**PENCARIAN KOORDINAT *END-EFFECTOR***

**PADA *ARM MANIPULATOR 6 DOF***

**DENGAN METODE DENAVIT-HARTENBERG**

**DAN PENGENDALI *MASTER CONTROLLER***

**PROYEK AKHIR**

Disusun sebagai salah satu syarat untuk

menyelesaikan pendidikan Diploma III

**Oleh:**

Fachri Maulana Fulchan

219341029



**PROGRAM STUDI TEKNIK MEKATRONIKA**

**JURUSAN TEKNIK OTOMASI MANUFAKTUR DAN MEKATRONIKA**

**POLITEKNIK MANUFAKTUR BANDUNG**

**2022**

**LEMBAR PENGESAHAN**

**PENCARIAN KOORDINAT *END-EFFECTOR***

**PADA *ARM MANIPULATOR 6 DOF***

**DENGAN METODE DENAVIT-HARTENBERG**

**DAN PENGENDALI *MASTER CONTROLLER***

**Oleh:**

Fachri Maulana Fulchan

219341029

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

Robotika sangat berguna di dunia industri, salah satu aplikasi robotika di dunia industri adalah *arm manipulator*. *Arm manipulator* adalah mekanik elektronik yang menyerupai lengan manusia sehingga lebih sering disebut dengan lengan robot. Lengan robot terdiri dari lengan (*link*) dan sendi (*joint*).

Pada proyek akhir ini dibuat purwarupa *arm* *manipulator 6* DOF dengan pengendalinya yaitu *master controller*. *Master controller* akan memberikan sinyal derajat pada masing – masing *joint manipulator* sehingga dapat bergerak ke posisi tertentu. Atau dengan kata lain, kendali robot yang dipergunakan adalah *Forward Kinematics* (masukan: derajat, keluaran: posisi dan rotasi). Untuk menyederhanakan perhitungan *Forward Kinematics*, diterapkanlah metode *Denavit-Hartenberg*. Dengan begitu, masukan derajat pada tiap *joint* dapat diterjemahkan menjadi rotasi dan posisi dari *end-effector* dengan relatif mudah.

Berdasarkan percobaan yang telah dilakukan, *manipulator* 6 DOF dapat mengetahui posisi dimana *end-effector* nya berada dengan besar *error* rata -rata di sumbu x = 1,94 %, di sumbu y = 0,85 %, dan sumbu z = 3,36 %. Rotasi *end-effector* berdasarkan *base frame* juga memperoleh hasil dengan validitas 100%.

**Kata Kunci :** *Arm Manipulator 6 DOF, Master Controller, Forward Kinematics, Denavit-Hartenberg Method.*

**ABSTRACT**

*Robotics is very useful in the industrial world, one of the applications of robotics in the industrial world is the arm manipulator. Arm manipulator is an electronic mechanic that resembles a human arm so it’s more often called a robotic arm. The robot arm consists of an arm (link) and a joint (joint).*

*In this final project, a 6 DOF arm manipulator prototype was made with the controller, namely the master controller. The master controller will give a degree signal to each joint manipulator so that it can move to a certain position. Or in other words, the robot control used is Forward Kinematics (input: degree, output: position amd rotation). To simplify the calculation of Forward Kinematics, the Denavit-Hartenberg method is applied. In this way, the degree input at each joint can be translated into rotation and position of the end-effector with relative ease.*

*Based on the experiments that have been carried out, the 6 DOF manipulator can find out the position where the end-effector is with the average error on the x-axis = 1.94%, on the y-axis = 0.85%, and the z-axis = 3.36% . The end-effector rotation based on the base frame also obtained results with 100% validity.*

***Keywords****: Arm Manipulator 6 DOF, Master Controller, Forward Kinematics, Denavit-Hartenberg Method.*

# KATA PENGANTAR

Segala puji bagi Allah yang hanya kepada-Nya kami memuji, memohon pertolongan, dan mohon ampunan. Kami berlindung kepada-Nya 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 jalan-Nya 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, Alhamdulillah penulis dapat menyelesaikan proyek akhir ini dengan judul: “PENCARIAN KOORDINAT *END-EFFECTOR* PADA *ARM MANIPULATOR 6 DOF* DENGAN METODE DENAVIT-HARTENBERG DAN 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, Riska Wulandari 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, 15 Juli 2022

Penulis

# DAFTAR ISI

[ABSTRAK i](#_Toc110322689)

[KATA PENGANTAR iii](#_Toc110322690)

[DAFTAR ISI v](#_Toc110322691)

[DAFTAR GAMBAR vi](#_Toc110322692)

[DAFTAR TABEL viii](#_Toc110322693)

[BAB I 1](#_Toc110322694)

[1.1. Tujuan 1](#_Toc110322695)

[1.2. Teknologi 1](#_Toc110322696)

[1.3. Batasan Pembahasan 2](#_Toc110322697)

[BAB II 3](#_Toc110322698)

[2.1. Landasan Teori 3](#_Toc110322699)

[2.1.1. Lengan Robot (*Arm Manipulator*) 3](#_Toc110322700)

[2.1.2. *Master Controller* 3](#_Toc110322701)

[2.1.3. *Stick diagram* 4](#_Toc110322702)

[2.1.4. *Kinematics* 5](#_Toc110322703)

[2.1.5. *Denavit-Hartenberg Method* 9](#_Toc110322704)

[2.1.6. *Yaw, Pitch, Roll Angles (Tait-Bryan Angles)* 10](#_Toc110322705)

[2.2. Metodologi Penelitian 12](#_Toc110322706)

[2.3. Gambaran Umum Sistem 13](#_Toc110322707)

[2.4. Diagram Alir Sistem 14](#_Toc110322708)

[2.5. Gambaran Sub Sistem 16](#_Toc110322709)

[2.5.1. Perangkat Mekanik 16](#_Toc110322710)

[2.5.2. Perangkat Elektrik 22](#_Toc110322711)

[2.5.3. Pencarian Posisi dan Rotasi *End-Effector* 24](#_Toc110322712)

[2.5.4. Program Implementasi 28](#_Toc110322713)

[2.6. Hasil Pengujian 33](#_Toc110322714)

[2.6.1. Pengujian Posisi *End-Effector* Terhadap *Base Frame* 33](#_Toc110322715)

[2.6.2. Pengujian Rotasi *End-Effector* Terhadap *Base Frame* 38](#_Toc110322716)

[BAB III 42](#_Toc110322717)

[3.1. Kesimpulan Hasil Implementasi 42](#_Toc110322718)

[3.2. Saran 43](#_Toc110322719)

[DAFTAR PUSTAKA 44](#_Toc110322720)

[LAMPIRAN 45](#_Toc110322721)

# DAFTAR GAMBAR

[**Gambar 2. 1** *Articulated Manipulator* dan *Scara Manipulator* 3](#_Toc110322729)

[**Gambar 2. 2** *Master Controller* Pabrikan *Schilling Robotics* 3](#_Toc110322730)

[**Gambar 2. 3** *Stick diagram* dari *Articulated Manipulator* dan *Scara Manipulator* 4](#_Toc110322731)

[**Gambar 2. 4** Gambaran *Forward Kinematics* 5](#_Toc110322732)

[**Gambar 2. 5** Orientasi dan Posisi 5](#_Toc110322733)

[**Gambar 2. 6** Rotasi Sumbu 6](#_Toc110322734)

[**Gambar 2. 7** Format *Rotation Matrix* 6](#_Toc110322735)

[**Gambar 2. 8** Perkalian *Dot Product* Untuk Mendapatkan R0-6 6](#_Toc110322736)

[**Gambar 2. 9** Perpindahan Sumbu dan Format *Displacement Vector* 7](#_Toc110322737)

[**Gambar 2. 10** Format *Homogeneous Transformation Matrix* 7](#_Toc110322738)

[**Gambar 2. 11** Perkalian *Dot Product* Untuk Mendapatkan H0-6 8](#_Toc110322739)

[**Gambar 2. 12** Cara Kerja *Forward Kinematics* dan *Invers Kinematics* 8](#_Toc110322740)

[**Gambar 2. 13** *Denavit-Hartenberg Parameter Table* 9](#_Toc110322741)

[**Gambar 2. 14** Format Pengisian *DH Parameter* pada HTM 10](#_Toc110322742)

[**Gambar 2. 15** *Yaw, Pitch, Roll* Pada Aeronautika 11](#_Toc110322743)

[**Gambar 2. 16** Rumus Mendapatkan *φ(Roll)*, *θ (Pitch)*,  *ψ(Yaw)* Dari *Rotation Matriks* 11](#_Toc110322744)

[**Gambar 2. 17** Diagram Alir Perancangan dan Pembuatan 12](#_Toc110322745)

[**Gambar 2. 18** Desain *Arm Manipulator* dan *Master Controller* 13](#_Toc110322746)

[**Gambar 2. 19** Gambaran Umum Sistem *Arm Manipulator* dan *Master Controller* 13](#_Toc110322747)

[**Gambar 2. 20** Diagram Alir Sistem Secara Umum 15](#_Toc110322748)

[**Gambar 2. 21** *Kinematics Diagram Arm Manipulator 6 DOF* 16](#_Toc110322749)

[**Gambar 2. 22** *Joint* Pada *Arm Manipulator* 17](#_Toc110322750)

[**Gambar 2. 23** Dimensi *Master Controller* 18](#_Toc110322751)

[**Gambar 2. 24** *Panel Master* dan *Master Arm* 19](#_Toc110322752)

[**Gambar 2. 25** Ukuran Area Kerja *Arm Manipulator* 20](#_Toc110322753)

[**Gambar 2. 26** Pergerakan Positif dan Negatif Pada Setiap *Joint* 21](#_Toc110322754)

[**Gambar 2. 27** Pengasutan Elektrik *Arm Manipulator* 22](#_Toc110322755)

[**Gambar 2. 28** *Motherboard Master Controller* 23](#_Toc110322756)

[**Gambar 2. 29** *Flow Chart* Pencarian HTM Menggunakan *DH Method* 24](#_Toc110322757)

[**Gambar 2. 30** *Stick diagram Arm Manipulator* *6 DOF* 25](#_Toc110322758)

[**Gambar 2. 31** Format Pengisian *DH Parameter* 26](#_Toc110322759)

[**Gambar 2. 32** Perkalian *Dot Product* Untuk Menghasilkan H0-6 27](#_Toc110322760)

[**Gambar 2. 33** Bentuk Hasil Matriks Homogen yang Dihasilkan 27](#_Toc110322761)

[**Gambar 2. 34** Rumus Mendapatkan *φ(Roll)*, *θ (Pitch)*,  *ψ(Yaw)* Dari *Rotation Matriks* 28](#_Toc110322762)

[**Gambar 2. 35** Matriks *H0-6* yang Dihasilkan 30](#_Toc110322763)

[**Gambar 2. 36** Nilai *Yaw, Pitch, Roll* yang Dihasilkan 31](#_Toc110322764)

[**Gambar 2. 37** Display LCD Menampilkan Parameter 32](#_Toc110322765)

[**Gambar 2. 38** Visualisasi *Arm Manipulator*, *Master Controller*, dan *Motor Driver* 33](#_Toc110322766)

[**Gambar 2. 39** Pengujian Posisi Sampel Ke-1 35](#_Toc110322767)

[**Gambar 2. 40** Pengujian Posisi Sampel ke-5 35](#_Toc110322768)

[**Gambar 2. 41** Pengujian Posisi Sampel ke-10 36](#_Toc110322769)

[**Gambar 2. 42** Pengujian Posisi Sampel ke-15 36](#_Toc110322770)

[**Gambar 2. 43** Pengujian Posisi Sampel Ke-20 37](#_Toc110322771)

[**Gambar 2. 44** Pengujian Rotasi Sampel Ke-1 39](#_Toc110322772)

[**Gambar 2. 45** Pengujian Rotasi Sampel Ke-3 40](#_Toc110322773)

[**Gambar 2. 46** Pengujian Rotasi Sampel Ke-5 40](#_Toc110322774)

[**Gambar 2. 47** Pengujian Rotasi Sampel Ke-7 41](#_Toc110322775)

[**Gambar 2. 48** Pengujian Rotasi Sampel Ke-10 41](#_Toc110322776)

# DAFTAR TABEL

[**Tabel 2. 1** *DH Parameter* yang Berhasil Didapatkan 26](#_Toc110322777)

[**Tabel 2. 2** Pengujian Posisi *End-Effector* 34](#_Toc110322778)

[**Tabel 2. 3** Pengujian Rotasi *End-Effector* 39](#_Toc110322779)

# BAB I

**PENDAHULUAN**

## Tujuan

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

1. Mempelajari struktur dan pergerakan dari lengan robot 6 *degree of freedom*.
2. Robot mampu bergerak dengan dikendalikan oleh sebuah *pendant* yang disebut *Master Controller*.
3. Robot dapat mengetahui posisi dan rotasi dari *end-effector* nya menggunakan kendali *Forward Kinematics* dengan metode *Denavit-Hartenberg*.
4. Informasi yang dibutuhkan seperti masukkan derajat, posisi *end-effector,* dan rotasi *manipulator* dapat tertampil dalam sebuah *interface* LCD.

## 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 *Stepper Driver Module*

b. TB6600 *Stepper Driver Module*

1. Sensor : a. Potensiometer 100K Ohm

b. *Analog Joystick Module*

1. *Digital Input* : a. *Tactile Push Button*

b. Saklar *Toggle* *ON-OFF-ON*

1. *Digital Output* : a. LED (*Light Emitting Diode*)

Perangkat Lunak :

1. Arduino IDE
2. Solidworks 2020
3. Eagle PCB Design
4. Python 3.10.0

## Batasan Pembahasan

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

1. Menggunakan desain rangka robot yang bersifat *open source*.
2. Robot hanya digerakkan secara *Forward Kinematics* (masukan derajat menjadi keluaran posisi).
3. Robot tidak mempunyai fitur *teaching*.
4. *Workload* / beban kerja dari lengan robot tidak menjadi parameter yang diperhitungkan.
5. Komunikasi antara lengan robot dan *master controller* menggunakan komunikasi *serial UART Arduino* dengan segala keterbatasannya.

# BAB II

**LAPORAN TEKNIK**

* 1. Landasan Teori

### Lengan Robot (*Arm Manipulator*)

**Gambar 2. 1** *Articulated Manipulator* dan *Scara Manipulator*

*Arm manipulator* adalah konstruksi 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 industri 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. Gambar 2.1 adalah beberapa contoh *arm manipulator* yang sering diterapkan pada industri yaitu *articulated manipulator* dan *scara* manipulator. 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.*

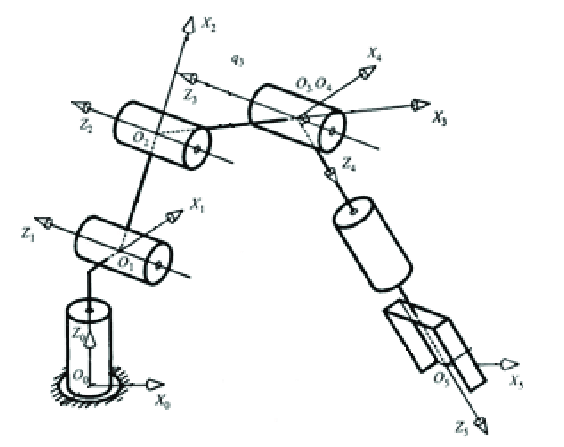
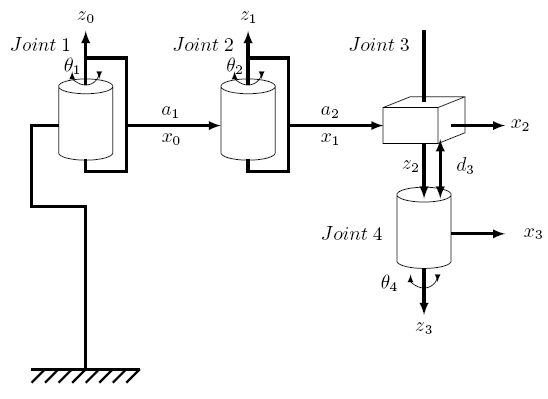
### *Master Controller*



**Gambar 2. 2** *Master Controller* Pabrikan *Schilling Robotics*

*Master controller* merupakan replika tangan berukuran kecil yang difungsikan untuk menggerakan *manipulator*. Setiap *joint* dari *manipulator* atau DOF (*Degree of Freedom*), digerakkan oleh satu sensor yang berada pada *master arm*. Dengan menggunakan *controller* ini, dapat membuat pergerakan dari *manipulator* menjadi simpel dan intuitif. Pergerakan dari masing – masing *joint* pun dapat di batasi sehingga *manipulator* tidak menjangkau area atau konfigurasi sistem yang dapat menimbulkan masalah. Fitur lain yang dimiliki *controller* ini adalah *individual joint freeze*, *joint diagnostics*, *error checking*, dan lain sebagainya. Gambar 2.2 merupakan salah satu *master controller* pabrikan sebuah perusahaan bernama *Schilling Robotics*.

### *Stick diagram*

**Gambar 2. 3** *Stick diagram* dari *Articulated Manipulator* dan *Scara Manipulator*

*Stick diagram* adalah sebuah diagram yang menampilkan hubungan antara *links* dan *joints* di dalam sebuah *manipulator*. Biasanya diagram ini hanya mengandung objek geometris seperti garis, tabung, kubus, dan lain sebagainya. Tujuan dari penggambaran diagram ini adalah untuk memperoleh persamaan matematis untuk mengontrol lengan robot. Contoh *stick diagram* tertera pada Gambar 2.3. Aturan penting dalam penggambaran *stick diagram* adalah gambar berada pada keadaan ketika semua *joint* bernilai 0.

Kinematik diagram terdiri dari beberapa komponen antara lain :

1. *Joints*, merupakan komponen yang menimbulkan pergerakan dalam suatu *manipulator*. Biasanya digambar dalam bentuk tabung untuk *revolute joint* dan kubus untuk *prismatic joint*.
2. *Links*, merupakan komponen kaku dalam suatu *manipulator* dan digambar dalam bentuk garis pada *stick diagram*. Panjang setiap *links* biasanya sudah diketahui dan bisa langsung dimasukkan kedalam formulasi.
3. *Coordinate Frame*, merupakan sumbu *frame* x, y, dan z yang berguna untuk merepresentasikan gerakan lengan robot. Setidaknya butuh 3 jenis *coordinate frame* dalam *stick diagram* :
4. *Coordinate frame* pada base *manipulator (base frame*).
5. *Coordinate frame* pada masing-masing joint *manipulator.*
6. *Coordinate frame* pada *end-effector manipulator*.
7. Pergerakan positif, merupakan arah pergerakan positif dari masing masing *joint* *manipulator*, biasanya dilambangkan dengan *theta* (θ). Cara mencari pergerakan positif ini adalah dengan menggunakan kaidah tangan kanan. Arah jempol mengarah pada sumbu z, maka arah sisa jari yang telungkup merupakan arah pergerakan positif dari *joint* tersebut.

### *Kinematics*

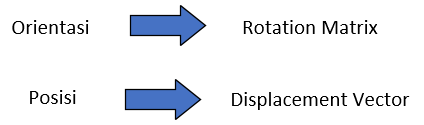
Dalam fisika, kinematika didefinisikan sebagai ilmu yang mempelajari gerak tanpa mempersoalkan penyebabnya. Dalam kasus lengan robot, maka kinematika adalah studi analitis pergerakan lengan robot yang bergerak tanpa memperhatikan gaya yang menyebabkan pergerakan tersebut. Di robotika, kita mengenal dua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

1. *Forward Kinematics* (*angle to positions*)



**Gambar 2. 4** Gambaran *Forward Kinematics*

Secara sederhana, *forward kinematics* dapat didefinisikan sebagai pergerakan dengan memasukan nilai derajat pada masing-masing *joint* lengan robot sehingga menimbulkan keluaran yakni posisi dan orientasi dari *end effector* terhadap *base frame*. Skema *forward kinematics* tertera pada Gambar 2.4.

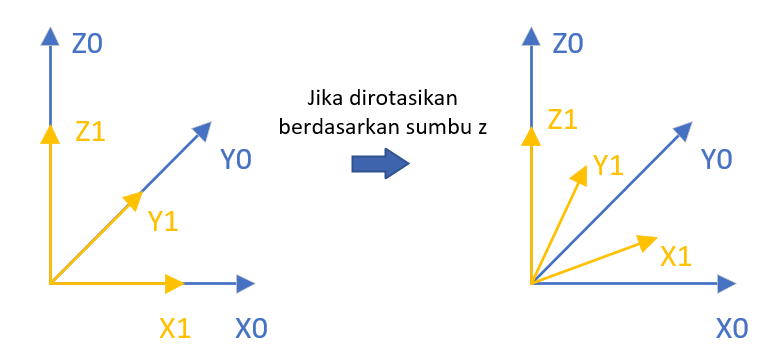


**Gambar 2. 5** Orientasi dan Posisi

Seperti pada Gambar 2.5, nilai orientasi dan posisi dalam *end-effector* dapat dijabarkan secara matematis kedalam bentuk matriks. Nilai orientasi *end-effector* relatif terhadap *base frame* di tuangkan dalam bentuk matriks yang bernama *rotation matrix ()* sedangkan nilai posisi *end-effector* relatif terhadap *base frame* disajikan ke dalam bentuk vektor yang bernama *displacement vector* ).

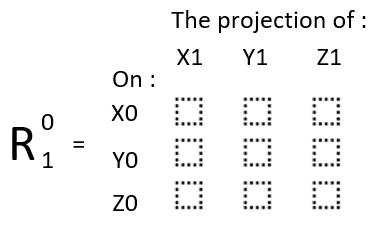
1. *Rotation Matrix*

*Rotation matrix* menjelaskan bagaimana *coordinate frame* yang satuberotasi relatif terhadap *coordinat frame* acuannya. Konsep dasar dari *matrix* ini adalah proyeksi vektor sederhana dalam matematika.



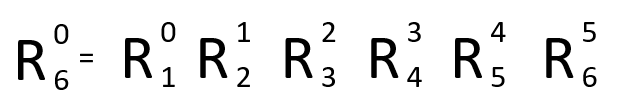
**Gambar 2. 6** Rotasi Sumbu

Gambar 2.6 menggambarkan bagaimana sebuah sumbu berotasi terhadap sumbu lainnya. Maka didapatkan proyeksi X1 terhadap X0, X1 terhadap Y0, X1 terhadap Z0, Y1 terhadap X0 dan seterusnya.



**Gambar 2. 7** Format *Rotation Matrix*

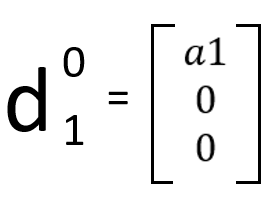
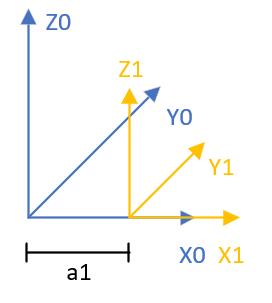
Setelah semua nilai proyeksi didapatkan, nilai tersebut dimasukkan ke dalam sebuah matriks 3x3 yang formatnya dapat dilihat pada Gambar 2.7. Nilai-nilai tersebut merupakan nilai rotasi suatu *frame* terhadap *frame* acuannya.



**Gambar 2. 8** Perkalian *Dot Product* Untuk Mendapatkan R0-6

Pada kasus *manipulator* 6 DOF, untuk menemukan nilai rotasi *end-effector* relatif terhadap *base frame* dibutuhkan 6 matriks rotasi. Keenam matriks tersebut di *dot product* kan sehingga menghasilkan matriks rotasi R0-6 seperti terlihat pada Gambar 2.8.

1. *Displacement Vector*

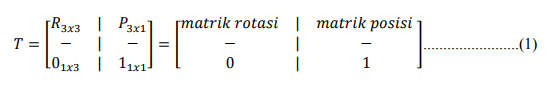


**Gambar 2. 9** Perpindahan Sumbu dan Format *Displacement Vector*

*Displacement vector* menjelaskan bagaimana titik pusat *coordinate frame* yang satu berjarak terhadap titik pusat *coordinate frame* acuannya seperti yang ditunjukkan pada Gambar 2.9.

Namun pada kasus robot lengan 6 DOF, untuk mendapatkan d0-6 tidak dapat dilakukan dengan cara mengalikan secara *dot product* keseluruhan *displacement vector* yang didapat dari *joint* 1 sampai *joint* 6. Pencarian d0-6 harus dilakukan menggunakan *homogeneous transformation matrix*.

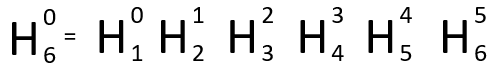
1. *Homogeneous Transformation Matrix*

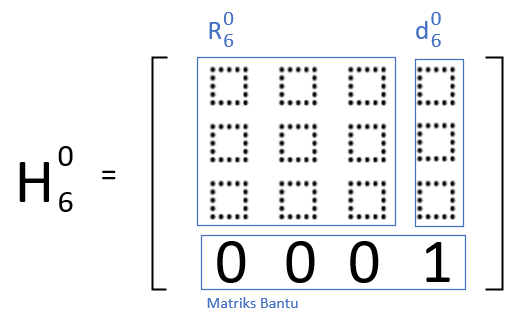


**Gambar 2. 10** Format *Homogeneous Transformation Matrix*

*Homogeneous transformation matrix* (HTM)merupakan matriks yang didalamnya tersusun dari *rotation matrix* dan *displacement matrix*. Strukturnya seperti ditunjukkan pada Gambar 2.10.

Dimana merupakan *rotation matrix* dan merupakan *displacement vector.* dan merupakan *matrix* bantu agar kedua matriks tersebut dapat disatukan. Kelebihan dari HTM ini adalah dapat dikalikan secara *dot product* seperti *rotation matriks*

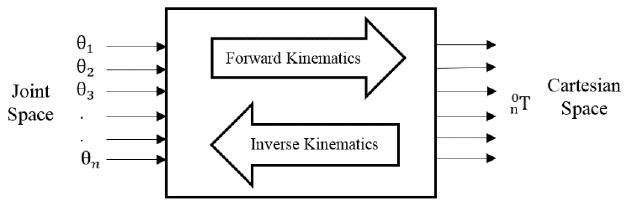




**Gambar 2. 11** Perkalian *Dot Product* Untuk Mendapatkan H0-6

Dengan begitu, maka kalkulasi posisi *end-effector* terhadap *base frame manipulator* 6 DOF yakni d0-6 dapat ditemukan (bisa dilihat pada Gambar 2.11). Begitulah pendekatan *forward kinematics* untuk mencari posisi dan orientasi *end effector*.

1. *Inverse Kinematics (positions to degree)*



**Gambar 2. 12** Cara Kerja *Forward Kinematics* dan *Invers Kinematics*

Dimana :

θ = variabel joint

x = posisi dan orientasi end effector

Gambar 2.12 menjelaskan bahwa *invers kinematics* merupakan kebalikan dari *forward* kinematics. *Inverse kinematics* digunakan untuk mencari kombinasi sudut yang harus diberikan pada *arm manipulator* agar *end effector* bergerak menuju koordinat posisi yang ditentukan. Berbeda dengan konsep *forward kinematics*, dimana sudut dari tiap *joint* sudah ditentukan untuk mencari koordinat posisi dari *end effector*. Ciri khas dari penyelesaian menggunakan persamaan *inverse kinematics* adalah mempunyai lebih dari satu solusi penyelesaian, karena dimungkinkan memiliki lebih dari satu kombinasi sudut tiap *joint* untuk mendapatkan koordinat posisi dari *end effector*.

### *Denavit-Hartenberg Method*

*Denavit-Hartenberg method* merupakan salah satu metode pendekatan untuk mengetahui hubungan antara *links* dan *joints* pada *kinematics diagram* sehingga diketahui pergerakan kinematikanya. Atau secara aplikatif, DH *method* ini digunakan pada aplikasi *forward kinematics* sebagai *shortcut* untuk mencari *homogeneous transformation matrix*. Sehingga nantinya tidak perlu mencari *rotation matrix* maupun *displacement vector* terlebih dahulu untuk mendapatkan HTM.

Metode ini diperkenalkan pada tahun 1955 oleh Jacques Denavit dan Richard Hartenberg. Untuk menggunakan *shortcut* ini ada beberapa tahapan antara lain :

1. Menggambar *coordinate frame* pada *stick diagram* berdasarkan 4 *Denavit-Hartenberg Rules*.
2. *Rule* #1

*the Z axis must be the axis of rotation for a revolute joint, ot the direction of motion for a prismatic joint.*

1. *Rule* #2

*the X axis must be perpendicular both to its own Z axis, and the Z axis of the frame before it.*

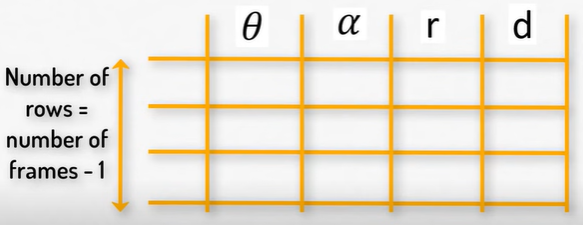
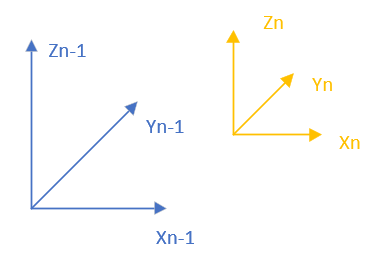
1. *Rule* #3

*all frames must follow the right hand rule*.

1. *Rule* #4

*each X axis must intersect the Z axs of the frame before* it

1. Mengisikan *Denavit-Hartenberg Parameter Table* sesuai dengan aturan.

**Gambar 2. 13** *Denavit-Hartenberg Parameter Table*

Format dari DH *Parameter Table* tertera pada Gambar 2.13. Terdapat 4 parameter yang harus dicari, yakni :

1. θ

Merupakan berapa banyak rotasi yang terjadi di Zn-1 untuk membuat axis Xn-1 sejajar dengan axis Xn. *Theta rotation* pun harus diikutkan.

1. α

Merupakan berapa banyak rotasi yang terjadi di Xn untuk membuat axis Zn-1 sejajar dengan axis Zn. *Theta rotation* tidak perlu diikutkan. Walaupun Xn yang menjadi sumbu putarnya, namun yang seolah berputar adalah axis n-1.

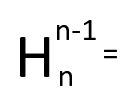
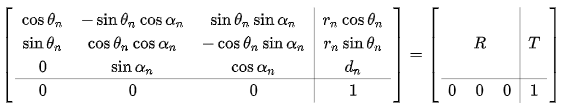
1. r

Merupakan jarak antara titik pusat axis n-1 dan axis n sepanjang sumbu Xn.

1. d

Merupakan jarak antara titik pusat axis n-1 dan n sepanjang sumbu Zn-1.

1. Mengisikan DH *Parameter* yang telah didapatkan ke dalam *homogeneous transformation matrix* dengan aturan :

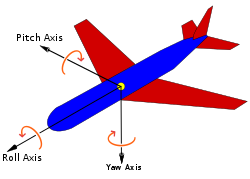
 

**Gambar 2. 14** Format Pengisian *DH Parameter* pada HTM

Parameter DH yang telah dicari kemudian di *assign* kedalam HTM dengan format sesuai dengan Gambar 2.14. Dengan begitu didapatkan *homogeneous transformation matrix* yang mengandung *rotation matrix* dan *displacement vector* di dalamnya sebagai pendekatan untuk mengetahui posisi dan rotasi *end-effector* terhadap *base frame* pada lengan robot.

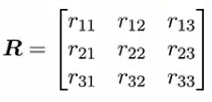
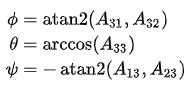
### *Yaw, Pitch, Roll Angles (Tait-Bryan Angles)*

*Yaw, Pitch, Roll* merupakan tiga *angle* yang menjelaskan orientasi dari sebuah benda (*rigid* body) mengacu kepada suatu *coordinate system*. Ketiga *angle* ini dikemukakan oleh Peter Guthrie Tait dan George H. Bryan sebagai penjabaran lebih lanjut terhadap konsep *Euler angles.* Aplikasi dari konsep ini sering diterapkan pada bidang aeronautika dan *engineering*.



**Gambar 2. 15** *Yaw, Pitch, Roll* Pada Aeronautika

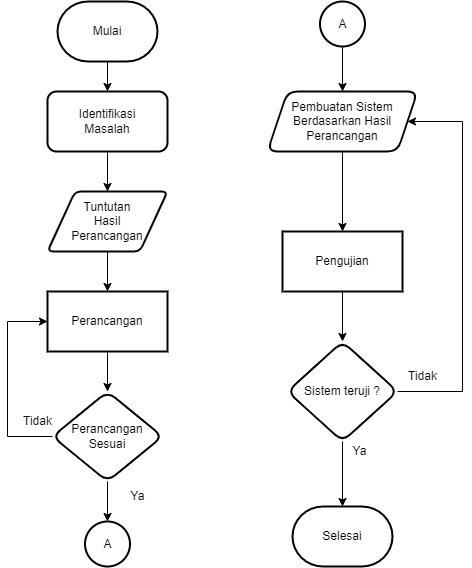
Rotasi *yaw*, *pitch*, dan *roll* dapat tergambarkan pada Gambar 2.15. Rotasi di sekitar sumbu yang terbentang dari depan ke belakang disebut *roll,* rotasi di sekitar sumbu yang terbentang dari kiri ke kanan disebut *pitch*, sedangkan rotasi pada sumbu vertical disebut *yaw*. *Yaw, pitch, roll* juga dapat diterapkan pada bidang robotika. *Angle* - *angle* ini akan menjelaskan bagaimana robot bergerak dan berotasi berdasarkan sumbu acuanya.

**Gambar 2. 16** Rumus Mendapatkan *φ(Roll)*, *θ (Pitch)*,  *ψ(Yaw)* Dari *Rotation Matriks*

Nilai *angle yaw, pitch,* dan *roll* dapat di didapatkan dari *rotation matrix* dengan menggunakan berbagai pendekatan. Salah satu pendekatan nya ialah *x-convention* untuk 3-1-3 *extrinsic euler angle* seperti yang tertera pada Gambar 2.16.

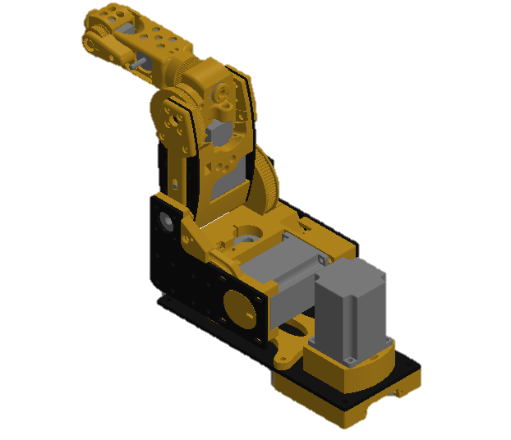
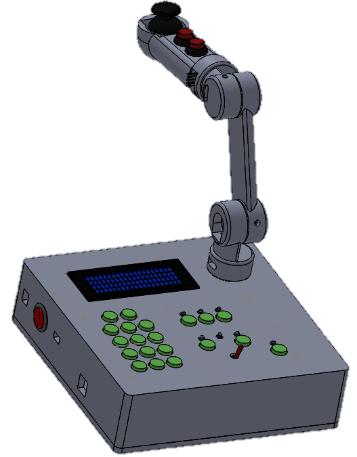
* 1. Metodologi Penelitian



**Gambar 2. 17** Diagram Alir Perancangan dan Pembuatan

Gambar 2.17 menjelaskan bagaimana tahapan - tahapan dalam pembuatan alat. Pertama- tama masalah di identifikasi yang solusinya akan dipecahkan oleh alat ini. Tuntutan hasil perancangan dipertimbangkan karena ada syarat minimal dalam penyelesaian proyek akhir. Kemudian adalah tahap perancangan dan hasil perancangan pun dibuat. Alat yang telah selesai kemudian di tanamkan program dan algoritma kedalam nya yang selanjutnya akan diuji.

* 1. Gambaran Umum Sistem

**Gambar 2. 18** Desain *Arm Manipulator* dan *Master Controller*

*Arm manipulator 6 DOF* merupakan robot berbentuk lengan yang memiliki 6 sendi pergerakan. Motor stepper dipergunakan sebagai aktuator untuk menggerakan sendi-sendinya. Pengendalian *manipulator* ini bergantung pada sebuah *pendant* yang dinamakan *master controller*. Setiap *joint* pada *manipulator* akan digerakkan oleh masing – masing satu sensor yang terpasang pada *master controller.*

*Master controller* akan menghasilkan data derajat dari pembacaan sensor. Data sensor tersebut kemudian dikalkulasikan menggunakan pendekatan *Denavit-Hartenberg Method* untuk memprediksi posisi dan orientasi *end-effector manipulator* berdasarkan *base frame* nya. Desain dari *arm manipulator* dan *master controller* yang telah di rancang tertera pada Gambar 2.18.

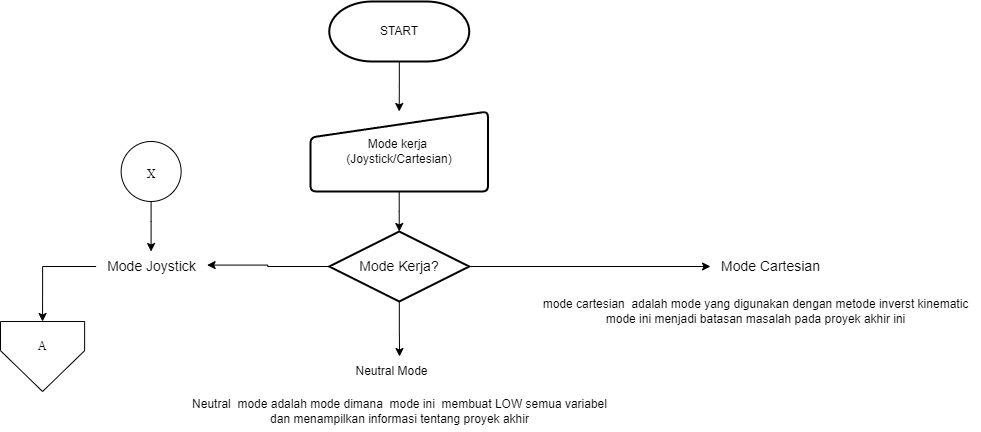


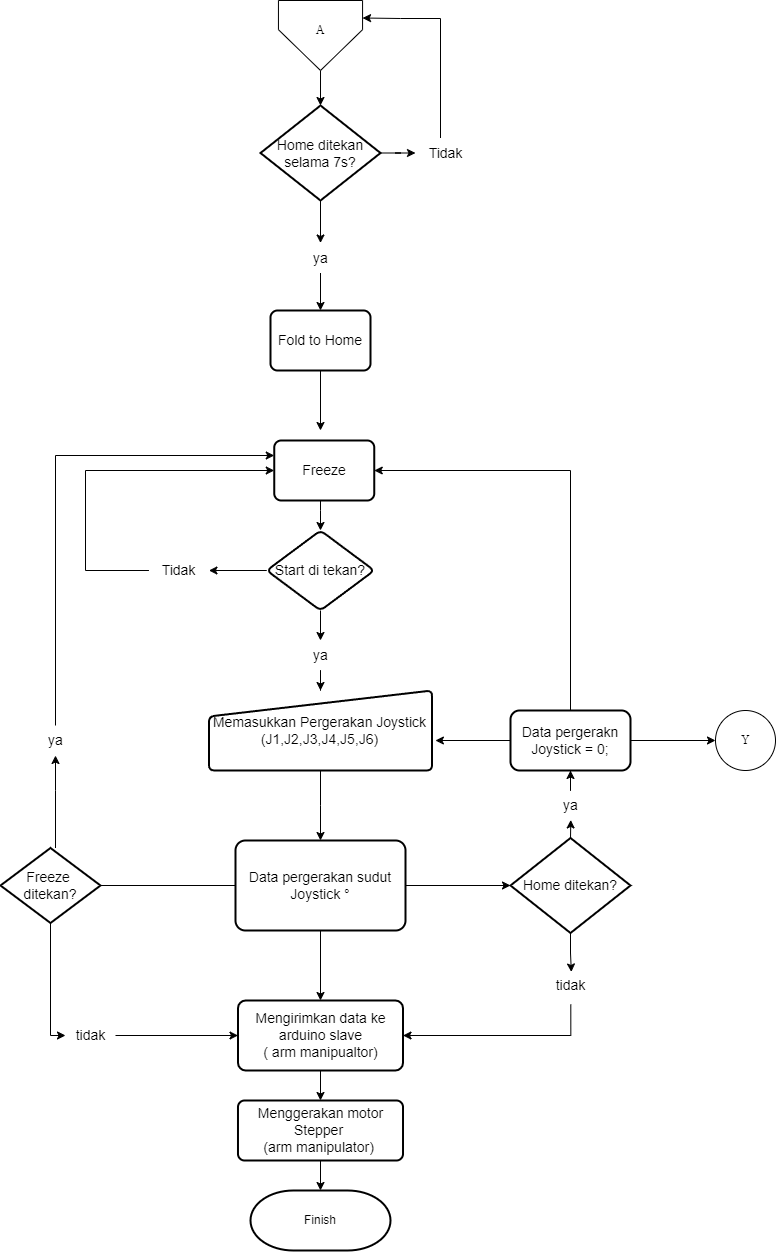
**Gambar 2. 19** Gambaran Umum Sistem *Arm Manipulator* dan *Master Controller*

Terlihat pada Gambar 2.14, *master controller* dibangun atas potensiometer, modul joystick, *push button*, dan saklar toggle. Data yang dihasilkan dari komponen – komponen tersebut kemudian akan diproses oleh Arduino (*master*). Data hasil pemroresan akan di tampilkan pada LCD sekaligus dikirim melalui komunikasi *serial* *UART* kepada *slave*.

Data yang dikirim akan diterima oleh Arduino (*slave*). Data tersebut dalam bentuk derajat. Kemudian data derajat tersebut akan diubah menjadi pulsa digital yang diperuntukkan untuk driver motor sehingga dapat menggerakkan motor stepper yang berada pada setiap *joint* *arm manipulator* / lengan robot.

* 1. Diagram Alir Sistem



**Gambar 2. 20** Diagram Alir Sistem Secara Umum

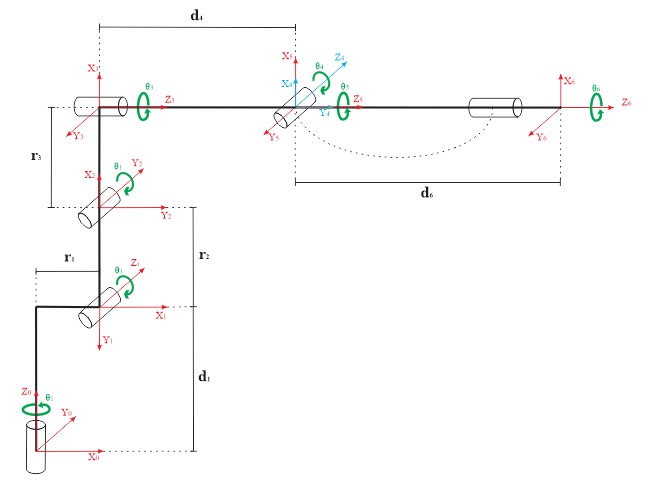
Diagram 2.20 menjelaskan bagaimana sistem pengendalian *manipulator* oleh *master controller* pada mode *joystick*. Saklar toggle harus diatur untuk memilih mode joystick. *Manipulator* mula-mula berada pada kondisi *fold* atau melipat. Ketika tombol *home* ditekan selama 7 detik, maka lengan robot akan bergerak ke posisi *steady*/*home* (keadaan dimana robot tergambar pada *stick 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 0dan tombol *home* ditekan selama 7 detik maka *manipulator* akan bergerak ke posisi melipat / *fold*.

* 1. Gambaran Sub Sistem

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

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*



**Gambar 2. 21** *Kinematics Diagram Arm Manipulator 6 DOF*

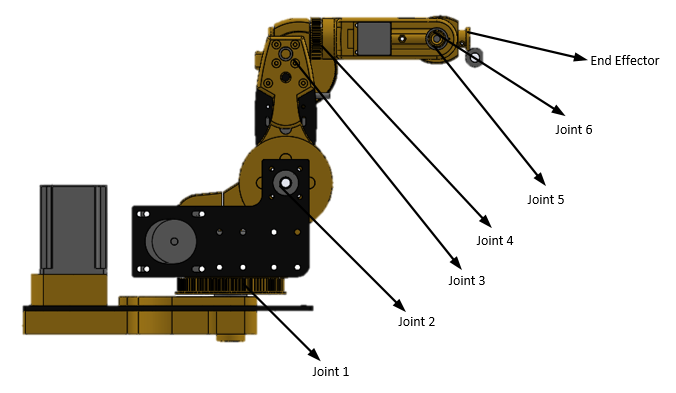
*Manipulator* terdiri dari *links* dan *joints*. Rangka dari *manipulator* 6 DOF ini dapat tergambarkan pada stick diagram di Gambar 2.21.

Panjang link :

r1 = 47 mm d1 = 133 mm

r2 = 110 mm d4 = 117.5 mm

r3 = 26 mm d6 = 28 mm



**Gambar 2. 22** *Joint* Pada *Arm Manipulator*

Motor stepper pun dipasang pada masing – masing *joint* sebagai sumber pergerakan (seperti yang bisa dilihat pada Gambar 2.22). Berikut merupakan spesifikasi dari masing – masing motor stepper di setiap *joint* :

*Joint* 1 dan 2

Jenis : Nema 23 5718HB2401

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

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

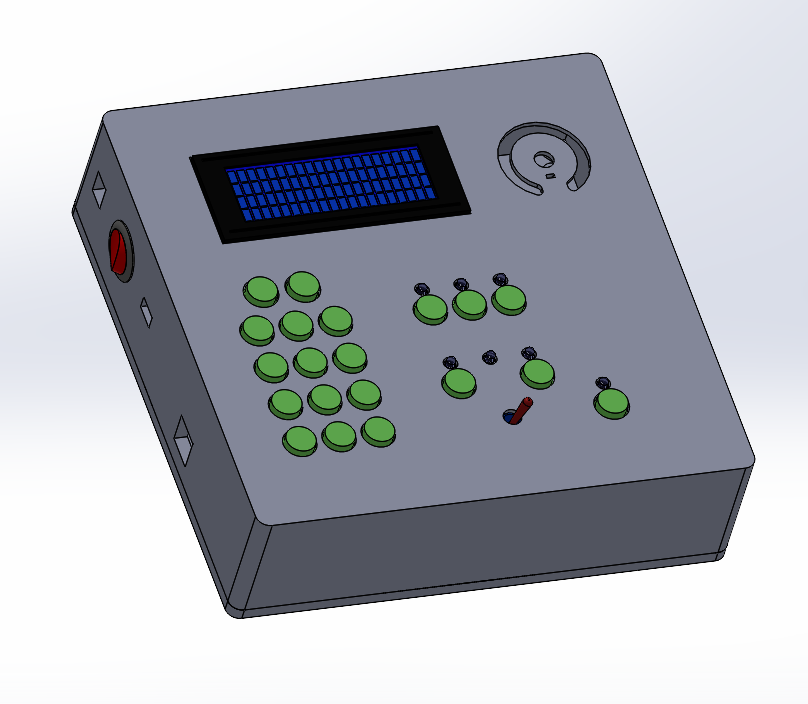
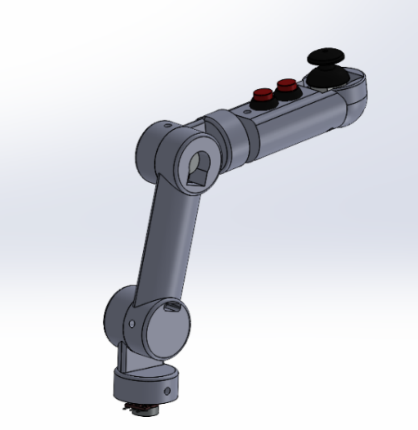
1. *Master Controller*



**Gambar 2. 23** Dimensi *Master Controller*

*Master controller* di rancang agar *master arm* dapat dioperasikan dengan memaksimalkan sudut perputaran pada potensiometer. Adapun dimensi pada *Master Controller* dapat dilihat pada Gambar 2.23.

*Controller* ini terdiri dari 2 bagian utama yaitu *panel master* dan *master arm*. *Panel master* memiliki beberapa tombol yang difungsikan sebagai pemilih mode pergerakan, mode kecepatan, mode *start, mode freeze,* dan lain sebagainya. Pada *panel master* pun terdapat display LCD yang akan menampilkan sudut pergerakan dari *manipulator*. Sedangkan *master arm* merupakan replika tangan berukuran kecil yang difungsikan untuk menggerakan *manipulator* secara intuitif.

**Gambar 2. 24** *Panel Master* dan *Master Arm*

Gambar 2.24 menampilkan visualisasi dari *panel master* dan *master arm*. Berikut merupakan penjabaran dari dimensi – dimensinya :

Panel *master controller* :

Dimensi : 197x197x55 mm

Thickness : 5 mm

Material : PLA ( Polilactic Acid )

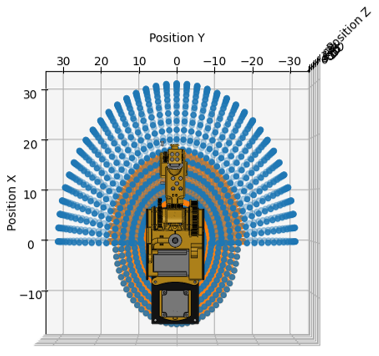
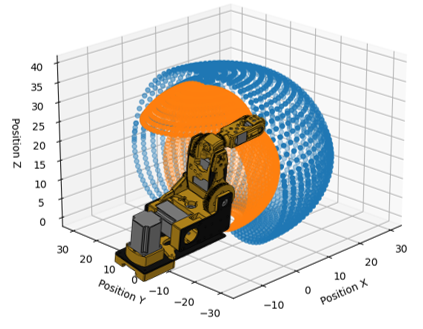
*Joystick master controller* :

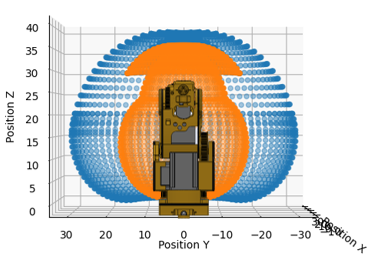
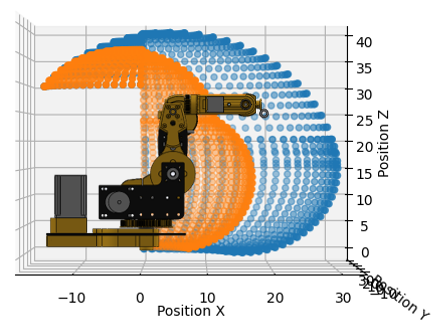
Diameter : 40 mm

Total length of joint : 300 mm

Material : PLA ( Polilactic Acid)

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

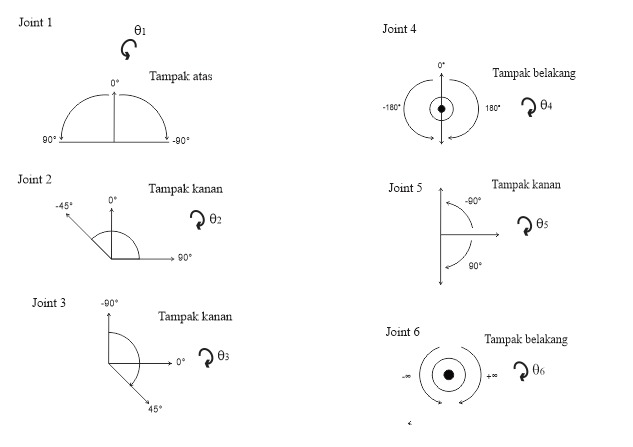
 

**Gambar 2. 25** Ukuran Area Kerja *Arm Manipulator*

Gambar 2.25 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.

Pergerakan dari masing – masing *joint manipulator* dibatasi pada sudut tertentu. Ini dimaksudkan agar *manipulator* tidak menjangkau area atau konfigurasi sistem yang dapat menimbulkan masalah.

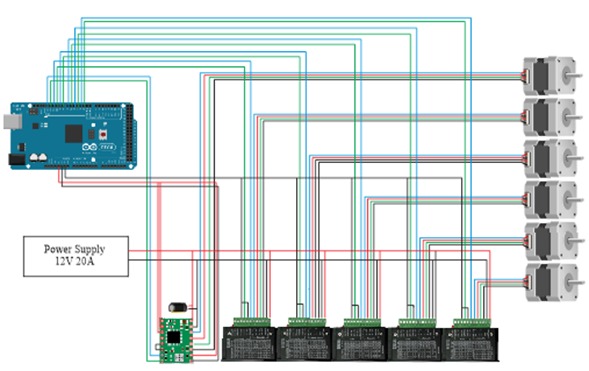


**Gambar 2. 26** Pergerakan Positif dan Negatif Pada Setiap *Joint*

Jangkauan sudut masing-masing sendi robot tergambarkan pada gambar 2.26. Lebih detailnya adalah sebagai berikut :

* *Joint* 1: 90o ≤ θ1 ≤ -90o
* *Joint* 2: -45o ≤ θ2 ≤ 90o
* *Joint* 3: -90o ≤ θ3 ≤ 45o
* *Joint* 4: -180 o ≤ θ4 ≤ 180o
* *Joint* 5: -90 o ≤ θ5 ≤ 90 o
* *Joint* 6: -tak hingga ≤ θ6 ≤ tak hingga

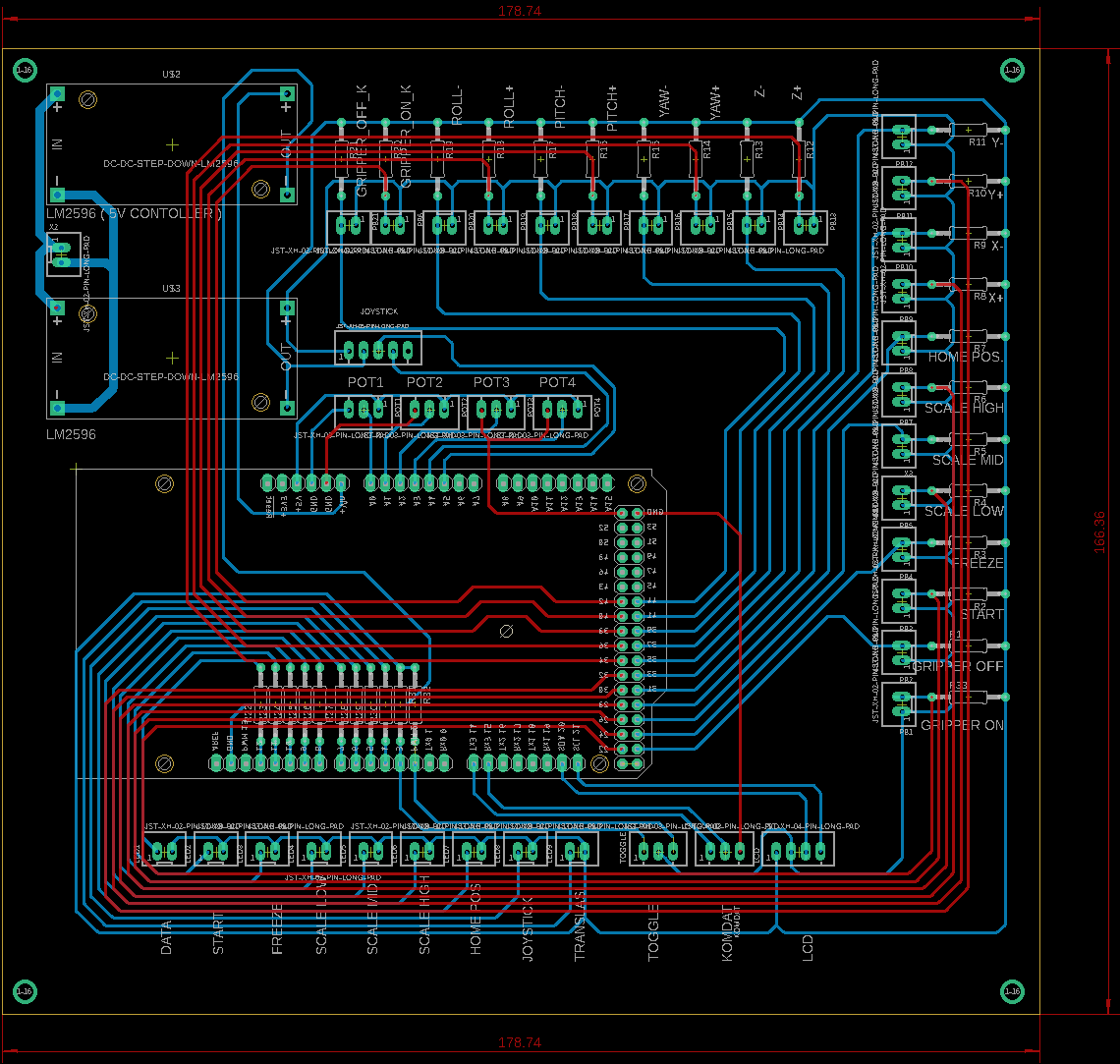
1. Perangkat Elektrik
2. *Arm Manipulator*

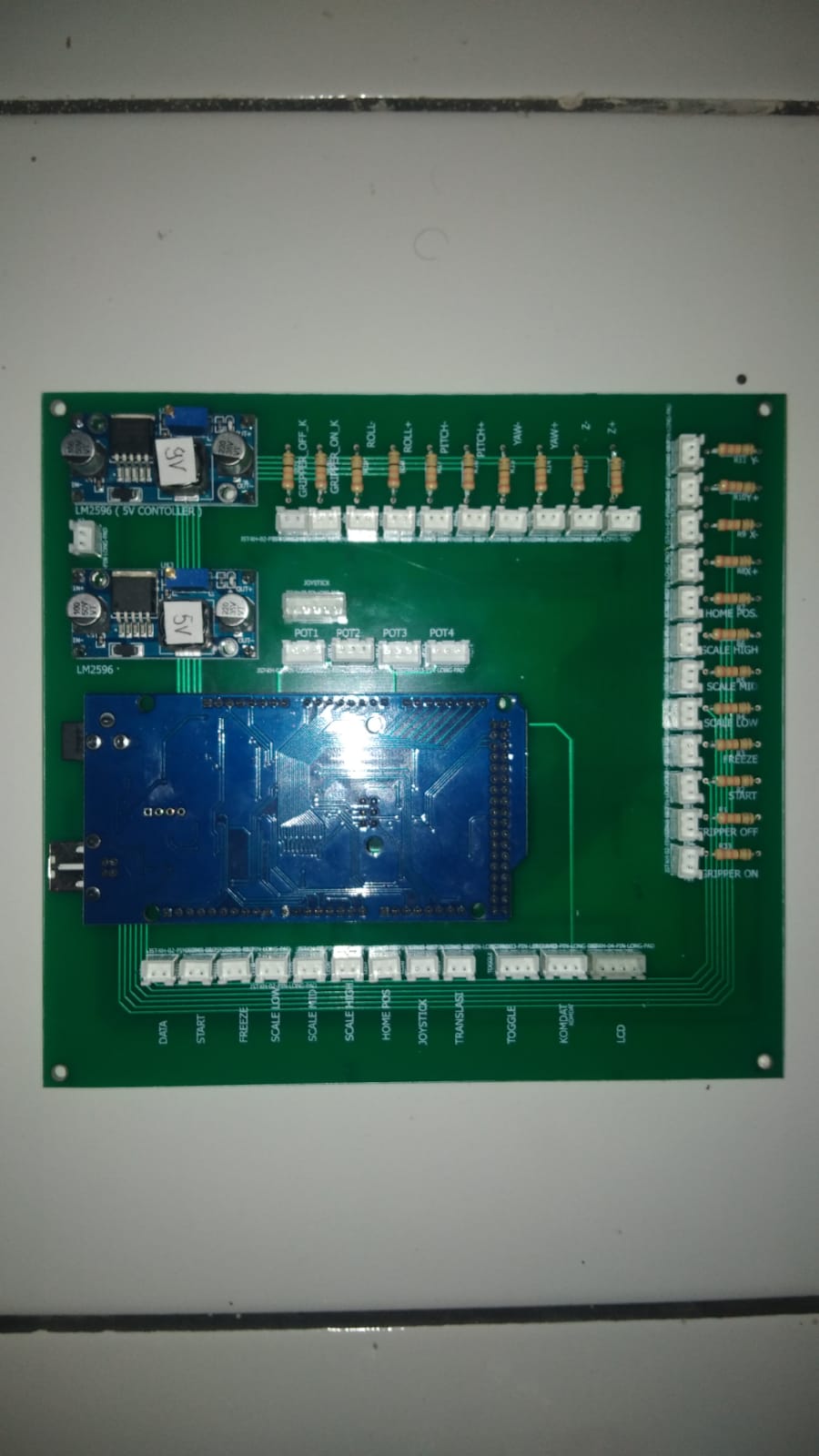


**Gambar 2. 27** Pengasutan Elektrik *Arm Manipulator*

Komponen elektrik *manipulator* terdiri dari motor stepper, motor driver A4988, motor driver TB6600, *power supply*, dan *microcontroller arduino*. Semua motor driver di atur pada *microstepping 1600* step/rev. Pengasutannya tertera pada Gambar 2.27.

1. *Master Controller*

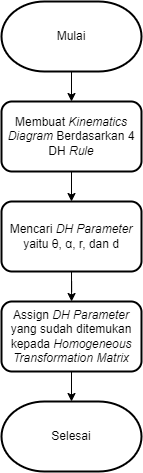


****

**Gambar 2. 28** *Motherboard Master Controller*

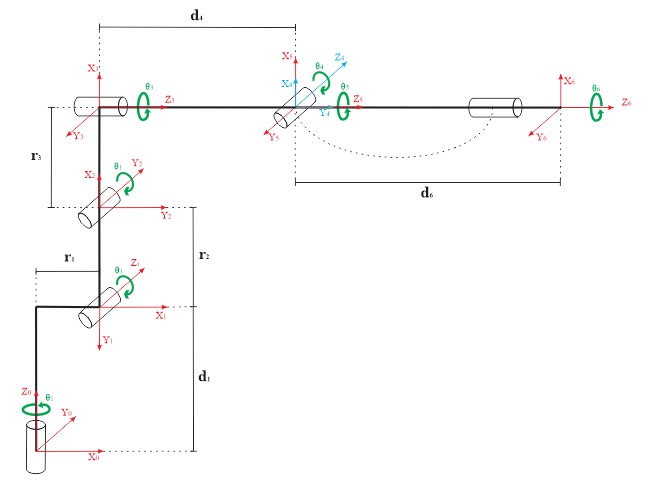
Pengasutan elektrik pada *master controller* dibuat secara terpusat pada suatu rangkaian *motherboard*. Rangkaian ini terdiri dari *microcontroller arduino*, sensor potensiometer, sensor *analog joystick*, *push buttons, pull down resistors* saklar toggle, *power supply unit for microcontroller, power supply unit for sensors*, LCD I2C *connections,* dan UART *communication to slave arduino*. Hasil rancangan PCB dapat dilihat pada gambar 2.28.

1. Pencarian Posisi dan Rotasi *End-Effector*



**Gambar 2. 29** *Flow Chart* Pencarian HTM Menggunakan *DH Method*

Pada bagian ini akan dijelaskan bagaimana pencarian nilai *homogeneous transformation matrix* yang mengandung *rotation matrix* dan *displacement vector* dengan pendekatan *Denavit-Hartenberg method*. Hasil dari parameter ini digunakan untuk menggerakan robot secara *forward kinematics*. Gambar 2.29 merupakan diagram alir penyelesaiannya.



**Gambar 2. 30** *Stick diagram Arm Manipulator* *6 DOF*

*Stick diagram* dari robot *manipulator* yang dibuat dapat dilihat pada Gambar 2.30

Panjang link :

r1 = 47 mm d1 = 133 mm

r2 = 110 mm d4 = 117.5 mm

r3 = 26 mm d6 = 28 mm

Penggambaran *stick diagram* ini telah mengikuti 4 *Denavit-Hartenberg rules* yakni:

1. Rule #1

*The Z axis must be the axis of rotation for a revolute joint, ot the direction of motion for a prismatic joint.*

1. Rule #2

*The X axis must be perpendicular both to its own Z axis, and the Z axis of the frame before it.*

1. Rule #3

*All frames must follow the right hand rule*.

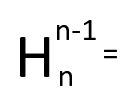
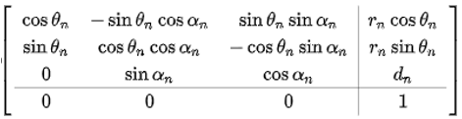
1. Rule #4

*Each X axis must intersect the Z axis of the frame before* it.

Setelah *stick diagram* dibuat, selanjutnya adalah mencari DH *parameter* yakni θ, α, r, dan d. Diperlukan pemahaman definisi yang dalam terhadap 4 parameter tersebut agar nilai yang didapat tidak keliru. Baris pada DH *table* ini berjumlah 6 dikarenakan *frame* pada *manipulator* berjumlah 7 buah. Aturannya adalah (*row = number frame – 1).* Nilai DH *parameter* yang didapat adalah tertera pada Tabel 2.1 berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **θ** | **α** | **r** | **d** |
| **1** | θ1 | -90 o | r1 | d1 |
| **2** | -90o+ θ2 | 0 o | r2 | 0 |
| **3** | θ3 | -90 o | r3 | 0 |
| **4** | θ4 | 90 o | 0 | d4 |
| **5** | θ5 | -90 o | 0 | 0 |
| **6** | θ6 | 0 o | 0 | d6 |

**Tabel 2. 1** *DH Parameter* yang Berhasil Didapatkan

**Gambar 2. 31** Format Pengisian *DH Parameter*

Selanjutnya adalah memasukkan parameter DH yang telah didapat ke dalam *homogeneous transformation matrix* dengan format seperti Gambar 2.31.

H0-1 =

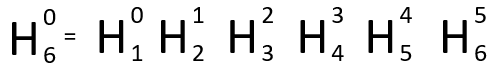
H1-2 =

H2-3 =

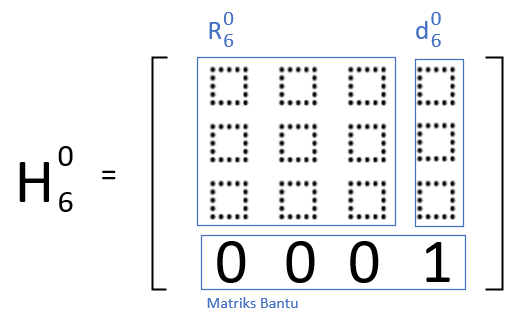
H3-4 =

H4-5 =

H5-6 =

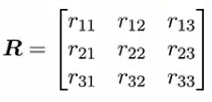
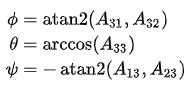


**Gambar 2. 32** Perkalian *Dot Product* Untuk Menghasilkan H0-6



**Gambar 2. 33** Bentuk Hasil Matriks Homogen yang Dihasilkan

Setelah mendapatkan formulasi keseluruhan nilai matriks. Matriks H0-6 dapat dicari dengan cara mengalikan secara *dot product* keseluruhan matriks yang didapat. Formulasinya tertera pada Gambar 2.32. Didalam matriks H0-6 mengandung *rotation matrix* R0-6 dan *displacement vector* d0-6 yang digunakan untuk mengetahui posisi dan rotasi dari *end-effector* lengan robot terhadap *base frame*. Skemanya dapat dilihat pada Gambar 2.33.

**Gambar 2. 34** Rumus Mendapatkan *φ(Roll)*, *θ (Pitch)*,  *ψ(Yaw)* Dari *Rotation Matriks*

*Displacement vector* (d0-6) yang merupakan representasi posisi *end-effector,* nilainya dapat dengan mudah dipahami. Sedangkan *rotation matrix* (R0-6), nilai akhirnya sukar dipahami karena itu merupakan hasil proyeksi satu sumbu (*end-effector frame*) dengan sumbu yang lainnya (*base* frame) dan menghasilkan 6 nilai proyeksi yang cukup rumit. Untuk memudahkan pemahaman rotasi lengan robot maka diterapkan konsep *yaw, pitch,* dan *roll angle* yang nilainya dapat didapatkan dari *rotation matrix*. Rumus tersebut tertera pada Gambar 2.34.

1. Program Implementasi

Berikut merupakan program implementasi kalkulasi *homogeneous transformation matrix* untuk mencari posisi dan rotasi dari *end-effector* terhadap *base frame*nya. Program ditulis menggunakan bahasa C Arduino.

// Determine End-Effector Position

generateHTMWithDH (steppJ1, steppJ2, steppJ3, steppJ4, yawJ5, rollJ6);

tampilMatriks (H0\_6); // buat nampilin Matriks H0-6

Program tersebut merupakan 2 fungsi utama untuk menghasilkan nilai posisi dan rotasi pergerakan *manipulator*. Fungsi *generateHTMWithDH* merupakan fungsi untuk mendapatkan *homogeneous transformation matrix end effector* terhadap *base frame* nya (H0-6) menggunakan DH *method*. Parameter yang diberikan yakni *steppJ1, steppJ2, steppJ3, steppJ4, yawJ5, rollJ6* merupakan nilai derajat dari *master arm*. Sedangkan *tampilMatriks* adalah fungsi untuk menampilkan matriks H0-6 pada serial monitor.

// Generate H0\_6 menggunakan DH parameter

void generateHTMWithDH (float T1, float T2, float T3, float T4, float T5, float T6) {

// Degree to radiant

T1 = (T1 / 180.00) \* PI;

T2 = (T2 / 180.00) \* PI;

T3 = (T3 / 180.00) \* PI;

T4 = (T4 / 180.00) \* PI;

T5 = (T5 / 180.00) \* PI;

T6 = (T6 / 180.00) \* PI;

Di dalam fungsi *generateHTMwithDH*, nilai derajat diubah terlebih dahulu ke dalam satuan radian. Ini dilakukan karena operasi matematika *sohcatoa* (sin, tan, cos) di dalam arduino tereksekusi menggunakan satuan radianbukan derajat.

// Parameter DH

float PT [6][4] {

{ T1, (-90.00 / 180.00)\*PI, r1, d1},

{ ((-90.00 / 180.00)\*PI) + T2, 0, r2, 0},

{ T3, (-90.00 / 180.00)\*PI, r3, 0},

{ T4, (90.00 / 180.00)\*PI, 0, d4},

{ T5, (-90.00 / 180.00)\*PI, 0, 0},

{ T6, 0, 0, d6}

};

Parameter DH yang telah dicari sebelumnya pun didefinisikan dalam *array* 2 dimensi yang terdiri 6 baris dan 4 kolom.

// Generate Homogeneous Transformation Matrix

int i = 0;

float H0\_1[4][4] = {

{cos(PT[i][0]), -sin(PT[i][0]) \* cos(PT[i][1]), sin(PT[i][0]) \* sin(PT[i][1]), PT[i][2] \* cos(PT[i][0])},

{sin(PT[i][0]), cos(PT[i][0]) \* cos(PT[i][1]), -cos(PT[i][0]) \* sin(PT[i][1]), PT[i][2] \* sin(PT[i][0])},

{0, sin(PT[i][1]), cos(PT[i][1]), PT[i][3]},

{0, 0, 0, 1}

};

int j = 1;

float H1\_2[4][4] = {

{cos(PT[j][0]), -sin(PT[j][0]) \* cos(PT[j][1]), sin(PT[j][0]) \* sin(PT[j][1]), PT[j][2] \* cos(PT[j][0])},

{sin(PT[j][0]), cos(PT[j][0]) \* cos(PT[j][1]), -cos(PT[j][0]) \* sin(PT[j][1]), PT[j][2] \* sin(PT[j][0])},

{0, sin(PT[j][1]), cos(PT[j][1]), PT[j][3]},

{0, 0, 0, 1}

};

int k = 2;

float H2\_3[4][4] = {

{cos(PT[k][0]), -sin(PT[k][0]) \* cos(PT[k][1]), sin(PT[k][0]) \* sin(PT[k][1]), PT[k][2] \* cos(PT[k][0])},

{sin(PT[k][0]), cos(PT[k][0]) \* cos(PT[k][1]), -cos(PT[k][0]) \* sin(PT[k][1]), PT[k][2] \* sin(PT[k][0])},

{0, sin(PT[k][1]), cos(PT[k][1]), PT[k][3]},

{0, 0, 0, 1}

};

int m = 3;

float H3\_4[4][4] = {

{cos(PT[m][0]), -sin(PT[m][0]) \* cos(PT[m][1]), sin(PT[m][0]) \* sin(PT[m][1]), PT[m][2] \* cos(PT[m][0])},

{sin(PT[m][0]), cos(PT[m][0]) \* cos(PT[m][1]), -cos(PT[m][0]) \* sin(PT[m][1]), PT[m][2] \* sin(PT[m][0])},

{0, sin(PT[m][1]), cos(PT[m][1]), PT[m][3]},

{0, 0, 0, 1}

};

int n = 4;

float H4\_5[4][4] = {

{cos(PT[n][0]), -sin(PT[n][0]) \* cos(PT[n][1]), sin(PT[n][0]) \* sin(PT[n][1]), PT[n][2] \* cos(PT[n][0])},

{sin(PT[n][0]), cos(PT[n][0]) \* cos(PT[n][1]), -cos(PT[n][0]) \* sin(PT[n][1]), PT[n][2] \* sin(PT[n][0])},

{0, sin(PT[n][1]), cos(PT[n][1]), PT[n][3]},

{0, 0, 0, 1}

};

int o = 5;

float H5\_6[4][4] = {

{cos(PT[o][0]), -sin(PT[o][0]) \* cos(PT[o][1]), sin(PT[o][0]) \* sin(PT[o][1]), PT[o][2] \* cos(PT[o][0])},

{sin(PT[o][0]), cos(PT[o][0]) \* cos(PT[o][1]), -cos(PT[o][0]) \* sin(PT[o][1]), PT[o][2] \* sin(PT[o][0])},

{0, sin(PT[o][1]), cos(PT[o][1]), PT[o][3]},

{0, 0, 0, 1}

};

Berikut merupakan pencarian nilai HTM di masing – masing *joint* yang berurutan pada *manipulator*. Nilai matriks yang didapat adalah H0-1, H1-2, H2-3, H3-4, H4-5, H5-6.

dotProductMatriks (1, H0\_1, H1\_2);

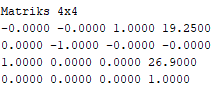
dotProductMatriks (2, H2\_3, H3\_4);

dotProductMatriks (3, H4\_5, H5\_6);

dotProductMatriks (4, H0\_2, H2\_4);

dotProductMatriks (5, H0\_4, H4\_6);

Nilai matriks H0-1, H1-2, H2-3, H3-4, H4-5, H5-6 harus di dot product kan satu persatu untuk mendapatkan matriks H0-6. Fungsi *dotProductMatriks* merupakan fungsi *user*-*defined* yang mengalikan 2 matriks secara *dot product*.

**Gambar 2. 35** Matriks *H0-6* yang Dihasilkan

Gambar 2.35 merupakan matriks H0-6 yang dihasilkan dan tampil di serial monitor. Dengan studi kasus semua *joint* bernilai 0o (kiri) dan ketika *joint* 1 bernilai 90o sedangkan yang lainnya bernilai 0o (kanan).

// calculation from HTM / Rotation Matrix

roll = atan2(H0\_6[2][0], H0\_6[2][1]);

pitch = acos(H0\_6[2][2]);

yaw = -atan2(H0\_6[0][2], H0\_6[1][2]);

// Ubah ke derajat

roll = roll/PI \* 180.00;

pitch = pitch/PI \* 180.00;

yaw = yaw/PI \* 180.00;

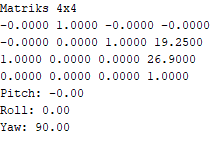
// Kembali ke posisi 0 (sesuai base frame)

pitch = pitch - 90.00;

roll = roll - 90.00;

yaw = yaw + 90.00;

Agar informasi rotasi dari *manipulator* terpahami dengan baik oleh operator, nilai kalkulasi *rotation matrix* dikonversi terlebih dahulu menjadi *yaw, pitch,* dan *roll angles* menggunakan program diatas.



**Gambar 2. 36** Nilai *Yaw, Pitch, Roll* yang Dihasilkan

Gambar 2.36 merupakan nilai *yaw, pitch, roll* yang dihasilkan dari *rotation matrix* dengan studi kasus ketika nilai *joint* 1 diputar 90o sedangkan *joint* lainnya bernilai 0o.

**Gambar 2. 37** Display LCD Menampilkan Parameter

Parameter masukkan derajat dan parameter posisi *end-effector* pun ditampilkan ke dalam display LCD 20x4 seperti pada Gambar 2.37. Tujuannya adalah untuk memudahkan operator menggerakan *manipulator* dan agar mengetahui dimana posisi *manipulator* berada.

* 1. Hasil Pengujian



**Gambar 2. 38** Visualisasi *Arm Manipulator*, *Master Controller*, dan *Motor Driver*

Setelah menyelesaikan pembuatan mekanik, elektrik, dan program dari robot (visualnya tertera pada Gambar 2.38), maka dilakukan pengujian untuk mengetahui keakurasian pergerakan lengan robot secara aktual dibandingkan dengan formulasi matematis telah dihitung.

1. Pengujian Posisi *End-Effector* Terhadap *Base Frame*

Ini merupakan pengujian posisi *end-effector manipulator* relatif terhadap *base frame*nya. Nilai θ1-θ6 akan menjadi masukkan, kemudian formulasi akan menghasilkan nilai posisi x,y,z. Nilai tersebut kemudian dibandingkan dengan hasil pengukuran langsung di *plant*.

Penjelasan parameter dalam tabel 2.2 adalah :

Sudut Masukkan : Merupakan sudut-sudut *setpoint* yang masukkan ke *manipulator* dari *joint* 1 (θ1) – *joint 6* (θ1).

Posisi Perhitungan : Merupakan hasil kalkulasi perhitungan posisi *manipulator* menggunakan *homogeneous transformation matrix* di sumbu x,y, dan z*.*

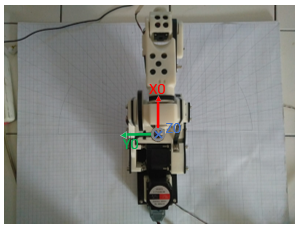
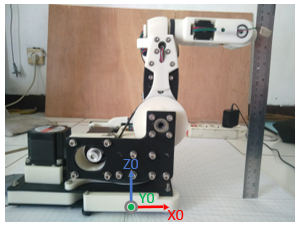
Posisi Aktual : Merupakan posisi *manipulator* yang terukur secara aktualdi sumbu x,y, dan z.

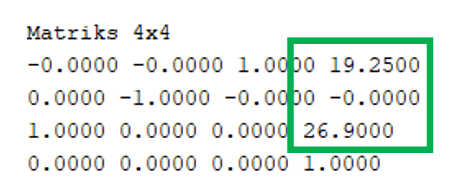
*Error* : Merupakan perbandingan kesalahan antara posisi aktual dan posisi perhitungan.

Berdasarkan data pengujian yang ditunjukkan pada tabel 2.2. Pengujian dilakukan secara berulang sebanyak 20 kali dengan nilai θ1-θ6yang variatif.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Sudut Masukkan** | | | | | | **Posisi Perhitungan** | | | **Posisi Aktual** | | | ***Error* (%)** | | |
| **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** | **X** | **Y** | **Z** | **X** | **Y** | **Z** | **X** | **Y** | **Z** |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 19,25 | 0 | 26,9 | 19,2 | 0 | 26,5 | 0,26 | 0,00 | 1,51 |
| 2 | 90 | 0 | 0 | 0 | 0 | 0 | 0 | -19,25 | 26,9 | 0 | -19,6 | 26,4 | 0,00 | 1,79 | 1,89 |
| 3 | -90 | 0 | 0 | 0 | 0 | 0 | 0 | 19,25 | 26,9 | 0 | 19,8 | 26,5 | 0,00 | 2,78 | 1,51 |
| 4 | 0 | 80 | 0 | 0 | 0 | 0 | 20,61 | 0 | 1,33 | 21,6 | 0 | 1,3 | 4,76 | 0,00 | 2,31 |
| 5 | 90 | 80 | 0 | 0 | 0 | 0 | 0 | 20,61 | 1,33 | 0 | 21 | 1 | 0,00 | 1,81 | 33,00 |
| 6 | 0 | 0 | -90 | 0 | 0 | 0 | 2,1 | 0 | 38,85 | 2,4 | 0 | 39,2 | 14,29 | 0,00 | 0,89 |
| 7 | -90 | 80 | -90 | 0 | 0 | 0 | 0 | -29,41 | 20,02 | 0 | -31 | 19,4 | 0,00 | 5,13 | 3,20 |
| 8 | 0 | -30 | 0 | 0 | 0 | 0 | 10,5 | 0 | 32,352 | 10,1 | 0 | 32,3 | 3,81 | 0,00 | 0,16 |
| 9 | 0 | 45 | -45 | 0 | 0 | 0 | 27,02 | 0 | 23,678 | 26,5 | 0 | 23,3 | 1,95 | 0,00 | 1,62 |
| 10 | 0 | 0 | 45 | 0 | 0 | 0 | 16,82 | 0 | 15,85 | 16,4 | 0 | 15,5 | 2,50 | 0,00 | 2,26 |
| 11 | 0 | 45 | -30 | 90 | 0 | 0 | 27,2 | 0 | 19,823 | 26,7 | 0 | 19,8 | 1,84 | 0,00 | 0,12 |
| 12 | 0 | 45 | -30 | 0 | 90 | 0 | 23,77 | 0 | 17,843 | 23,5 | 0 | 17,5 | 1,16 | 0,00 | 1,96 |
| 13 | 0 | 45 | -30 | 0 | -90 | 0 | 25,22 | 0 | 23,253 | 24,6 | 0 | 23,7 | 2,48 | 0,00 | 1,89 |
| 14 | 0 | 45 | -30 | 90 | 90 | 0 | 24,5 | 2,8 | 20,54 | 24,2 | 2,8 | 20,3 | 1,22 | 0,00 | 1,18 |
| 15 | 0 | 45 | -30 | -90 | 90 | 0 | 24,5 | -2,8 | 20,54 | 24,7 | -2,8 | 20,3 | 0,82 | 0,00 | 1,18 |
| 16 | 0 | 45 | -30 | 0 | 0 | 90 | 27,2 | 0 | 19,823 | 26,5 | 0 | 19,4 | 2,59 | 0,00 | 2,18 |
| 17 | 0 | 45 | -30 | 90 | 90 | 90 | 24,5 | 2,8 | 20,54 | 24,2 | 2,8 | 20,3 | 1,22 | 0,00 | 1,18 |
| 18 | 90 | 45 | -30 | 90 | 90 | 90 | -2,8 | 24,5 | 20,54 | -2,8 | 24,7 | 19,8 | 0,00 | 0,81 | 3,74 |
| 19 | -90 | 45 | -30 | 90 | 90 | 90 | 2,8 | -24,5 | 20,54 | 2,8 | -25,1 | 20 | 0,00 | 2,39 | 2,70 |
| 20 | -90 | 45 | -30 | -90 | -90 | -90 | 2,8 | -24,5 | 20,54 | 2,8 | -25,1 | 20 | 0,00 | 2,39 | 2,70 |
| **Rata-Rata :** | | | | | | | | | | | | | **1,94** | **0,85** | **3,36** |

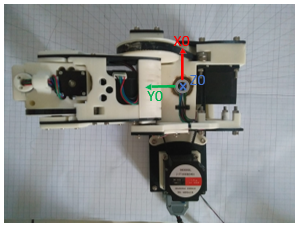
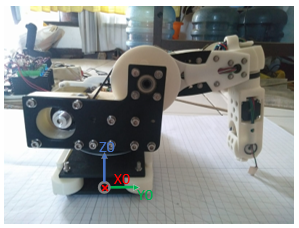
**Tabel 2. 2** Pengujian Posisi *End-Effector*

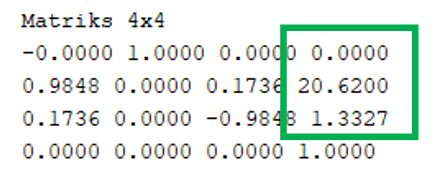
 



**Gambar 2. 39** Pengujian Posisi Sampel Ke-1

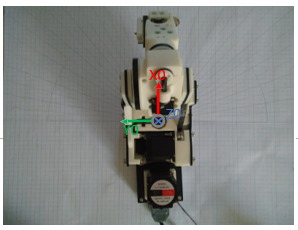
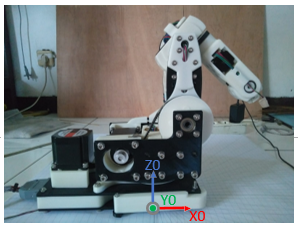
Gambar 2.39 merupakan pengujian posisi sampel pertama. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 0, θ3 = 0, θ4 = 0, θ5 = 0, θ6 = 0. Dan menghasilkan posisi 19.25 cm di sumbu X, 0 cm di sumbu Y dan 26.90 cm di sumbu Z.

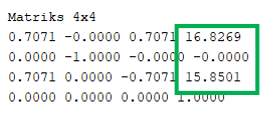
 



**Gambar 2. 40** Pengujian Posisi Sampel ke-5

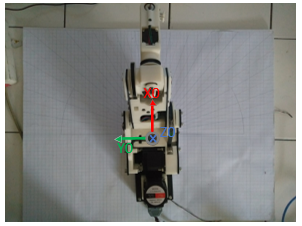
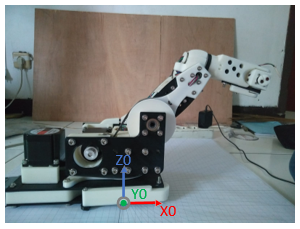
Gambar 2.40 merupakan pengujian posisi sampel kelima. Parameter *theta* yang diberikan adalah θ1 = 90, θ2 = 80, θ3 = 0, θ4 = 0, θ5 = 0, θ6 = 0. Dan menghasilkan posisi 0 cm di sumbu X, 20.62 cm di sumbu Y dan 1.3327 cm di sumbu Z.

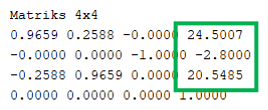
 



**Gambar 2. 41** Pengujian Posisi Sampel ke-10

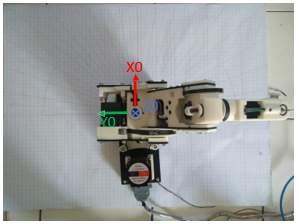
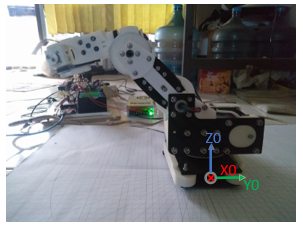
Gambar 2.41 merupakan pengujian posisi sampel kesepuluh. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 0, θ3 = 45, θ4 = 0, θ5 = 0, θ6 = 0. Dan menghasilkan posisi 16.82 cm di sumbu X, 0 cm di sumbu Y dan 15.85 cm di sumbu Z.

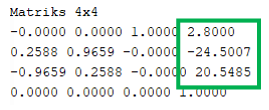
 



**Gambar 2. 42** Pengujian Posisi Sampel ke-15

Gambar 2.42 merupakan pengujian posisi sampel kelima belas. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 45, θ3 = -30, θ4 = -90, θ5 = 90, θ6 = 0. Dan menghasilkan posisi 24.50 cm di sumbu X, -2.8 cm di sumbu Y dan 20.54 cm di sumbu Z.



**Gambar 2. 43** Pengujian Posisi Sampel Ke-20

Gambar 2.43 merupakan pengujian posisi sampel kelima belas. Parameter *theta* yang diberikan adalah θ1 = -90, θ2 = 45, θ3 = -30, θ4 = -90, θ5 = -90, θ6 = -90. Dan menghasilkan posisi 2.80 cm di sumbu X, -24.50 cm di sumbu Y dan 20.54 cm di sumbu Z.

Analisa :

Setelah dilakukan pengujian, *error* dariposisi *end-effector* yang didapatkan relatif kecil. Jika dirata-ratakan, *error* yang terjadi di sumbu x sebesar 1.94%, 0.85% pada sumbu y, dan 3.36 % pada sumbu z. Ini mengindikasikan bahwa *denavit-hartenberg method* dapat di implementasikan pada pergerakan *forward kinematics* *manipulator* 6 DOF. Selain itu, *stick diagram* dan DH *parameter* yang penulis dapatkan secara analitis sudah tepat. *Error* yang terjadi mungkin diakibatkan dari faktor mekanik lengan robot dan pengukuran yang kurang presisi.

1. Pengujian Rotasi *End-Effector* Terhadap *Base Frame*

Ini merupakan pengujian rotasi e*nd-effector manipulator* berdasarkan *base frame*nya. Untuk memudahkan pengukuran, hanya nilai sudut 0o dan kelipatan 90o (90, 180, 270, dst) yang akan menjadi masukan di tiap *joint*. Namun sudut – sudut ini sudah merepresentasikan kebenaran rotasi dari *end-effector.*

Penjelasan parameter di dalam tabel 2.3 adalah :

Sudut Masukkan : Merupakan sudut-sudut *setpoint* sebagai masukkan ke *manipulator* dari *joint* 1 (θ1) – *joint* 6(θ1).

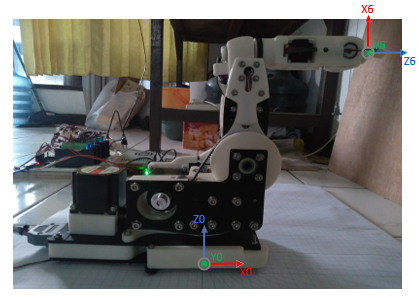
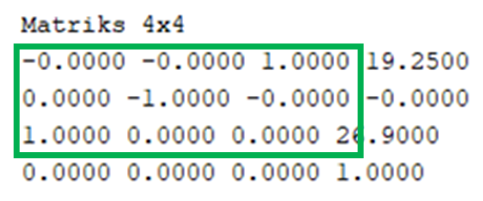
Proyeksi Rotasi Hitung : Merupakan proyeksi rotasi sumbu 6 (*end effector)* terhadap sumbu 0 (*base frame*) dalam sumbu XYZ. Proyeksi X6 terhadap X0, Y0, dan Z0. Proyeksi Y6 terhadap X0, Y0, dan Z0. Proyeksi Z6 terhadap X0, Y0, dan Z0.

Proyeksi Rotasi Ukur : Merupakan proyeksi rotasi *end-effector* terhadap *base frame* dalam sumbu XYZ yang diamati secara visual.

Ket : Merupakan keterangan validitas rotasi ukur dibandingkan dengan rotasi hitung.

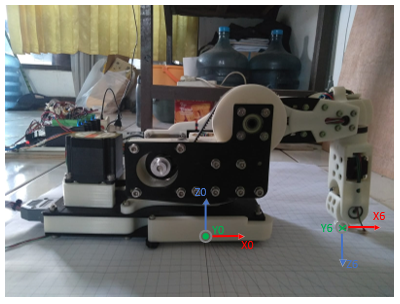
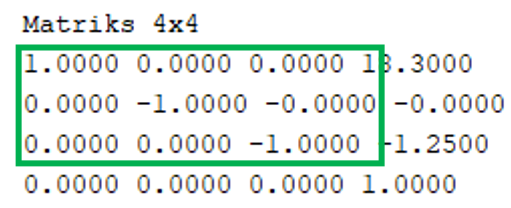
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Sudut Masukkan** | | | | | | **Proyeksi Rotasi Hitung** | | | | **Proyeksi Rotasi Ukur** | | | | **Ket** |
| 1 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 0 | 0 | 0 | 0 | 0 | **X0** | 0 | 0 | 1 | **X0** | 0 | 0 | 1 |
| **Y0** | 0 | -1 | 0 | **Y0** | 0 | -1 | 0 |
| **Z0** | 1 | 0 | 0 | **Z0** | 1 | 0 | 0 |
| 2 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 90 | 0 | 0 | 0 | 0 | 0 | **X0** | 0 | 1 | 0 | **X0** | 0 | 1 | 0 |
| **Y0** | 0 | 0 | 1 | **Y0** | 0 | 0 | 1 |
| **Z0** | 1 | 0 | 0 | **Z0** | 1 | 0 | 0 |
| 3 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 90 | 0 | 0 | 0 | 0 | **X0** | 1 | 0 | 0 | **X0** | 1 | 0 | 0 |
| **Y0** | 0 | -1 | 0 | **Y0** | 0 | -1 | 0 |
| **Z0** | 0 | 0 | -1 | **Z0** | 0 | 0 | -1 |
| 4 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 0 | 0 | 90 | 0 | 0 | **X0** | 0 | 0 | 1 | **X0** | 0 | 0 | 1 |
| **Y0** | -1 | 0 | 0 | **Y0** | -1 | 0 | 0 |
| **Z0** | 0 | -1 | 0 | **Z0** | 0 | -1 | 0 |
| 5 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 0 | 0 | 0 | 90 | 0 | **X0** | 1 | 0 | 0 | **X0** | 1 | 0 | 0 |
| **Y0** | 0 | -1 | 0 | **Y0** | 0 | -1 | 0 |
| **Z0** | 0 | 0 | -1 | **Z0** | 0 | 0 | -1 |
| 6 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 0 | 0 | 0 | 0 | 90 | **X0** | 0 | 0 | 1 | **X0** | 0 | 0 | 1 |
| **Y0** | -1 | 0 | 0 | **Y0** | -1 | 0 | 0 |
| **Z0** | 0 | -1 | 0 | **Z0** | 0 | -1 | 0 |
| 7 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 90 | 90 | -90 | 0 | 0 | 0 | **X0** | 0 | 1 | 0 | **X0** | 0 | 1 | 0 |
| **Y0** | 0 | 0 | 1 | **Y0** | 0 | 0 | 1 |
| **Z0** | 1 | 0 | 0 | **Z0** | 1 | 0 | 0 |
| 8 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 90 | -90 | 180 | 0 | 0 | **X0** | 0 | 0 | 1 | **X0** | 0 | 0 | 1 |
| **Y0** | 0 | 1 | 0 | **Y0** | 0 | 1 | 0 |
| **Z0** | -1 | 0 | 0 | **Z0** | -1 | 0 | 0 |
| 9 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 90 | -90 | 180 | 90 | 0 | **X0** | 1 | 0 | 0 | **X0** | 1 | 0 | 0 |
| **Y0** | 0 | 1 | 0 | **Y0** | 0 | 1 | 0 |
| **Z0** | 0 | 0 | 1 | **Z0** | 0 | 0 | 1 |
| 10 | **θ1** | **θ2** | **θ3** | **θ4** | **θ5** | **θ6** |  | **X6** | **Y6** | **Z6** |  | **X6** | **Y6** | **Z6** | **Valid** |
| 0 | 90 | -90 | 180 | 90 | 180 | **X0** | -1 | 0 | 0 | **X0** | -1 | 0 | 0 |
| **Y0** | 0 | -1 | 0 | **Y0** | 0 | -1 | 0 |
| **Z0** | 0 | 0 | 1 | **Z0** | 0 | 0 | 1 |

**Tabel 2. 3** Pengujian Rotasi *End-Effector*

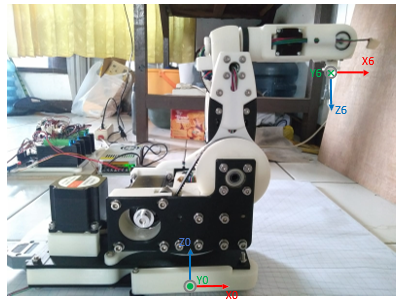
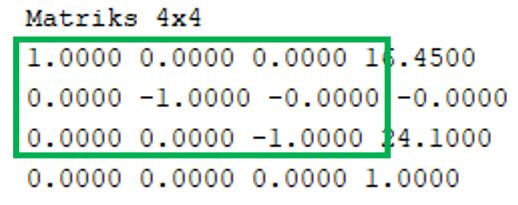
**Gambar 2. 44** Pengujian Rotasi Sampel Ke-1

Gambar 2.44 merupakan pengujian rotasi sampel kesatu. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 0, θ3 = 0, θ4 = 0, θ5 = 0, θ6 = 0. Dan menghasilkan X6 yang sejajar dengan Z0 (X6Z0 = 1), Y6 yang berlawanan dengan Y0 (Y6Y0 = -1), dan Z6 yang sejajar dengan X0 (Z6X0 = 1).

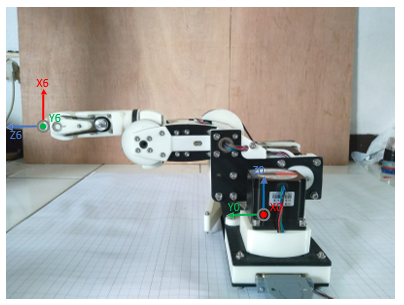
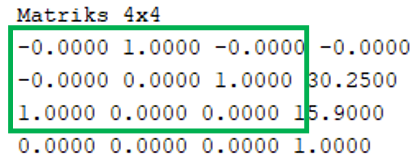
**Gambar 2. 45** Pengujian Rotasi Sampel Ke-3

Gambar 2.45 merupakan pengujian rotasi sampel ketiga. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 90, θ3 = 0, θ4 = 0, θ5 = 0, θ6 = 0. Dan menghasilkan X6 yang sejajar dengan X0 (X6X0 = 1), Y6 yang berlawanan dengan Y0 (Y6Y0 = -1), dan Z6 yang berlawanan dengan Z0 (Z6Z0 = -1).

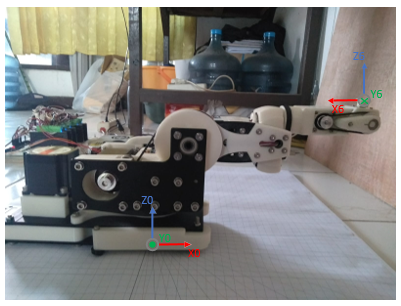
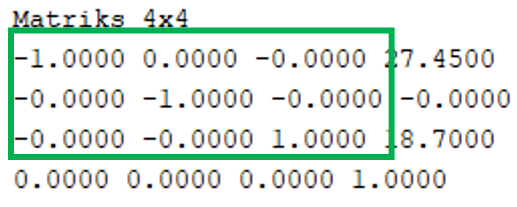
**Gambar 2. 46** Pengujian Rotasi Sampel Ke-5

Gambar 2.46 merupakan pengujian rotasi sampel kelima. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 0, θ3 = 0, θ4 = 0, θ5 = 90, θ6 = 0. Dan menghasilkan X6 yang sejajar dengan X0 (X6X0 = 1), Y6 yang berlawanan dengan Y0 (Y6Y0 = -1), dan Z6 yang berlawanan dengan Z0 (Z6Z0 = -1).

**Gambar 2. 47** Pengujian Rotasi Sampel Ke-7

Gambar 2.47 merupakan pengujian rotasi sampel ketujuh. Parameter *theta* yang diberikan adalah θ1 = 90, θ2 = 90, θ3 = -90, θ4 = 0, θ5 = 0, θ6 = 0. Dan menghasilkan X6 yang sejajar dengan Z0 (X6Z0 = 1), Y6 yang sejajar dengan X0 (Y6X0 = 1), dan Z6 yang sejajar dengan Y0 (Z6Y0 = 1).

**Gambar 2. 48** Pengujian Rotasi Sampel Ke-10

Gambar 2.48 merupakan pengujian rotasi sampel kesepuluh. Parameter *theta* yang diberikan adalah θ1 = 0, θ2 = 90, θ3 = -90, θ4 = 180, θ5 = 90, θ6 = 180. Dan menghasilkan X6 yang berlawanan dengan X0 (X6X0 = -1), Y6 yang berlawanan dengan Y0 (Y6Y0 = -1), dan Z6 yang sejajar dengan Z0 (Z6Z0 = 1).

Analisa :

Dalam pengujian ini, dilakukan pengamatan bagaimana perputaran sumbu *end-effector* terhadap *base frame* dari *manipulator*. Dikarenakan hasil perhitungan berupa *rotation matrix* yang akan sukar diukur secara aktual. Maka pengamatan dilakukan secara visual dan menggunakan sudut-sudut yang mudah untuk diamati seperti 0o, 90o, 180o. Maka hasil yang didapatkan adalah kevalidan 100% antara perputaran *end-effector* terhadap *base frame* nya yang beberapa sampelnya bisa diamati pada gambar gambar diatas.

# BAB III

**KESIMPULAN DAN SARAN**

1. Kesimpulan Hasil Implementasi

Berdasarkan pembahasan dan pengujian mengenai “Pencarian *End-Effector* Pada *Arm Manipulator 6 DOF* Dengan Metode *Denavit-Hartenberg* dan Pengendali *Master Controller*” maka dapat disimpulkan bahwa :

1. Dengan menggunakan *Denavit-Hartenberg Method* ini, kalkulasi *forward kinematics* untuk menggerakan *manipulator 6 DOF* dapat disederhanakan sehingga nilai posisi dan rotasi dari *end-effector* dapat diketahui dengan relatif mudah. Matriks homogen yang dihasilkan menggunakan *DH method* ini persis sama dengan kalkulasi yang dihasilkan melalui pendekatan secara manual.
2. Untuk dapat mengaplikasikan *Denavit-Hartenberg Method* pada lengan robot, diperlukan pengetahuan dalam menggambar *kinematics diagram* yang benar sesuai dengan 4 aturan *Denavit-Hartenberg*. Kemudian diperlukan juga untuk mengetahui definisi dasar dari *DH parameters* yakni θ, α, r, dan d untuk mengisi *DH parameters table*.
3. Robot dapat mengetahui posisi *end-effector* nyaterhadap *base frame*. Besar kesalahan posisi pada sumbu X sebesar 1,94%, pada sumbu Y sebesar 0,85%, dan pada sumbu Y sebesar 3,36 %. Robot pun dapat mengetahui rotasi dari *end-effector* nya dengan validitas 100 %.
4. Saran

Berdasarkan pembahasan dan pengujian mengenai “Pencarian *End-Effector* Pada *Arm Manipulator 6 DOF* Dengan Metode *Denavit-Hartenberg* dan Pengendali *Master Controller*”, maka ada beberapa saran diantaranya sebagai berikut :

1. Menggunakan *mikrokontroler* atau *mini computer* dengan *processor* yang lebih tinggi meminimalkan latensi pengiriman data.
2. Menggunakan *belt* yang *close loop* secara pabrikan sebagai transmisi putaran agar tidak mudah putus.
3. Menggunakan jenis motor torsi daya angkatnya lebih besar
4. Menggunakan material yang lebih kuat seperti *acrylic* atau *steel* agar konstruksi mekanik nya kokoh.
5. Menggunakan sensor seperti *limit switch* pada setiap *joint manipulator* sebagai titik *home* dari masing – masing *joint*. Penyebabnya adalah karena motor stepper tidak memiliki karakteristik untuk mengetahui posisi sebelumnya apabila sumber dayanya terputus.
6. Walaupun pergerakan motor stepper sangat presisi, faktor beban dan lingkungan dapat mempengaruhi pergerakannya sehingga dapat menimbulkan selip. Selip ini dapat membuat perhitungan posisi dari stepper menjadi tidak tepat. Maka dari itu dibutuhkan pula sensor pada masing – masing *joint* sebagai *feedback* kembali ke sistem.

# DAFTAR PUSTAKA

1. Dipankar Mitra, “*How to Improve Motion Smoothness and Accuracy of Stepper Motor*”, *Texas Instruments*.
2. Agus B.D, Lina, “Penerapan Metode *Denavit-Hartenberg* Pada Perhitungan *Inverse Kinematics* Gerakan Lengan Robot”, *Jurnal Muara Sains Teknologi, Kedokteran, dan Ilmu Kesehatam Vol.1, No.1, April 2017: hlm 287-292.*
3. Robert D. Christ dan Robert L. Wernli Sr. *The ROV Manual (A User Guide for Remotely Operated Vehicle Second Edition*). Amerika: Elsevier Ltd. 2014.
4. Anandhya Tradana, “Rancang Bangun Aplikasi Python Dengan Raspberry Untuk Kendali Robot Lengan,” *Proyek Akhir Diploma*, Politeknik Manufaktur Bandung, 2020.
5. Tazkia Rusdiana, “Pengendalian *End Effector* Robot Lengan Untuk Aplikasi *Pick And Place* Benda Posisi Acak Dengan Metode Pengolahan Citra Digital”, *Proyek Akhir Diploma*, Politeknik Manufaktur Bandung, 2021.
6. 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.
7. Daniel. C.S, “Perancangan Jaringan Saraf Tiruan Untuk Menyelesaikan Kinematika Balik Manipulator Robot Denso 6-DOF”. Tugas Akhir, Institut Teknologi Sepuluh November, 2018.
8. “Schilling Robotics”. Titan 3 Manipulator System, Costeau Ct Amerika Serikat
9. (2020) *6 Axis Robotic Arm*. [Online]. Diakses pada 8 Agustus 2022, Tersedia: https://www.youtube.com/playlist?list=PLh-o7Bm9fcVy2MhcVkaGOmANr8j7dZ78-
10. (2018) *Robotics* 1 2017.[Online]. Diakses pada 8 Agustus 2022, Tersedia: <https://www.youtube.com/playlist?list=PLT_0lwItn0sDBE98BsbaZezflB96ws12b>
11. *Rotation formalism in three dimensions* [Online].Diakses pada 8 Agustus 2022, Tersedia:

https://en.wikipedia.org/wiki/Rotation\_formalisms\_in\_three\_dimensions#Conversion\_formulae\_between\_formalisms

# LAMPIRAN

Semua dokumentasi mekanik, elektrik dan program dihimpunkan ke dalam folder github yang berada pada tautan : <https://github.com/FachriMFulchan/Arm_Manipulator_7_Functions>