

## BAB II LANDASAN TEORI

### 2.1 Robotika

Robotika merupakan bidang ilmu yang mempelajari perancangan, pembuatan, pengendalian, dan penerapan robot. Bidang ini menggabungkan beberapa disiplin seperti mekanika, elektronika, kontrol, komputer, dan kecerdasan buatan. Robot dapat didefinisikan sebagai sistem mekanik yang mampu melakukan tugas tertentu secara otomatis atau semiotomatis berdasarkan instruksi yang diberikan. Dalam konteks pendidikan, robotika sering digunakan sebagai sarana untuk memahami konsep sensor, aktuator, algoritma, dan interaksi antara perangkat keras dan perangkat lunak. Craig (2005) dalam bukunya "*Introduction to Robotics: Mechanics and Control*" menjelaskan bahwa robotika modern membutuhkan pemahaman analitis agar gerakan robot dapat dijelaskan, diprediksi, dan dikembangkan [11].

Robot lengan atau manipulator adalah jenis robot yang menyerupai lengan manusia dalam fungsi gerakannya. Robot ini memiliki *Link* dan *joint* yang saling terhubung. *Link* adalah bagian kaku yang membentuk struktur lengan, sedangkan *joint* adalah sambungan yang memungkinkan gerakan relatif antar *Link*. Pada sistem sederhana, *joint* dapat digerakkan oleh motor *stepper*. Kombinasi gerakan *joint* akan menghasilkan posisi *end-effector* tertentu. *End-effector* adalah bagian paling ujung dari robot yang berinteraksi dengan objek atau lingkungan. Siciliano et al. (2009) dalam "*Robotics: Modelling, Planning and Control*" mengembangkan pemodelan kinematika robot planar dan spatial secara sistematis menggunakan parameter Denavit-Hartenberg [12].

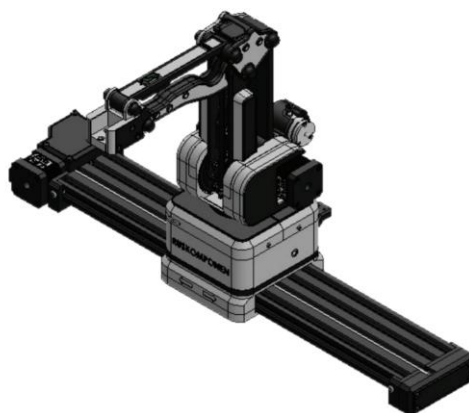
Dalam robotika, pemahaman posisi dan orientasi menjadi sangat penting. Robot harus mengetahui posisi bagian ujungnya agar dapat menjalankan tugas secara akurat. Untuk robot lengan, posisi *end-effector* ditentukan oleh konfigurasi sendi. Jika robot memiliki lebih banyak sendi, maka perhitungan menjadi semakin kompleks. Namun prinsip dasarnya tetap sama, yaitu memodelkan hubungan antara parameter sendi dan posisi geometris robot. Niku (2011) dalam

"*Introduction to Robotics: Analysis, Control, Applications*" mengimplementasikan *forward kinematics* pada robot planar 2 DOF dan 3 DOF menggunakan metode trigonometri sederhana [13].

Metode Denavit-Hartenberg (DH) yang dikembangkan oleh Jacques Denavit dan Richard Hartenberg pada tahun 1955 menjadi standar untuk mendefinisikan kerangka koordinat pada setiap *joint* robot manipulator. Jazar (2010) dalam "*Theory of Applied Robotics: Kinematics, Dynamics, and Control*" membahas kinematika *forward* robot manipulator menggunakan metode DH dan matriks transformasi, serta menjelaskan hubungan antara metode DH dan persamaan trigonometri sederhana untuk robot planar [14].

Penelitian mengenai visualisasi *forward kinematics* telah dilakukan oleh Corke (2017) dalam "*Robotics, Vision and Control: Fundamental Algorithms in MATLAB*" yang mengintegrasikan *forward kinematics* dengan visualisasi 3D menggunakan MATLAB untuk robot manipulator 6 DOF. Penelitian ini membuktikan bahwa visualisasi 3D membantu pemahaman hubungan antara sudut sendi dan posisi *end-effector* secara intuitif. Lynch dan Park (2017) dalam "*Modern Robotics: Mechanics, Planning, and Control*" mengembangkan metode *forward kinematics* menggunakan produk eksponensial sebagai alternatif metode selain DH parameter [15].

## 2.2 Robot Arm RNV3



Gambar 2.1 Robot RNV3

RNV3 merupakan platform lengan robotik berbasis sumber terbuka (*open-source*) yang mengintegrasikan komunikasi serial dengan antarmuka desktop untuk memberikan fleksibilitas pengembangan yang tinggi bagi penggunanya. Robot ini beroperasi menggunakan sistem koordinat Kartesian yang sangat intuitif, sehingga pengguna, baik pemula maupun profesional, dapat dengan mudah mengontrol pergerakan hanya dengan memasukkan koordinat titik tujuan pada antarmuka perangkat lunak. Sistem pengendaliannya dioptimalkan melalui integrasi bahasa pemrograman Python sebagai bahasa tingkat tinggi untuk menyusun skrip kontrol yang cerdas, efisien, dan mampu mengotomatiskan tugas-tugas berulang (*repetitive tasks*) secara presisi [16].

Antarmuka khusus tersebut juga memfasilitasi pemantauan koneksi serial secara *real-time* guna menjamin keakuratan transmisi data antara komputer dan perangkat keras robot. Dengan kemampuan adaptasi yang luar biasa, RNV3 sangat ideal diimplementasikan dalam berbagai skenario kerja seperti pemrosesan *pick and place*, inspeksi visual, hingga kebutuhan riset di laboratorium dan institusi pendidikan guna meningkatkan produktivitas serta efisiensi operasional secara keseluruhan. Quigley et al. (2015) dalam "*Programming Robots with ROS*" mengimplementasikan *forward kinematics* pada robot berbasis ROS dan menunjukkan bahwa *software framework* dapat mempermudah pengembangan sistem robotika [16].

## **2.3 Komponen Utama dan Elektronika**

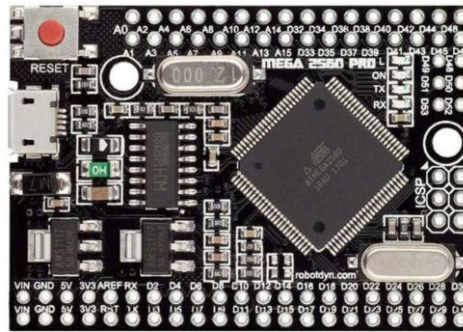
Sistem robot lengan 3 DOF terdiri dari beberapa komponen elektronika utama yang saling terintegrasi. Setiap komponen memiliki fungsi spesifik dalam mendukung operasi robot, mulai dari pengendalian motor, deteksi posisi, pengaturan daya, hingga antarmuka dengan pengguna. Berikut adalah penjelasan masing-masing komponen:

### **2.3.1 Arduino**

Arduino adalah platform mikrokontroler yang banyak digunakan untuk prototipe sistem elektronika dan robotika. Arduino menyediakan lingkungan pemrograman yang sederhana, pustaka pendukung, serta antarmuka input-output

digital dan analog. Dalam implementasi robot lengan, Arduino digunakan untuk mengendalikan *Stepper* dan melakukan perhitungan *forward kinematics*. Komunikasi serial digunakan agar pengguna dapat memasukkan nilai sudut dan membaca hasil koordinat melalui serial monitor [17].

Program Arduino pada sistem ini memuat pustaka *AccelStepper/Stepper* dan *math*. Pustaka *AccelStepper/Stepper* digunakan untuk mengirim pulsa step dan arah ke driver motor *Stepper*, sedangkan pustaka *math* digunakan untuk fungsi trigonometri seperti *cos* dan *sin*. Nilai sudut dalam derajat dikonversi menjadi radian karena fungsi trigonometri pada bahasa C/C++ menggunakan satuan radian.



Gambar 2.2 Arduino Mega 2560 Mini Pro

Komponen ini bertindak sebagai "otak" atau unit pemroses pusat dari seluruh sistem robot. Arduino Mega 2560 Mini Pro dipilih karena memiliki jumlah pin I/O yang sangat banyak, memungkinkannya untuk mengendalikan banyak motor *Stepper*, membaca input dari berbagai sensor endstop, serta mengelola komunikasi serial dengan komputer secara simultan.

Tabel 2.1 Spesifikasi Arduino Mega 2560 Mini Pro

Parameter	Spesifikasi
Tegangan Operasi	5V DC
Tegangan Input (Vin)	7V - 12V DC (Disuplai melalui jalur

	adaptor robot)
<b>Digital I/O Pins</b>	54 pin (15 pin mendukung output PWM)
<b>Analog Input Pins</b>	16 pin
<b>Arus DC per I/O Pin</b>	20 mA
<b>Flash Memory</b>	256 KB (8 KB digunakan untuk bootloader)
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Clock Speed</b>	16 MHz
<b>Komunikasi Serial</b>	4 Hardware UART (TX/RX)

### 2.3.2 Motor *Stepper*

Motor *Stepper* adalah aktuator yang mampu bergerak menuju sudut tertentu berdasarkan sinyal kendali. *Stepper* banyak digunakan pada robot kecil karena mudah dikendalikan, memiliki rangkaian kontrol internal, dan dapat langsung menerima perintah sudut dari mikrokontroler. *Stepper* standar umumnya bergerak pada rentang sudut kerja mekanik yang ditentukan saat kalibrasi. Pada program robot, nilai sudut dikirim melalui pulsa step dan sinyal direction sehingga poros *Stepper* bergerak menuju posisi yang diminta.

Kelebihan *Stepper* adalah kemudahan implementasi dan kestabilan posisi untuk aplikasi ringan. Namun *Stepper* juga memiliki keterbatasan seperti torsi terbatas, akurasi bergantung pada kualitas mekanik, *backlash* pada *gearbox*, serta kemungkinan selisih antara sudut perintah dan sudut aktual. Dalam penelitian *forward kinematics*, sudut yang digunakan adalah sudut perintah. Oleh karena itu, hasil perhitungan bersifat teoritis berdasarkan asumsi bahwa *Stepper* mencapai sudut yang diperintahkan [18].



Gambar 2.3 Motor *Stepper* Nema17  
17HS4401

Robot RNV3 menggunakan motor *Stepper* sebagai penggerak presisi pada setiap sumbu geraknya. Berbeda dengan motor DC biasa, motor *Stepper* memungkinkan kontrol posisi yang sangat akurat tanpa memerlukan *feedback* eksternal yang rumit, karena bergerak dalam satuan langkah (*step*) yang diskrit.

Secara keseluruhan, NEMA 17 adalah pilihan yang paling tepat untuk duplikasi robot ini karena keseimbangan antara ukuran yang kompak dan torsi yang dihasilkan cukup kuat untuk menggerakkan lengan sepanjang 140mm.

Tabel 2.2 Spesifikasi Motor *Stepper* Nema17 17HS4401

Parameter	Spesifikasi
<b>Type Motor</b>	Bipolar <i>Stepper</i>
<b>Sudut Langkah (<i>Step Angle</i>)</b>	1.8° per <i>step</i> (200 <i>steps</i> per putaran penuh)
<b>Arus Rating (<i>Rated Current</i>)</b>	1.5 A per fase
<b>Torsi Penahan (<i>Holding Torque</i>)</b>	400 mN.m
<b>Resistansi Fase (<i>Phase Resistance</i>)</b>	1.5 Ohm
<b>Induktansi Fase (<i>Phase Inductance</i>)</b>	2.8 mH

<b>Voltase Rating (<i>Rated Voltage</i>)</b>	3.6 V
<b>Inersia Rotor (<i>Rotor Inertia</i>)</b>	54 g.cm <sup>2</sup>
<b>Berat Motor</b>	280 gram

### 2.3.3 Driver Motor (A4988 Stepper Driver)



Gambar 2.4 Driver Motor (A4988 Stepper Driver)

Driver motor A4988 berfungsi sebagai perantara antara mikrokontroler dan motor *Stepper*. Komponen ini bertugas menyuplai arus listrik yang cukup besar ke motor berdasarkan sinyal kontrol (Step dan Direction) yang dikirim oleh Arduino. Driver ini juga mendukung fitur *microstepping*, yang membuat pergerakan robot menjadi lebih halus dan mengurangi getaran mekanis [19].

Tabel 2.3 Spesifikasi Driver Motor (A4988 Stepper Driver)

<b>Parameter</b>	<b>Spesifikasi</b>
<b>Tegangan Operasi Motor (VMOT)</b>	8 V hingga 35 V
<b>Tegangan Operasi Logika (VDD)</b>	3 V hingga 5.5 V (cocok untuk Arduino 5V)
<b>Arus Kontinu per Fase</b>	1.0 A (tanpa <i>heatsink</i> / kipas tambahan)
<b>Arus Maksimum per Fase</b>	<b>2.0 A</b> (wajib menggunakan <i>heatsink</i> )

	dan pendingin aktif)
<b>Proteksi Internal</b>	<i>Thermal Shutdown, Under-Voltage Lockout, dan Cross-Current Protection</i>
<b>Fitur Kendali</b>	Otomatis mendeteksi <i>Current Decay Mode</i> (Mixed, Fast, atau Slow)

### 2.3.4 Sensor Posisi (Endstop/Limit Switch)



Gambar 2.5 Sensor A3144

Robot dilengkapi dengan beberapa sensor *endstop* (tipe magnetik A3144 atau mekanik) yang dipasang pada setiap sumbu. Fungsi Kalibrasi: Sensor ini digunakan saat proses homing (perintah G28) untuk menentukan titik referensi nol absolut robot. Keamanan Mekanik: Mencegah robot bergerak melampaui batas fisik yang dapat merusak struktur mekanik atau motor. Halliday et al. (2013) dalam "*Fundamentals of Physics*" menjelaskan bahwa sensor *Hall Effect* seperti A3144 bekerja berdasarkan prinsip medan magnet yang menghasilkan tegangan output saat mendeteksi kutub magnet [20].

1. **Fungsi Kalibrasi:** Sensor ini digunakan saat proses *homing* (perintah G28) untuk menentukan titik referensi nol absolut robot.
2. **Keamanan Mekanik:** Mencegah robot bergerak melampaui batas fisik yang dapat merusak struktur mekanik atau motor.

Robot ini memiliki beberapa titik sensor krusial sesuai dengan jalur kabel pada *Wiring Guide*:

1. ***Upper Shank Endstop***: Terletak di sendi lengan atas untuk mendeteksi batas sudut maksimal.
2. ***Lower Shank Endstop***: Terletak di sendi lengan bawah.
3. ***Rotation Base Endstop***: Terletak di bagian basis untuk mengkalibrasi putaran robot.
4. ***Rail Endstop***: Terletak di ujung rel linier (E0) untuk mendeteksi posisi *Home* pada jarak 0mm.

Setiap sensor dihubungkan melalui konektor DB25 atau DB15 pada pin tertentu:

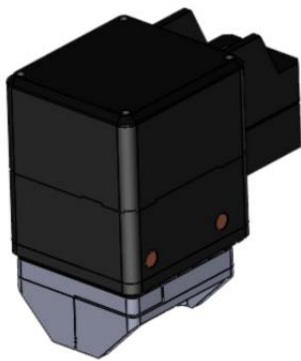
1. **Sensor 1, 2, dan 3**: Terhubung ke pin 2, 3, dan 4.
2. **Sensor Stop E0 (Rail)**: Terhubung ke pin 21 pada konektor DB25.
3. **Catu Daya**: Sensor ini memerlukan tegangan **5V** dan **GND** agar dapat beroperasi.

Tabel 2.4 Spesifikasi Sensor A3144

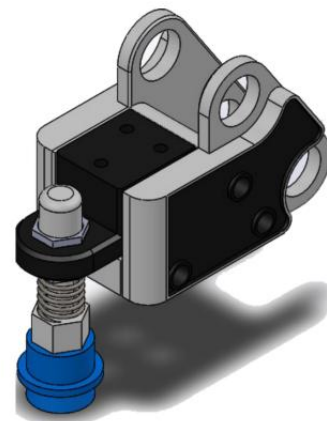
Parameter	Spesifikasi
<b>Type Sensor</b>	<i>Digital Output Hall Effect Switch</i> (Sensitivitas Magnetik)
<b>IC Utama</b>	A3144 / A3144E
<b>Tegangan Operasi (VCC)</b>	4.5 V hingga 24 V DC (sangat stabil dihubungkan ke 5V Arduino)
<b>Arus Output Maksimum</b>	25 mA
<b>Jenis Output</b>	Digital (Logika <i>HIGH / LOW</i> )
<b>Kondisi Output Default</b>	<b><i>HIGH (5V)</i></b> saat tidak ada medan

	magnet (menggunakan resistor <i>pull-up</i> )
<b>Kondisi Output Aktif</b>	<b>LOW (0V)</b> saat mendeteksi kutub Selatan ( <i>South Pole</i> ) magnet
<b>Suhu Operasi</b>	-40°C hingga 85°C

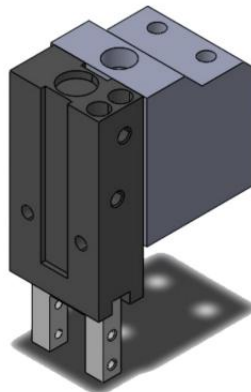
### 2.3.5 End-effector (Varian Gripper)



Gambar 2.7 Gripper Servo



Gambar 2.6 Gripper Vakum

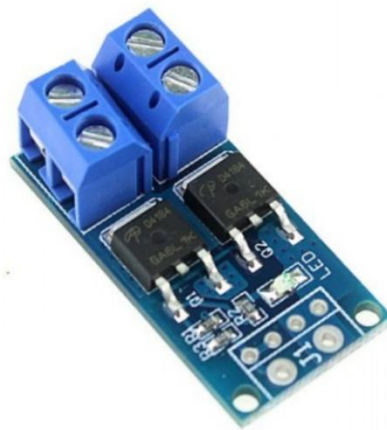


Gambar 2.8 Gripper Pneumatic

Bagian ujung robot yang bersentuhan langsung dengan objek kerja. RNV3 memiliki desain basis gripper standar yang memungkinkan penggantian secara praktis.

1. **Servo Gripper:** Menggunakan motor servo untuk mekanisme capit.
2. **Vacuum Gripper:** Dilengkapi dengan pompa vakum dan katup solenoid (valve 5v) untuk mengangkat objek datar melalui daya hisap.
3. **Pneumatic/Pen/Laser:** Pilihan lain untuk kebutuhan penanganan material bertenaga udara, menggambar, atau penandaan.

### 2.3.6 Power Management & Switching (Mosfet LG1-LG3)



Gambar 2.9 MOSFET PWM *Trigger Switch* Module 15A

Papan kontrol RNV3 dilengkapi dengan *logic IC Mosfet* yang mampu menangani arus hingga 5A. Komponen ini berfungsi sebagai sakelar elektronik untuk mengendalikan perangkat eksternal seperti konveyor, lampu indikator, atau pompa tanpa membebani mikrokontroler secara langsung. Output ini dapat diaktifkan melalui perintah khusus seperti M1, M206, atau M6 [21].

#### 1. Karakteristik Teknis Mosfet

- a) **Kapasitas Arus:** Mosfet ini didesain sebagai *main switch* yang mampu menangani beban arus hingga **5A**.

- b) **Tegangan Kerja:** Modul ini dapat mengontrol perangkat dengan tegangan yang bervariasi, mulai dari **5V, 12V, hingga 24V** (menggunakan daya eksternal).
- c) **Tipe Sakelar:** Mosfet ini bekerja sebagai sakelar pada jalur **GND (Ground)**. Artinya, beban dihubungkan langsung ke kutub positif (V+), sementara Mosfet memutuskan atau menyambung jalur negatifnya.

## 2. Identifikasi Sumbu Kendali (LG1 - LG3)

Sistem switching ini dibagi menjadi tiga kanal independen yang dapat dikendalikan melalui antarmuka atau perintah G-Code spesifik:

- a) **LG1 (Logic GND 1):** Diaktifkan dengan perintah **M1** dan dimatikan dengan **M2**. Biasanya digunakan untuk beban ringan seperti konveyor 5V.
- b) **LG2 (Logic GND 2):** Diaktifkan dengan perintah **M206** dan dimatikan dengan **M207**. Dalam konfigurasi gripper vakum, LG2 sering digunakan untuk mengontrol *Valve* (katup) 5V.
- c) **LG3 (Logic GND 3):** Diaktifkan dengan perintah **M6** dan dimatikan dengan **M7**. Pada sistem vakum, LG3 digunakan untuk menggerakkan *Pump* (pompa) 12V/5V.

## 3. Indikator Visual dan Proteksi

Berdasarkan diagram skematik (Halaman 26), setiap kanal Mosfet dilengkapi dengan komponen pendukung:

- a) **LED Indikator:** Terdapat LED (LED1, LED2, LED3) yang akan menyala sebagai penanda visual saat jalur Mosfet sedang dalam kondisi *ON* atau aktif.
- b) **Resistor Proteksi:** Menggunakan kombinasi resistor (seperti 10k dan 100k) untuk memastikan gerbang (*gate*) Mosfet mendapatkan sinyal logika yang stabil dari pin Arduino (D8, D9, D10) tanpa terjadi kebocoran arus.

#### 4. Koneksi Fisik

Akses ke output Mosfet ini disediakan melalui dua konektor utama:

- a) **DB25:** Pin 7 (LG1), Pin 8 (LG2), dan Pin 9 (LG3).
- b) **DB15:** Pin 7, 8, dan 9 yang terhubung secara paralel dari DB25 untuk memudahkan pemasangan aksesoris langsung pada area gripper.

Dengan spesifikasi arus 5A dan kontrol *low-side switching* ini, robot RNV3 memiliki fleksibilitas tinggi untuk diintegrasikan dengan berbagai perangkat otomasi tambahan di luar pergerakan lengannya sendiri.

Tabel 2.5 Spesifikasi MOSFET PWM *Trigger Switch Module* 15A

Parameter	Spesifikasi
Tipe Komponen	N-Channel Power MOSFET
Tegangan Beban Maksimum	Hingga 100 V DC (Sangat aman untuk tegangan robot 12V/24V)
Tegangan Kendali Gerbang	20 V (Aktif penuh pada tingkat logika 5V dari Arduino)
Arus Kontinu Maksimum	9.4 A hingga 33 A (Tergantung tipe IC yang terpasang)
Resistansi Internal Aktif	Sangat rendah (0.04 Ohm), mengurangi panas saat bekerja
Konfigurasi Output Modul	3 Kanal Terpisah (LG1, LG2, LG3)
Isolasi Sinyal	Dilengkapi <i>Optocoupler</i> (pada beberapa varian board) untuk melindungi Arduino dari lonjakan arus balik ( <i>back-</i>

## 2.4 Robot Manipulator Tiga Derajat Kebebasan

Robot manipulator tiga derajat kebebasan adalah robot yang memiliki tiga variabel gerak independen. Pada robot planar, tiga derajat kebebasan biasanya berupa tiga sendi putar. Sendi pertama mengatur arah *Link* pertama terhadap titik dasar, sedangkan sendi kedua mengatur arah *Link* kedua relatif terhadap *Link* pertama. Gerakan kedua sendi membentuk konfigurasi lengan yang dapat mencapai beberapa titik dalam area kerja tertentu [22].

Area kerja robot dua *Link* ditentukan oleh panjang *Link* dan batas sudut sendi. Jika kedua *Link* memiliki panjang  $L_1$  dan  $L_2$ , jarak maksimum yang dapat dicapai adalah  $L_1 + L_2$  ketika kedua *Link* lurus searah. Jarak minimum secara teoritis adalah nilai absolut  $L_1 - L_2$  ketika *Link* saling berlawanan arah, meskipun pada sistem *Stepper* nyata batas mekanik dapat membatasi kondisi tersebut. Dalam penelitian ini,  $L_1$  bernilai 14 cm dan  $L_2$  bernilai 14 cm sehingga jangkauan maksimum teoritis adalah 28 cm [22].

Robot tiga derajat kebebasan sering digunakan sebagai model dasar karena cukup sederhana untuk dianalisis tetapi tetap memperlihatkan prinsip penting robot manipulator. Melalui model ini, pengguna dapat memahami pengaruh sudut bahu dan siku terhadap posisi akhir. Model ini juga menjadi pintu masuk menuju pembahasan lebih lanjut seperti *inverse kinematics*, trajectory planning, dan kontrol closed-loop.

## 2.5 Trigonometri pada Robot Planar

Trigonometri merupakan dasar utama dalam perhitungan *Forward kinematics* robot lengan 3 DOF. Untuk sebuah *Link* dengan panjang  $L$  dan sudut theta terhadap sumbu X, komponen koordinat X dihitung dengan  $L \cos \theta$ , sedangkan komponen koordinat Y dihitung dengan  $L \sin \theta$ . Jika terdapat dua *Link* utama setelah *base*, posisi ujung *Link* kedua merupakan penjumlahan vektor dari *Link* pertama dan *Link* kedua. *Link* kedua memiliki sudut total  $\theta_1 + \theta_2$

karena orientasinya relatif terhadap *Link* pertama, sedangkan  $\theta_0$  memutar hasil jangkauan radial ke arah sumbu X dan Y [23].

Persamaan robot 3 DOF dapat ditulis sebagai  $r = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)$ ,  $z = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)$ ,  $x = r \cos(\theta_0)$ , dan  $y = r \sin(\theta_0)$ . Persamaan ini menunjukkan bahwa posisi *end-effector* tidak hanya dipengaruhi oleh sudut sendi kedua, tetapi juga oleh sudut sendi pertama. Perubahan  $\theta_1$  akan memutar seluruh struktur robot, sedangkan perubahan  $\theta_2$  akan mengubah konfigurasi relatif *Link* kedua terhadap *Link* pertama.

## 2.6 Forward kinematics

*Forward kinematics* adalah proses menghitung posisi dan orientasi *end-effector* dari parameter sendi yang diketahui. Pada robot dengan *joint revolute*, parameter yang digunakan adalah sudut. Pada robot dengan *joint prismatic*, parameter yang digunakan adalah perpindahan linear. Dalam robot lengan tiga motor *Stepper*, kedua *joint* bersifat *revolute* sehingga masukan utamanya adalah  $\theta_0$ ,  $\theta_1$ , dan  $\theta_2$  [24].

Kelebihan *Forward kinematics* adalah perhitungannya langsung dan deterministik. Jika panjang *Link* dan sudut sendi diketahui, maka posisi dapat dihitung tanpa proses iterasi. Kelemahannya adalah metode ini tidak menjawab pertanyaan sebaliknya, yaitu sudut apa yang diperlukan agar robot mencapai posisi tertentu. Pertanyaan sebaliknya merupakan domain *inverse kinematics*. Meski demikian, *Forward kinematics* tetap menjadi dasar wajib karena *inverse kinematics* juga membutuhkan model geometri yang sama.

Dalam implementasi perangkat lunak, *Forward kinematics* dapat dibuat sebagai fungsi yang menerima tiga parameter sudut dan mengembalikan struktur data berisi koordinat X dan Y. Fungsi ini membuat program lebih mudah diuji karena perhitungan dipisahkan dari proses membaca input serial atau menggerakkan motor *Stepper*.

### 2.6.1 Parameter Denavit-Hartenberg

Denavit-Hartenberg (DH) merupakan metode standar yang digunakan untuk mendefinisikan kerangka koordinat pada setiap *joint* robot manipulator. Metode ini dikembangkan oleh Jacques Denavit dan Richard Hartenberg pada tahun 1955 dan hingga saat ini masih menjadi acuan utama dalam pemodelan kinematika robot. Metode DH menggunakan empat parameter yang menggambarkan hubungan geometris antara dua *Link* yang berdekatan:

1.  $\theta_i$  (*joint angle*): sudut rotasi dari sumbu  $x_{i-1}$  ke sumbu  $x_i$  terhadap sumbu  $z_{i-1}$
2.  $d_i$  (*Link offset*): jarak dari titik potong sumbu  $z_{i-1}$  dan  $x_i$  ke titik potong sumbu  $z_i$  dan  $x_i$ , diukur sepanjang sumbu  $z_{i-1}$
3.  $a_i$  (*Link length*): jarak dari titik potong sumbu  $z_{i-1}$  dan  $x_i$  ke titik potong sumbu  $z_i$  dan  $x_i$ , diukur sepanjang sumbu  $x_i$
4.  $\alpha_i$  (*Link twist*): sudut rotasi dari sumbu  $z_{i-1}$  ke sumbu  $z_i$  terhadap sumbu  $x_i$

Untuk robot lengan 3 DOF yang digunakan dalam penelitian ini, parameter DH didefinisikan sebagai berikut:

Tabel 2.6 Parameter D-H

<i>Joint</i>	$\theta$ ( <i>sudut joint</i> )	$d$ ( <i>offset Link</i> )	$a$ ( <i>panjang Link</i> )	$\alpha$ ( <i>twist Link</i> )
1	$\theta_1$	0	0	$90^\circ$
2	$\theta_2$	0	140 mm	$0^\circ$
3	$\theta_3$	0	140 mm	$0^\circ$
<i>End-effector</i>	—	0	77 mm	$0^\circ$

**Keterangan setiap parameter:**

1.  $\theta_1, \theta_2, \theta_3$  = sudut *joint* yang merupakan variabel (berubah sesuai perintah motor)
2.  $d$  = *offset* sepanjang sumbu Z (0 untuk semua *joint* karena konfigurasi planar)
3.  $a$  = panjang *Link* sepanjang sumbu X (140 mm untuk *Link* 1 dan 2, 77 mm untuk *end-effector*)

4.  $\alpha$  = sudut *twist* antara sumbu Z ( $90^\circ$  untuk *joint* 1 karena perubahan orientasi dari vertikal ke horizontal)

Penjelasan lebih lanjut mengenai setiap parameter adalah sebagai berikut. Parameter  $\theta_1$  merepresentasikan rotasi base robot terhadap sumbu vertikal. Pada *joint* 1, nilai  $\alpha_1 = 90^\circ$  karena sumbu rotasi base (vertikal) harus diubah menjadi sumbu horizontal untuk *Link* 1. Parameter  $\theta_2$  dan  $\theta_3$  masing-masing merepresentasikan sudut lengan atas dan lengan bawah terhadap horizontal. Karena robot ini bersifat planar (gerakan pada satu bidang), semua nilai  $d$  bernilai nol.

### 2.6.2 Matriks Transformasi Homogen

Matriks transformasi homogen merupakan matriks berukuran  $4 \times 4$  yang digunakan untuk menggambarkan posisi dan orientasi suatu kerangka koordinat relatif terhadap kerangka koordinat lainnya. Matriks ini menggabungkan informasi rotasi dan translasi dalam satu representasi matematis yang kompak. Matriks transformasi homogen memiliki bentuk umum:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

di mana  $R$  merupakan matriks rotasi  $3 \times 3$  dan  $p$  merupakan vektor translasi  $3 \times 1$ . Elemen  $R$  menggambarkan orientasi kerangka koordinat tujuan relatif terhadap kerangka acuan, sedangkan elemen  $p$  menggambarkan posisi titik asal kerangka koordinat tujuan.

Untuk setiap *joint* pada robot, matriks transformasi homogen dihitung menggunakan parameter DH. Matriks transformasi dari kerangka  $i-1$  ke kerangka  $i$  dinyatakan sebagai:

$$T_i^{i-1} = Rot_z(\theta_i) \cdot Trans_z(d_i) \cdot Trans_x(a_i) \cdot Rot_x(\alpha_i)$$

yang menghasilkan matriks:

$$\begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Berdasarkan parameter DH pada Tabel 2.6, matriks transformasi untuk masing-masing *joint* adalah sebagai berikut:

**Matriks Transformasi *Joint* 1 ( $\alpha_1 = 90^\circ$ ):**

Pada matriks *joint* 1, nilai  $\cos 90^\circ = 0$  dan  $\sin 90^\circ = 1$  disubstitusikan ke dalam matriks DH standar. Matriks ini merepresentasikan rotasi base robot dari sumbu vertikal ke sumbu horizontal.

$$\begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Matriks Transformasi *Joint* 2 ( $\alpha_2 = 0^\circ$ ):**

$$\begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 140 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 140 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Matriks Transformasi *Joint* 3 ( $\alpha_3 = 0^\circ$ ):**

Kedua matriks *joint* 2 dan 3 memiliki bentuk yang serupa karena keduanya memiliki  $\alpha = 0^\circ$ , yang berarti tidak ada perubahan orientasi sumbu Z antara *Link* yang berdekatan. Perbedaan hanya terletak pada indeks sudut *joint* ( $\theta_2$  dan  $\theta_3$ ).

$$\begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 140 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 140 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 2.6.3 Matriks Transformasi Total

Matriks transformasi total dari base ke *end-effector* merupakan hasil perkalian seluruh matriks transformasi homogen dari setiap *joint*. Matriks total diperoleh melalui:

$$T_0^3 = T_0^1 \times T_1^2 \times T_2^3$$

Hasil perkalian matriks transformasi homogen tersebut menghasilkan matriks 4×4 yang komponen posisinya merepresentasikan koordinat kartesian *end-effector*. Komponen translasi dari matriks total adalah sebagai berikut:

$$X = 140 \cos(\theta_1) \cos(\theta_2) + 140 \cos(\theta_1) \cos(\theta_2 + \theta_3) + 77 \sin(\theta_1)$$

$$Y = 140 \sin(\theta_1) \cos(\theta_2) + 140 \sin(\theta_1) \cos(\theta_2 + \theta_3) + 77 \cos(\theta_1)$$

$$Z = 140 \sin(\theta_2) + 140 \sin(\theta_2 + \theta_3)$$

Persamaan yang dihasilkan dari perkalian matriks transformasi homogen tersebut konsisten dengan persamaan *forward kinematics* berbasis trigonometri yang diimplementasikan dalam program. Hubungan antara keduanya dapat dijelaskan sebagai berikut:

Suku  $140 \cos(\theta_2) + 140 \cos(\theta_2 + \theta_3)$  merepresentasikan  $r\_arm$  (jarak radial lengan), yang merupakan proyeksi horizontal total dari kedua *Link*. Suku 77 merepresentasikan  $ee\_offset$  (offset *end-effector*), yaitu jarak dari ujung *Link* 2 ke titik pusat alat. Penjumlahan keduanya menghasilkan  $r\_total$  (jarak radial total) dari base ke *end-effector*.

Dengan demikian, persamaan posisi *end-effector* dapat ditulis sebagai:

$$X = r\_total \cdot \sin(\theta_1) = (r\_arm + ee\_offset) \cdot \sin(\theta_1)$$

$$Y = r\_total \cdot \cos(\theta_1) = (r\_arm + ee\_offset) \cdot \cos(\theta_1)$$

$$Z = 140 \sin(\theta_2) + 140 \sin(\theta_2 + \theta_3)$$

Persamaan ini identik dengan rumus *forward kinematics* yang diimplementasikan dalam program `gui_forward_kinematics.py`, di mana:

1.  $r\_arm = L_1 \cdot \cos(\theta_2) + L_2 \cdot \cos(\theta_2 + \theta_3)$
2.  $r\_total = r\_arm + ee\_offset$

3.  $X = r\_total \cdot \sin(\theta_1)$
4.  $Y = r\_total \cdot \cos(\theta_1)$
5.  $Z = L_1 \cdot \sin(\theta_2) + L_2 \cdot \sin(\theta_2 + \theta_3)$

Konsistensi antara kedua pendekatan tersebut membuktikan bahwa perhitungan *forward kinematics* yang diimplementasikan dalam program sudah benar. Pendekatan trigonometri yang digunakan dalam program merupakan penyederhanaan dari perkalian matriks transformasi homogen Denavit-Hartenberg, yang menghasilkan persamaan yang sama.

## 2.7 Komunikasi Serial

Komunikasi serial digunakan untuk pertukaran data antara komputer dan mikrokontroler. Pada sistem ini, pengguna mengetik tiga angka yang dipisahkan spasi, misalnya 0 0 30 45. Program membaca karakter dari serial monitor, menyimpannya dalam buffer, lalu memprosesnya ketika karakter baris baru diterima. Jika format benar, program mengubah teks menjadi angka, menggerakkan motor *Stepper*, dan mencetak hasil *forward kinematics* [25].

Pendekatan serial sangat cocok untuk pengujian awal karena sederhana dan tidak membutuhkan antarmuka tambahan. Pengguna dapat mencoba banyak kombinasi sudut dengan cepat. Data yang dihasilkan juga dapat dicatat untuk analisis laporan.