.

**Tabel 2.5:** *Syntax* Koordinat Mouse

| No | <i>Syntax</i>  | Keterangan   |
|----|----------------|--|
| 1  | mouseX         | Variabel sistem X memiliki nilai dari koordinat horizontal mouse secara real time    |
| 2  | mouseY         | Variabel sistem Y memiliki nilai dari koordinat horizontal mouse secara real time.   |
| 3  | mouseButton()  | Bila tombol mouse ditekan, nilai variable sistem disetel ke LEFT,RIGHT, atau CENTER. |
| 4  | mouseClicked() | Dipanggil setelah tombol mouse ditekan dan kemudian dilepaskan.                      |
| 5  | mouseDragged() | Dipanggil setelah tombol mouse bergerak saat ditekan.                                |

### 2.7.1.5 Compile Sketch

Program pada Processing IDE dinamakan sketch. Tujuan diberi nama sketch adalah untuk membuat pemrograman bergaya Java ini terasa lebih seperti *scripting*, dan mengadopsi proses *scripting* untuk menulis kode dengan cepat. Sketch disimpan dalam bentuk Sketchbook, folder yang digunakan sebagai default penyimpanan semua proyek Processing IDE. Sketch yang tersimpan dalam Sketchbook dapat diakses dari File - Sketchbook. Sebagai alternatif, File - Open.. dapat digunakan untuk membuka sketch dari tempat lain pada sistem. Gambar 2. 18 adalah opsi toolbar untuk menjalankan sketch. Run (Ctrl + R) adalah opsi untuk menjalankan sketch yang telah dibuat, begitu pula dengan Tweak. Present (Ctrl + Shift + R) berfungsi untuk menjalankan sketch dengan layer penuh (*Fullscreen*). Stop berfungsi untuk menghentikan sketch.



**Gambar 2.18:** Toolbar Sketch

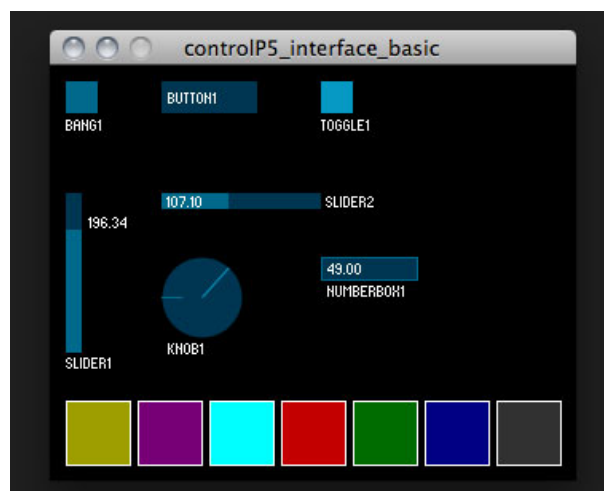
## 2.7.2 Library untuk Processing IDE

### 2.7.2.1 Memuat Huruf (*Font*) pada Processing IDE

Pfont adalah class yang digunakan untuk penggunaan font pada Processing IDE. Untuk membuat *font* yang akan digunakan dengan Processing IDE, dengan memilih "*Create Font ...*" dari menu *Tools*. Ini akan membuat *font* dalam format Processing IDE yang membutuhkan dan juga menambahkannya ke direktori data sketsa saat ini. Pengolahan menampilkan font menggunakan format font .vlw. Fungsi loadFont() digunakan untuk membuat *font* baru dan textFont () digunakan agar *font* baru yang telah dibuat tersebut aktif. *Syntax* list() berfungsi untuk membuat daftar *font* yang terinstal di komputer, yang merupakan informasi yang berguna untuk digunakan dengan fungsi createFont () untuk mengubah *font* secara dinamis menjadi format yang bisa digunakan di Processing IDE.

### 2.7.2.2 Memuat ControlP5 pada Processing IDE

ControlP5 adalah library yang berfungsi sebagai pengontrol untuk membuat antarmuka dengan user, ControlP5 berupa grafis di dalam sketch Processing IDE yang meliputi Slider, Button, Toggles, Knobs, Radiobutton, dan dapat dengan mudah ditambahkan ke sketch pengolahan. ControlP5 ditulis oleh Andreas Schlegel untuk pemrograman Processing IDE. Library ControlP5 bisa diunduh di laman <https://github.com/sojamo/controlp5>, atau unduh di library Processing IDE secara langsung. Beberapa contoh dari penggunaan ControlP5 ditunjukkan pada Gambar 3.2



Gambar 2.19: ControlP5

Setiap *tools* yang ada pada ControlP5 memiliki fungsi yang berbeda-beda sesuai dengan kegunaannya. ControlP5 dapat dipakai atau saling dikombinasikan agar memudahkan dan memaksimalkan dari GUI yang telah dibuat. ControlP5 nantinya akan bekerja sesuai dengan program yang sudah dimasukkan. Pada Tabel 3.3 merupakan beberapa fungsi dari *tools* yang ada pada ControlP5.

**Tabel 2.6:** Fungsi *Tools* ControlP5

|   |           |  |                                       |
|---|-----------|--|---------------------------------------|
| 1 | Bang      | Memicu sebuah tindakan ketika ditekan                                    | controlP5.addBang("bang1");           |
| 2 | Button    | Memicu sebuah tindakan ketika sudah dilepas                              | controlP5.addButton("button1");       |
| 3 | Toggle    | Memiliki dua status.<br>TRUE memiliki nilai 1 dan FALSE memiliki nilai 0 | controlP5.addToggle("toggle1")        |
| 4 | Slider    | Mengatur nilai dengan cara menggeser secara horisontal maupun vertikal   | controlP5.addSlider("slider2");       |
| 5 | Knob      | Mengatur sebuah nilai dengan tombol putaran 360°                         | controlP5.addKnob("knob1");           |
| 6 | NumberBox | Kotak yang menampilkan angka dan dapat diubah dengan klik di dalam kotak | controlP5.addNumberbox("numberbox1"); |



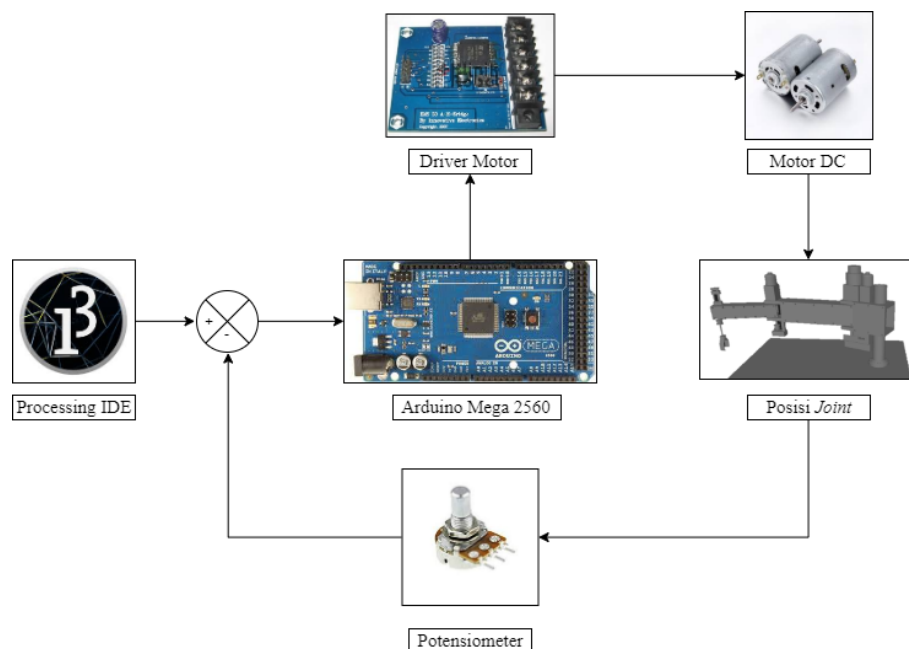
## BAB III

### PERANCANGAN SISTEM

Pada bab ini akan disajikan mekanisme perancangan alat, baik perangkat keras ataupun perangkat lunak untuk mewujudkan sebuah robot lengan. Tahapan perancangan dimulai dari perancangan diagram blok sistem, perancangan perangkat keras, perancangan perangkat lunak, perancangan kinematika balik, dan integrasi keseluruhan program.

#### 3.1 Diagram Blok Sistem

Secara garis besar, pada tahapan implementasi dari kinematika pada *arm manipulator robot SCARA* ini menggunakan output atau penggerak berupa motor DC dengan *feedback* posisi berupa potensiometer dan pada bagian input yang berasal dari Processing IDE yang mengirimkan sebuah koordinat yang digunakan untuk menentukan pergerakan robot berdasarkan fungsi *inverse kinematic*. Sistem kerja *arm manipulator robot SCARA* ditunjukkan pada Gambar 3.1.



**Gambar 3.1:** Diagram Blok Sistem

Pada blok diagram yang disajikan pada Gambar 3.1 sistem terdiri dari bagian

– bagian yang meliputi bagian masukan, bagian kendali, bagian keluaran, dan bagian penampil. Pada bagian masukan menggunakan GUI yang dirancang pada Processing IDE yang digunakan sebagai *forward kinematic* serta *inverse kinematic* dimana robot akan bergerak menyesuaikan dengan posisi atau sudut yang diinputkan melalui Processing IDE. Pada bagian kontrol menggunakan Arduino Mega 2560 sebagai mikrokontroler yang mengendalikan seluruh operasi dari robot. *Power supply* Arduino sebesar 5 volt DC didapatkan dari regulator tegangan yang menurunkan tegangan dari 12 volt DC ke 5 volt DC.

Pada bagian keluaran, pin pulse with modulation (PWM) pada Arduino Mega 2560 dihubungkan dengan driver motor yang digunakan untuk mengontrol arah pergerakan dari motor DC serta kecepatan pergerakan dari motor DC. Pergerakan arah putaran motor DC bergantung pada *feedback* posisi setiap sendi yang diberikan oleh potensiometer. Tiga buah pin digital arduino dihubungkan rangkaian *switch* yang menggunakan IC TIP31 yang berfungsi sebagai kontrol dari *End-Effector* dari robot SCARA.

Pada bagian penampil merupakan bentuk dari rancangan GUI yang dirancang dalam Processing IDE melalui sebuah bentuk pemrograman. Dalam tampilan GUI nya, terdapat beberapa *tools* yang dapat untuk mengatur pergerakan robot SCARA. Pada GUI juga dapat menampilkan nilai dari sudut, dan posisi pada kondisi langsung dari pergerakan robot SCARA.

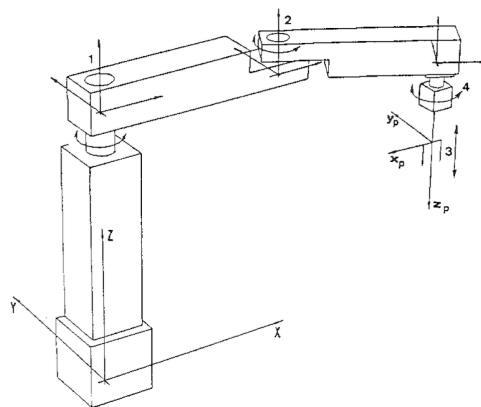
## 3.2 Perancangan Perangkat Keras

Perancangan perangkat keras pada *arm manipulator robot* SCARA terdiri dari dua bagian yaitu bagian mekanik dan elektronis. Bagian mekanik merupakan bagian *hardware* yang meliputi desain, bahan dan bentuk dari arm manipulator robot SCARA dan bagian elektronis merupakan bagian *hardware* yang meliputi sistem – sistem yang berkaitan dengan rangkaian pada I SCARA seperti rangkain pada desain *board* serta komponen – komponen elektronis.

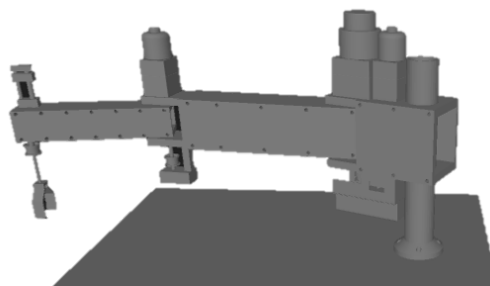
### 3.2.1 Sistem Mekanik

Sistem mekanika dari robot lengan bergantung dari konfigurasi robot lengan. Konfigurasi robot lengan terbagi menjadi enam, yaitu konfigurasi *Articulated*, konfigurasi SCARA, konfigurasi *Spherical*, konfigurasi *Cylindrical*, dan konfigurasi *Cartesian*. Pada tugas ini, konfigurasi robot lengan yang digunakan adalah konfigurasi

SCARA, dengan dua joint revolute dan satu *end-effector*. Sistem mekanik dari lengan robot tiga DOF sangat berpengaruh dan mendominasi sistem karena bentuk dan pergerakan dari mekanik akan mempengaruhi elektronis, serta program. Sistem mekanik yang baik sangat mendukung dari pergerakan robot, oleh karena itu perancangan mekanik haruslah proporsional dari panjang setiap lengan, lebar serta tinggi robot. Pada Gambar 3.2 disajikan *free body* dari robot SCARA dan Gambar 3.3 merupakan bentuk fisik dari robot SCARA yang digunakan pada penelitian kali ini.



**Gambar 3.2:** *Free Body* Robot SCARA



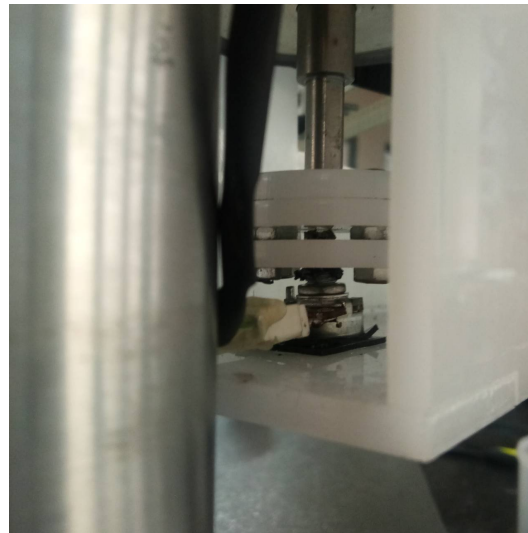
**Gambar 3.3:** Bentuk Fisik Robot SCARA

Robot SCARA merupakan robot yang memiliki tiga buah derajat kebebasan (DOF) yang terletak pada *shoulder*, *elbow*, dan pada *end-effector*. Pada *shoulder* dan *elbow* menggunakan sebuah motor DC dan pada *end-effector* menggunakan bantuan angin yang dikontrol menggunakan *valve pneumatic*. Pergerakan pada masing-masing *joint* memiliki jangkauan maksimum yang berbeda-beda. Jangkauan juga dipengaruhi oleh panjangnya lengan yang dimiliki oleh robot SCARA tersebut. Tabel 3.1 merupakan spesifikasi dari robot SCARA yang digunakan.

**Tabel 3.1:** Spesifikasi Robot SCARA

|                      |        |
|----------------------|--------|
| Main arm length      | 360 mm |
| Fore arm length      | 290 mm |
| Shoulder movement    | 180 °  |
| Elbow movement       | 200 °  |
| Wrist rotation       | 360 °  |
| Up & down movement   | 150 mm |
| Maximum tip velocity | 3.0 kg |

Desain pada *arm manipulator robot SCARA* berbahan besi dengan tebal 2mm dengan tiga derajat kebebasan yang meliputi bagian *shoulder*, *elbow*, serta *end-effector*. Pada desainnya robot SCARA terbagi menjadi dua bagian. Bagian utama adalah box panel yang berisi sistem elektronis utama dan pada bagian yang lain merupakan lengan dari robot SCARA sendiri. Terdapat juga tiga buah selang yang berfungsi untuk mentransformasikan angin untuk pergerakan dari *end-effector* yang berasal dari sebuah kompresor. Dalam komunikasinya dengan komputer personal, robot SCARA dihubungkan dengan konektor USB. Gambar 3.4 merupakan bentuk fisik dari box panel pada Robot SCARA.

**Gambar 3.4:** Box Panel Robot SCARA

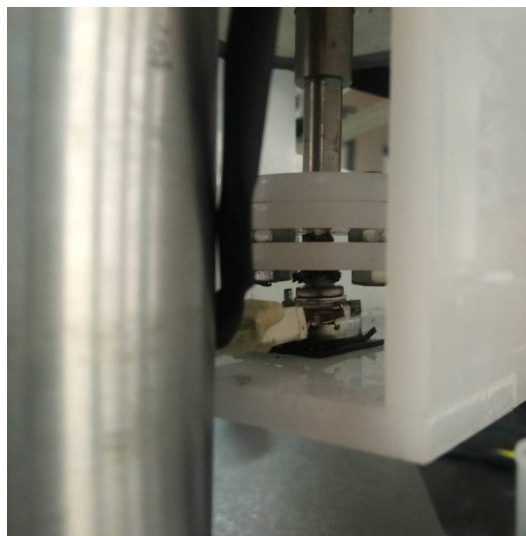
Pada *arm manipulator robot SCARA* menggunakan penggerak berupa motor DC 12 Volt dengan diberikan *gearbox* sehingga mampu mengangkat beban berat ka-

rena torsi pada motor bertambah besar. Motor DC dikontrol dari driver motor EMS 30A H-Bridge melalui mikrokontroler Arduino Mega 2560. Tabel 3.2 merupakan spesifikasi dari motor DC yang digunakan pada robot SCARA.

**Tabel 3.2:** Spesifikasi Motor DC pada Robot SCARA

|   |                      |
|---|----------------------|
| Moments of inertia of the main arm ( $J_1$ )                              | $0.0980kgm^2$        |
| Moments of inertia of the fore arm ( $J_2$ )                              | $0.0115kgm^2$        |
| Masses of the main arm ( $m_1$ )  | $1.90kg$             |
| Masses of the fore arm ( $m_2$ )  | $0.93kg$             |
| Motor and equivalent inertias ( $J_m$ )                                   | $3.3 * 10^{-6}kgm^2$ |
| Back emf constants for main arm and fore arm motor ( $K_{e1} = K_{e2}$ )  | $0.047Nm/A$          |
| Armature resistance for main arm and fore arm motor ( $R_{a1} = R_{a2}$ ) | $3.5\Omega$          |
| Armatures inductances for main and fore arm motor ( $L_{a1} = L_{a2}$ )   | $1.3mH$              |

Pada bagian *gearbox* pada masing-masing motor DC terdapat *potensiometer* sebagai sensor posisi. *Potensiometer* ditempatkan pada bawah motor DC yang terhubung langsung. Setiap pergerakan dari motor DC, *potensiometer* akan secara otomatis juga ikut bergerak dan kemudian mengirimkan nilai data analog ke Arduino Mega. Nilai ini dalam Arduino Mega dilakukan sebuah perhitungan untuk menghasilkan sebuah posisi berupa besar sudut dari setiap motor DC saat dioperasikan. Gambar 3.5 merupakan bentuk fisik dari motor DC serta pemasangan *potensiometer*.



**Gambar 3.5:** Motor DC dengan Potensiometer

Pada bagian *end-effector* menggunakan pergerakan translasi. Pergerakan translasi merupakan pergerakan garis lurus dengan sebuah sumbu. Pada robot SCARA pergerakan ini ada pada bagian *end-effector* yang bergerak secara vertikal atau naik turun. Dengan pergerakan ini, posisi *end-effector* mengalami perubahan pada posisi tingginya. Pergerakan translasi juga terdapat pada bagian *end-effector* yang menyebabkan sebuah *end-effector* dapat membuka dan menutup karena sebuah sistem mekanik yang telah ada. Selain dari pergerakan translasi, pergerakan pada *end-effector* juga terdapat rotasi. Pergerakan ini dilakukan oleh satu buah motor DC yang ditempatkan pada bagian base dengan dihubungkan melalui sebuah belt. Pengoperasian pada motor DC ini juga dilakukan oleh driver motor EMS 30A H-Bridge. Gambar 3.6 merupakan bentuk fisik dari *end-effector*.



**Gambar 3.6:** End-Effector Robot SCARA

Semua pergerakan pada *end-effector* ditenagai oleh sebuah tekanan udara yang bersumber dari sebuah kompresor. Tekanan udara diaplikasikan pada sebuah *pneumatic* dengan sistem kerja translasi yang dapat menyebabkan sebuah objek dapat bergerak pada sebuah garis lurus. Gambar 3.7 merupakan bentuk fisik dari *pneumatic* yang digunakan pada robot SCARA.



**Gambar 3.7:** Bentuk Fisik Pneumatic

Kompresor yang digunakan untuk menghasilkan sebuah tekanan udara merupakan sebuah kompresor listrik dengan kapasitas 8 bar. Kompresor ini dioperasikan menggunakan sumber tegangan AC. Ketika kapasitas udara sudah terpenuhi, kompresor ini dapat digunakan tanpa menggunakan sumber tegangan tetapi hanya sebatas kapasitas udara yang disimpan. Gambar 3.8 merupakan bentuk kompresor yang digunakan.



**Gambar 3.8:** Bentuk Fisik Kompresor

Rancangan robot secara keseluruhan ditampilkan pada Gambar 3.9 yang merupakan rancangan tampak samping, Gambar 3.10 merupakan rancangan tampak atas

dan Gambar 3.11 merupakan rancangan dimensi robot.



## **BAB IV**

### **PENUTUP**

## DAFTAR PUSTAKA

- [1] G. Younkin, *Industrial Servo Control Systems: Fundamentals And Applications, Revised And Expanded*. USA: Marcel Dekker, INC, 2002.