

**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 kehadiran 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 terima kasih 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.

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

HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	vi
DAFTAR GAMBAR	viii
Intisari	ix
<i>Abstract</i>	x
I PENDAHULUAN	1
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
II LANDASAN TEORI	6
2.1 Gambaran Umum Robot Lengan	6
2.1.1 <i>Degree 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 Mega 2560	15
2.6 <i>Driver</i> Motor H-Bridge EMS 30A	16

2.7	Processing <i>Integrated Development Environment</i> (IDE)	18
2.7.1	<i>Syntax</i> dalam Processing IDE	18
2.7.2	<i>Library</i> untuk Processing IDE	21
III	PERANCANGAN SISTEM	24
3.1	Diagram Blok Sistem	24
3.2	Perancangan Perangkat Keras	25
3.2.1	Sistem Mekanis	25
3.2.2	Rangkaian Elektronika	32
3.3	Perancangan Perangkat Lunak	37
3.3.1	ControlP5	38
3.3.2	<i>Shape</i>	39
3.4	Sistem Kinematika	39
3.4.1	Prinsip Kerja Kinematika Maju	40
3.4.2	Prinsip Kerja Kinematika Balik	40
3.5	Perancangan Sistem Keseluruhan	43
IV	PENGUJIAN DAN PEMBAHASAN	45
4.1	Pengujian Fungsional	45
4.1.1	Pengujian DC - to - DC Converter	46
4.1.2	Pengujian Motor DC	46
4.1.3	Pengujian <i>Driver</i> Motor H – <i>Bridge</i>	47
4.1.4	Pengujian Nilai <i>Analog Potensiometer</i>	48
4.1.5	Pengujian Rangkaian <i>Switching Valve Pneumatic</i>	49
4.1.6	Pengujian Kinematika Maju	50
4.1.7	Pengujian Kinematika Balik	50
4.1.8	Pengujian GUI	58
4.2	Pengujian Keseluruhan	60
4.2.1	Pengujian Akurasi Robot Lengan	60
4.2.2	Pengujian Simulasi Robot Lengan	61
V	PENUTUP	64
5.1	Kesimpulan	64
5.2	Saran	64
	DAFTAR PUSTAKA	65

DAFTAR TABEL

Tabel 2.1	Spesifikasi Arduino Mega 2560	16
Tabel 2.2	Pin pada <i>Driver</i> 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 <i>Mouse</i>	21
Tabel 2.6	Fungsi dari <i>tools</i> ControlP5	23
Tabel 3.1	Spesifikasi Robot SCARA	27
Tabel 3.2	Spesifikasi Motor DC pada Robot SCARA	28
Tabel 3.3	Pin pada Arduino Mega 2560	36
Tabel 3.4	Keterangan Tampilan pada GUI Robot SCARA	38
Tabel 3.5	Keterangan Sistem Keseluruhan	44
Tabel 4.1	Hasil Tegangan Keluaran Dari Tegangan DC-DC Converter .	46
Tabel 4.2	Hasil Pengujian Motor DC	46
Tabel 4.3	Hasil Pengujian <i>Driver</i> Motor H-Bridge	47
Tabel 4.4	Hasil Pengujian <i>Potensiometer</i>	48
Tabel 4.5	Hasil Pengujian <i>Potensiometer</i> Menggunakan Program <i>Moving Avarage</i>	49
Tabel 4.6	Hasil Pengujian Rangkaian <i>Switching Vavle Pnemuatic</i>	49
Tabel 4.7	Pengujian Koordinat X	51
Tabel 4.8	Hasil Pengujian Koordinat Y	53
Tabel 4.9	Hasil Pengujian <i>Joint Shoulder</i>	55
Tabel 4.10	Hasil Pengujian <i>Joint Elbow</i>	57
Tabel 4.11	Hasil Pengujian Akurasi Robot Secara Keseluruhan	62

DAFTAR GAMBAR

Gambar 1.1	Diagram Alir 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	Bentuk Fisik Motor DC	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	<i>Driver</i> Motor EMS 30A H-Bridge	17
Gambar 2.17	Tampilan Processing IDE	18
Gambar 2.18	<i>Toolbar Sketch</i>	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 <i>Potensiometer</i>	28
Gambar 3.6	<i>End-Effector</i> Robot SCARA	29
Gambar 3.7	Bentuk Fisik <i>Pneumatic</i>	30
Gambar 3.8	Bentuk Fisik Kompresor	30
Gambar 3.9	Rancangan Tampak Samping	31
Gambar 3.10	Rancangan Tampak Atas	31
Gambar 3.11	Dimensi Robot	32
Gambar 3.12	Rangkaian Utama <i>Driver</i> Motor EMS 30A H-Bridge	33

Gambar 3.13 Rangkaian Arduino antara <i>Driver</i> Motor dan Motor DC	33
Gambar 3.14 Bentuk Fisik dari <i>Valve Pneumatic</i>	34
Gambar 3.15 Rangkaian <i>Valve Pneumatic</i> dengan Rangkaian TIP31	34
Gambar 3.16 Rangkaian Minimum Sistem Arduino	35
Gambar 3.17 Rangkaian Catu Daya	37
Gambar 3.18 Tampilan GUI Sistem Kinematika Robot SCARA	38
Gambar 3.19 Trigonometri Sisi Atas	41
Gambar 3.20 Sistem Secara Keseluruhan	43
 Gambar 4.1 Grafik Pengujian Koordinat X	 52
Gambar 4.2 Grafik Pengujian Koordinat Y	54
Gambar 4.3 Grafik Pengujian <i>Joint Shoulder</i>	56
Gambar 4.4 Grafik Pengujian <i>Joint Elbow</i>	58
Gambar 4.5 Perbandingan Gambaran Robot SCARA	59
Gambar 4.6 Perbandingan Gambaran Robot SCARA	59
Gambar 4.7 Perbandingan Gambaran Robot SCARA	60
Gambar 4.8 Grafik Pengujian Akurasi Robot Secara Keseluruhan	61
Gambar 4.9 Simulasi Robot Lengan Menggunakan GUI	63

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. Besar objek yang dapat dibawa oleh *gripper* robot tidak dapat diubah.
4. Komunikasi antara Processing IDE dan *hardware* menggunakan komunikasi serial yang dihubungkan dengan kabel USB.

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, transformator AC 5A, *Converter* AC to DC, modul DC-to-DC *converter* LM2596, IC TIP 31, multimeter, *valve pneumatic*, *Driver* Motor EMS 30A dan catu daya AC 220 Volt. 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.

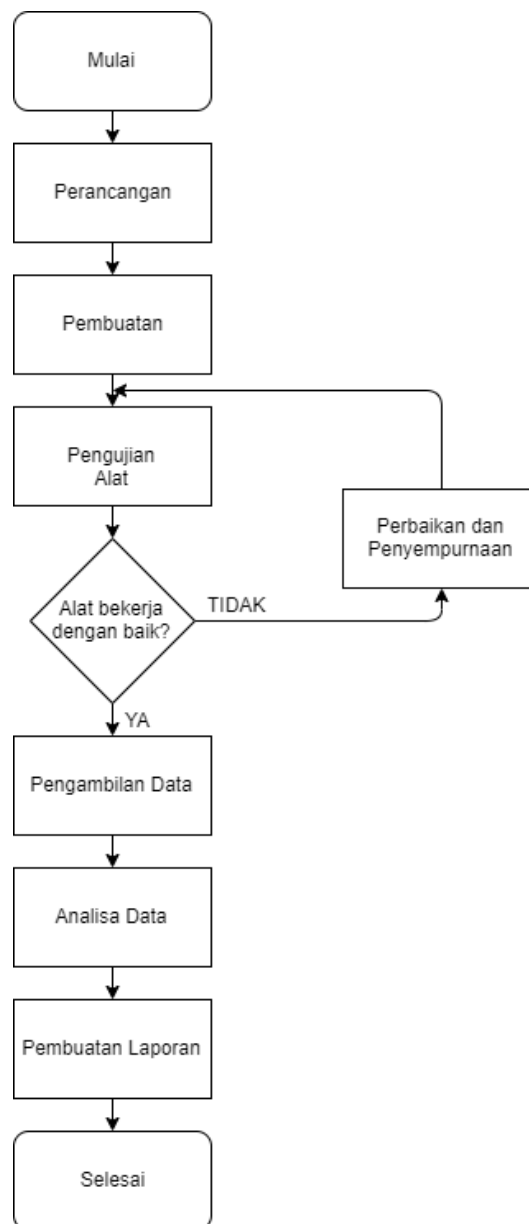
4. Metode Penelitian

Metode penelitian yang dilakukan meliputi: Pengujian tegangan DC to DC *converter*, pengujian motor DC, pengujian *Driver* Motor DC EMS 30A, pengujian rangkaian *switching valve pneumatic*, pengujian kinematika balik, pengujian

Graphical User Interface (GUI) pada Processing IDE, 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. Gambar 1.1 merupakan diagram alir prosedur penelitian.



Gambar 1.1: Diagram Alir 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 tegangan DC to DC *converter*, pengujian motor DC, pengujian *Driver Motor DC EMS 30A*, pengujian rangkaian *switching valve pneumatic*, pengujian kinematika balik, pengujian *Graphical User Interface* (GUI) pada Processing IDE, serta pengambilan data dan analisa data hingga pada pengujian 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

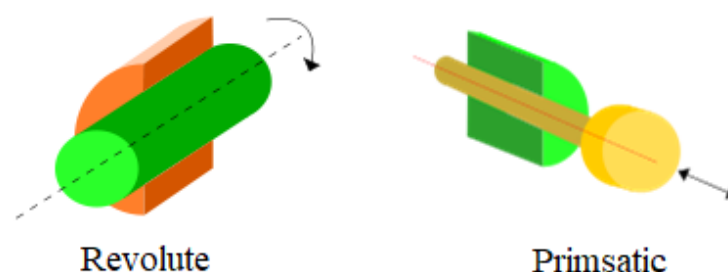
LANDASAN TEORI

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, 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). Gambar 2.1 merupakan jenis-jenis dari *joint*.



Gambar 2.1: Jenis-Jenis *Joint*

2.1.1 *Degree of Freedom*

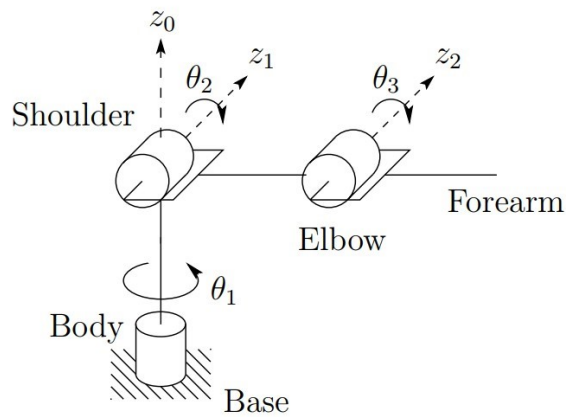
Degree of Freedom (DOF) merupakan sebuah konfigurasi yang dapat meminimalkan spesifikasi dengan menggunakan parameter yang dapat menyatakan posisi suatu sistem pada setiap saat. Umumnya robot lengan mempunyai paling sedikit enam independen derajat kebebasan, tiga derajat kebebasan untuk translasi dan 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)*

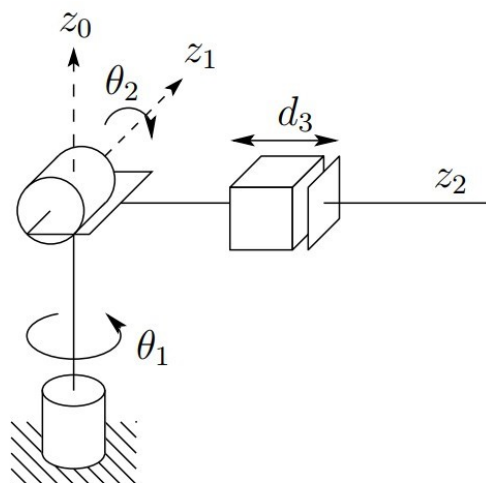
Articulated manipulator ini pada dasarnya mempunyai jenis *revolute joint* pada ketiga *joint* robot lengan (*base, shoulder, elbow*). Dengan konfigurasi ini, robot lengan dengan konfigurasi *Articulated* dapat memiliki variasi DOF yang banyak. DOF yang dapat dihasilkan dengan robot lengan dengan konfigurasi 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. Gambar 2.2 merupakan konfigurasi dari *Articulated*.



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*. Gambar 2.3 merupakan struktur dari konfigurasi *Spherical*.

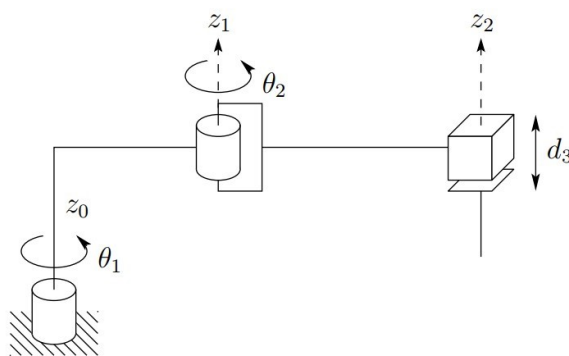


Gambar 2.3: Struktur dari Konfigurasi *Spherical*

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*

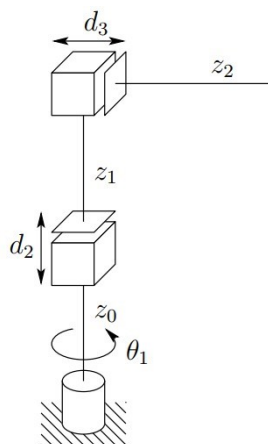
prismatic sama seperti konfigurasi *Spherical*. Meskipun SCARA memiliki struktur *joint revolute – revolute – prismatic* (RRP) sama seperti konfigurasi yang dimiliki *Spherical*, struktur ini sedikit berbeda dengan konfigurasi *Spherical* dari tampilannya maupun dari jarak *workspace* nya. Tidak seperti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



Gambar 2.4: Struktur dari Konfigurasi SCARA

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

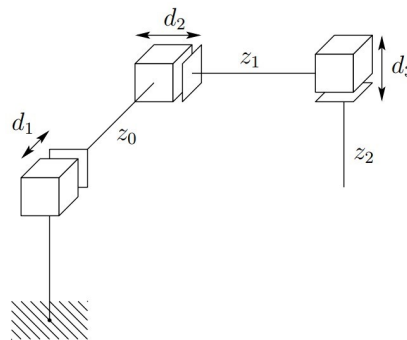
Konfigurasi *Cylindrical* merupakan konfigurasi yang mempunyai satu buah *joint revolute* dan dua buah *joint prismatic*. *Joint revolute* menghasilkan pergerakan rotasi di *base*, sementara *joint prismatic* berada di bagian *shoulder* dan *elbow*. Gambar 2.5 merupakan struktur dari konfigurasi *Cylindrical*.



Gambar 2.5: Struktur dari Konfigurasi *Cylindrical*

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

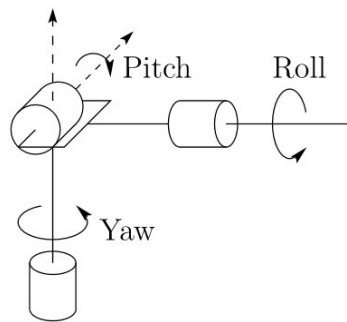
Konfigurasi *Cartesian* mempunyai tiga buah *joint prismatic*. Variabel *joint* dari konfigurasi *prismatic* adalah koordinat *Cartesian* dari *end-effector* dengan memperhatikan letak *base* dari robot lengan. Kinematika dari jenis konfigurasi ini adalah yang paling sederhana dari semua konfigurasi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



Gambar 2.6: Struktur dari Konfigurasi *Cartesian*

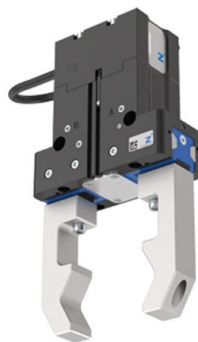
2.1.3 Wrist dan *End-effector*

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*. Gambar 2.7 merupakan struktur dari konfigurasi *Spherical Wrist*.



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. Gambar 2.8 merupakan bentuk dari *end-effector*.

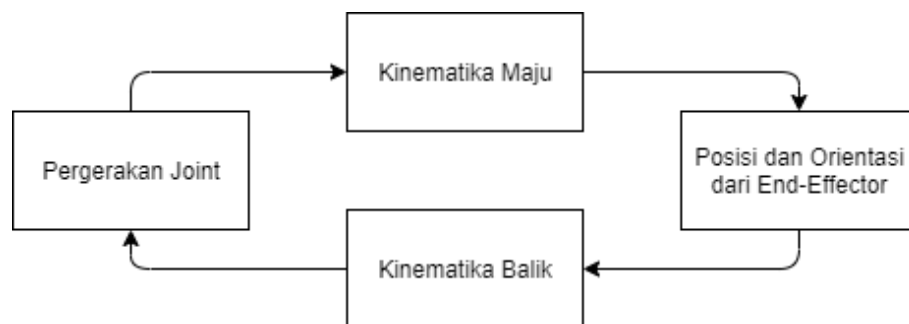


Gambar 2.8: Struktur *End-Effector*

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 yang tergantung dari tujuan dari setiap robot. Kinematika yang dijelaskan pada penelitian ini adalah kinematika yang khusus mempelajari dan menganalisa pergerakan robot lengan.

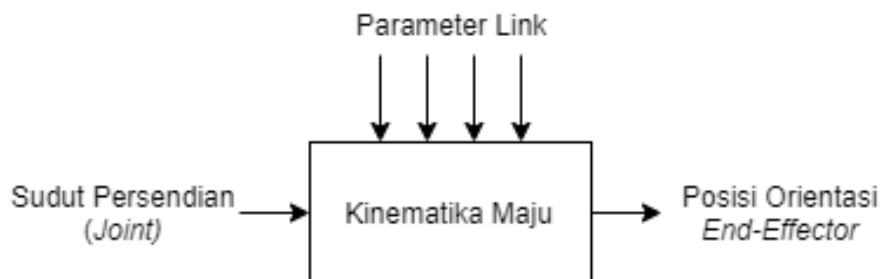
Pada kinematika robot terdapat dua buah pembahasan kinematika. Pembahasan pertama adalah kinematika maju yang merupakan proses menghitung orientasi dan posisi dari *end-effector* berdasarkan nilai sudut pada masing-masing *joint*. Sedangkan kinematika balik sebaliknya dari kinematika maju, diberikan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



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 . Gambar 2.10 merupakan proses dari kinematika maju.

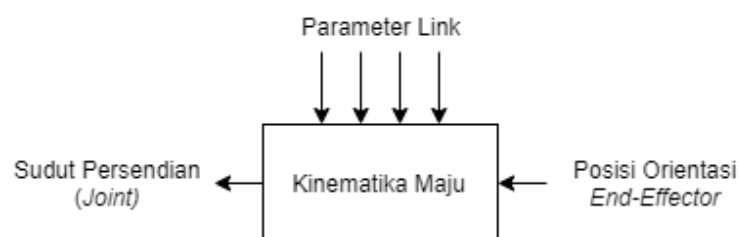


Gambar 2.10: Kinematika Maju

2.2.2 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*. 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 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. Gambar 2.11 merupakan proses dari kinematika balik.



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

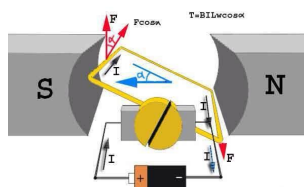
putaranya searah jarum jam (*clock wise*) atau berkebalikan dengan arah jarum jam (*counter clock wise*) bergantung dengan kutub atau polaritas dari catu daya yang diberikan pada motor DC. Gambar ?? merupakan bentuk fisik dari motor DC.



Gambar 2.12: Bentuk Fisik Motor DC

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. Gambar 2.13 merupakan prinsip kerja dari motor DC.



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 kekurangannya 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 . Gambar 2.14 merupakan bentuk fisik dari regulator *Buck* LM2596.

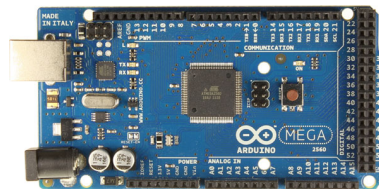


Gambar 2.14: Regulator *Buck* LM2596

2.5 Arduino Mega 2560

Arduino Mega 2560 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 Modu-*

lation (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 ?? merupakan bentuk fisik dari Arduino Mega 2560.



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, atau baterai dengan nilai tegangan sesuai dengan spesifikasi pada Arduino Mega tersebut. Tabel 2.1 merupakan spesifikasi dari Arduino Mega 2560.

Tabel 2.1: Spesifikasi Arduino Mega 2560

Keterangan	Nilai
Mikrokontroler	Atmega2560
Tegangan Operasional	5 Volt
Tegangan Masukan	5-12 Volt
Pin Digital I/O	54 Buah
Pin PWM	15 Buah
Arus I/O	20 mA
Memori Flash	256 KB
Kecepatan Clock	16 Mhz
Dimensi	101.52 x 53.3 mm

2.6 Driver Motor H-Bridge EMS 30A

Driver Motor H-Bridge EMS 30A 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 H-Bridge* EMS 30A. Umumnya *driver* motor memiliki beberapa jenis tergantung dari spesifikasi dari motor DC yang digunakan. Salah satu jenis *driver* motor adalah *Driver Motor H-Bridge* EMS 30A. Gambar 3.12 merupakan bentuk fisik dari *Driver Motor H-Bridge* EMS 30A.



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

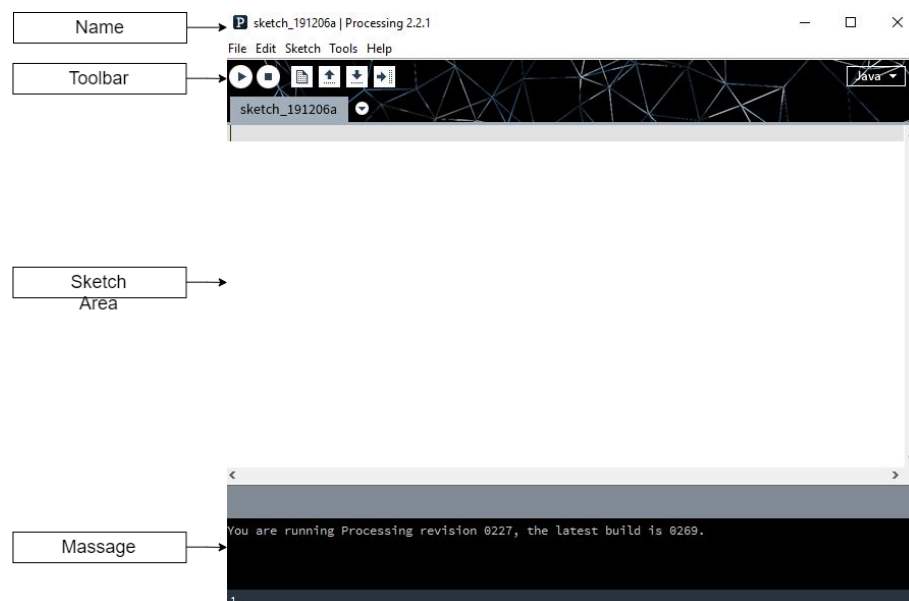
Driver Motor EMS 30A H-Bridge merupakan *driver* motor yang dapat mengoperasikan sebuah Motor DC dengan batasan arus hingga 30 Ampere. Driver ini memiliki 10 buah pin data yang dapat dihubungkan dengan sebuah mikrokontroler. Tabel 2.2 merupakan alokasi pin pada *Driver Motor H-Bridge* EMS 30A..

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 diberi logika HIGH untuk mengaktifkan half H-Bridge 1, diberi logika LOW untuk menonaktifkannya
4	MEN2	I/O	Pin enable untuk output MOUT2 diberi logika HIGH untuk mengaktifkan half H-Bridge 2, 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. Gambar 2.17 merupakan tampilan pada Processing IDE.



Gambar 2.17: Tampilan 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

"*height*". Untuk objek yang ukurannya tergantung pada layar, selalu gunakan variabel lebar dan tinggi, bukan angka. Ini mencegah masalah saat parameter fungsi `size()` diubah.

2.7.1.2 Transformasi Bentuk dalam 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 2.3 adalah contoh penulisan *syntax* dari Transformasi.






Tabel 2.3: *Syntax* Transformasi

No	Syntax	Keterangan
1	<code>Translate(x,y)</code>	x = transisi kiri/kanan 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 2.4 merupakan contoh dari beberapa *syntax* shape.

Tabel 2.4: *Syntax Shape*

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

2.7.1.4 Akses Koordinat Mouse

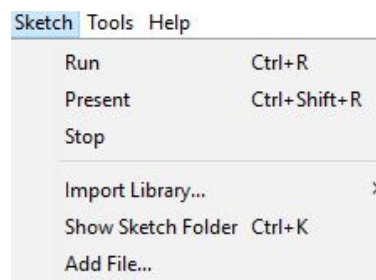
Processing IDE mempunyai fungsi khusus yaitu dapat melacak posisi mouse baik secara vertical maupun horizontal. Namun, Processing IDE hanya dapat melacak posisi mouse saat sketch dijalankan saja. Nilai default mouseX dan mouseY adalah 0, jadi 0 akan dikembalikan sampai mouse bergerak di depan jendela sketsa. Setelah mouse bergerak menjauh dari jendela, mouseX akan terus melaporkan posisi terakhirnya. Tabel 2.5 merupakan contoh penulisan *syntax* dari mouse.

Tabel 2.5: *Syntax Koordinat Mouse*

No	Syntax	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*. Gambar 2.18 merupakan *toolbar* dari *sketch*.

**Gambar 2.18:** *Toolbar Sketch*

2.7.2 *Library untuk Processing IDE*

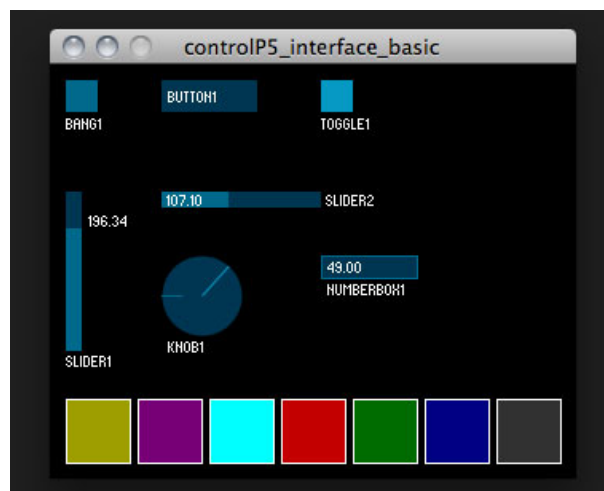
2.7.2.1 *Memuat Huruf (Font) pada Processing IDE*

Pfont adalah *class* yang digunakan untuk penggunaan *font* pada Processing IDE. Untuk membuat *font* yang akan digunakan dengan Processing IDE, diawali 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. Gambar 2.19 merupakan beberapa contoh gambaran dari ControlP5.



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 ?? merupakan beberapa fungsi dari *tools* yang ada pada ControlP5.

Tabel 2.6: Fungsi dari *tools* ControlP5

No	ControlP5	Fungsi
1	Bang	Memicu sebuah tindakan ketika ditekan <code>controlP5.addBang("bang1");</code>
2	Button	Memicu sebuah tindakan ketika sudah dilepas <code>controlP5.addButton("button1");</code>
3	Toggle	Memiliki dua status. TRUE memiliki nilai 1 dan FALSE memiliki nilai 0 <code>controlP5.addToggle("toggle1")</code>
4	Slider	Mengatur nilai dengan cara menggeser secara horisontal maupun vertikal <code>controlP5.addSlider("slider2");</code>
5	Knob	Mengatur sebuah nilai dengan tombol putaran 360° <code>controlP5.addKnob("knob1");</code>
6	NumberBox	Kotak yang menampilkan angka dan dapat diubah dengan klik di dalam kotak <code>controlP5.addNumberbox("numberbox1");</code>

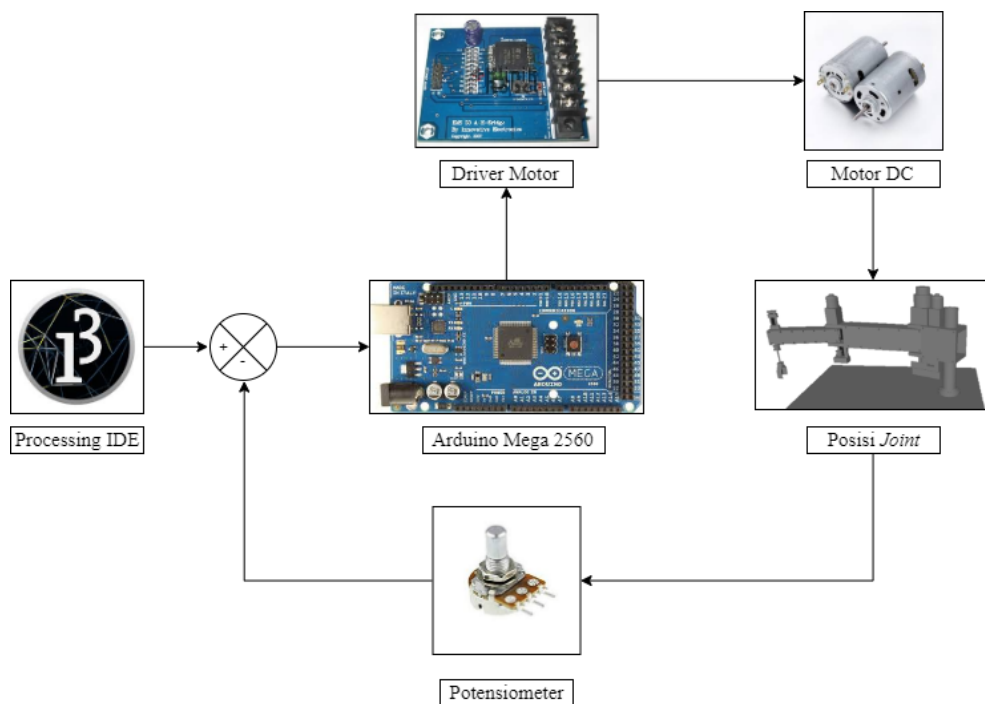
BAB III

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, perancangan GUI, 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* sedangkan pada bagian *input* yang berasal dari GUI yang dibuat pada Processing IDE mengirimkan sebuah koordinat yang digunakan untuk menentukan pergerakan robot berdasarkan fungsi *inverse kinematic*. Gambar 3.1 merupakan diagram blok sistem secara keseluruhan.



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 24 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 pada rangkaian *switch* yang menggunakan IC TIP31 yang berfungsi sebagai kontrol dari *End-Effector* yang dioperasikan menggunakan tekanan udara yang dikontrol oleh *valve pneumatic*.

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

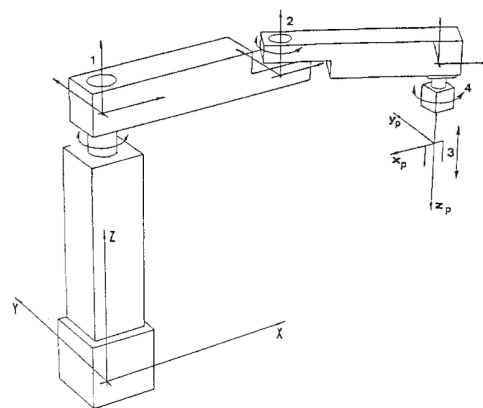
3.2 Perancangan Perangkat Keras

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

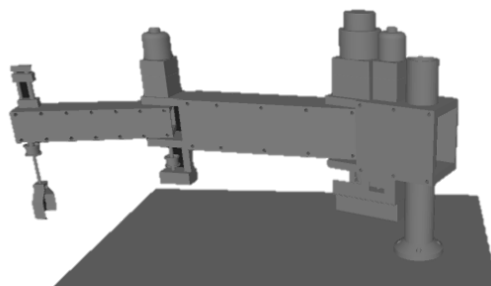
3.2.1 Sistem Mekanis

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 *Car-*

tesian. Pada Kerja Praktik 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 harus proporsional dari panjang setiap lengan, lebar serta tinggi robot. Gambar 3.2 merupakan *free body* dari robot SCARA. 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

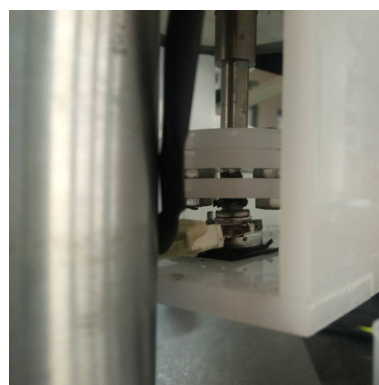
Robot SCARA merupakan robot yang memiliki tiga buah derajat kebebasan (DOF) yang terletak pada *shoulder*, *elbow*, dan pada *end-effector*. Seluruh derajat kebebasan menggunakan sebuah motor DC yang didalamnya terdapat *gear box* untuk motor DC yang ada pada *shoulder* dan *elbow* untuk melakukan pergerakan. Sedangkan motor DC pada *end-effector* dibantu oleh sebuah *belt* untuk menyalurkan putaran dari motor yang terletak pada *shoulder*. Pergerakan pada masing-masing *joint* me-

miliki 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

No	Keterangan	Nilai
1	Main arm length	360 mm
2	Fore arm length	290 mm
3	Shoulder movement	180 °
4	Elbow movement	200 °
5	Wrist rotation	360 °
6	Up & down movement	150 mm
7	Maximum tip velocity	3.0 kg

Desain pada *arm manipulator robot* SCARA berbahan besi dengan tebal 2 mm dengan tiga derajat kebebasan yang meliputi bagian *shoulder*, *elbow* serta *end-effector*. Desain 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 saluran udara yang berfungsi untuk mentransformasikan tekanan udara untuk pergerakan vertikal 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

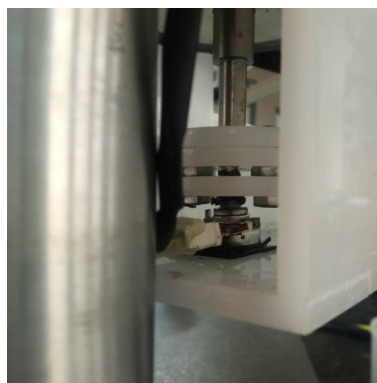
Pada *arm manipulator robot* SCARA menggunakan penggerak berupa motor DC 12 Volt dengan *gearbox* sehingga mampu mengangkat beban berat karena torsi

pada motor bertambah besar. Motor DC dikontrol oleh *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

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

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* secara otomatis ikut bergerak dan kemudian mengirimkan nilai data analog ke Arduino Mega. Data yang berhasil diterima oleh Arduino Mega 2560 kemudian diolah dan mendapatkan hasil berupa besar sudut pada pergerakan *joint* tersebut. 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 pada robot *end-effector* merupakan gerak translasi pada sumbu Y. 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 *gripper end-effector* dapat membuka dan menutup karena sebuah sistem mekanik yang telah ada di dalamnya. Selain dari pergerakan translasi, pergerakan pada *end-effector* juga terdapat rotasi. Pergerakan ini dilakukan oleh satu buah motor DC yang ditempatkan pada bagian *shoulder* 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* pada robot SCARA yang digunakan.



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 220 Volt. 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 fisik dari 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.



Gambar 3.9: Rancangan Tampak Samping



Gambar 3.10: Rancangan Tampak Atas



Gambar 3.11: Dimensi Robot

3.2.2 Rangkaian Elektronika

3.2.2.1 Rangkaian Motor DC

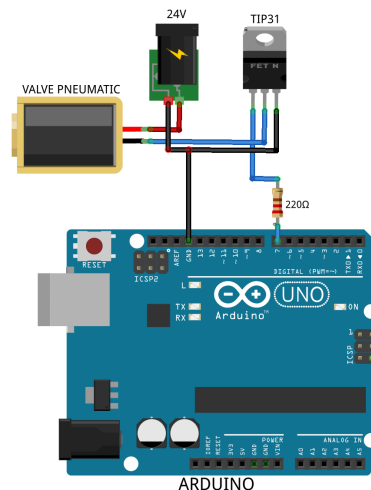
Rancangan kendali pada *arm manipulator robot* SCARA ini menggunakan tiga buah motor DC dengan masing – masing dilengkapi dengan *gearbox* untuk memperkuat torsi yang dihasilkan oleh motor DC. Pada *gearbox* masing – masing motor DC diberikan sensor *potensiometer* sebagai *feedback* untuk memberikan posisi motor DC pada keseluruhan sistem. Motor DC diletakkan pada *shoulder* untuk *end-effector* satu buah yang dihubungkan melalui *belt*, pada *shoulder* satu buah, dan pada *elbow* satu buah. Ketiga motor DC tersebut masing-masing menggunakan *Driver Motor* EMS 30A H-Bridge dalam sistem kerjanya. Pada masing – masing motor DC membutuhkan catu daya 12 Volt DC. Rangkaian *driver* motor mendapat sumber tegangan DC 12V untuk disalurkan ke pada motor DC dan 5 Volt untuk kerja dari driver motor sendiri. 12 Volt didapat dari tegangan keluaran yang dihasilkan oleh regulator *Buck* yang berseumber dari tegangan DC 24 Volt setelah dilakukan *converter* AC to DC. Ragkaian utama *Driver Motor* EMS 30A H-Bridge ditunjukkan pada Gambar 3.12 dan pada Gambar 3.13 merupakan rangkaian antara motor DC, driver motor, dan Arduino Mega.

End-Effector menggunakan tekanan udara dalam melakukan pergerakannya. Tekanan udara ini dikontrol menggunakan sebuah *valve relay*. *Valve relay* yang digunakan dapat mengontrol tekanan udara hingga delapan bar. *Valve relay* dapat bekerja pada tegangan DC 24 Volt. Tegangan 24 Volt pada *valve relay* didapat dari keluaran dari rangkaian AC-DC yang dilakukan oleh dioda *bridge* dengan masukan awalnya adalah tegangan AC 24 Volt yang diberikan oleh sebuah transformator. Dengan besaran tegangan 24 Volt maka sebuah arduino tidak dapat mengontrolnya. Oleh karena itu, diberi sebuah rangkaian pembantu yang prinsipinya bekerja seperti saklar. Rangkaian tersebut diotaki oleh IC TIP31 yang nantinya akan menerima sinyal data digital

dari Arduino Mega 2560 dan akan membuka jalur untuk tegangan 24 Volt. Gambar 3.14 merupakan bentuk fisik dari *valve relay* yang digunakan dan Gambar 3.15 merupakan rangkaian dari *valve pneumatic* dengan rangkaian TIP31.



Gambar 3.14: Bentuk Fisik dari *Valve Pneumatic*

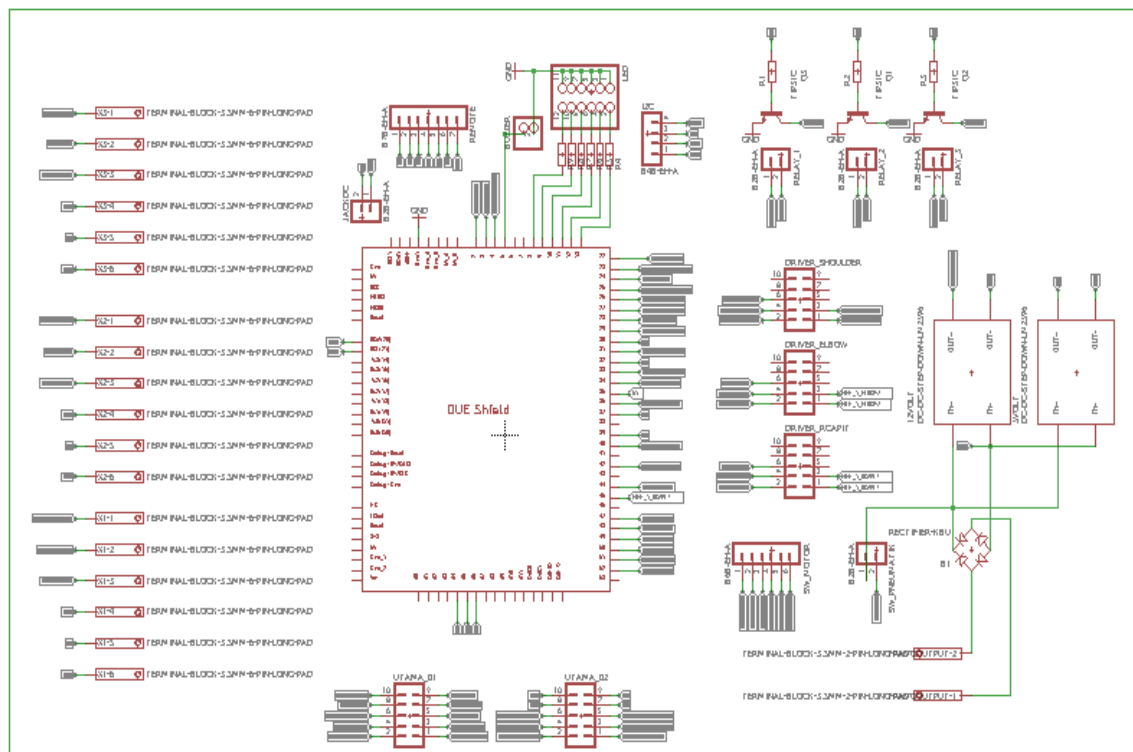


Gambar 3.15: Rangkaian *Valve Pneumatic* dengan Rangkaian TIP31

3.2.2.3 Rangkaian Minimum Sistem Arduino

Arduino Mega 2560 digunakan sebagai mikrokontroler utama untuk mengendalikan seluruh sistem pada *arm manipulator robot SCARA*. Arduino 2560 memiliki banyak pin keluaran dan masukan digital dan analog yang dapat digunakan sebagai pengendali fungsi – fungsi dari setiap komponen. Pin pada Arduino 2560 dapat mencukupi kebutuhan masukan dan keluaran untuk sistem kerja robot. Pin analog yang pada Arduino digunakan untuk menerima *feedback* masukkan dari potensio yang ada pada setiap motor robot. Beberapa pin digital pada Arduino Mega 2560 juga memiliki fungsi lain yaitu sebagai pin *Pulse With Modulation* (PWM) yang dapat digunakan

sebagai keluaran analog sehingga dapat digunakan untuk mengatur nilai tegangan kaluaran dari Arduino 2560. Pin PWM yang cukup digunakan untuk kontrol kecepatan pada *driver* motor. Selain itu pin PWM pada Arduino Mega 2560 digunakan untuk memberikan kontrol direksi pada *driver* motor untuk memberikan masukan arah putar kanan ataupun putar kiri. Gambar 3.16 merupakan rangkaian sistem minimum Arduino Mega 2560 pada robot SCARA ini. Tabel 3.3 merupakan fungsi dari masing-masing pin yang ada pada Arduino Mega 2560.



Gambar 3.16: Rangkaian Minimum Sistem Arduino

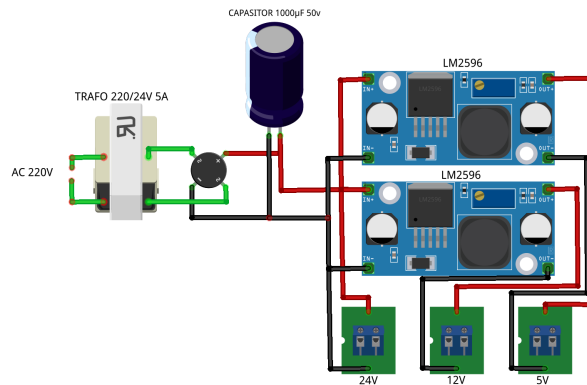
Tabel 3.3: Pin pada Arduino Mega 2560

No	Pin Arduino Mega 2560	Fungsi
1	A1	Feedback potensiometer shoulder
2	A2	Feedback potensiometer elbow
3	A3	Feedback potensiometer end-effector
4	D16, D18	Kontrol aktif driver motor shoulder
4	D20, D22	Kontrol driver motor shoulder
5	D24, D26	Kontrol aktif driver motor elbow
6	D28, D30	Kontrol driver motor elbow
7	D32, D34	Kontrol aktif driver motor end-effector
8	D36, D38	Kontrol driver motor end-effector
9	D4	Kontrol PWM driver motor shoulder
10	D5	Kontrol PWM driver motor elbow
11	D6	Kontrol PWM driver motor end-effector
12	D7	Kontrol valve relay naik
13	D8	Kontrol valve relay turun
14	D9	Kontrol valve relay buka-tutup
15	D15	Kontrol LED Shoulder aktif high
16	D17	Kontrol LED elbow aktif high
17	D19	Kontrol LED end-effector naik aktif high
18	D21	Kontrol LED end-effector turun aktif high
19	D23	Kontrol LED end-effector buka-tutup aktif high
20	D25	Buzzer aktif high

3.2.2.4 Rangkaian Catu Daya

Rangkaian catu daya merupakan hal yang sangat penting dalam sebuah sistem. Pada *arm manipulator robot SCARA* terdapat tiga buah nilai tegangan *supply* yang berbeda. Tegangan yang ditunjukkan untuk Arduino Mega 2560 dan beberapa sensor membutuhkan tegangan 5 Volt, motor DC membutuhkan tegangan 12 Volt dan *valve pneumatic* membutuhkan tegangan 24 Volt. Catu daya diawali dengan tegangan AC 220 Volt dari listrik PLN. Tegangan tersebut kemudian diturunkan menggunakan sebuah trafo 3A menjadi 24 Volt AC. Komponen yang digunakan dalam rangkaian

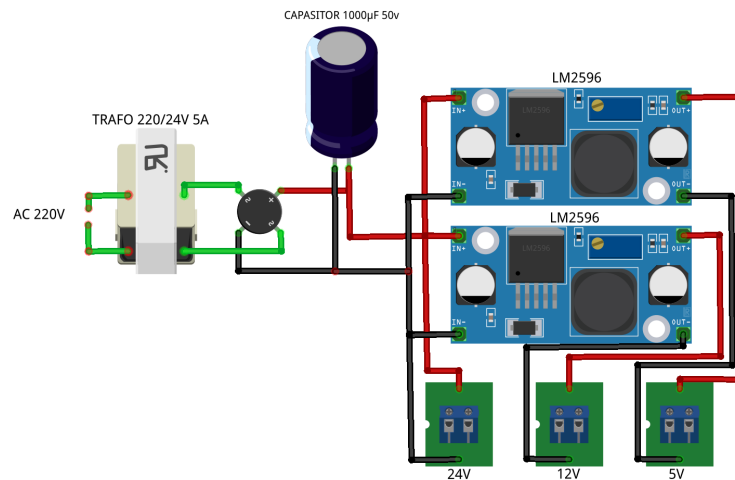
merupakan komponen yang membutuhkan tegangan DC, maka tegangan 24 Volt AC diubah menjadi 24 Volt DC menggunakan rangkaian dioda *bridge*. Tegangan 24 Volt DC diarahkan menuju dua buah regulator *Buck* LM2596 yang masing-masing menghasilkan tegangan 12 Volt dan 5 Volt. Gambar 3.17 merupakan rangkaian catu daya secara keseluruhan.



Gambar 3.17: Rangkaian Catu Daya

3.3 Perancangan Perangkat Lunak

Perangkat lunak yang digunakan merupakan *software* Processing IDE. Processing IDE diprogram agar menghasilkan sebuah GUI yang cocok sesuai dengan fungsi robot SCARA. Di dalam GUI terdapat dua bagian yang diantaranya merupakan bagian kontrol, dan bagian *display*. Pada bagian kontrol pada GUI berfungsi untuk mengontrol mulai dari pergerakan dan posisi dari robot SCARA. Sedangkan pada bagian penampil, GUI menampilkan nilai dari sudut, posisi, serta animasi terkait robot SCARA. Gambar 3.18 merupakan tampilan dari GUI secara keseluruhan. Tabel 3.4 menyajikan setiap komponen yang ditampilkan di dalam GUI processing IDE.



Gambar 3.18: Tampilan GUI Sistem Kinematika Robot SCARA

Tabel 3.4: Keterangan Tampilan pada GUI Robot SCARA

No	Pin Arduino Mega 2560	Fungsi
1	A1	Feedback potensiometer shoulder
2	A2	Feedback potensiometer elbow
3	A3	Feedback potensiometer end-effector
4	D16, D18	Kontrol aktif driver motor shoulder
4	D20, D22	Kontrol driver motor shoulder

3.3.1 ControlP5

ControlP5 merupakan salah satu *library* yang berguna dalam membuat sebuah GUI. Dalam *library* yang disediakan terdapat banyak pilihan terkait kontrol sistem dengan berbagai jenis pilihan. Kontrol sistem yang disediakan tersedia dua pilihan yaitu untuk mengendalikan dan untuk menampilkan sebuah data. Masing-masing kontrol sistem ini dapat digunakan dengan cara memanggilnya pada program yang ditulis.

Pada perancangan kinematika robot SCARA ini, menggunakan empat buah kontrol sistem. Empat kontrol sistem yang digunakan merupakan jenis kontrol sistem yang berguna untuk memberikan sebuah data yang dikirimkan pada Arduino Mega 2560. Kontrol sistem tersebut diantaranya:

1. *Slider Control*

Slider Control berupa tampilan kontrol GUI yang menggunakan sistem geser

dalam memberikan data. Data memiliki batasan atas dan batas bawah yang dimasukkan dalam program. Keuntungan menggunakan kontrol sistem jenis ini adalah mudahnya dalam memberikan sebuah data yang berbeda.

2. *Textfield Control*

Textfield control berupa tampilan kontrol GUI yang menggunakan masukan nilai data sesuai apa yang dituliskan atau diketikkan secara langsung. Kelebihan menggunakan kontrol sistem jenis ini karena dapat memasukkan nilai data lebih spesifik secara langsung sesuai keinginan.

3. *Bang Control*

Bang control Berbeda dengan kontrol sistem sebelumnya, *bang control* berupa kotak yang bekerja pada saat mulai ditekan. Saat *bang* mulai ditekan data akan dikirimkan sesuai dengan nilai yang sudah dimasukkan kedalam program.

4. *Toggle Control*

Toggle control *Toggle control* memiliki sistem kontrol seperti saklar on-off. Pada saat ditekan maka *toggle control* menyimpan data berupa kondisi pertama dan berubah saat ditekan kembali.

3.3.2 *Shape*

Shape pada GUI berfungsi sebagai penampil animasi tiga dimensi dari robot SCARA. Robot SCARA dapat ditampilkan dalam berbagai ukuran, letak dan juga dapat bergerak sesuai dengan data yang diberikan. Pergerakan dari robot SCARA merupakan implementasi dari wujud aslinya dengan pergerakan yang sama.

Dalam pengoperasiannya, desain dari robot SCARA yang ditampilkan harus dimasukkan ke dalam folder yang sama dengan program processing IDE. File yang dapat ditampilkan merupakan file dengan jenis obj. yang berarti file objek. Pada program robot SCARA file dari dimensi tiga dari robot SCARA memiliki tiga buah file dimana ketiganya adalah *base*, *shoulder*, dan *elbow*.

3.4 Sistem Kinematika

Persamaan kinematika terbagi dua, yaitu kinematika maju dan kinematika balik. Kinematika maju digunakan untuk menentukan posisi dan orientasi *end-effector*

apabila variabel sudut *joint*-nya telah diketahui. Kinematika balik digunakan untuk mencari *joint* robot dalam menentukan posisi dan orientasi dari *end-effector*.

3.4.1 Prinsip Kerja Kinematika Maju

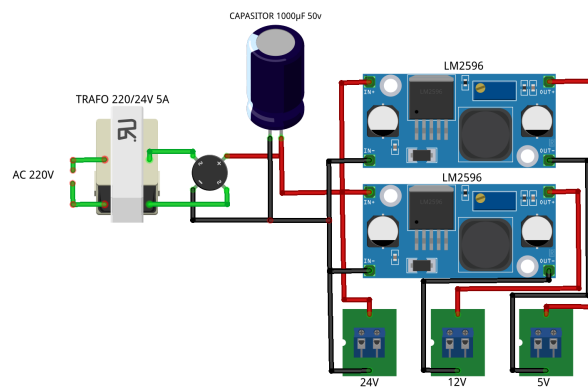
Metode Denavit-Hartenberg merupakan metode yang menggabungkan proses perhitungan rotasi dan translasi menjadi sebuah matriks yang menyertakan nilai-nilai sudut putar dan jarak sendi dari sebuah lengan robot. Dalam beberapa aplikasi, metode Denavit-Hartenberg umumnya digunakan dalam perhitungan *Forward Kinematics*. Dalam penelitian ini dirancang aplikasi yang menggunakan metode Denavit-Hartenberg untuk menghitung sudut-sudut tiap sendi pada *Invers Kinematics* untuk menggerakkan sebuah lengan robot. Matrik Denavit-Hartenberg yang berisi perhitungan rotasi dan translasi digunakan untuk mendapatkan nilai-nilai sudut untuk menggerakkan tiap motor sendi. Empat Aturan Frame Denavit-Hartenberg yaitu Sumbu Z harus menjadi sumbu rotasi atau translasi dari sebuah joint. Sumbu X harus tegak lurus dari sumbu Z frame sebelumnya. Sumbu X harus memotong atau menyilang dari sumbu Z frame sebelumnya. Sumbu Y harus digambarkan sesuai dengan aturan tangan kanan setelah sumbu X dan sumbu Z setiap frame digambarkan.

3.4.2 Prinsip Kerja Kinematika Balik

Kinematika balik adalah perhitungan untuk mencari variabel sudut (*joint*) robot dalam menentukan posisi dan orientasi dari *end-effector*. Dalam menentukan koordinat *end-effector*, kinematika balik harus disesuaikan dengan batas area kerja (*workspace*) dari jangkauan robot. Penyelesaian kinematika balik ini dapat diselesaikan dengan menggunakan persamaan kinematika balik yang di dalamnya menggunakan hukum *pythagoras* dan aturan *cosinus*. Secara garis besar metode *inverse kinematic* 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.

Analisis persamaan kinematik dapat diselesaikan dengan cara yang paling dasar yaitu menggunakan trigonometri dengan bantuan grafik. Pada penelitian ini ka-

rena menggunakan sebuah GUI yang didalamnya terdapat animasi dari bentuk fisik robot SCARA maka koordinat didapat dari Processing IDE. Robot SCARA yang terdapat di dalam Processing IDE menggunakan skala tertentu agar tetap sesuai dari dimensi aslinya. Penyelesaian kinematika dalam robot SCARA cukup diselesaikan menggunakan satu sisi, yaitu sisi atas (*top view*) dari struktur robot lengan. Pada sisi atas derajat sudut *joint shoulder*, dan sudut *joint elbow* dapat ditemukan. Gambar 3.19 merupakan gambaran untuk mendapatkan sebuah persamaan yang akan menghasilkan nilai sudut pada setiap *joint*.



Gambar 3.19: Trigonometri Sisi Atas

Pada robot SCARA, sudut dari *joint* yang dicari merupakan *joint* dari *shoulder* dan juga *elbow*. Kedua *joint* tersebut dapat ditemukan dengan melalui persamaan *pythagoras* dan juga hukum *cosinus*. Pada Gambar 3.19 terlihat bahwa sudut *shoulder* merupakan besar sudut diantara lengan *shoulder* dan juga sumbu x dan sudut *elbow* merupakan besar sudut antara lengan *elbow* dengan garis bantu dari lengan *shoulder*. Keduanya dapat ditentukan besar nilainya melalui beberapa persamaan dari *cosinus* dan juga hukum *pythagoras*. l_1 merupakan panjang lengan *shoulder* dan l_2 merupakan panjang lengan dari *elbow*. Serta θ_1 merupakan sudut dari *shoulder* dan θ_2 merupakan sudut dari *elbow*.

1. Dengan menggunakan hukum *cosinus*, didapatkan sebuah persamaan 3.1

$$(x^2 + y^2) = l_1^2 + l_2^2 - 2l_1l_2\cos(180 - \theta_2) \quad (3.1)$$

2. Pada persamaan 3.2 terdapat \cos yang dapat diubah sesuai dengan prinsip dari

hukum *cosinus* menjadi seperti pada persamaan 3.2

$$(x^2 + y^2) = l_1^2 + l_2^2 - 2l_1l_2\cos(\theta_2) \quad (3.2)$$

3. Inti persamaan akan mencari sebuah nilai dari θ_2 , maka untuk memudahkannya maka persamaan menjadi seperti pada persamaan 3.3

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \quad (3.3)$$

4. Nilai dari θ_2 dapat dengan mudah diketahui dengan melanjutkan seperti pada persamaan 3.4

$$\theta_2 = \arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \quad (3.4)$$

Dengan persamaan 3.4 maka nilai dari θ_2 atau sudut dari *elbow* dapat diketahui dengan cara memasukkan nilai posisi x , posisi y serta panjang dari *shoulder* dan *elbow* ke dalam persamaan. Nilai posisi x dan posisi y merupakan posisi akhir dari *end-effector*.

5. Dalam menentukan sudut lainnya yaitu sudut *shoulder* yang ditandai dengan simbol θ_1 menggunakan persamaan *cosinus* yang dituliskan pada persamaan 3.5

$$\frac{\sin(\beta)}{l_2} = \frac{\sin(\gamma)}{\sqrt{x^2 + y^2}}; \alpha = \arctan\left(\frac{y}{x}\right) \quad (3.5)$$

6. Pada persamaan 3.5 beberapa dapat diubah sesuai dengan hukum *cosinus* $\sin(\gamma) = \sin(180 - \theta_2) = \sin(\theta_2)$ dengan mengubah $\sin(\gamma)$ menjadi $\sin(\theta_2)$ maka akan persamaan menjadi seperti yang ditunjukkan pada persamaan 3.6

$$\beta = \arcsin\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2 + y^2}}\right) \quad (3.6)$$

7. Jika dilihat pada gambar 3.19 maka besar dari sudut *shoulder* yang ditandai dengan θ_1 yang artinya $\theta_1 = \beta + \alpha$ maka dapat diselesaikan dengan persamaan 3.7

$$\theta_1 = \arcsin\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2 + y^2}}\right) + \arctan\left(\frac{y}{x}\right) \quad (3.7)$$

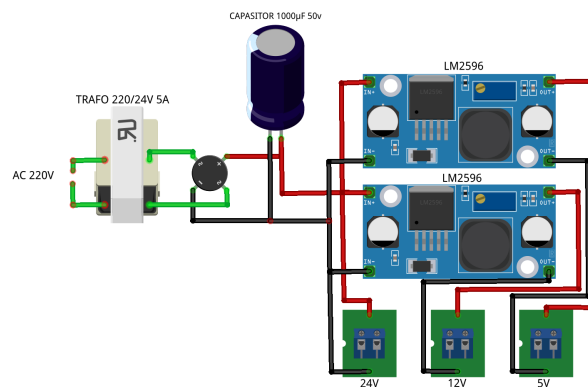
8. Dengan penyelesaian seperti pada persamaan 3.7 besar sudut dari *shoulder* da-

pat diketahui dengan memasukkan nilai panjang *elbow*, sudut *elbow* dan juga posisi dari *end-effector*.

Dalam penelitian kinematika robot SCARA ini, segala perhitungan kinematika dilakukan di dalam program processing IDE. Di dalam processing IDE terdapat animasi dari bentuk asli robot SCARA dengan dimensi yang sama hanya saja ditampilkan dalam bentuk berskala. Pada posisi *end-effector* didapat posisi x dan y yang diprogram pada dalam processing IDE. Dua buah data sudut yang didapat pada perhitungan merupakan sudut *shoulder* dan *elbow* kemudian dikirimkan kepada Arduino Mega 2560 untuk diproses dan diimplementasikan secara langsung pada robot lengan.

3.5 Perancangan Sistem Keseluruhan

Rancangan sistem keseluruhan merupakan gabungan dari perancangan perangkat keras dan perangkat lunak yang diintegrsikan sesuai dengan diagram blok sistem. Gambar 3.20 adalah gambaran tentang perancangan sistem secara keseluruhan. Tabel 3.5 menunjukkan keterangan setiap komponen yang ada di dalam sistem keseluruhan *arm manipulator* robot SCARA.



Gambar 3.20: Sistem Secara Keseluruhan

Tabel 3.5: Keterangan Sistem Keseluruhan

No	Keterangan
1	Robot Lengan SCARA
2	Box Panel
3	Arduino Mega 2560
4	Personal Computer
5	GUI Processing IDE
6	Workspace Robot SCARA
7	Kompresor
8	Objek

Processing IDE merupakan piranti yang digunakan sebagai masukan data. Masukan dari Processing IDE adalah hasil dari perhitungan kinematika maju dan atau kinematika balik yang dimasukkan melalui GUI. Objek berada dalam cakupan *workspace* yang telah ditentukan luasnya yang ditinjau dari cangkupan dari *end-effector*. GUI dari Processing IDE ini diproses menggunakan komputer personal.

Processing IDE juga menampilkan nilai derajat setiap *joint*, koordinat x dan y dari *end-effector*. Koordinat x dan y yang telah didapatkan dari processing IDE dioalah dalam perhitungan kinematika balik sehingga dapat ditemukan nilai besar derajat sudut setiap *joint*. Setelah nilai besar derajat sudut setiap *joint* didapatkan, Processing IDE akan mengirimkan nilai-nilai tersebut ke Arduino IDE melewati komunikasi Serial. Arduino Mega 2560 mengirimkan nilai-nilai tersebut sesuai dengan pin motor DC yang tersambung dengan Arduino Mega 2560 melalui *Driver Motor H-Bridge* EMS 30A, kemudian nilai-nilai *joint* tersebut dikirimkan Arduino Mega 2560 kembali ke Processing IDE melalui koneksi langsung menggunakan kabel USB.

Arduino Mega 2560 merupakan mikrokotroler yang menggerakkan setiap motor DC sesuai dengan besaran nilai yang telah didapatkan. Robot lengan SCARA akan mencapai koordinat x dan y dari objek, lalu mencengkram objek tersebut menggunakan *Gripper*.

BAB IV

PENGUJIAN DAN PEMBAHASAN

Pada bab pengujian dan pembahasan ini, penulis akan melakukan pengujian sistem kendali *arm manipulator* robot SCARA berdasarkan spesifikasi sistem yang telah dijelaskan pada bab sebelumnya. Tujuan pengujian ini adalah untuk membuktikan apakah sistem yang diimplementasikan telah memenuhi spesifikasi dan rancangan yang sudah direncanakan sebelumnya. Hasil dari pengujian akan dimanfaatkan untuk menyempurnakan kinerja dari sistem dan sekaligus digunakan dalam pengembangan sistem lebih lanjut.

Metode pengujian menggunakan dua macam metode, yaitu pengujian fungsionalitas dari setiap komponen dan pengujian sistem secara keseluruhan. Pengujian fungsionalitas digunakan untuk membuktikan apakah sistem yang diimplementasikan dapat memenuhi persyaratan dari fungsi operasional yang telah dirancang dan direncanakan sebelumnya. Sedangkan pengujian sistem secara keseluruhan bertujuan untuk memperoleh beberapa parameter yang dapat menunjukkan kemampuan dan keandalan dari sistem secara keseluruhan dalam menjalankan fungsi operasionalnya. Pada *sistem arm manipulator* robot SCARA dilakukan terlebih dahulu pengujian terhadap fungsional dari beberapa komponen seperti bagian *DC to DC converter*, arah gerakan motor DC, *feedback potentiometer*, fungsi rangkaian *switching* pada *valve pneumatic* dan keakuratan setiap *joint* untuk bergerak sesuai sudut yang diinginkan berdasarkan kinematika balik maupun kinematika maju dengan menggunakan kontrol dari GUI. Kemudian setelah pengujian fungsional terpenuhi maka dilakukan pengujian sistem secara keseluruhan untuk mengetahui keakuratan dan simulasi dari sistem *arm manipulator* robot SCARA.

4.1 Pengujian Fungsional

Pengujian fungsional digunakan untuk menguji bagian – bagian dari sistem yang terdiri dari *DC to DC converter*, arah gerakan motor DC, *feedback potentiometer*, fungsi rangkaian *switching* pada *valve pneumatic* dan keakuratan setiap *joint*, pengujian GUI Processing dan pengujian program.

4.1.1 Pengujian DC - to - DC Converter

Pengujian DC – to – DC *converter* dilakukan untuk mengetahui tegangan masukan pada Arduino Mega 2560, sensor *potensiometer*, motor DC dan juga sumber tegangan untuk *valve pneumatic*. Tegangan masukan dari catu daya utama sebesar 24 Volt DC yang nantinya dibagi ke tiga buah nilai tegangan. Tabel 4.1 merupakan tegangan keluaran DC – to - DC.

Tabel 4.1: Hasil Tegangan Keluaran Dari Tegangan DC-DC Converter

No	Tegangan Masukan (V)	Tegangan Keluaran Buck Converter (12 V)	Tegangan Keluaran Buck Converter (5 V)
1	12	12.1	4.9
2	14	12.1	4.9
3	16	12.1	4.9
4	18	12.1	4.9
5	20	12.1	4.9

Pada hasil yang ditunjukkan oleh Tabel 4.1 terlihat bahwa nilai tegangan telah sesuai dengan yang dibutuhkan. Pada saat mengubah besar tegangan keluaran yang dilakukan oleh Regulator *Buck* LM2596 dilakukan dengan cara memutar *potensiometer* yang terdapat pada Regulator *Buck*. Diputar searah dengan jarum jam sesuai hingga pada tegangan yang diinginkan.

4.1.2 Pengujian Motor DC

Pengujian motor DC dilakukan untuk mengetahui apakah motor DC dalam keadaan baik atau tidak. Pengujian dilakukan dengan memberikan tegangan kerja pada motor DC yang ada pada *shoulder*, *elbow*, dan juga *end-effector* yang nantinya diukur arus yang dihasilkan pada masing-masing motor DC. Tabel 4.2 merupakan hasil dari pengujian pada masing-masing motor DC.

Tabel 4.2: Hasil Pengujian Motor DC

No	Joint	Tegangan(V)	Arus(mA)
1	<i>Shoulder</i>	12.1	
2	<i>Elbow</i>	12.1	
3	<i>End-Effector</i>	12.1	

Pada hasil yang ditunjukkan oleh tabel 4.2 menunjukkan bahwa setiap motor DC mempunyai nilai arus yang berbeda-beda. Motor DC yang terletak pada *end-effector* merupakan motor DC yang menghasilkan arus paling besar. Hal ini disebabkan karena motor DC yang terpasang pada *end-effector* dibantu dengan bantuan *belt* untuk menyalurkan putaran pada *end-effector*. Dengan begitu penggunaan *belt* pada motor DC ini menyebabkan beban yang dikerjakan oleh motor DC pada *end-effector* menjadi lebih besar dari pada motor DC yang lain yang langsung menggerakkan pada masing-masing *joint*. Secara keseluruhan motor DC dapat dioperasikan dan masih dalam keadaan baik.

4.1.3 Pengujian Driver Motor H – Bridge

Pengujian *Driver Motor H – bridge* dilakukan untuk mengetahui keberfungsian dari *driver* motor apakah sesuai dengan perancangan atau tidak. Pada *driver* motor juga dilakukan pengujian untuk melihat direksi dari arah pergerakan motor DC dari *output driver* ketika diberikan masukan berupa sinyal *high* dan *low* pada arduino. Tabel 4.3 menunjukkan hasil dari pengujian *driver* motor H-Bridge.

Tabel 4.3: Hasil Pengujian *Driver* Motor H-Bridge

No	Joint	Sinyal Arduino		Kondisi	Arus (mA)
		MEN1	MEN2		
1	Shoulder	HIGH	HIGH	Tidak berputar	
		HIGH	LOW	Berputar searah putaran jam	
		LOW	HIGH	Berputar berlawanan arah jam	
		LOW	LOW	Tidak berputar	
2	Elbow	HIGH	HIGH	Tidak berputar	
		HIGH	LOW	Berputar searah putaran jam	
		LOW	HIGH	Berputar berlawanan arah jam	
		LOW	LOW	Tidak berputar	
3	End-Effector	HIGH	HIGH	Tidak berputar	
		HIGH	LOW	Berputar searah putaran jam	
		LOW	HIGH	Berputar berlawanan arah jam	
		LOW	LOW	Tidak berputar	

Pada *driver* motor H-Bridge EMS 30A sinyal digital *high* dan *low* dihubungkan pada pin MEN1 dan MEN2. Dari hasil pengujian *driver* motor H-Bridge seper-

ti yang ditunjukkan pada tabel 4.3 terlihat bahwa ketika sinyal *high* diberikan pada MEN1 dan *low* diberikan MEN2 maka pergerakan motor akan berputar searah dengan arah jarum jam dan sebaliknya jika diberikan *low* pada MEN1 dan *high* pada MEN2 maka arah pergerakan motor akan berlawanan arah. Pada tabel 4.3 juga terlihat bahwa nilai arus dapat dialirkan pada driver motor beragam dari 1 Ampere hingga 1.5 Ampere. Terlihat bahwa *driver* motor dapat mengoperasikan driver motor dengan baik.

4.1.4 Pengujian Nilai Analog Potensiometer

Pengujian nilai *analog potensiometer* berfungsi untuk mengetahui apakah *potensiometer* bekerja dengan baik dan nilai yang diberikan dalam keadaan yang normal. Pada *potensiometer* nilai data yang dikirimkan berupa data analog yang dihasilkan oleh pembagian tegangan yang diatur pada setiap putaran resistornya yang dapat diimplementasikan sebagai posisi. Posisi ini dapat diimplementasikan kepada *joint* pada robot SCARA dengan cara mengatur batasan minimal dan maksimal melalui program arduino. Pada hasil akhirnya nilai data yang dikirimkan oleh *potensiometer* kemudian dilakukan *mapping* data sesuai besaran sudut yang dapat dilakukan oleh *joint* yaitu dari 0-360 derajat. Pada tabel ?? merupakan hasil pengujian dari *potensiometer*.

Tabel 4.4: Hasil Pengujian *Potensiometer*

No	Pengujian	Joint Shoulder		Joint Elbow		End-effector	
		Max	Min	Max	Min	Max	Min
1	Pengujian Ke-1						
2	Pengujian Ke-2						
3	Pengujian Ke-3						
4	Pengujian Ke-4						
5	Pengujian Ke-5						

Pada hasil pengujian yang ditunjukkan oleh Tabel 4.4 terlihat bahwa pada saat nilai-nilai tertentu, *potensiometer* mengirimkan nilai data analog yang berubah-ubah. Hal tersebut dipengaruhi oleh pembacaan *potensiometer* yang belum stabil. Untuk membuat data analog yang dikirimkan oleh *potensiometer* menjadi lebih stabil maka pada program arduino ditambahkan program *moving avarage*. *Moving avarage*

berfungsi untuk membuat rata-rata nilai dari hasil data pembacaan nilai data pada *potensiometer* yang menyebabkan nilai menjadi lebih stabil. Tabel 4.5 merupakan hasil pengujian nilai data analog *potensiometer* setelah dilakukan *moving average*.

Tabel 4.5: Hasil Pengujian *Potensiometer* Menggunakan Program *Moving Average*

No	Pengujian	Joint Shoulder		Joint Elbow		End-effector	
		Max	Min	Max	Min	Max	Min
1	Pengujian Ke-1						
2	Pengujian Ke-2						
3	Pengujian Ke-3						
4	Pengujian Ke-4						
5	Pengujian Ke-5						

Setelah diberikan sebuah program *moving average* terlihat pada Tabel 4.5 nilai data yang diterima oleh Arduino Mega 2560 menjadi lebih stabil. Nilai yang stabil akan membantu sebuah sistem dalam melakukan pekerjaannya menjadi lebih baik.

4.1.5 Pengujian Rangkaian *Switching Valve Pneumatic*

Pengujian rangkaian *switching* berfungsi untuk mengetahui apakah rangkaian dapat bekerja dengan baik. Fungsi utama dari rangkaian *switching* yang diperuntukkan untuk *valve pneumatic* yaitu sebagai saklar penghubung dan pemutus daya yang masuk untuk *valve pneumatic*. Rangkaian dapat memutus dan menghubungkan daya dengan *trigger* dari sinyal data yang diberikan kepada *Gate* yang berupa sinyal digital *HIGH* dan *LOW* dari Arduino Mega 2560. Pengujian dilakukan dengan mengukur keberhasilan rangkaian sebagai rangkaian *switching* dengan variasi tegangan yang dilewatkan. Tabel 4.6 merupakan hasil pengujian dari rangkaian *switching* yang diotaki oleh TIP31.

Tabel 4.6: Hasil Pengujian Rangkaian *Switching Valve Pneumatic*

No	Sinyal dari Arduino	Tegangan <i>Valve Pneumatic</i>	Kondisi
1	HIGH	24 Volt	Valve bekerja
	LOW	0 Volt	Valve tidak bekerja

Pada hasil pengujian yang ditunjukkan pada tabel 4.6 terlihat bahwa rangkaian dapat berfungsi dengan baik dan dapat menghubungkan daya menuju *valve pneumatic*.

tic. Terlihat bahwa jika sinyal *HIGH* diberikan pada *Gate* TIP31 maka rangkaian *switching* akan menjadi rangkaian tertutup dan daya dapat dialirkan yang berarti *valve pneumatic* menjadi hidup dan siap beroperasi. Sebaliknya, jika pada *Gate* TIP31 diberikan sinyal *LOW* maka rangkaian *switching* menjadi rangkaian terbuka dan *valve pneumatic* tidak dapat bekerja.

4.1.6 Pengujian Kinematika Maju

4.1.7 Pengujian Kinematika Balik

Pengujian kinematika balik pada *arm manipulator* robot SCARA dilakukan dengan cara membandingkan posisi koordinat x , dan y yang aktual dengan jarak koordinat x dan y yang ada di dalam program Processing IDE. Setiap koordinat dimasukkan ke dalam perhitungan kinematika balik di dalam Processing IDE dan menghasilkan keluaran titik koordinat x dan y pada *end-effector*. Motor DC akan menggerakkan lengan *shoulder* dan *elbow* menuju posisi koordinat x dan y sesuai dari yang diperintahkan dalam program.

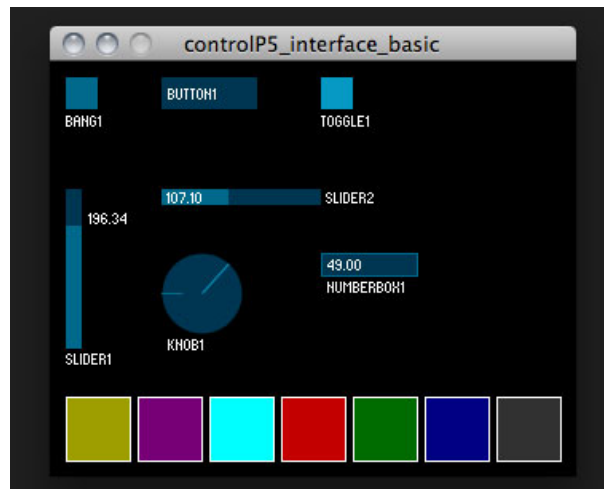
Pengujian ini dilakukan bertujuan untuk mengetahui *workspace* dari *arm manipulator* robot SCARA dan juga mengetahui akurasi dari posisi *end-effector* dengan perhitungan kinematika yang ada. Dalam pengujian ini, pengujian terdiri dari pengujian posisi koordinat posisi *end-effector*, dan juga hasil keluaran sudut pada masing-masing *joint* apakah sesuai dengan posisi *end-effector* yang diberikan. Sudut yang terdiri dari sudut *shoulder* dan sudut *elbow* yang dibandingkan dari pergerakan aslinya dan juga pergerakan pada processing IDE.

4.1.7.1 Pengujian Koordinat X

Pengujian koordinat x dilakukan untuk mengetahui batas minimal dan batas maksimal *end-effector* dari *arm manipulator* robot SCARA relatif terhadap sumbu x . Pengujian koordinat x dilakukan dengan cara mengujicobakan setiap titik dari ? cm sampai dengan total panjang *shoulder* dan *elbow* yaitu ? cm menggunakan perhitungan kinematika balik. Perhitungan posisi x diujicobakan menggunakan program Processing IDE, sementara untuk pengukuran posisi diukur secara faktual dari posisi *end-effector* dengan membuat sebuah diagram kartesius. Tabel 4.7 menunjukkan pengujian koordinat x dan Gambar 4.1 menunjukkan grafik dari *sampling* koordinat x .

Tabel 4.7: Pengujian Koordinat X

No	Koordinat Masukan (cm)		Koordinat Terukur (cm)		Error
	X	Y	X	Y	
1	1	0			
2		0			
3		0			
4		0			
5		0			
6		0			
7		0			
8		0			
9		0			
10		0			
11		0			
12		0			
13		0			
14		0			
15		0			
16		0			
17		0			
18		0			
19		0			
20		0			
21		0			
22		0			
23		0			
24		0			
25		0			
26		0			
27		0			
28		0			
29		0			
30		0			
31		0			
32		0			
33		0			
34		0			
35		0			
36		0			
37		0			
38		0			
39		0			
40		0			



Gambar 4.1: Grafik Pengujian Koordinat X

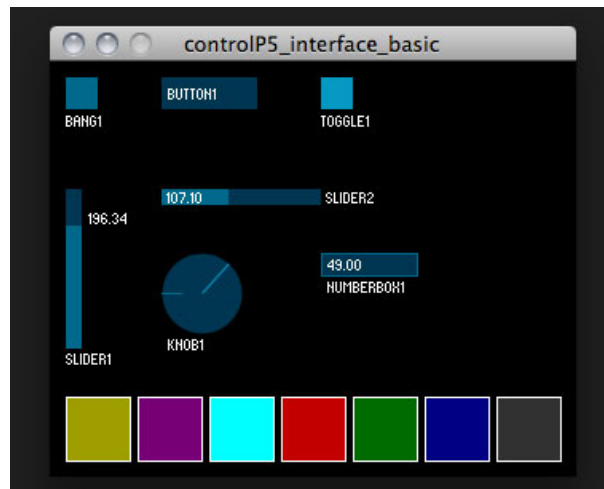
Dari hasil pengujian yang ditampilkan pada Tabel 4.7 dan juga Gambar 4.1 terlihat bahwa posisi *end-effector* yang dihasilkan pada robot SCARA dibandingkan dengan data sesuai perhitungan kinematika memiliki sedikit perbedaan pada beberapa data. Perbedaan data ini dapat ditoleransi karena dengan *sampling* koordinat x yang diuji hanya kurang dari lima persen nilai data yang berbeda. Pada hasil yang ditunjukkan diketahui bahwa posisi *end-effector* memiliki batas minimum dan juga maksimum. Batas minimum posisi *end-effector* pada sumbu x yaitu ? cm dari titik pusat dan batas maksimum posisi *end-effector* pada posisi sumbu x adalah ? cm yang merupakan panjang dari lengan *shoulder* dan juga lengan *elbow*.

4.1.7.2 Pengujian Koordinat Y

Pengujian koordinat y dilakukan untuk mengetahui batas minimal dan batas maksimal *end-effector* dari *arm manipulator* robot SCARA relatif terhadap sumbu y . Pengujian koordinat y dilakukan dengan cara mengujicobakan setiap titik dari ? cm sampai dengan total panjang *shoulder* dan *elbow* yaitu ? cm menggunakan perhitungan kinematika balik. Perhitungan posisi y diujicobakan menggunakan program Processing IDE, sementara untuk pengukuran posisi diukur secara faktual dari posisi *end-effector* dengan membuat sebuah diagram kartesius. Tabel 4.8 menunjukkan pengujian koordinat y dan Gambar 4.2 menunjukkan grafik dari *sampling* koordinat y .

Tabel 4.8: Hasil Pengujian Koordinat Y

No	Koordinat Masukan (cm)		Koordinat Terukur (cm)		Error
	X	Y	X	Y	
1	1	0			
2		0			
3		0			
4		0			
5		0			
6		0			
7		0			
8		0			
9		0			
10		0			
11		0			
12		0			
13		0			
14		0			
15		0			
16		0			
17		0			
18		0			
19		0			
20		0			
21		0			
22		0			
23		0			
24		0			
25		0			
26		0			
27		0			
28		0			
29		0			
30		0			
31		0			
32		0			
33		0			
34		0			
35		0			
36		0			
37		0			
38		0			
39		0			
40		0			



Gambar 4.2: Grafik Pengujian Koordinat Y

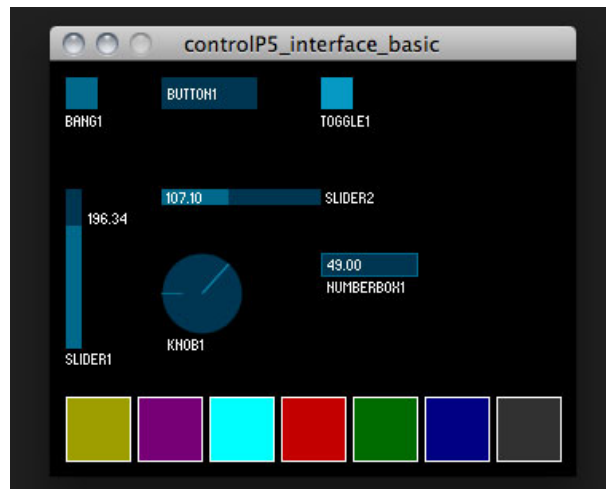
Dari hasil pengujian yang ditampilkan pada Tabel 4.8 dan juga Gambar 4.2 terlihat bahwa posisi *end-effector* yang dihasilkan pada robot SCARA dibandingkan dengan data sesuai perhitungan kinematika memiliki sedikit perbedaan pada beberapa data. Perbedaan data ini dapat ditoleransi karena dengan *sampling* koordinat y yang diuji hanya kurang dari lima persen nilai data yang berbeda. Pada hasil yang ditunjukkan diketahui bahwa posisi *end-effector* memiliki batas minimum dan juga maksimum. Batas minimum posisi *end-effector* pada sumbu x yaitu ? cm dari titik pusat dan batas maksimum posisi *end-effector* pada posisi sumbu y adalah ? cm yang merupakan panjang dari lengan *shoulder* dan juga lengan *elbow*.

4.1.7.3 Pengujian *Joint Shoulder*

Pengujian dilakukan dengan cara memberikan sebuah posisi koordinat x dan y dengan nilai yang bervariasi. Nilai tersebut dimasukkan ke dalam sebuah perhitungan kinematika balik yang ada pada program Processing IDE sesuai dengan persamaan pada bab sebelumnya. Hasil yang dihasilkan oleh perhitungan merupakan nilai sudut dari *joint shoulder* dan juga *elbow*. Perhitungan aktual dilakukan dengan cara mengukur sudut pada setiap *joint* menggunakan bantuan busur atau *software* sensor kemiringan yang di-*install* pada smartphone. Tabel ?? merupakan hasil pengujian dari *joint shoulder* berdasarkan *sampling* posisi yang diambil. Gambar 4.3 merupakan grafik hubungan dari sudut aktual dan sudut berdasarkan perhitungan kinematika.

Tabel 4.9: Hasil Pengujian *Joint Shoulder*

No	Sudut Hasil Perhitungan Kinematika Balik(deg)	Sudut Aktual (deg)	Error
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			



Gambar 4.3: Grafik Pengujian *Joint Shoulder*

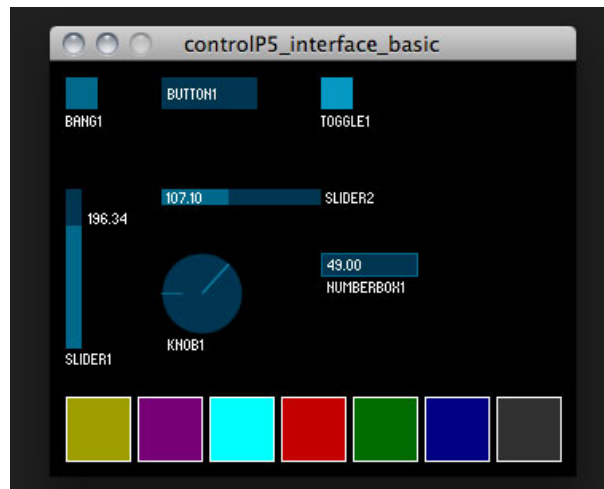
Pada hasil yang ditunjukkan oleh Tabel 4.9 dan Gambar 4.3 terlihat bahwa secara keseluruhan nilai sudut yang dihasilkan oleh aktual pada robot SCARA dibandingkan dengan perhitungan kinematika oleh Processing IDE tidak begitu terdapat perbedaan. Terlihat bahwa nilai error memiliki nilai minimum persen dan nilai maksimum ... persen. Persentase error didapat dari data yang tidak sesuai dibagi oleh kekeliruan data yang diuji coba. Dapat dikatakan bahwa pengujian sudut *joint shoulder* cukup baik.

4.1.7.4 Pengujian *Joint Elbow*

Pengujian *joint elbow* dilakukan dengan membandingkan nilai sudut yang dihasilkan oleh perhitungan kinematika yang ada pada program Processing IDE dengan sudut yang dihasilkan dalam aktual robot SCARA. Pengujian dilakukan dengan cara memberikan sebuah posisi koordinat x dan y dengan nilai yang bervariasi. Nilai tersebut dimasukkan ke dalam sebuah perhitungan kinematika balik yang ada pada program Processing IDE. Hasil yang dihasilkan oleh perhitungan merupakan nilai sudut dari *joint shoulder* dan juga *elbow*. Perhitungan aktual dilakukan dengan cara mengukur sudut pada setiap *joint* menggunakan bantuan busur atau *software* sensor kemiringan yang di-*install* pada smartphone. Tabel 4.10 merupakan hasil pengujian dari *joint elbow* berdasarkan *sampling* posisi yang diambil. Gambar 4.4 merupakan grafik hubungan dari sudut aktual dan sudut berdasarkan perhitungan kinematika.

Tabel 4.10: Hasil Pengujian *Joint Elbow*

No	Sudut Hasil Perhitungan Kinematika Balik(deg)	Sudut Aktual (deg)	Error
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			

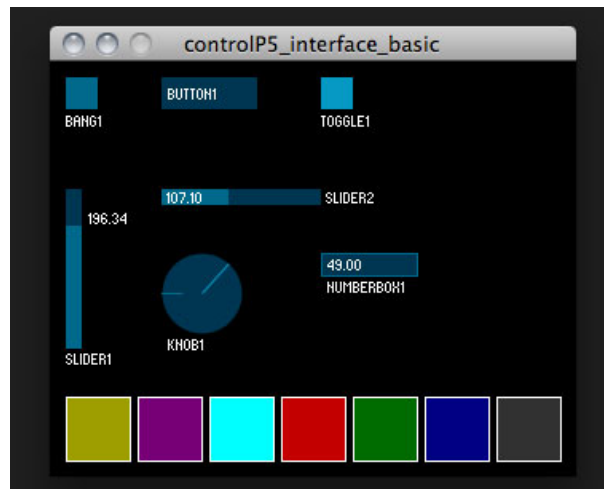


Gambar 4.4: Grafik Pengujian *Joint Elbow*

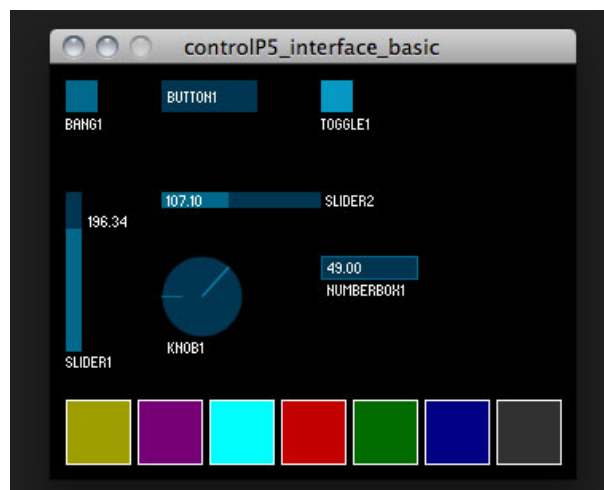
Pada hasil yang ditunjukkan oleh Tabel 4.10 dan Gambar 4.4 terlihat bahwa secara keseluruhan nilai sudut yang dihasilkan oleh aktual pada robot SCARA dibandingkan dengan perhitungan kinematika oleh Processing IDE tidak begitu terdapat perbedaan. Terlihat bahwa nilai error memiliki nilai minimum persen dan nilai maksimum ... persen. Persentase error didapat dari data yang tidak sesuai dibagi oleh kekeliruan data yang diuji coba. Nilai tersebut dapat ditoleransi dan pengujian sudut *joint elbow* dapat dikatakan baik.

4.1.8 Pengujian GUI

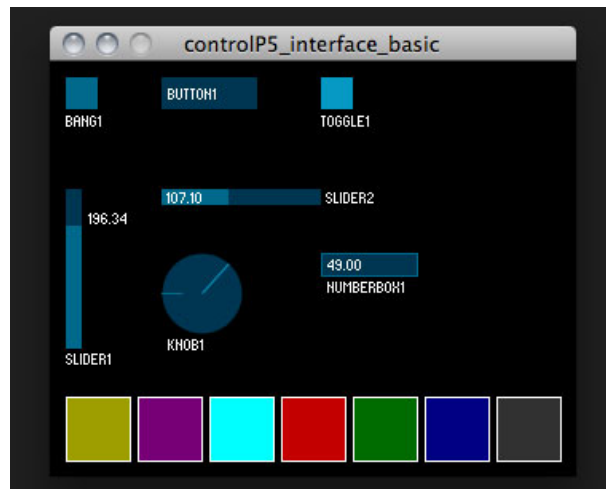
Pengujian GUI dilakukan untuk membandingkan posisi dari aktual robot SCARA dengan posisi yang ada pada Processing IDE. Pengujian dilakukan dengan memberikan *sampling* posisi yang kemudian dilihat posisi dari robot secara aktual maupun Processing IDE. Gambar 4.5, Gambar 4.6, dan Gambar 4.7 merupakan hasil *sampling* perbandingan posisi dari robot SCARA secara aktual dan secara animasi.



Gambar 4.5: Perbandingan Gambaran Robot SCARA



Gambar 4.6: Perbandingan Gambaran Robot SCARA



Gambar 4.7: Perbandingan Gambaran Robot SCARA

Dari hasil yang ditunjukkan oleh Gambar 4.5, Gambar 4.6, dan Gambar 4.7 terlihat bahwa perbedaan posisi dari robot SCARA aktual dan pada Processing IDE hanya beberapa dan dapat ditoleransi. Terlihat bahwa posisi koordinat *end-effector* yang ditunjukkan oleh robot SCARA aktual dan robot SCARA processing IDE terlihat beberapa perbedaan yang masih dapat ditoleransi. Sudut dari *shoulder* dan *elbow* yang masing-masing ditunjukkan terlihat sama. Dengan begitu, pengujian GUI dapat dikatakan baik.

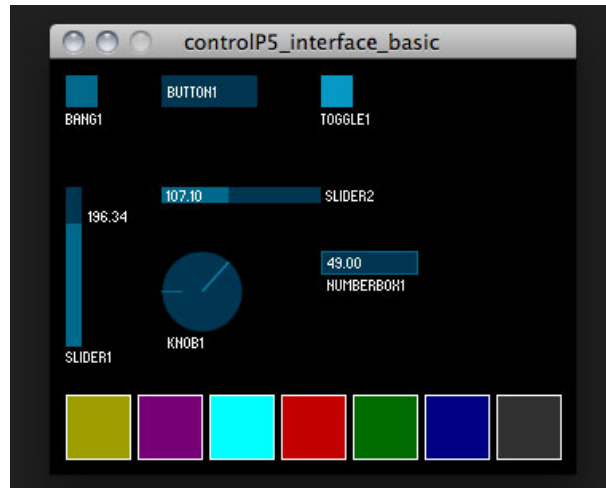
4.2 Pengujian Keseluruhan

Setelah semua komponen sistem yang diuji dapat bekerja dengan baik, pengujian selanjutnya adalah pengujian sistem secara keseluruhan dengan beberapa parameter uji yang diberikan. Parameter uji tersebut diantaranya adalah pengujian akurasi robot lengan, dan simulasi robot lengan.

4.2.1 Pengujian Akurasi Robot Lengan

Pada pengujian akurasi robot lengan dilakukan pengujian dengan memberikan posisi koordinat x dan y dan melihat kembali posisi x dan y secara aktual, melihat besar sudut pada masing-masing joint dengan perbandingan perhitungan kinematika. Pengujian akurasi robot lengan ini akan menghasilkan sebuah hasil secara luas dan dapat ditarik untuk sebuah kesimpulan yang menentukan robot SCARA dapat bekerja baik atau buruk secara keseluruhan. Tabel 4.11 merupakan hasil dari pengujian

akurasi robot lengan secara keseluruhan. Gambar 4.8 merupakan grafik dari hasil pengujian akurasi robot lengan secara keseluruhan.



Gambar 4.8: Grafik Pengujian Akurasi Robot Secara Keseluruhan

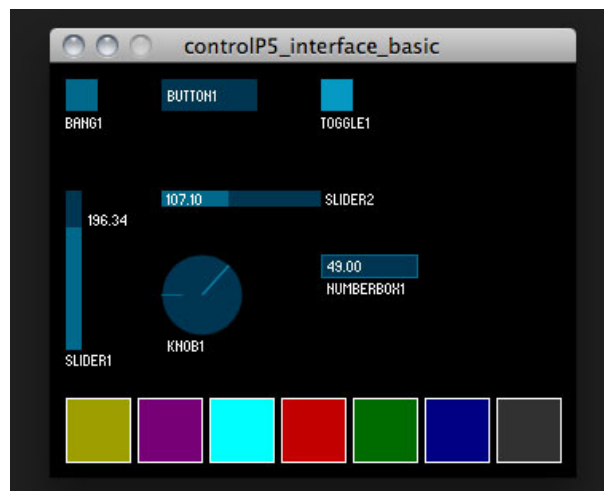
Pada hasil pengujian yang ditunjukkan pada Tabel 4.11 dan Gambar 4.8 terlihat bahwa pengujian mendapat beberapa data yang beragam. Terlihat bahwa nilai kesalahan paling banyak terdapat pada bagian posisi koordinat x dan y dan kesalahan paling sedikit pada bagian sudut *shoulder*. Sedangkan untuk sistem keseluruhan kesalahan total pada akurasi robot memiliki nilai ... persen. Nilai sebesar ini merupakan nilai yang cukup baik bagi sebuah sistem. Dengan nilai tersebut maka sistem kinematika robot SCARA ini dapat dikatakan baik.

4.2.2 Pengujian Simulasi Robot Lengan

Pada pengujian ini, dilakukan serangkaian simulasi robot lengan untuk melakukan pergerakan sesuai dengan data yang diberikan melalui GUI yang telah dibuat. Simulasi ini menggunakan semua komponen yang ada di dalam perancangan, diantaranya robot SCARA, *workspace*, catu daya, komputer personal, GUI Processing IDE, dan objek.

Pertama, robot SCARA berada posisi normal yang berarti pada posisi lurus searah dengan sumbu y . Robot SCARA akan mulai bergerak pada saat nilai data pada Processing IDE telah didapatkan dan kemudian dikirimkan ke Arduino Mega 2560. Proses pemilihan data pada Processing IDE dilalui pada sebuah GUI yang telah dibuat sesuai dengan fungsi dari robot SCARA. Pada GUI terdapat beberapa pilihan dalam memberikan sebuah nilai masukan.

Kedua, robot SCARA akan mulai bergerak menuju posisi seperti data yang dimasukkan dan kemudian berhenti pada posisi tersebut dan mengambil objek yang dibawa oleh *end-effector* pada posisi akhir. Pergerakan *end-effector* ditenagai oleh tekanan udara pada sebuah kompresor yang dikontrol oleh *valve pneumatic* yang diotaki oleh TIP31. Posisi akhir merupakan posisi dimana semua objek dikumpulkan. Setelah melakukan satu pekerjaan tersebut, robot SCARA kembali pada posisi normal dan menunggu data yang diberikan kembali oleh Processing IDE. Gambar 4.9 merupakan simulasi robot lengan SCARA menggunakan masukan pada GUI yang telah dibuat.



Gambar 4.9: Simulasi Robot Lengan Menggunakan GUI

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian dan pembahasan terhadap alat dan sistem “Kinematika dan Antarmuka Robot SCARA berbasis Processing IDE” yang telah dirancang dan dibuat ini, maka dapat disimpulkan sebagai berikut:

1. Sistem kendali robot lengan dengan Arduino Mega 2560 dapat digunakan dengan baik dan dapat menerapkan sistem kinematika dengan benar. Hasil dari nilai data yang diperoleh pada Processing IDE selanjutnya digunakan sebagai masukan untuk kinematika balik yang hasilnya dikirimkan pada Arduino Mega 2560 yang diteruskan pada motor DC.
2. GUI yang dibuat sebagai jembatan antara *user* dengan hardware memiliki beberapa masukan data dan juga tampilan data berupa animasi dari robot SCARA.
3. Implementasi dari kinematika robot SCARA dapat berjalan dengan baik meskipun terdapat beberapa data yang tidak akurat.
4. Hal-hal yang menyebabkan robot lengan SCARA tidak akurat dalam menentukan posisi *end-effector* diantaranya sebagai berikut:

Terdapat toleransi sudut pada joint *shoulder* dan joint *elbow* yang menyebabkan θ_1, θ_2 , memiliki nilai toleransi sebesar 3 derajat.

Keakuratan pembacaan nilai *analog potentiometer* pada *feedback* posisi.

5.2 Saran

Setelah mengambil beberapa kesimpulan dan melihat dari sistem secara keseluruhan, terdapat beberapa saran disampaikan untuk menambah mutu dan kualitas dari sistem kendali robot lengan SCARA adalah sebagai berikut:

1. Untuk menghasilkan kinematika yang baik maka toleransi besar sudut pada motor DC dapat diperkecil dan ditambahkan kontrol PID ke dalam program.
2. Untuk menghasilkan tampilan yang nyata maka pada sistem diberi tambahan kamera yang dapat ditampilkan pada GUI.

DAFTAR PUSTAKA