

**RANCANG BANGUN PENGONTROL GERAKAN ROBOT
OPENMANIPULATOR DENGAN MATLAB**

TUGAS AKHIR



DANIEL KELVIN WINARTA

NIM : 311710007

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2024**

LEMBAR PENGESAHAN

Judul :

RANCANG BANGUN PENGONTROL GERAKAN ROBOT OPENMANIPULATOR DENGAN MATLAB

Oleh :

Daniel Kelvin Winarta

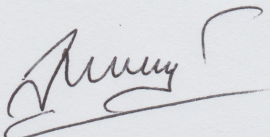
311710007

Dari :

**Program Studi Teknik Informatika
Fakultas Teknologi dan Desain
Universitas Ma Chung**

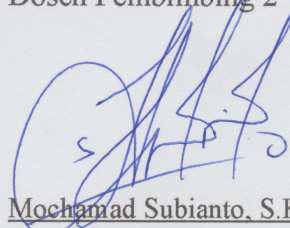
Telah dinyatakan lulus dalam melaksanakan Tugas Akhir (TA) sebagai syarat kelulusan dan berhak mendapatkan gelar Sarjana Komputer (S.Kom)

Dosen Pembimbing 1



Prof. Dr. Eng. Romy Budhi, S.T., M.T., M.Pd.
NIP. 20070035

Dosen Pembimbing 2



Mochamad Subianto, S.Kom., M.Cs.
NIP. 20100002

Dekan Fakultas Teknologi dan Desain



Prof. Dr. Eng. Romy Budhi, S.T., M.T., M.Pd.
NIP. 20070035

PERNYATAAN KEASLIAN

Dengan ini saya, Daniel Kelvin Winarta menyatakan bahwa seluruh isi dari laporan tugas akhir saya yang berjudul “Rancang Bangun Pengontrol Gerakan Robot Openmanipulator dengan MATLAB” merupakan hasil penulisan saya sendiri dan telah diselesaikan tanpa menggunakan konten-konten ilegal serta bukan karya orang lain yang saya klaim sebagai karya saya. Referensi terkait yang saya gunakan dalam penyusunan laporan tugas akhir ini telah saya cantumkan di daftar pustaka. Apabila laporan tugas akhir ini terbukti bukanlah karya saya melainkan karya orang lain, saya bersedia menerima sanksi sesuai dengan aturan yang berlaku di Universitas Ma Chung.

Malang, 26 Juli 2024



Handwritten signature of Daniel Kelvin Winarta.

Daniel Kelvin Winarta

RANCANG BANGUN PENGONTROL GERAKAN ROBOT OPENMANIPULATOR DENGAN MATLAB

Daniel Kelvin Winarta, Romy Budhi Widodo, Mochammad Subianto

Universitas Ma Chung

Abstrak

Seiring dengan berkembangnya ilmu pengetahuan dan teknologi, tuntutan masyarakat pada industri agar dapat menghasilkan barang berkualitas baik serta cepat diproduksi tentu meningkat. Oleh karena itu, banyak industri mulai meninggalkan sistem produksi manual dan beralih ke sistem produksi otomatis agar dapat melakukan proses produksi yang lebih efisien serta mendapat hasil yang lebih konsisten. Agar tujuan tersebut tercapai, industri menggunakan robot dalam proses produksinya. Salah satu jenis robot yang digunakan adalah lengan robot. Pada penelitian ini akan dilakukan pengembangan lengan robot agar dapat melakukan gerakan memindah objek yang bentuknya sering ditemui dalam kehidupan sehari-hari. Kontrol gerakan pada lengan robot dilakukan menggunakan aplikasi MATLAB. Prototipe lengan robot sendiri terdiri dari beberapa perangkat keras, yaitu U2D2 *board*, *power adapter*, dan servo *dynamixel*. Pengujian dilakukan dengan melakukan gerakan lengan robot pada posisi yang sudah ditentukan sebelumnya. Pengujian berhasil 100% melakukan seluruh gerakan pindah posisi pada lengan robot, namun lengan robot masih belum dapat memegang dan mengambil objek dalam geraknya sehingga kedepannya perlu dilakukan penyempurnaan agar lengan robot dapat berfungsi sebagaimana mestinya.

Kata kunci : *dynamixel*, MATLAB, lengan robot, OpenManipulator

RANCANG BANGUN PENGONTROL GERAKAN ROBOT OPENMANIPULATOR DENGAN MATLAB

Daniel Kelvin Winarta, Romy Budhi Widodo, Mochammad Subianto

Universitas Ma Chung

Abstract

Along with the development of science and technology, the demands of society on industry to be able to produce good quality goods and be produced quickly certainly increase. Therefore, many industries have begun to abandon manual production systems and switch to automated production systems in order to carry out more efficient production processes and get more consistent results. In order to achieve this goal, the industry uses robots in its production process. One type of robot used is the robot arm. In this study, the development of a robot arm will be carried out so that it can move objects whose shapes are often encountered in everyday life. Movement control on the robot arm is carried out using the MATLAB application. The robot arm prototype itself consists of several hardware, namely U2D2 board, power adapter, and servo dynamixel. Testing is carried out by moving the robot arm in a predetermined position. The test was 100% successful in carrying out all position movement on the robot arm, but the robot arm still cannot hold and take objects in its movement so that in the future improvements need to be made so that the robot arm can function properly.

Keywords : dynamixel, MATLAB, OpenManipulator, robot arm

KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, penelitian ini bisa terlaksana dengan baik. Penelitian yang telah dilakukan selanjutnya disusun ke dalam laporan tugas akhir yang menjadi salah satu syarat kelulusan bagi mahasiswa dan mahasiswi Universitas Ma Chung. Dukungan moral, materi, dan waktu juga telah diberikan dari banyak pihak selama penyusunan laporan tugas akhir ini. Secara khusus, Penulis ingin berterima kasih kepada :

1. Bapak Prof. Dr. Eng. Romy Budhi, S.T., M.T., M.Pd., selaku Dekan Fakultas Teknologi dan Desain Universitas Ma Chung, Dosen Pembimbing I, dan Dosen Penguji I.
2. Bapak Hendry Setiawan, S.T., M.Kom., selaku Kepala Program Studi Teknik Informatika Universitas Ma Chung dan Ketua Penguji.
3. Bapak Mochamad Subianto, S.Kom., M.Cs., selaku Dosen Program Studi Teknik Informatika Universitas Ma Chung, Dosen Pembimbing II dan Dosen Penguji II.
4. Universitas Ma Chung yang telah memberikan kesempatan kepada Penulis untuk dapat memperoleh ilmu pengetahuan dan pengalaman.
5. Orang tua, saudara, dan teman-teman yang selalu memberikan berbagai macam dukungan kepada Penulis sehingga penelitian dan laporan tugas akhir ini dapat diselesaikan dengan baik.

Penulis sadar bahwa hasil dari penelitian dan laporan tugas akhir ini masih belum bisa dikatakan sempurna serta banyak kekurangan. Besar harapan Penulis agar mendapatkan kritik dan saran yang membangun agar kedepannya laporan tugas akhir ini dapat bermanfaat bagi seluruh sivitas akademika Universitas Ma Chung serta semua pihak lain.

Malang, 26 Juli 2024

Daniel Kelvin Winarta

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
DAFTAR LAMPIRAN.....	vi
Bab I : Pendahuluan	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah.....	2
1.4 Perumusan Masalah.....	3
1.5 Tujuan Penelitian.....	3
1.6 Luaran.....	3
1.7 Manfaat Penelitian.....	3
1.8 Sistematika Penulisan.....	4
Bab II : Tinjauan Pustaka	5
2.1 Robot <i>Arm</i>	5
2.2 <i>Dynamixel</i>	8
2.3 U2D2	9
2.4 U2D2 <i>Power Hub</i>	10
2.5 MATLAB.....	11
2.6 <i>Dynamixel Wizard</i>	12
Bab III : Analisis dan Perancangan Sistem	13
3.1 Tahapan Penelitian	13
3.2 Proses Pengerjaan.....	14
3.3 Metode Evaluasi.....	15
3.4 Tabel Hasil Pengujian	16
Bab IV : Hasil dan Pembahasan.....	17

4.1	Rincian Penelitian	17
4.1.1	Tempat dan waktu	17
4.1.2	Alat dan komponen.....	17
4.2	Pengambilan Data	19
4.3	Hasil Pengujian	20
4.4	Diskusi.....	24
Bab V : Simpulan dan Saran		25
5.1	Simpulan.....	25
5.2	Saran.....	25
DAFTAR PUSTAKA		26
LAMPIRAN		

DAFTAR TABEL

Tabel 3.1 Format Penulisan Hasil Pengujian	16
Tabel 4.1 Data Posisi Robot <i>Arm</i>	19
Tabel 4.2 Hasil Pengujian Gerakan Robot <i>Arm</i>	20

DAFTAR GAMBAR

Gambar 2.1 Robot <i>Arm</i>	5
Gambar 2.2 Spesifikasi OpenMANIPULATOR-X	6
Gambar 2.3 Dimensi OpenMANIPULATOR-X	7
Gambar 2.4 Servo <i>Dynamixel</i>	8
Gambar 2.5 U2D2 <i>Board</i>	9
Gambar 2.6 U2D2 <i>Power Hub</i>	10
Gambar 2.7 Program MATLAB	11
Gambar 2.8 Program <i>Dynamixel Wizard</i>	12
Gambar 3.1 Tahapan Pembuatan Prototipe Robot <i>Arm</i>	13
Gambar 3.2 Rangkaian Prototipe Robot <i>Arm</i>	14
Gambar 3.3 Ilustrasi Posisi Robot <i>Arm</i>	15
Gambar 3.4 Objek Tabung	16
Gambar 3.5 Objek Bola	16
Gambar 4.1 Rancangan Prototipe	18
Gambar 4.2 Konfigurasi pada <i>Device Manager</i>	18
Gambar 4.3 Penjelasan ID Servo <i>Dynamixel</i>	20
Gambar 4.4 Data Posisi Robot <i>Arm</i>	21
Gambar 4.5 Robot <i>Arm</i> pada Posisi <i>Home</i>	21
Gambar 4.6 Robot <i>Arm</i> pada Posisi B1	22
Gambar 4.7 Robot <i>Arm</i> pada Posisi B2	22
Gambar 4.8 Robot <i>Arm</i> pada Posisi B3	23
Gambar 4.9 Robot <i>Arm</i> pada Posisi B4	23

DAFTAR LAMPIRAN

Lampiran A. Kode Sumber MATLAB

Bab I

Pendahuluan

1.1 Latar Belakang

Saat ini, perkembangan ilmu pengetahuan dan teknologi telah meningkat dengan sangat pesat. Peningkatan ini terjadi seiring dengan adanya tuntutan permintaan yang tinggi dari masyarakat kepada industri agar dapat memproduksi barang dengan cepat dan tepat namun tetap memiliki kualitas yang baik. Untuk memenuhi permintaan tersebut, sebagian besar industri mulai meninggalkan sistem manual yang masih banyak melibatkan peran manusia dan beralih ke sistem otomatis. Hal ini dilakukan untuk mendapatkan efisiensi waktu dan penggandaan hasil produksi yang lebih besar. Robot adalah salah satu contoh alat yang seringkali digunakan oleh industri dalam penerapan sistem otomatis.

Robotika adalah bidang studi yang memfokuskan pada pembuatan, pengembangan, dan aplikasi robot (Sari, 2024). Bidang studi inilah yang kemudian melahirkan dan memopulerkan robot sebagai alat pintar yang bisa membantu meringankan pekerjaan manusia. Menurut Muslim (2022), robot adalah seperangkat alat mekanik yang bisa melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu (kecerdasan buatan). Berdasarkan bentuknya, robot terbagi menjadi beberapa jenis seperti *humanoid* (robot yang memiliki bentuk seperti manusia), *fixed* robot (robot yang tidak dapat berpindah), *mobile* robot (robot yang dapat berpindah secara dinamis karena memiliki roda atau kaki), *bug* robot (robot yang bentuknya mirip dengan binatang), dan *combination* (robot dengan bentuk gabungan dari keempat robot sebelumnya) (Insani, 2023). Namun, salah satu jenis robot yang paling sering digunakan dalam industri adalah robot *arm* atau robot manipulator.

Robot *arm* atau robot manipulator adalah robot yang memiliki bentuk dan fungsi menyerupai lengan manusia. Menurut Abidin dan Suprianto (2020), robot *arm* terdiri dari beberapa bagian, yaitu *base*, *joint*, *link*, dan *end effector*. Bagian yang pertama adalah *base*. Bagian ini merupakan dasar atau pijakan pada robot *arm*.

Bagian yang kedua adalah *joint*. Bagian ini merupakan sendi atau titik yang mampu menghasilkan gerakan yang terkontrol. Bagian yang ketiga adalah *link*. Bagian ini merupakan rangka penyusun robot *arm*. Bagian yang terakhir adalah *end effector*. Bagian ini merupakan bagian paling ujung robot *arm* yang melakukan tugas tertentu, seperti pencapit dan lain sebagainya. Dengan adanya keempat bagian penting tersebut, robot *arm* pada akhirnya dapat diprogram dan digunakan untuk melakukan berbagai macam pekerjaan tertentu, seperti memilih dan memindahkan (*pick and place*) objek tertentu hingga melakukan proses pengelasan (Ayega dkk., 2021).

Berdasarkan diskusi dengan Bapak Prof. Dr. Eng. Romy Budhi, S.T., M.T., M.Pd., selaku Kepala Pusat Studi *Human Machine Interaction* (HMI) Universitas Ma Chung, beliau mengharapkan agar robot *arm* dapat menjadi sebuah alat bantu yang digunakan pada laboratorium khususnya laboratorium yang biasa menggunakan bahan kimia berbahaya.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, identifikasi masalah yang ada pada penelitian ini adalah diperlukannya robot pembantu pada laboratorium agar dapat digunakan sewaktu petugas laboratorium mengambil atau memindahkan bahan kimia berbahaya sehingga dapat meminimalisir cedera.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

1. Menggunakan program MATLAB.
2. Prototipe yang digunakan adalah *dynamixel* dengan 4 servo.
3. Perangkat belum di uji coba langsung di lapangan dan masih menggunakan objek tiruan.
4. Pengujian dilakukan di Laboratorium *Human Machine Interaction* (HMI) Universitas Ma Chung.

1.4 Perumusan Masalah

Berdasarkan permasalahan tersebut, rumusan masalah pada penelitian ini adalah bagaimana prototipe dapat melakukan kontrol gerakan robot *arm* agar dapat digunakan sebagai pengganti tangan manusia ketika melakukan percobaan kimia.

1.5 Tujuan Penelitian

Tujuan penelitian ini adalah membuat sebuah prototipe untuk melakukan kontrol gerakan robot *arm* atau robot manipulator dengan aplikasi MATLAB.

1.6 Luaran

Luaran dari penelitian ini adalah prototipe untuk melakukan kontrol gerakan robot *arm* atau robot manipulator dengan aplikasi MATLAB dan publikasi ilmiah.

1.7 Manfaat Penelitian

Manfaat yang diharapkan pada penulisan penelitian ini adalah :

1. Bagi penulis, mendapatkan kesempatan untuk menerapkan ilmu yang sudah dipelajari di lingkup universitas maupun diluar universitas agar mendapatkan wawasan dan pengalaman baru.
2. Bagi Universitas Ma Chung, khususnya Program Studi Teknik Informatika, dapat mempersiapkan lulusan yang berkompeten dan siap kerja dengan memberikan bekal kepada mahasiswa maupun mahasiswi melalui kegiatan penelitian yang berlangsung.
3. Bagi peneliti selanjutnya, mendapatkan kesempatan untuk melakukan pengembangan dari penelitian ini untuk mendapatkan hasil penelitian berikutnya yang lebih baik.

1.8 Sistematika Penulisan

Sistematika penulisan pada penelitian ini adalah sebagai berikut :

Bab I : Pendahuluan

Bab ini akan dijelaskan latar belakang penelitian, berupa rumusan masalah, identifikasi masalah, tujuan, batasan, sistematika penulisan, luaran dan juga manfaat dari penelitian yang dilakukan.

Bab II : Tinjauan Pustaka

Bab ini akan menjelaskan mengenai teori pendukung pengembangan prototipe perangkat untuk melakukan kontrol gerakan robot *arm* atau robot manipulator dengan aplikasi MATLAB meliputi perangkat yang digunakan, pengujian, dan teori tentang sistem yang digunakan untuk penelitian ini.

Bab III : Analisis dan Perancangan Sistem

Bab ini akan menjelaskan lagi secara lebih rinci mengenai alur penelitian, desain sistem serta penjelasan untuk kondisi dan pelaksanaan pengujian perangkat.

Bab IV Hasil dan Pembahasan

Bab ini akan menjelaskan hasil yang didapat dari hasil pembuatan prototipe serta pengujian alat yang telah dilakukan pada kondisi tertentu.

Bab V Simpulan dan Saran

Bab ini berisi tentang kesimpulan dari hasil penelitian, pembuatan prototipe yang sudah dilakukan serta saran bagi peneliti selanjutnya.

Bab II

Tinjauan Pustaka

2.1 Robot *Arm*

Robot merupakan suatu alat mekanik yang dapat melakukan tugas fisik dengan kontrol maupun pengawasan manusia. Robot dapat diartikan dengan mesin yang dapat menyelesaikan tugas fisik dengan program yang telah dirancang dan didefinisikan. Robot *arm* atau biasa disebut dengan robot lengan merupakan salah satu jenis robot yang sering digunakan pada suatu industri untuk memindahkan *material, part*, atau beberapa peralatan lainnya. Robot lengan dapat memudahkan suatu industri dalam mengurangi biaya dan meningkatkan keunggulan kompetitif (Rahman dkk., 2020).

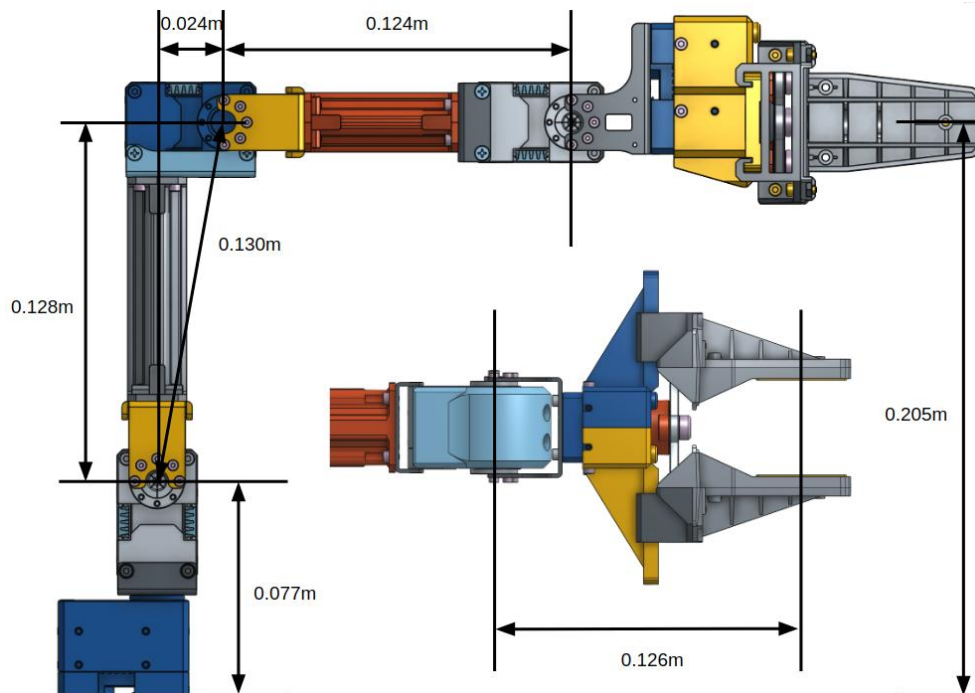


Gambar 2.1 Robot *Arm*

Items	Unit	OpenMANIPULATOR-X
Actuator		DYNAMIXEL XM430-W350-T
Input Voltage	V	12
DOF	-	5 (4 DOF + 1 DOF Gripper)
Payload	g	500
Repeatability	mm	< 0.2
Speed(Joint)	RPM	46
Weight	kg (lb)	0.70 (1.54)
Reach	mm (in)	380 (14.9)
Gripper Stroke	mm (in)	20~75 (0.79~2.95)
Communication	-	TTL Level Multidrop BUS
Software	-	ROS, DYNAMIXEL SDK, Arduino, Processing
Main Controller	-	PC, OpenCR

Gambar 2.2 Spesifikasi OpenMANIPULATOR-X

Manfaat dari robot lengan adalah untuk meningkatkan produktivitas dan efisiensi yang tinggi. Pada suatu industri, robot lengan dirancang untuk beroperasi selama 24 jam sehingga dapat meningkatkan *output*. Manfaat dari robot lengan selanjutnya adalah untuk meningkatkan akurasi dan kemampuan beradaptasi dan fleksibilitas. Robot lengan dapat dirancang secara konsisten dan efisien layaknya pekerjaan yang dilakukan oleh tangan manusia. Jumlah kemungkinan gerakan pada sendi robot merupakan definisi dari *degree of freedom* (DoF). Teori dari *degree of freedom* pada robot adalah semakin banyak jumlah DoF-nya maka robot tersebut memiliki banyak gerakan yang dapat dilakukan. Jumlah dari DoF tergantung dengan konfigurasi *hardware* (Pamungkas & Noviansyah, 2021).



Gambar 2.3 Dimensi OpenMANIPULATOR-X

Robot lengan (robot *arm*) merupakan suatu robot yang dirancang sama seperti cara kerja lengan manusia. Robot lengan terdiri dari beberapa komponen utama yaitu *link*, *joint*, *arm*, dan *end effector*. Komponen *link* adalah panjang yang dimiliki atau dirancang pada lengan robot. *Arm* merupakan komponen yang digerakkan oleh motor atau servo, dimana gerakannya akan disesuaikan oleh perangkat kontroler. Sedangkan *joint* adalah pertemuan antara *link* dengan *link* pada suatu *base* lengan robot yang akan membuat *degree of freedom* (DoF). Sedangkan *end effector* adalah suatu *gripper* untuk melakukan fungsi tertentu (Oktama dkk., 2018).

Pada robot lengan atau robot manipulator, terdapat dua permodelan kinematik yang membahas terkait gerakan benda tanpa mempermasalahkan gaya yang menyebabkan gerakan. Kinematik tersebut adalah *forward kinematics* dan *inverse kinematics*. *Forward kinematics* merupakan suatu teori yang menjelaskan terkait proses perhitungan orientasi dan posisi dari *end effector*, dimana proses perhitungan tersebut didasari oleh sudut *joint*. Sedangkan *inverse kinematics* adalah pemberian posisi dari *end effector*. *Output* yang dihasilkan adalah perubahan

besaran sudut pada tiap *joint* agar mencapai posisi *end effector* (Utomo & Munadi, 2013).

Parameter Denavit Hattenberg merupakan suatu metode yang menyatukan antara proses perhitungan rotasi dengan tranlasi. *Output* yang dihasilkan pada metode ini adalah suatu matriks yang dapat menyertakan nilai sudut putar serta jarak sendi lengan robot. Parameter Denavit Hattenberg suatu notasi yang mendeskripsikan parameter hubungan antara satu sendi dengan sendi lainnya. Pada metode Denavit Hattenberg banyak digunakan oleh robot yang menggunakan perhitungan *forward kinematics*. Pada notasi tersebut dijelaskan bahwa terdapat 4 parameter yaitu (a_i , α_i , d_i , θ_i). Denavit Hattenberg (D-H) berfungsi untuk menggambarkan beberapa hubungan *link* pada suatu robot, dimana *link* yang ada pada robot berbentuk benda tegar (Purwoto dkk., 2020).

2.2 *Dynamixel*

Dynamixel merupakan komponen robot yang banyak digunakan oleh perusahaan karena memiliki kemampuan dalam ekspansi yang serbaguna. *Dynamixel* memiliki berbagai fungsi dan data *feedback* terkait tegangan, suhu, kecepatan, dan posisi. *Dynamixel* banyak digunakan pada robot yang menggunakan sistem *multi joint*, salah satunya adalah robot *arm*. Servo *dynamixel* atau motor servo merupakan suatu komponen yang menggunakan sistem *closed feedback* yang nantinya akan diinformasikan kembali kepada rangkaian kontrol yang ada pada motor servo (Pambudi dkk., 2018).



Gambar 2.4 Servo *Dynamixel*

Motor servo terbagi menjadi dua jenis yaitu motor servo digital dan servo analog. Terdapat beberapa perbedaan dalam dua jenis motor servo. Perbedaan yang pertama yaitu sinyal yang dihasilkan oleh servo digital lebih kuat dan dapat menggerakkan servo sesuai dengan sudut yang diinginkan sehingga memberikan respon yang cepat. Perbedaan yang kedua yaitu servo digital memiliki akurasi pemosisian yang tinggi dibandingkan dengan servo analog. Perbedaan ketiga yaitu servo digital memiliki kemampuan anti interferensi yang kuat (Putri & Thamrin, 2019).

2.3 U2D2

U2D2 adalah suatu modul yang dapat digunakan untuk alat komunikasi pengirim perintah berupa paket instruksi terhadap *actuator dynamixel*. U2D2 merupakan alat yang memiliki fungsi untuk merubah komunikasi data dari *full duplex* menjadi *half duplex*. *Full duplex* merupakan suatu media komunikasi dua pihak yang akan mengirimkan serta menerima informasi pada waktu yang sama. *Half duplex* merupakan suatu media komunikasi dua pihak yang akan mengirimkan dan menerima informasi dengan cara bergantian (Pambudi dkk., 2018).



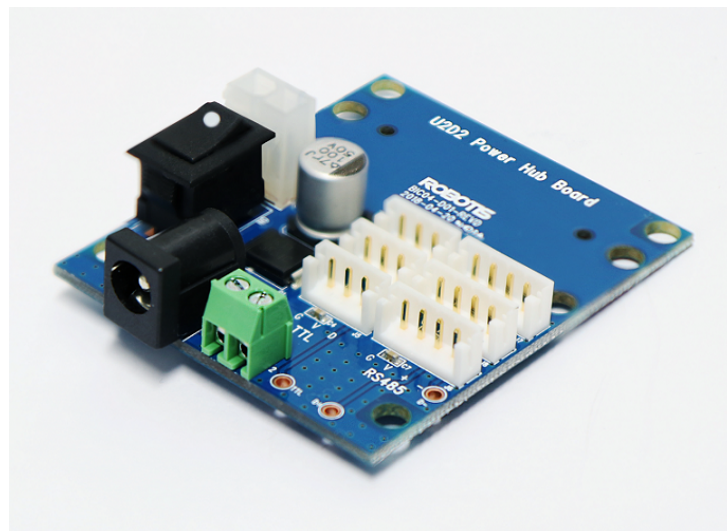
Gambar 2.5 U2D2 Board

U2D2 merupakan salah satu perangkat USB *converter* yang dapat digunakan sebagai penghubung *dynamixel* ke PC atau laptop. U2D2 dapat digunakan sebagai

pengontrol serta jalur komunikasi berbagai varian servo *dynamixel*. Pada U2D2 terdapat beberapa *port* diantara lain adalah 4 pin UART, 3 pin TTL, 4 pin TTL, status LED, dan Micro-B USB. 4 pin UART digunakan untuk mengubah USB dan UART, 3 pin TTL adalah *dynamixel port* dengan 3 pin komunikasi TTL, 4 pin TTL adalah *dynamixel port* dengan 4 pin komunikasi RS-485, status LED digunakan untuk menampilkan status *power supply*, TxD, dan RxD, dan Micro-B USB digunakan untuk menghubungkan ke PC melalui USB (Pambudi dkk., 2018).

2.4 U2D2 Power Hub

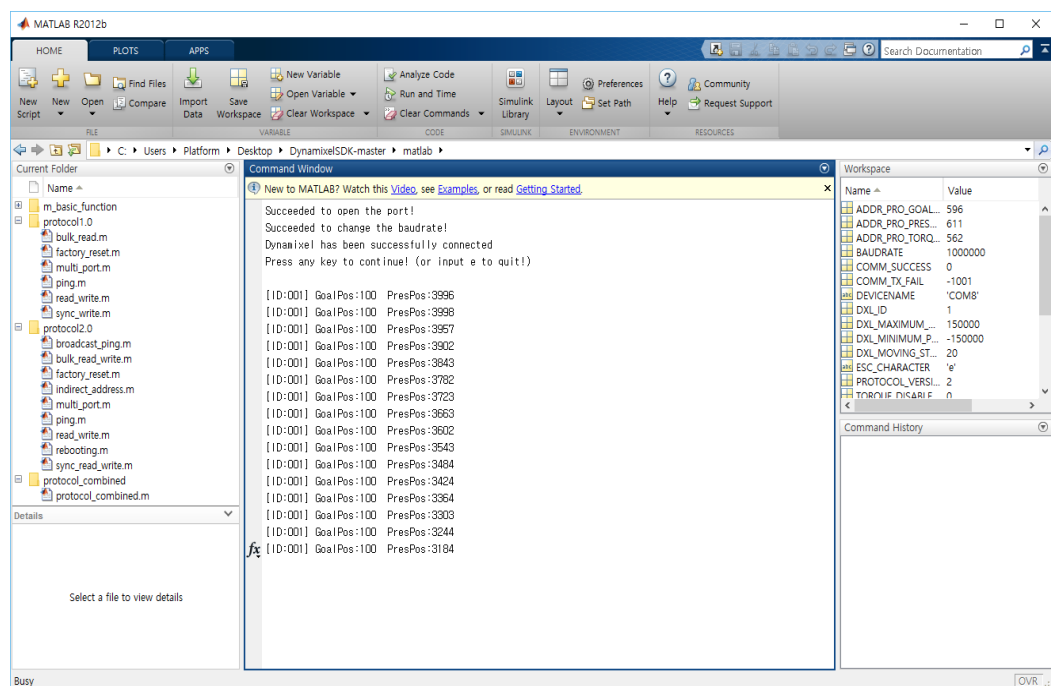
U2D2 *power hub* merupakan suatu *board* yang dapat memberikan catu daya eksternal pada servo *dynamixel*. U2D2 *power hub* merupakan *board* yang dapat memberikan suplai daya terhadap servo *dynamixel* dari SMPS *port*, konektor daya 2 pin, dan *terminal block*. *Board* U2D2 *power hub* dapat dirangkai dengan sistem *rivet* serta memiliki lubang untuk meletakkan modul U2D2. U2D2 *power hub* dipasangkan dengan U2D2 sebagai pemasok sumber daya eksternal ke *dynamixel*. Pada U2D2 *power hub* memiliki beberapa komponen diantara lain adalah *dynamixel connectors*, *power connectors*, dan TTL atau RS485 *test points* (Fadlillah dkk., 2023).



Gambar 2.6 U2D2 Power Hub

2.5 MATLAB

MATLAB merupakan suatu platform yang banyak digunakan untuk bahasa pemrograman serta mengolah angka. Selain itu, MATLAB merupakan platform pemrograman dengan bahasa yang berbasis matriks sehingga memudahkan manusia dalam menganalisis data, menciptakan pemodelan atau aplikasi, serta membuat algoritma. Terdapat beberapa fitur yang dimiliki oleh MATLAB diantaranya adalah editor dan *debugger*, toolbox, pemrograman paralel, simulink, integrasi dengan bahasa pemrograman, dan sumber daya komunitas. Pemodelan dan analisis data menggunakan MATLAB akan menghasilkan solusi dalam menghadapi berbagai tantangan (Purwoto dkk., 2020).



Gambar 2.7 Program MATLAB

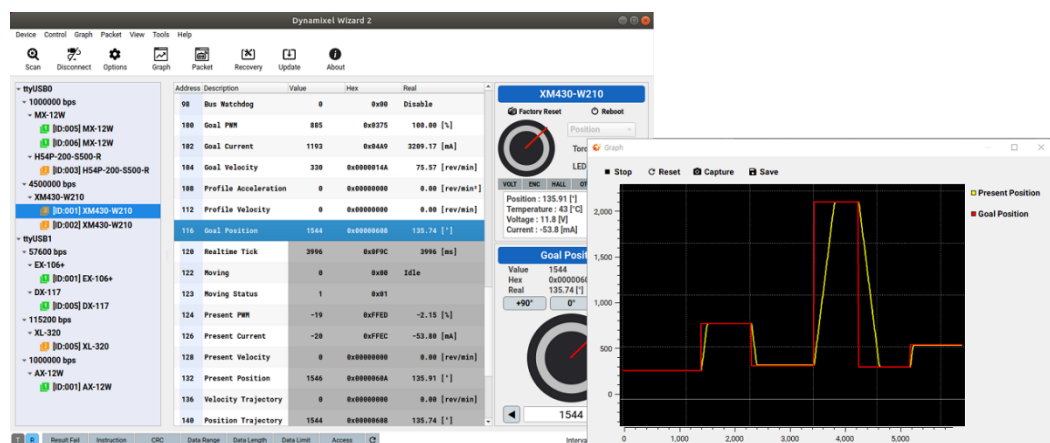
Penggunaan MATLAB pada robotik, dapat digunakan untuk mengevaluasi serta mensimulasikan kinematika robot. Selain itu, MATLAB berfungsi untuk memodelkan robot dengan fitur robotik *system toolbox*. MATLAB sudah banyak dikenal oleh suatu industri dengan tujuan untuk memudahkan perusahaan dalam mengembangkan bisnisnya. Sistem dari MATLAB bekerja untuk menjalankan

simulasi atau sebagai dasar dalam implementasi suatu prototipe pengembangan produk.

Selain itu, MATLAB dapat digunakan untuk meningkatkan peluang keberhasilan suatu teknologi baru. Kelebihan dari MATLAB adalah memberikan perhitungan *array* dan matriks yang tepat dan konsisten serta penugasan variabel yang sederhana terhadap nilai *array*. Dimana ukuran dari tipe *array* akan disesuaikan dengan *array* yang dialokasikan secara otomatis (Utomo & Munadi, 2013).

2.6 Dynamixel Wizard

Dynamixel wizard merupakan suatu aplikasi yang dibuat oleh robotis yang digunakan untuk akses baca tulis memori MCU *dynamixel*. Penggunaan *dynamixel wizard* bertujuan untuk mengetahui tingkat keberhasilan program yang dirancang. *Dynamixel wizard* yang terhubung dengan *dynamixel* akan bekerja dengan cara menguji, menentukan, mengatur dan memperbarui fitur aktuator *dynamixel*. Selain itu, *dynamixel wizard* digunakan untuk mengendalikan motor servo yang bergerak pada tiap *joint* (Kautsar & Darwito, 2015).



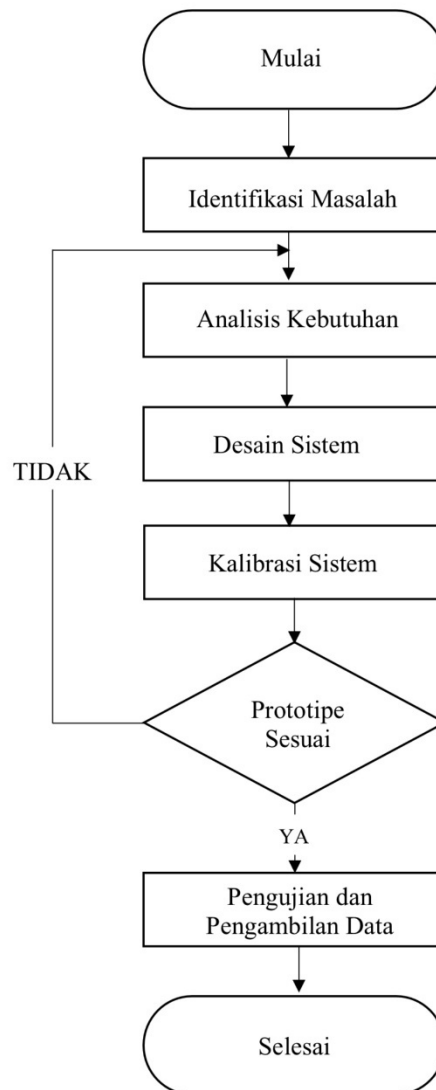
Gambar 2.8 Program *Dynamixel Wizard*

Bab III

Analisis dan Perancangan Sistem

3.1 Tahapan Penelitian

Pada Gambar 3.1 menunjukkan tahapan pembuatan sebuah prototipe perangkat robot *arm*.



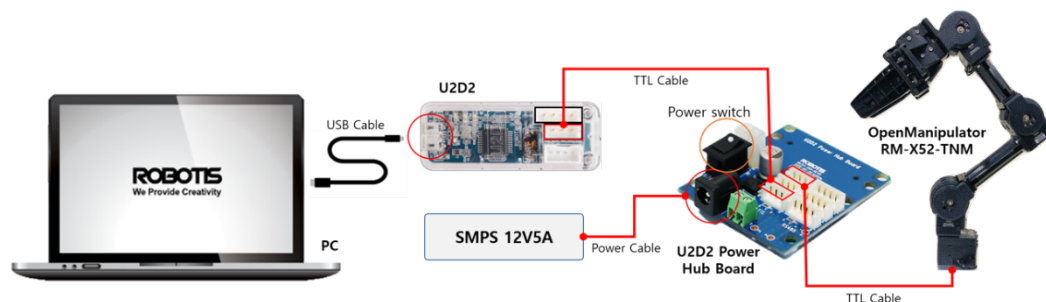
Gambar 3.1 Tahapan Pembuatan Prototipe Robot *Arm*

Pada saat melakukan identifikasi masalah, ditemukan bahwa diperlukannya robot pembantu pada laboratorium agar dapat digunakan sewaktu petugas laboratorium mengambil atau memindahkan bahan kimia berbahaya sehingga dapat

meminimalisir cedera. Melalui identifikasi masalah tersebut, dilakukan analisis kebutuhan yang dimulai dengan melakukan analisis pada perangkat lunak dan perangkat keras apa saja yang dibutuhkan dalam penelitian. Setelah melakukan analisis menyeluruh terkait perangkat-perangkat yang dibutuhkan, dilakukan proses desain sistem. Pada proses ini, dilakukan perancangan secara menyeluruh terkait sistem yang akan dibuat. Selanjutnya, dilakukan proses kalibrasi sistem dimana pada proses ini dilakukan pengetesan pada prototipe untuk mengetahui apakah prototipe yang dibuat sudah sesuai dengan desain awal. Jika prototipe sudah sesuai bisa dilanjutkan pada proses pengujian dan pengambilan data. Apabila pengetesan yang dilakukan gagal harus dilakukan analisis kebutuhan kembali untuk mengetahui apakah ada yang belum sesuai dengan tujuan awal dibuatnya prototipe robot *arm*. Setelah semua proses tersebut selesai, tahapan akhir yang harus dilakukan adalah pengujian dan pengambilan data. Pengujian dan pengambilan data ini dilakukan dengan tujuan agar mendapatkan kesimpulan dari penelitian dengan hasil yang menyeluruh serta dapat mengukur tingkat keberhasilan penelitian yang telah dilakukan.

3.2 Proses Pengerjaan

Pada penelitian ini, Peneliti berfokus pada pembuatan prototipe penggerak robot *arm*. Robot *arm* ini sendiri bertujuan untuk melakukan gerakan menyerupai tangan manusia pada sebuah lingkungan tertutup atau lingkungan yang berbahaya bagi manusia. Prototipe robot *arm* yang dipakai terdiri dari 3 komponen yang harus dirangkai dahulu yaitu U2D2, *power hub*, serta motor servo *dynamixel*. Berikut adalah skema rangkaian pada prototipe robot *arm* :



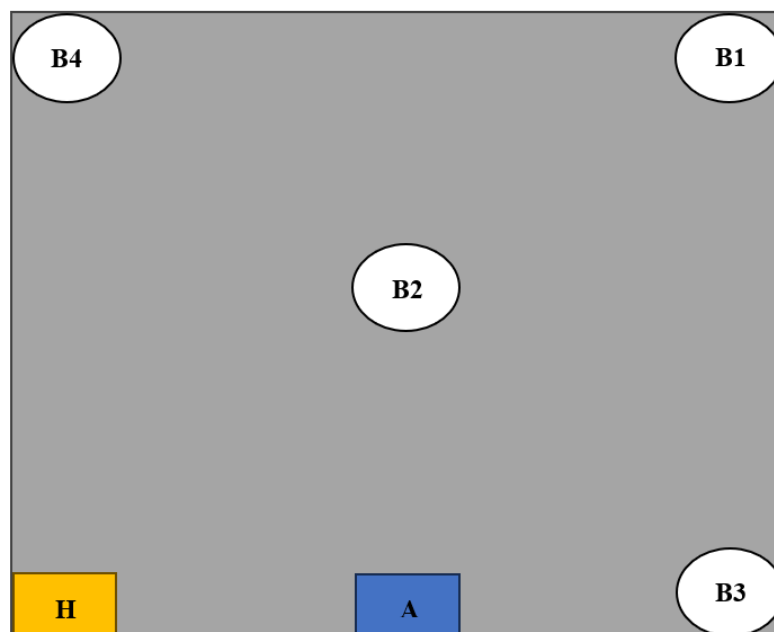
Gambar 3.2 Rangkaian Prototipe Robot *Arm*

Pada Gambar 3.2 mengilustrasikan rangkaian PC peneliti yang dilengkapi dengan program MATLAB dan *dynamixel wixard* ke sebuah kabel USB yang terhubung dengan U2D2 *board*. Kemudian terdapat U2D2 *Power Hub Board* yang dihubungkan dengan *board* U2D2 dengan kabel TTL serta sebuah power adaptor 12V memakai *jack* DC. Robot *arm* dihubungkan dengan U2D2 *Power Board* dengan sebuah kabel TTL.

Robot *arm* terdiri dari 4 buah servo *dynamixel* yang bekerja sebagai *joint*, *base*, dan *joint arm* lalu ada sebuah servo *dynamixel* yang bekerja sebagai *end effector*. Dimana antar servo tersebut disambungkan dengan menggunakan rangka yang telah disediakan, lalu antar koneksi antar servo terhubung menggunakan kabel 3 pin.

3.3 Metode Evaluasi

Pada proses pengujian prototipe, robot *arm* diharapkan mampu bergerak antar posisi yang sudah ditentukan. Posisi *base* dari robot *arm* ditunjukkan dengan huruf A. Posisi benda ditunjukkan dengan huruf B1, B2, B3, B4. Posisi *home* dari robot *arm* yang ditunjukkan dengan huruf H merupakan posisi awal dari robot *arm* setiap kali akan bergerak.



Gambar 3.3 Ilustrasi Posisi Robot *Arm*

Keterangan:

1. A merupakan *base* dari robot *arm*.
2. H merupakan posisi awal robot *arm* setiap kali akan bergerak.
3. B1 sampai dengan B4 merupakan posisi dan tujuan benda ketika dipindahkan.

3.4 Tabel Hasil Pengujian

Pengujian akan dilakukan dengan menggerakkan robot *arm* dari satu titik ke titik lainnya. Pada pengujian ini titik posisi robot *arm* dibatasi menjadi 4 titik, dimana 1 titik merupakan titik awal dan 3 titik merupakan titik tujuan gerak robot *arm*. Apabila proses gerak berhasil akan diberi tanda $\surd$ dan apabila gagal akan diberi tanda X pada tabel format penulisan hasil pengujian.

Tabel 3.1 Format Penulisan Hasil Pengujian

	B1	B1	B1	B2	B2	B2	B3	B3	B3	B4	B4	B4
Gerak	ke	ke	ke	ke	ke	ke	ke	ke	ke	ke	ke	ke
	B2	B3	B4	B1	B3	B4	B1	B2	B4	B1	B2	B3
Berhasil	-	-	-	-	-	-	-	-	-	-	-	-
Gagal	-	-	-	-	-	-	-	-	-	-	-	-

Bab IV

Hasil dan Pembahasan

4.1 Rincian Penelitian

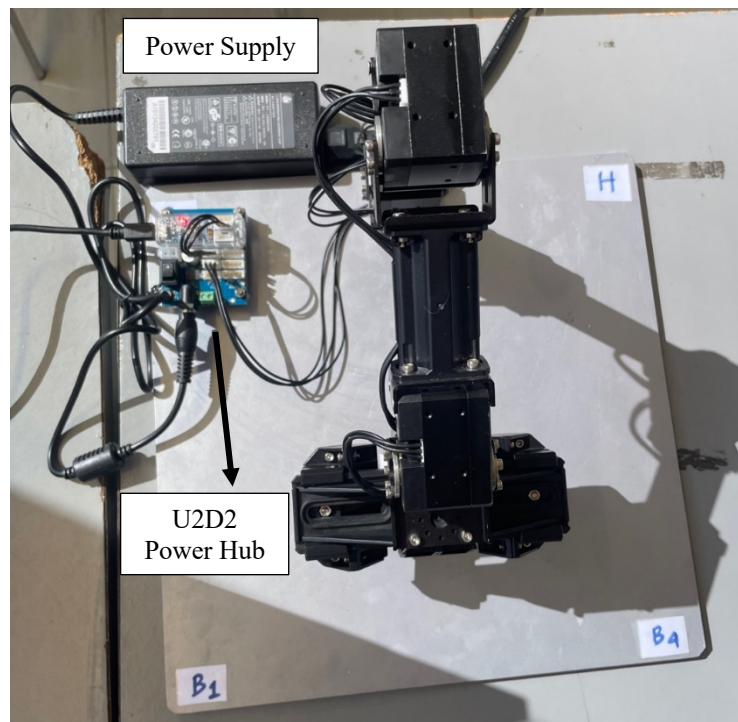
Rincian pada penelitian ini terdiri dari tempat dan waktu penelitian serta topologi alat yang digunakan dalam penelitian yang meliputi perangkat lunak dan juga perangkat keras yang digunakan.

4.1.1 Tempat dan waktu

Penelitian ini dilakukan di Laboratorium *Human Machine Interaction* (HMI) Gedung *Research and Development* Lantai 6 Universitas Ma Chung. Universitas Ma Chung sendiri beralamat di Jalan Villa Puncak Tidar Blok N No. 1 Karangwido, Kecamatan Dau, Kabupaten Malang, Jawa Timur. Penelitian ini dimulai sejak hari Senin, 3 Juni 2024 sampai dengan hari Kamis, 25 Juli 2024. Hasil akhir penelitian ini dilakukan uji pada robot *arm* agar dapat melakukan gerakan sesuai dengan instruksi yang diberikan melalui aplikasi MATLAB.

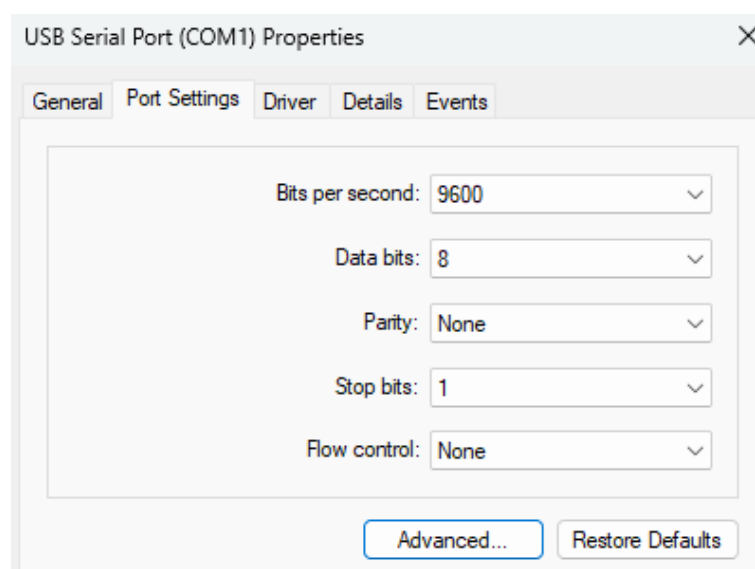
4.1.2 Alat dan komponen

Alat dan komponen yang digunakan dalam penelitian ini adalah sebuah komputer yang memiliki *port Universal Serial Bus* (USB) agar dapat dikoneksikan dengan rangkaian prototipe melalui sebuah kabel *Universal Serial Bus* (USB). Pada rangkaian robot *arm* sendiri terdiri dari sebuah U2D2 *power hub* yang terhubung dengan servo *dynamixel* yang sudah dirangkai menyerupai *arm* manusia. Kemudian terdapat juga sebuah *power adaptor* 12V 5A untuk memberi suplai tegangan pada servo *dynamixel* seperti pada Gambar 4.1.



Gambar 4.1 Rancangan Prototipe

Komputer yang digunakan memiliki perangkat lunak MATLAB dan telah memiliki *library dynamixel* SDK agar dapat berkomunikasi dengan robot *arm dynamixel*. Komunikasi yang digunakan adalah komunikasi serial dengan kecepatan 9600 *bit per second*, seperti pada Gambar 4.2.



Gambar 4.2 Konfigurasi pada *Device Manager*

Data bits merupakan jumlah bit data yang dikirimkan dalam satu frame data. Biasanya peralatan modern menggunakan 8 data bit. *Parity* merupakan cara pengecekan pada data itu sendiri apabila memiliki error. *Stop bits* adalah jumlah bit data yang digunakan untuk menandai akhir data.

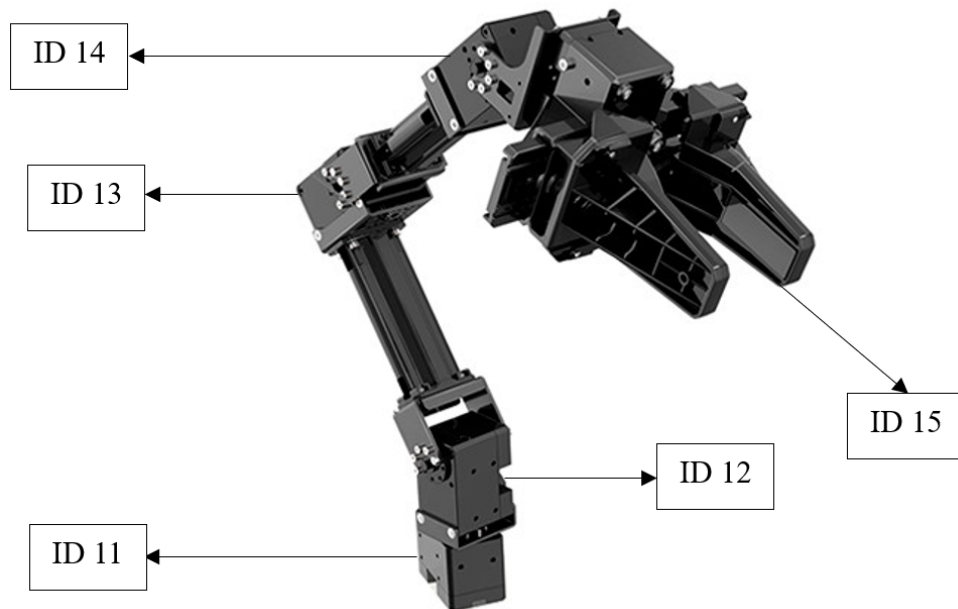
4.2 Pengambilan Data

Pengambilan data dilakukan dengan menggunakan *function* Read2Byte pada MATLAB dengan memposisikan robot *arm* sesuai dengan posisi yang telah ditentukan sebelumnya. Kemudian data posisi robot *arm* tersebut akan digunakan sebagai dasar untuk melakukan perbaikan posisi robot *arm* sehingga posisinya lebih presisi dan *arm* bisa bergerak dengan lebih lancar. Dari proses ini didapatkan data seperti pada Tabel 4.1. *Range* servo 0 - 4095 menurut *datasheet* dari ROBOTIS.

Tabel 4.1 Data Posisi Robot *Arm*

ID Servo	Posisi <i>Home</i>	Posisi B1	Posisi B2	Posisi B3	Posisi B4
ID 11	3084	1730	2050	1020	2350
ID 12	1500	2600	1990	1880	2600
ID 13	2450	1290	2300	2450	1290
ID 14	2770	3120	2800	2770	3120
ID 15	1500	1800	1800	1800	1800

Pada Gambar 4.3 menjelaskan nomor ID servo *dynamixel* yang dimaksud pada Tabel 4.1.



Gambar 4.3 Penjelasan ID Servo *Dynamixel*

4.3 Hasil Pengujian

Penelitian dilakukan dengan menguji gerakan robot *arm* sesuai dengan posisi yang telah ditentukan seperti pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian Gerakan Robot *Arm*

	B1	B1	B1	B2	B2	B2	B3	B3	B3	B4	B4	B4
Gerak	ke	ke	ke	ke	ke	ke	ke	ke	ke	ke	ke	ke
	B2	B3	B4	B1	B3	B4	B1	B2	B4	B1	B2	B3
Berhasil	√	√	√	√	√	√	√	√	√	√	√	√
Gagal	-	-	-	-	-	-	-	-	-	-	-	-

Gambar 4.4 menunjukkan potongan *coding* yang menjelaskan proses pengambilan data posisi dari masing-masing servo sesuai ID yang sudah diberikan pada robot *arm*.

```

1  dxl_present_position_ID11      =      read2ByteTxRx(port_num,
2  PROTOCOL_VERSION, DXL_ID11, ADDR_MX_PRESENT_POSITION);
3      dxl_comm_result_ID11      =      getLastTxRxResult(port_num,
4  PROTOCOL_VERSION);
5      dxl_error_ID11            =      getLastRxPacketError(port_num,
6  PROTOCOL_VERSION);
7      if dxl_comm_result_ID11 ~= COMM_SUCCESS
8          fprintf('%s\n',      getTxRxResult(PROTOCOL_VERSION,
9  dxl_comm_result_ID11));
10     elseif dxl_error_ID11 ~= 0
11         fprintf('%s\n',      getRxPacketError(PROTOCOL_VERSION,
12  dxl_error_ID11));
13     end
14     fprintf('[ID:%03d]  GoalPos:%03d      PresPos:%03d\n',  DXL_ID11,
15  dxl_goal_position_ID11(index), dxl_present_position_ID11);

```

Gambar 4.4 Data Posisi Robot *Arm*

Keberhasilan pengujian gerakan robot *arm* pada Tabel 4.2 kemudian dapat dilihat melalui beberapa gambar berikut :



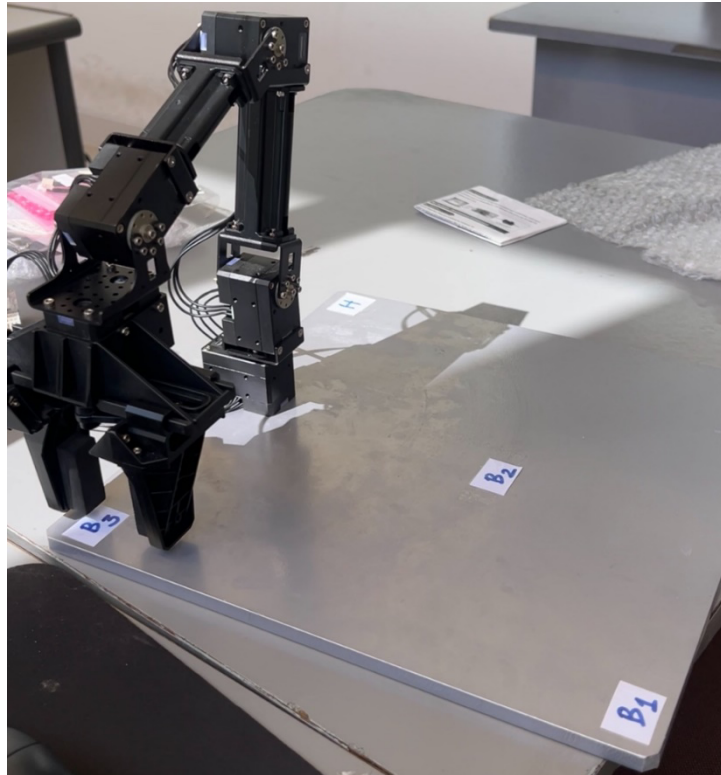
Gambar 4.5 Robot *Arm* pada Posisi *Home*



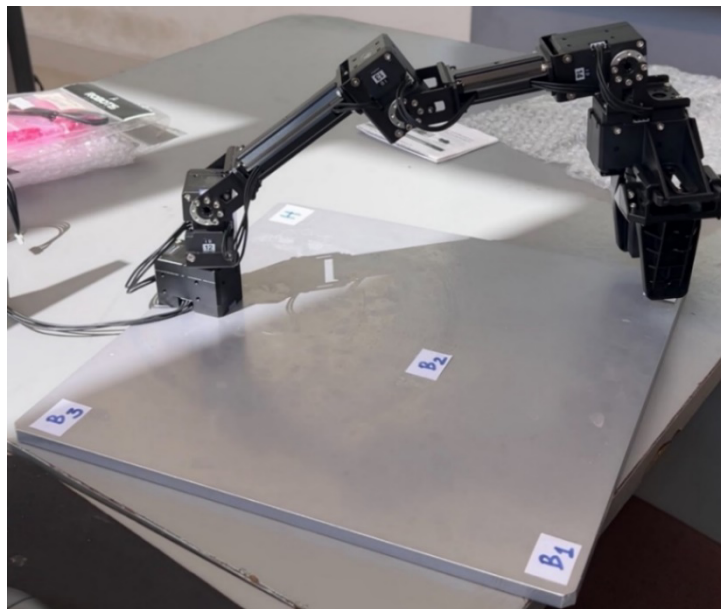
Gambar 4.6 Robot *Arm* pada Posisi B1



Gambar 4.7 Robot *Arm* pada Posisi B2



Gambar 4.8 Robot *Arm* pada Posisi B3



Gambar 4.9 Robot *Arm* pada Posisi B4

4.4 Diskusi

Penelitian ini dilakukan dengan tujuan agar robot *arm* mampu bergerak menyerupai tangan manusia. Pada penelitian ini, robot *arm* sudah mampu bergerak sesuai dengan lokasi yang telah ditentukan. Gerakan antar *joint* juga sudah bisa bergerak dengan baik. Namun, gerakan dari robot *arm* ini masih kaku dan sedikit kasar sehingga robot *arm* belum bisa memegang objek seperti yang diinginkan.

Pada program robot *arm*, kontrol gerakan masih harus dilakukan secara manual dengan merubah indeks posisi pada *.m file*. Gerakan yang masih kaku dan sedikit kasar ini kemungkinan disebabkan oleh *function* yang digunakan ketika menggerakkan robot. Agar robot *arm* mampu bergerak dengan lebih halus dan bisa memegang objek dengan baik mungkin perlu ditambahkan posisi transit. Selain itu, perlu ditambahkan juga sebuah *function* pada *.m file* agar dapat menggerakkan robot *arm* sebagai satu rangkaian gerakan sehingga gerakannya lebih halus.

Bab V

Simpulan dan Saran

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan, maka kesimpulan yang dapat ditarik adalah penelitian ini berhasil membuat sebuah prototipe pengontrol gerakan robot OpenManipulator dengan aplikasi MATLAB. Prototipe yang dikembangkan sudah mampu melakukan fungsi gerak sesuai dengan lokasi gerak yang ditentukan meskipun dilakukan lewat *.m file* pada MATLAB. Prototipe masih belum bisa memegang objek karena setiap kali gerak mengambil objek, objek terlempar karena tersenggol *gripper* pada robot *arm*, perlu dicari sebuah cara untuk mengurangi kecepatan gerak robot arm.

5.2 Saran

Saran yang dapat diberikan untuk mengembangkan robot *arm* pada penelitian selanjutnya adalah sebagai berikut :

1. Pengembangan *software* untuk gerakan robot *arm* masih perlu dilakukan penelitian lebih lanjut agar gerakan robot *arm* menjadi lebih halus dan terarah.
2. Robot *arm* dapat memindahkan objek dengan bentuk yang menyerupai benda yang digunakan dalam kehidupan sehari-hari.
3. Gerakan robot *arm* bisa dilakukan secara langsung tanpa harus merubah indeks posisi pada *.m file* MATLAB.

DAFTAR PUSTAKA

- Abidin, M., & Suprianto, B. (2020). Rancang Bangun Trainable Servo Robotic Arm 4 DOF (Degree of Freedom). *Jurnal Teknik Elektro*, 9(2), 321-329. Retrieved Juni 2024
- Ayega, I. H., Tamba, T. A., & Arthaya, B. M. (2023, Juli). Rancang Bangun Purwarupa Manipulator Lengan. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(3), 796-810. Retrieved Juni 2024
- Fadlillah, R., Rusdinar, A., & Siregar, S. (2023, Oktober). Implementasi Inverse Kinematics Untuk Pola Jalan Robot Penari Humanoid. *e-Proceeding of Engineering*, 10(5), 3967-3973. Retrieved Mei 2024
- Insani, D. N. (2023, Juni 23). *Teknologi Robotik dan Artificial Intelligence untuk Dunia Pendidikan*. Retrieved Mei 2024, from BalaiTekKomDik: btkp-diy.or.id
- Kautsar, S., & Darwito, P. A. (2015, Februari 6-8). Pergerakan Robot Lengan Pengambil Objek Dengan Sistem Perekam Gerak Berbasis Komputer. *Seminar Nasional Teknologi Informasi dan Multimedia 2015*, 37-42. Retrieved Mei 2024
- Muslim, A. I. (2022, Oktober). ROBOT. Retrieved Mei 2024
- Oktama, R., Maulana, R., & Setyawan, G. E. (2018, Agustus). Implementasi Robot Lengan Pemindah Barang 3 DOF Menggunakan Metode Inverse Kinematics. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(8), 2810-2816. Retrieved Mei 2024
- Pambudi, N. S., Wiharta, D. M., & Sastra, N. P. (2018, Desember). Analisa Kestabilan Gerakan Statis pada Robot Humanoid. *E-Journal SPEKTRUM*, 5(2), 253-262. Retrieved Mei 2024
- Pamungkas, D. S., & Noviansyah, M. S. (2021). Simulator Robot Lengan Dua Derajat Kebebasan. *Seminar Nasional Terapan Riset Inovatif (SENTRINOV) ke-VII : Series: Engineering and Science*, 7(1), 51-57. Retrieved Mei 2024

- Purwoto, B. H., Rhokhim, D. R., & Indraswari, D. (2020, September). Pemodelan Robot Kinematik Manipulator Menggunakan Matlab. *Emitor: Jurnal Teknik Elektro*, 20(2), 141-146. Retrieved Mei 2024
- Putri, U. W., & Thamrin. (2019, September). Perancangan Pergerakan Kaki Robot Humanoid Menggunakan Servo Dynamixel Berbasis OpenCM 9.04. *Jurnal Vokasional Teknik Elektronika dan Informatika (VOTEKNIKA)*, 7(3), 76-84. Retrieved Mei 2024
- Rahman, F., Faridah, Nur, A. I., & Makkaraka, A. N. (2020, April). Rancang Bangun Prototipe Manipulator Lengan Robot Menggunakan Motor Servo Berbasis Mikrokontroler. *ILTEK : Jurnal Teknologi*, 15(1), 42-46. Retrieved Mei 2024
- Sari, R. P. (2024, Maret 24). *Mengenal Apa itu Robotik? Pengertian dan Contohnya*. Retrieved Mei 2024, from Cloud Computing Indonesia: www.cloudcomputing.id
- Utomo, B., & Munadi. (2013). Analisa Forward dan Inverse Kinematics pada Simulator Arm Robot 5 Derajat Kebebasan. *Jurnal Teknik Mesin S-1*, 1(3), 11-20. Retrieved Mei 2024

LAMPIRAN

Lampiran A. Kode Sumber MATLAB

```
clc;  
clear all;
```

%fungsi untuk membaca library yang digunakan

```
lib_name = '';  
  
if strcmp(computer, 'PCWIN')  
    lib_name = 'dxl_x86_c';  
elseif strcmp(computer, 'PCWIN64')  
    lib_name = 'dxl_x64_c';  
elseif strcmp(computer, 'GLNX86')  
    lib_name = 'libdxl_x86_c';  
elseif strcmp(computer, 'GLNXA64')  
    lib_name = 'libdxl_x64_c';  
elseif strcmp(computer, 'MACI64')  
    lib_name = 'libdxl_mac_c';  
end
```

% Load Libraries

```
if ~libisloaded(lib_name)  
    [notfound, warnings] = loadlibrary(lib_name,  
    'dynamixel_sdk.h', 'addheader', 'port_handler.h',  
    'addheader', 'packet_handler.h');  
end
```

%fungsi untuk memberikan default value spek dynamixel yang berbeda tiap model servo yang digunakan kontrol

table address (SESUAIKAN TIPE SERVO)

```
ADDR_MX_TORQUE_ENABLE      = 64;
ADDR_MX_GOAL_POSITION      = 116;
ADDR_MX_PRESENT_POSITION   = 132;
```

%protocol version

```
PROTOCOL_VERSION           = 2.0;
```

%deklarasi ID servo robot arm

```
DXL_ID11                   = 11;
DXL_ID12                   = 12;
DXL_ID13                   = 13;
DXL_ID14                   = 14;
DXL_ID15                   = 15;
```

%konfigurasi serial port yang digunakan, sesuaikan dengan tipe servo

```
BAUDRATE                   = 1000000;
DEVICENAME                 = 'COM1';
```

```
TORQUE_ENABLE              = 1;           % Value for
enabling the torque
TORQUE_DISABLE              = 0;           % Value for
disabling the torque
```

%fungsi untuk deklarasi minimum dan maksimum value dari servo dynamixel value berbeda untuk setiap tipe servo dynamixel

```
DXL_MINIMUM_POSITION_VALUE_ID11 = 100;
```

```
DXL_MAXIMUM_POSITION_VALUE_ID11 = 4000
```

```
DXL_MINIMUM_POSITION_VALUE_ID12 = 100
```

```
DXL_MAXIMUM_POSITION_VALUE_ID12 = 4000;
```

```
DXL_MINIMUM_POSITION_VALUE_ID13 = 100;
```

```
DXL_MAXIMUM_POSITION_VALUE_ID13 = 4000;
```

```
DXL_MINIMUM_POSITION_VALUE_ID14 = 100;
```

```
DXL_MAXIMUM_POSITION_VALUE_ID14 = 4000;
```

```
DXL_MINIMUM_POSITION_VALUE_ID15 = 100;
```

```
DXL_MAXIMUM_POSITION_VALUE_ID15 = 4000;
```

```
DXL_MOVING_STATUS_THRESHOLD = 10;
```

%fungsi untuk exit dari program

```
ESC_CHARACTER = 'e'; % Key for  
escaping loop
```

```
COMM_SUCCESS = 0; % Communication  
Success result value
```

```
COMM_TX_FAIL = -1001; % Communication  
Tx Failed
```

```
port_num = portHandler(DEVICENAME);  
packetHandler();
```

```
index = 1;
```

```
dxl_comm_result = COMM_TX_FAIL;           % Communication
result
```

%deklarasi array untuk target position

```
%dxl_target_position_IDXX = [HOME B1 B2 B3 B4];

dxl_target_position_ID11 = [3084 1730 2050 1020 2350];
%%id11=base
dxl_target_position_ID12 = [1500 2600 1990 1880 2600];
dxl_target_position_ID13 = [2450 1290 2300 2450 1290];
dxl_target_position_ID14 = [2770 3120 2800 2770 3120];
dxl_target_position_ID15 = [1500 1800 1800 1800 1800];
%%id15=gripper objek

dxl_error = 0;                               % Dynamixel
error
dxl_present_position = 0;                     % Present
position
```

%fungsi open port serial dan set baudrate

% Open port

```
if (openPort(port_num))
    fprintf('Succeeded to open the port!\n');
else
    unloadlibrary(lib_name);
    fprintf('Failed to open the port!\n');
    input('Press any key to terminate...\n');
    return;
end
```

% Set port baudrate

```
if (setBaudRate(port_num, BAUDRATE))
    fprintf('Succeeded to change the baudrate!\n');
else
    unloadlibrary(lib_name);
    fprintf('Failed to change the baudrate!\n');
    input('Press any key to terminate...\n');
    return;
end
```

%fungsi untuk enable dynamixel torque agar robot arm bisa kaku

% Enable Dynamixel Torque_ID11

```
writelByteTxRx(port_num,  PROTOCOL_VERSION,  DXL_ID11,
ADDR_MX_TORQUE_ENABLE, TORQUE_ENABLE);
dxl_comm_result_ID11      =    getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error_ID11           =    getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID11 ~= COMM_SUCCESS
    fprintf('%s\n',    getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result_ID11));
elseif dxl_error_ID11 ~= 0
    fprintf('%s\n',    getRxPacketError(PROTOCOL_VERSION,
dxl_error_ID11));
else
    fprintf('Dynamixel  ID11  has  been  successfully
connected \n');
```

end

% Enable Dynamixel Torque_ID12

```
writelByteTxRx(port_num,  PROTOCOL_VERSION,  DXL_ID12,
ADDR_MX_TORQUE_ENABLE, TORQUE_ENABLE);
dxl_comm_result_ID12    =    getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error_ID12          =    getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID12 ~= COMM_SUCCESS
    fprintf('%s\n',      getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result_ID12));
elseif dxl_error_ID12 ~= 0
    fprintf('%s\n',  getRxPacketError(PROTOCOL_VERSION,
dxl_error_ID12));
else
    fprintf('Dynamixel  ID12  has  been  successfully
connected \n');
end
```

% Enable Dynamixel Torque_ID13

```
writelByteTxRx(port_num,  PROTOCOL_VERSION,  DXL_ID13,
ADDR_MX_TORQUE_ENABLE, TORQUE_ENABLE);
dxl_comm_result_ID13    =    getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error_ID13          =    getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID13 ~= COMM_SUCCESS
    fprintf('%s\n',      getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result_ID13));
elseif dxl_error_ID13 ~= 0
    fprintf('%s\n',  getRxPacketError(PROTOCOL_VERSION,
dxl_error_ID13));
```

```

else
    fprintf('Dynamixel ID13 has been successfully
connected \n');
end

```

% Enable Dynamixel Torque_ID14

```

write1ByteTxRx(port_num,  PROTOCOL_VERSION,  DXL_ID14,
ADDR_MX_TORQUE_ENABLE, TORQUE_ENABLE);
dxl_comm_result      =      getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error            =      getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result ~= COMM_SUCCESS
    fprintf('%s\n',      getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));
elseif dxl_error ~= 0
    fprintf('%s\n',  getRxPacketError(PROTOCOL_VERSION,
dxl_error));
else
    fprintf('Dynamixel ID14 has been successfully
connected \n');
end

```

% Enable Dynamixel Torque_ID15

```

write1ByteTxRx(port_num,  PROTOCOL_VERSION,  DXL_ID15,
ADDR_MX_TORQUE_ENABLE, TORQUE_ENABLE);
dxl_comm_result      =      getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error            =      getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result ~= COMM_SUCCESS

```

```

        fprintf('%s\n',      getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));
elseif dxl_error ~= 0
        fprintf('%s\n',  getRxPacketError(PROTOCOL_VERSION,
dxl_error));
else
        fprintf('Dynamixel  ID15  has  been  successfully
connected \n');
end

```

%fungsi untuk melakukan pemindahan posisi robot arm

```

while 1
    if input('Press any key to continue! (or input e to
quit!)\n', 's') == ESC_CHARACTER
        break;
    end

```

% Write Goal Position (ID11)

```

    write4ByteTxRx(port_num,          PROTOCOL_VERSION,
DXL_ID11,          ADDR_MX_GOAL_POSITION,
dxl_target_position_ID11(1));
    dxl_comm_result_ID11 = getLastTxRxResult(port_num,
PROTOCOL_VERSION);
    dxl_error_ID11      =  getLastRxPacketError(port_num,
PROTOCOL_VERSION);
    if dxl_comm_result_ID11 ~= COMM_SUCCESS
        fprintf('%s\n', getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));

```

```

elseif dxl_error_ID11 ~= 0
    fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error));
end

```

% Write Goal Position (ID12)

```

write4ByteTxRx(port_num,          PROTOCOL_VERSION,
DXL_ID12,          ADDR_MX_GOAL_POSITION,
dxl_target_position_ID12(1));
dxl_comm_result_ID12 = getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error_ID12      =  getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID12 ~= COMM_SUCCESS
    fprintf('%s\n', getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));
elseif dxl_error_ID12 ~= 0
    fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error));    end

```

% Write Goal Position (ID13)

```

write4ByteTxRx(port_num,          PROTOCOL_VERSION,
DXL_ID13,          ADDR_MX_GOAL_POSITION,
dxl_target_position_ID13(1));
dxl_comm_result_ID13 = getLastTxRxResult(port_num,
PROTOCOL_VERSION);
dxl_error_ID13      =  getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID13 ~= COMM_SUCCESS

```

```

        fprintf('%s\n', getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));
    elseif dxl_error_ID13 ~= 0
        fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error));
    end

```

% Write Goal Position (ID14)

```

        write4ByteTxRx(port_num,          PROTOCOL_VERSION,
DXL_ID14,          ADDR_MX_GOAL_POSITION,
dxl_target_position_ID14(1));
    dxl_comm_result_ID14 = getLastTxRxResult(port_num,
PROTOCOL_VERSION);
    dxl_error_ID14      =  getLastRxPacketError(port_num,
PROTOCOL_VERSION);
    if dxl_comm_result_ID14 ~= COMM_SUCCESS
        fprintf('%s\n', getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));
    elseif dxl_error_ID14 ~= 0
        fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error));
    end

```

% Write Goal Position (ID15)

```

        write4ByteTxRx(port_num,          PROTOCOL_VERSION,
DXL_ID15,          ADDR_MX_GOAL_POSITION,
dxl_target_position_ID15(1));
    dxl_comm_result_ID15 = getLastTxRxResult(port_num,
PROTOCOL_VERSION);
    dxl_error_ID15      =  getLastRxPacketError(port_num,
PROTOCOL_VERSION);

```

```

        if dxl_comm_result_ID15 ~= COMM_SUCCESS
            fprintf('%s\n', getTxRxResult(PROTOCOL_VERSION,
dxl_comm_result));
        elseif dxl_error_ID15 ~= 0
            fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error));
        end

```

%fungsi untuk membaca posisi saat ini dari servo robot arm

% Read Present Position (ID11)

```

        dxl_present_position_ID11 =
read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL_ID11,
ADDR_MX_PRESENT_POSITION);
        dxl_comm_result_ID11 =
getLastTxRxResult(port_num, PROTOCOL_VERSION);
        dxl_error_ID11 = getLastRxPacketError(port_num,
PROTOCOL_VERSION);
        if dxl_comm_result_ID11 ~= COMM_SUCCESS
            fprintf('%s\n',
getTxRxResult(PROTOCOL_VERSION, dxl_comm_result_ID11));
        elseif dxl_error_ID11 ~= 0
            fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error_ID11));
        end
        fprintf('[ID:%03d] GoalPos:%03d PresPos:%03d\n',
DXL_ID11, dxl_goal_position_ID11(index),
dxl_present_position_ID11);

```

% Read Present Position (ID12)

```
dxl_present_position_ID12 = read2ByteTxRx(port_num,
PROTOCOL_VERSION, DXL_ID12, ADDR_MX_PRESENT_POSITION);
dxl_comm_result_ID12 =
getLastTxRxResult(port_num, PROTOCOL_VERSION);
dxl_error_ID12 = getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID12 ~= COMM_SUCCESS
    fprintf('%s\n',
getTxRxResult(PROTOCOL_VERSION, dxl_comm_result_ID12));
elseif dxl_error ~= 0
    fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error_ID12));
end
fprintf('[ID:%03d]    GoalPos:%03d        PresPos:%03d\n',
DXL_ID12,                dxl_goal_position_ID12(index),
dxl_present_position_ID12);
```

% Read Present Position (ID13)

```
dxl_present_position_ID13 = read2ByteTxRx(port_num,
PROTOCOL_VERSION, DXL_ID13, ADDR_MX_PRESENT_POSITION);
dxl_comm_result_ID13 =
getLastTxRxResult(port_num, PROTOCOL_VERSION);
dxl_error_ID13 = getLastRxPacketError(port_num,
PROTOCOL_VERSION);
if dxl_comm_result_ID13 ~= COMM_SUCCESS
    fprintf('%s\n',
getTxRxResult(PROTOCOL_VERSION, dxl_comm_result_ID13));
elseif dxl_error ~= 0
    fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error_ID13));
```

```

        end
fprintf('[ID:%03d]    GoalPos:%03d        PresPos:%03d\n',
DXL_ID13,                dxl_goal_position_ID13(index),
dxl_present_position_ID13);

```

% Read Present Position (ID14)

```

    dxl_present_position_ID14  =  read2ByteTxRx(port_num,
PROTOCOL_VERSION, DXL_ID14, ADDR_MX_PRESENT_POSITION);
    dxl_comm_result_ID14      =
getLastTxRxResult(port_num, PROTOCOL_VERSION);
    dxl_error_ID14 = getLastRxPacketError(port_num,
PROTOCOL_VERSION);
    if dxl_comm_result_ID14 ~= COMM_SUCCESS
        fprintf('%s\n',
getTxRxResult(PROTOCOL_VERSION, dxl_comm_result_ID14));
    elseif dxl_error ~= 0
        fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error_ID14));
    end
fprintf('[ID:%03d]    GoalPos:%03d        PresPos:%03d\n',
DXL_ID14,                dxl_goal_position_ID14(index),
dxl_present_position_ID14);

```

% Read Present Position (ID15)

```

    dxl_present_position_ID15  =  read2ByteTxRx(port_num,
PROTOCOL_VERSION, DXL_ID15, ADDR_MX_PRESENT_POSITION);
    dxl_comm_result_ID15      =
getLastTxRxResult(port_num, PROTOCOL_VERSION);
    dxl_error_ID15 = getLastRxPacketError(port_num,
PROTOCOL_VERSION);

```

```

        if dxl_comm_result_ID15 ~= COMM_SUCCESS
            fprintf('%s\n',
getTxRxResult(PROTOCOL_VERSION, dxl_comm_result_ID15));
        elseif dxl_error ~= 0
            fprintf('%s\n',
getRxPacketError(PROTOCOL_VERSION, dxl_error_ID15));
        end
        fprintf('[ID:%03d]    GoalPos:%03d        PresPos:%03d\n',
DXL_ID15,                dxl_goal_position_ID15(index),
dxl_present_position_ID15);

end

```

%fungsi untuk melakukan penghentian program yang berjalan

% Close Port

```
closePort(port_num);
```

% Unload Library

```
unloadlibrary(lib_name);
```

```
fclose all;
```

```
clear all;
```