

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Pada saat ini, perkembangan teknologi sudah sangat pesat. Perkembangannya telah mencakup di segala bidang, terutama pada bidang teknologi komputer, elektronika, dan kontrol. Salah satu contoh perkembangan pada bidang komputer dan bidang elektronika kontrol adalah penggunaan robot. Penggunaan robot kini semakin meningkat dari waktu ke waktu untuk menangani berbagai tugas. Baik tugas yang tidak bisa ditangani manusia ataupun tugas-tugas yang telah umum dilakukan oleh manusia (Hamidah, 2008).

Di sisi lainnya, perkembangan yang juga terjadi di bidang industri menuntut penggunaan alat bantu yang dapat mengoptimalkan sumber-sumber daya yang ada agar dapat bersaing di pasar bebas. Robot merupakan salah satu alat bantu yang dalam kondisi tertentu sangat diperlukan dalam industri. Terdapat kondisi-kondisi tertentu dalam industri yang tidak mungkin ditangani oleh manusia seperti kebutuhan akan akurasi yang tinggi, tenaga yang besar, kecepatan yang tinggi atau resiko yang tinggi. Keadaan-keadaan ini dapat diatasi dengan penggunaan robot (Masykuri, 2010).

Robot memiliki banyak kelebihan yang tidak dimiliki manusia. Kelebihan robot diantaranya 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. Selain itu robot juga lebih sedikit melakukan kesalahan dibandingkan manusia dan kinerjanya lebih mudah untuk dimonitor. Penggunaan robot dapat menghemat biaya produksi dan dapat memberikan berbagai keuntungan lainnya (Widayanti, 2009).

Diantara robot yang sering digunakan dalam dunia industri adalah robot lengan. Robot lengan merupakan robot yang meniru gerak lengan pada manusia sehingga mampu melakukan pemindahan barang dari satu tempat ke tempat lain. Robot lengan diharapkan dapat diprogram ulang secara fleksibel oleh pengguna. Pemrograman ulang pada robot lengan membuat kita membutuhkan antarmuka

antara robot lengan dengan pengguna melalui komputer. Dengan adanya antarmuka tersebut diharapkan robot lengan akan lebih mudah diprogram ulang sesuai kebutuhan ( Seif, 2012 ).

Pada skripsi ini, digunakan robot lengan *Smart Arm Robotic AX-12A*. Robot ini memiliki empat derajat kebebasan dan tujuh buah servo sebagai aktuator. Aktuator yang dimaksud yaitu 1 servo untuk *pivot*, 2 servo untuk lengan, 2 servo untuk siku, 1 servo pergelangan dan 1 servo sebagai penjapit. Robot ini merupakan salah satu peralatan percobaan yang digunakan pada Laboratorium Robotika dan Mekatronika Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Sedangkan untuk perhitungan kinematika digunakan metode *invers* dan *forward*. Penggunaan *invers* dan *forward* pada *Geometry - Based Kinematic* adalah untuk memberikan informasi posisi *end effector* melalui pendekatan trigonometri sehingga dapat memudahkan pengguna. *Invers* kinematik digunakan untuk mencari sudut pada tiap *joint* dengan memanfaatkan posisi yang diberikan pada *end effector*. Sedangkan *forward* kinematik digunakan untuk mencari posisi *end effector* dengan memanfaatkan sudut yang diperoleh dari nilai yang diberikan pada masing-masing *joint*.

Mahasiswa selama ini merasa kesulitan memahami dinamika gerak robot lengan, hal ini disebabkan pemahaman kinematika gerak pada robot lengan membutuhkan pengetahuan penunjang seperti konsep matrik, vektor dan lain-lain. Oleh karena itu, antarmuka robot lengan dengan pengguna yang baik nantinya diharapkan dapat digunakan sebagai modul pembelajaran mata kuliah robotika.

## 1.2 Rumusan Masalah

Rumusan masalah dari latar belakang yang telah dipaparkan adalah sebagai berikut:

1. Bagaimana merancang pengendalian tujuh motor servo untuk mengubah dan menyesuaikan posisi *end-effector*?
2. Bagaimana merancang sistem antarmuka Qt untuk komunikasi antara robot lengan (Arduino) dengan komputer?

3. Bagaimana menerapkan metode *invers kinematic* dan *forward kinematic* pada gerakan robot untuk memindahkan benda dari satu tempat ke tempat yang berbeda pada posisi yang kita inginkan?

### 1.3 Batasan Masalah

Dalam penulisan skripsi ini terdapat batasan masalah sebagai berikut:

1. Robot lengan yang digunakan adalah *Smart Arm Robotic AX-12A* dengan 4 derajat kebebasan ditambah dengan sebuah *gripper* serta kontroler yang digunakan yaitu *Arduino Mega*.
2. Aktuator yang digunakan untuk mengubah posisi lengan menggunakan motor servo *Dynamixel AX-12A*.
3. Program antarmuka yang digunakan adalah *Qt*.
4. Berat benda yang digunakan cukup ringan dan berbentuk bola sehingga tidak memerlukan orientasi pada *end effector*.
5. Pembahasan ditekankan pada kinematika gerak robot.
6. Lintasan gerakan *end-effector* robot lengan ditentukan oleh penulis.
7. Pembahasan mengenai dinamika robot, mekanik robot serta rangkaian elektronika tidak dibahas secara mendalam.

### 1.4 Tujuan

1. Merancang sistem kendali untuk pergerakan robot lengan *Smart Arm Robotic AX-12A* melalui pendekatan *Geometry - Based Kinematic* dimana robot lengan nantinya akan mengambil sebuah benda untuk dipindahkan dengan memperhatikan posisi objek.
2. Merancang antarmuka sistem kendali pada robot lengan menggunakan *Arduino Mega* dan *Qt*.

### 1.5 Sistematika Pembahasan

Skripsi ini terdiri atas enam bab dengan sistematika pembahasan sebagai berikut:

- BAB I           Pendahuluan**  
Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.
- BAB II           Tinjauan Pustaka**  
Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan sistem.
- BAB III          Metode Penelitian**  
Membahas tentang metode yang digunakan dalam penyusunan skripsi ini.
- BAB IV          Perancangan dan Pembuatan Alat**  
Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.
- BAB V          Pengujian dan Analisis**  
Memuat hasil pengujian menggunakan metode *invers kinematic* dan *forward kinematic* terhadap ketepatan posisi dari robot lengan (*end-effector*).
- BAB VI          Penutup**  
Memuat kesimpulan dan saran-saran untuk pengembangan lebih lanjut dari sistem yang telah dibuat.

## BAB II

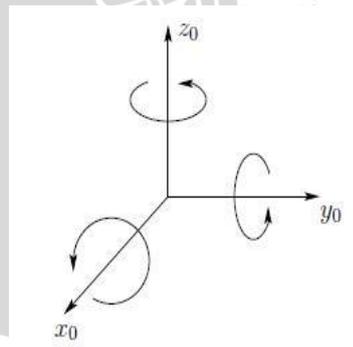
### TINJAUAN PUSTAKA

#### 2.1 Kinematika Robot Lengan

Saat ini, ilmu robotika dan implementasinya di berbagai bidang berkembang dengan pesat. Robotika tidak lagi hanya diminati oleh kalangan peneliti di bidang robotika, namun juga diminati oleh peneliti dan siswa dari berbagai disiplin ilmu, serta kalangan penggemar robotika. Salah satu dasar dari ilmu robotika adalah pemahaman mengenai kinematika dan dinamika robot. Dinamika robot merupakan pengetahuan atau teori tentang pergerakan objek dengan memperhitungkan gaya-gaya yang dapat menyebabkan objek bergerak, contohnya torsi pada aktuator, percepatan yang dialami oleh benda pejal dan distribusi massa. Sedangkan kinematika robot merupakan pengetahuan atau teori tentang pergerakan objek tanpa memperhitungkan gaya-gaya yang menyebabkan benda itu bergerak. Kinematika robot terdiri atas rotasi dan translasi.

##### 2.1.1 Rotasi

Rotasi yaitu gerakan berputar pada sebuah sumbu yang tetap, gerakan tersebut dapat berputar pada sumbu  $x$ ,  $y$  maupun  $z$ . seperti yang terlihat dalam Gambar 2.1.



**Gambar 2.1 Rotasi pada Tiap-tiap Sumbu Koordinat Kartesian (Spong, Mark W, 2007)**

$Z_0$  adalah sumbu utama rotasi pada arah  $Z$ ,  $Y_0$  adalah sumbu utama rotasi pada arah  $Y$ ,  $X_0$  adalah sumbu utama rotasi pada arah  $X$

Matrik transformasi rotasi pada tiap-tiap sumbu terlihat dalam persamaan dibawah ini (Persamaan 2.1, 2.2 dan 2.3),  $\theta$  untuk sumbu z,  $\alpha$  untuk sumbu x dan  $\phi$  untuk sumbu y.

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots(2.1)$$

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \dots\dots(2.2)$$

$$R_{y,\phi} = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \dots\dots(2.3)$$

Dimana

$R_{z,\theta}$  = Matriks transformasi rotasi pada sumbu z sepanjang  $\theta$

$R_{x,\alpha}$  = Matriks transformasi rotasi pada sumbu x sepanjang  $\alpha$

$R_{y,\phi}$  = Matriks transformasi rotasi pada sumbu y sepanjang  $\phi$

Jadi untuk merepresentasikan sebuah posisi ( $p_1$ ) dari sumbu koordinat yang telah berotasi ke dalam posisi ( $p_0$ ) sumbu koordinat semula dirumuskan dalam Persamaan 2.4 sampai dengan 2.6.

$$p_0 = \begin{bmatrix} p_{0x} \\ p_{0y} \\ p_{0z} \end{bmatrix} \dots\dots(2.4)$$

$$p_1 = \begin{bmatrix} p_{1x} \\ p_{1y} \\ p_{1z} \end{bmatrix} \dots\dots(2.5)$$

$$p_0 = R_0^1 p_1 \dots\dots(2.6)$$

Dimana :

$p_0$  = posisi sumbu koordinat semula.

$p_{0x}$  = posisi sumbu koordinat x semula.

$p_{0y}$  = posisi sumbu koordinat y semula.

$p_{0z}$  = posisi sumbu koordinat z semula.

$p_{1x}$  = posisi sumbu koordinat x yang telah mengalami translasi pada jarak tertentu dari sumbu koordinat semula.

$p_{1y}$  = posisi sumbu koordinat y yang telah mengalami translasi pada jarak tertentu dari sumbu koordinat semula.

$p_{1z}$  = posisi sumbu koordinat z yang telah mengalami translasi pada jarak tertentu dari sumbu koordinat semula.

$R_0^1$  = sumbu koordinat semula yang telah mengalami rotasi dengan sudut tertentu.

$d_0^1$  = sumbu koordinat semula yang telah mengalami pergeseran sejauh jarak tertentu.

$H$  = matrik transformasi homogen yang terdiri dari gerakan rotasi dan translasi.

### 2.1.2 Translasi

Sedangkan pada translasi, artinya terdapat pergeseran sumbu koordinat pada jarak tertentu dari sumbu koordinat semula. Jadi untuk merepresentasikan posisi ( $p_1$ ) dari sumbu koordinat translasi kedalam sumbu koordinat semula dirumuskan dalam Persamaan 2.7 dan 2.8.

$$p_0 = p_1 + d_0^1 \quad \dots(2.7)$$

$$\begin{bmatrix} p_{0x} \\ p_{0y} \\ p_{0z} \end{bmatrix} = \begin{bmatrix} p_{1x} \\ p_{1y} \\ p_{1z} \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \quad \dots(2.8)$$

Dimana :

$d_x$  = translasi sejauh jarak tertentu pada sumbu x

$d_y$  = translasi sejauh jarak tertentu pada sumbu y

$d_z$  = translasi sejauh jarak tertentu pada sumbu z

Apabila terjadi dua gerakan yaitu rotasi dan translasi secara bersamaan, artinya sumbu koordinat semula diputar (rotasi) dengan sudut tertentu ( $R_0^1$ ) kemudian digeser (translasi) sejauh jarak tertentu ( $d_0^1$ ). Dapat diselesaikan dengan

menggabungkan persamaan rotasi dan translasi (Persamaan 2.6 dan 2.7) seperti dalam persamaan 2.9.

$$p_0 = R_0^1 p_1 + d_0^1 \quad \dots(2.9)$$

Persamaan 2.9 dapat kita representasikan kedalam bentuk matrik agar lebih mudah perhitungan dan sederhana penulisannya.

### 2.1.3 Denavit -Hartenberg (D-H) Parameter

Representasi bentuk matrik H yang terdiri dari matrik rotasi dan translasi, kita sebut sebagai *homogenous transformation*. *Homogenous transformation* dapat memudahkan kita untuk mendapat posisi suatu titik yang diperlihatkan dari koordinat *frame i* kedalam bentuk koordinat *frame j* yang telah dilakukan gerakan translasi dan rotasi.

$$p_0 = R_0^1 p_1 + d_0^1$$

$$\begin{bmatrix} p_0 \\ 1 \end{bmatrix} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad \dots(2.10)$$

Pada umumnya untuk menentukan *frame* atau koordinat memiliki aturan-aturan tertentu agar nantinya memudahkan kita dalam menganalisa posisi dari pergerakan robot tersebut, Diantaranya adalah *Denavit – Hartenberg parameter* Pada *D-H parameter*, suatu matrik *homogenous transformation*  $A_i$  merepresentasikan hasil dari empat transformasi dasar. Secara matematis ditunjukkan dalam Persamaan 2.11 sampai 2.14 berikut ini.

$$A_i = Rot_{z_i, \theta_i} Trans_{z_i, d_i} Trans_{x_i, a_i} Rot_{x_i, \alpha_i} \quad \dots(2.11)$$

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots(2.12)$$

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \cos \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & \cos \theta_i \cos \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots(2.13)$$

$$T_0^n = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots(2.14)$$

Dimana :

$Rot_{z\theta}$  = Matrik rotasi sumbu koordinat z semula yang telah mengalami rotasi dengan sudut  $\theta$  tertentu.

$Rot_{x\alpha}$  = Matrik rotasi sumbu koordinat semula yang telah mengalami rotasi dengan sudut  $\alpha$  tertentu.

$Trans_{zd}$  = Matrik translasi sumbu koordinat semula yang telah mengalami translasi sejauh jarak  $d$  tertentu.

$Trans_{xd}$  = Matrik translasi sumbu koordinat semula yang telah mengalami translasi sejauh jarak  $d$  tertentu.

$A$  = Matrik transformasi homogen yang terdiri atas empat transformasi dasar transformasi untuk menentukan frame / koordinat.

Pada *D-H parameter*, terdapat beberapa parameter yaitu  $a_i$ ,  $\alpha_i$ ,  $d_i$ , dan  $\theta_i$ .  $a_i$  kita sebut sebagai *length*,  $\alpha_i$  disebut sebagai *twist*,  $d_i$  disebut sebagai *offset*, dan  $\theta_i$  disebut sebagai *angle*.

### 2.1.4 Forward kinematic

Apabila kita menghendaki untuk menghitung posisi (dari koordinat dasar) robot lengan dengan memberikan nilai pada *D-H parameter*, maka hal itu disebut *forward kinematic*. *Forward kinematic* adalah suatu metode perhitungan posisi sebagai fungsi sudut dengan menghitung setiap *link*. Jika kita memiliki *n-link* maka kita bisa mencari matriks transformasi homogen melalui Persamaan 2.15.

$$T_0^n = A_1 \dots A_n \dots (2.15)$$

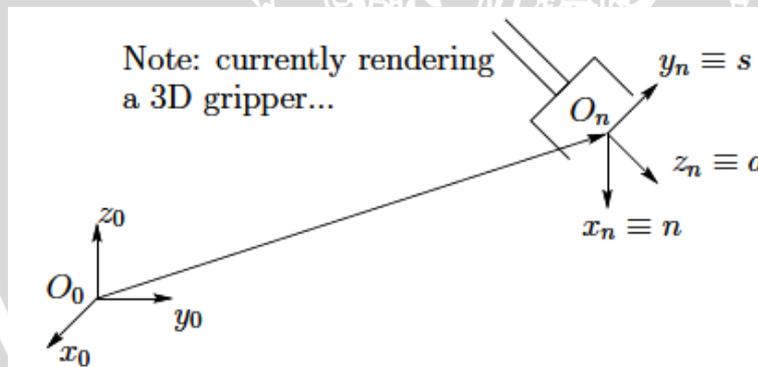
Matrik  $n = [n_x \ n_y \ n_z]^T$  adalah vektor yang merepresentasikan arah sumbu  $o_1x_1$  kedalam  $o_0x_0y_0z_0$ ,  $n$  adalah arah normal (keatas).

Matrik  $s = [s_x \ s_y \ s_z]^T$  adalah vektor yang merepresentasikan arah sumbu  $o_1y_1$  kedalam  $o_0x_0y_0z_0$ ,  $s$  adalah arah *sliding* (pergeseran).

Matrik  $a = [a_x \ a_y \ a_z]^T$  adalah vektor yang merepresentasikan arah sumbu  $o_1z_1$  kedalam  $o_0x_0y_0z_0$ ,  $a$  adalah *approach* (pendekatan).

Matrik  $p = [p_x \ p_y \ p_z]^T$  adalah vektor yang merepresentasikan dari sumbu asal  $o_0$  ke sumbu asal  $o_1$  direpresentasikan kedalam sumbu koordinat  $o_0x_0y_0z_0$

Dalam Gambar 2.2 berikut ini diperlihatkan gambaran umum penempatan *frame* robot pada bagian dasar dan sebuah *gripper* menggunakan *forward kinematic*.



Gambar 2.2 Gambaran Umum Penempatan Frame Pada *Forward Kinematic* (Spong, Mark W, 2007)

### 2.1.5 Invers Kinematic

Kebalikan dari metode *forward kinematic* adalah *invers kinematic*. Pada *forward kinematic* kita memberikan nilai pada setiap sudut motor servo dan mendapatkan posisi *end-effector*, sedangkan pada *invers kinematic* kita memberi masukan berupa posisi dan robot lengan akan mencari sudut pada tiap motor servo agar tepat sesuai dengan posisi yang kita inginkan. Solusi umum dari *invers kinematic* memang tidak ada, hal ini disebabkan solusinya sangat bergantung pada

peletakan *frame* (sumbu koordinat). Adapun pendekatan yang dilakukan untuk menentukan solusi sudut motor servo yaitu dengan pendekatan *Geometric Solution* dan *Algebraic Solution*.

Langkah awal untuk menentukan sudut pada tiap-tiap link adalah dengan melakukan *invers* pada matrik *forward kinematic*. Persamaan-persamaan yang dihasilkan dari *invers* matrik *forward kinematic* kemudian dicari solusinya dengan melakukan substitusi dan eliminasi berdasarkan dua pendekatan di atas. Atau jika ditulis secara matematis akan menghasilkan

$$A_1^{-1}T_0^n = A_1^{-1}A_1 \dots A_n \quad \dots(2.16)$$

sehingga:

$$A_{n-1}^{-1} \dots A_1^{-1}T_0^n = A_{n-1}^{-1} \dots A_1^{-1}A_1 \dots A_n \quad \dots(2.17)$$

## 2.2 Sistem Robot Lengan

Sistem robot lengan pada umumnya merupakan suatu struktur mekanik yang tersusun atas beberapa batang kaku yang sering disebut dengan *link*. *Link* terbuat dari logam, plastik, dan lain sebagainya. Antara *link* satu dengan *link* lainnya dihubungkan oleh persendian yang disebut *joint*. Umumnya *Prismatic Joint* dan *Flat Joint* dapat menghasilkan pergeseran. Sedangkan *Spheris Joint* dan *Revolute Joint* dapat menghasilkan *Degree of Freedom* (DOF) atau derajat kebebasan. *Degree of Freedom* (DOF) atau derajat kebebasan adalah jumlah arah yang independen dimana *actuator* dari sebuah robot dapat bergerak dan menghasilkan gerakan berputar. DOF dapat dihitung tiap sendi dan tidak termasuk *end effector*. Sedangkan *end effector* adalah piranti yang terpasang pada lengan robot untuk melaksanakan fungsi-fungsi tertentu. *End effector* terbagi menjadi dua yaitu *gripper* dan *tool*. *End effector* dan keseluruhan bagian robot lengan bekerja pada *workspace* tertentu, tergantung kemampuan robot yang digunakan

Pada robot *Smart Arm Robotic AX-12A* memiliki 4 derajat kebebasan ditambah sebuah *gripper*. Bagian-bagian pada robot lengan diantaranya yaitu batang-batang logam aluminium yang berfungsi sebagai link, 6 buah motor DC servo yang berfungsi sebagai aktuator dan sebuah servo pada bagian *gripper* yang

berfungsi sebagai *end effector*. Pengendalian lengan dibantu dengan suatu sistem mikrokontroler dan PC sebagai *interface*. Gambar 2.3 menunjukkan mekanik robot lengan *Smart Arm Robotic AX-12*.

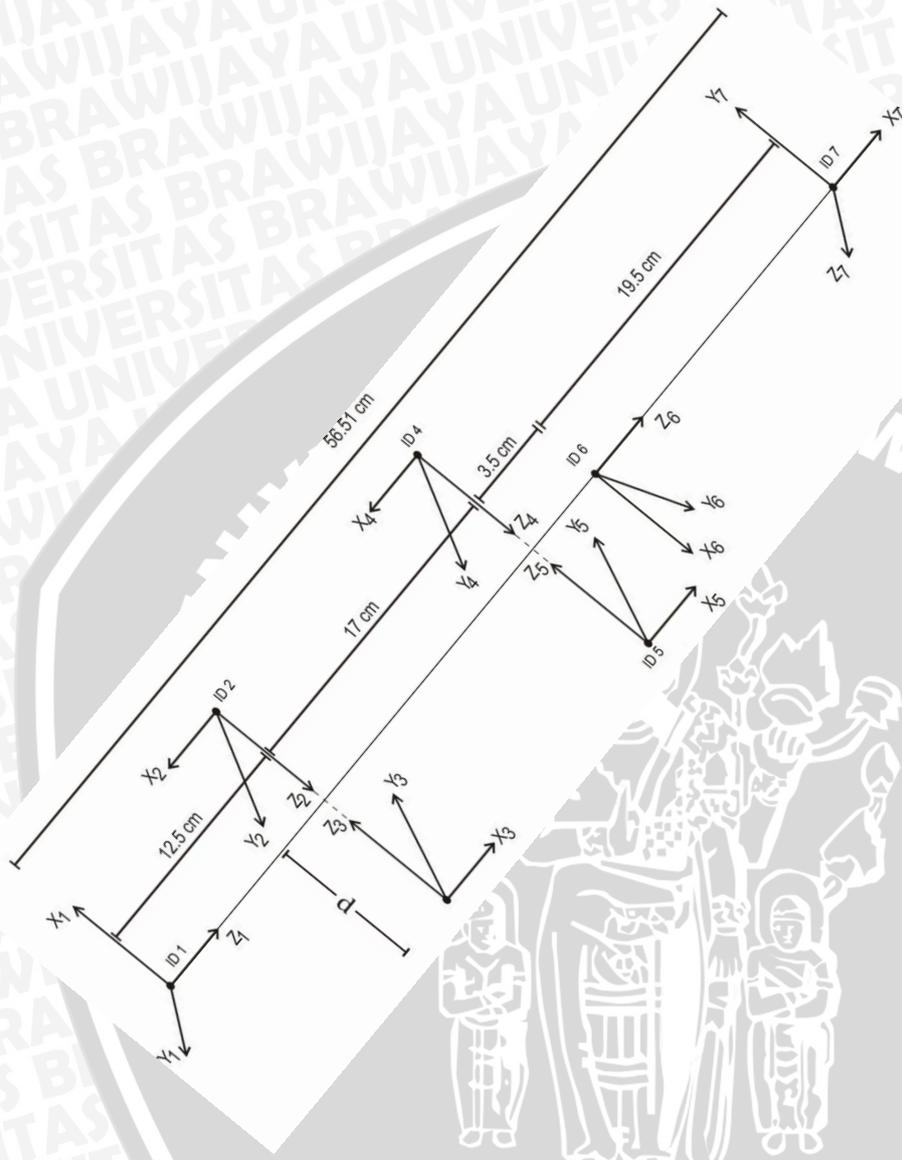


**Gambar 2.3 Mekanik dari robot lengan *Smart Arm Robotic AX-12A***  
(<http://www.crustcrawler.com>)

Sistem koordinat robot lengan mengacu pada standar yang digunakan Mark W. Spong, 2004, setiap *link* diberi nomor dimulai dari bagian *basement*. *Basement* disebut *link 0*. Batang yang bergerak dan berhubungan dengan *basement* disebut *link 1*. Batang yang bergerak dan berhubungan dengan *link 1* disebut *link 2* dan seterusnya. Sehingga secara keseluruhan lengan robot terbentuk dari *link 0*, *link 1*, *link 2*, *link 3* dan *end effector*. Antara *link* yang satu dengan *link* tetangganya dihubungkan dengan *joint*. *Joint i+1* adalah *joint* yang menghubungkan *link i* dengan *link i+1*. Sehingga *joint 1* menghubungkan *link 0* dan *link 1*. Sedangkan *link 1* dan *link 2* dihubungkan dengan *joint 2* dan seterusnya.

Dari keempat *joint* yang dimiliki lengan robot lengan, seluruhnya memiliki satu derajat kebebasan. *Frame* merupakan sistem koordinat yang menggambarkan posisi sebuah *link* relatif terhadap *link* lainnya. Sistem koordinat ini melekat pada *link*. Penomoran *frame* sesuai dengan penomoran *link* yang dilekatinya. *Frame* ditempatkan pada *link* sehingga sumbu *Z* dari *frame {i}* yaitu  $Z_i$ , sejajar dengan

sumbu putar (poros) dari *joint i*. Pendeklarasian frame ditunjukkan dalam gambar 2.4 berikut :



**Gambar 2.4 Pendeklarasian frame pada Robot Lengan ( Perancangan)**

Beberapa parameter yang digunakan yaitu  $a_i$ ,  $\alpha_i$ ,  $d_i$ , dan  $\theta_i$ . Jarak dari  $Z_{i-1}$  ke  $Z_i$  diukur sepanjang  $X_i$  disebut  $a_i$ . Sudut antara  $Z_{i-1}$  dengan  $Z_i$  diukur sepanjang  $X_i$  disebut  $\alpha_i$ . Parameter  $d_i$  adalah jarak dari  $X_{i-1}$  ke  $X_i$  diukur sepanjang  $Z_i$ . Tabel 2.1 berikut ini menunjukkan parameter link dari lengan robot sesuai dengan peletakan *frame* yang ditunjukkan Gambar 2.4



**Tabel 2.1** Parameter *Link* pada Lengan Robot

L	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
0	12.5 cm	0	0	0
1	17.5 cm	0	0	$\theta_1$
2	23 cm	0	0	$\theta_2$
3	0 cm	$\alpha_3$	0	0

### 2.3 Smart Arm Robotic AX-12A

*Smart Arm Robotic AX-12A* merupakan robot lengan berbasis motor servo produksi dari *Crust Crawler Robotis USA*. Fitur *Smart Arm Robotic AX-12A* di desain dengan kecepatan dan torsi yang lebih bagi pengguna yang ingin performa maksimal dari lengan robot mereka. Robot ini memiliki konstruksi bahan rangka yang seluruhnya terbuat dari alumunium yang kokoh dan kuat serta tahan terhadap korosi. Robot ini memiliki empat derajat kebebasan ditambah sebuah *gripper*.

*Smart Arm Robotic AX-12A* menggunakan 7 buah motor servo sebagai aktuator. Yakni yang terdiri dari 1 servo untuk pivot, 2 servo untuk lengan, 2 servo untuk siku, 1 servo pergelangan dan 1 servo (penjapit). Selain itu, robot ini juga memiliki berbagai pilihan *gripper*. Secara keseluruhan, spesifikasi dari *Smart Arm Robotic AX-12A* adalah sebagai berikut :

- Panjang keseluruhan ( dari dasar ) - 22,25 " ( 56.51cm )
- Ketinggian Base - 1 " ( 2.54cm )
- Turntable Axis ke Axis 2 - 4,875 " ( 12.38cm )
- Axis 2 ke Axis 3 - 6,875 " ( 17.46cm )
- Axis 3 dengan sumbu 4 - 2,625 " ( 6.67cm )
- Axis 4 sampai akhir gripper ( terbuka ) - 5,25 " ( 13.33cm )
- Axis 4 sampai akhir gripper ( ditutup ) - 6,875 " ( 17.46cm )
- Panjang keseluruhan dengan gripper ditutup - 22.25 " ( 56.51cm )
- Max lebar 5.5 " ( 13.97cm ) dengan posisi gripper terbuka
- Max lebar lengan 4,4 " ( 11.18cm )
- Rotasi dasar servo berpusat di persegi 4 " X 4 " (10.16 cm X 10.16 cm)
- Setiap sumbu memiliki range 300 °, 1024 langkah (0,29 ° / step)

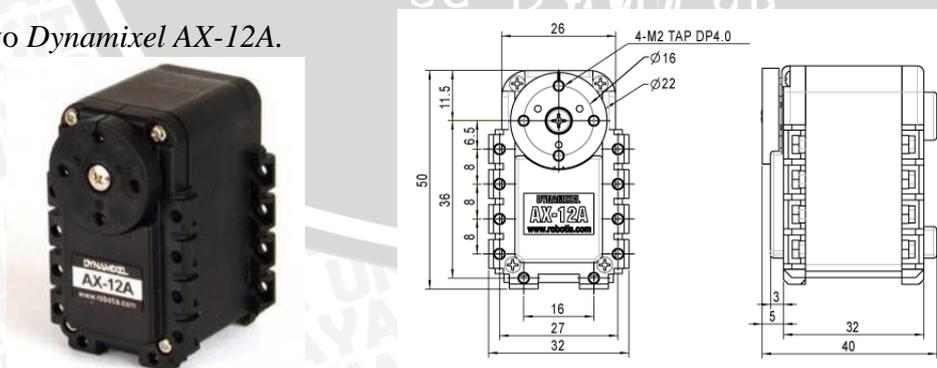
- Bantalan presisi dari bola baja karbon.
- Memiliki 7 servo *Dynamixel AX-12A / AX-12+* ( 4 sumbu + 1 gripper ).

## 2.4 Motor Servo

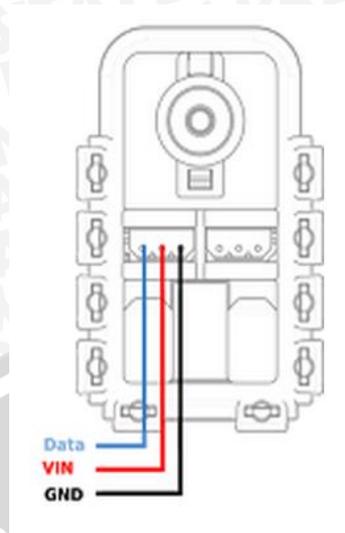
Motor merupakan bagian utama dari sebuah robot. Hampir semua jenis robot kecuali yang menggunakan *muscle wire* (kawat otot) selalu menggunakan motor. Terdapat beberapa jenis motor di pasaran dan untuk merancang sebuah robot maka kita harus dapat memilih motor yang tepat sesuai fungsinya pada robot tersebut.

Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor servo akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor servo terdiri dari sebuah motor, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo.

Motor servo mampu bekerja dua arah dimana arah dan sudut pergerakan motornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle Pulse Width Modulation (PWM)* pada bagian pin kontrolnya. Pada robot, motor servo sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar. Dalam Gambar 2.5 berikut ini diperlihatkan bentuk fisik motor servo *Dynamixel AX-12A*. Sedangkan dalam Gambar 2.6 berikut ini diperlihatkan konfigurasi pin motor servo *Dynamixel AX-12A*.



Gambar 2.5 Motor Servo Dynamixel AX-12A (<http://support.robotis.com>)



Gambar 2.6 Pin Out Kabel Motor Servo *Dynamixel AX-12A*

Motor servo mampu bekerja dua arah yaitu CW (*clockwise*) atau searah jarum jam dan CCW (*counter clockwise*) atau berlawanan arah jarum jam yang arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Secara umum terdapat dua jenis motor servo berdasarkan kemampuannya, yaitu:

1. Motor Servo *Standard 180°*

Motor servo jenis ini merupakan motor yang hanya mampu bergerak dua arah (CW dan CCW) dan mempunyai *defleksi* masing-masing sudut mencapai  $90^\circ$  sehingga total *defleksi* sudut dari kanan – tengah – kiri adalah  $180^\circ$ .

2. Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) dan tanpa batasan *defleksi* sudut putar (dapat berputar secara kontinu) sehingga motor ini berputar  $360^\circ$ .

Paling tidak ada dua varian motor servo yang saat ini beredar di pasaran, yaitu servo analog dan digital. Motor servo analog menerima sinyal dari sebuah penguat, kurang lebih 50 Hz. Sinyal ini membuat penguat untuk meng-*update* posisi motor. Servo digital menggunakan frekuensi lebih tinggi untuk meng-*update* posisi motor, kurang lebih 300 Hz. Dengan meng-*update* posisi motor lebih sering, servo digital dapat memberikan torsi penuh sejak awal pergerakan, dan meningkatkan daya genggam servo. *Refresh* yang lebih cepat juga membuat

servo digital memiliki *deadband* yang lebih kecil, respons yang lebih cepat dan akselerasi yang lebih halus. Namun, harga yang harus dibayar untuk kemampuan ini adalah konsumsi daya yang lebih tinggi.

Servo analog dan digital memiliki tiga kabel. Satu kabel untuk catu, kabel kedua untuk *ground*, dan kabel ketiga merupakan kabel data. Kontroler berkomunikasi dengan servo melalui kabel ini, melalui sinyal berupa pulsa. Melihat ukuran fisik servo, paling tidak ada tiga ukuran servo yang beredar di pasaran umum yaitu: *micro*, *standard* dan *giant* atau *1/4 scale*.

Hal lain adalah spesifikasi *speed* dan *torque rating*. *Speed* adalah kecepatan putar per derajat tertentu, biasanya per  $60^\circ$ . Sebagai contoh,  $0.12 \text{ sec}/60^\circ$ , berarti perlu waktu 0.12 detik untuk memutar lengan servo atau roda sejauh  $60^\circ$ . *Torque* menentukan gaya putar maksimum yang dapat diberikan sebuah servo. Spesifikasi ini diukur dalam *ounces-inch (oz-in)* atau kilogram-centimeter (kg-cm). Bila kita memiliki sebuah servo dengan spesifikasi 4 kgf.cm, maka dengan lengan sepanjang 2 cm, gaya maksimum yang dapat diberikan di ujung lengan adalah 2 kg.

Pada skripsi ini, motor servo yang digunakan adalah motor servo *Dynamixel AX-12A*. *Dynamixel AX-12A* merupakan motor servo cerdas yang memiliki torsi hingga 15kgf.cm dan dilengkapi kemampuan *networking* melalui antarmuka UART TTL *half duplex multidrop*. *Dynamixel AX-12A* terdiri dari *reduction gear*, *precision DC Motor*, dan rangkaian kontrol yang telah dilengkapi dengan kemampuan komunikasi (*networking*) dalam satu kemasan. AX-12A merupakan versi terbaru dari AX-12+ dengan kemampuan yang sama tetapi dengan penambahan desain eksternal. Sedangkan AX-12+ sendiri merupakan versi perbaikan dari AX-12 dengan perbaikan pada desain rangkaian, material, dan khususnya perbaikan pada *wheel gear*. Saat ini jenis yang tersedia adalah AX-12A.

Motor servo ini adalah jenis motor servo digital dengan fitur *daisy-chained*. Dimana satu servo dan yang lainnya, bisa dirangkai secara *serial daisy-chained*. Masing-masing servo, yang memiliki ID yang unik, diperintah melalui satu jalur serial, dan tidak memerlukan sinyal kontrol yang kontinu. Motor servo *Dynamixel AX-12A* memiliki spesifikasi utama sebagai berikut :

- Posisi dan kecepatan dapat diatur dengan resolusi hingga 1024 step.
- *Running Degree* :
  - 0°-300° (*Actuator Mode*).
  - Endless Turn (Wheel Mode)*.
- Resolusi putaran : 0.29° (@ 0-300°)
- *Gear Reduction Ratio* : 254 : 1.
- *Stall Torque* : 12kgf.cm(@12.0V, 1.5A).
- Kecepatan (tanpa beban) : 59 rpm (@12V), 0.269/60° @ 7V, 0.196/60° @ 10V.
- *Axis Bearing*, meminimalkan losses efisiensi torsi untuk beban yang berat.
- Dilengkapi dengan *alarm system*, jika terjadi penyimpangan nilai parameter sistem akan memberikan *alert* (peringatan) atau otomatis langsung menangani masalah (misalnya menon-aktifkan torsi).
- Antarmuka *UART TTL level Multi Drop* dengan pengkabelan menggunakan tipe *daisy chain*.
- Tipe protokol : *half duplex Asynchronous Serial Communication* (8 data bit, 1 stop bit, tanpa paritas).
- Kecepatan komunikasi data : 7343bps - 1 Mbps.
- *Commands Signal* : *Digital Packet*.
- Nomor ID : 254 ID (0 - 253).
- *Data feedback* : posisi, temperatur, tegangan input, kuat arus dll.
- *Running temperature* : -5°C - +70°C.
- Tegangan : 9 - 12 VDC (direkomendasikan 11.1 VDC) / 900mA (Max) atau 50mA (Stand-By).
- Material : *Engineering Plastic*.
- Beberapa parameter seperti posisi, kecepatan, dan torsi dapat diatur dengan sebuah paket perintah (*command packet*) untuk memudahkan kontroler utama mengendalikan beberapa unit modul.
- Dilengkapi LED indikator status *error* dan *alarm system*.
- Kompatibel dengan *OPEN CM 9.04*, *CM-5*, *CM-5100*, *CM-700*, *USB2Dynamixel*, dan *frame* untuk Bioloid.

## 2.5 Arduino Mega

*Arduino Mega* adalah board mikrokontroler berbasis *ATmega1280*. Memiliki 54 pin input dan output digital, dimana 14 pin diantara pin tersebut dapat digunakan sebagai output PWM dan 16 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *board Arduino Mega* ke komputer dengan menggunakan kabel USB atau dengan adaptor AC-DC untuk menjalankannya.

Secara fisik, ukuran *Arduino Mega* hampir kurang lebih 2 kali lebih besar dari *Arduino Uno*, ini untuk mengakomodasi lebih banyaknya pin digital dan analog pada *board Arduino Mega* tersebut. Tampilan *Arduino Mega* dapat dilihat dalam Gambar 2.7 berikut ini.



Gambar 2.7 Tampak depan Arduino Mega (<http://arduino.cc/en/Main/ArduinoBoardMega>)

### 2.5.1 Daya

*Arduino Mega* dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya akan dipilih secara otomatis. Eksternal (*non-USB*) daya dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan sebuah 2.1 mm konektor pusat-positif ke konektor listrik pada board. Baterai dapat dimasukkan ke dalam header pin Gnd dan Vin dari konektor daya.

*Board* dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. apabila dicatu kurang dari 7V, pin 5V dapat menyediakan kurang dari 5V dan board bisa menjadi tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Kisaran yang disarankan adalah 7 sampai 12 volt.

Pin untuk daya adalah sebagai berikut:

- VIN. Tegangan input ke *board Arduino* ketika itu menggunakan sumber daya eksternal (sebagai lawan 5 volt dari koneksi USB atau sumber daya lain). Anda dapat menyediakan tegangan melalui pin ini, atau jika memasok tegangan melalui konektor listrik, mengaksesnya melalui pin ini.
- 5V. catu daya yang digunakan untuk daya mikrokontroler dan komponen lainnya di dalam board. Ini bisa masuk dari VIN melalui regulator on-board, atau dipasok oleh USB atau suplai 5V lainnya.
- 3V- 3,3 Volt pasokan yang dihasilkan oleh chip FTDI *on-board*. Menarik arus maksimum 50 mA.
- GND. Pin *ground*.

### 2.5.2 Pemrograman

*Arduino Mega* dapat diprogram dengan software Arduino. *ATmega 1280* pada *Arduino Mega* dilengkapi dengan bootloader yang memungkinkan untuk meng-*upload* kode baru untuk tanpa menggunakan programmer hardware eksternal.

### 2.6 Qt

Qt Framework sudah sejak lama digunakan untuk mengembangkan aplikasi lintas platform. Qt sendiri dibuat pada tahun 1996 oleh perusahaan dari swedia yang bernama Trolltech. Karena sifatnya yang lintas platform sehingga dapat membuat aplikasi yang berjalan diatas platform Windows, Linux, dan Mac. Dengan Qt kode yang sama dapat dijalankan pada target platform yang berbeda.



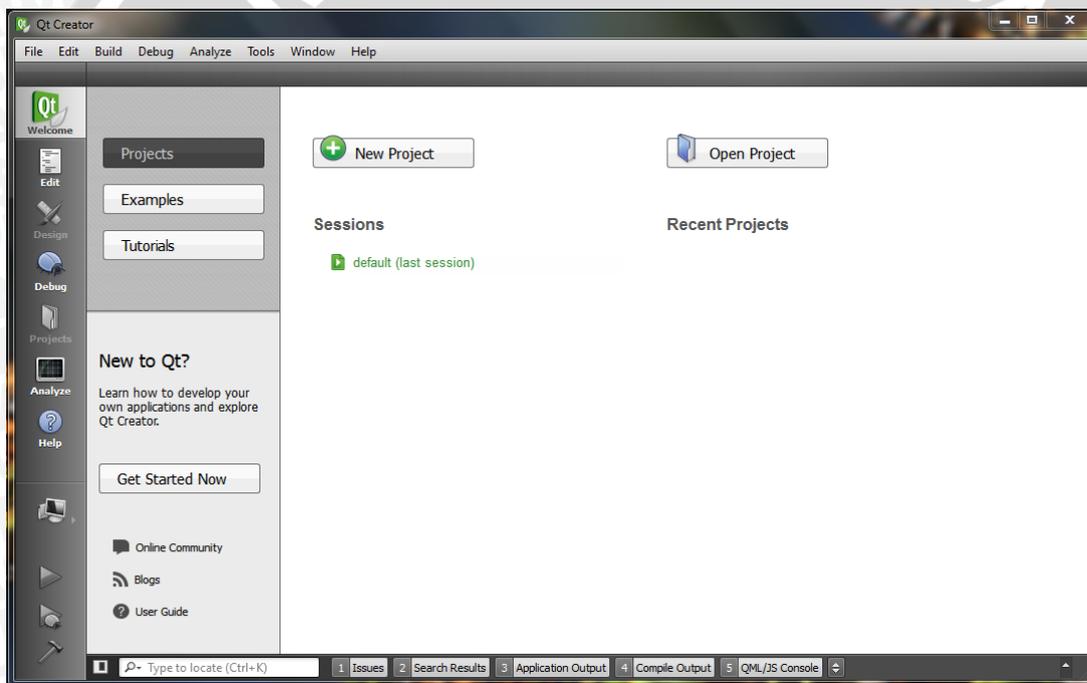
Gambar 2.8 Logo *Qt Framework* (<http://qt.digia.com>)

*Qt Framework* sudah didesain sedemikian rupa sehingga mudah digunakan oleh developer tanpa harus mengorbankan fleksibilitas dan efisiensi. *Qt* mendukung pengembangan dengan dua bahasa utama yaitu *Object Oriented C++* dan *Java*. Pada perancangan ini antarmuka pada skripsi dibuat dari aplikasi *Qt* ini.

*Qt Framework* merupakan framework application cross platform yang cocok untuk para *C++ programmer*. Sebagai *cross platform development* program, *Qt* dapat membuat program untuk berbagai platform. *Qt* tersedia untuk versi *Windows*, *Linux* dan *Mac*. *Qt* ini merupakan salah satu *framework* yang dapat mempermudah kita dalam membuat suatu aplikasi dengan pemrograman *C++*. *Qt framework* sendiri merupakan sebuah frame yang akan membantu kita dalam menyelesaikan program aplikasi yang ingin kita buat. *Qt* memiliki *Framework Codeigniter*. Dengan *Codeigniter* inilah maka semua plugin, module, dan trik-trik penulisan cepat sudah ada dalam satu paket. Sehingga akan sangat membantu untuk *developer*. Bisa dibayangkan jika tidak menggunakan *framework*, maka pengguna harus mengetik sendiri kode-kode tersebut per-huruf, sehingga pembuatan software membutuhkan waktu yang cukup lama hingga siap digunakan.

*Qt* adalah *cross-platform application* dan *UI Framework* untuk *developer* yang menggunakan bahasa *C++* atau *QML*. *QML* itu adalah bahasa *C++* dari *Qt* yang mirip seperti *css* dan *javascript*. Untuk *IDE* dari *Qt* telah disediakan *Qt Creator*, tetapi bisa juga menggunakan *IDE* lain seperti *CodeBlocks* dan *VisualStudio*. *Qt Creator* merupakan bagian dari *Nokia Qt SDK* (sejak tahun 2008 *Qt* diakuisisi oleh *Nokia*). *Qt* adalah *Framework* dibawah lisensi *LGPL opensource* yang dapat kita gunakan untuk membuat sebuah aplikasi yang gratis maupun aplikasi komersial.

Versi *Qt* yang digunakan adalah versi *Qt 5.0*. *Qt 5.0* merupakan versi yang direlease akhir 2012, ini adalah versi terbaik dan perbaikan dari versi-versi sebelumnya. Versi ini lebih cepat dari versi sebelumnya sekaligus mendukung paradigma *User Interface* yang sedang trend saat ini yaitu *Touch Screen* dan *Tablets*. *Qt 5.0* akan sangat memanjakan penggunaanya karena sangat mudah dan fleksibel dengan dukungan *Javascript* dan *QML* yang masih berpedoman pada *C++*. Untuk *Javascript Backend* sendiri didukung oleh *V8* dan *Qt Webkit 2* yang akan sangat membantu sekali terutama untuk *Web Developer HTML5*. Ada satu kelebihan lain dibandingkan dengan versi sebelumnya, portabilitas *cross platform* dalam *Qt 5* jauh lebih simple. Gambar 2.9 berikut ini memperlihatkan *Qt Creator* pada mode *Windows*.



Gambar 2.9 *Qt Creator* Windows mode (Blanchette, Jasmin. 2006 : 54)

### BAB III

#### METODE PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang suatu sistem pengendalian pergerakan robot lengan *Smart Arm Robotic AX-12A* melalui pendekatan *Geometry - Based Kinematic* yang dikendalikan oleh *Arduino Mega* dan antarmuka menggunakan *Qt*. Selain itu robot lengan *Smart Arm Robotic AX-12A* dikendalikan agar dapat mengambil benda untuk dapat dipindahkan dengan memperhatikan posisi objek mula-mula ke posisi yang dituju sesuai dengan yang direncanakan. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

#### 3.1. Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian pergerakan robot lengan *Smart Arm Robotic AX-12A* dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan menggunakan peralatan di Laboratorium Mekatronika dan Robotika serta Laboratorium Sistem Kontrol Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, meliputi:

Penentuan sistem yang digunakan.

- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
  - a. Penentuan deskripsi kerja sistem secara keseluruhan.
  - b. Rangkaian elektronik pendukung.
- Perancangan program pada *Arduino Mega* sebagai pusat pengontrol sistem.

Pelaksanaan perancangan sistem dilakukan di Laboratorium Mekatronika dan Robotika dan Laboratorium Sistem Kontrol Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

#### 3.2. Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Penyempurnaan mekanik robot lengan *Smart Arm Robotic AX-12A* secara keseluruhan yang meliputi penentuan batasan area kerja robot lengan *Smart Arm Robotic AX-12A*.
- Pembuatan rangkaian elektronika meliputi rangkaian *buffer* dengan menggunakan *IC 74LS241N*.
- Pembuatan program pengontrolan menggunakan *Arduino Mega* dan *Qt*.

### 3.3. Pengujian dan Analisa Data

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut:

1. Pengujian pada motor servo yang digunakan.

Pengujian motor servo dilakukan dengan cara mencatat dan membandingkan hasil pembacaan sudut motor servo yang terukur menggunakan busur dengan sudut motor servo yang sebenarnya dihasilkan sesuai kalibrasi datasheet dan hasil konversi secara manual.

2. Pengujian pada tiap-tiap blok rangkaian.

Pengujian tersebut meliputi pengujian *forward kinematic* dan *invers kinematic*.

3. Pengujian *stall torque*.

Pengujian *stall torque* atau torsi tanpa pembebanan dilakukan pada motor servo ID1 sampai ID7 sebanyak 20 kali tiap ID dan mencatat hasil pengukuran *stall torque* oleh torsimeter Lutron TQ-8800 serta melihat *LED indicator* dalam kondisi menyala atau mati kemudian hasilnya ditabulasikan.

4. Pengujian torsi pembebanan

Pengujian torsi pembebanan dilakukan pada motor servo ID1 sampai ID7 menggunakan torsimeter Lutron TQ-8800. Pengujian dilakukan dengan menambahkan beban seberat 500 gram, 700 gram, dan 900 gram. Pengujian ini dilakukan sebanyak 10 kali tiap ID dan mencatat hasil pengukuran *stall torque* oleh torsimeter Lutron TQ-8800 serta melihat

*LED indicator* dalam kondisi menyala atau mati kemudian hasilnya ditabulasikan.

5. Mengadakan pengujian sistem secara keseluruhan.

Pengujian ini dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat dan memberikan analisa terhadap kinerja alat. Mekanismenya, pengujian dilakukan dengan 4 macam posisi yang berbeda-beda. Pengujian tersebut dilakukan dengan menggunakan algoritma *invers kinematic* dan *forward kinematic* yang sudah ditentukan sebelumnya untuk melihat apakah sistem bisa berjalan tanpa ada *error* atau dengan *error* seminimal mungkindengan penggunaan algoritma yang sama dengan cara mengubah posisi awal.

6. Mengevaluasi hasil pengujian sistem secara keseluruhan.

Evaluasi dilakukan untuk mengetahui kinerja alat secara keseluruhan untuk mengetahui apa yang menjadi kekurangan dalam sistem tersebut.

#### **3.4. Pengambilan Kesimpulan dan Saran**

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.

## BAB IV

### PERANCANGAN SISTEM DAN PEMBUATAN PROGRAM

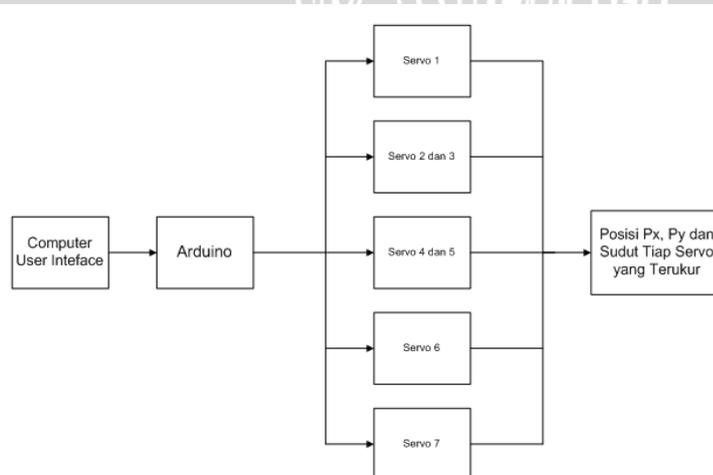
#### 4.1 Perancangan Sistem

Perancangan sistem terlebih dahulu kita lakukan agar memudahkan pemahaman tentang pengendalian dan prinsip kerja dari robot lengan. Perancangan sistem robot lengan meliputi:

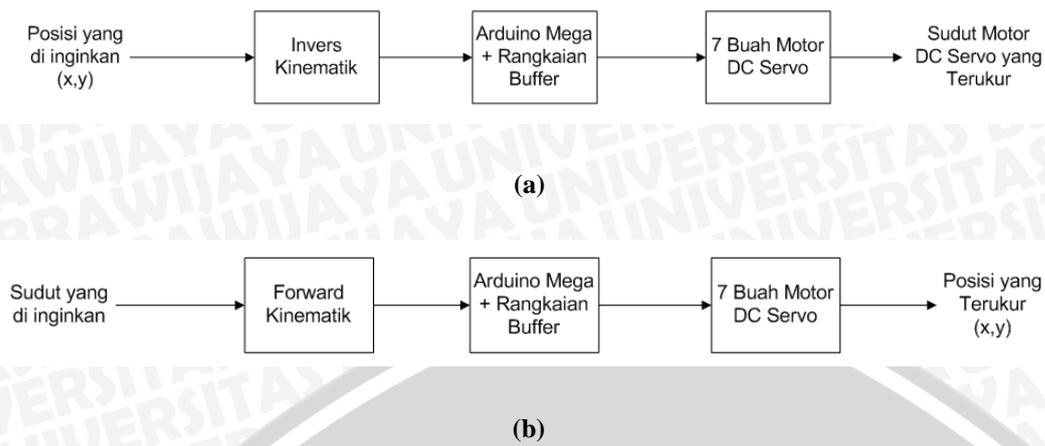
- Diagram balok dan prinsip kerja robot lengan.
- Deskripsi *frame* robot lengan

##### 4.1.1 Diagram Balok dan Prinsip Kerja Robot Lengan

Robot lengan ini pada dasarnya memiliki empat *degree of freedom* (DOF) yang akan dimanfaatkan untuk penerapan metode *forward* dan *invers* pada skripsi ini. Gambar 4.1 adalah diagram balok perangkat keras sedangkan Gambar 4.2 adalah diagram balok sistem pengaturan. Komputer sebagai antarmuka dengan pengguna. Pengguna dapat menggunakan dua metode yaitu *forward kinematic* dan *invers kinematic*. Blok mikrokontroler yaitu sebuah modul Arduino Mega 2560 (seperti tampak dalam Gambar 4.1) yang memiliki fungsi sebagai driver untuk mengatur posisi sudut motor servo. Motor servo sebagai aktuator penggerak *end-effector*. *End-effector* adalah perangkat pada ujung robot lengan, yang dirancang untuk berinteraksi dengan lingkungan (berupa penjepit). Posisi *end-effector* dapat dilihat berdasarkan papan penunjuk koordinat *end-effector*.



Gambar 4.1 Diagram Balok Perangkat Keras (Perancangan)



**Gambar 4.2 Diagram Balok Sistem Pengaturan (Perancangan)**

(a) **Invers Kinematik** (b) **Forward Kinematik**

Dalam Gambar 4.3 berikut ini diperlihatkan mekanik robot lengan *Smart Arm Robotic AX-12A*

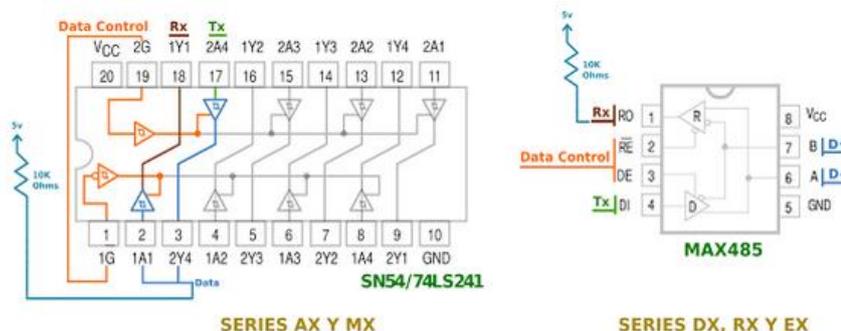


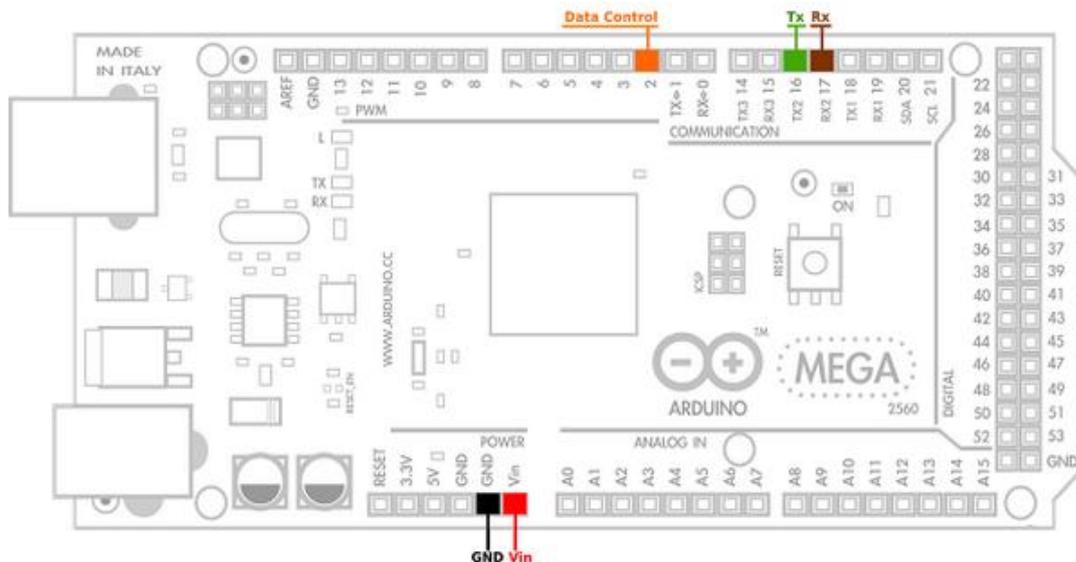
**Gambar 4.3 Smart Arm Robotic AX-12A (www.crustcrawler.com)**



Gambar 4.4 Modul *Arduino Mega* (<http://arduino.cc>)

Komunikasi serial *Arduino* berbasis *full duplex* atau dengan 2 jalur data, 1 mengirim (pin TX) dan 1 menerima (pin RX). Sedangkan komunikasi serial pada *Dynamixel* berbasis *half duplex* dimana hanya butuh 1 jalur data untuk berkomunikasi. Oleh karena itu masalah pertama yang muncul saat akan mengkomunikasikan *Arduino* dengan *Dynamixel* adalah ketidaksesuaian antara komunikasi serial dalam *Arduino* dengan komunikasi serial pada *Dynamixel*. Untuk itu diperlukan sebuah perangkat lain untuk menghubungkan *Arduino* dengan *Dynamixel* atau dengan kata lain untuk mengubah *full duplex* ke *half duplex*. Pada kasus ini diperlukan IC 74LS241N sebagai pengendali atau pengkonversi jalur data. Jadi, sederhananya IC 74LS241N adalah sebuah gerbang yang memilah-milah apakah data dialamatkan sebagai data *transfer* (TX data) atau data *receiver* (RX data). Konfigurasi IC 74LS241N dengan *Board Arduino Mega* ditunjukkan dalam Gambar 4.5 berikut ini:





Gambar 4.5 Konfigurasi IC 74LS241N dengan Board Arduino Mega (<http://savageelectronics.blogspot.com>)

#### 4.1.2 Deskripsi Frame Robot Lengan

Perancangan program dilakukan dengan terlebih dahulu mendefinisikan *frame* pada tiap-tiap *link* pada robot lengan, berikut pendefinisian *frame*. Pada robot Smart Arm Robotic AX-12A. Bagian-bagian pada robot lengan diantaranya batang-batang logam aluminium yang berfungsi sebagai *link*, enam buah motor servo yang berfungsi sebagai aktuator dan sebuah motor servo berfungsi sebagai penjepit pada *end effector*. Pengendalian lengan dibantu dengan suatu sistem mikrokontroler dan PC sebagai *interface*.

Pada skripsi ini hanya memanfaatkan empat derajat kebebasan. Sistem koordinat robot lengan mengacu pada standar yang digunakan Mark W. Spong, 2004, setiap *link* diberi nomor dimulai dari bagian *basement* yang bergerak. *Basement* disebut *link 0*. Batang yang bergerak dan berhubungan dengan *basement* disebut *link 1*. Batang yang bergerak dan berhubungan dengan *link 1* disebut *link 2* dan seterusnya. Sehingga secara keseluruhan lengan robot terbentuk dari *link 0*, *link 1*, *link 2*, dan *end-effector*. Antara *link* yang satu dengan *link* tetangganya dihubungkan dengan *joint*. *Joint i+1* adalah *joint* yang menghubungkan *link i* dengan *link i+1*. Sehingga *joint 1* menghubungkan *link 0* dan *link 1*. Sedangkan *link 1* dan *link 2* dihubungkan dengan *joint 2* dan

seterusnya. Dari keempat *joint* yang dimiliki lengan robot lengan, seluruhnya memiliki satu derajat kebebasan. *Frame* merupakan sistem koordinat yang menggambarkan posisi sebuah *link* relatif terhadap *link* lainnya. Sistem koordinat ini melekat pada *link*. Penomoran *frame* sesuai dengan penomoran *link* yang dilekatinya. *Frame* ditempatkan pada *link* sehingga sumbu Z dari *frame* {*i*} yaitu  $Z_i$ , sejajar dengan sumbu putar (poros) dari *joint* *i*. Berikut ini pendeklarasian frame robot lengan seperti yang tampak pada Gambar 4.6 :



Gambar 4.6 Pendeklarasian *Frame* Pada Robot Lengan *Smart Arm Robotic AX-12*



Beberapa parameter yang digunakan yaitu  $a_i$ ,  $\alpha_i$ ,  $d_i$ , dan  $\theta_i$ . Jarak dari  $Z_{i-1}$  ke  $Z_i$  diukur sepanjang  $X_i$  disebut  $a_i$ . Sudut antara  $Z_{i-1}$  dengan  $Z_i$  diukur sepanjang  $X_i$  disebut  $\alpha_i$ . Parameter  $d_i$  adalah jarak dari  $X_{i-1}$  ke  $X_i$  diukur sepanjang  $Z_i$ . Parameter  $\theta_i$  adalah sudut antara  $X_{i-1}$  dengan  $X_i$  diukur sepanjang  $Z_i$ . Tabel 4.1 menunjukkan parameter *link* dari lengan robot sesuai dengan peletakan *frame* yang ditunjukkan dalam Gambar 4.6.

**Tabel 4.1 Parameter *Link* pada Lengan Robot**

L	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
0	12.5 cm	0	0	0
1	17.5 cm	0	0	$\theta_1$
2	23 cm	0	0	$\theta_2$
3	0 cm	$\alpha_3$	0	0

Pada skripsi ini yang digunakan hanya link 1 dan 2, dengan matrik homogen sebagai berikut :

$$A_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots\dots(4.1)$$

$$A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots\dots(4.2)$$

## 4.2 Perancangan Kinematika Robot Lengan

Kinematika robot lengan merupakan pengetahuan atau teori mengenai pergerakan objek tanpa memperhitungkan gaya-gaya yang menyebabkan objek itu bergerak. Kinematika robot lengan terdiri dari pergerakan rotasi dan translasi. Pada penelitian ini, tinjauan kinematika akan digunakan dalam menentukan perhitungan *Forward Kinematic* dan *Invers Kinematic*.

#### 4.2.1 Forward Kinematic

Secara garis besar *forward kinematic* adalah memberikan sudut kepada motor servo setelah itu kita dapatkan posisi *end-effector*. Perancangan *forward kinematic* kita hitung terlebih dahulu *homogenous* matrik semua *link*. Berikut ini adalah penyusunan matrik transformasi homogen dari matrik homogen di tiap-tiap link yang akan dipergunakan dalam perhitungan *forward kinematic*. Penyusunan matrik homogen diperlihatkan dalam Persamaan 4.3 sampai 4.16 berikut ini:

$$A_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.3)$$

$$A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.4)$$

$$A_1 \cdot A_2 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.5)$$

$$A_1 \cdot A_2 = \begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} \\ e_{21} & e_{22} & e_{23} & e_{24} \\ e_{31} & e_{32} & e_{33} & e_{34} \\ e_{41} & e_{42} & e_{43} & e_{44} \end{bmatrix} \dots\dots(4.6)$$

Dengan:

$$e_{11} = \{(\cos \theta_1 \cdot \cos \theta_2) + (-\sin \theta_1 \cdot \sin \theta_2) + 0 + 0\}$$

$$= \{(\cos \theta_1 \cdot \cos \theta_2) - (\sin \theta_1 \cdot \sin \theta_2)\}$$

$$e_{12} = \{(\cos \theta_1 \cdot (-\sin \theta_2)) + ((-\sin \theta_1) \cdot \cos \theta_2) + 0 + 0\}$$

$$= -\{(\cos \theta_1 \cdot \sin \theta_2) + (\sin \theta_1 \cdot \cos \theta_2)\}$$

$$e_{13} = 0$$

$$e_{14} = \{(\cos \theta_1 \cdot a_2 \cos \theta_2) + ((-\sin \theta_1) \cdot a_2 \cos \theta_2) + 0 + (a_1 \cos \theta_1)\}$$

$$= \{(\cos \theta_1 \cdot a_2 \cos \theta_2) - (\sin \theta_1 \cdot a_2 \cos \theta_2) + (a_1 \cos \theta_1)\}$$

$$e_{21} = \{(\sin \theta_1 \cdot \cos \theta_2) + (\cos \theta_1 \cdot \sin \theta_2) + 0 + 0\}$$

$$= \{(\sin \theta_1 \cdot \cos \theta_2) + (\cos \theta_1 \cdot \sin \theta_2)\}$$

$$e_{22} = \{(\sin \theta_1 \cdot (-\sin \theta_2)) + (\cos \theta_2 \cdot \cos \theta_1) + 0 + 0\}$$

$$\begin{aligned}
 &= \{(\sin \theta_1 \cdot (-\sin \theta_2)) + (\cos \theta_2 \cdot \cos \theta_1)\} \\
 e_{23} &= 0 \\
 e_{24} &= \{(\sin \theta_1 \cdot a_2 \cos \theta_2) + (\cos \theta_1 \cdot a_2 \sin \theta_2) + 0 + (a_1 \sin \theta_1)\} \\
 &= \{(\sin \theta_1 \cdot a_2 \cos \theta_2) + (\cos \theta_1 \cdot a_2 \sin \theta_2) + (a_1 \sin \theta_1)\} \\
 e_{31} &= 0 \\
 e_{32} &= 0 \\
 e_{33} &= 1 \\
 e_{34} &= 0 \\
 e_{41} &= 0 \\
 e_{42} &= 0 \\
 e_{43} &= 0 \\
 e_{44} &= 1
 \end{aligned}$$

Dengan mengacu pada persamaan trigonometri dasar tentang jumlah dan selisih dua sudut berikut :

$$\cos(\theta_1 - \theta_2) = (\cos \theta_1 \cdot \cos \theta_2) + (\sin \theta_1 \cdot \sin \theta_2) \dots\dots(4.7)$$

$$\cos(\theta_1 + \theta_2) = (\cos \theta_1 \cdot \cos \theta_2) - (\sin \theta_1 \cdot \sin \theta_2) \dots\dots(4.8)$$

$$\sin(\theta_1 - \theta_2) = (\sin \theta_1 \cdot \cos \theta_2) - (\cos \theta_1 \cdot \sin \theta_2) \dots\dots(4.10)$$

$$\sin(\theta_1 + \theta_2) = (\sin \theta_1 \cdot \cos \theta_2) + (\cos \theta_1 \cdot \sin \theta_2) \dots\dots(4.11)$$

Maka didapatkan :

$$\begin{aligned}
 &A_1 \cdot A_2 \\
 &= \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & \{(a_1 \cos \theta_1) + (a_2 \cos(\theta_1 + \theta_2))\} \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & \{(a_1 \sin \theta_1) + (a_2 \sin(\theta_1 + \theta_2))\} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.12)
 \end{aligned}$$

$$T_1^0 = A_1 \dots\dots(4.13)$$



$$T_2^0 = A_1 \cdot A_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -s \sin(\theta_1 + \theta_2) & 0 & \{(a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2))\} \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & \{(a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2))\} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.14)$$

$$T_2^0 = A_1 \cdot A_2 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.15)$$

Dengan memasukkan nilai a<sub>1</sub> dan a<sub>2</sub> yang ada pada Tabel 4.1 maka didapatkan :

$$A_1 = \begin{bmatrix} \cos(\theta_1) & -s \sin(\theta_1) & 0 & (17 \cos(\theta_1)) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & (17 \sin(\theta_1)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \cos(\theta_2) & -s \sin(\theta_2) & 0 & (23 \cos(\theta_2)) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & (23 \sin(\theta_2)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^0 = A_1$$

$$T_2^0 = A_1 \cdot A_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -s \sin(\theta_1 + \theta_2) & 0 & \{(17 \cos(\theta_1) + 23 \cos(\theta_1 + \theta_2))\} \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & \{(17 \sin(\theta_1) + 23 \sin(\theta_1 + \theta_2))\} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = A_1 \cdot A_2 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.16)$$

$$n_x = \cos(\theta_1 + \theta_2)$$

$$n_y = \sin(\theta_1 + \theta_2)$$

$$n_z = 0$$

$$o_x = -s \sin(\theta_1 + \theta_2)$$

$$o_y = \cos(\theta_1 + \theta_2)$$

$$o_z = 0$$



$$\begin{aligned} a_x &= 0 \\ a_y &= 0 \\ a_z &= 1 \end{aligned}$$

$$\begin{aligned} P_x &= \{(17 \cos(\theta_1) + 23 \cos(\theta_1 + \theta_2))\} \\ P_y &= \{(17 \sin(\theta_1) + 23 \sin(\theta_1 + \theta_2))\} \\ P_z &= 0 \end{aligned}$$

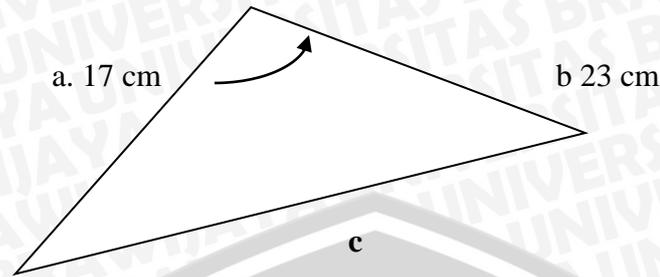
Matrik *forward* memberikan informasi orientasi dan posisi dari *end-effector*. Pada skripsi ini tidak membahas mengenai orientasi *end-effector* hanya posisi *end-effector*. Dapat dilihat pada persamaan diatas, posisi koordinat (Px, Py) *end-effector* dapat kita hitung dengan memberikan masukan sudut pada motor servo untuk setiap *link*. Pergerakan robot merupakan kombinasi perubahan  $\theta_1$  dan  $\theta_2$  secara bersamaan. Namun melihat kondisi yang paling memungkinkan terkait keterbatasan fisik robot maka perubahan pada masing-masing sudut  $\theta_1$  dan  $\theta_2$  dibatasi agar tidak merusak robot.

#### 4.2.2 *Invers Kinematic*

Pada *invers kinematic* kita menggunakan aturan *cosinus* untuk mendapatkan solusi sudut pada posisi tertentu *end-effector*. Aturan *cosinus* berlaku untuk sudut dalam segitiga, jadi langkah awal yang perlu dilakukan adalah membentuk segitiga pada tiap *link*.

##### 4.2.2.1 Membentuk Segitiga untuk Tiap-tiap *Link*

Pada mulanya kita lakukan pembentukan segitiga pada bagian *link* I dan *link* II untuk mencari jangkauan minimum dan maksimum dari c.. Seperti yang kita ketahui panjang  $a_1 = a = 17$  cm,  $a_2 = b = 23$  cm (lihat Tabel 4.1) dengan kemampuan maksimum dan minimum sudut motor servo untuk berputar 0 derajat sampai 300 derajat dan netral pada 150 derajat. Kemampuan motor servo untuk membentuk sudut akan sangat menentukan panjang sisi c (lihat Gambar 4.7). Nilai c minimal didapatkan dengan memasukkan sudut  $0^\circ$  ke *cosinus*  $\phi$  pada Persamaan untuk menghitung nilai c. Sedangkan nilai c maksimal didapatkan dengan memasukkan sudut  $90^\circ$  ke *cosinus*  $\phi$  pada Persamaan 4.17 untuk menghitung nilai c. Pada Persamaan 4.10 didapatkan nilai c minimal 28.6 cm dan maksimal 40 cm.



Gambar 4.7 Segitiga *Link* Penghubung Bagian Atas (Perancangan)

$$c^2 = a^2 + b^2 - 2ab \cos \phi \quad \dots\dots(4.17)$$

C minimal

$$c^2 = 17^2 + 23^2 - 2 * 17 * 23 * \cos 90^0$$

$$c = 28.6 \text{ cm}$$

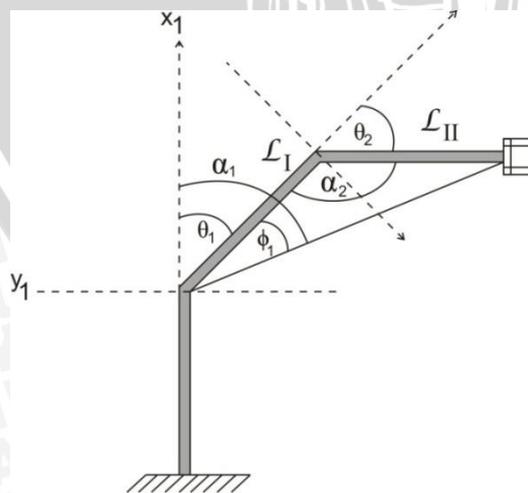
C maksimal

$$c^2 = 17^2 + 23^2 - 2 * 17 * 23 * \cos 180^0$$

$$c = 40 \text{ cm}$$

$$28.6 \text{ cm} \leq c \leq 40 \text{ cm}$$

Pembentukan segitiga dilakukan dengan cara dua menghubungkan *joint* terbawah oleh suatu garis imajinasi ke posisi yang diinginkan, dan *link* II (bagian atas) mengarahkan secara langsung *end-effector*. Dalam Gambar 4.8 berikut ini ditunjukkan pembentukan segitiga untuk mencari solusi *invers kinematic*:



Gambar 4.8 *Invers Kinematic* dengan Metode Segitiga (Perancangan)

#### 4.2.2.2 Solusi Sudut Motor Servo

*Variable* pada *frame* robot lengan didefinisikan kembali untuk memudahkan penyusunan solusi-solusi sudut pada motor servo.

- Solusi untuk motor servo *link* 1

$$\tan \alpha_1 = \frac{y_0}{x_0}$$

$$\alpha_1 = \arctan \frac{y_0}{x_0}$$

$$\phi_1 = \arccos \frac{(a^2 + c^2 - b^2)}{2 * a * c}$$

$$a = 17 \text{ cm}$$

$$b = 23 \text{ cm}$$

$$c = \sqrt{y_0^2 + x_0^2}$$

Jika alpha 1 positif (+) maka

$$\theta_1 = \alpha_1 - \phi_1$$

Jika alpha 1 negatif (-) maka

$$\theta_1 = \alpha_1 + \phi_1$$

- Solusi untuk motor servo *link* 2

Jika alpha 1 positif (+) maka

$$\theta_2 = 180 - \alpha_2$$

Jika alpha 1 negatif (-) maka

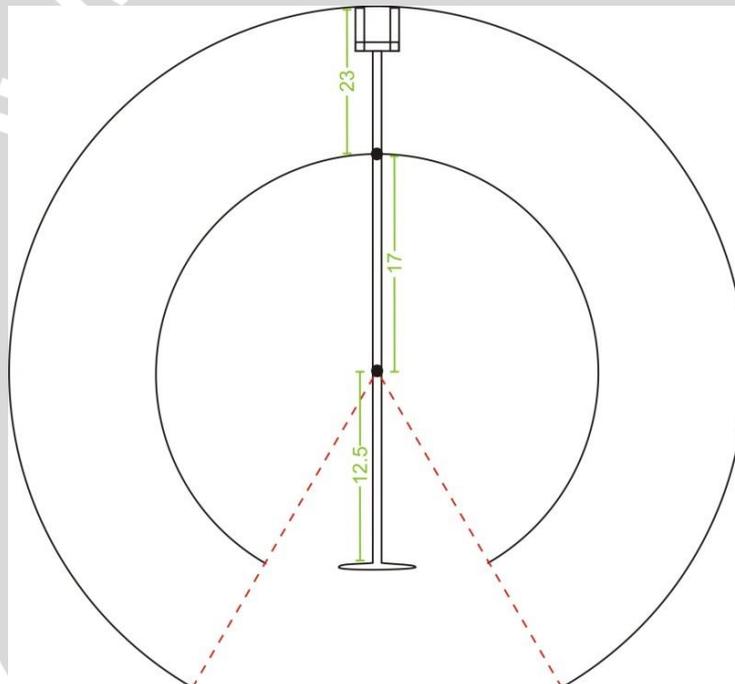
$$\theta_2 = -(180 - \alpha_2)$$

### 4.3 Perancangan Pergerakan Robot Lengan

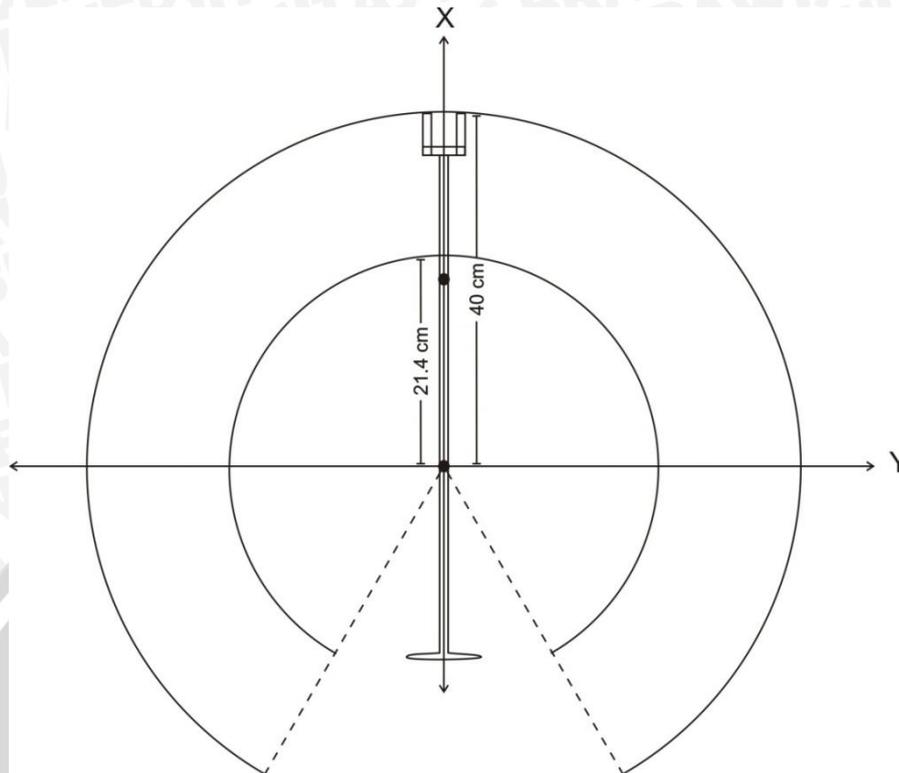
Perancangan pergerakan robot lengan meliputi penentuan area kerja robot lengan yang dipergunakan. Selain itu diperlukan juga penentuan desain dimensi atau ukuran medan robot lengan. Hal ini dimaksudkan agar pengguna dapat mengetahui daerah jangkauan minimum mdan maksimum gerakan robot lengan pada medan yang telah ditentukan.

### 4.3.1 Workspace Robot Lengan

*Workspace* robot lengan adalah total luas yang memungkinkan terlewati (tersapu) oleh gerakan robot lengan. *Workspace* dapat kita tentukan dengan cara menggerakkan tiap motor servo secara berurutan sehingga kita dapat mengetahui jangkauan robot lengan dengan keterbatasan fisik robot.. Gambar 4.9 menunjukkan *workspace* robot lengan *Smart Arm Robotic AX-12A* beserta panjang *link* dari posisi *base* hingga *gripper*. Sedangkan Gambar 4.10 memperlihatkan *workspace* robot lengan dengan jangkauan minimum dan maksimum pergerakan robot lengan tersebut.



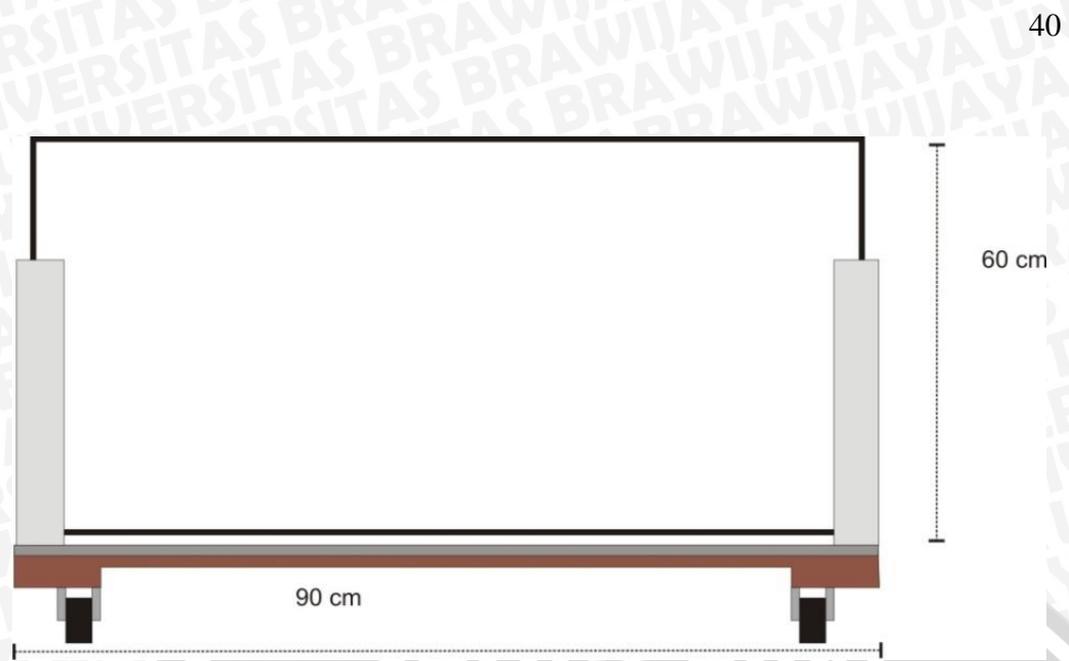
Gambar 4.9 *Workspace* Robot Lengan (Perancangan)



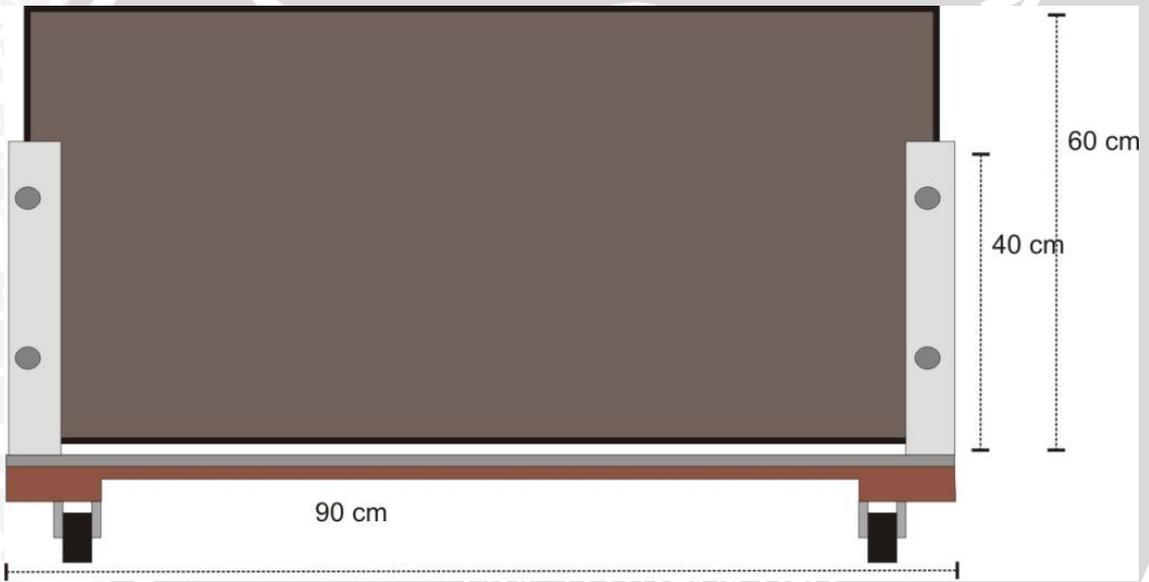
**Gambar 4.10 Jangkauan min-max Robot Lengan (Perancangan)**

Motor servo yang kita gunakan memiliki keterbatasan sudut yaitu antara 0 derajat sampai 300 derajat. Motor servo paling bawah kita gerakkan (dari sudut min hingga max) maka akan terbentuk juring lingkaran dengan jari-jari sekitar 40 cm. Kondisikan motor servo bagian bawah pada kondisi sudut maks atau min lalu gerakan motor servo bagian tengah sehingga kita dapatkan ruang gerak baru yang melewati (tersapu), yaitu seperempat lingkaran dengan jari-jari 24 cm.

Dalam Gambar 4.9, menunjukkan *workspace* robot lengan tanpa ada batasan medan. Setelah kita ketahui *workspace* robot lengan maka dapat kita lakukan perancangan medan robot lengan untuk mengambil objek. Medan robot lengan kami sesuaikan dengan kemampuan jangkauan robot lengan (lihat Gambar 4.10). Penempatan papan difungsikan untuk mengukur posisi dari *end-effector*.

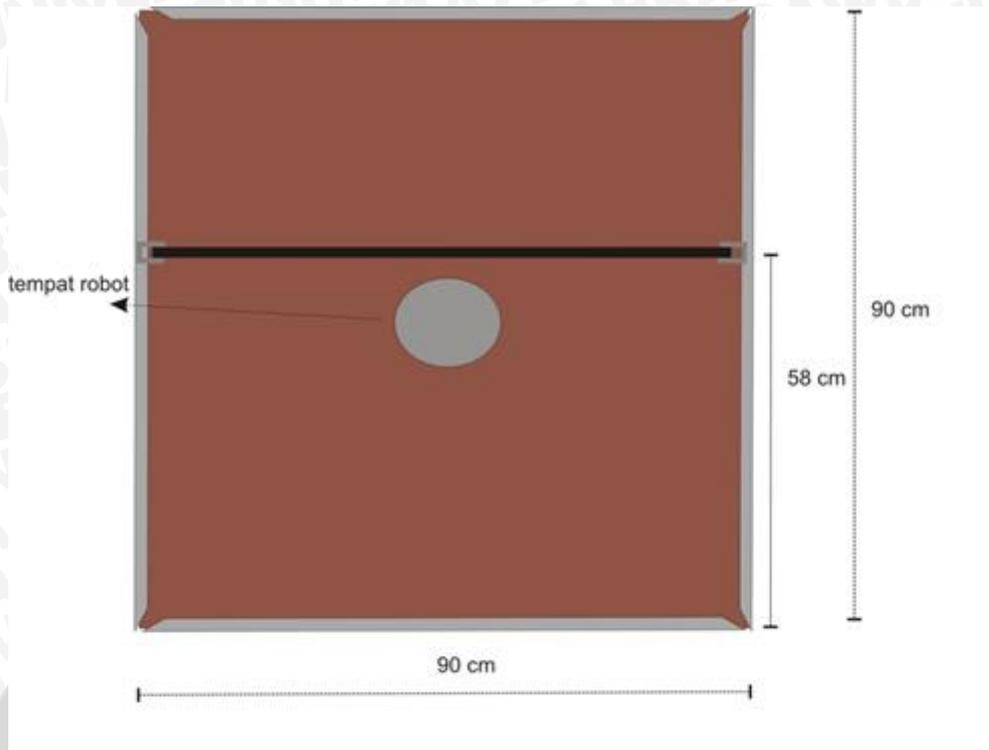


(a) Tampak depan

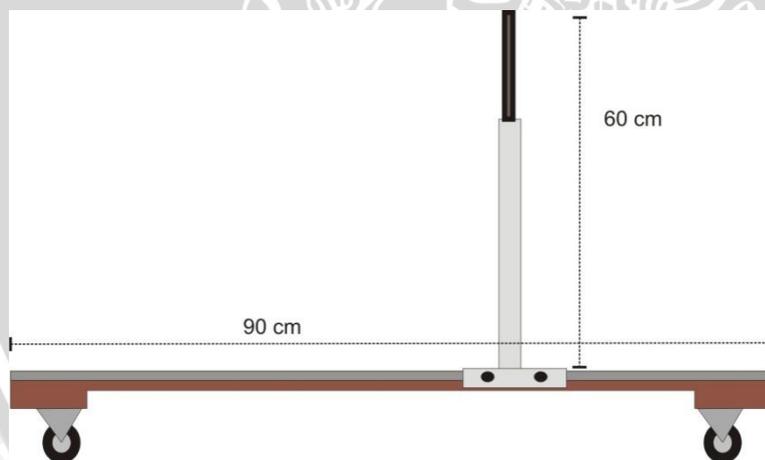


(b) Tampak belakang





(c) Tampak atas



(d) Tampak samping

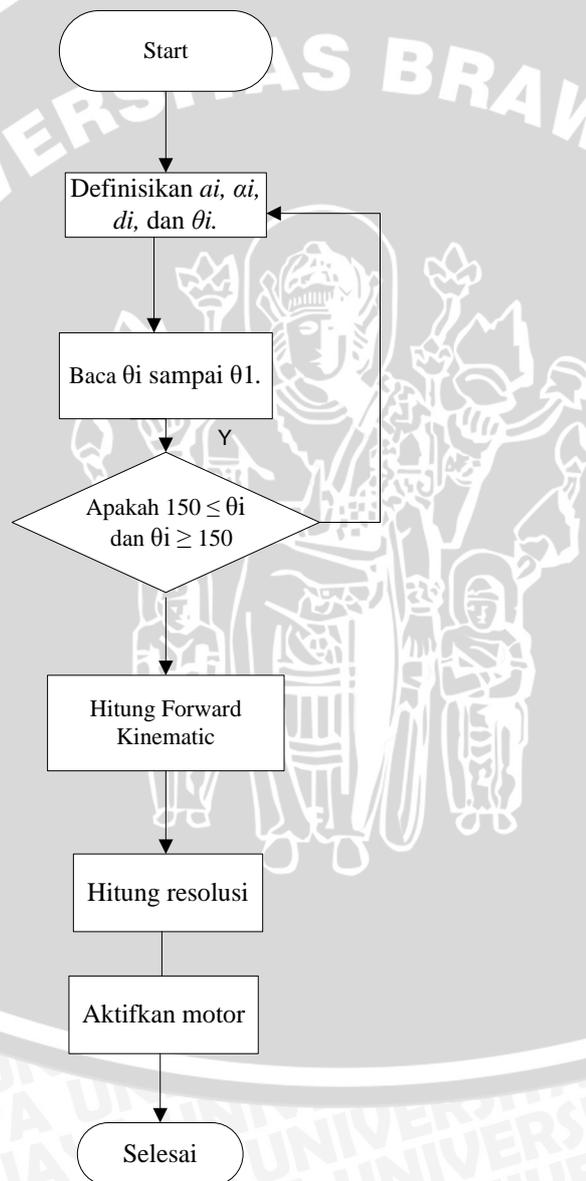
Gambar 4.11 Rancangan Medan Robot Lengan (Perancangan)

### 4.3.2 Skema Gerak Robot Lengan

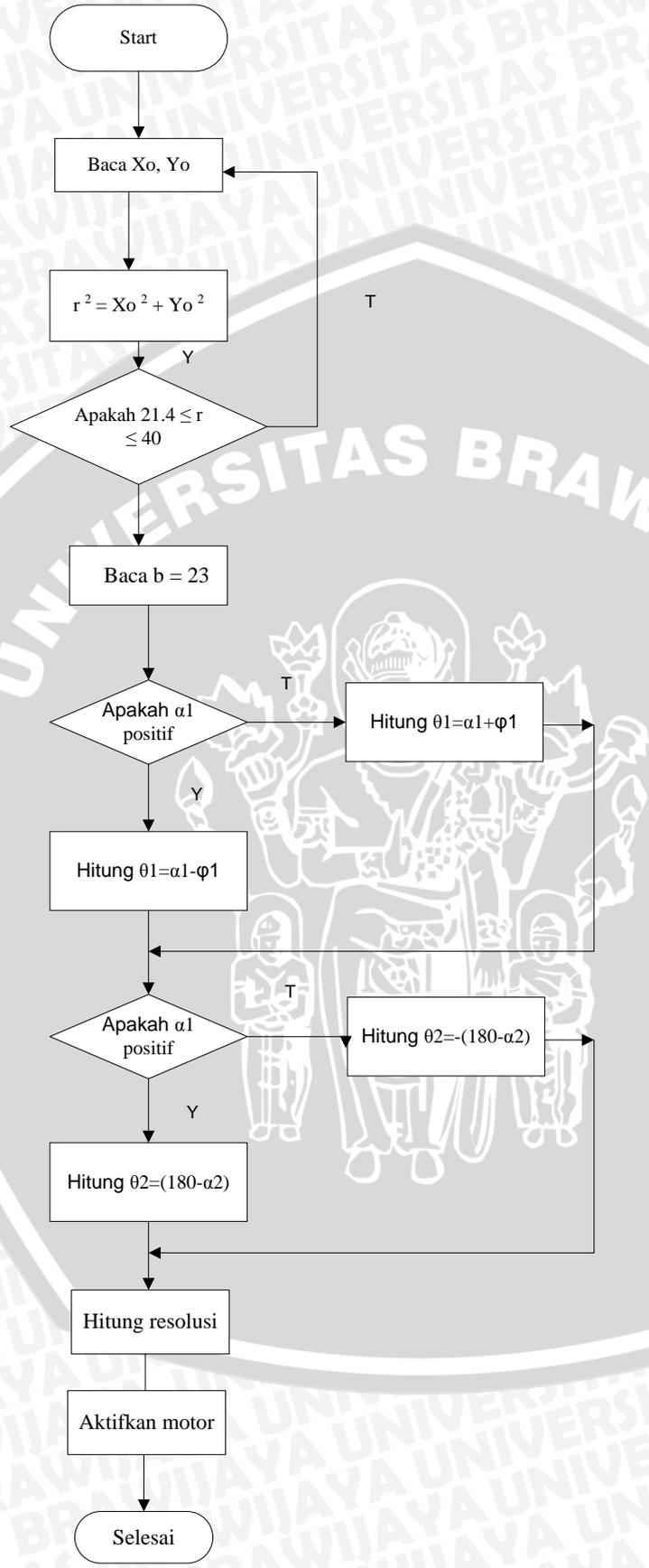
Gerak robot lengan sangat dipengaruhi kemampuan motor servo robot lengan dan juga jangkauan posisi yang mungkin dituju oleh metode *invers kinematic* dengan solusi geometri (segitiga). Perancangan *invers kinematic* dengan solusi geometri (segitiga) diketahui memiliki jangkauan posisi

antara 28.6 cm sampai 40 cm dari koordinat dasar, selama objek berada pada jangkauan tersebut maka robot lengan akan mampu mengambil objek tersebut. Objek diletakkan pada posisi dalam jangkauan, posisi penghalang dan objek diatur sesuai dengan kemampuan robot lengan. Posisi objek dapat ditaruh pada semua posisi dalam area solusi *invers kinematic*.

#### 4.3.3 Perancangan Perangkat Lunak



Gambar 4.12 Flowchart Program Forward Kinematic (Perancangan)



Gambar 4.13 Flowchart Program Invers Kinematic (Perancangan)



#### 4.3.4 Konversi

Motor servo yang digunakan pada skripsi ini merupakan jenis motor servo digital yang bekerja dari rentang sudut 0 derajat sampai 300 derajat atau dalam posisi resolusi 0 bit sampai 1023 bit. Karena pada program antarmuka untuk *forward kinematic* pengguna terbiasa memberikan input berupa sudut dalam satuan derajat, maka diperlukan perhitungan dalam bentuk perbandingan sederhana untuk memudahkan pemrograman. Berikut ini adalah perhitungan konversi :

- Konversi dari resolusi (bit) ke sudut (derajat)

$$\frac{n \text{ (bit)}}{1024} = \frac{x^0}{300^0} \quad \dots\dots(4.18)$$

misalkan  $n = 512$ , maka :

$$\frac{512}{1024} = \frac{x^0}{300^0}$$

$$x = \frac{512 \cdot 300^0}{1024}$$

$$x = 150^0$$

- Konversi dari sudut (derajat) ke resolusi (bit)

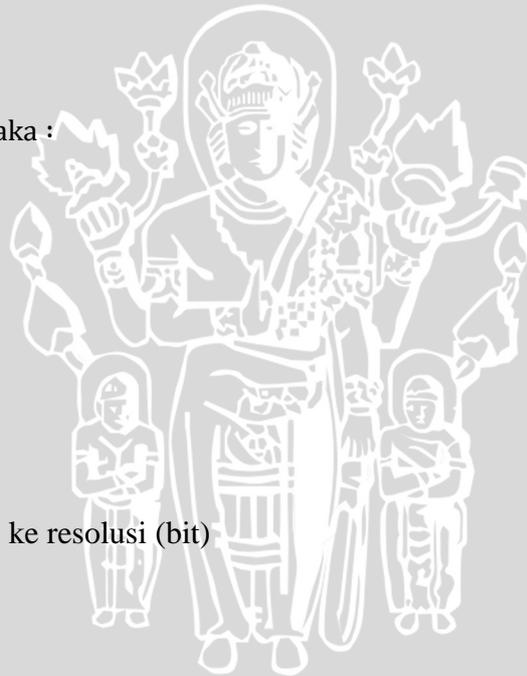
$$\frac{x \text{ (bit)}}{1024} = \frac{n^0}{300^0} \quad \dots\dots(4.19)$$

misalkan  $n = 150^0$ , maka :

$$\frac{x \text{ (bit)}}{1024} = \frac{150^0}{300^0}$$

$$x = \frac{150 \cdot 1024^0}{300}$$

$$x = 512 \text{ bit}$$



Dari perhitungan konversi perbandingan antara hasil konversi dari perbandingan manual dan dari *datasheet* yang ditabulasikan dalam Table 4.1 berikut :

**Tabel 4.1 Perbandingan Hasil Konversi Manual dan *Datasheet***

No	Resolusi	Perhitungan	<i>Datasheet</i>
1	0	150	148.48
2	64	131.25	129.92
3	128	112.5	111.36
4	192	93.75	92.8
5	256	75	74.24
6	320	56.25	55.68
7	384	37.5	37.12
8	448	18.75	18.46
9	512	0	0
10	576	-18.75	-18.46
11	640	-37.5	-37.12
12	704	-56.25	-55.68
13	768	-75	-74.24
14	832	-93.75	-92.8
15	896	-112.5	-111.36
16	960	-131.25	-129.92
17	1023	-150	-148.48

Berdasarkan perbandingan hasil konversi manual dan yang berasal dari *datasheet* akan dicari nilai koefisien korelasi antara keduanya. Dengan mengetahui nilai koefisien korelasi antara fungsi yang dibentuk oleh hasil perhitungan dan fungsi yang dibentuk oleh hasil *datasheet*, maka dapat dilihat hubungan yang dibentuk oleh keduanya saling terkait erat atau tidak. Koefisien korelasi biasa dilambangkan dengan huruf  $r$  dimana nilai  $r$  dapat bervariasi dari  $-1$  sampai  $+1$ . Nilai  $r$  yang mendekati  $-1$  atau  $+1$  menunjukkan hubungan yang kuat antara dua fungsi tersebut dan nilai  $r$  yang mendekati  $0$  mengindikasikan lemahnya hubungan antara dua fungsi tersebut. Sedangkan tanda  $+$  (positif) dan  $-$

(negatif) memberikan informasi mengenai arah hubungan antara dua fungsi tersebut. Jika bernilai + (positif) maka kedua fungsi tersebut memiliki hubungan yang searah. Dalam arti lain peningkatan X akan bersamaan dengan peningkatan Y dan begitu juga sebaliknya. Jika bernilai - (negatif) artinya korelasi antara kedua variabel tersebut bersifat berlawanan. Peningkatan nilai X akan bersamaan dengan penurunan Y.

Pada skripsi ini, dalam menentukan koefisien korelasi digunakan metode *Product Moment Coefficient of Correlation* oleh Karl Pearson (Dass, 2002). Persamaan yang dipergunakan seperti yang ditunjukkan dalam Persamaan 4.20 berikut ini:

$$r = \frac{\sum((x - \bar{x}) \cdot (y - \bar{y}))}{\sqrt{\sum((x - \bar{x}))^2 \cdot \sum(y - \bar{y})^2}} \quad \dots(4.20)$$

Dimana :

$x$  = nilai sudut hasil perhitungan manual

$\bar{x}$  = rata-rata nilai sudut hasil perhitungan manual

$y$  = nilai sudut sesuai *datasheet*

$\bar{y}$  = rata-rata nilai sudut sesuai *datasheet*

Mengacu pada hasil yang tertera pada Tabel 4.1, maka didapatkan nilai koefisien korelasi melalui tabulasi perhitungan seperti yang tampak dalam Tabel 4.2 berikut ini :

**Tabel 4.2 Tabulasi Perhitungan Koefisien Korelasi**

Resolusi	Perhitungan (x)	Datasheet (y)	$(x - \bar{x})$	$(y - \bar{y})$	$(x - \bar{x})^2$	$(y - \bar{y})^2$	$(x - \bar{x}) \cdot (y - \bar{y})$
0	0	0	-150	-148.4741176	22500	22044.5636	22271.11765
64	18.75	18.46	-131.25	-130.0141176	17226.563	16903.6708	17064.35294
128	37.5	37.12	-112.5	-111.3541176	12656.25	12399.7395	12527.33824
192	56.25	55.68	-93.75	-92.79411765	8789.0625	8610.74827	8699.448529
256	75	74.24	-75	-74.23411765	5625	5510.70422	5567.558824
320	93.75	92.8	-56.25	-55.67411765	3164.0625	3099.60738	3131.669118
384	112.5	111.36	-37.5	-37.11411765	1406.25	1377.45773	1391.779412
448	131.25	129.92	-18.75	-18.55411765	351.5625	344.255282	347.8897059

512	150	148.48	0	0.005882353	0	3.4602E-05	0
576	168.75	167.04	18.75	18.56588235	351.5625	344.691988	348.1102941
640	187.5	185.6	37.5	37.12588235	1406.25	1378.33114	1392.220588
704	206.25	204.16	56.25	55.68588235	3164.0625	3100.91749	3132.330882
768	225	222.72	75	74.24588235	5625	5512.45105	5568.441176
832	243.75	241.28	93.75	92.80588235	8789.0625	8612.9318	8700.551471
896	262.5	259.84	112.5	111.3658824	12656.25	12402.3598	12528.66176
960	281.25	278.4	131.25	129.9258824	17226.563	16880.7349	17052.77206
1023	300	296.96	150	148.4858824	22500	22048.0573	22272.88235
$\Sigma$	2550	2524.06	0	-2.27374E-13	143437.5	140571.222	141997.125
<b>MEAN</b>	150	148.4741176					
<b>R</b>	0.999999971						

Berdasarkan tabulasi perhitungan diatas, diperoleh nilai koefisien korelasi antara hasil perhitungan menggunakan konversi manual dan hasil yang berasal dari *datasheet* yaitu nilai (r) sebesar 0.999999971. Nilai koefisien korelasi yang dihasilkan mendekati nilai 1. Hal ini menunjukkan bahwa fungsi yang terbentuk dari perhitungan menggunakan konversi manual dan fungsi yang terbentuk dari informasi *datasheet* memiliki keterkaitan yang kuat dan layak digunakan.

Selain menghitung koefisien korelasi, perlu juga mengetahui nilai *standard error of prediction* ( $E_{yx}$ ) dari fungsi yang didapatkan melalui hasil perhitungan menggunakan konversi manual. *Standard error of prediction* ( $E_{yx}$ ), yang paling baik adalah yang nilainya terletak diantara interval 0 sampai 1. Jika nilai *standard error of prediction* ( $E_{yx}$ ) semakin kecil atau semakin mendekati 0, maka fungsi yang ditinjau memiliki keakuratan nilai yang semakin baik. Dalam menentukan *standard error of prediction* ( $E_{yx}$ ), mula-mula dicari terlebih dahulu fungsi regresi untuk hasil perhitungan konversi manual dan fungsi regresi untuk hasil yang berasal dari *datasheet*. Fungsi regresi yang digunakan adalah fungsi regresi linear. Dan persamaan-persamaan yang digunakan adalah seperti yang tampak dalam Persamaan 4.21 sampai 4.23 berikut ini :

$$na_0 + a_1 \sum x_i = \sum y_i \quad \dots(4.21)$$

$$a_0 \sum x_i + a_1 \sum (x_i)^2 = \sum x_i y_i \quad \dots(4.22)$$

$$y = a_0 + a_1 x_i \quad \dots(4.23)$$

Tabel 4.3 berikut ini digunakan dalam menentukan fungsi regresi untuk nilai-nilai yang diperoleh dari *datasheet* :

**Tabel 4.3 Tabulasi Perhitungan Fungsi Regresi Nilai *Datasheet***

Resolusi ( $x_i$ )	Datasheet ( $y_i$ )	$(x_i)^2$	$(x_i y_i)$	
0	0	0	0	
64	18.46	4096	1181.44	
128	37.12	16384	4751.36	
192	55.68	36864	10690.6	
256	74.24	65536	19005.4	
320	92.8	102400	29696	
384	111.36	147456	42762.2	
448	129.92	200704	58204.2	
512	148.48	262144	76021.8	
576	167.04	331776	96215	
640	185.6	409600	118784	
704	204.16	495616	143729	
768	222.72	589824	171049	
832	241.28	692224	200745	
896	259.84	802816	232817	
960	278.4	921600	267264	
1023	296.96	1046529	303790	
$\Sigma$	8703	2524.06	6125569	1776705

Berdasarkan Persamaan 4.19 dan Persamaan 4.20, maka didapatkan persamaan-persamaan berikut :

$$2524.06 = 17a_0 + 8703 a_1$$

$$1776705 = 8703 a_0 + 6125569 a_1$$

setelah diselesaikan melalui proses eliminasi dan substitusi maka didapatkan nilai – nilai sebagai berikut :

$$a_0 = - 0.047696$$

$$a_1 = 0.290115$$

dan fungsi regresi untuk nilai – nilai yang didapatkan dari *datasheet* yaitu

$$y = - 0.047696 + 0.290115x \dots(4.24)$$



Tabel 4.4 berikut ini digunakan dalam menentukan fungsi regresi untuk nilai-nilai yang diperoleh dari konversi manual :

**Tabel 4.4 Tabulasi Perhitungan Fungsi Regresi Nilai Konversi Manual**

Resolusi ( $x_i$ )	Datasheet ( $y_i$ )	$(x_i)^2$	$(x_i y_i)$
0	0	0	0
64	18.75	4096	1200
128	37.5	16384	4800
192	56.25	36864	10800
256	75	65536	19200
320	93.75	102400	30000
384	112.5	147456	43200
448	131.25	200704	58800
512	150	262144	76800
576	168.75	331776	97200
640	187.5	409600	120000
704	206.25	495616	145200
768	225	589824	172800
832	243.75	692224	202800
896	262.5	802816	235200
960	281.25	921600	270000
1023	300	1046529	306900
$\Sigma$	8703	6125569	1794900

Berdasarkan Persamaan 4.19 dan Persamaan 4.20, maka didapatkan persamaan-persamaan berikut :

$$2550 = 17a_0 + 8703 a_1$$

$$1794900 = 8703 a_0 + 6125569 a_1$$

setelah diselesaikan melalui proses eliminasi dan substitusi maka didapatkan nilai – nilai sebagai berikut :

$$a_0 = - 0.56958$$

$$a_1 = 0.294115$$

dan fungsi regresi untuk nilai – nilai yang didapatkan dari konversi manual yaitu

$$y = - 0.56958 + 0.290115x \quad \dots\dots(4.25)$$

Dalam Tabel 4.5 berikut ini diperlihatkan hasil tabulasi perhitungan untuk mendapatkan *standard error of prediction* ( $E_{yx}$ ) untuk hasil konversi manual :

**Tabel 4.5 Tabulasi Perhitungan *Standard Error of Prediction* ( $E_{yx}$ ) Untuk Hasil Konversi Manual**

Resolusi (x)	Perhitungan (y)	$x^2$	xy	$y_r$	$y-y_r$	$(y-y_r)^2$	$E_{yx}$
0	0	0	0	-0.56959	0.56959	0.3244328	0.3371092
64	18.75	4096	1200	18.25377	0.49623	0.2462442	
128	37.5	16384	4800	37.07713	0.42287	0.178819	
192	56.25	36864	10800	55.90049	0.34951	0.1221572	
256	75	65536	19200	74.72385	0.27615	0.0762588	
320	93.75	102400	30000	93.54721	-0.20279	0.0411238	
384	112.5	147456	43200	112.37057	0.12943	0.0167521	
448	131.25	200704	58800	131.19393	0.05607	0.0031438	
512	150	262144	76800	150.01729	-0.01729	0.0002989	
576	168.75	331776	97200	168.84065	-0.09065	0.0082174	
640	187.5	409600	120000	187.66401	-0.16401	0.0268993	
704	206.25	495616	145200	206.48737	-0.23737	0.0563445	
768	225	589824	172800	225.31073	-0.31073	0.0965531	
832	243.75	692224	202800	244.13409	-0.38409	0.1475251	
896	262.5	802816	235200	262.95745	-0.45745	0.2092605	
960	281.25	921600	270000	281.78081	-0.53081	0.2817593	
1023	300	1046529	306900	300.310055	-0.31006	0.0961341	
$\Sigma$	8703	6125569	1794900	2549.99982	0.000185	1.9319241	

Dalam Tabel 4.6 berikut ini diperlihatkan hasil tabulasi perhitungan untuk mendapatkan *standard error of prediction* ( $E_{yx}$ ) untuk hasil dari *datasheet* :

**Tabel 4.6 Tabulasi Perhitungan *Standard Error of Prediction* ( $E_{yx}$ ) Untuk Hasil dari *Datasheet***

Resolusi (x)	<i>Datasheet</i> (y)	$x^2$	Xy	$y_r$	$y-y_r$	$(y-y_r)^2$	$E_{yx}$
0	0	0	0	-0.047696	0.0477	0.002275	0.064019237
64	18.46	4096	1181.44	18.519664	-0.05966	0.00356	
128	37.12	16384	4751.36	37.087024	0.03298	0.001087	
192	55.68	36864	10690.56	55.654384	0.02562	0.000656	
256	74.24	65536	19005.44	74.221744	0.01826	0.000333	
320	92.8	102400	29696	92.789104	0.0109	0.000119	

	384	111.36	147456	42762.24	111.35646	0.00354	1.25E-05
	448	129.92	200704	58204.16	129.92382	-0.00382	1.46E-05
	512	148.48	262144	76021.76	148.49118	-0.01118	0.000125
	576	167.04	331776	96215.04	167.05854	-0.01854	0.000344
	640	185.6	409600	118784	185.6259	-0.0259	0.000671
	704	204.16	495616	143728.6	204.19326	-0.03326	0.001106
	768	222.72	589824	171049	222.76062	-0.04062	0.00165
	832	241.28	692224	200745	241.32798	-0.04798	0.002302
	896	259.84	802816	232816.6	259.89534	-0.05534	0.003063
	960	278.4	921600	267264	278.4627	-0.0627	0.003932
	1023	296.96	1046529	303790.1	296.73995	0.22005	0.048422
$\Sigma$	8703	2524.06	6125569	1776705	2524.06	-1.3E-05	0.069674

Dalam Tabel 4.5 dan Tabel 4.6 dapat dilihat bahwa nilai *standard error of prediction* ( $E_{yx}$ ) untuk hasil dari konversi manual adalah sebesar 0.3371092 dan nilai *standard error of prediction* ( $E_{yx}$ ) untuk hasil dari *datasheet* adalah sebesar 0.064019237. Hal ini menunjukkan bahwa nilai *standard error of prediction* ( $E_{yx}$ ) dari keduanya sangat bagus karena berada pada rentang 0 sampai 1 . Pada skripsi ini seluruh perhitungan yang akan diterapkan pada program akan mengacu pada konversi manual. Karena nilai *standard error of estimate* ( $E_{yx}$ ) untuk hasil dari konversi manual masih ada dalam batas toleransi maka Persamaan 4.18 dan 4.19 yang digunakan untuk mengkonversi nilai sudut ke bentuk resolusi serta persamaan yang digunakan untuk mengkonversi nilai resolusi ke bentuk sudut masih layak dipergunakan.

## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan.

#### 5.1 Pengujian Motor Servo

##### a. Tujuan

Mengetahui tingkat kelinieran dari motor servo dalam melakukan perubahan sudut dari tiap sendi pada robot lengan.

##### b. Peralatan yang digunakan

- Penggaris busur.
- Robot lengan lengkap dengan motor servo pada masing-masing *link*.
- Board Arduino Mega 2560
- Komputer/laptop.
- Kabel komunikasi serial *USB Type A to B*
- Catu daya 8 volt
- Software Servo Calibrator

##### c. Langkah pengujian

1. Peralatan dirangkai seperti dalam Gambar 5.1.
2. Berikan masukan bit mulai 0 – 1023 bit pada *software Servo Calibrator*.
3. Busur derajat digunakan untuk mengukur sudut *link*.
4. Data keluaran sudut dari motor servo dicatat dan digambarkan grafiknya untuk melihat kelinierannya
5. Agar data pengukuran mendekati keadaan yang sebenarnya langkah 2, 3 dan 4 dilakukan sebanyak tujuh belas kali.



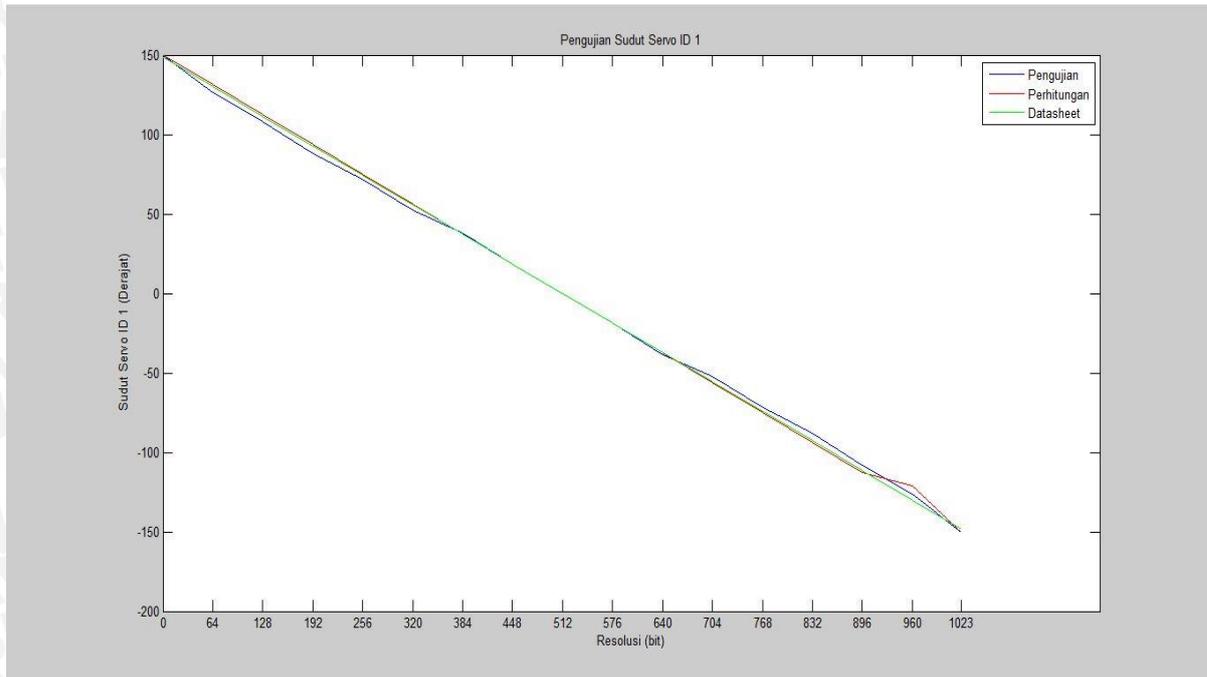
**Gambar 5.1** Blok Pengujian Motor Servo

#### d. Hasil Pengujian

Hasil sudut keluaran dari motor servo sangat dipengaruhi oleh mekanik robot lengan yang kurang ideal. Data hasil pengujian motor servo ditunjukkan pada Tabel 5.1, Tabel 5.2 dan Tabel 5.3 berikut:

**Tabel 5.1** Hasil Pengujian Motor Servo pada *ID 1*

No	Resolusi	Pengujian Sudut Servo ID 1 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	150	150	152	150	150.5	151	150.5	150.6
2	64	125	126	127	126.5	126	128	126	126.33333333
3	128	108	108	108	108	108	108	108	108
4	192	88	88	89	88	89	88	88	88.28571429
5	256	72.5	72	72.5	72	72	72.5	73	72.35714286
6	320	52	52	52	52	52	52	52	52
7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	19	19	18.5	18	18.4
9	512	0	0	0.5	0	0.5	0	0.5	0
10	576	-18	-18	-19	-18	-19	-18.5	-18.5	-18.42857143
11	640	-39	-39	-40	-38	-38.5	-38	-38	-38.64285714
12	704	-53	-52	-52	-52.5	-52	-52.5	-52	-52.28571429
13	768	-72.5	-72	-72	-71	-71	-72	-72.5	-71.85714286
14	832	-88.5	-88	-89	-88	-89	-88	-88	-88.35714286
15	896	-108.5	-108	-108	-109	-108	-108	-108.5	-108.2857143
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150.5	-150	-152	-148	-151	-151	-150.5	-150.3571429



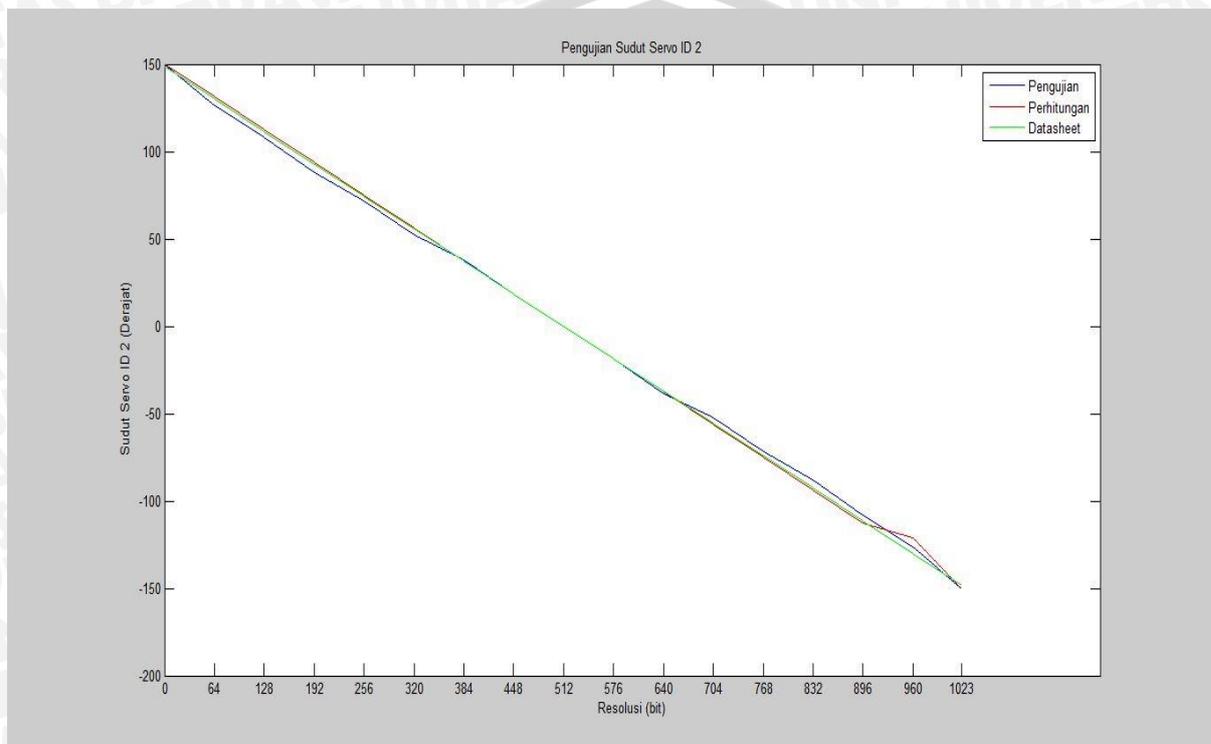
Gambar 5.2 Grafik Sudut Terhadap Resolusi Motor Servo ID 1

Tabel 5.2 Hasil Pengujian Motor Servo pada ID 2

No	Resolusi	Pengujian Sudut Servo ID 2 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	149	150	151	148	150.5	151	150.5	149.8
2	64	125	126	127	126.5	126	128	126	126.33333333
3	128	108	108	108	108	108	108	108	108
4	192	88	88	89	88	89	88	88	88.28571429
5	256	72	72	72	72	72	72	72	72
6	320	52	52	52	52	52	52	52	52
7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	18	19	18.5	18	18.2
9	512	0	0	0.5	0	0.5	0	0.5	0
10	576	-19	-18.5	-18	-18	-19	-18.5	-18	-18.42857143
11	640	-38	-39	-38	-38	-38.5	-38	-39	-38.35714286
12	704	-53	-52	-52	-52	-52	-52	-52	-52.14285714
13	768	-72	-73	-72	-71	-72	-71	-72	-71.85714286

14	832	-88	-88	-89	-88	-89	-88	-88	-88.28571429
15	896	-108	-110	-108	-109	-108	-108	-107	-108.2857143
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150	-150	-151	-150	-151	-150	-150.5	-150.2857143

**Gambar 5.3** Grafik Sudut Terhadap Resolusi Motor Servo ID 2

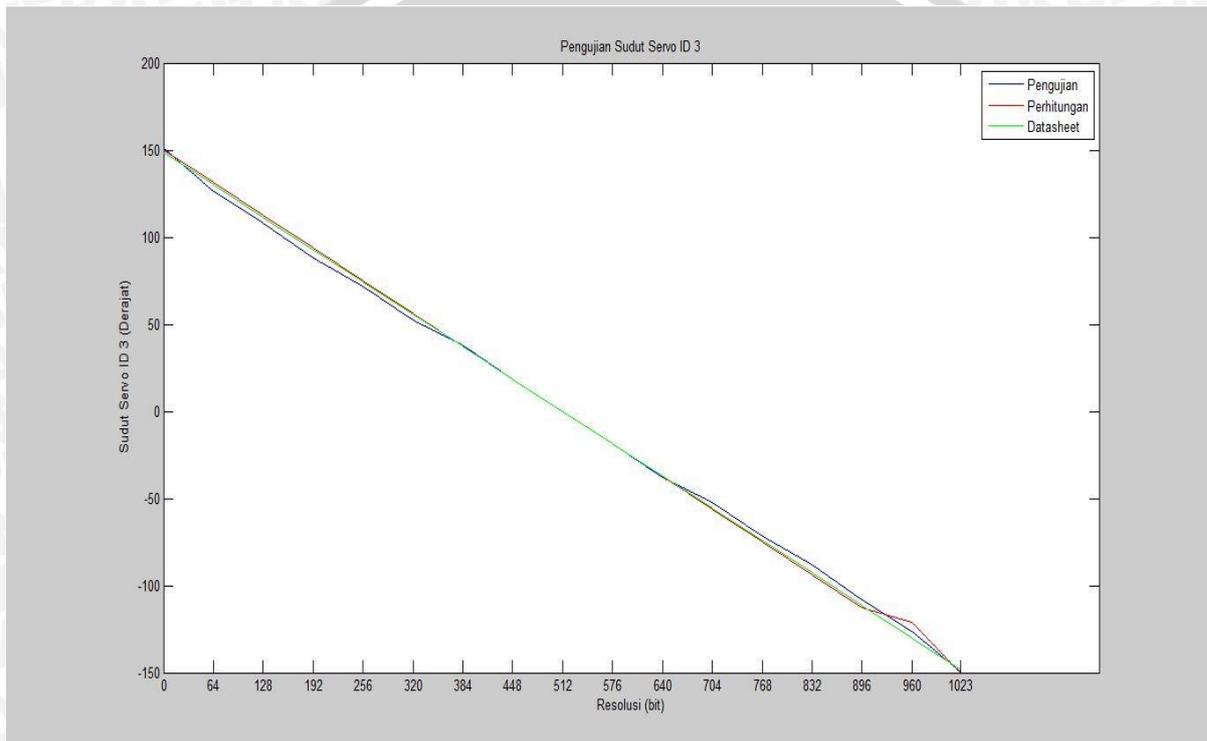


**Tabel 5.3** Pengujian Motor Servo pada ID 3

No	Resolusi	Pengujian Sudut Servo ID 3 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	150.5	150	152	150	150.5	151	150.5	150.75
2	64	125	126	127	126.5	126	128	126	126.3333333
3	128	108	108	108	108	108	108	108	108
4	192	88	88	89	88	89	88	88	88.28571429
5	256	72	72	72	72	72	72	72	72
6	320	52	52	52	52	52	52	52	52
7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	18	19	18.5	18	18.2
9	512	0	0	0	0	0.5	0	0.5	0
10	576	-19	-18.5	-18	-18	-19	-18.5	-18	-18.42857143

11	640	-38	-39	-38	-37	-38.5	-38	-39	-38.21428571
12	704	-53	-51	-52	-52	-52	-52	-52	-52
13	768	-72	-73	-72	-71	-72	-71	-72	-71.85714286
14	832	-88	-88	-89	-88	-89	-88	-88	-88.28571429
15	896	-108	-108.5	-108	-109	-109	-108	-107	-108.2142857
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150	-150	-151	-147	-151	-150	-150.5	-149.8571429

Gambar 5.4 Grafik Sudut Terhadap Resolusi Motor Servo ID 3

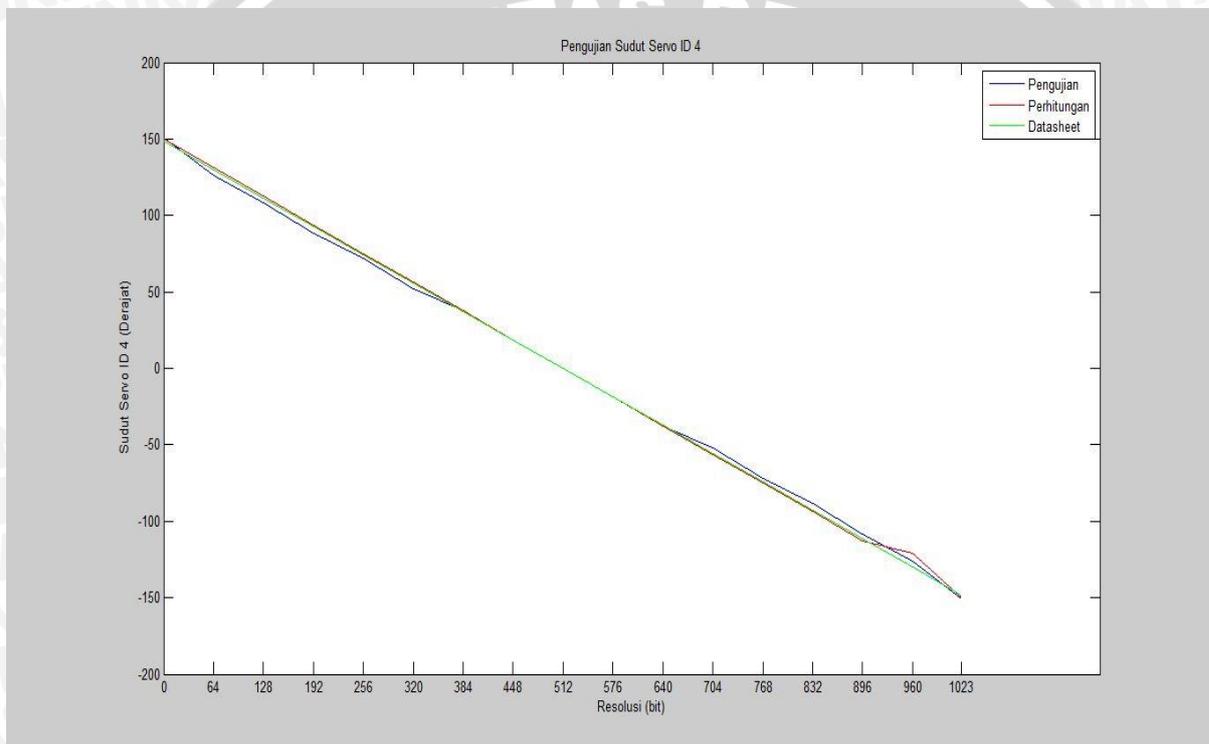


Tabel 5.4 Pengujian Motor Servo pada ID 4

No	Resolusi	Pengujian Sudut Servo ID 4 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	150.5	150	152	150	150.5	151	150.5	150.75
2	64	125	126	127	126.5	126	128	126	126.3333333
3	128	108	108	108	108	108	108	108	108
4	192	88	88	89	88	89	88	88	88.28571429
5	256	72	72	72	72	72	72	72	72
6	320	52	52	52	52	52	52	52	52
7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	18	19	18.5	18	18.2
9	512	0	0	0.5	0	0.5	0	0.5	0

10	576	-18	-18.5	-18	-18	-19	-18.5	-18	-18.28571429
11	640	-38	-38	-38	-38	-38	-38	-38	-38
12	704	-52	-52	-52	-52	-52	-52	-52	-52
13	768	-72	-72	-72	-72	-72	-72	-72	-72
14	832	-88	-88	-89	-88	-89	-88	-88	-88.28571429
15	896	-108	-108	-108	-108	-108	-108	-108	-108
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150.5	-150	-153	-150	-151	-151	-150.5	-150.7142857

Gambar 5.5 Grafik Sudut Terhadap Resolusi Motor Servo ID 4

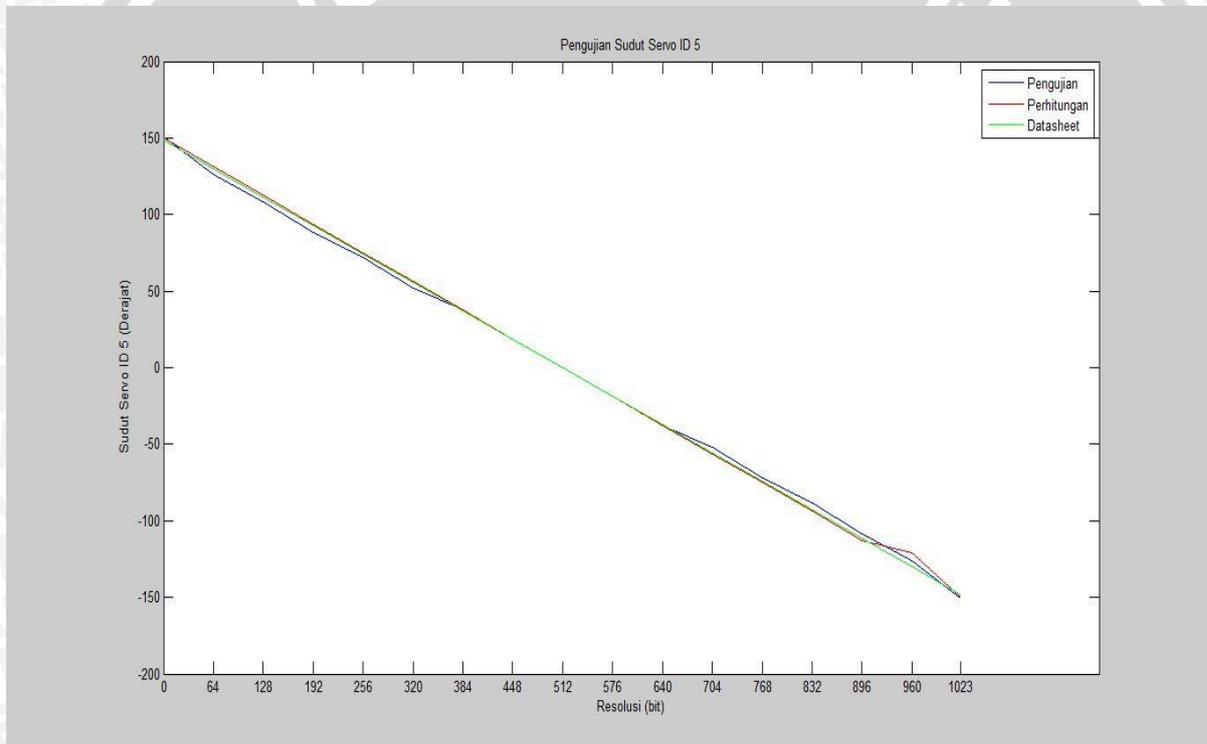


Tabel 5.5 Pengujian Motor Servo pada ID 5

No	Resolusi	Pengujian Sudut Servo ID 5 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	150.5	150	152	150	150.5	151	150.5	150.75
2	64	125	126	127	126.5	126	128	126	126.3333333
3	128	108	108	108	108	108	108	108	108
4	192	88	88	89	88	89	88	88	88.28571429
5	256	72	72	72	72	72	72	72	72
6	320	52	52	52	52	52	52	52	52

7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	18	19	18.5	18	18.2
9	512	0	0	0.5	0	0.5	0	0.5	0
10	576	-18	-18.5	-18	-18	-19	-18.5	-18	-18.28571429
11	640	-38	-38	-38	-38	-38	-38	-38	-38
12	704	-52	-52	-52	-52	-52	-52	-52	-52
13	768	-72	-72	-72	-72	-72	-72	-72	-72
14	832	-88	-88	-89	-88	-89	-88	-88	-88.28571429
15	896	-108	-108	-108	-108	-108	-108	-108	-108
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150.5	-151	-152	-150	-151	-151	-150.5	-150.7857143

Gambar 5.6 Grafik Sudut Terhadap Resolusi Motor Servo ID 5

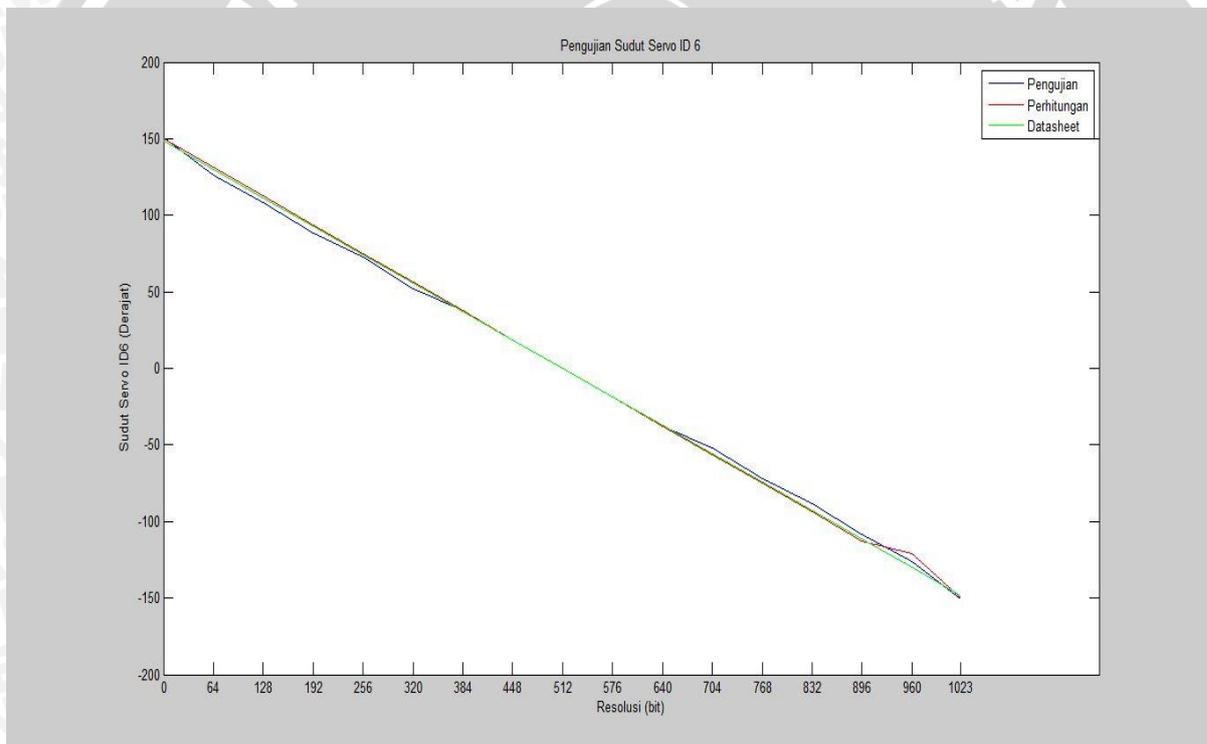


Tabel 5.6 Pengujian Motor Servo pada ID 6

No	Resolusi	Pengujian Sudut Servo ID 6 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	150	150	152	150	150.5	151	150.5	150.6
2	64	125	126	127	126.5	126	128	126	126.3333333
3	128	108	108	108	108	108	108	108	108
4	192	88	88	89	88	89	88	88	88.28571429

5	256	72.5	72	72.5	72	72	72.5	73	72.35714286
6	320	52	52	52	52	52	52	52	52
7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	19	19	18.5	18	18.4
9	512	0	0	0.5	0	0.5	0	0.5	0
10	576	-18	-18.5	-18	-18	-19	-18.5	-18	-18.28571429
11	640	-38	-38	-38	-38	-38	-38	-38	-38
12	704	-52	-52	-52	-52	-52	-52	-52	-52
13	768	-72	-72	-72	-72	-72	-72	-72	-72
14	832	-88	-88	-89	-88	-89	-88	-88	-88.28571429
15	896	-108	-108	-108	-108	-108	-108	-108	-108
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150.5	-150	-152	-150	-151	-151	-150.5	-150.6428571

Gambar 5.7 Grafik Sudut Terhadap Resolusi Motor Servo ID 6

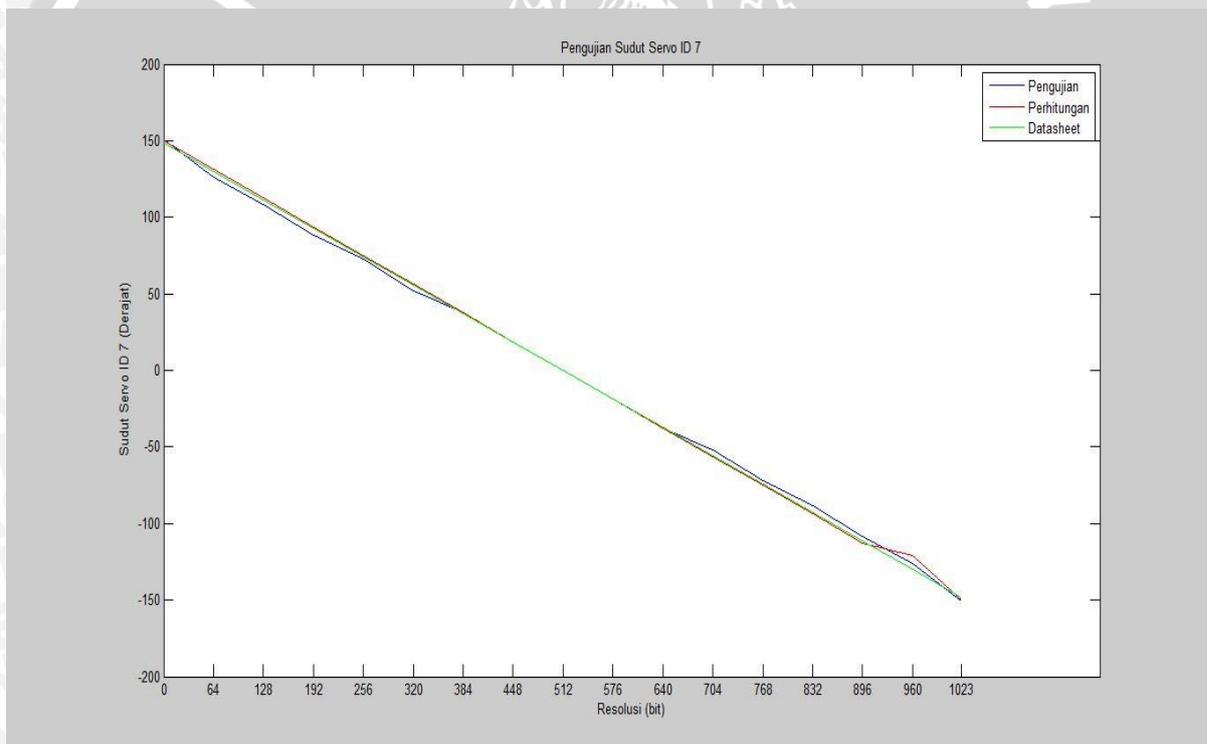


Tabel 5.7 Pengujian Motor Servo pada ID 7

No	Resolusi	Pengujian Sudut Servo ID 7 ( Derajat )							Rata-rata Pengujian
		1	2	3	4	5	6	7	
1	0	150.5	150	152	150	150.5	151	150.5	150.75
2	64	125	126	127	126.5	126	128	126	126.3333333
3	128	110	108	107.5	108	108.5	109	108	108.4285714

4	192	88	88	89	88	89	88	88	88.28571429
5	256	72	72	72.5	72	74	73	72	72.5
6	320	52	52	52	52	52	52	52	52
7	384	38	38	38	38	38	38	38	38
8	448	18	18.5	18	18	19	18.5	18	18.2
9	512	0	0	0.5	0	0.5	0	0.5	0
10	576	-18	-18.5	-18	-18	-19	-18.5	-18	-18.28571429
11	640	-38	-38	-38	-38	-38	-38	-38	-38
12	704	-52	-52	-52	-52	-52	-52	-52	-52
13	768	-72	-72	-72	-72	-72	-72	-72	-72
14	832	-88	-88	-89	-88	-89	-88	-88	-88.28571429
15	896	-108	-108	-108	-108	-108	-108	-108	-108
16	960	-125	-126	-127	-126.5	-126	-128	-126	-126.3571429
17	1023	-150.5	-150	-152	-148	-151	-151	-150.5	-150.3571429

Gambar 5.8 Grafik Sudut Terhadap Resolusi Motor Servo ID 7



Dari hasil pengujian dapat kita lihat rata-rata *error* sudut yang terukur terhadap hasil perhitungan adalah  $\pm 1 - 6$  derajat. Sedangkan *error* sudut yang terukur terhadap hasil kalibrasi *datasheet* adalah  $\pm 0.3 - 6$  derajat. Hasil pengujian ini menunjukkan kelinearitasan servo dari ID 1 sampai dengan ID 7. Dapat dilihat

bahwa motor servo memberikan kelinearitasan yang cukup baik pada resolusi antara 320 sampai 704 bit dan kurang linear pada interval resolusi 0 sampai 256 bit serta 768-1023 bit. Jika diasumsikan bahwa hasil kalibrasi *datasheet* adalah hasil yang paling mendekati nilai sesungguhnya dan ada simpangan tertentu antara hasil perhitungan manual dengan nilai hasil kalibrasi *datasheet*, maka ada nilai pembobot tertentu pada hasil kalibrasi yang menyebabkan adanya perbedaan nilai diantara keduanya.

## 5.2 Pengujian *Forward Kinematic*

### a. Tujuan

Untuk mengetahui ketelitian posisi *end-effector* dengan metode *forward kinematic*.

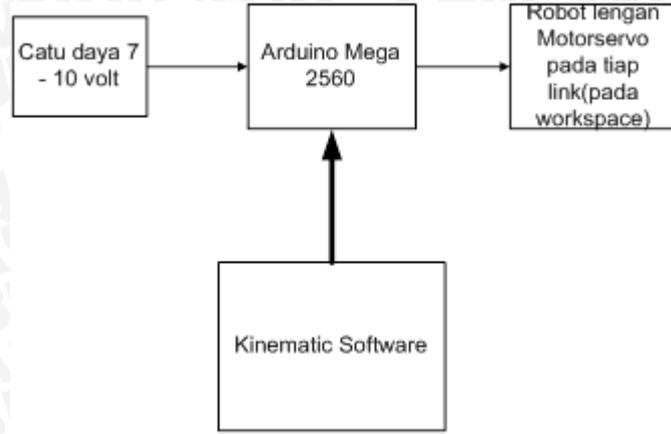
### b. Peralatan yang digunakan

- Robot lengan Smart Arm Robotic AX-12A
- *Arduino Mega 2560*
- Catu daya 7 – 10 volt
- Kabel komunikasi serial *USB Type A to B* Komputer/laptop
- Penggaris siku

### c. Medan robot lengan

### c. Langkah pengujian

1. Rangkai peralatan seperti dalam Gambar 5.9.
2. *Kinematic Software* dinyalakan dan diberikan nilai sudut yang diinginkan pada masing-masing *link*.
3. Tombol *forward* ditekan, perhitungan secara teoritis akan diperlihatkan oleh *software*
4. Ukur *end-effector* posisi dengan dua buah penggaris siku dan papan penunjuk posisi



Gambar 5.9 Blok Pengujian *Forward Kinematic*

d. Hasil pengujian

Hasil pengujian *forward kinematic* ditunjukkan dalam Tabel 5.8

Tabel 5.8 Pengujian *Forward Kinematic*

No	Sudut (derajat)		Posisi Pengukuran		Posisi Teori		Error	
	Link 1	Link 2	Px (cm)	Py (cm)	Px (cm)	Py (cm)	Px (cm)	Py (cm)
1	15	24	34.3	19	34.2951	18.87435	-0.0049	-0.12565
			34.3	19			-0.0049	-0.12565
			34.3	19			-0.0049	-0.12565
			34.2	18.5			0.0951	0.37435
			34.3	19			-0.0049	-0.12565
			34.3	19			-0.0049	-0.12565
			34.2	18.5			0.0951	0.37435
2	19	38	28.5	24.75	28.6005	24.82407	0.1005	0.07407
			28.6	24.8			0.0005	0.02407
			28.6	24.8			0.0005	0.02407
			28.6	24.8			0.0005	0.02407
			28.5	24.75			0.1005	0.07407
			28.6	24.8			0.0005	0.02407
			28.6	24.8			0.0005	0.02407
3	-25	-20	31.6	-23.4	31.6707	-23.448	0.07068	-0.04796
			31.65	-23.45			0.02068	0.00204
			31.6	-23.45			0.07068	0.00204
			31.6	-23.45			0.07068	0.00204
			31.7	-23.5			-0.02932	0.05204
			31.7	-23.5			-0.02932	0.05204
			31.7	-23.5			-0.02932	0.05204



			26	-27			-0.30501	0.34058
			26	-27			-0.30501	0.34058
			25.5	-26.5			0.19499	-0.15942
			25.7	-26.6			-0.00501	-0.05942
			25.7	-27.6			-0.00501	0.94058
			25.7	-27.6			-0.00501	0.94058
<b>4</b>	-20	-45	25.7	-27.6	25.695	-26.659	-0.00501	0.94058

Dari hasil pengujian dapat kita lihat rata-rata *error* posisi pada sumbu x yaitu sebesar  $\pm 0.05608$  cm, sedangkan rata-rata *error* yang terjadi pada sumbu y yaitu  $\pm 0.19919$  cm. *Error* yang terjadi diakibatkan oleh mekanik robot lengan pada konstruksi pasangan servo pada joint 2 dan 3 yang kurang selaras sehingga *error* yang ditimbulkan dapat mempengaruhi ketepatan posisi akhir pengukuran.

Contoh perhitungan tabel 5.4 (lihat tabel 4.1 dan persamaan 4.8):

$$a_1 = 17a_2 = 23$$

$$\theta_1 = 15^\circ$$

$$\theta_2 = 24^\circ$$

$$P_x = (a_2 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2))$$

$$P_y = (a_2 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2))$$

$$P_x = (17 \cos 15^\circ + 23 \cos(15^\circ + 24^\circ)) = 16.42074 + 17.87436 = 34.2951$$

$$P_y = (17 \sin 15^\circ + 23 \sin(15^\circ + 24^\circ)) = 4.39999 + 14.47436 = 18.87435$$

Untuk selanjutnya, nilai perhitungan secara teoritis dihitung dengan program sederhana menggunakan bantuan MATLAB.

### 5.3 Pengujian *Invers Kinematic*

a. Tujuan

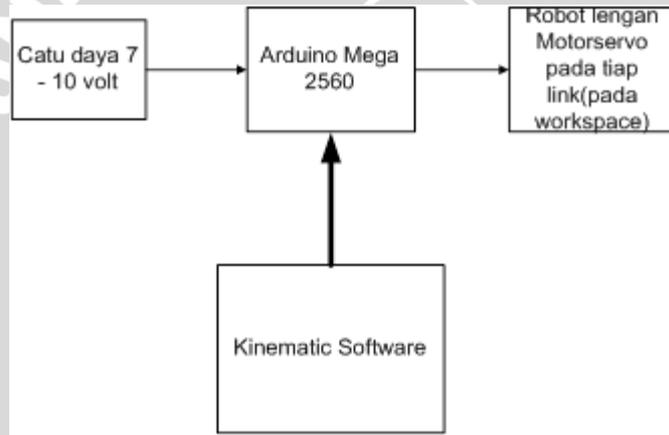
Untuk mengetahui ketelitian posisi *end-effector* dengan metode *forward kinematic*.

b. Peralatan yang digunakan

- Robot lengan Smart Arm Robotic AX-12A
- *Arduino Mega 2560*
- Catu daya 7 – 10 volt



- Kabel komunikasi serial *USB Type A to B* Komputer/laptop
  - Penggaris siku
  - Medan robot lengan
- c. Langkah pengujian
1. Rangkai peralatan seperti dalam Gambar 5.10.
  2. *Kinematic Software* dinyalakan dan diberikan nilai sudut yang diinginkan pada masing-masing *link*.
  3. Tombol *invers* ditekan, perhitungan secara teoritis akan diperlihatkan oleh *software*
  4. Ukur *end-effector* posisi dengan dua buah penggaris siku dan papan penunjuk posisi



Gambar 5.10 Blok Pengujian *Invers Kinematic*

d. Hasil pengujian

Pada Tabel 5.9 Menunjukkan hasil pengujian *invers kinematic*

Tabel 5.9 Pengujian *Invers Kinematic*

No	Posisi Tujuan		Solusi Sudut		Posisi terukur		Error	
	Px (cm)	Py (cm)	Link 1	Link 2	Px (cm)	Py (cm)	Px (cm)	Py (cm)
1	34.2	18.8	15.02624	23.99952	34.3	18.8	-0.3	0.2
					34.1	18.7	-0.1	0.3
					34.3	18.8	-0.3	0.2
					34.3	18.8	-0.3	0.2
					34.3	18.8	-0.3	0.2

					34.2	18.7	-0.2	0.3
					34.2	18.7	-0.2	0.3
2	28.6	25	18.99987	38.62078	28.6	24.8	0	0.2
					28.6	24.8	0	0.2
					28.6	24.8	0	0.2
					28.5	25	0.1	0
					28.5	25	0.1	0
					28.6	24.8	0	0.2
					28.5	25	0.1	0
3	31.7	-23	-25	-20	31.6	-23.2	0.1	0.2
					31.7	-23.4	0	0.4
					31.7	-23.4	0	0.4
					31.7	-23.4	0	0.4
					31.7	-23.4	0	0.4
					31.7	-23.4	0	0.4
					31.6	-23.2	0.1	0.2
4	25.7	-26.7	-20	-45	25.7	-26.6	0	0
					25.7	-26.6	0	0
					25.7	-26.6	0	0
					25.7	-26.6	0	0
					25.7	-26.6	0	0
					25.7	-26.6	0	0
					25.7	-26.6	0	0

Dari hasil pengujian dapat di lihat bahwa *error* posisi pada sumbu x yaitu sebesar  $\pm 0.08928$  cm sedangkan *error* yang terjadi pada sumbu y yaitu sebesar  $\pm 0.175$  cm. *Error* yang terjadi diakibatkan oleh mekanik robot lengan pada konstruksi pasangan servo pada joint 2 dan 3 yang kurang selaras sehingga *error* yang ditimbulkan dapat mempengaruhi ketepatan posisi akhir pengukuran. Selain itu solusi sudut *iners* yang kurang tepat juga dapat memberikan *error*.

Contoh perhitungan tabel 5.5 (lihat sub bab 4.2.2.2)

Untuk solusi sudut servo Link I

$$\tan \alpha_1 = \frac{Y_0}{X_0}$$

$$\tan \alpha_1 = \frac{18.8}{34.2}$$

$$\tan \alpha_1 = 0.55036$$

$$\alpha_1 = 28.82624^{\circ}$$

$$C = \sqrt{Y_0^2 + X_0^2}$$

$$C = \sqrt{18.8^2 + 34.2^2}$$

$$C = \sqrt{1529.93} = 39.11431$$

$$\phi_1 = \arccos \frac{(a^2 + c^2 - b^2)}{2 \cdot a \cdot c}$$

$$\phi_1 = \arccos \frac{(17^2 + 39.11^2 - 23^2)}{2 \cdot 17 \cdot 39.11}$$

$$\phi_1 = \arccos \left( \frac{1289.5921}{1329.74} \right)$$

$$\phi_1 = \arccos 0.971026$$

$$\phi_1 = 13.8^{\circ}$$

$\alpha_1$  (+), maka

$$\theta_1 = \alpha_1 - \phi_1 = 28.82624^{\circ} - 13.8^{\circ}$$

$$\theta_1 = 15.02624^{\circ} \approx 15^{\circ}$$

Untuk solusi sudut servo Link II

$$\cos \alpha_2 = \frac{(a^2 + b^2 - c^2)}{2 \cdot a \cdot b}$$

$$\alpha_2 = \arccos \frac{(a^2 + b^2 - c^2)}{2 \cdot a \cdot b}$$

$$\alpha_2 = \arccos \frac{(17^2 + 23^2 - 39.11^2)}{2 \cdot 17 \cdot 23}$$

$$\alpha_2 = \arccos \left( \frac{-711.5921}{782} \right)$$

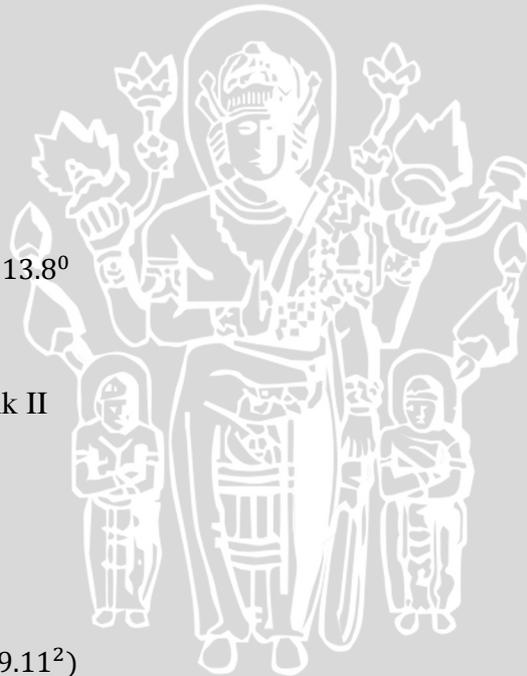
$$\alpha_2 = \arccos(-0.91354)$$

$$\alpha_2 = 156.00048^{\circ}$$

$\alpha_1$  (+), maka

$$\theta_2 = 180^{\circ} - \alpha_2 = 180^{\circ} - 156.00048^{\circ}$$

$$\theta_2 = 23.99952^{\circ} \approx 24^{\circ}$$



Untuk selanjutnya, nilai perhitungan secara teoritis dihitung dengan program sederhana menggunakan bantuan MATLAB.

#### 5.4 Pengujian *Stall Torque* Pada Motor Servo

##### a. Tujuan

Mengetahui nilai torsi rata-rata pada masing-masing motor servo yang digunakan saat bekerja tanpa pembebanan.

##### b. Peralatan yang digunakan

- Mur sepanjang 2 cm berat 7 mg.
- Motor servo Dynamixel AX-12A ID 1 – ID 7
- Board Arduino Mega 2560
- Komputer/laptop.

##### c. Langkah pengujian

- Catu daya 8 volt
- Torsimeter Lutron TQ-8800
- Software Servo Calibrator
- Peralatan dirangkai seperti pada Gambar 5.7.
- Berikan masukan bit dari 0 menuju 1023 bit pada *software Servo Calibrator*.
- Busur derajat digunakan untuk mengukur sudut *link*.
- Data keluaran torsi dari torsimeter dicatat dan di tabulasikan
- Agar data pengukuran mendekati keadaan yang sebenarnya langkah 2, 3 dan 4 dilakukan sebanyak dua puluh kali untuk masing-masing ID .



**Gambar 5.11** Blok Pengujian *Stall Torque* Motor Servo

##### d. Hasil Pengujian

Pada Tabel 5.10 Menunjukkan hasil pengujian *Stall Torque* Motor Servo

**Tabel 5.6** Pengujian *Stall Torque* Pada Motor Servo



Dari hasil pengujian dapat kita lihat rata-rata *stall torque* untuk ID 1 – ID 7 yang terukur oleh torsimeter memberikan hasil yang hampir sama yaitu  $\pm 11.8$  kgf.cm. Selain itu dalam dua puluh kali pengujian untuk tiap ID, dapat dilihat bahwa LED dalam kondisi tidak menyala. Hal ini membuktikan bahwa setiap motor servo yang di uji tidak mengalami over position, over current, over voltage, over torque, maupun over temperature. Akan tetapi perlu diingat bahwa sebelum melakukan pengujian *stall torque* motor harus dipastikan bahwa tegangan dari power supply harus dalam rentang antara 7 sampai 8 volt. Karena menurut *datasheet*, nilai *stall torque* pada motor servo tidak hanya dipengaruhi oleh bobot motor tanpa beban dan panjang poros motor, akan tetapi juga dipengaruhi oleh temperatur dan besarnya catu daya yang diberikan.

### 5.5 Pengujian Torsi Pembebanan Pada Motor Servo

#### a. Tujuan

Mengetahui besarnya torsi maksimum yang dapat diberikan oleh motor servo terhadap perlakuan beban yang berbeda-beda.

#### b. Peralatan yang digunakan

- Mur sepanjang 2 cm berat 7 mg.
- Beban neraca seberat 4x100 gr dan 1x500gr
- Robot lengan lengkap dengan motor servo pada masing-masing *link*.
- Board Arduino Mega 2560
- Komputer/laptop.
- Catu daya 8 volt
- Torsimeter Lutron TQ-8800
- Software Servo Calibrator

#### c. Langkah pengujian

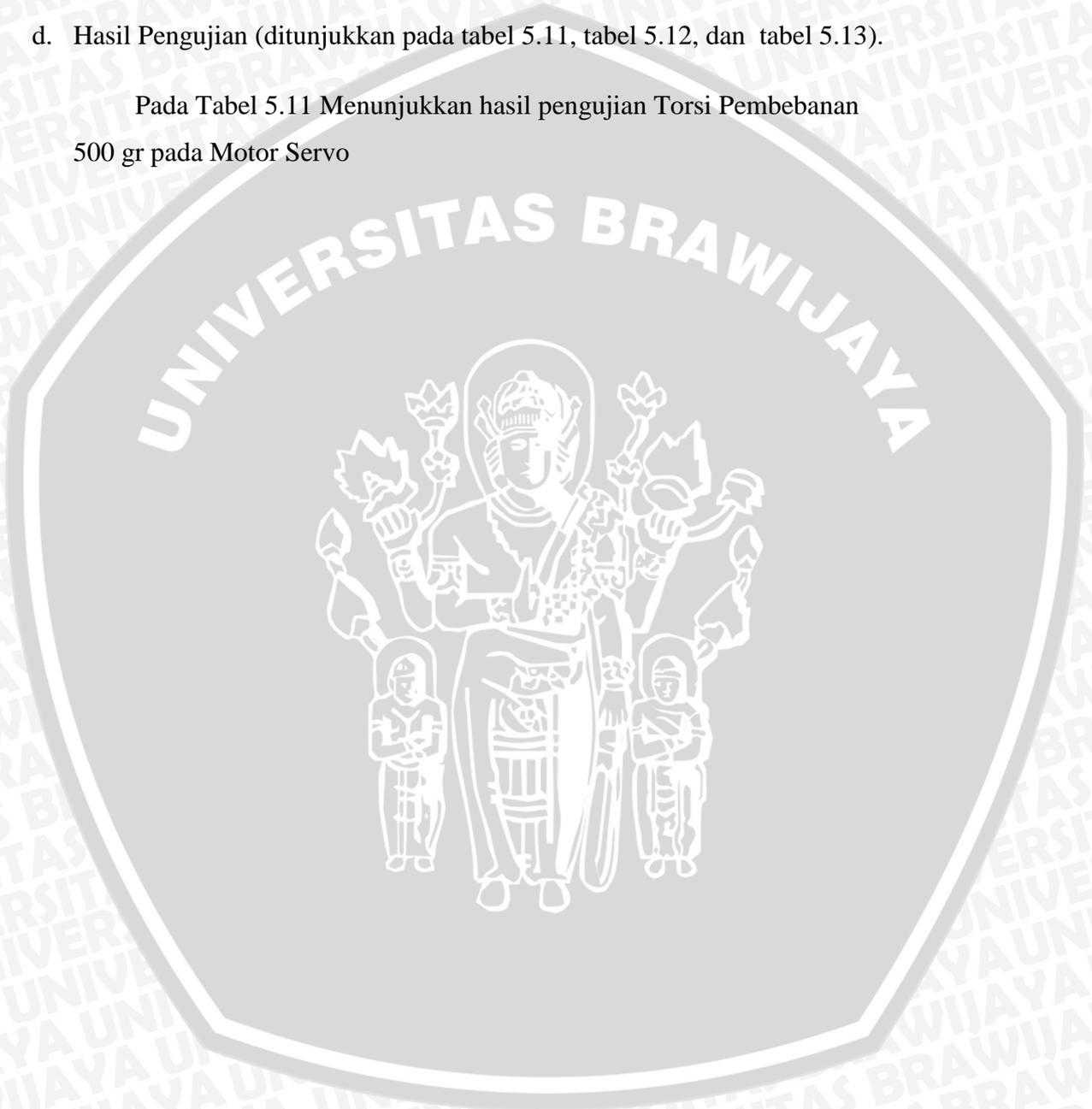
- Peralatan dirangkai seperti pada Gambar 5.6.
- Berikan masukan bit mulai 0 – 1023 bit pada *software Servo Calibrator*.
- Busur derajat digunakan untuk mengukur sudut *link*.
- Data keluaran torsi dari torsimeter dicatat dan di tabulasikan.
- Agar data pengukuran mendekati keadaan yang sebenarnya langkah 2, 3 dan 4 dilakukan sebanyak sepuluh kali untuk masing-masing ID .



**Gambar 5.12** Blok Pengujian Torsi Pembebanan Pada Motor Servo

d. Hasil Pengujian (ditunjukkan pada tabel 5.11, tabel 5.12, dan tabel 5.13).

Pada Tabel 5.11 Menunjukkan hasil pengujian Torsi Pembebanan 500 gr pada Motor Servo





Dari hasil pengujian dapat kita lihat rata-rata torsi pembebanan 500 gr untuk ID 1 – ID 7 yang terukur oleh torsimeter memberikan hasil yang hampir sama yaitu  $\pm 9.8$  kgf.cm. Selain itu dalam sepuluh kali pengujian untuk tiap ID, dapat dilihat bahwa LED dalam kondisi tidak menyala. Hal ini membuktikan bahwa setiap motor servo yang di uji tidak mengalami *over position*, *over current*, *over voltage*, *over torque*, maupun *over temperature*. Seperti yang dilakukan pada pengujian stall torque sebelum melakukan pengujian torsi pembebanan motor harus dipastikan bahwa tegangan dari power supply harus dalam rentang antara 7.7 sampai 8 volt. Karena menurut *datasheet*, nilai torsi pembebanan pada motor servo tidak hanya dipengaruhi oleh bobot motor tanpa beban dan panjang poros motor, akan tetapi juga dipengaruhi oleh temperatur dan besarnya catu daya yang diberikan.

Pada Tabel 5.12 Menunjukkan hasil pengujian Torsi Pembebanan 700 gr pada Motor Servo





Dari hasil pengujian dapat kita lihat rata-rata torsi pembebanan 700gr untuk ID 1 – ID 7 yang terukur oleh torsimeter memberikan hasil yang hampir sama yaitu  $\pm 17.88$  kgf.cm. Akan tetapi dalam sepuluh kali pengujian untuk tiap ID, dapat dilihat bahwa ada beberapa data yang memperlihatkan kondisi LED yang menyala. Hal ini mengindikasikan bahwa motor servo yang di uji sedang mengalami salah satu dari kondisi *over position*, *over current*, *over voltage*, *over torque*, maupun *over temperature*. Akan tetapi karena tidak adanya modul untuk membaca kondisi motor servo maka penyebab dari LED menyala tidak dapat diketahui secara pasti.

Pada Tabel 5.13 Menunjukkan hasil pengujian Torsi Pembebanan 900 gr pada Motor Servo





Dari hasil pengujian dapat kita lihat rata-rata torsi pembebanan 900 gr untuk ID 1 – ID 7 yang terukur oleh torsimeter memberikan hasil yang hampir sama yaitu  $\pm 17.87$  kgf.cm. Selain itu dalam sepuluh kali pengujian untuk tiap ID, dapat dilihat bahwa LED dalam kondisi menyala. Hal ini mengindikasikan bahwa motor servo yang di uji sedang mengalami salah satu dari kondisi *over position*, *over current*, *over voltage*, *over torque*, maupun *over temperature*. Akan tetapi karena tidak adanya modul untuk membaca kondisi motor servo maka penyebab dari LED menyala tidak dapat diketahui secara pasti.

Motor servo Dynamixel sendiri sebenarnya tergolong motor servo cerdas. Hal ini disebabkan motor servo ini mampu memberikan *feedback* kepada *user* mengenai torsi, temperatur, konsumsi arus dan tegangan serta posisi yang dituju ( dalam resolusi / bit). *Library-library* khusus Arduino untuk menjalankan fungsi pembacaan informasi mengenai torsi, temperatur, konsumsi arus dan tegangan serta posisi yang dituju ( dalam resolusi / bit)-pun sebenarnya sudah dapat diunduh secara *free* melalui *website* resmi perusahaan yang memproduksinya. Akan tetapi untuk membaca *feedback* tersebut, perusahaan yang memproduksi motor servo ini menjual modul tambahan yang harus digunakan bersama pada rangkaian kendali elektriknya. Modul-modul ini dipasaran dikenal sebagai modul OPEN CM 9.04, OPEN-Rob, CM- 700, CM-500 dan sejenisnya.

Penggunaan modul tambahan seperti yang disebutkan sebelumnya bertujuan tidak lain adalah untuk memperkuat performa *master-controller* ( dalam skripsi ini Arduino Mega 2560 ), menyamakan baudrate, mengubah protokol sistem komunikasi *full duplex* menjadi *half duplex* serta membaca *feedback* dari motor servo *Dynamixel series*. Namun sayangnya selama pengerjaan skripsi ini penulis sangat kesulitan mendapatkan modul tersebut. Sehingga dalam pengerjaan skripsi ini, sistem pengendalian yang dijalankan sejauh ini hanya dapat mengirimkan data untuk diproses agar dapat menggerakkan aktuator robot dan belum bisa membaca kondisi aktuator. Kondisi-kondisi yang dimaksud termasuk informasi mengenai torsi, temperatur, konsumsi arus dan tegangan serta posisi yang dituju ( dalam resolusi / bit).

Selama pengambilan data hasil pengujian torsi pembebanan dan pengambilan data hasil pengujian *invers kinematic* serta *forward kinematic* , LED



Dari hasil pembacaan tersebut dapat diketahui bahwa program yang dijalankan sudah dalam algoritma yang benar namun dalam prosedur yang salah. Prosedur yang salah dalam hal ini adalah program tetap dijalankan meskipun tidak tersedianya modul tambahan seperti OPEN CM 9.04, OPEN-Rob, CM- 700, CM-500 dan sejenisnya. Akibatnya yang terbaca pada serial monitor hanya berupa *dummy data* yaitu angka -1 atau -2.



## BAB VI PENUTUP

### 6.1 Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada robot lengan Smart Arm Robotic AX-12A maka dapat diambil kesimpulan sebagai berikut:

1. Pengendalian enam motor servo menggunakan metode *forward* dan *invers kinematic*, haruslah mempunyai linieritas yang baik antara sudut yang terbentuk dan sudut yang diberikan. Berdasarkan data pengujian motor servo tiap ID, motor servo membentuk sudut dengan penyimpangan antara 0.5 – 2 derajat .
2. Perancangan sistem antarmuka Qt untuk komunikasi antara robot lengan (Arduino Mega 2560) dengan PC sementara ini hanya dapat digunakan untuk mengirim nilai sudut dan posisi karena adanya keterbatasan baudrate Arduino Mega 2560 untuk dapat membaca kondisi servo Dynamixel AX-12A.
3. Hasil penerapan metode *forward* dan *invers kinematic* pada robot lengan. Pada pengujian *invers kinematic* dapat dilihat rata-rata error posisi pada sumbu x yaitu sebesar  $\pm 0.10476$  cm dan pada sumbu y yaitu  $\pm 0.23333$  cm. Pada pengujian *forward kinematic* dapat dilihat rata-rata error posisi pada sumbu x yaitu sebesar  $\pm 0.00282$  cm dan pada sumbu y yaitu  $\pm 0.13526$  cm.
4. Penerapan metode *forward* dan *invers kinematic* pada robot lengan yang digunakan untuk memindahkan benda dari satu tempat ke tempat yang berbeda, memiliki kelemahan *error* posisi karena konfigurasi mekanik robot sehingga memerlukan penyesuaian posisi agar benda dapat dipindahkan serta adanya faktor konversi bit dalam sudut yang kurang sesuai.

## 6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan:

1. Menggunakan perangkat pembantu lain (contohnya OpenCM 9.04) untuk menyesuaikan *baudrate* pada servo dan Arduino Mega 2560 agar dapat membaca suhu, tegangan, dan posisi (dalam resolusi / bit) masing-masing servo.
2. Mengembangkan dengan solusi *invers dynamic* agar memiliki ketepatan yang lebih baik dari pada solusi *invers geometry* serta dapat mengetahui hal-hal yang berkaitan dengan torsi, distribusi massa dan percepatan yang dialami robot.
3. Skema pengendalian yang lebih baik dengan mengkombinasikan *invers* dan *forward kinematic* dengan *feedback* pada tiap-tiap *link* untuk mengukur posisi, torsi, suhu, tegangan dan lain sebagainya.
4. Membuat antarmuka dengan desain grafis yang lebih interaktif, misalnya dalam bentuk gambar tiga dimensi dengan *trackmouse input*.