**IMPLEMENTASI MOTOR STEPPER BIPOLAR**

**PADA *ARM MANIPULATOR 6 DOF* DENGAN PENGENDALI**

***MASTER CONTROLLER***

**PROYEK AKHIR**

Disusun sebagai salah satu syarat untuk

menyelesaikan pendidikan Diploma III

**Oleh:**

Riska Wulandari

219341044



**PROGRAM STUDI TEKNIK MEKATRONIKA**

**JURUSAN TEKNIK OTOMASI MANUFAKTUR DAN MEKATRONIKA**

**POLITEKNIK MANUFAKTUR BANDUNG**

**2022**

**LEMBAR PENGESAHAN**

**IMPLEMENTASI MOTOR STEPPER BIPOLAR**

**PADA *ARM MANIPULATOR 6 DOF* DENGAN PENGENDALI**

***MASTER CONTROLLER***

**Oleh:**

Riska Wulandari

219341044

Program Studi Teknik Mekatronika

Jurusan Teknik Otomasi Manufaktur dan Mekatronika

Politeknik Manufaktur Bandung

Menyetujui

Tim Pembimbing

|  |  |
| --- | --- |
| Pembimbing I,  **Dr. Setyawan Ajie Sukarno, S.S.T., M.T., M.Sc. Eng**  **NIP. 198004282008101001** | Pembimbing II,  **Dr. Susetyo Bagas Bhaskoro, S.S.T., M.T.**  **NIP. 198706222015041002** |

# ABSTRAK

*Arm Manipulator* merupakan salah satu jenis robot yang banyak digunakan pada dunia industri. *Arm Manipulator* adalah sebuah robot yang menggunakan motor penggerak dengan bentuk seperti tangan untuk mempermudah pekerjaan manusia di bidang industry. Penerapan Arm Manipulator yang telah banyak digunakan yaitu untuk pengelasan, pengecatan, pengeboran, dan pemindahan. Pada Proyek Akhir ini dibuat sebuah Arm Manipulator 6 DoF atau dengan kata lain adalah lengan robot yang memiliki 6 derajat kebebasan. Arm manipulator yang dibuat ini menggunakan penggerak motor stepper yang dikendalikan oleh joystick yang disebut dengan master controller. Pengendaliannya dilakukan secara remote/jarak jauh menggunakan komunikasi serial. Dalam hal ini, dilakukan pengujian yang bertujuan untuk mengetahui bagaimana implementasi stepper motor pada arm manipulator 6 DoF serta mengetahui keakurasian motor stepper berdasarkan pulsa yang diberikan. Hasil yang didapat dari pengujian tersebut yaitu error pada *joint* 1 sebesar 0,27%, *joint* 2 sebesar 0,4%, *joint* 3 sebesar 0,13%, *joint* 4 sebesar 0,09%, *joint* 5 sebesar 0,15%, dan *joint* 6 sebesar 0,03%. Dari pengujian tehadap keenam *joint* tersebut didapat sebuah rata-rata error keseluruhan yaitu sebesar 0,14%.

**Kata kunci** : *Arm Manipulator, Master Controller*, motor stepper

# *ABSTRACT*

*Arm Manipulator is one type of robot that is widely used in the industrial world. Arm Manipulator is a robot that uses a motor with a shape like a hand to facilitate human work in industry. Arm Manipulator applications that have been widely used are for welding, painting, drilling, and moving. In this final project, a 6 DoF Arm Manipulator or in other words is a robotic arm that has 6 degrees of freedom. This arm manipulator uses a stepper motor driven by a joystick called the master controller. Control is done remotely using serial communication. In this case, a test is carried out which aims to find out how to implement a stepper motor on a 6 DoF arm manipulator and determine the accuracy of the stepper motor based on the pulse given. The results obtained from the test are the error at joint 1 is 0.27%, joint 2 is 0.4%, joint 3 is 0.13%, joint 4 is 0.09%, joint 5 is 0.15%, and joint 6 by 0.03%. From the testing of the six joints, an overall average error of 0.14% was obtained*.

***Keywords*** *: Arm Manipulator, Mater Controller, Stepper Motor*

# KATA PENGANTAR

Segala puji bagi Allah yang hanya kepadaNya kami memuji, memohon pertolongan, dan mohon keampunan. Kami berlindung kepadaNya dari kekejian diri dan kejahatan amalan kami. Barang siapa yang diberi petunjuk oleh Allah maka tidak ada yang dapat menyesatkan, dan barang siapa yang tersesat dari jalanNya maka tidak ada yang dapat memberinya petunjuk. Dan aku bersaksi bahwa tiada sembahan yang berhak disembah melainkan Allah saja, yang tiada sekutu bagi-Nya. Dan aku bersaksi bahwa Muhammad adalah hamba-Nya dan Rasul-Nya.

Atas petunjukan dan pertolongan-Nya, Alhamdlillah penulis dapat menyelesaikan proyek akhir ini dengan judul: “IMPLEMENTASI STEPPER MOTOR BIPOLAR PADA *ARM MANIPULATOR 6 DOF* DENGAN PENGENDALI *MASTER CONTROLLER*”.

Proyek akhir dibuat dalam rangka memenuhi salah satu syarat menyelesaikan pendidikan Diploma III pada Program Studi Teknik Mekatronika di Politeknik Manufaktur Bandung.

Terselesaikannya proyek akhir ini tidak terlepas dari bantuan banyak pihak, sehingga pada kesempatan ini dengan segala kerendahan hati dan penuh rasa hormat penulis menghaturkan terima kasih yang sebesar-besamya bagi semua pihak yang telah memberikan bantuan moril maupun materil baik langsung maupun tidak langsung dalam penyusunan skripsi ini hingga selesai, terutama kepada yang saya hormati:

1. Direktur Politeknik Manufaktur Bandung, Bapak Mohammad Nurdin, S.T., M.A.B.
2. Ketua Jurusan Teknik Otomasi Manufaktur dan Mekatronika, Bapak Ismail Rokhim, ST., MT.
3. Ketua Program Studi D-III Teknik Mekatronika, Ibu Siti Aminah, S.T., M.T
4. Kedua Pembimbing proyek akhir Bapak Dr. Setyawan Ajie Sukarno, S.S.T., M.T., M.Sc. dan Bapak Susetyo Bagas Bhaskoro, S.S.T., M.T.
5. Panitia proyek akhir Ibu Hilda Khoirunnisa. S.Tr.T., M.Sc.Eng, dan Bapak Afaf Fadhil Rifa’i, S.T., M.T.
6. Teristimewa kepada Orang Tua serta keluarga besar penulis yang selalu mendoakan, memberikan motivasi dan pengorbanannya baik dari segi moril, materi kepada penulis sehingga penulis dapat menyelesaikan proyek akhir ini.
7. Rekan kelompok 9 Raihan Ramadhan, Fachri Maulana Fulchan serta rekan kelas AEB 2019 dan rekan – rekan mahasiswa AE 2019 yang saling mendukung dan menyemangati dalam proses penyelesaian proyek akhir ini

Akhir kata penulis menyadari bahwa dalam penulisan proyek akhir ini masih jauh dari kesempurnaan. Karena itu, penulis memohon saran dan kritik yang sifatnya membangun demi kesempurnaannya dan semoga bermanfaat bagi kita semua. Aamiiiin Ya Robbal Alamin.

Bandung, 20 Juli 2022

Penulis

# DAFTAR ISI

[ABSTRAK i](#_Toc111787780)

[*ABSTRACT* ii](#_Toc111787781)

[KATA PENGANTAR iii](#_Toc111787782)

[DAFTAR ISI v](#_Toc111787783)

[DAFTAR GAMBAR vii](#_Toc111787784)

[DAFTAR TABEL viii](#_Toc111787785)

[BAB I 1](#_Toc111787786)

[PENDAHULUAN 1](#_Toc111787787)

[1.1 Tujuan 1](#_Toc111787788)

[1.2 Teknologi 1](#_Toc111787789)

[1.2.1 Pemilihan Komponen dan Bahan 2](#_Toc111787790)

[1.3 Batasan Pembahasan 4](#_Toc111787791)

[BAB II 5](#_Toc111787792)

[LAPORAN TEKNIK 5](#_Toc111787793)

[2.1 Landasan Teori 5](#_Toc111787794)

[2.1.1 Latar Belakang Proyek 5](#_Toc111787795)

[2.1.2 Lengan Robot (Arm Manipulator) 6](#_Toc111787796)

[2.1.3 Kinematik 7](#_Toc111787797)

[2.1.4 Motor Stepper 9](#_Toc111787798)

[2.1.5 Driver Motor Stepper 14](#_Toc111787799)

[2.1.6 Komunikasi Serial UART Arduino 20](#_Toc111787800)

[2.2 Metodologi Penelitian 21](#_Toc111787801)

[2.3 Gambaran Umum Sistem 22](#_Toc111787802)

[2.4 Perangkat Mekanik 23](#_Toc111787803)

[2.4.1 Konstruksi Arm Manipulator 6 DOF dan Master Controller 23](#_Toc111787804)

[2.4.2 Ukuran dan Area Kerja Arm Manipulator 6 DOF 28](#_Toc111787805)

[2.4.3 Diagram Alir Sistem 29](#_Toc111787806)

[2.4.4 Diagram Alir Sistem Secara Umum 29](#_Toc111787807)

[2.5 Gambaran Sub Sistem 31](#_Toc111787808)

[2.5.1 Konversi Data Sudut ke Data Step 33](#_Toc111787809)

[2.5.2 Pergerakan Motor Stepper 37](#_Toc111787810)

[2.5.3 Rangkaian Elektrik Motor Stepper dan Driver 40](#_Toc111787811)

[2.6 Pengujian 41](#_Toc111787812)

[2.6.1 Pengujian pergerakan stepper 41](#_Toc111787813)

[BAB III 46](#_Toc111787814)

[KESIMPULAN DAN SARAN 46](#_Toc111787815)

[3.1 Kesimpulan 46](#_Toc111787816)

[3.2 Saran 46](#_Toc111787817)

[DAFTAR PUSTAKA 47](#_Toc111787818)

[LAMPIRAN 49](#_Toc111787819)

# DAFTAR GAMBAR

[**Gambar 2. 1** Master Controller 6](#_Toc111787820)

[**Gambar 2. 2** Lengan robot 6 DoF 7](#_Toc111787821)

[**Gambar 2. 3** Motor Stepper 9](#_Toc111787822)

[**Gambar 2. 4** Pergerakan full step one phase 10](#_Toc111787823)

[**Gambar 2. 5** Pergerakan full step two phase 11](#_Toc111787824)

[**Gambar 2. 6** Pergerakan half step 11](#_Toc111787825)

[**Gambar 2. 7** Rangkaian motor stepper unipolar 12](#_Toc111787826)

[**Gambar 2. 8** Lilitan pada Bipolar Stepper Motor 13](#_Toc111787827)

[**Gambar 2. 9** Rangkaian Bipolar Stepper Motor 13](#_Toc111787828)

[**Gambar 2. 10** Driver stepper motor TB6600 15](#_Toc111787829)

[**Gambar 2. 11** Driver motor stepper A4988 17](#_Toc111787830)

[**Gambar 2. 12** Pin A4988 18](#_Toc111787831)

[**Gambar 2. 13** Ilustrasi serial komunikasi Rx Tx 20](#_Toc111787832)

[**Gambar 2. 14** Flowchart Metodologi Penelitian 21](#_Toc111787833)

[**Gambar 2. 15** Arm Manipulator 6 Dof dan Master Controller 22](#_Toc111787834)

[**Gambar 2. 16** Gambaran Umum Sistem Arm Manipulator dan Mater Controller 23](#_Toc111787835)

[**Gambar 2. 17** Stick Diagram Arm Manipulator 6 DOF 24](#_Toc111787836)

[**Gambar 2. 18** Joint Pada Arm Manipulator 25](#_Toc111787837)

[**Gambar 2. 19** Dimensi Master Controller 26](#_Toc111787838)

[**Gambar 2. 20** Design Panel Master Controller dan Joystick Master Controller 27](#_Toc111787839)

[**Gambar 2. 21** Area Kerja Arm Manipulator 28](#_Toc111787840)

[**Gambar 2. 22** Diagram Alir system umum 29](#_Toc111787841)

[**Gambar 2. 23** Diagram Alir Sistem secara umum A 30](#_Toc111787842)

[**Gambar 2. 24** Ilustrasi Pulley yang digunakan 35](#_Toc111787843)

[**Gambar 2. 25** Rangkaian stepper motor dan driver motor 40](#_Toc111787844)

[**Gambar 2. 26** Salah satu pengujian yaitu pada joint 1 menggunakan busur digital 44](#_Toc111787845)

# DAFTAR TABEL

[**Tabel 2. 1** Perbandingan TB6600 dan TB6560 2](#_Toc111787846)

[**Tabel 2. 2** Perbedaan Inverse Kinematic dan Forward Kinematic 9](#_Toc111787847)

[**Tabel 2. 3** Urutan eksitasi motor stepper full step one phase 10](#_Toc111787848)

[**Tabel 2. 4** Urutan eksitasi motor stepper full step two phase 11](#_Toc111787849)

[**Tabel 2. 5** Spesifikasi TB6600 15](#_Toc111787850)

[**Tabel 2. 6** Setting MicroStep 17](#_Toc111787851)

[**Tabel 2. 7** Setting current TB6600 17](#_Toc111787852)

[**Tabel 2. 8** Spesifikasi A4988 18](#_Toc111787853)

[**Tabel 2. 9** Setting Micro Step 19](#_Toc111787854)

[**Tabel 2. 10** Jangkauan sudut pergerakan arm manipulator 33](#_Toc111787855)

[**Tabel 2. 11** Jumlah gigi pada pulley yang digunakan 35](#_Toc111787856)

[**Tabel 2. 12** Pengujian Sudut Pada Joint 1 42](#_Toc111787857)

[**Tabel 2. 13** Pengujian Sudut Pada Joint 2 42](#_Toc111787858)

[**Tabel 2. 14** Pengujian Sudut Pada Joint 3 43](#_Toc111787859)

[**Tabel 2. 15** Pengujian Sudut Pada Joint 4 43](#_Toc111787860)

[**Tabel 2. 16** Pengujian Sudut Pada Joint 5 43](#_Toc111787861)

[**Tabel 2. 17** Pengujian Sudut Pada Joint 6 44](#_Toc111787862)

[**Tabel 2. 18** Hasil Pengujian Rata-Rata Persen Error 45](#_Toc111787863)

# BAB I

# PENDAHULUAN

* 1. Tujuan

Tujuan dari penulisan karya tulis proyek akhir ini adalah sebagai berikut:

* + - 1. Mengetahui bagaimana mengkonversi data *degree* menjadi data *step*
      2. Mengetahui *setting microstep* dan *setting current motor driver*
      3. Mencari pergerakan stepper motor agar lebih halus
      4. Menguji keakurasian motor stepper terhadap masukan pulsa
  1. Teknologi

Teknologi yang digunakan para proyek akhir ini adalah:

1. Interfacing : LCD Display 20x4
2. Proses : Microcontroller - Arduino Mega 2560

Perangkat keras:

1. Motor Stepper : a. Nema 8 (8HS11-0204S)

b. Nema 11 (28HD1411-02)

c. Nema 17 (17HS4401)

d. Nema 23 (5718HB2401)

1. Driver Motor Stepper : a. A4988 Driver Stepper Module

b. TB6600 Driver Stepper Module

1. Input Digital : a. Push Button

b. Saklar Toggle

1. Output Digital : LED
2. Sensor : Potensiometer 100K Ohm

Perangkat Lunak:

1. Arduino IDE
2. Solidworks 2020
3. Eagle PCB Design
   * 1. Pemilihan Komponen dan Bahan

Berdasarkan teknologi yang digunakan, terdapat pertimbangan dan batasan dalam pemilihan teknologi tersebut. Adapun pemilihan komponen pada proyek ini adalah :

* + - 1. Motor Stepper Bipolar sebagai penggerak

Untuk memilih suatu penggerak yang sesuai pada tiap *joint Arm Manipulator* perlu beberapa pertimbangan. Namun alasan utama pemilihan motor stepper yaitu motor stepper memiliki variasi sudut *step*. Dengan adanya variasi sudut step tersebut akan lebih memudahkan untuk melakukan pengontrolan serta pengontrolannya dapat langsung menggunakan sinyal digital tanpa perlu menggunakan rangkaian *closed-loop feedback* untuk memonitor posisinya. Dengan alasan inilah maka motor stepper banyak digunakan sebagai aktuator yang menerapkan rangkaian digital sebagai pengontrol (*driver*), ataupun untuk *interfacing* ke piranti yang berbasis mikrokontroler. Alasan pengguanaan motor stepper bipolar yaitu memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama.

* + - 1. TB6600 dan A4988 sebagai driver motor stepper
         1. TB6600

Sebelum memilih *driver* motor yang pas untuk digunakan, dilakukan perbandingan antara *driver* TB6600 dengan TB6500. Perbandingan tersebut digunakan sebagai acuan pemilihan driver motor, dapat dilihat pada tabel 2.1 dibawah ini.

**Tabel 2. 1** Perbandingan TB6600 dan TB6560

|  |  |  |
| --- | --- | --- |
|  | TB6600 | TB6560 |
| *Operating voltage* | 10 – 35 VDC, 24 VDC *recommended* | 9 – 42 VDC, 36 VDC *recommended* |
| *Output current (peak)* | 5.0 A | 3.5 A |
| *Microstep resolution* | *Full*, ½, ¼, 1/8, 1/16, dan 1/32 | *Full*, ½, 1/8, dan 1/16 |

Berdasarkan tabel 2.1, didapat sebuah kesimpulan yaitu perbedaan yang cukup jelas adalah kapasitas peak current TB

6600 yang lebih besar, sehingga kuat untuk mengangkat motor stepper Nema 23 keatas. Kelebihan lain TB6600 dibandingkan TB6560 yaitu *wiring* yang lebih mudah sehingga tidak memerlukan banyak kabel. Pada *controller* TB6600 hanya perlu menghubungkan CW, CLK, +5V, sedangkan TB6560 *controller* CW-, CLK-, dan EN- perlu dihubungkan ke *ground*.

Keistimewaan *driver* TB6600 yaitu dapat membuat pergerakan stepper motor menjadi lebih halus dengan membagi 360 putaran dengan jumlah pulsa yang lebih banyak (lebih dari 200 pulsa).

* + - * 1. A4988

Pada rencana awal, dilakukan riset mendalam antara driver A4988 dan L298N. Dari hasil riset tersebut didapatkan sebuah kesepakatan untuk memakai A4988. Alasan utama penggunaan A4988 dibandingkan L298N yaitu dapat mengendalikan banyak motor stepper. Kelebihan lainnya yaitu A4988 dapat mengontrol kecepatan dan arah putaran motor steper bipolar seperti Nema 17 hanya dengan menggunakan 2 pin.

* + - 1. Bahan

Seluruh rangka *arm manipulator* dan *master controller* dibuat dengan menggunakan *3D printer Ender 3 V2*. Adapun bahan yang digunakan adalah jenis bahan *Polylactic acid* . *Polylactic acid* (PLA) adalah bahan yang berpotensi menggantikan polimer tersebut. PLA merupakan salah satu biopolimer yang diproduksi dari fermentasi gula pada bahan yang dapat diperbaharui seperti tebu dan jagung. Oleh karena itu PLA sangat aman digunakan dan ramah lingkungan. Keunggulan PLA jika dibandingkan dengan biopolimer lainnya adalah memiliki kemampuan proses *thermal* yang lebih baik jika dibandingkan dengan bahan lain seperti *polyhydroxyl alkanoate (PHA), polyethylene glycol (PEG),* dan *polycaprolactone (PCL),* memiliki kekuatan tarik tinggi, dan dapat diproses menggunakan peralatan.

* 1. Batasan Pembahasan

Fokus pembahasan yang diuraikan pada karya tulis ini dibatasi. Pembatasan bahasan pada karya tulis ini adalah sebagai berikut :

1. Kinerja robot bergantung pada *hardware* yang digunakan.
2. Lengan robot menggunakan desain robot yang bersifat *open source* pada laman *SkyentificGit/SmallRobotArm - GitHub*
3. *Master controller* dan *arm manipulator* (lengan robot) terhubung dengan komunikasi serial.
4. Tidak terdapat fitur *Teaching* untuk lengan robot.
5. *Design* lengan robot yang digunakan bersifat *open-source*.
6. *Arm manipulator* digerakkan menggunakan mode *Joystick Master Controller.*
7. Robot digerakkan secara *Forward Kinematics* (masukan derajat menjadi keluaran posisi).

# BAB II

# LAPORAN TEKNIK

* 1. Landasan Teori
     1. Latar Belakang Proyek

Teknologi robot mengalami suatu kemajuan yang sangat pesat dewasa kini. Robot telah menggantikan peralatan-peralatan manual yang membutuhkan banyak tenaga manusia. Dalam dunia industri, robot merupakan salah satu alat bantu yang dalam kondisi tertentu sangat diperlukan. Terdapat beberapa kondisi tertentu dalam bidang industri yang tidak mungkin ditangani oleh manusia. Robot memiliki banyak kelebihan yang tidak dimiliki manusia, yaitu menghasilkan kualitas yang sama ketika mengerjakan suatu pekerjaan secara berulang–ulang, tidak mudah lelah, dan dapat diprogram ulang sehingga dapat difungsikan untuk beberapa tugas yang berbeda. Dan diantara robot yang sering digunakan dalam bidang industri adalah *Arm Manipulator*. Penerapan *Arm Manipulator* banyak ditemukan di industri, seperti mengelas, mengecat pada industri otomotif dan *packaging* dan *filling* pada industri kimia.

Pada tugas akhir ini, dibuatlah sebuah *Arm Manipulator* yang memungkinkan untuk membantu pekerjaan manusia. Arm Manipulator yang dibuat memiliki 6 degree of freedom (DOF), pada masing– masing DOF digerakan oleh sebuah motor stepper yang dikendalikan oleh mikrokontroler Arduino Mega. .*Arm manipulator* dikendalikan oleh semacam pendantyang disebut dengan *master controller*. Pengendaliannya dilakukan secara remote/jarak jauh menggunakan komunikasi serial. *Master controller* memiliki kurang lebih 7 sensor yang terpasang padanya yang masing-masing mengendalikan aktuator yang berada pada *joint arm manipulator.* Pada *master controller* pun terdapat banyak tombol yang diperuntukkan untuk berbagai fitur seperti *scalling* pergerakan*, start, stop,* pergerakan translasi, dan lain-lain. Berikut merupakan contoh dari master controller seperti pada gambar 2.1.



**Gambar 2. 1** Master Controller

Dalam rangka sedikit sumbangsih untuk kemajuan teknologi pada industry di Indonesia dan dengan tujuan pembelajaran bagi penulis serta generasi selanjutnya. Dirancang lah proyek akhir ini yang merupakan lengan robot 6 DOF yang dikendalikan oleh master controller. Dikontrol menggunakan 2 Arduino Mega yang dikomunikasikan secara serial dengan tampilan *interface* berupa LCD *display* untuk menampilkan posisi dari lengan robot.

* + 1. Lengan Robot (Arm Manipulator)

*Arm Manipulator* adalah kontruksi robot yang memiliki bentuk hanya sebuah lengan saja. Robot ini memiliki aktuator berupa motor untuk menggerakan seluruh bagian robot tersebut. Robot manipulator ini sudah banyak dipakai di bidang industry terutama dalam bagian packing suatu pabrik. Selain untuk menghemat waktu robot ini digunakan karena hasil yang diperoleh juga lebih baik daripada hasil yang dikerjakan oleh manusia.

Lengan robot banyak digunakan di industri yang membutuhkan ketepatan tinggi dan pekerjaan yang repetitif. Namun dalam pengaplikasian di ROV (*Remotely Operated Vehicle*), lengan robot lebih ditekankan pada mobilitas nya dibandingkan ketepatannya dikarenakan lingkungan pengaplikasiannya yang dinamis. Untuk ketepatan, nantinya akan dipantau oleh operator yang mengoperasikan lengan robot. Seperti pada gambar 2.2 merupakan contoh dari *Arm Manipulator 6 DoF*.



**Gambar 2. 2** Lengan robot 6 DoF

Beberapa istilah dalam arm manipulator yaitu :

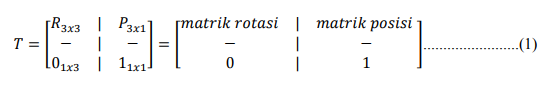
1. *Joint* (Sendi) yaitu koneksi antar *link* yang dapat menentukan pergerakan.
2. *Link* merupakan bagian-bagian kerangka yang kaku yang dihubungkan secara bersamaan sehinga membentuk suatu rangkaian *kinematic.*
3. Derajat kebebasan (*Degree Of Freedom* atau DOF) merupakan setiap gerakan linier atau putaran sepanjang atau sekitar pada sebuah sumbu (axis).
   * 1. Kinematik

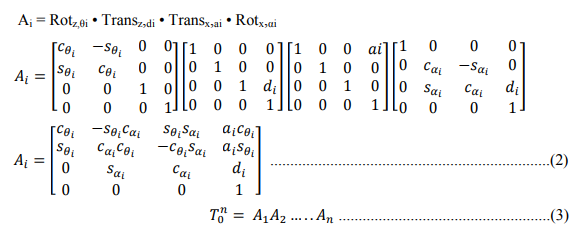
Dalam fisika, kinematika didefiniskan sebagai ilmu yang mempelajari gerak tanpa mempersoalkan penyebabnya. Dalam kasus lengan robot, maka kinematika adalah studi analitis pergerekan lengan robot yang bergerak tanpa memperhatikan gaya yang menyebabkan pergerakan tersebut. Di robotika, kita mengenal dua macam pendekatan kinematika yakni *forward kinematic* dan *invers kinematic*.

1. Forward Kinematik

Koordinat posisi dan orientasi dari *end effector* merupakan hal yang penting dalam pergerakan manipulator. Orientasi menyatakan rotasi *end effector* terhadap *base frame* sedangkan koordinat posisi menyatakan letak *end effector* (x,y,z) terhadap *base frame*. Ciri khas dari penyelesaian menggunakan persamaan *forward kinematics* adalah hanya mempunyai satu solusi penyelesaian yang menyatakan

hubungan antara *joint* terkait atau *end effector* terhadap *base frame* sehingga dapat diketahui koordinat posisi dan orientasi dari *end effector*.

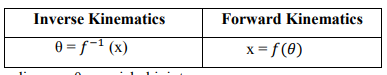


*Forward kinematic* dinyatakan dalam matriks transformasi homogen seperti pada persamaan 1 dimana merupakan matriks rotasi dan merupakan matriks posisi yang menyatakn pergerakan *end effector* terhadap *base frame*. Langkah untuk mencari formard kinematic pertama kali adalah mencari terlebih dahulu parameter tiap joint dengan aturan *Denavit Hartenberg* (DH). Parameter *joint* ini kemudian dibentuk menjadi matriks transformasi homogen tiap joint seperti pada persamaan (2), untuk selanjutnya dibentuk menjadi matriks transformasi seperti pada persamaan (3) yang menyatakan hubungan antar *joint*.

1. Inverse Kinematik

*Inverse kinematics* digunakan untuk mencari kombinasi sudut yang harus diberikan pada *arm manipulator* agar *end effector* bergerak menuju koordinat posisi yang ditentukan. Hal ini berbeda dengan konsep *forward kinematics*, dimana sudut dari tiap *joint* sudah ditentukan untuk mencari koordinat posisi dari *end effector*, perbedaan keduanya dapat dilihat pada Tabel 2.2. Ada beberapa cara untuk menyelesaikan *inverse kinematics*, salah satunya dalah dengan metode aljabar. Ciri khas dari penyelesaian menggunakan persamaan *inverse kinematics* adalah mempunyai lebih dari satu solusi penyelesaian, karena dimungkinkan memiliki lebih dari satu kombinasi sudut tiap *joiunt* untuk mendapatkan koordinat posisi dari *end effector*.

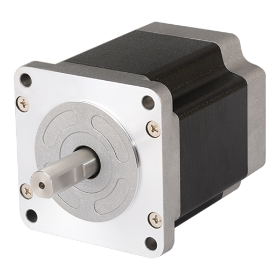
**Tabel 2. 2** Perbedaan Inverse Kinematic dan Forward Kinematic



Dimana : θ = variabel joint

x = posisi dan orientasi end effector

* + 1. Motor Stepper



**Gambar 2. 3** Motor Stepper

*Motor stepper* adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor *stepper* merupakan motor listrik yang dikendalikan dengan pulsa-pulsa digital, bukan dengan memberikan tegangan yang terus-menerus. Motor *stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkannya diperlukan pengendali motor *stepper* yang membangkitkan pulsa-pulsa periodik.

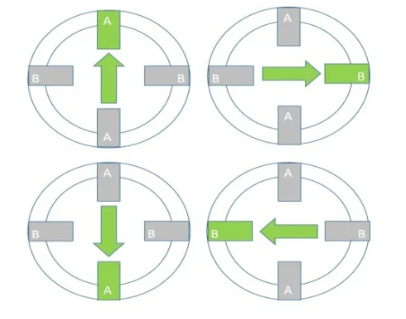
Setiap motor *stepper* akan memiliki spesifikasi utama yaitu berapa derajat motor akan berputar setiap kali diberikan 1 pulse, atau disebut dengan *Step/pulse*. Motor *stepper* mempunyai besaran derajat per *step*, mulai dari 0.72° per *step*, 1.8° per *step*, 3.6° per *step*, 7.5° per *step*, 15° per *step* dan bahkan ada yang 90° per step. Untuk menggerakkan setiap *step* pada *stepper* ada beberapa metode yang dapat dilakukan, diantaranya yaitu *full step* dan *half step*. Metode *full step* berarti motor *stepper* berputar sesuai dengan spesifikasi derajat per *step*nya, sedangkan metode *half step* berarti motor *stepper* berputar setengah derajat per *step* dari spesifikasi motor stepper tersebut.

1. **Full Step**

Pada metode ini, motor stepper akan bergerak sesuai dengan sudut step dasarnya, dalam metode ini juga terdapat dua tipe mode penguatan yaitu :

1. *Single Phase*

Motor dioperasikan hanya dengan satu *phase* yang aktif pada satu waktu. Mode ini membutuhkan daya paling sedikit dari penggerak dibandingkan mode penguatan yang lain.



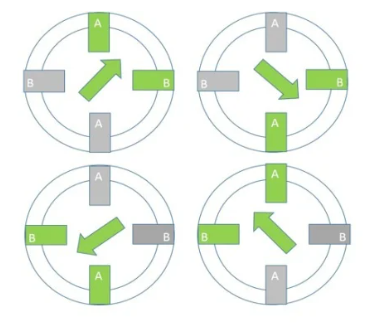
**Gambar 2. 4** Pergerakan full step one phase

**Tabel 2. 3** Urutan eksitasi motor stepper full step one phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Phase** | | | |
| **A** | **B** | **A̅** | **B̅** |
| **1** | 1 | 0 | 0 | 0 |
| **2** | 0 | 1 | 0 | 0 |
| **3** | 0 | 0 | 1 | 0 |
| **4** | 0 | 0 | 0 | 1 |

1. *Two Phase*

Motor dioperasikan dengan kedua *phase* aktif pada saat bersamaan. Mode ini memberikan torsi dan kecepatan yang lebih baik. Mode ini menghasilkan torsi 30% sampai 40% lebih banyak daripada mode *single phase*, namun membutuhkan daya dua kali lebih banyak dari penggerak.



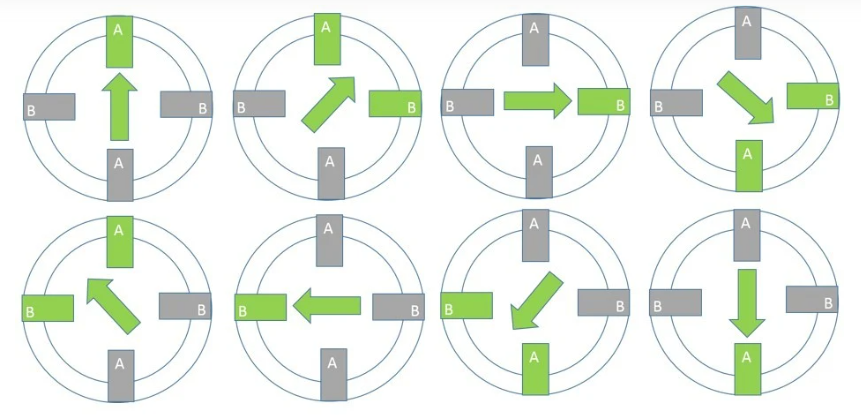
**Gambar 2. 5** Pergerakan full step two phase

**Tabel 2. 4** Urutan eksitasi motor stepper full step two phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Phase** | | | |
| **A** | **B** | **A̅** | **B̅** |
| **1** | 1 | 1 | 0 | 0 |
| **2** | 0 | 1 | 1 | 0 |
| **3** | 0 | 0 | 1 | 1 |
| **4** | 1 | 0 | 0 | 1 |

1. **Half Step**

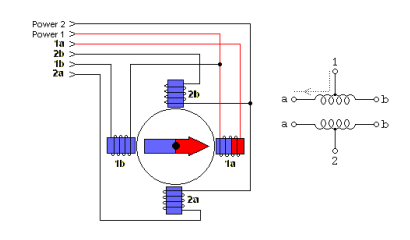
Pada metode ini, menggabungkan mode penguatan *one-phase on full step* dan *two-phase on full step* sehingga sudut pergerakan yang dihasilkan merupakan setengah dari sudut dasar. Karena sudut langkah yang lebih kecil, metode ini memberikan dua kali resolusi dan operasi yang lebih halus. *Half stepping* menghasilkan torsi kira-kira 15% lebih sedikit daripada *two-phase on full step.*



**Gambar 2. 6** Pergerakan half step

Berdasarkan metode perancangan rangkaian pengendalinya, motor stepper dapat dibagi menjadi 2 jenis yaitu motor stepper unipolar dan motor stepper bipolar.

**Motor Stepper Unipolar**

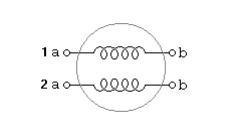
****

**Gambar 2. 7** Rangkaian motor stepper unipolar

Motor stepper dengan tipe unipolar merupakan motor stepper yang memiliki 2 buah lilitan yang masing-masing lilitan di tengah-tengahnya diberikan sebuah tap. Rangkaian pengendali motor stepper unipolar hanya memerlukan satu signal On Off dengan menggunakan *switch* / transistor pada masing – maisng lilitannya. Motor ini mempunyai *step* sebesar 90 derajat dan mempunyai dua buah lilitan yang didistribusikan berseberangan 180 derajat di antara kutub pada stator. Sedangkan pada rotornya menggunakan satu pasang magnet permanen. Gambar 2.9 merupakan konsep dasar sebuah motor stepper yang memiliki sudut tiap-tiap *step* (langkah) atau juga sering disebut dengan *step size* adalah sebesar 90 derajat. Prinsip kerja secara mendasar untuk arah putaran searah jarum jam (CW) berdasarkan diagram pada gambar 2.9 adalah sebagai berikut: ketika lilitan stator 1a dialiri arus listrik, maka pada gigi stator 1a akan ditimbulkan medan magnet. Dengan demikian kutub magnet pada rotor akan tertarik ke arah stator 1a. Urutan pemberian arus listrik pada lilitan stator selanjutnya adalah pada gigi stator 2a, 1b an 2b. Proses tersebut akan menghasilkan satu putaran. Agar motor stepper dapat berputar secara terus menerus maka proses pemberian arus listrik pada lilitan stator juga harus dilakukan secara terus menerus.

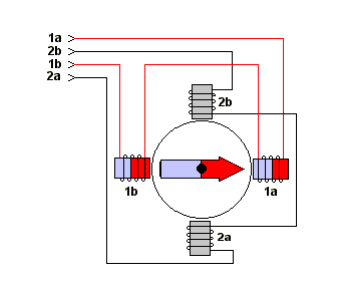
**Motor Stepper Bipolar**

Bipolar Stepper Motor hampir sama atau mirip dengan Unipolar Stepper Motor, namun pada motor stepper jenis bipolar tidak menggunakan sambungan *center tap.* Oleh karena itu Motor stepper jenis ini memerlukan jenis rangkaian kontrol yang berbeda. Prinsip dasar dari motor stepper bipolar ditunjukkan oleh gambar 2.11.



**Gambar 2. 8** Lilitan pada Bipolar Stepper Motor

Pada gambar 2.10 ditunjukkan sebuah Bipolar Stepper Motor yang terdiri dari 2 lilitan. Kedua lilitan memiliki spesifikasi yang sama (identik) dan tidak terhubung ke sumber daya. Model motor stepper bipolar secara konseptual dapat dilihat pada gambar 2.11.



**Gambar 2. 9** Rangkaian Bipolar Stepper Motor

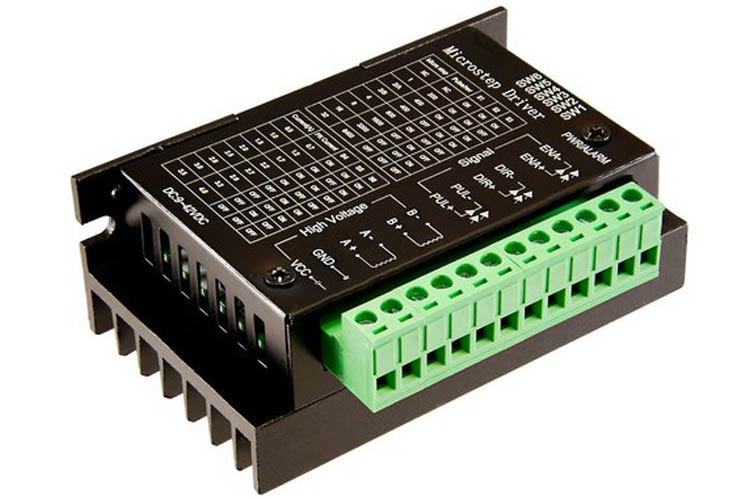
Prinsip kerja dari motor srepper jenis bipolar dapat dijelaskan sebagai berikut: semua lilitan diaktifkan sesuai dengan urutannya untuk menggerakkan rotor yang digambarkan sebagai anak panah pada gambar 2.11. Pada diagram konseptual ditunjukkan bahwa motor stepper memiliki 90 derajat *step* per phasa. Kita asumsikan bahwa terminal 1a dihubungkan ke sumber daya polaritas positif dan 1b ke negatif, maka rotor akan bergerak ke arah lilitan 1a. Bila polaritas lilitan 1 dibalik (1a terhubung ke negatip dan 1b ke positip) maka rotor akan bergerak ke arah sebaliknya (lilitan 1b). Dalam konsep urutan *”wave drive”* searah putaran jarum jam (CW), lilitan 1 di-non aktifkan dan lilitan 2 dieksitasi untuk mengaktifkan phasa berikutnya. Rotor akan dibawa atau digerakkan kearah lilitan yang dieksitasi sehingga akan menghasilkan putaran yang kontinyu. Bila dua lilitan yang berurutan dieksitasi secara bersama-sama maka rotor akan bergerak ke arah atau posisi diantara kedua lilitan tersebut.

* + 1. Driver Motor Stepper

Secara teoritis, sebuah motor stepper dapat digerakkan langsung oleh mikrokontroller (Trianto, 2005). Dalam kenyataanya, arus dan tegangan yang dikeluarkan oleh mikrokontroller terlalu kecil untuk menggerakkan sebuah motor stepper. Gerbang-gerbang Transistor Transistor Logic (TTL) mikrokontroller hanya mampu mengeluarkan arus dalam orde mili-ampere dan tegangan antara 2 sampai 2.5 Volt. Sementara itu untuk menggerakkan motor stepper diperlukan arus yang lebih besar (dalam orde ampere) dan tegangan berkisar 5 sampai 24 Volt. Untuk mengatasi masalah tersebut, diperlukan sebuah piranti tambahan yang memenuhi kebutuhan arus dan tegangan yang cukup besar. Rangkaian *driver* motor stepper merupakan rangkaian “*open collector*”, dimana output rangkaian ini terhubung dengan ground untuk mencatu lilitan-lilitan motor stepper. Rangkaian pengendali motor stepper ini berfungsi sebagai *interface* antara komputer dengan motor stepper.

Berdasarkan jurnal *Texas Instrument,* driver motor yang digunakan pada motor stepper dengan pencocokan arus yang lebih baik dan tingkat microstepping yang tinggi dapat membantu meningkatkan akurasi secara signifikan pada motor stepper. Laporan tersebut menjelaskan bagaiman fitur driver motor tertentu dapat meningkatkan kualitas gerak untuk berbagai macam system dan aplikasi motor stepper. *Driver* motor stepper yang digunakan pada arm manipulator ini yaitu TB6600 dan A4988.

* + - 1. **Driver Motor Stepper TB6600**



**Gambar 2. 10** Driver stepper motor TB6600

Penggerak motor stepper bipolar dua fasa berfungsi untuk membangkitkan pulsa-pulsa periodik dan pengatur arah arus dari positif ke negatif pada masing-masing kumparan dengan kecepatan yang tinggi. Penggerak motor stepper yang dipilih adalah Toshiba TB6600. TB6600 adalah driver motor stepper profesional yang mudah digunakan yang dapat Anda sesuaikan langkah mikronya. Modul ini dapat mengontrol motor stepper dua fase. Fitur utama dari modul ini adalah dapat mengubah pengaturan *microstep* dengan sakelar bawaan pada *driver.*

Modul-modul ini memiliki beberapa fungsi keselamatan sebagai berikut:

* Perlindungan *over current*
* *Under voltage current*
* Perlindungan *over heating*

*Driver* motor ini mempunyai daya tinggi dengan arus maksimum 4,5 ampere dan arus starting hingga 5A. Tegangan untuk driver ini mampu menampung hingga 45 Volt. Spesifikasi tersebut menyatakan driver ini mampu menggerakan motor stepper dengan menghasilkan kecepatan serta *torque* yang diinginkan. Pada project ini, TB6600 digunakan untuk menggerakkan motor stepper NEMA 23, NEMA 17, dan NEMA 11.

Berikut adalah tabel spesifikasi dari driver TB6600 :

**Tabel 2. 5** Spesifikasi TB6600

|  |  |
| --- | --- |
| Tegangan masukan | 9 – 40 V |
| Arus masukan | 0 – 5 A |
| Arus keluaran | 0 – 4 A |
| Daya maksimal | 160 W |
| Tegangan sinyal | 5 V |
| Berat | 200 g |

Pada TB6600 terdapat beberapa pin yang mempunyai fungsi berbeda-beda, berikut adalah pin dari TB6600:

Tegangan tinggi

VCC : Catu daya motor 9-42V

GND : Tanah

A+ : Pin positif kumparan 1

A- : Pin negatif kumparan 1

B+ : Pin positif kumparan 2

B- : Pin negatif kumparan 2

Sinyal

PUL : Pin untuk memberikan sinyal/pulse

DIR : Pin untuk mengontrol arah putaran

ENA : Aktifkan pin driver

5V : Tegangan 5V

**Micro Step Setting**

Motor stepper biasanya memiliki ukuran langkah 1,8° atau 200 langkah per putaran, ini mengacu pada langkah penuh. Sehingga ketika diberi 1 pulsa maka motor stepper akan bergerak 1.8 derajat (gerak per pulsa = 360/200 = 1.8 derajat). Kita dapat mengatur microstep motor melalui tiga *dip switches* yaitu S1, S2, dan S3. Sakelar ini digunakan untuk mengontrol resolusi *microstep* dan membatasi arus driver. Keistimewaan *driver* ini yaitu dapat membuat pergerakan stepper motor menjadi lebih halus dengan membagi 360 putaran dengan jumlah pulsa yang lebih banyak (lebih dari 200 pulsa). Sebagai contoh pada motor driver TB6600 ini maksimal mampu memberikan 6400 pulsa untuk 1 kali putaran. dengan kata lain berarti gerak per pulsanya adalah 360/6400 = 0.05625 derajat. Tabel 2.6 menunjukkan langkah *Micro driver*.

**Tabel 2. 6** Setting MicroStep

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Micro Step | Pulse/Rev | S1 | S2 | S3 |
| NC | NC | ON | ON | ON |
| 1 | 200 | ON | ON | OFF |
| 2/A | 400 | ON | OFF | ON |
| 2/B | 400 | OFF | ON | ON |
| 4 | 800 | ON | OFF | OFF |
| 8 | 1600 | OFF | ON | OFF |
| 16 | 3200 | OFF | OFF | ON |
| 32 | 6400 | ON | OFF | OFF |

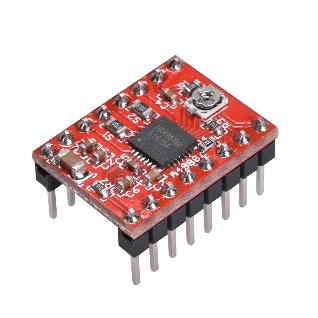
**Current Control Setting**

TB6600 dapat mengatur arus yang masuk ke motor saat sedang berjalan dengan mengatur dip switch S4, S5, dan S6 ke on atau off. Berikut adalah tabel untuk mengatur arus yang diberikan driver untuk masuk ke dalam motor.

**Tabel 2. 7** Setting current TB6600

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Current (A) | Peak Current | S4 | S5 | S6 |
| 0.5 | 0.7 | ON | ON | ON |
| 1.0 | 1.2 | ON | OFF | ON |
| 1.5 | 1.7 | ON | ON | OFF |
| 2.0 | 2.2 | ON | OFF | OFF |
| 2.5 | 2.7 | OFF | ON | ON |
| 2.8 | 2.9 | OFF | OFF | ON |
| 3.0 | 3.2 | OFF | ON | OFF |
| 3.5 | 4.0 | OFF | OFF | OFF |

* + - 1. **Driver Motor Stepper A4988**



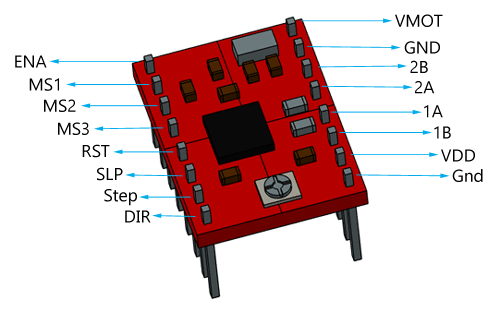
**Gambar 2. 11** Driver motor stepper A4988

A4988 adalah driver motor *microstepping* dengan penerjemah bawaan dalam pengoperasian motor stepper*. Driver* A4988 dirancang untuk mengoperasikan motor stepper bipolar dalam mode *full-step, half-step, quarter-step, eighth-step* dan *sixteenth-step*. Dengan kapasitas output hingga 35V dan ± 2A. Terdapat regulator arus yang tetap ketika motor stepper dalam keadaan berhenti. Terdapat rangkaian pengaman pada driver A4988 yaitu pengontrol panas atau suhu dengan kontrol histerisis, penguncian dibawah tegangan (UVLO), proteksi arus silang dan tidak diperlukan urutan power on khusus.

Driver ini memiliki kapasitor keramik ESR rendah, yang membuatnya rentan terhadap lonjakan tegangan. Dalam beberapa kasus, lonjakan ini dapat melebihi 35V (peringkat tegangan maksimum A4988), berpotensi merusak papan dan bahkan motor secara permanen. Salah satu cara untuk melindungi *driver* dari lonjakan tersebut adalah dengan menempatkan kapasitor elektrolitik 100μF (setidaknya 47μF) di pin catu daya motor. Berikut adalah tabel spesifikasi dari driver motor A4988.

**Tabel 2. 8** Spesifikasi A4988

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Karakteristik | Simbol | Keterangan | Nilai | Unit |
| Supply Tegangan Beban | VBB |  | 35 | V |
| Arus Keluaran | IOUT |  | ±2 | A |
| Masukkan Tegangan Logika | VIN |  | -0.3 to 5.5 | V |
| Supply Tegangan Logika | VDD |  | -0.3 to 5.5 | V |
| VBBx to OUTx |  |  | 35 | V |
| Tegangan Sense | VSENSE |  | 0.5 | V |
| Tegangan Referensi | VREF |  | 5.5 | V |
| Suhu Operasi | TA | Jarak S | -20 to 85 | °C |
| Suhu Persimpangan Maksimum | TJ (max) |  | 150 | °C |
| Suhu Penyimpanan | Tstg |  | -55 to 150 | °C |



**Gambar 2. 12** Pin A4988

Pada gambar 2.14 diatas merupakan pin-pin yang terdapat pada Driver Motor A4988. Driver A4988 memungkinkan *microstepping* dengan membagi satu step menjadi langkah yang lebih kecil. Hal tersebut dilakukan dengan memberi sinyal pada kumparan melalui pin MS1, MS2, dan MS3. Berikut adalah tabel 2.9 merupakan *setting microstepping* A4988.

**Tabel 2. 9** Setting Micro Step

|  |  |  |  |
| --- | --- | --- | --- |
| **MS1** | **MS2** | **MS3** | **Microstep Resolution** |
| Low | Low | Low | Full Step |
| High | Low | Low | Half Step |
| Low | High | Low | Quarter Step |
| High | High | Low | Eighth Step |
| High | High | High | Sixteenth Step |

Harus dilakukan sedikit penyesuaian sebelum menjalankan motor saat menggunakan *driver* ini. Perlunya dibatasi jumlah arus maksimum yang mengalir melalui kumparan stepper dan mencegahnya melebihi *rated current* motor.

Untuk mengatur batas saat ini, potensiometer pemangkas kecil disediakan pada driver A4988.

Ada dua metode untuk melakukan penyesuaian ini:

Metode 1

Dalam metode ini batas arus ditentukan dengan mengukur tegangan (Vref) pada pin “ref”.

1. Lihatlah datasheet untuk motor stepper yang digunakan. Catat *rated current*.
2. Letakkan *driver* dalam mode *full-step* dengan melepaskan tiga pin pilihan *microstep*.
3. Tahan motor pada posisi tetap tanpa mencatat input *step.*
4. Ukur tegangan (Vref) pada metal trimmer pot saat menyesuaikannya.
5. Sesuaikan tegangan Vref menggunakan rumus,

Metode 2

Dalam metode ini batas arus ditentukan dengan mengukur arus yang mengalir melalui kumparan.

Lihatlah datasheet untuk motor stepper yang digunakan. Catat *rated current.*

Letakkan driver dalam mode *full-step* dengan melepaskan tiga pin pilihan *microstep*.

Tahan motor pada posisi tetap tanpa mencatat input step. Jangan biarkan input *step* mengambang, sambungkan ke catu daya logika (5V).

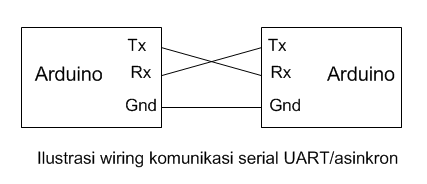
Tempatkan ammeter secara seri dengan salah satu kumparan pada motor stepper dan ukur arus aktual yang mengalir.

Ambil obeng kecil dan sesuaikan potensiometer batas arus hingga mencapai *rated current*.

* + 1. Komunikasi Serial UART Arduino

Secara sederhana, komunikasi serial adalah sebuah komunikasi yang terjadi dengan mengirimkan data per-bit secara berurutan dan bergantian. Dengan adanya komunikasi serial, maka Arduino bisa melakukan pembagian tugas dengan Arduino lainnya untuk mengendalikan sistem agar pengolahan data menjadi ringan.

Umumnya semua *board* Arduino telah memiliki minimal 1 *port* serial yang biasa dikenal dengan jenis USART atau UART (*Universal Asynchronous Receiver-Transmitter)*. Proses komunikasi serial menggunakan dua pin, yaitu pin RX untuk menerima data dan pin TX untuk mengirimkan data. Itulah alasan mengapa terkadang ada yang menyebut istilah komunikasi RX TX Arduino. Biasanya pin RX di Arduino adalah pin0 dan pin TX adalah pin 1.

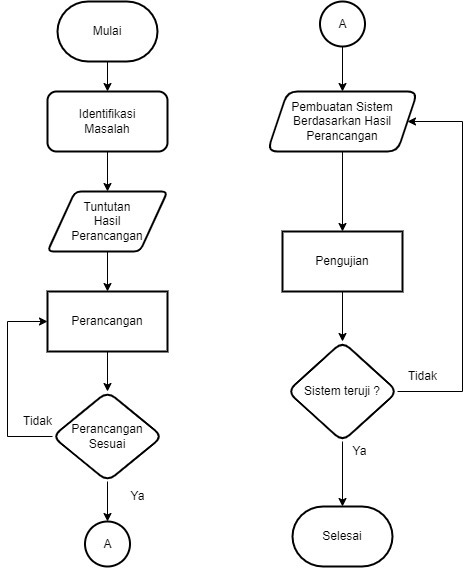


**Gambar 2. 13** Ilustrasi serial komunikasi Rx Tx

Pengiriman data pada komunikasi serial ditentukan oleh *baudrate*. *Baudrate* adalah kecepatan transfer data dalam bit per satuan detik atau *bit per second (bps)*. Beberapa pilihan *baudrate* pada Arduino yaitu 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800 38400, 57600 dan 115200.

* 1. Metodologi Penelitian

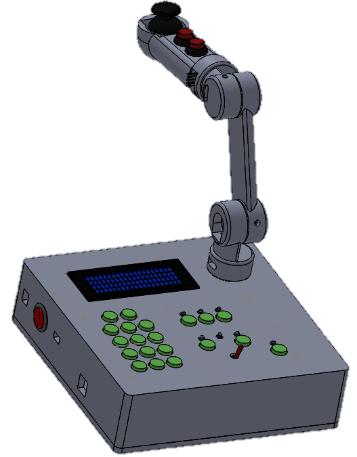
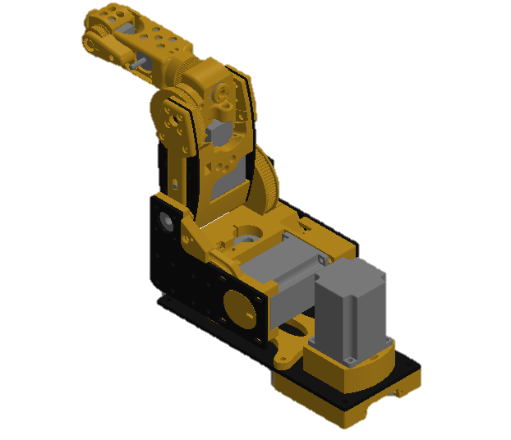
Pertama penulis membuat identifikasi masalah, kemudian membuat tuntutan hasil perancangan yang akan dicapai, setelah itu membuat konsep sistem perancangan. Setelah terkonsep secara keseluruhan, jika sistem yang dibuat belum sesuai, maka dilakukan pengkonsepan ulang, jika sistem sudah sesuai, maka dilakukan instalasi dan pengujian sistem. Jika sistem yang sudah di uji belum sesuai, maka dilakukan pengujian ulang, jika sistem yang sudah diuji sesuai, maka sistem sudah baik untuk di implementasikan.



**Gambar 2. 14** Flowchart Metodologi Penelitian

* 1. Gambaran Umum Sistem

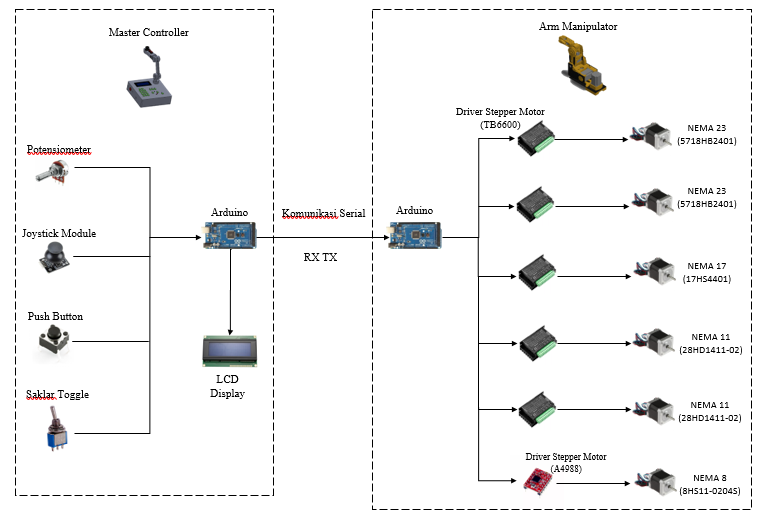
Secara hardware, sistem proyek akhir ini terdiri dari *arm manipulator* *6* *DOF* dan *master controller*. *Arm manipulator 6 DOF* merupakan robot berbentuk lengan yang memiliki 6 sendi pergerakan (*joint*). Sedangkan *master controller* adalah sebuah pendant yang mengendalikan pergerakan dari *arm manipulator*. Pergerakan *joint* pada *manipulator* dihasilkan dari pembacaan sensor *joystick* yang terpasang pada *master controller*. Satu *joint* pada sensor *joystick* akan mengendalikan satu *joint* pada *arm manipulator*. *Master controller* akan menghasilkan data derajat dari pembacaan sensor. Data sensor tersebut akan dikirim ke *Arduino slave* untuk dikonversi menjadi data *pulse* untuk menggerakkan stepper motor.

****

**Gambar 2. 15** Arm Manipulator 6 Dof dan Master Controller

Secara teknis lengan robot dikendalikan oleh Master *controller* dan data hasil pembacaan yang sudah berupa besaran sudut akan ditampilkan pada display LCD sekaligus mengirimkan data secara serial kepada *Slave Controller Arduino mega* untuk pemprosessan data selanjutnya.

*User* menggerakkan lengan robot secara manual dengan menggerakkan *joystick* yang terdapat pada *master controller.* Kemudian pergerakkan yang terbaca oleh potensiometer berupa data diolah oleh program dan ditampilkan kedalam bentuk sudut.



**Gambar 2. 16** Gambaran Umum Sistem Arm Manipulator dan Mater Controller

Terlihat pada Gambar 2.18 diatas *master controller* dibangun atas berbagai inputan komponen dan sensor, adapun inputan komponennya terdiri dari Potensiometer, modul *Joystick, Push button*, dan saklar Toggle. Data yang dihasilkan kemudian akan di proses oleh Arduino (master) sebagai controller. Lalu data akan di tampilkan melalui LCD sekaligus dikirim melalui komunikasi serial UART.

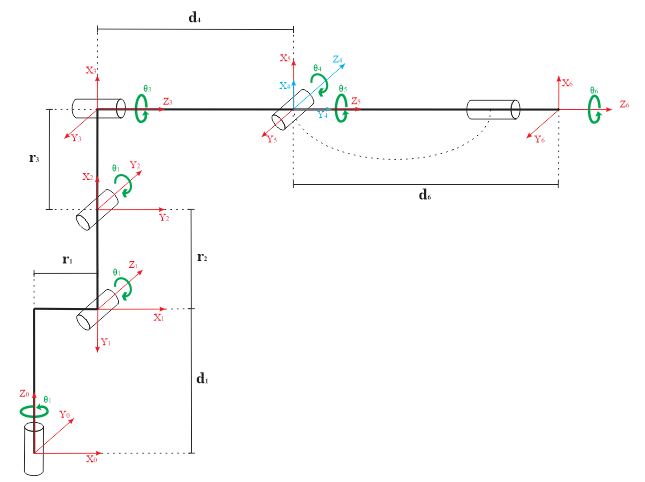
Data yang dikirim berupa sudut akan diterima oleh Arduino (slave) dan akan diubah menjadi pulsa digital untuk kemudian driver motor akan menggerakkan tiap-tiap aktuator motor stepper yang berada pada setiap *joint arm manipulator*/lengan robot.

* 1. Perangkat Mekanik
     1. Konstruksi Arm Manipulator 6 DOF dan Master Controller

*Arm manipulator* dan *master controller* ini terbangun dari sistem kontrol, perangkat elektronik, dan perangkat mekanik yang masing-masing entitas tersebut saling berkesinambungan menjalankan fungsi setiap bagiannya. Sebagian besar konstruksi rangka dari *arm manipulator* dan *master controller* di manufaktur oleh *3D printer* dengan bahan PLA. Berikut perangkat – perangkat mekanik pada alat ini:

1. *Arm Manipulator*

*Manipulator* terdiri dari *links* dan *joints*. Rangka dari *manipulator* 6 DOF ini dapat tergambarkan pada stick diagram seperti ini :



**Gambar 2. 17** Stick Diagram Arm Manipulator 6 DOF

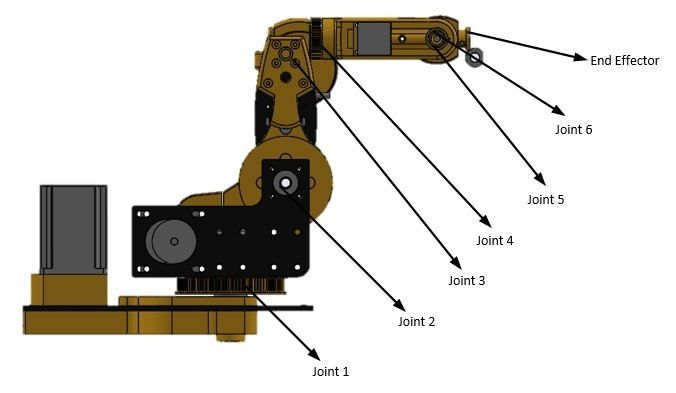
Panjang link :

r1 = 47 mm d1 = 133 mm

r2 = 110 mm d4 = 117.5 mm

r3 = 26 mm d6 = 28 mm

Motor stepper pun dipasang pada masing – masing *joint* sebagai sumber pergerakan. Berikut merupakan spesifikasi dari masing – masing motor stepper di setiap *joint* :



**Gambar 2. 18** Joint Pada Arm Manipulator

*Joint* 1 dan 2

Jenis : Nema 23 5718HB2401 bipolar stepper motor

Dimensi : 54x54x55 mm

*Shaft diameter* : 6.35 mm

*Step angle* : 1.8o

*Holding torque* : 1.3 N.m

*Phase current* : 3.5 A

Joint 3

Jenis : Nema 17 17HS4401 Bipolar stepper motor

Dimensi : 42x42x40 mm

*Shaft diameter / length* : 5 mm x 22 mm

*Step angle* : 1.8o

*Holding torque* : 4 Kg.cm

*Phase current* : 1.7A

Joint 4 dan 5

Jenis : Nema 11 28HD1411-02

Dimensi : 28x28x28 mm

*Shaft diameter / length* : 5 mm x 18 mm

*Step angle* : 1.8o

*Holding torque* : 6.5 N.cm

*Phase current* : 0.8 A

Joint 6

Jenis : Nema 8 8HS11-0204S

Dimensi : 20x20x28 mm

*Shaft diameter / length* : 4 mm x 10 mm

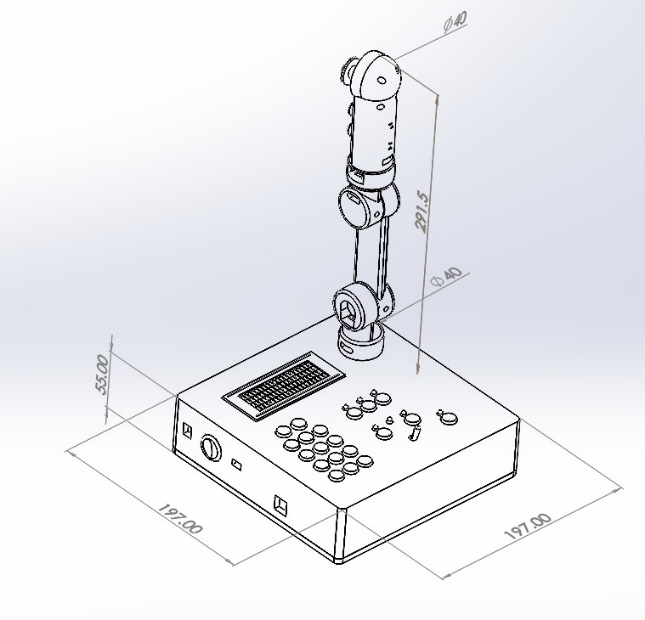
*Step angle* : 1.8o

*Holding torque* : 1.6 N.cm

*Phase current* : 0.2 A

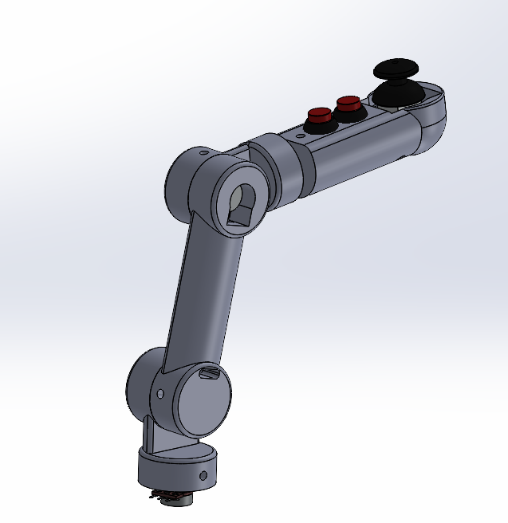
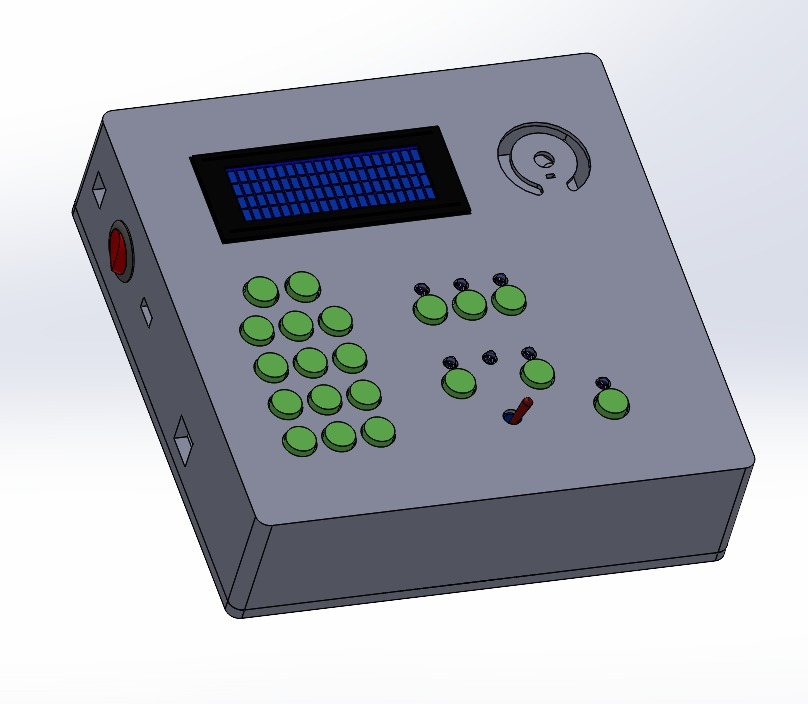
1. *Master Controller*

*Master Controller* memiliki dimensi yang di rancang agar *Master Arm* dapat dioperasikan dengan memaksimalkan sudut perputaran pada potensiometer. Adapun dimensi pada *Master Controller* adalah seperti pada gambar dibawah ini:



**Gambar 2. 19** Dimensi Master Controller

*Master controller* terdiri dari dua bagian, yaitu bagian Panel Master yang berfungsi sebagai *casing panel, display* pergerakan *arm manipulator* dan tombol operasi. Kemudian bagian selanjutnya yaitu *Joystick Master* yang berfungsi sebagai pengendali lengan robot pada mode *Joystick*.



**Gambar 2. 20** Design Panel Master Controller dan Joystick Master Controller

Panel Master controller

Dimensi : 197x197x55 mm

Thickness : 5 mm

Material : PLA ( Polilactic acid )

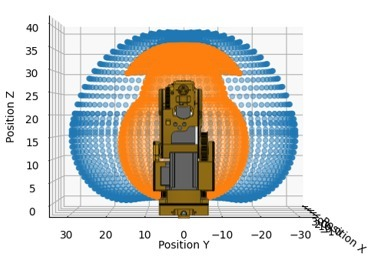
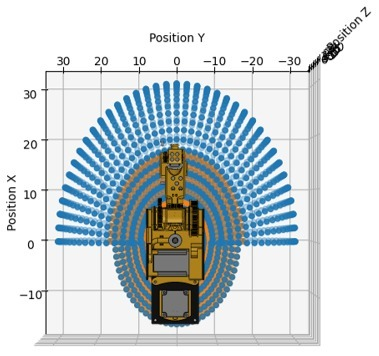
Joystick Master controller

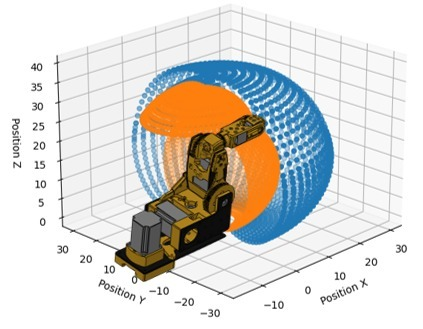
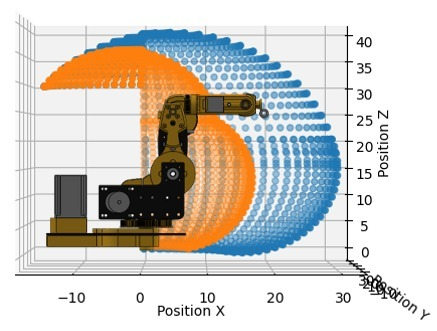
Diameter : 40 mm

Total length of joint : 300 mm

Material : PLA ( Polilactic acid )

* + 1. Ukuran dan Area Kerja Arm Manipulator 6 DOF





**Gambar 2. 21** Area Kerja Arm Manipulator

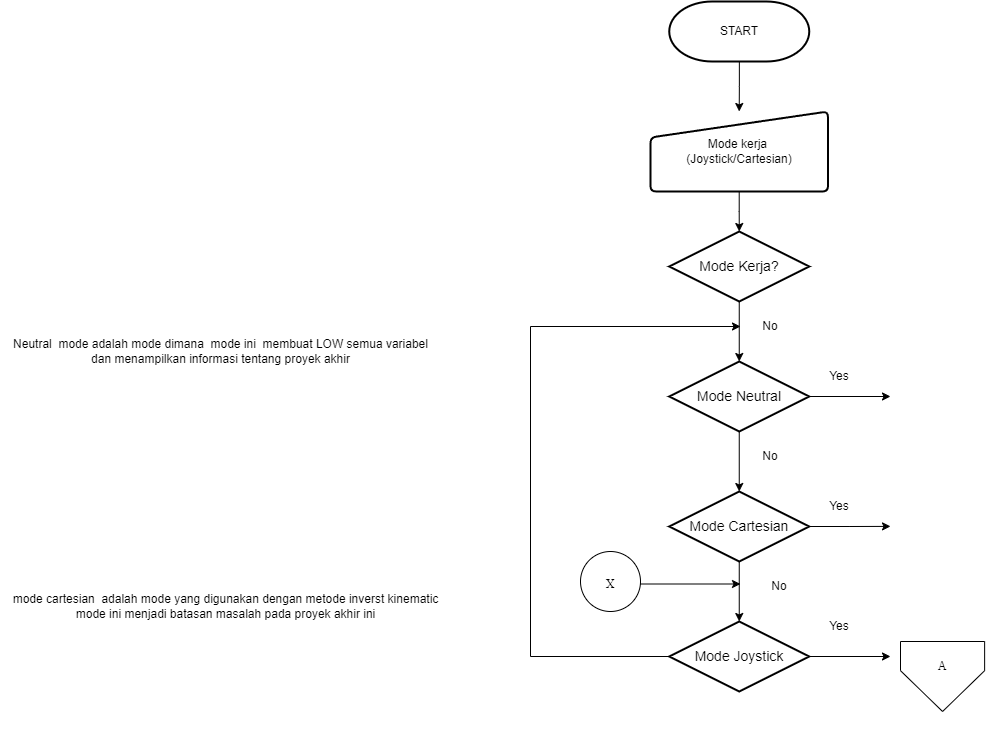
Gambar 2.23 menjelaskan area kerja dari arm manipulator. Area kerja dari manipulator berbentuk seperempat bola dengan area yang berwarna biru merupakan capaian maksimal manipulator sedangkan area yang berwarna jingga merupakan capaian minimum nya.

* + 1. Diagram Alir Sistem

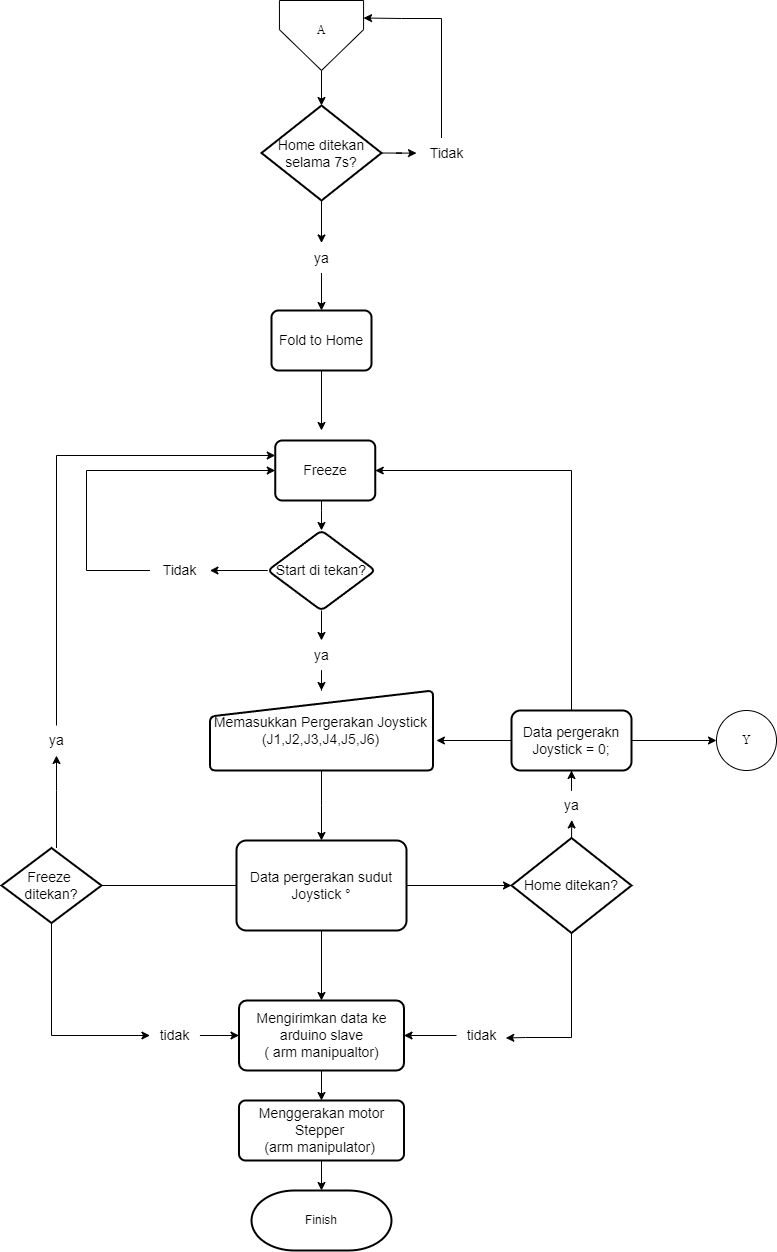
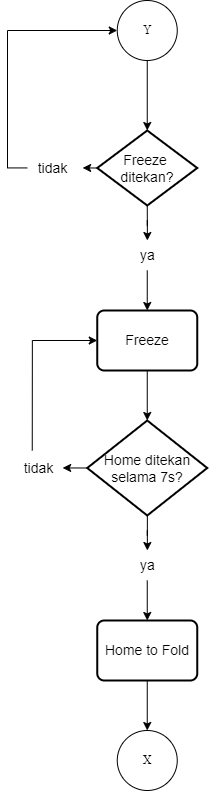
Untuk mulai menjalankan sebuah *Master controller, User* terlebih dahulu memilih mode pergerakkan yang akan dipakai. Terdapat dua mode pergerakkan yaitu mode Cartesian dan mode Joystick. Mode Neutral merupakan mode tambahan dimana pada mode ini semua variable menjadi berlogika LOW. Pada diagram alir sistem di bawah ini akan menjelaskan bagaimana mode Joystick dapat bekerja untuk mengendalikan suatu *arm manipulator.*

.

* + 1. Diagram Alir Sistem Secara Umum



**Gambar 2. 22** Diagram Alir system umum



**Gambar 2. 23** Diagram Alir Sistem secara umum A

Diagram pada gambar 2.24 dan gambar 2.25 diatas menjelaskan bagaimana sistem pengendalian manipulator oleh *master controller* pada mode joystick. Saklar toggle harus diatur untuk memilih mode joystick. Manipulator mula-mula berada pada kondisi *fold* atau melipat. Ketika tombol home ditekan selama 7 detik, maka lengan robot akan bergerak ke posisi *steady/home* (keadaan dimana robot tergambar pada kinematic diagram). Untuk menggerakan manipulator tombol *start* perlu ditekan, sehingga *master arm* akan menghasilkan data derajat sesuai dengan masukkan dari operator.

Data derajat masing – masing joint dan posisi *end-effector* dari robot tertampil pada display LCD yang tersedia. Jika tombol *freeze* ditekan, *master arm* tidak akan menghasilkan data derajat dan tidak akan menggerakan lengan robot bagaimanapun master arm digerakkan sedemikian rupa oleh operator. Robot akan bergerak ke posisi *home* atau posisi ketika semua joint bernilai 0 adalah ketika tombol *home* ditekan. Ketika semua joint bernilai 0 dan tombol *home* ditekan selama 7 detik maka manipulator akan bergerak ke posisi melipat / *fold*.

* 1. Gambaran Sub Sistem

Pada bagian ini akan dijelaskan bagaimana data *degree* yang diterima dari master akan dikonversi menjadi data step untuk menggerakkan *arm manipulator, setting driver* motor yang digunakan oleh masing-masing *joint*, dan bagaimana pergerakan yang dihasilkan dari pulsa yang diberikan.

Diketahui bahwa digunakan beberapa motor stepper pada setiap *joint arm manipulator* untuk menghasilkan pergerakan dan dibutuhkan motor driver yang berguna untuk membangkitkan pulsa-pulsa periodik pada motor stepper, berikut merupakan pengaturan yang diatur pada driver motor:

Driver Motor Joint 1 dan 2

Motor Stepper : Nema 23 5718HB2401 bipolar stepper motor

Driver Motor : TB6600

Pulse/rev : 1600 (OFF-ON-OFF)

Current : 2.5 A (OFF-ON-ON)

Driver Motor Joint 3

Motor Stepper : Nema 17 17HS4401 bipolar stepper motor

*Driver Motor* : TB6600

*Pulse/rev* : 1600 (OFF-ON-OFF)

*Current* : 1.5 A (ON-ON-OFF)

Driver Motor Joint 4 dan 5

Motor Stepper : Nema 11 28HD1411-02 bipolar stepper motor

*Driver Motor* : TB6600

*Pulse/rev*  : 1600 (OFF-ON-OFF)

*Current* : 3.5 A (ON-ON-ON)

Driver Motor Joint 6

Motor stepper : Nema 8 8HS11-0204S bipolar stepper motor

Driver Motor : A4988

Vref : 0.08 V

Berdasarkan keterangan diatas, seluruh setting *microstep* pada driver TB6600 menggunakan *Pulse/rev* sebanyak 1600. Alasan dipilihnya *Pulse/rev* sebesar 1600 dikarenakan berdasarkan perhitungan ,

Dimana 360 merupakan sudut satu putaran penuh dan 1600 merupakan banyaknya *pulse/rev* yang digunakan. Sehingga, diketahui bahwa gerak per pulsanya yaitu sebesar 0.225°. Ketika diberi 1 pulsa maka motor akan bergerak sebesar 0.225° yang membuat pergerakan motor stepper menjadi lebih halus. *Current* (arus) yang diatur pada TB6600 berberda-beda pada tiap *joint*-nya, pengaturan arus pada TB6600 berpacu pada spesifikasi dari masing-masing motor stepper yang digunakan.

Sedangkan, pada joint 6 yaitu motor stepper Nema 8 8HS11-0204S menggunakan *driver* motor A4988. Untuk menggunakan *driver* ini, perlu diatur arus untuk membatasi arus melebihi *rated current* motor. Dilakukan perhitungan,

Diketahui:

*Rated current* motor stepper Nema 8 8HS11-0204S pada joint 6 yaitu 0.2 A.

Maka didapat Vref = 0.08 V.

*Microstepping* yang digunakan yaitu *eighth step* yang berarti 1600 *microstep/revolution.*

* + 1. Konversi Data Sudut ke Data Step

Setiap *joint* dalam *arm manipulator 6 DOF* menggunakan stepper motor untuk membuat pergerakan. Agar stepper motor dapat bergerak diperlukan pulsa-pulsa yang diberikan kepada motor. *Arm Manipulator 6 DOF* ini bergerak mengikuti pergerakan *joystick* pada *master controller* menggunakan komunikasi serial Arduino dimana hanya menggunakan 2 pin yaitu RX dan TX. *Master controller* menerima data dari sensor berupa data *degree* (derajat), data tersebut akan dikirimkan ke *slave Arduino* agar dikonversi menjadi data pulsa dan dieksekusi untuk dapat menggerakkan motor stepper yang ada pada tiap joint. Hal pertama yang dilakukan untuk mengubah data *degree* ke data step yaitu dilakukan perhitungan untuk mendapat parameter *step* pada sudut tertentu. Berikut adalah tabel jangkauan sudut pergerakan dari setiap *joint*.

**Tabel 2. 10** Jangkauan sudut pergerakan arm manipulator

|  |  |
| --- | --- |
|  | Joint 1  Sudut Maksimal : 90°  Sudut Minimal : -90° |
|  | Joint 2  Sudut Maksimal : 90°  Sudut Minimal : -45° |
|  | Joint 3  Sudut Maksimal : 45°  Sudut Minimal : -90° |
|  | Joint 4  Sudut Maksimal : 180°  Sudut Minimal :-180° |
|  | Joint 5  Sudut Maksimal : 90°  Sudut Minimal : -90° |
|  | Joint 6  Sudut Maksimal : 360°  Sudut Minimal : -360° |

Tabel 2.10 merupakan sudut-sudut jangkauan pergerakan yang diinginkan. Pergerakan yang dihasilkan dari motor stepper akan ditransmisikan pada pulley penggerak. Perhitungan yang dilakukan untuk mengetahui parameter step pada sudut tertentu yaitu berdasarkan pulley yang digunakan. Perhitungan ini berfokus pada banyaknya gigi-gigi yang ada pada pulley. Berikut adalah analogi perhitungan yang dilakukan,

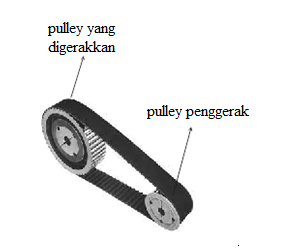
Keterangan:

A = jumlah gigi pada *pulley* penggerak

B = jumlah gigi pada *pulley* yang digerakkan

C = *pulse/rev* yang digunakan

X = *pulse/rev pulley* penggerak



**Gambar 2. 24** Ilustrasi Pulley yang digunakan

Berdasarkan analogi perhitungan pada gambar 2.28 diatas, akan didapat parameter *step* untuk menggerakkan tiap *joint*, berikut adalah data *pulley* berdasarkan jumlah gigi yang digunakan pada tiap *joint*. Analogi perhitungan diatas merupakan analogi yang diambil dari perhitungan kecepatan roda gigi yang dirasa hamper sama penggunaanya pada pulley.

**Tabel 2. 11** Jumlah gigi pada pulley yang digunakan

|  |  |  |
| --- | --- | --- |
| Joint | Pulley Penggerak (gigi) | Pulley yang digerakkan (gigi) |
| 1 | 20 | 96 |
| 2 | 20 | 80 |
| 3 | 20 | 100 |
| 4 | 20 | 56 |
| 5 | 20 | 43 |

Catatan : untuk *joint* 6 tidak menggunakan *pulley* karena merupakan *end effector*.

Pada tabel 2.11 diketahui bahwa tiap joint memanfaatkan *pulley* dengan jumlah gigi penggerak seluruhnya 20 joint dan jumlah gigi pada *pulley* yang digerakkan berbeda-beda pada tiap joint-nya.

Hasil dari perhitungan merupakan *step* untuk menggerakkan *pulley* untuk satu putaran atau 360°. Seluruh hasil tersebut dijadikan parameter pulse untuk menggerakan sudut tertentu dengan memasukkannya pada program.

// Parameter Angle

int maxAngleJ1 = 90;

int minAngleJ1 = -90;

int maxAngleJ2 = 90;

int minAngleJ2 = -45;

int maxAngleJ3 = 45;

int minAngleJ3 = -90;

int maxAngleJ4 = 180;

int minAngleJ4 = -180;

int maxAngleJ5 = 90;

int minAngleJ5 = -90;

int maxAngleJ6 = 360; // infinite

int minAngleJ6 = -360; // infinite

Pada program diatas, batas atas dan batas bawah sudut dari tiap-tiap *joint* dideklarasikan dengan *variable* tipe data *int* dimana tipe data ini digunakan untuk menampung data bilangan bulat.

Berikut merupakan data step hasil dari perhitungan untuk dijadikan parameter step pada program utama.

// Parameter step sampai sudut tertentu

int batasJ1Plus = 1920; // 90 derajat J1

int batasJ1Min = -1920; // -90 derajat J1

int batasJ2Plus = 1600; // 90 derajat J2

int batasJ2Min = -800; // -45 derajat J2

int batasJ3Plus = 1000; // 45 derajat J3

int batasJ3Min = -2000; // 90 derajat J3

int batasJ4Plus = 2240; // 180 derajat J4

int batasJ4Min = -2240; // -180 derajat J4

int batasJ5Plus = 860; // 90 derajat J5

int batasJ5Min = -860; // -90 derajat J5

int batasJ6Plus = 1600; // 180 derajat J6

int batasJ6Min = -1600; // -180 derajat J6

Hasil dari perhitungan tersebut dimasukkan ke dalam program utama untuk dijadikan parameter step sudut tertentu. Setelah didapat parameter sudut dan parameter *step*, konversi dapat dilakukan dengan menggunakan fungsi *mapping*. Fungsi *map* ini digunakan untuk mengkonversi suatu bilangan dalam rentang ke rentang bilangan lain, berikut merupakan program yang digunakan :

// ==== convert data degree ke step

currentStateJ1 = thetaConversions(T1, minAngleJ1, maxAngleJ1, batasJ1Min, batasJ1Plus);

currentStateJ2 = thetaConversions(T2, minAngleJ2, maxAngleJ2, batasJ2Min, batasJ2Plus);

currentStateJ3 = thetaConversions(T3, minAngleJ3, maxAngleJ3, batasJ3Min, batasJ3Plus);

currentStateJ4 = thetaConversions(T4, minAngleJ4, maxAngleJ4, batasJ4Min, batasJ4Plus);

currentStateJ5 = thetaConversions(T5, minAngleJ5, maxAngleJ5, batasJ5Min, batasJ5Plus);

currentStateJ6 = thetaConversions(T6, minAngleJ6, maxAngleJ6, batasJ6Min, batasJ6Plus);

// ==== Fungsi untuk converting degree ke step ====

int thetaConversions (int angle, int minAngle, int maxAngle, int minStep, int maxStep) {

int compare = 0;

compare = map(angle, minAngle, maxAngle, minStep, maxStep);

if (compare > maxStep) {

compare = 0;

}

else if ( compare < minStep) {

compare = 0;

}

return compare;

}

Pada program diatas, *thetaConversions* merupakan sebuah fungsi untuk memetakan nilai yang akan dikonversi. Di dalam kurung terdapat T1-T6 yang merupakan nilai hasil konversi dari data sudut ke data *step*. *MinAngle* dan *maxAngle* merupakan rentang nilai sudut yang akan dikonversi sesuai parameter yang telah dideklarasikan diatas, sedangkan batasMin dan BatasPlus merupakan rentang nilai step yang diinginkan. Sebagai contoh,

currentStateJ1 = thetaConversions(T1, minAngleJ1, maxAngleJ1, batasJ1Min, batasJ1Plus);

artinya *minAngleJ1* merupakan batas bawah dari sudut sebesar -90 derajat, *maxAngleJ1* merupakan batas bawah dari sudut yaitu sebesar 90 derajat. Rentang tersebut akan dikonversi menjadi *step* dengan batasJ1min yaitu batas bawah dari *step* sebesar -1940 dan batasJ1Plus merupakan batas atas dari *step* yaitu sebesar 1940. Hasil dari konversi tersebut masuk dalam T1. Lalu, didapatlah hasil konversi yaitu *currentStateJ1* atau posisi saat ini.

* + 1. Pergerakan Motor Stepper

Telah didapat data pulse hasil konversi untuk dapat menggerakkan motor stepper. Data pulsa tersebut akan dieksekusi oleh *Arduino slave*. Didapatkan 2 cara untuk dapat pergerakan motor stepper.

* + - 1. Algoritma Val

Algoritma ini merupakan fungsi yang dibuat sendiri untuk mendapatkan pergerakan stepper. Berikut merupakan program algoritma val yang digunakan,

// ==== Fungsi untuk menggenerate sinyal yang PRESISI ke stepper ====

void rotateStepper (int gerak, int currentState) {

// Pergerakan J1

if (gerak == 1) {

if (currentState != lastStateJ1) {

if (currentState > lastStateJ1) {

int val = 0;

int i;

val = currentState - lastStateJ1;

for (i = 0; i < val; i++) {

counter1++;

// Putar motor searah jarum jam sesuai dengan step yang diberikan

rotateCW(dirPinJ1, stepPinJ1);

}

}

else if (currentState < lastStateJ1) {

int val = 0;

int i;

val = lastStateJ1 - currentState;

for (i = 0; i < val; i++) {

counter1--;

rotateCCW(dirPinJ1, stepPinJ1);

}

}

}

lastStateJ1 = currentState;

}

Program diatas merupakan algoritma yang dibuat sendiri untuk menggerakkan motor stepper. Dibuat sebuah fungsi yang dapat menggenerate sinyal yang presisi ke stepper. Program diatas merupakan program untuk pergerakan *joint* 1. Jika nilai sekarang (*currentState*) lebih besar dari nilai sebelumnya (*lastStateJ1*), maka untuk mendapatkan *val* yaitu mengurangi nilai sekarang dengan nilai sebelumnya. Hasil *val* yang didapat akan dieksekusi dalam fungsi *for*. Sebaliknya jika nilai sekarang lebih kecil dari nilai sebelumnya, maka untuk mendapatkan nilai *val* yaitu dengan mengurangi nilai sebelumnya dengan nilai sekarang dan selanjutnya dieksekusi oleh fungsi *for*. Hasil akhir didapat bahwa nilai sebelumnya tidak sama dengan nilai sekarang. Pada *joint-joint* yang lain pun menggunakan program yang sama.

* + - 1. *Library Accel Stepper*

Cara yang kedua untuk membuat pergerakan pada motor stepper yaitu menggunakan *library Accel Stepper*. *Library* ini memungkinkan untuk menghubungkan beberapa motor stepper dengan akselerasi dan deselerasi yang terkontrol. Library ini lebih mudah digunakan karena sudah tersedia dari Arduino. Library ini dapat digunakan untuk mengelola beberapa stepper dan membuat semuanya pindah ke posisi yang sama pada waktu yang sama untuk gerakan linier 2d (atau 3d). Berikut penjelasan program *library Accel Stepper*

#include <AccelStepper.h>

#include <MultiStepper.h>

Program diatas menyatakan *library* yang digunakan.

//inisial pin Driver stepper TB6600

AccelStepper stepper1(AccelStepper::DRIVER, 3, 2); // step,DIR

AccelStepper stepper2(AccelStepper::DRIVER, 5, 4); // step,DIR

AccelStepper stepper3(AccelStepper::DRIVER, 7, 6); // step,DIR

AccelStepper stepper4(AccelStepper::DRIVER, 9, 8); // step,DIR

AccelStepper stepper5(AccelStepper::DRIVER, 11, 10); // step,DIR

AccelStepper stepper6(AccelStepper::DRIVER, 13, 12); // step,DIR

MultiStepper steppers;

Digunakan untuk menginisial pin *step* dan *direction* dari driver motor yang digunakan dimana pada project ini menggunakan TB6600 dan A4988.

// Configure each stepper Speed

stepper1.setMaxSpeed(500);

stepper2.setMaxSpeed(500);

stepper3.setMaxSpeed(500);

stepper4.setMaxSpeed(500);

stepper5.setMaxSpeed(500);

stepper6.setMaxSpeed(500);

Digunakan untuk mengatur kecepatan masing-masing stepper motor.

// Then give them to MultiStepper to manage

steppers.addStepper(stepper1);

steppers.addStepper(stepper2);

steppers.addStepper(stepper3);

steppers.addStepper(stepper4);

steppers.addStepper(stepper5);

steppers.addStepper(stepper6);

Selanjutnya akan dieksekusi oleh *library MultiStepper*

long positions[6]; // Array of desired stepper positions

positions[0] = currentStateJ1; // posisi pertama stepper 1

positions[1] = currentStateJ2; // posisi pertama stepper 2

positions[2] = currentStateJ3; // posisi pertama stepper 3

positions[3] = currentStateJ4; // posisi pertama stepper 4

positions[4] = currentStateJ5; // posisi pertama stepper 5

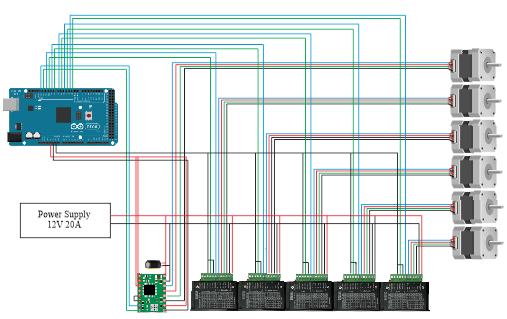
positions[5] = currentStateJ6; // posisi pertama stepper 6

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

Program diatas digunakan untuk menyimpan beberapa data posisi yang diinginkan dari masing-masing stepper

* + 1. Rangkaian Elektrik Motor Stepper dan Driver



**Gambar 2. 25** Rangkaian stepper motor dan driver motor

Rangkaian pada gambar 2.29 diatas merupakan rangkaian motor stepper dan driver motor yang digunakan pada *arm manipulator 6 DoF*. *Power supply* memberikan tegangan 12V pada masing-masing driver. Sebanyak lima motor stepper menggunakan motor driver TB6600 dan satu motor stepper menggunakan A4988. Terdapat kapasitor 4.7µF pada A4988 untuk menghindari lonjakan tegangan yang besar.

* 1. Pengujian
     1. Pengujian pergerakan stepper

Pengujian dilakukan untuk melihat bagaimana data *pulse* hasil perhitungan *pulley* dan keakurasian pada sudut yang ditentukan. Berdasarkan perhitungan *roda gigi* menggunakan jumlah gigi yang ada, hasil yang didapat yaitu berupa *step* untuk mencapai 1 putaran roda gigi atau 360° dengan analogi rumus berikut,

Keterangan:

A = jumlah gigi pada *pulley* penggerak

B = jumlah gigi pada *pulley* yang digerakkan

C = *pulse/rev* yang digunakan

x = *pulse/rev pulley* penggerak

Untuk mendapat hasil *step* pada sudut tertentu, diperlukan sedikit perhitungan tambahan. Berikut merupakan perhitungan tambahan untuk mendapatkan hasil *step* pada sudut tertentu.

**Hasil Pengujian**

Parameter-parameter yang diujikan yaitu terlampir pada tabel 2.12 sampai dengan tabel 2.17 dimana penjelasannya sebagai berikut,

* *Pulse* yang diberikan merupakan masukan pada arm manipulator berupa *pulse* yang didapatkan dari hasil perhitungan.
* Sudut yang dihasilkan merupakan sudut *arm manipulator* yang seharusnya didapat dari masukan pulsa.
* Sudut aktual merupakan sudut yang dihasilkan pada *arm manipulator* dari hasil pengujian
* *Error* merupakan perbandingan kesalahan antara sudut yang dihasilkan dan sudut aktual.
* Pengukuran pada pengujian didasarkan pada alat ukur busur digital. Busur digital yang digunakan dalam pengujian ini memiliki resolusi sebesar 0.05º dan memiliki kearuasian sebesar ±0.03º.

**Tabel 2. 12** Pengujian Sudut Pada Joint 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1920 | 90 | 89.95 | 0.06 |
| 2 | 960 | 45 | 45 | 0.00 |
| 3 | 640 | -30 | -30.05 | 0.17 |
| 4 | 1280 | -60 | -59.5 | 0.83 |
| Rata-Rata Error | | | | 0.27 |

**Tabel 2. 13** Pengujian Sudut Pada Joint 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | Error  (%) |
| 1 | 1600 | 90 | 90 | 0.00 |
| 2 | 800 | 45 | 45.05 | 0.11 |
| 3 | 533 | -30 | -30.1 | 0.33 |
| 4 | 800 | -45 | -45.05 | 0.11 |
| Rata-Rata Error | | | | 0.14 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1000 | 45 | 45 | 0.00 |
| 2 | 555 | 25 | 24.95 | -0.20 |
| 3 | 1333 | -60 | -60.15 | 0.25 |
| 4 | 2000 | -90 | -90.05 | 0.06 |
| Rata-Rata Error | | | | 0.13 |

**Tabel 2. 14** Pengujian Sudut Pada Joint 3

**Tabel 2. 15** Pengujian Sudut Pada Joint 4

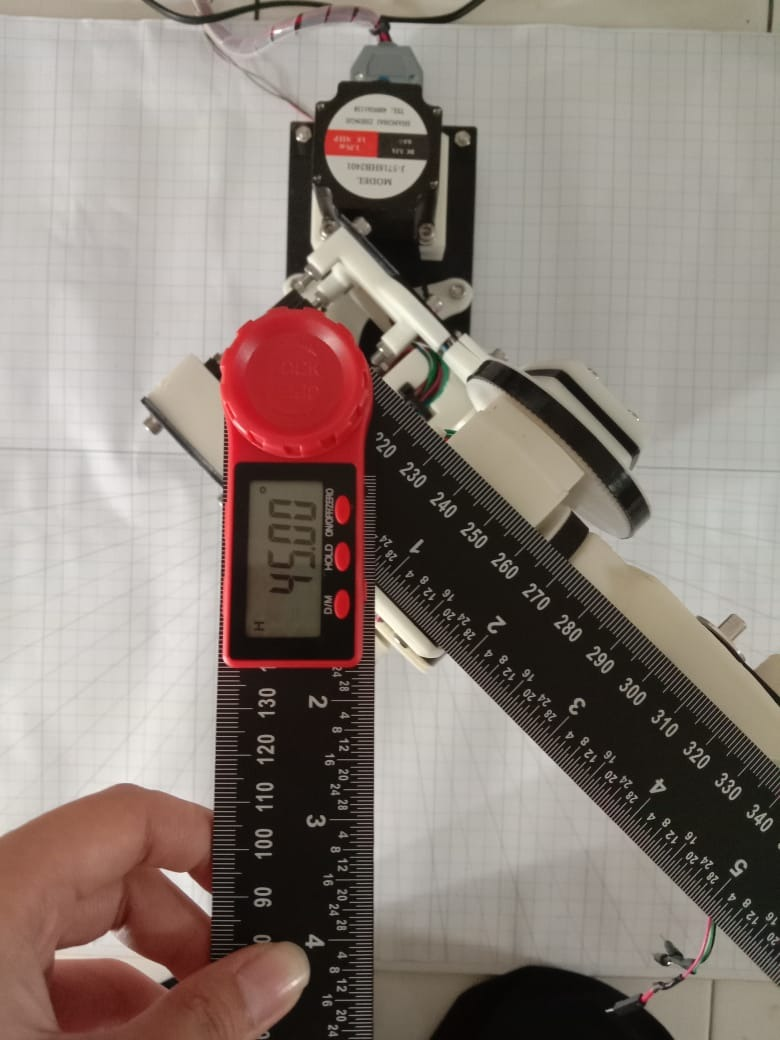
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1867 | 150 | 150.1 | 0.07 |
| 2 | 1120 | 90 | 90 | 0.00 |
| 3 | 560 | -45 | 45.05 | 0.11 |
| 4 | 747 | -60 | 60.1 | 0.17 |
| Rata-Rata Error | | | | 0.09 |

**Tabel 2. 16** Pengujian Sudut Pada Joint 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 860 | 90 | 90 | 0.00 |
| 2 | 430 | 45 | 45.15 | 0.33 |
| 3 | 287 | -30 | 29.95 | -0.17 |
| 4 | 573 | -60 | 60.05 | 0.08 |
| Rata-Rata Error | | | | 0.15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual  (°) | *Error*  (%) |
| 1 | 133 | 30 | 30.00 | 0.00 |
| 2 | 266 | 60 | 60.05 | 0.08 |
| 3 | 800 | 180 | 180.00 | 0.00 |
| 4 | 933 | 210 | 210.05 | 0.02 |
| Rata-Rata Error | | | | 0.03 |

**Tabel 2. 17** Pengujian Sudut Pada Joint 6



**Gambar 2. 26** Salah satu pengujian yaitu pada joint 1 menggunakan busur digital

Gambar 2.30 merupakan salah satu pengujian yang dilakukan yaitu pada joint 1. Pada pengujian diatas dimasukkan pulse sebesar 960. Dari masukan tersebut seharusnya dihasilkan sudut sebesar 45º dan berdasarkan pengujian pun didapatkan hasil sudut actual pada arm manipulator sebesar 45º. Pengujian tersebut dilakukan menggunakan alat ukur busur digital dengan resolusi 0.05º dan akurasi ±0.03º.

**Rata-Rata Persen Error**

**Tabel 2. 18** Hasil Pengujian Rata-Rata Persen Error

|  |  |
| --- | --- |
| **Joint** | **Rata-Rata Error** |
| 1 | 0.27 |
| 2 | 0.14 |
| 3 | 0.13 |
| 4 | 0.09 |
| 5 | 0.15 |
| 6 | 0.16 |
| **Rata-Rata Error Keseluruhan** | 0.27 |

Berdasarkan pengujian yang telah dilakukan, didapatkan hasil pada tabel 2.18 bahwa keakurasian motor stepper berdasarkan *pulse* yang diberikan terbilang sangat baik. Terlihat pada persen *error* yang didapat, pada tiap joint mendapat error hanya pada kisaran nol koma sekian persen. Terlebih lagi setelah di rata-ratakan didapat hasil *error* keseluruhan *joint* yaitu pada 0.14%.

# BAB III

# KESIMPULAN DAN SARAN

3.1 Kesimpulan

Kesimpulan dari proyek akhir ini adalah :

*Arm Manipulator 6 DoF* merupakan lengan robot yang memiliki 6 *joint* dan dikendalikan menggunakan sebuah *master controller*. *Arm Manipulator* ini digerakkkan dengan menggunakan mode *Joystick*.

Motor stepper dipergunakan sebagai aktuator pada proyek ini. Pergerakan motor stepper pada tiap joint dapat digerakkan secara bersamaan dalam satu gerakan. Digunakan setting *microstep* 1600 *pulse/rev* pada tiap driver motor karena optimal diaplikasikan pada *arm manipulator 6 DoF*.

Pengkonversian data *degree* menjadi data *step* memanfaatkan perhitungan rasio jumlah gigi pada pulley yang digunakan untuk dijadikan parameter.

Keakurasian motor stepper yang diaplikasikan pada arm manipulator 6 DoF terhadap pulse yang diberikan menunjukkan bahwa motor stepper memiliki keakurasian sebesar 99,86% dengan rata-rata error terhadap seluruh joint yaitu sebesar 0,14%.

## 3.2 Saran

Berdasarkan pengerjaan mengenai proyek akhir ini, maka terdapat beberapa saran diantaranya sebagai berikut :

* + 1. Dalam penelitian kedepannya di harapkan dapat merancang Mode Kartesian dengan metode inverse kinematic agar robot dapat bergerak berdasarkan koordinat masukan yang diinginkan.
    2. Menggunakan jenis motor dengan torsi daya angkatnya lebih besar.
    3. Menggunakan sensor seperti *limit switch* pada setiap *joint manipulator* yang difungsikan sebagai titik *home* dari masing – masing *joint*. Ini digunakan karena motor stepper tidak memiliki karakteristik untuk mengetahui dia ada di posisi mana apabila sumber daya terputus.

# DAFTAR PUSTAKA

1. Dipankar Mitra, “*How to Improve Motion Smoothness and Accuracy of Stepper Motor*”, *Texas Instruments*.
2. Robert D. Christ dan Robert L. Wernli Sr. *The ROV Manual (A User Guide for Remotely Operated Vehicle Second Edition*). Amerika: Elsevier Ltd. 2014.
3. Anandhya Tradana, “Rancang Bangun Aplikasi Python Dengan Raspberry Untuk Kendali Robot Lengan,” *Proyek Akhir Diploma*, Politeknik Manufaktur Bandung, 2020.
4. Tazkia Rusdiana, “Pengendalian *End Effector* Robot Lengan Untuk Aplikasi *Pick And Place* Benda Posisi Acak Dengan Metode Pengolahan Citra Digital”, *Troyek Akhir Diploma*, Politeknik Manufaktur Bandung, 2021.
5. F. M. Fulchan, R. Ramadhan, R. Wulandari, “Rancang Bangun Purwarupa *Arm Manipulator* *7 Functions* dengan *Master Controller* Untuk Pengaplikasian Pada *Remotely Operated Underwater Vehicle*”. *Proposal Program Kreativitas Mahasiswa,* Politeknik Manufaktur Bandung, 2022.
6. “Schilling Robotics”. Titan 3 Manipulator System, Costeau Ct Amerika Serikat
7. (2020) *6 Axis Robotic Arm*. [Online]. Tersedia: https://www.youtube.com/playlist?list=PLh-o7Bm9fcVy2MhcVkaGOmANr8j7dZ78-
8. (2018) *Robotics* 1 2017.[Online]. Tersedia: https://www.youtube.com/playlist?list=PLT\_0lwItn0sDBE98BsbaZezflB96ws12b
9. (2020) *Parsing Data* Komunikasi Arduino. Tersedia: https://www.youtube.com/watch?v=3MTR3yrYUZA
10. (2021) Serial.read() Arduino. Tersedia: https://www.youtube.com/watch?v=qCjCRBLv\_VM&t=11s
11. (2019) *Big Steppers with Arduino*. [Online]. Tersedia: https://www.youtube.com/watch?v=iY\_4YOlpqyI
12. (2021) Kontrol Motor Stepper dengan *Driver* TB6600. Tersedia: <https://www.cronyos.com/cara-kontrol-motor-stepper-dengan-driver-tb6600-tb6500-arduino/>
13. (2022) *Control Stepper Motor with* A4988 *Driver Module*. Tersedia: <https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/>
14. (2018) Mengenal ROV (*Remotely Operated Vehicle*). Tersedia: http://majalah1000guru.net/2018/03/remotely-operated-vehicle/
15. Arduino Mega 2560 Rev3, Tersedia: <https://store.arduino.cc/products/arduino-mega-2560-rev3>.
16. (2020) *Resistors: Pull-up and pull-down resistors*. Tersedia: https://www.seeedstudio.com/blog/2020/09/22/resistors-pull-up-and-pull-down-resistors/#:~:text=To%20calculate%20the%20pull%2Ddown,so%20our%20formula%20remains%20constant
17. (2022) *Using Functions in a Sketch*. Tersedia: https://docs.arduino.cc/learn/programming/functions
18. (2017) LCD *Display* 20x4. Tersedia: https://www.youtube.com/watch?v=F9IVtKa8C7Q

# LAMPIRAN

Semua dokumentasi mekanik, elektrik, dan program dihimpunkan ke dalam folder github yang berada pada tautan :

<https://github.com/FachriMFulchan/Arm_Manipulator_7_Functions>