

BAB II

DASAR TEORI

Bab ini membahas teori dasar yang berkaitan dengan lengan robot dan sistem kendalinya. Pembahasan meliputi konsep lengan robot, degree of freedom, *inverse kinematics*, mikrokontroler, komunikasi serial, aktuator *stepper* dan servo, sensor posisi referensi, transmisi *timing belt-pulley*, torsi, catu daya, serta antarmuka TFT. Teori pada bab ini disajikan secara umum sebagai dasar untuk memahami hubungan antara geometri manipulator, pengendalian aktuator.

2.1. Tinjauan Pustaka

Tinjauan pustaka digunakan untuk membandingkan penelitian yang dilakukan dengan penelitian terdahulu yang memiliki keterkaitan pada aspek lengan robot, *inverse kinematics*, sistem pemetikan buah, serta pengolahan koordinat target. Pemilihan literatur mempertimbangkan kedekatan objek penelitian, konfigurasi manipulator, metode kinematika, sistem persepsi, dan mekanisme *end-effector*. Hasil tinjauan pustaka disajikan pada Tabel 2.1

Tabel 2. 1 Tinjauan Pustaka

Judul/Penuli	Keterangan	Persamaan dan Perbedaan
		Persamaan: sama-sama menggunakan manipulator 5
	Penelitian menguji kelayakan pemanenan chile pepper	DOF, koordinat target tiga dimensi, dan <i>inverse kinematics</i>
A Study on the Feasibility of Robotic Harvesting for Chile Pepper/M. U. Masood dan M. Haghshenas-Jaryani / 2021	menggunakan manipulator serial 5 DOF, kamera depth, forward dan <i>inverse kinematics</i> , analisis <i>workspace</i> , serta <i>end-effector</i> pemotong. Koordinat buah digunakan untuk menentukan gerakan manipulator menuju target [2].	untuk pemetikan cabai. Perbedaan: penelitian pembandingan menggunakan mekanisme pemotong dan program berbasis MATLAB, sedangkan penelitian ini menggunakan STM32F411CEU6, kombinasi motor stepper-servo, sistem homing, dan <i>gripper</i> .

Judul/Penuli	Keterangan	Persamaan dan Perbedaan
Simplified 4-DOF Manipulator for Rapid Robotic Apple Harvesting/G. Hu et al. / 2022	Penelitian mengembangkan manipulator 4 DOF untuk robot pemanen apel dengan fokus pada penyederhanaan struktur, sistem kendali manipulator, serta evaluasi kinerja pada kebun apel [7].	Persamaan: sama-sama merancang manipulator untuk membawa <i>end-effector</i> menuju posisi buah. Perbedaan: penelitian pembandingan menggunakan 4 DOF untuk apel, sedangkan penelitian ini menggunakan 5 DOF untuk cabai merah dengan pengendali STM32F411CEU6.
Design of a Chili Pepper Harvesting Device for Hilly Chili Fields/W. Han et al. / 2025	Penelitian mengembangkan perangkat pemanen cabai dengan rekonstruksi <i>point cloud</i> 3D dan mekanisme leveling untuk mengatasi oklusi daun, ketidaksejajaran lengan, dan keterbatasan jangkauan pada lahan berbukit [3].	Persamaan: sama-sama membahas pemanenan cabai dengan koordinat tiga dimensi sebagai dasar pendekatan target. Perbedaan: penelitian pembandingan berfokus pada rekonstruksi lingkungan dan mekanisme leveling di lahan berbukit, sedangkan penelitian ini berfokus pada implementasi <i>inverse kinematics</i> dan kendali lengan 5 DOF.
Design, Kinematic Modeling, and Hybrid AI-Based Optimization of a Vision-Guided 6-DOF Robotic Arm for Capsicum Harvesting in Greenhouse Environments/W. Han et al. / 2025	Penelitian merancang lengan robot 6 DOF untuk pemanenan capsicum di greenhouse dengan pemodelan kinematik, sistem vision-guided, aktuator stepper dan servo, serta optimasi untuk mencapai target buah [4].	Persamaan: sama-sama membahas robot pemanen Capsicum, pemodelan kinematik, motor stepper, servo, dan pencapaian koordinat target. Perbedaan: penelitian pembandingan menggunakan 6 DOF dan optimasi berbasis AI, sedangkan penelitian ini menggunakan 5 DOF dan menitikberatkan implementasi <i>inverse kinematics</i> pada STM32F411CEU6.

Kajian lain menunjukkan bahwa pengembangan manipulator untuk aplikasi pertanian perlu mempertimbangkan konfigurasi mekanik, jangkauan kerja, sistem kendali, serta kinerja robot pada kondisi pengujian. Van Henten et al. [8] membahas optimasi desain manipulator untuk robot pemanen mentimun, sedangkan Zhang et al. [9] mengembangkan sistem dan kendali robot pemanen apel yang kemudian dievaluasi lebih lanjut pada kondisi lapangan [30]. Pengembangan lengan robot dengan desain terbuka dan *end-effector* yang dapat disesuaikan juga ditunjukkan melalui platform PARA [10]. Selain itu, pengembangan mobile robotic *arm* dengan kendali berbasis gestur dibahas pada [16].

2.2. Lengan Robot

Lengan robot atau robot manipulator merupakan sistem mekanik yang tersusun atas *link*, *joint*, dan *end-effector* untuk menghasilkan gerak terkontrol pada ruang kerja. Hubungan antara konfigurasi *joint* dan pose *end-effector* menjadi dasar pemodelan kinematika manipulator. *Forward kinematics* menentukan pose *end-effector* dari variabel *joint*, sedangkan *inverse kinematics* menentukan variabel *joint* yang diperlukan untuk mencapai pose target [31].

Jumlah gerak independen pada manipulator dinyatakan sebagai degree of freedom (DOF). Manipulator serial dapat memiliki beberapa *joint* revolute maupun prismatic sesuai kebutuhan geraknya. Pada konfigurasi lima DOF, misalnya, gerak dapat terdiri atas rotasi *base*, *shoulder*, *elbow*, *wrist pitch*, dan *wrist yaw*. Gerak buka-tutup *gripper* umumnya diperlakukan sebagai fungsi *end-effector* yang terpisah dari DOF utama penentu pose.

2.2.1. Batang (*Link*)

Link merupakan elemen struktural kaku yang menghubungkan satu *joint* dengan *joint* lainnya. Panjang, *offset*, dan orientasi *link* menentukan geometri rantai kinematik, jangkauan kerja, serta posisi *end-effector* yang dapat dicapai. Parameter geometris *link* digunakan secara langsung dalam pembentukan model kinematika robot [31].

2.2.2. Sendi (*Joint*)

Joint merupakan sambungan antar-*link* yang memungkinkan terjadinya gerak relatif. *Joint* dapat berupa revolute *joint* yang menghasilkan gerak rotasi atau

prismatic joint yang menghasilkan gerak translasi. Pada *joint revolute*, variabel kinematik dinyatakan sebagai sudut, sedangkan pada *joint prismatic* variabel kinematik dinyatakan sebagai perpindahan linear [31].

2.2.3. End-Effector

End-effector merupakan bagian pada ujung lengan yang berinteraksi langsung dengan objek kerja. Pose *end-effector* ditentukan oleh konfigurasi *joint* serta geometri *link*. Bentuk *end-effector* dapat berupa *gripper*, suction cup, alat potong, *tool* pengelasan, atau mekanisme lain sesuai fungsi yang dibutuhkan oleh suatu aplikasi robot [31].

2.3. Degree of Freedom (DOF) dan Inverse Kinematics

Degree of Freedom (DOF) menyatakan jumlah gerak independen yang dimiliki manipulator. Pada manipulator serial, variabel *joint* biasanya dinyatakan sebagai q_1, q_2 , hingga q_n sesuai jumlah DOF. Sebagai contoh, konfigurasi lima DOF dapat direpresentasikan oleh lima variabel *joint* yang mengatur rotasi *base*, *shoulder*, *elbow*, *wrist pitch*, dan *wrist yaw*. Pose target dapat dinyatakan menggunakan koordinat *Cartesian* X, Y, dan Z, sedangkan orientasi *end-effector* dapat dinyatakan dengan sudut orientasi yang diinginkan, seperti *pitch* (φ_d) dan *yaw* (ψ_d) [31].

Inverse kinematics digunakan untuk memperoleh variabel *joint* dari posisi dan orientasi target *end-effector*. Pendekatan geometris menyelesaikan hubungan tersebut menggunakan geometri segitiga, teorema Pythagoras, aturan cosinus, serta fungsi trigonometri [31]. Persamaan pada subbab ini merupakan bentuk dasar untuk manipulator serial dengan *joint revolute*. Parameter seperti panjang *link*, tinggi sumbu, dan *offset tool* dinyatakan sebagai parameter geometri yang nilainya dapat berbeda pada setiap konfigurasi manipulator. Pendekatan geometris juga tetap menjadi salah satu dasar dalam pengembangan metode *inverse kinematics* modern [11].

2.3.1. Sistem Koordinat dan Penentuan Sudut Joint Base

Pada model geometris dasar, sudut *joint base* ditentukan dari proyeksi koordinat target pada bidang horizontal X-Y. Jarak radial target diperoleh menggunakan hubungan Pythagoras, sedangkan sudut azimuth *base* dihitung

menggunakan fungsi atan2 agar informasi tanda koordinat dan kuadran target tetap terjaga [31].

$$r = \sqrt{X^2 + Y^2} \quad (2.1)$$

Keterangan Persamaan (2.1):

r = jarak radial horizontal target terhadap sumbu *base* (mm);

X = koordinat target pada sumbu X (mm);

Y = koordinat target pada sumbu Y (mm).

$$q_{1,ideal} = \text{atan2}(Y, X) \quad (2.2)$$

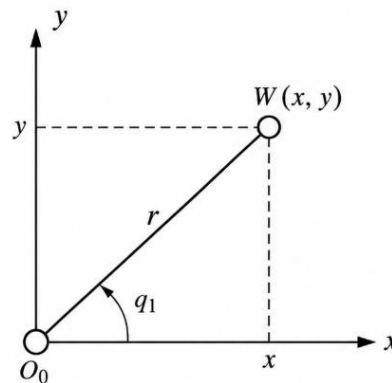
Keterangan Persamaan (2.2):

$q_{1,ideal}$ = sudut *base* ideal;

$\text{atan2}(Y, X)$ = fungsi arctangent dua argumen yang mempertahankan informasi kuadran;

X dan Y = koordinat target pada bidang horizontal.

Persamaan (2.1) menyatakan jarak radial target terhadap sumbu *base*, sedangkan Persamaan (2.2) menentukan sudut azimuth target. Kedua persamaan tersebut merupakan hubungan dasar antara koordinat *Cartesian* pada bidang horizontal dan sudut rotasi *base*. Definisi *frame* koordinat dan arah positif setiap sumbu harus ditetapkan secara konsisten agar hasil sudut memiliki makna yang sama pada seluruh proses kinematika. Visualisasi proyeksi koordinat target pada bidang horizontal X-Y yang digunakan untuk menentukan sudut *joint base* ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Proyeksi koordinat target pada bidang horizontal untuk menentukan sudut *joint base* diolah berdasarkan [31]

2.3.2. Inverse Kinematics Geometris Shoulder dan Elbow

Setelah sudut *base* diperoleh, penyelesaian *shoulder* dan *elbow* dapat direduksi menjadi persoalan geometri dua-link pada bidang radial-vertikal. Titik yang digunakan sebagai target dua-link dinyatakan sebagai *wrist center*, yaitu titik acuan sebelum bagian *wrist* dan *end-effector*. Posisi *wrist center* diperoleh dari posisi target setelah memperhitungkan tinggi *shoulder* dan *offset tool* secara geometris. Selanjutnya jarak *shoulder* menuju *wrist center* dihitung menggunakan teorema Pythagoras, sedangkan sudut *elbow* dan *shoulder* ditentukan menggunakan aturan cosinus dan fungsi trigonometri [31].

$$r_w = r - \Delta r_{tool} \quad (2.3)$$

Keterangan Persamaan (2.3):

r_w = komponen radial *wrist center* pada model ideal (mm);

r = jarak radial target (mm);

Δr_{tool} = proyeksi *offset tool* pada arah radial (mm).

$$z_w = (Z - H_s) - \Delta z_{tool} \quad (2.4)$$

Keterangan Persamaan (2.4):

Z_w = komponen vertikal *wrist center* terhadap *shoulder* (mm);

Z = koordinat vertikal target (mm);

H_s = tinggi sumbu *shoulder* terhadap *base* reference (mm);

Δz_{tool} = proyeksi *offset tool* pada arah vertikal (mm).

$$d = \sqrt{r_w^2 + z_w^2} \quad (2.5)$$

Keterangan Persamaan (2.5):

d = jarak lurus dari *shoulder* ke *wrist center* (mm);

r_w = komponen radial *wrist center* (mm);

z_w = komponen vertikal *wrist center* (mm).

$$D = \frac{(d^2 - L_1^2 - L_2^2)}{(2L_1L_2)} \quad (2.6)$$

Keterangan Persamaan (2.6):

D = variabel bantu aturan cosinus, tidak bersatuan;

d = jarak *shoulder-wrist* (mm);

L_1 = panjang link *shoulder-elbow* (mm);

$L2$ = panjang *link elbow-wrist pitch* (mm).

Persamaan (2.3) dan Persamaan (2.4) menyatakan komponen radial dan vertikal *wrist center*. Persamaan (2.5) menentukan jarak dari *shoulder* menuju *wrist center*, sedangkan Persamaan (2.6) membentuk variabel bantu berdasarkan aturan cosinus untuk memperoleh sudut *elbow*. Besaran Δr_{tool} dan Δz_{tool} merepresentasikan komponen geometris antara titik target *end-effector* dan *wrist center* sehingga nilainya bergantung pada bentuk serta dimensi manipulator yang dianalisis.

$$q_3 = \text{atan2}\left(\pm\sqrt{1 - D^2}, D\right) \quad (2.7)$$

Keterangan Persamaan (2.7):

q_3 = sudut *joint elbow*;

D = variabel bantu aturan cosinus; tanda $\pm$ menyatakan dua kemungkinan konfigurasi *elbow*;

atan2 = fungsi arctangent dua argumen.

Persamaan (2.7) menunjukkan bahwa satu posisi target dapat memiliki lebih dari satu solusi konfigurasi *elbow*. Pemilihan solusi dapat mempertimbangkan postur manipulator yang diinginkan, batas gerak *joint*, serta *workspace*. Setelah sudut *elbow* ditentukan, sudut *shoulder* dapat dihitung menggunakan Persamaan (2.8) [31].

$$q_2 = \text{atan2}(z_w, r_w) - \text{atan2}(L2 \sin(q_3), L1 + L2 \cos(q_3)) \quad (2.8)$$

Keterangan Persamaan (2.8):

q_2 = sudut *joint shoulder*;

r_w, z_w dan = koordinat *wrist center* pada bidang radial-vertikal;

$L1$ dan $L2$ = panjang *link* pertama dan kedua;

q_3 = sudut *elbow*;

$\sin$ dan $\cos$ = fungsi trigonometri.

$$|L_1 - L_2| \leq d \leq L_1 + L_2 \quad (2.9)$$

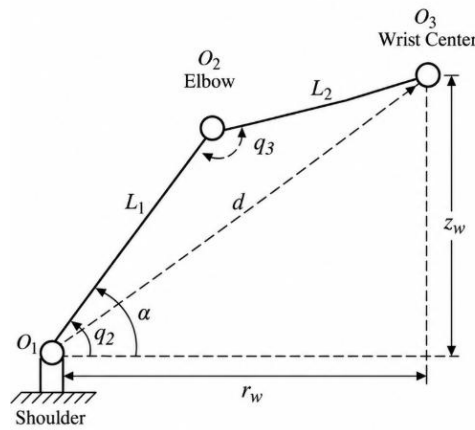
Keterangan Persamaan (2.9):

$L1$ = panjang *link shoulder-elbow*;

$L2$ = panjang *link elbow-wrist pitch*;

d = jarak *shoulder* ke *wrist center*. Ketaksamaan menyatakan syarat target berada dalam jangkauan geometris dua-link.

Persamaan (2.9) merupakan syarat keterjangkauan geometris mekanisme dua-link. Suatu target termasuk dalam *workspace* geometris apabila jarak *shoulder* ke *wrist center* berada di antara selisih dan jumlah panjang kedua *link*. Hubungan tersebut digunakan untuk menentukan batas jangkauan teoritis manipulator [31]. Geometri dua-link pada bidang radial-vertikal yang menjadi dasar penentuan sudut *shoulder* dan *elbow* ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Diagram geometris dua-link pada bidang radial-vertikal untuk menentukan sudut *shoulder* dan *elbow* diolah berdasarkan [31]

2.3.3. Orientasi *Wrist* pada Model Ideal

$$q_4 = \varphi_d - q_2 - q_3 \quad (2.10)$$

Keterangan Persamaan (2.10):

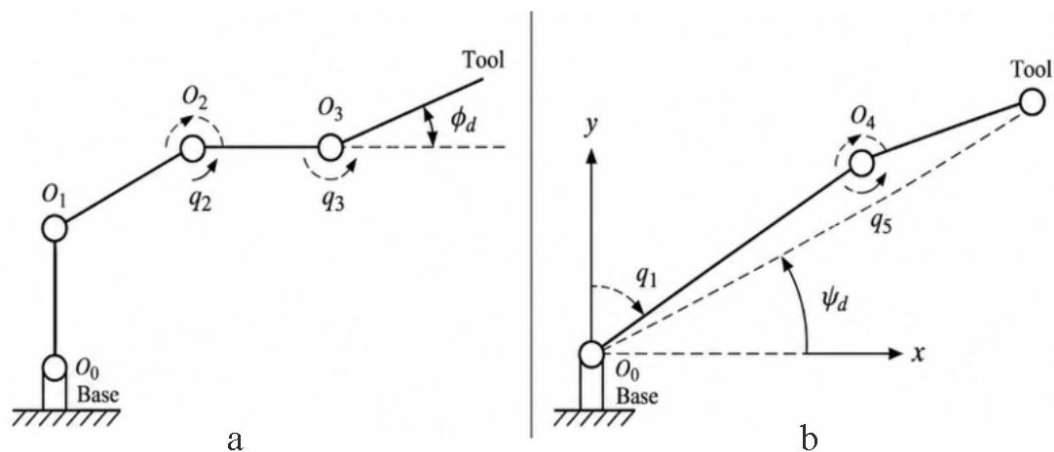
- q_4 = sudut *wrist pitch* pada model ideal;
- φ_d = orientasi *pitch* yang diinginkan pada *end-effector*;
- q_2 = sudut *shoulder*;
- q_3 = sudut *elbow*.

$$q_5 = \psi_d - q_1 \quad (2.11)$$

Keterangan Persamaan (2.11):

- q_5 = sudut *wrist yaw* pada model ideal;
- ψ_d = orientasi *yaw* yang diinginkan terhadap *frame* acuan;
- q_1 = sudut *base*.

Persamaan (2.10) dan Persamaan (2.11) menunjukkan hubungan dasar antara orientasi *end-effector* dan orientasi *joint* sebelumnya pada rantai kinematik. Pada model serial, sudut *wrist* dapat digunakan untuk melengkapi orientasi yang telah dibentuk oleh *base*, *shoulder*, dan *elbow*. Tanda serta urutan penjumlahan sudut dapat berbeda bergantung pada definisi *frame* dan konvensi kinematika, tetapi prinsip dasarnya tetap berupa hubungan antara orientasi target dan variabel *joint* [31]. Hubungan orientasi *wrist* divisualisasikan pada Gambar 2.3, yaitu (a) *wrist pitch* pada tampak samping dan (b) *wrist yaw* pada tampak atas.



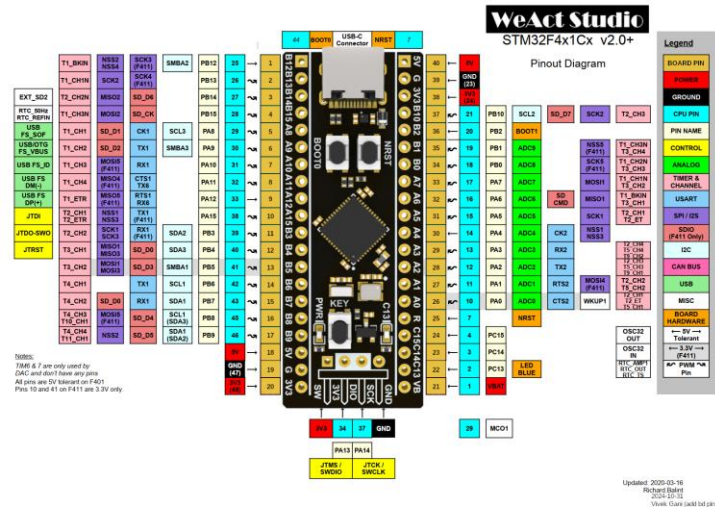
Gambar 2. 3 Gambar (a) Hubungan sudut *wrist pitch* dan Gambar (b) *wrist yaw* terhadap orientasi *tool* pada model ideal diolah berdasarkan [31]

2.4. STM32F411CEU6

STM32F411CEU6 merupakan mikrokontroler 32-bit berbasis Arm Cortex-M4 dengan FPU dan instruksi DSP, frekuensi CPU hingga 100 MHz, Flash 512 KB, SRAM 128 KB, ADC 12-bit, serta antarmuka komunikasi seperti USART, SPI/I²S, I²C, USB OTG FS, dan SDIO. Spesifikasi pada subbab ini mengacu pada datasheet resmi STM32F411xC/xE dari STMicroelectronics [12].

STM32F411CEU6 dapat digunakan sebagai pengendali tertanam untuk berbagai fungsi kendali gerak. Timer dapat dimanfaatkan untuk menghasilkan pulsa atau PWM, GPIO digunakan untuk masukan dan keluaran digital, USART untuk komunikasi serial, sedangkan SPI dapat digunakan untuk berkomunikasi dengan perangkat periferal. Kombinasi periferal tersebut mendukung penerapan kendali aktuator, pembacaan sensor, komunikasi data, dan antarmuka pengguna [12], [13].

Pada sistem ini STM32F411CEU6 digunakan pada papan pengembangan Black Pill; diagram *pinout* papan tersebut ditunjukkan pada Gambar 2.4, sedangkan spesifikasi utama mikrokontroler dirangkum pada Tabel 2.2.



Gambar 2. 4 Diagram *pinout* papan STM32F411CEU6 Black Pill [32]

Tabel 2. 2 Spesifikasi STM32F411CEU6

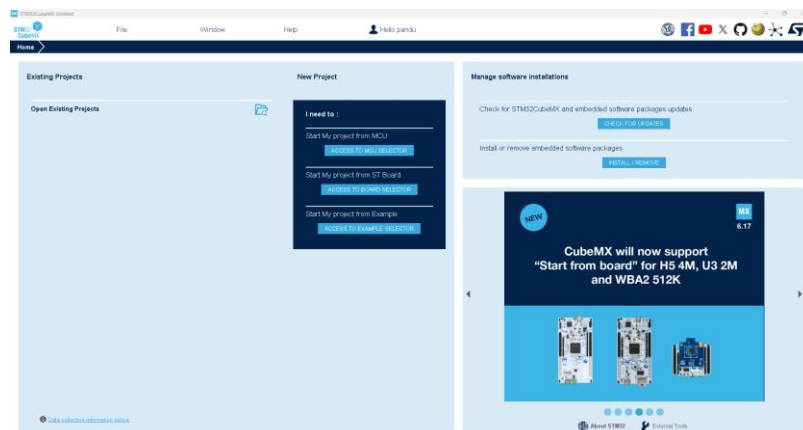
Parameter	Spesifikasi
Core	Arm Cortex-M4 32-bit + FPU dan DSP
Frekuensi CPU maksimum	100 MHz
Flash memory	512 KB
SRAM	128 KB
Tegangan operasi IC	1,7-3,6 V
ADC	12-bit
Antarmuka komunikasi	USART/UART, SPI/I ² S, I ² C, USB OTG FS, SDIO
Package STM32F411CEU6	UFQFPN48

2.5. Perangkat Lunak STM32F411CEU6

Pengembangan *firmware* STM32F411CEU6 menggunakan perangkat lunak resmi STMicroelectronics untuk konfigurasi perangkat keras, penulisan dan kompilasi program, serta proses pemrograman dan *debugging*. Perangkat yang digunakan pada tahapan tersebut meliputi STM32CubeMX, STM32CubeIDE, dan ST-LINK.

2.5.1. STM32CubeMX

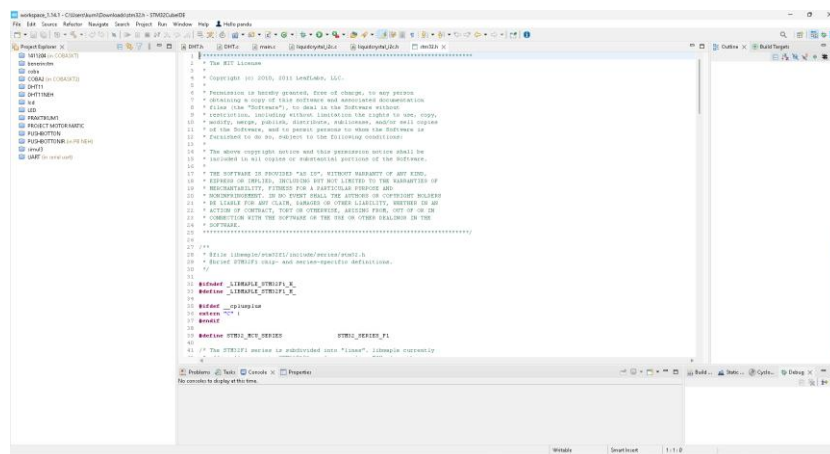
STM32CubeMX merupakan perangkat lunak konfigurasi grafis resmi STMicroelectronics yang digunakan untuk mengatur *pinout*, *clock*, periferal, *middleware*, dan menghasilkan kode inisialisasi proyek STM32. Konfigurasi timer, USART, SPI, GPIO, dan periferal lain dapat disiapkan melalui antarmuka grafis sebelum kode aplikasi dikembangkan lebih lanjut [19]. Tampilan STM32CubeMX yang digunakan pada proses konfigurasi ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Antarmuka STM32CubeMX untuk konfigurasi STM32F411CEU6

2.5.2. STM32CubeIDE

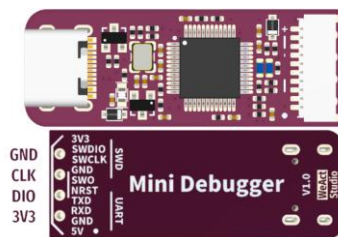
STM32CubeIDE merupakan integrated development environment (IDE) resmi STMicroelectronics untuk pengembangan mikrokontroler STM32. Perangkat lunak ini menyediakan editor kode, proses kompilasi, pemrograman perangkat, *debugging*, serta integrasi dengan konfigurasi proyek STM32 [20]. Lingkungan pengembangan *firmware* yang digunakan ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Lingkungan pengembangan *firmware* pada STM32CubeIDE

2.5.3. ST-LINK dan Serial Wire Debug

ST-LINK merupakan antarmuka *debugger* dan *programmer* yang digunakan pada pengembangan mikrokontroler STM32. Pada implementasi sistem ini digunakan modul WeAct MiniDebugger dengan *firmware* STLink V2.1. Modul tersebut menyediakan antarmuka Serial Wire Debug (SWD) untuk pemrograman dan *debugging* STM32; jalur SWDIO digunakan untuk data, sedangkan SWCLK digunakan sebagai sinyal *clock*. Prinsip antarmuka SWD mengacu pada dokumentasi STMicroelectronics [21], sementara bentuk modul WeAct MiniDebugger yang digunakan ditunjukkan pada Gambar 2.7 [33].



Gambar 2. 7 Modul WeAct MiniDebugger STLink V2.1 [33]

2.6. Protokol Komunikasi UART

Universal Asynchronous Receiver/Transmitter (UART) merupakan metode komunikasi serial asinkron yang menggunakan jalur TX untuk mengirim data dan RX untuk menerima data. Karena tidak menggunakan *clock* bersama, kedua perangkat harus menggunakan parameter komunikasi yang sama, seperti baud rate, jumlah bit data, parity, dan stop bit. Data dikirim dalam bentuk *frame* yang umumnya terdiri atas start bit, bit data, parity opsional, dan stop bit [13].

2.7. NVIDIA Jetson Orin Nano Super

NVIDIA Jetson Orin Nano Super Developer Kit 8 GB merupakan platform komputer *embedded* untuk komputasi edge yang mendukung aplikasi *artificial intelligence*, *computer vision*, dan robotika. Perangkat ini menggabungkan CPU, GPU, memori, serta antarmuka I/O dalam satu platform komputasi berukuran ringkas [14].

Konfigurasi Super menggunakan GPU NVIDIA Ampere dengan 1024 CUDA cores dan 32 Tensor Cores, CPU 6-core Arm Cortex-A78AE, memori 8 GB

LPDDR5 dengan bandwidth hingga 102 GB/s, serta performa AI hingga 67 INT8 TOPS. Rentang daya modul dapat dikonfigurasi 7-25 W [14].

NVIDIA Jetson Orin Nano Super merupakan komputer *embedded* yang dirancang untuk pemrosesan komputasi intensif, termasuk aplikasi *artificial intelligence*, *computer vision*, dan robotika. Pada arsitektur robot, komputer *embedded* dapat berperan sebagai pengolah tingkat tinggi untuk persepsi, pengolahan data sensor, dan pembentukan target, sedangkan pengendali tertanam menangani tugas kendali aktuator yang membutuhkan respons waktu lebih deterministik [14]. Bentuk perangkat NVIDIA Jetson Orin Nano Super Developer Kit ditunjukkan pada Gambar 2.8, sedangkan spesifikasi utama versi 8 GB dirangkum pada Tabel 2.3 [14].



Gambar 2. 8 NVIDIA Jetson Orin Nano Super Developer Kit [14]

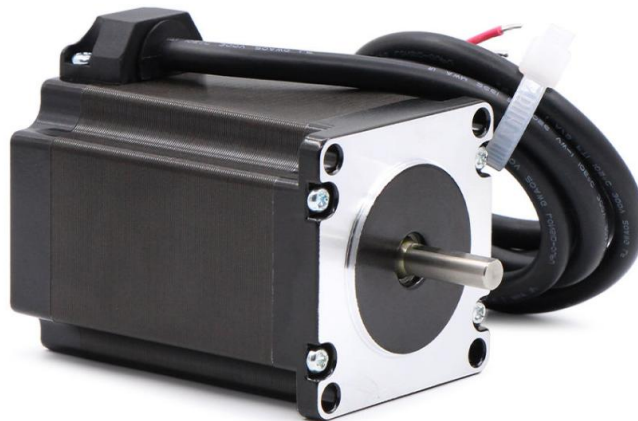
Tabel 2. 3 Spesifikasi NVIDIA Jetson Orin Nano Super Developer Kit 8 GB [14]

Parameter	Spesifikasi
AI performance	hingga 67 INT8 TOPS
GPU	NVIDIA Ampere, 1024 CUDA cores, 32 Tensor Cores
CPU	6-core Arm Cortex-A78AE, hingga 1,7 GHz
Memori	8 GB 128-bit LPDDR5
Bandwidth memori	hingga 102 GB/s
Penyimpanan	microSD; dukungan NVMe melalui M.2 Key M
Jaringan	1 × Gigabit Ethernet
Rentang daya	7-25 W

2.8. Motor *Stepper* NEMA 23

Motor *stepper* menghasilkan perpindahan sudut diskrit sesuai urutan eksitasi fasa. Posisi poros dapat dikendalikan berdasarkan jumlah pulsa langkah, sedangkan kecepatan dipengaruhi oleh frekuensi pulsa. Karakteristik tersebut sesuai untuk penggerak *joint* yang memerlukan pengaturan posisi berbasis pulsa [15].

NEMA 23 merupakan standar ukuran *frame* motor *stepper* dengan dimensi muka sekitar 57 mm, bukan satu nilai spesifikasi listrik dan mekanik yang tetap. Setiap seri dapat memiliki panjang bodi, arus nominal, *holding torque*, serta diameter poros yang berbeda. Salah satu varian G-Penny NEMA 23 memiliki konfigurasi dua fasa, *step angle* $1,8^\circ$, arus nominal 3 A, *holding torque* 2,2 N·m, ukuran sekitar 57×82 mm, dan diameter poros 8 mm [28]. Contoh visual motor NEMA 23 dengan dimensi dan karakteristik yang sesuai dengan varian 57HS8230A4D8 ditunjukkan pada Gambar 2.9, sedangkan spesifikasi contoh varian tersebut dirangkum pada Tabel 2.4 [28], [39].



Gambar 2. 9 Contoh motor *stepper* NEMA 23 varian 57HS8230A4D8 [39]

Tabel 2. 4 Contoh spesifikasi motor *stepper* NEMA 23 G-Penny [28]

Parameter	Spesifikasi
Merek/tipe	G-Penny NEMA 23 CNC Stepper Motor
Jumlah fasa	2 fasa
<i>Step angle</i>	$1,8^\circ/\text{step}$

Parameter	Spesifikasi
Arus nominal	3 A
<i>Holding torque</i>	2,2 N·m
Ukuran motor	sekitar 57 × 82 mm
Diameter poros	8 mm

2.9. Driver TB6600

Driver TB6600 merupakan modul pengendali motor *stepper* bipolar dua fasa yang berfungsi sebagai penghubung antara sinyal kendali digital dan kumparan motor. *Driver* menerima perintah langkah dan arah dari pengendali, kemudian mengatur arus pada fasa motor menggunakan metode chopper agar motor dapat bergerak secara bertahap [22].

Driver TB6600 umumnya menyediakan pengaturan arus keluaran dan resolusi *microstepping* melalui DIP switch. *Microstepping* membagi satu *full-step* menjadi beberapa bagian yang lebih kecil, sehingga resolusi gerak meningkat dan perubahan posisi dapat berlangsung lebih halus. Pemilihan level arus disesuaikan dengan karakteristik motor yang dikendalikan [22].

Sinyal PUL atau STEP menentukan jumlah langkah motor, DIR menentukan arah putaran, sedangkan ENABLE digunakan untuk mengaktifkan atau menonaktifkan keluaran *driver*. Dengan ketiga sinyal tersebut, pengendali dapat menentukan arah, besar perpindahan, dan kondisi aktif motor *stepper*. Bentuk modul *driver* TB6600 yang dibahas ditunjukkan pada Gambar 2.10, sedangkan karakteristik umumnya dirangkum pada Tabel 2.5.



Gambar 2. 10 Modul *driver* motor *stepper* TB6600 [34]

Tabel 2. 5 Karakteristik umum *driver* TB6600 [34]

Parameter	Keterangan umum
Jenis perangkat	<i>Driver</i> motor stepper bipolar dua fasa
Sinyal kendali	PUL/STEP, DIR, dan ENABLE
Metode pengaturan arus	<i>Current chopper</i> / pengendalian arus fasa
<i>Microstepping</i>	Pembagian <i>full-step</i> menjadi beberapa bagian yang lebih kecil
Pengaturan	Level arus dan <i>microstep</i> umumnya dipilih melalui DIP switch
Fungsi utama	Mengubah sinyal logika pengendali menjadi arus penggerak kumparan motor
Aplikasi	Pengendalian posisi dan kecepatan motor stepper berbasis pulsa

2.9.1. Sinyal STEP/PUL, Direction, dan *Microstepping* pada Motor Stepper

Pengendalian motor *stepper* melalui *driver* TB6600 dilakukan menggunakan sinyal digital STEP/PUL, DIR, dan ENABLE. Sinyal STEP/PUL mewakili perpindahan langkah, DIR menentukan arah putaran, dan ENABLE mengendalikan keadaan keluaran *driver*. Hubungan jumlah pulsa, *microstepping*, dan resolusi sudut digunakan untuk mengonversi target sudut menjadi perintah gerak motor [22].

Microstepping membagi satu *full-step* menjadi bagian. Karena setiap pulsa STEP mewakili satu *microstep*, resolusi sudut motor untuk satu pulsa terlebih dahulu ditentukan dari sudut *full-step* [22].

$$\Delta\theta_m = \frac{\theta_s}{M} \quad (2.12)$$

Keterangan Persamaan (2.12):

$\Delta\theta_m$ = resolusi sudut poros motor untuk satu pulsa STEP pada mode *microstepping* ($^{\circ}/pulse$)

θ_s = sudut langkah dasar motor untuk satu *full-step* ($^{\circ}/full-step$)

M = faktor *microstepping* (*microstep/full-step*).

N_p Jika motor menerima pulsa, sudut total poros motor merupakan hasil perkalian jumlah pulsa dengan resolusi satu *microstep*. Dengan mensubstitusikan Persamaan (2.12), diperoleh:

$$\theta_m = N_p \cdot \Delta\theta_m = N_p \cdot \frac{\theta_s}{M} \quad (2.13)$$

Keterangan Persamaan (2.13):

θ_m = sudut putaran poros motor *stepper* (°)

N_p = jumlah pulsa STEP/PUL yang diterima *driver* (*pulse*)

$\Delta\theta_m$ = resolusi sudut poros motor untuk satu pulsa STEP (°/*pulse*)

θ_s = sudut *full-step* motor (°)

M = faktor *microstepping*.

Jumlah *full-step* dalam satu putaran diperoleh dari perbandingan satu putaran penuh terhadap sudut *full-step*:

$$N_s = \frac{360^\circ}{\theta_s} \quad (2.14)$$

Keterangan Persamaan (2.14):

N_s = jumlah *full-step* dalam satu putaran (step/rev)

360° = satu putaran penuh

θ_s = sudut langkah dasar motor (°/*full-step*).

Pada mode *microstepping*, jumlah pulsa untuk satu putaran menjadi:

$$N_{rev} = N_s \cdot M \quad (2.15)$$

Keterangan Persamaan (2.15):

N_{rev} = jumlah pulsa STEP yang diperlukan untuk satu putaran motor pada mode *microstepping* (*pulse/rev*)

N_s = jumlah *full-step* per putaran

M = faktor *microstepping*.

$$n_m = 60 \cdot \frac{f_p}{N_{rev}} \quad (2.16)$$

Keterangan Persamaan (2.16):

n_m = kecepatan putar motor (rpm)

f_p = frekuensi pulsa STEP (*pulse/s* atau Hz)

N_{rev} = jumlah pulsa per putaran (*pulse/rev*)

faktor 60 mengonversi detik menjadi menit.

2.10. Motor Servo DS3225

DS3225 merupakan RC digital servo yang dikendalikan menggunakan sinyal PWM. Datasheet resmi DSSERVO mencantumkan rentang tegangan operasi 4,8-6,8 V, ukuran $40 \times 20 \times 40,5$ mm, massa sekitar 60 g, dan tingkat perlindungan IP66 [23].

Pada 5 V, DS3225 memiliki kecepatan tanpa beban sekitar 0,15 s/60° dan *stall torque* 21 kg·cm; pada 6,8 V, kecepatan tanpa beban sekitar 0,13 s/60° dan *stall torque* 24,5 kg·cm. Arus stall masing-masing sekitar 1,9 A dan 2,3 A [23].

Sinyal kontrol DS3225 menggunakan *pulse width* 500-2500 μ s dengan posisi netral 1500 μ s, *deadband* 3 μ s, dan frekuensi operasi 50-330 Hz. Perubahan lebar pulsa digunakan untuk memberikan perintah posisi sudut pada servo di dalam rentang gerakanya [23]. Bentuk servo DS3225 yang dilengkapi horn 25T ditunjukkan pada Gambar 2.11, sedangkan spesifikasi listrik dan mekaniknya dirangkum pada Tabel 2.6.



Gambar 2. 11 Motor servo DS3225 dengan servo horn 25T [35]

Tabel 2. 6 Spesifikasi Motor Servo DS3225 [23]

Parameter	Spesifikasi
Model	DS3225 RC Digital Servo
Tegangan operasi	4,8-6,8 V
Ukuran	$40 \times 20 \times 40,5$ mm
Massa	60 g
<i>Stall torque</i> @ 5 V	21 kg·cm
<i>Stall torque</i> @ 6,8 V	24,5 kg·cm

Parameter	Spesifikasi
Kecepatan @ 5 V	0,15 s/60°
Kecepatan @ 6,8 V	0,13 s/60°
Stall current @ 5 V / 6,8 V	1,9 A / 2,3 A
Pulse width	500-2500 μ s
Frekuensi kontrol	50-330 Hz
Deadband	3 μ s
Perlindungan	IP66

2.10.1. Pulse Width Modulation (PWM) pada Servo DS3225

Pulse Width Modulation (PWM) merupakan metode pembentukan sinyal periodik dengan mengatur lamanya kondisi aktif (HIGH) dan tidak aktif (LOW) dalam satu siklus. Sinyal PWM dapat dijelaskan melalui frekuensi, periode, *pulse width*, dan *duty cycle*. Pada servo posisi, informasi utama perintah sudut direpresentasikan oleh lebar pulsa aktif yang dikirim secara berulang, sedangkan frekuensi menentukan seberapa sering pulsa tersebut diulang [23].

Frekuensi PWM menyatakan jumlah siklus sinyal dalam satu detik, sedangkan periode PWM menyatakan waktu yang diperlukan untuk menyelesaikan satu siklus. Karena keduanya saling berbalik, hubungan periode dan frekuensi dinyatakan pada Persamaan (2.17).

$$T_{PWM} = \frac{1}{f_{PWM}} \quad (2.17)$$

Keterangan Persamaan (2.17):

T_{PWM} = periode pengulangan sinyal PWM (s)

f_{PWM} = frekuensi PWM (Hz).

$$D_{PWM} = \left(\frac{t_p}{T_{PWM}} \right) \times 100\% \quad (2.18)$$

Keterangan Persamaan (2.18):

D_{PWM} = *duty cycle* sinyal PWM (%)

t_p = lebar pulsa aktif/*pulse width* (s atau μ s)

T_{PWM} = periode PWM (s).

Jika hubungan antara sudut perintah dan lebar pulsa diasumsikan linier di antara dua batas perintah, lebar pulsa untuk suatu sudut dapat ditentukan melalui interpolasi linier seperti ditunjukkan pada Persamaan (2.19).

$$t_p = t_{min} + \left(\frac{(\theta_{cmd} - \theta_{min})}{(\theta_{max} - \theta_{min})} \right) \cdot (t_{max} - t_{min}) \quad (2.19)$$

Keterangan Persamaan (2.19):

t_p = lebar pulsa perintah servo (μs)

t_{min} dan t_{max} = lebar pulsa pada batas minimum dan maksimum perintah (μs)

θ_{cmd} = sudut yang diperintahkan ($^\circ$)

θ_{min} dan θ_{max} = batas minimum dan maksimum sudut pada model kendali servo ($^\circ$).

2.11. Sensor Hall A3144

A3144 merupakan *Hall-effect switch* unipolar yang digunakan untuk mendeteksi keberadaan medan magnet. Keluaran sensor berubah keadaan ketika medan magnet melewati nilai *operate point* dan kembali ketika medan turun melewati *release point*. Karakteristik tersebut memungkinkan sensor Hall digunakan sebagai pendeteksi posisi tanpa kontak mekanis [24].

Datasheet Allegro mencantumkan rentang tegangan operasi 4,5-24 V, kemampuan sink keluaran hingga 25 mA, serta keluaran *open-collector* yang memerlukan resistor pull-up. Histeresis antara *operate point* dan *release point* membantu mencegah perubahan logika berulang ketika medan magnet berada di sekitar ambang *switching* [24].

Dalam aplikasi penentuan posisi referensi, sensor Hall dapat dipasangkan dengan magnet pada bagian yang bergerak. Ketika magnet memasuki daerah aktivasi sensor, perubahan logika dapat dibaca oleh pengendali sebagai penanda posisi tertentu. Karena proses deteksi tidak memerlukan kontak mekanis, sensor Hall sesuai digunakan pada mekanisme yang membutuhkan pendeteksian posisi berulang [24]. Bentuk fisik sensor A3144 kemasan TO-92 ditunjukkan pada Gambar 2.12, sedangkan spesifikasi kelistrikannya dirangkum pada Tabel 2.7.



Gambar 2. 12 Sensor Hall A3144 dalam kemasan TO-92 [36]

Tabel 2. 7 Spesifikasi Sensor Hall A3144 [24]

Parameter	Spesifikasi
Tipe sensor	A3144 unipolar Hall-effect switch
Tegangan operasi	4,5-24 V
Jenis keluaran	Open-collector
Arus keluaran kontinu maksimum	25 mA
Tegangan saturasi keluaran	maks. 400 mV pada IO _{UT} = 20 mA
Arus suplai	tip. 4,4 mA; maks. 9 mA
Rentang temperatur	bergantung suffix: hingga -40...+85 °C atau -40...+150 °C
Karakter switching	Unipolar Hall switch dengan operate point, release point, dan histeresis

2.12. Sistem *Homing* Lengan Robot

Homing adalah proses menentukan posisi referensi awal suatu sumbu atau *joint* sebelum perintah posisi dijalankan. Pada sistem *stepper open-loop* tanpa *encoder* absolut, posisi mekanik setelah power-on tidak dapat diketahui hanya dari pencacahan pulsa sebelumnya sehingga diperlukan proses *homing* untuk membentuk acuan posisi yang konsisten [15].

Proses *homing* umumnya dilakukan dengan menggerakkan aktuator menuju sensor referensi, seperti limit switch atau sensor Hall. Setelah sensor aktif, gerakan dihentikan dan nilai posisi ditetapkan sesuai definisi koordinat pengendali. Setelah titik referensi diperoleh, perintah posisi berikutnya dapat dihitung relatif terhadap acuan tersebut.

2.13. *Timing Belt* HTD5M

Timing belt HTD5M merupakan *synchronous belt* dengan profil gigi curvilinear dan *pitch* 5 mm. Profil gigi memungkinkan belt berkaitan dengan *pulley*

secara sinkron sehingga perpindahan putaran dapat berlangsung tanpa slip relatif antara belt dan *pulley*. Karakteristik geometris profil HTD5M mencakup *pitch* 5 mm, tinggi belt sekitar 3,8 mm, dan tinggi gigi sekitar 2,1 mm [25].

Timing belt HTD5M tersedia dalam berbagai panjang dan lebar. Pemilihan lebar belt dipengaruhi oleh besar beban yang ditransmisikan, diameter *pulley*, kecepatan, serta faktor layanan. Lebar yang lebih besar umumnya memberikan kemampuan membawa beban yang lebih tinggi, selama tetap sesuai dengan *pulley* yang digunakan [25]. Karakteristik umum profil *timing belt* HTD5M yang digunakan sebagai dasar pemilihan ditunjukkan pada Tabel 2.8.

Tabel 2. 8 Karakteristik umum *timing belt* HTD5M [25]

Parameter	Spesifikasi umum
Profil	HTD 5M, gigi curvilinear
<i>Pitch</i>	5 mm
Tinggi belt	sekitar 3,8 mm
Tinggi gigi	sekitar 2,1 mm
Lebar belt	Tersedia dalam beberapa ukuran sesuai kebutuhan transmisi

2.14. Timing Pulley HTD5M

Timing *pulley* HTD5M merupakan *pulley* bergigi yang dirancang berpasangan dengan *timing belt* berprofil HTD5M dan *pitch* 5 mm. Parameter utama *pulley* meliputi jumlah gigi, *pitch* diameter, diameter *bore*, dan lebar muka *pulley*. Parameter tersebut menentukan kesesuaian mekanik *pulley* dengan poros, belt, serta rasio transmisi yang dibentuk.

Kesesuaian *pitch* dan profil gigi antara *pulley* dan belt diperlukan agar transmisi sinkron dapat bekerja dengan benar. Jumlah gigi memengaruhi *pitch* diameter dan rasio putaran, diameter *bore* ditentukan berdasarkan ukuran poros, sedangkan lebar muka *pulley* disesuaikan dengan lebar belt. Parameter umum timing *pulley* HTD5M dirangkum pada Tabel 2.9.

Tabel 2. 9 Parameter umum timing *pulley* HTD5M

Parameter	Nilai/Symbol	Fungsi	Keterangan
<i>Pitch</i>	5 mm	Kesesuaian profil gigi	Sama dengan <i>pitch</i> belt
Jumlah gigi	Z	Menentukan rasio transmisi	Dipilih sesuai kebutuhan
Bore / lebar	db / b	Menyesuaikan poros dan belt	Sesuai dimensi pasangan

2.15. Rasio Transmisi *Timing Belt* dan *Pulley*

Rasio transmisi *timing belt-pulley* menentukan hubungan antara gerak motor dan gerak *joint*. Untuk pasangan *pulley* dengan *pitch* yang sama, rasio reduksi dapat dinyatakan dari perbandingan jumlah gigi *pulley* driven terhadap *pulley driver*. Hubungan ini digunakan untuk menghitung perubahan kecepatan, torsi ideal, dan resolusi sudut keluaran [25].

$$i = \frac{Z_{driven}}{Z_{driver}} \quad (2.20)$$

Keterangan Persamaan (2.20):

i = rasio reduksi transmisi, tidak bersatuan

Z_{driven} = jumlah gigi *pulley* pada sisi *joint*/driven

Z_{driver} = jumlah gigi *pulley* pada poros motor/*driver*.

$$r_p = \frac{1}{i} = \frac{Z_{driver}}{Z_{driven}} = \frac{\theta_{axis}}{\theta_{motor}} \quad (2.21)$$

Keterangan Persamaan (2.21):

r_p = rasio sudut *output axis* terhadap sudut motor, tidak bersatuan

i = rasio reduksi

Z_{driver} = jumlah gigi *pulley driver*

Z_{driven} = jumlah gigi *pulley driven*

θ_{axis} = sudut sumbu keluaran

θ_{motor} = sudut poros motor.

Kecepatan keluaran ideal turun sebesar rasio i :

$$n_{out} = \frac{n_{in}}{i} \quad (2.22)$$

Keterangan Persamaan (2.22):

n_{out} = kecepatan putar sumbu keluaran (rpm)

n_{in} = kecepatan putar poros motor (rpm)

i = rasio reduksi transmisi.

Dengan efisiensi mekanik η , estimasi torsi keluaran meningkat sesuai rasio reduksi:

$$\tau_{out} = \eta \cdot i \cdot \tau_{in} \quad (2.23)$$

Keterangan Persamaan (2.23):

τ_{out} = estimasi torsi pada sumbu keluaran (N·m)

η = efisiensi mekanik transmisi (0-1)

i = rasio reduksi

τ_{in} = torsi masukan dari motor (N·m).

Resolusi sudut *joint* untuk satu pulsa STEP diperoleh dengan mengalikan resolusi satu microstep pada Persamaan (2.12) dengan rasio sudut pada Persamaan (2.21)[25].

$$\Delta\theta_{axis} = \Delta\theta_m \cdot r_p = \left(\frac{\theta_s}{M}\right) \cdot \left(\frac{Z_{driver}}{Z_{driven}}\right) \quad (2.24)$$

Keterangan Persamaan (2.24):

$\Delta\theta_{axis}$ = resolusi sudut sumbu keluaran per pulsa ($^{\circ}/pulse$)

$\Delta\theta_m$ = resolusi sudut motor per *microstep* ($^{\circ}/pulse$)

r_p = rasio sudut *axis*/motor

θ_s = sudut *full-step* motor ($^{\circ}$)

M = faktor *microstepping*

Z_{driver} dan Z_{driven} = jumlah gigi *pulley driver* dan *driven*.

N_p Sudut *joint* akibat pulsa selanjutnya menjadi:

$$\theta_{joint} = N_p \cdot \Delta\theta_{axis} \quad (2.25)$$

Keterangan Persamaan (2.25):

θ_{joint} = besar perubahan sudut *joint* akibat pulsa yang diberikan ($^{\circ}$)

N_p = jumlah pulsa STEP

$\Delta\theta_{axis}$ = resolusi sudut sumbu keluaran per pulsa ($^{\circ}/pulse$).

Untuk menentukan jumlah pulsa yang diperlukan dari perubahan sudut *joint* yang diperintahkan, Persamaan (2.25) disusun ulang sehingga diperoleh:

$$N_p = \frac{|\Delta\theta_{target}|}{\Delta\theta_{axis}} \quad (2.26)$$

Keterangan Persamaan (2.26):

N_p = jumlah pulsa STEP yang diperlukan (*pulse*)

$\Delta\theta_{target}$ = besar perubahan sudut *joint* yang diperintahkan ($^{\circ}$)

$\Delta\theta_{axis}$ = resolusi sudut sumbu keluaran per pulsa ($^{\circ}/pulse$).

2.16. Torsi Lengan Robot

Kebutuhan torsi statis *joint* ditentukan oleh gaya berat komponen dan jarak tegak lurus garis kerja gaya terhadap sumbu rotasi. Besaran torsi merupakan hasil kali gaya dengan lengan momen, dengan satuan SI newton meter (N·m). Nilai percepatan gravitasi standar yang digunakan pada perhitungan adalah 9,80665 m/s² [29].

$$F_g = m \cdot g \quad (2.27)$$

Keterangan Persamaan (2.27):

F_g = gaya berat beban atau komponen (N)

m = massa beban atau komponen (kg)

g = percepatan gravitasi (m/s²), digunakan sekitar 9,81 m/s².

$$\tau_{joint} = F_g \cdot l_{\perp} \quad (2.28)$$

Keterangan Persamaan (2.28):

τ_{joint} = torsi statis terhadap sumbu *joint* (N·m)

F_g = gaya berat beban/komponen (N)

$l_{\perp}$ = jarak tegak lurus garis kerja gaya terhadap sumbu *joint* (m).

Pada *joint* dengan sumbu rotasi vertikal, gaya gravitasi tidak menghasilkan momen statis terhadap sumbu rotasi dengan cara yang sama seperti pada *joint* bersumbu horizontal. Torsi pada sumbu vertikal lebih berkaitan dengan momen inersia massa yang diputar, percepatan sudut, gesekan, dan rugi transmisi. Sebaliknya, pada *joint* dengan sumbu rotasi horizontal, gaya berat dapat menghasilkan momen statis yang besarnya bergantung pada massa dan jarak tegak lurus pusat beban terhadap sumbu rotasi.

2.17. Catu Daya dan *Buck Converter* XL4016

Daya listrik DC setiap beban dihitung dari hasil kali tegangan dan arus. Nilai ini digunakan untuk memperkirakan kebutuhan daya keseluruhan sistem sebelum menentukan rating sumber dan konverter daya.

$$P_i = V_i \cdot I_i \quad (2.29)$$

Keterangan Persamaan (2.29):

P_i = daya listrik komponen ke- i (W)

V_i = tegangan kerja komponen ke- i (V)

I_i = arus kerja komponen ke- i (A)

i = indeks komponen.

Kebutuhan daya total sistem merupakan penjumlahan seluruh beban:

$$P_{total} = \sum P_i = \sum (V_i \cdot I_i) \quad (2.30)$$

Keterangan Persamaan (2.30):

P_{total} = estimasi daya total beban (W)

P_i = daya komponen ke- i (W)

V_i = tegangan kerja komponen ke- i (V)

I_i = arus kerja komponen ke- i (A)

$\sum$ = operasi penjumlahan seluruh beban yang diperhitungkan

i = indeks komponen.

XL4016 merupakan integrated circuit buck DC-DC converter yang digunakan untuk menurunkan tegangan DC menjadi level keluaran yang lebih rendah. Datasheet XLSEMI mencantumkan frekuensi *switching* tetap sekitar 180 kHz, rentang tegangan operasi yang direkomendasikan 8-36 V, dan tegangan keluaran adjustable sekitar 1,25-32 V [26].

IC XL4016 memiliki kemampuan arus keluaran hingga 12 A pada kondisi desain yang memenuhi persyaratan termal dan komponen eksternal, efisiensi hingga sekitar 96%, serta proteksi current limit, thermal shutdown, dan short-circuit. Rating pada modul komersial dapat lebih rendah karena dipengaruhi *heatsink*, induktor, diode, konektor, dan tata letak PCB [26].

Pada rangkaian *buck converter*, tegangan keluaran diatur melalui jaringan umpan balik, sedangkan kemampuan arus harus disesuaikan dengan kebutuhan

beban dan kondisi termal. Efisiensi konverter dipengaruhi oleh tegangan masukan, tegangan keluaran, arus beban, komponen eksternal, serta rancangan pelepasan panas [26]. Contoh modul *buck converter* berbasis XL4016 ditunjukkan pada Gambar 2.13, sedangkan karakteristik IC XL4016 dirangkum pada Tabel 2.10.



Gambar 2. 13 Modul *buck converter* berbasis XL4016 [37]

Tabel 2. 10 Karakteristik IC XL4016 [26]

Parameter	Spesifikasi XL4016
Tipe	Buck DC-DC converter
Rentang tegangan operasi yang direkomendasikan	8-36 V DC
Tegangan keluaran adjustable	sekitar 1,25-32 V DC
Frekuensi switching	180 kHz
Kemampuan arus keluaran IC	hingga 12 A dengan desain termal yang sesuai
Efisiensi maksimum	hingga sekitar 96%
Minimum dropout	sekitar 0,3 V
Proteksi internal	current limit, thermal shutdown, short-circuit protection

2.18. TFT 2,8 Inch ILI9341 dan XPT2046

Antarmuka lokal menggunakan modul TFT 2,8inch dengan pengendali tampilan ILI9341 dan pengendali sentuh resistif XPT2046. ILI9341 merupakan *driver* TFT dengan resolusi $240\text{RGB} \times 320$ dan dukungan hingga 262K warna, serta menyediakan antarmuka paralel maupun serial 3-wire/4-wire [27]. XPT2046

merupakan controller layar sentuh resistif 4-wire dari XPTEK dengan resolusi ADC 12-bit dan antarmuka SPI [17].

ILI9341 menyediakan antarmuka paralel maupun serial untuk menerima data tampilan, sedangkan XPT2046 menggunakan antarmuka serial untuk membaca koordinat sentuhan dari panel resistif. Pada modul TFT, komunikasi SPI sering digunakan karena hanya memerlukan beberapa jalur data dan dapat dihubungkan dengan berbagai jenis mikrokontroler [17], [27].

Modul TFT dengan layar sentuh dapat digunakan sebagai antarmuka manusia-mesin untuk menampilkan informasi, status, menu, grafik, maupun menerima masukan sentuhan. *Breakout module* dapat menambahkan regulator tegangan, level shifter, slot memori, atau rangkaian pendukung lain, sehingga karakteristik kelistrikan modul perlu mengikuti skematik dan dokumentasi perangkat yang digunakan [17], [27]. Bentuk modul TFT 2,8inch yang menggunakan ILI9341 dan XPT2046 ditunjukkan pada Gambar 2.14, sedangkan spesifikasi kedua controller dirangkum pada Tabel 2.11.



Gambar 2. 14 Modul TFT 2,8inch berbasis ILI9341 dan XPT2046 [38]

Tabel 2. 11 Spesifikasi pengendali ILI9341 dan XPT2046 [17], [27]

Parameter	Spesifikasi
Ukuran modul	2,8inch
Pengendali tampilan	ILI9341
Resolusi <i>display controller</i>	240RGB × 320 piksel
Kedalaman warna ILI9341	hingga 262K warna
Antarmuka serial ILI9341	3-wire / 4-wire serial interface
Pengendali sentuh	XPT2046 (XPTEK)

Parameter	Spesifikasi
Tipe touch	4-wire resistive
Resolusi ADC XPT2046	12-bit
Tegangan suplai XPT2046	2,2-5,25 V
Antarmuka XPT2046	SPI

2.19. Ringkasan Dasar Komponen dan Kendali Lengan Robot

Pada sistem robot, perangkat komputasi tingkat tinggi, mikrokontroler, sensor, *driver*, aktuator, transmisi mekanik, dan antarmuka pengguna memiliki fungsi yang saling melengkapi. Komputer *embedded* dapat menangani pemrosesan data dan pembentukan target, sedangkan mikrokontroler menjalankan fungsi kendali waktu nyata dan menghasilkan sinyal untuk *driver* atau servo. Sensor memberikan informasi keadaan atau posisi referensi, transmisi mekanik meneruskan gerak dan torsi, sedangkan antarmuka pengguna menampilkan informasi serta menerima perintah. Koordinasi antarkomponen tersebut membentuk rantai proses dari informasi target hingga gerakan mekanik pada manipulator.