

# PENERAPAN KINEMATIKA TERBALIK PADA ROBOT LENGAN LIMA SENDI (5 DOF) DENGAN CITRA DIGITAL

Fahril Maula Tanzil Huda<sup>1\*</sup>, Yusril Ali Riza Pratama<sup>2</sup>, Fauzan Ra'is Saputra<sup>3</sup>, Rizanurfadli Hadiazzaka<sup>4</sup>, Ardy Seto Priambodo<sup>5</sup>

<sup>1,2,3,4,5</sup> Departemen Elektro dan Elektronika, Universitas Negeri Yogyakarta; Jl.Mandung Pengasih Kulonprogo, Telp. & Fax. (0274) 773906.

Received: 15 Desember 2024  
Accepted: 14 Januari 2025  
Published: 20 Januari 2025

## Keywords:

Robot lengan 5 DOF,  
Kinematika terbalik, Citra digital, Deteksi objek

## Correspondent Email:

fahrilmaula.2022@student.uny.ac.id

**Abstrak.** Perkembangan teknologi yang semakin maju memberi dampak pada sistem kendali robot, salah satunya ialah robot lengan. Sistem kendali yang sering digunakan pada lengan robot adalah kinematika terbalik dan kinematika maju. Penelitian ini membahas penerapan kinematika terbalik pada robot lengan dengan lima derajat kebebasan (5 DOF) yang diintegrasikan dengan teknologi citra digital untuk meningkatkan presisi dan efisiensi operasional. Deteksi objek dan warna dilakukan melalui pemrosesan citra yang digunakan sebagai masukan untuk algoritma kinematika terbalik. Dengan metode tersebut didapatkan keluaran berupa derajat untuk masing-masing aktuator. Berdasarkan hasil pengujian yang dilakukan kombinasi kinematika terbalik dan citra digital masih memiliki error yang cukup besar, seperti pada sumbu x dengan error tertingginya pada 4.11, sumbu y dengan error tertingginya 13 dan sumbu z dengan error 26.74. Meskipun nilai error masih cukup besar tetapi metode ini berhasil mengendalikan robot lengan 5-DOF.

**Abstract.** The development of advanced technology has an impact on robot control systems, one of which is the robot arm. Control systems that are often used on robot arms are inverse kinematics and forward kinematics. This research discusses the application of inverse kinematics to a robot arm with five degrees of freedom (5 DOF) integrated with digital image technology to improve operational precision and efficiency. Object and color detection is done through image processing which is used as input for the inverse kinematics algorithm. With this method, the output is the degree for each actuator. Based on the test results, the combination of inverse kinematics and digital images still has a fairly large error, such as on the x-axis with the highest error at 4.11, the y-axis with the highest error of 13 and the z-axis with an error of 26.74. Although the error value is still large, this method successfully controls the 5-DOF robot arm.

## 1. PENDAHULUAN

Robot telah berkembang secara signifikan dalam beberapa dekade terakhir. Robot kini menjadi bagian penting dalam berbagai industri, seperti manufaktur, kesehatan, dan transportasi. Kemampuan robot untuk bekerja secara berkelanjutan, meningkatkan kualitas produk, dan menjaga kinerja yang stabil

menjadikan robot sangat bermanfaat di berbagai bidang.

Pada industri, salah satu penggunaan robot adalah untuk mengambil dan meletakkan sebuah objek yang mungkin sulit dijangkau[1]. Robot dapat bekerja secara mandiri atau dapat digabungkan dalam jalur produksi untuk melakukan tugas kompleks yang melibatkan

banyak tahapan[2]. Salah satu robot yang sering digunakan di industri adalah robot lengan.

Robot lengan adalah robot yang dapat digunakan dalam berbagai pekerjaan pada industri[3]. Robot lengan dirancang untuk meniru pergerakan dan fungsi lengan manusia, sehingga cara kerjanya menyerupai tangan asli. Namun, robot lengan memiliki keunggulan yang tidak dimiliki manusia, seperti kemampuan untuk menghasilkan kualitas yang konsisten saat melakukan pekerjaan berulang, tidak mengenal lelah, dan dapat diprogram ulang untuk menjalankan berbagai tugas. Misalnya, robot lengan dapat digunakan untuk mendeteksi objek dan posisi dengan akurasi tinggi. Kemampuan ini menjadikannya sangat efisien, terutama dalam bidang industri. Hal ini dirasakan sangat efisien khususnya dalam bidang industri[4].

Dalam pengoperasiannya, robot lengan memerlukan sistem kontrol yang kompleks untuk dapat bekerja secara presisi, terutama saat harus memindahkan objek ke posisi yang ditentukan. Salah satu algoritma yang sering digunakan adalah kinematika terbalik (*inverse kinematics*). Kinematika terbalik adalah algoritma untuk mengendalikan ujung efektor secara langsung untuk menempatkannya pada lokasi yang diinginkan[5]. Penerapan kinematika terbalik memungkinkan robot untuk bekerja dengan efisiensi dan akurasi tinggi, khususnya pada robot lengan dengan derajat kebebasan (DOF) yang banyak, seperti robot lengan 5 DOF. Namun, penerapan kinematika terbalik pada robot lengan, khususnya dengan 5 derajat kebebasan (DOF), menghadapi sejumlah tantangan. Salah satu masalah utama adalah adanya beberapa solusi untuk konfigurasi sendi yang memenuhi posisi dan orientasi ujung efektor tertentu. Hal ini dapat menyulitkan pemilihan solusi optimal, terutama pada lingkungan yang dinamis[6][7].

Beberapa penelitian mencoba mengatasi kendala ini melalui pengembangan algoritma yang lebih adaptif dan efisien, namun aplikasi praktisnya masih memerlukan evaluasi lebih lanjut. Dengan demikian, penelitian ini bertujuan untuk mengintegrasikan algoritma kinematika terbalik dengan teknologi citra digital untuk meningkatkan presisi dan kemudahan penggunaan dalam pengoperasian robot lengan.

## 2. TINJAUAN PUSTAKA

### 2.1 Robot Lengan

Robot lengan atau sering disebut sebagai manipulator robotik adalah sistem mekanik yang dirancang untuk melakukan tugas-tugas seperti memindahkan, memegang, atau memanipulasi objek dalam lingkungan tertentu. Robot lengan terdiri dari beberapa bagian utama seperti segmen (*link*), sambungan (*joint*), dan aktuator, yang bekerja bersama untuk menghasilkan gerakan yang presisi. Gerakan ini biasanya diatur oleh kontroler yang mengelola sinyal listrik untuk menggerakkan motor atau aktuator. Robot lengan juga sering dilengkapi dengan *end-effector*, yaitu alat di ujung lengan yang dirancang sesuai dengan tugas tertentu, seperti penjepit, alat las, atau perangkat lain yang memungkinkan interaksi langsung dengan objek.

Pengendalian robot dapat dilakukan melalui berbagai metode, termasuk kinematika maju dan terbalik untuk menentukan posisi, serta algoritma kontrol yang memastikan gerakan lengan sesuai dengan kebutuhan operasional[13]. Derajat kebebasan (*Degree of Freedom*, DOF) pada robot lengan merujuk pada jumlah sumbu gerakan yang dimiliki oleh lengan tersebut. Semakin banyak DOF, semakin fleksibel robot lengan dalam menjangkau posisi dan orientasi tertentu di ruang tiga dimensi.

### 2.2 Kinematika Terbalik

Algoritma kinematika terbalik adalah metode analisis kinematika yang digunakan untuk mentransformasikan koordinat dari ruang kartesian ke ruang sendi robot[11]. Persamaan kinematika terbalik berfungsi untuk menentukan parameter dari setiap titik hubung pada robot agar dapat mencapai posisi yang diinginkan dan orientasi *gripper* yang diinginkan[12]. Kinematika terbalik mempermudah pengendalian gerakan robot lengan, karena posisi aktuator dapat dihitung berdasarkan posisi yang ditentukan. Selain itu, algoritma ini mengurangi waktu pengerjaan, karena tidak memerlukan pendekatan trial and error dalam menentukan posisi dan orientasi robot.

Algoritma kinematika terbalik juga memiliki beberapa tantangan. Salah satunya adalah transformasi posisi dan orientasi robot menjadi rotasi pada setiap titik hubung yang

membutuhkan perhitungan trigonometri invers untuk menyelesaikan persamaan - persamaan kinematika yang bersifat kompleks. Algoritma kinematika terbalik sangat bergantung pada desain mekanik robot dan persamaan yang digunakan. Gerakan robot menjadi terbatas dikarenakan desain robot dapat menyebabkan hilangnya beberapa derajat kebebasan dalam konfigurasi tertentu.

### 2.3 ESP32

ESP32 adalah mikrokontroler *System on Chip* (SoC) yang dirancang untuk mendukung berbagai aplikasi dengan fitur unggulan seperti *Wi-Fi* dan *Bluetooth*. Mikrokontroler ini memiliki prosesor terpadu, penyimpanan internal, serta akses ke GPIO, menjadikannya solusi serbaguna untuk proyek IoT dan otomatisasi. Tersedia dalam dua varian, 30 GPIO dan 36 GPIO, ESP32 menawarkan fleksibilitas dalam pengembangan perangkat dengan antarmuka *USB to UART* yang memudahkan pemrograman menggunakan *Arduino IDE*. Selain itu, *ESP32* memiliki efisiensi daya yang baik, kemudahan integrasi dengan perangkat keras lain, dan kemampuan menjalankan aplikasi *real-time*, sehingga cocok untuk berbagai keperluan teknis seperti pengembangan sistem pintar dan robotik[8].



Gambar 1. ESP32

### 2.4 ESP32-CAM

ESP32-CAM adalah modul berbasis mikrokontroler ESP32 yang dilengkapi dengan kamera, dirancang untuk aplikasi pemantauan dan kontrol otomatis yang terintegrasi. Modul ini mendukung konektivitas *Wi-Fi* (802.11 b/g/n) dan *Bluetooth* (v4.2 BR/EDR dan *Bluetooth Low Energy*), memungkinkan komunikasi dengan perangkat lain seperti *smartphone* atau *smartwatch*, serta pengiriman data dan notifikasi melalui jaringan *Wi-Fi*. Pengaturan ESP32-CAM dilakukan menggunakan perangkat *FTDI USB to TTL* untuk koneksi ke komputer dan pemrograman

melalui *Arduino IDE*. Dengan fitur-fitur unggulan dari ESP32, ESP32-CAM mendukung berbagai aplikasi otomatisasi, termasuk integrasi kamera untuk mendeteksi dan memproses data secara *real-time*[9].



Gambar 2. ESP32-CAM

### 2.5 Driver Stepper Motor A4988

A4988 adalah *driver* motor *stepper* yang dirancang dengan fitur *built-in translator*, mempermudah pengoperasian motor tanpa perlu mengatur urutan langkah secara manual. *Driver* ini tidak hanya mampu mengatur jumlah langkah yang dikirimkan ke motor *stepper*, tetapi juga memastikan tegangan dan arus yang stabil untuk operasi yang lancar. *Driver* ini mampu mengelola tegangan hingga 35V dan arus hingga 2A per fase, meskipun memiliki kapasitas arus yang lebih kecil dibanding beberapa *driver* lainnya, A4988 tetap menjadi pilihan populer karena efisiensinya dalam mendukung kinerja motor *stepper* pada berbagai kecepatan dan aplikasi seperti robotika dan mesin *CNC*. [10].



Gambar 3. Driver Stepper Motor A4988

### 2.6 Servo MG996R

Servo MG996R merupakan motor servo digital dengan sidit gerakan sebesar 180 derajat. Servo ini memiliki torsi sebesar 12 Kg/cm yang kompatibel dengan *board Arduino Uno*. MG996R adalah versi upgrade dari servo MG995 agar servo dapat bekerja lebih presisi. Servo MG996R menambahkan sistem kontrol *PCB* seperti *IC. Gearing* internal dan motornya juga ditingkatkan untuk meningkatkan *bandwith* mati dan pemusatan.



Gambar 4. Servo MG996R

### 2.7 NEMA 17HS4401S

Salah satu jenis motor *stepper 4-wire* yang memiliki 1,8 derajat per langkah untuk gerakan halus dan torsi penahan yang bagus. Motor ditentukan memiliki arus maksimal 1,7A/fase sehingga dapat digerakkan dengan mudah dengan pelindung motor umum untuk *Arduino* (atau driver motor lainnya) dan adaptor dinding atau baterai timbal-asam. Motor dilengkapi dengan kabel daya sepanjang 50 cm dengan konektor betina *Harwin 4-pin* yang sudah terpasang.

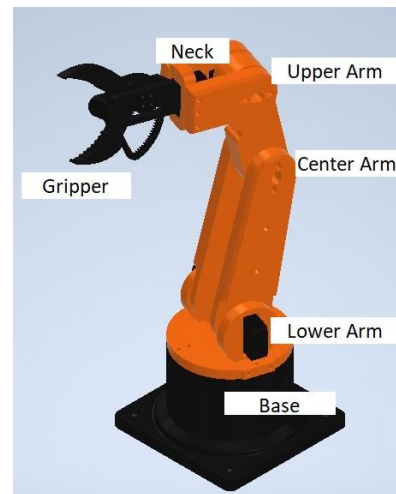


Gambar 5. NEMA 17HS4401S

## 3. METODE PENELITIAN

### 3.1 Perancangan Desain Mekanik Robot

Desain mekanik robot dirancang menggunakan perangkat lunak Autodesk Inventor 2024. Model 3D robot dikembangkan menjadi beberapa komponen utama, yaitu *base arm*, *lower arm*, *center arm*, *upper arm*, dan *gripper*. Dalam proses perancangan ini, ukuran setiap bagian harus diperhitungkan dengan cermat agar semua komponen dapat terhubung dan berfungsi dengan baik. Hal ini penting untuk memastikan kesesuaian dimensi dan kestabilan struktur mekanik robot secara keseluruhan. Gambar 6 adalah desain dan bagian dari robot lengan.



Gambar 6. Robot lengan

Tabel 1. Spesifikasi Mekanik Robot

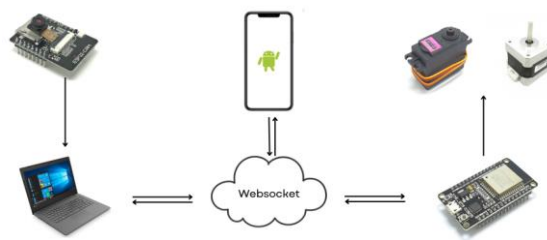
Segmen	Panjang (cm)	Derajat
Base	7	0 - 360°
Lower Arm	24	0 - 180°
Center Arm	17	0 - 180°
Upper Arm	11	0 - 180°
Neck	2	0 - 360°
Gripper	15	0 - 180°

### 3.2 Pembuatan Rangkaian Elektronik

Sistem Elektronik dirancang sebagai pendukung utama yang memastikan robot lengan dapat bekerja dengan stabil dan sesuai yang diinginkan. Komponen elektronik ini mencakup penyuplai daya, pengendali sinyal, dan instrumen pendukung lainnya yang bekerja secara terintegrasi untuk mengontrol setiap pergerakan robot lengan.

#### 3.2.1 Blok Diagram

Blok diagram dibuat untuk menentukan batasan rancangan seperti fitur yang akan dibuat serta alat dan bahan yang diperlukan untuk mendukung pembuatan robot lengan.



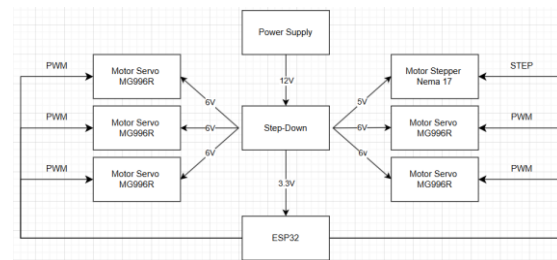
Gambar 7. Blok Diagram

Blok diagram pada Gambar 7 menjelaskan alur proses dimulai dari ESP32-CAM yang membaca citra. Citra yang diperoleh dikirimkan ke laptop melalui koneksi *Wi-Fi* untuk diproses lebih lanjut. Laptop kemudian memproses citra tersebut menggunakan bahasa Python, yang melibatkan tahap deteksi objek (kubus), identifikasi warna, serta penentuan lokasi objek. Lokasi objek dalam koordinat tiga sumbu (x, y, z) ini selanjutnya digunakan sebagai input untuk algoritma kinematika terbalik. Algoritma tersebut menghasilkan keluaran berupa besaran sudut dalam derajat untuk masing-masing aktuator (motor servo dan motor *stepper*) guna mengontrol pergerakan sistem.

Data hasil pengolahan dari laptop diteruskan ke ESP32 untuk diproses lebih lanjut dan digunakan sebagai dasar pengendalian aktuator pada sistem. Komunikasi antar perangkat dilakukan menggunakan metode *WebSocket*, yang menawarkan kemudahan dan fleksibilitas. Selain itu, sistem ini juga dilengkapi antarmuka yang dapat diakses dengan *Wi-Fi* lokal yang memungkinkan pengguna untuk mengatur posisi setiap aktuator pada robot lengan secara manual.

### 3.2.2 Perancangan Rangkaian

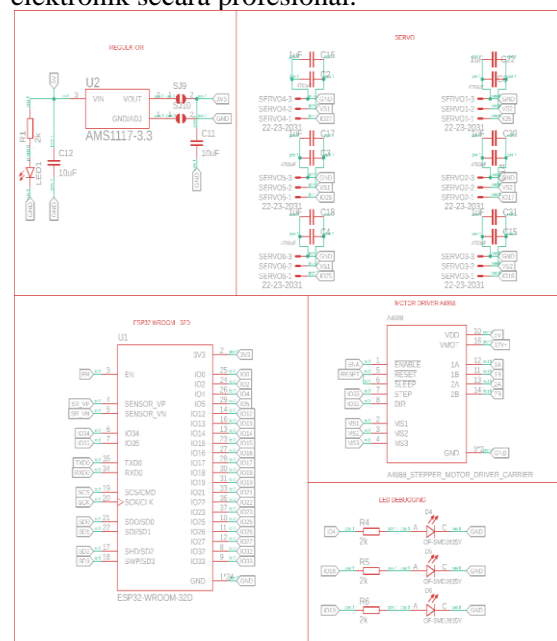
Berdasarkan rancangan sistem pada Gambar 7. Mikrokontroler yang digunakan untuk mengolah data adalah ESP32. ESP32 dapat bekerja dengan supply daya 3.0V - 3.6V yang sudah diturunkan menggunakan step-down AMS117 3.3V. Adapun *Power supply* 12VDC diturunkan tegangannya menjadi 6VDC melalui step down XL4015 yang akan menyuplai motor servo, sedangkan motor stepper akan mendapat tegangan dari step down AMS117 5V dengan menurunkan tegangan menjadi 5VDC sesuai spesifikasi kebutuhan motor stepper.



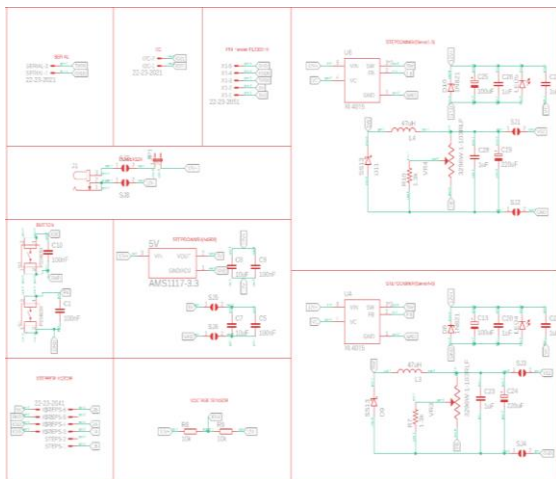
Gambar 8. Diagram Alir Perangkat Elektronik

Saat *power supply* dihidupkan, ESP32 akan mengirimkan sinyal PWM ke aktuator (motor servo dan motor *stepper*), sinyal tersebut berisi nilai informasi besaran sudut putar motor servo dan motor *stepper* yang dikirimkan oleh ESP32-CAM untuk menggerakkan sendi robot.

Pembuatan skematik rangkaian untuk robot lengan dimulai dengan riset mendalam mengenai komponen elektronik yang sesuai untuk mendukung pergerakan robot. Proses ini melibatkan analisis spesifikasi dan kinerja komponen agar memenuhi kebutuhan sistem secara efisien. Setelah menentukan komponen yang tepat, skematik dan tata letak PCB dirancang menggunakan perangkat lunak Eagle by Autodesk, yang menyediakan fitur lengkap untuk mendukung perancangan sirkuit elektronik secara profesional.

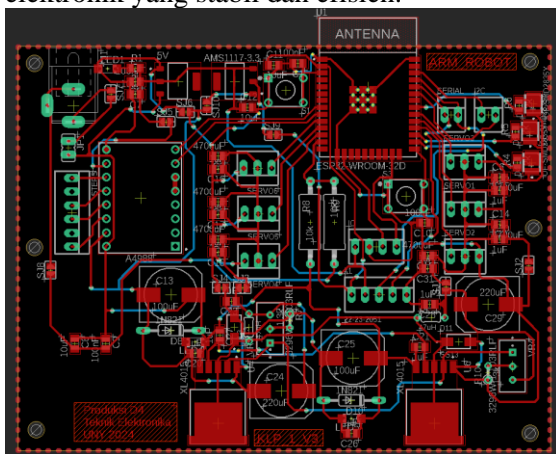






Gambar 9. Layout Schematic PCB

Pembuatan skematik PCB menggunakan beberapa komponen utama, yaitu ESP32 sebagai mikrokontroler, motor driver A4988 untuk mengontrol motor *stepper*, dan *step down* XL4015 untuk menurunkan tegangan. Pemilihan komponen ini didasarkan pada spesifikasi teknis yang sesuai untuk mendukung fungsi dan performa robot lengan yang dirancang. Komponen-komponen tersebut diintegrasikan untuk menciptakan sistem elektronik yang stabil dan efisien.



Gambar 10. Layout Board PCB

Perancangan layout PCB bertujuan untuk mempermudah pengaturan dan pengelolaan jalur rangkaian pada sistem robot lengan dengan 5 derajat kebebasan (5 DOF). Layout ini dirancang untuk menghubungkan mikrokontroler ESP32 dengan aktuator seperti motor servo dan *stepper* secara efisien, sehingga memungkinkan komunikasi yang lebih terorganisir dan stabil di antara komponen. Dengan pendekatan ini, proses

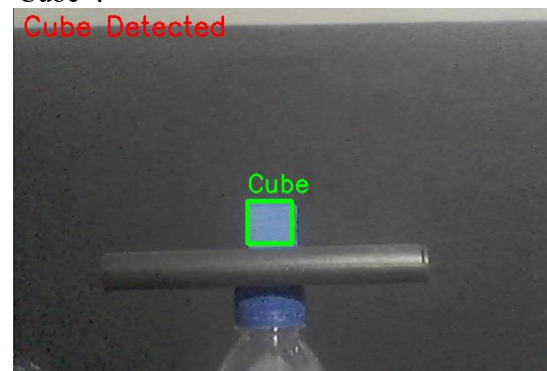
instalasi dan perawatan sistem menjadi lebih sederhana, mengurangi risiko kesalahan sambungan, serta meningkatkan keandalan dan performa keseluruhan robot dalam menjalankan tugas-tugas presisi tinggi. Layout PCB juga mendukung optimasi ruang dan integrasi komponen pada robot, menjadikannya solusi yang lebih praktis untuk aplikasi robotika modern.

### 3.3 Pengolahan Citra Digital

Citra digital adalah representasi visual dari suatu objek dalam bentuk matriks yang terdiri dari M kolom dan N baris. Setiap elemen pada matriks ini disebut piksel, yang merupakan elemen terkecil dari sebuah citra digital. Piksel memiliki dua parameter utama, yaitu koordinat (posisi) dan intensitas atau warna, yang bersama-sama merepresentasikan informasi visual pada citra tersebut[14]. Dalam penelitian ini, citra digital digunakan untuk mendeteksi target untuk robot lengan.

#### 3.3.1 Deteksi Objek

Deteksi objek dilakukan dengan bahasa pemrograman *Python* dengan pustaka utama yang digunakan yaitu OpenCV. Citra akan dibaca secara *real time* dan diolah secara *real time* juga. Setiap *frame* yang ditangkap diubah menjadi gambar *grayscale* agar lebih mudah diolah. Selanjutnya, algoritma *Canny* digunakan untuk mendeteksi tepi dengan membandingkan intensitas piksel, sehingga menghasilkan gambar berbentuk garis-garis tepi. Untuk meningkatkan hasil deteksi, dilakukan dilasi (memperbesar area tepi) dan erosi (menghilangkan derau) dengan kernel morfologi. Objek yang memenuhi kriteria akan ditandai dengan kotak hijau dan diberi label "Cube".



Gambar 11. Objek terdeteksi

### 3.3.2 Deteksi Warna

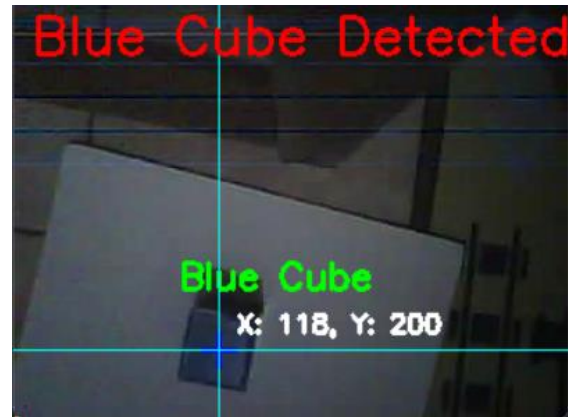
Deteksi warna dilakukan untuk mengetahui warna spesifik dari objek dengan membandingkan nilai HSV (*Hue, Saturation, Value*). Pada komputasi ini, warna yang digunakan dibatasi menjadi empat warna utama, yaitu merah, biru, kuning, dan hijau. Setiap warna memiliki rentang nilai HSV tertentu yang telah ditentukan. Proses dimulai dengan mengonversi citra dari format BGR menjadi format HSV, karena ruang warna HSV lebih stabil terhadap perubahan pencahayaan. Kemudian dilakukan masking untuk menyoroti piksel-piksel pada citra yang masuk ke dalam rentang HSV warna tertentu. Setelah itu, area-area yang sesuai dengan *masking* dianalisis untuk memastikan warna dominan objek tersebut, sehingga dapat diidentifikasi secara akurat sesuai dengan kategori warna yang ditentukan.



Gambar 12. Deteksi warna kubus

### 3.3.3 Mendapatkan Lokasi Objek

Lokasi objek digunakan sebagai target lokasi *end-effector*. Lokasi ini berbentuk koordinat yang mencakup 3 sumbu (X, Y dan Z). Objek yang diidentifikasi lokasinya adalah hasil dari deteksi objek dan deteksi warna sebelumnya.



Gambar 13. Deteksi lokasi kubus

### 3.4 Penerapan Algoritma Kinematika

Kinematika terbalik adalah algoritma utama yang digunakan oleh robot lengan dalam meminimalkan jarak *end-effector* dengan target. Kinematika terbalik sendiri menggunakan lokasi atau koordinat objek yang telah didapatkan sebelumnya sebagai masukan, dan derajat sebagai keluaran. Dalam aplikasinya kinematika terbalik perlu mengenali karakteristik robot sebelum melakukan operasi.

Inisialisasi awal dilakukan pada algoritma yaitu panjang setiap segmen dan batas rotasi setiap motor. Kemudian algoritma kinematika terbalik mulai menghitung sudut-sudut setiap sambungan (*joint*) untuk menggerakkan *end-effector* menuju posisi target. Perhitungan ini didasarkan pada model geometris robot, yang mencakup panjang setiap segmen serta hubungan antara posisi *end-effector* dan sudut rotasi motor. Untuk robot lengan 6-DOF, kinematika terbalik melibatkan model geometris dan optimisasi.

Setelah panjang lengan, batas rotasi, dan ukuran arena diketahui, robot akan menggunakan data posisi yang diperoleh dari kamera. Data tersebut kemudian dikonversi ke dalam koordinat aktual sesuai dengan area kerja robot. Koordinat aktual ini akan menjadi input bagi algoritma kinematika terbalik (*inverse kinematics*) untuk menentukan titik target. Selanjutnya, dilakukan perhitungan sudut rotasi untuk setiap sambungan (*link*) pada robot untuk mencapai posisi target dengan presisi. Perhitungan sudut rotasi untuk setiap link pada robot arm 5 DOF dilakukan menggunakan algoritma kinematika terbalik (*inverse kinematics*). Sudut rotasi *base* dihitung untuk menentukan arah horizontal.

$$\theta_1 = \arctan\left(\frac{y}{x}\right) \quad (1)$$

Selanjutnya, proyeksi jarak dari target pada bidang horizontal ( $r$ ) dihitung dengan persamaan berikut.

$$r = \sqrt{x^2 + y^2} \quad (2)$$

Untuk menentukan sudut pada *joint* pertama atau *lower arm*, digunakan hukum *cosinus*:

$$D = \frac{r^2 + z^2 - L_1^2 - L_2^2}{2 \cdot L_1 \cdot L_2} \quad (3)$$

$$\theta_2 = \arccos(D) \quad (4)$$

Sudut pada lengan tengah atau *center arm* dihitung dengan menambahkan sudut  $\alpha$  ( $\alpha$ ) dan  $\beta$  ( $\beta$ ), di mana:

$$\alpha = \arctan\left(\frac{z}{r}\right) \quad (5)$$

$$\beta = \arccos\left(\frac{r^2 + z^2 + L_1^2 - L_2^2}{2 \cdot L_1 \cdot \sqrt{r^2 + z^2}}\right) \quad (6)$$

$$\theta_3 = \alpha + \beta \quad (7)$$

Terakhir, *upper arm* dihitung untuk menyesuaikan orientasi ujung lengan (*end-effector*). Sudut adalah hasil pengurangan orientasi target ( $\phi$ ) dengan jumlah sudut.

$$\theta_4 = \phi - (\theta_2 + \theta_3) \quad (8)$$

Sedangkan rotasi *neck gripper* ( $\theta_5$ ) merupakan rotasi tambahan untuk menyesuaikan posisi *gripper* sesuai kebutuhan. Rumus-rumus ini memastikan setiap *link* pada robot bergerak secara presisi untuk mencapai posisi target yang telah ditentukan.

Setelah melakukan kinematika terbalik, diperlukan kinematika maju untuk mengetahui posisi dan orientasi *end-effector* dari sudut. Langkah pertama adalah menghitung posisi pada sumbu horizontal ( $x$ ) dan vertikal ( $z$ ) untuk setiap *joint*. Rumus posisi untuk setiap koordinat dihitung dengan memanfaatkan transformasi matriks homogen dan trigonometri.

$$x = (L_1 \cos \theta_2 + L_2 \cos(\theta_2 + \theta_3)) \cos \theta_1 \quad (9)$$

$$y = (L_1 \cos \theta_2 + L_2 \cos(\theta_2 + \theta_3)) \sin \theta_1 \quad (10)$$

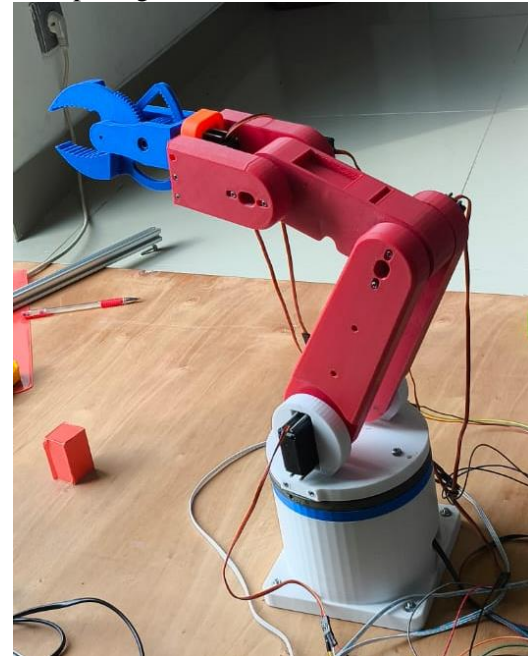
Sedangkan posisi pada sumbu vertikal ( $z$ ) dihitung dengan:

$$z = L_1 \sin \theta_2 + L_2 \sin(\theta_2 + \theta_3) \quad (11)$$

Orientasi *end-effector* ditentukan oleh kombinasi sudut ( $\theta_4$ ) dan ( $\theta_5$ ), yang merepresentasikan rotasi *gripper* untuk mencengkeram objek pada posisi yang diinginkan. Hasilnya akan dibandingkan dengan posisi target untuk mendapatkan nilai *error* baik pada sumbu  $x$ ,  $y$  dan  $z$ .

#### 4. HASIL DAN PEMBAHASAN

Robot lengan yang telah dibuat dapat dilihat pada gambar 12. Robot



Gambar 14. Robot lengan

Pada posisi *home*, robot lengan memiliki jarak 7 cm dari kamera ESP32-CAM. Kamera tersebut diposisikan pada ketinggian 70 cm dari permukaan arena pengujian. Arena pengujian berbentuk persegi dengan panjang sisi 100 cm. Dalam konfigurasi ini, ESP32-CAM mampu membaca area dengan lebar 45 cm dan panjang 40 cm. Hal ini memastikan bahwa area pengujian berada dalam jangkauan kamera untuk mendeteksi objek dengan akurasi optimal.

##### 4.1 Pengujian

Pengujian dilakukan menggunakan tiga objek dengan warna yang berbeda, yaitu merah, hijau, dan biru. Setiap objek diuji sebanyak lima kali dengan posisi target dan posisi awal robot



yang sama. Pengujian ini bertujuan untuk memastikan robot dapat berfungsi dengan optimal, terutama dalam mendeteksi warna, menghitung koordinat, dan melakukan pergerakan servo secara akurat dan konsisten. Hasil pengujian diharapkan menunjukkan bahwa sistem mampu bekerja secara andal dalam berbagai kondisi dengan tingkat keberhasilan yang tinggi.

## 4.2 Hasil Pengujian

### 4.2.1 Objek Merah

Pengujian robot untuk objek berwarna merah menunjukkan adanya *error* yang cukup signifikan antara posisi target dan posisi *end effector*, dengan *error* terbesar pada sumbu z mencapai 27.20 cm, sementara *error* pada sumbu x dan y masing-masing sebesar 4.15 cm dan 13.00 cm. Waktu pergerakan bervariasi antara 0.35 detik hingga 1.91 detik, yang kemungkinan dipengaruhi oleh jarak target dan kompleksitas pergerakan. Variasi waktu pergerakan juga menunjukkan ketidakkonsistenan yang perlu diminimalkan dengan optimasi lebih lanjut.

Tabel 2. Pengujian dengan Objek Merah

Posisi Target (x, y, z)	Posisi End Effector (x, y, z)	Error (x, y, z)	Waktu
-8.88, 6.94, 10.00	-20.57, 3.40, 20.60	11.70, 3.53, 10.60	1.1
6.88, 9.75, 10.00	-18.12, 4.45, 22.81	25.00, 5.30, 12.81	5.12
-6.75, 7.12, 10.00	-20.10, 3.33, 20.90	13.35, 3.80, 10.90	2.34
-13.38, 3.94, 10.00	-21.70, 2.36, 19.94	8.33, 1.58, 9.94	0.28
-8.50, 14.25, 10.00	-21.70, 2.36, 19.94	8.33, 1.58, 9.94	0.30
-13.62, 3.75, 10.00	-21.70, 2.36, 19.94	8.33, 1.58, 9.94	0.24

10.00	19.94	9.94	
-13.62, 3.94, 10.00	-21.70, 2.36, 19.94	8.33, 1.58, 9.94	1.90
-13.62, 3.75, 10.00	-21.81, 2.47, 19.90	8.19, 1.28, 9.90	3.96
-13.62, 1.75, 10.00	-21.45, 8.28, 20.99	12.95, 5.97, 10.99	0.57
-13.62, 2.75, 10.00	-21.45, 8.28, 20.99	12.95, 5.97, 10.99	0.57

Posisi target seharusnya selalu sama karena target tidak diubah sama sekali posisinya, tetapi perubahan koordinat target seperti terjadi karena adanya perubahan titik tengah pada kontur yang mengakibatkan koordinat berpindah ke titik yang lain. Perubahan yang terjadi tidak terlalu signifikan tetapi hal tersebut dapat menyebabkan *error* pada pergerakan robot yang mengakibatkan kegagalan dalam menjalankan program.

### 4.2.2 Objek Hijau

Pengujian robot untuk objek berwarna hijau menunjukkan adanya *error* yang cukup konsisten pada posisi end effector terhadap posisi target. *Error* pada sumbu x berkisar antara 4.08 cm hingga 4.21 cm, sedangkan pada sumbu y *error* mencapai 5.46 cm. Sedangkan *error* pada sumbu z cenderung signifikan, yaitu sebesar 26.74 cm pada semua pengujian. Waktu pergerakan bervariasi dari 0.13 detik hingga 4.56 detik, menunjukkan adanya variasi dalam durasi gerakan yang kemungkinan disebabkan oleh perbedaan jarak target atau kompleksitas jalur pergerakan.

Tabel 3. Pengujian dengan Objek Hijau

Posisi Target (x, y, z)	Posisi End Effector (x, y, z)	Error (x, y, z)	Waktu
3.25, 5.81, 10.00	-18.36, 2.80, 22.26	21.61, 3.01, 12.26	0.38

3.38, 6.00, 10.00	-18.35, 2.88, 22.28	21.73, 3.12, 12.28	0.37
3.50, 6.00, 10.00	-18.35, 2.88, 22.28	21.83, 3.12, 12.30	0.52
-7.75, 7.88, 10.00	-20.61, 5.19, 20.83	12.63, 4.00, 10.78	1.73
-7.75, 8.06, 10.00	-20.38, 3.87, 20.78	12.79, 3.31, 10.80	5.84
-7.75, 8.06, 10.00	-20.54, 4.75, 20.80	12.79, 3.31, 10.80	5.74
-7.75, 8.06, 10.00	-20.54, 4.75, 20.80	12.79, 3.31, 10.80	2.93
-7.75, 8.06, 10.00	-20.54, 4.75, 20.80	12.79, 3.31, 10.80	1.19
-7.88, 8.06, 10.00	-20.41, 3.98, 20.77	12.53, 4.08, 10.77	11.98
8.25, 11.44, 10.00	-18.22, 5.25, 22.91	26.47, 6.19, 12.91	2.12

Pembacaan koordinat target masih mengalami hasil yang sama yaitu terjadi perubahan pada koordinat posisi target.

#### 4.2.3 Objek Biru

Pengujian objek berwarna biru terlihat bahwa posisi *end effector* memiliki *error* yang cukup konsisten pada semua pengujian. *Error* pada sumbu x berkisar antara 3.91 cm hingga 4.03 cm, sementara *error* pada sumbu y berada di kisaran 3.87 cm hingga 4.05 cm. *Error* pada sumbu z relatif besar, yaitu sebesar 26.28 cm untuk seluruh pengujian. Waktu yang diperlukan untuk mencapai target bervariasi dari 0.13 detik hingga 1.09 detik, menunjukkan bahwa ada perbedaan dalam durasi gerakan tergantung pada posisi target dan jalur pergerakan yang ditempuh.

Tabel 4. Pengujian dengan Objek Biru

Posisi Target (x, y, z)	Posisi End Effector (x, y, z)	Error (x, y, z)	Waktu
-7.75, 8.81, 0.00	-20.61, 5.19, 20.83	12.86, 3.63, 20.83	0.54
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.30
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.39
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.29
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.45
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.55
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.63
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	0.27
-7.62, 9.00, 10.00	-20.61, 5.26, 20.84	12.98, 3.74, 10.84	0.65
-7.75, 8.81, 10.00	-20.61, 5.19, 20.83	12.86, 3.63, 10.83	1.05

## 5. KESIMPULAN

Setelah melakukan percobaan dan menganalisis hasil yang didapat dari penelitian penerapan kinematika terbalik pada robot lengan 5 DOF, maka dapat ditarik kesimpulan sebagai berikut :

- Hasil pengujian rangkaian elektronik membuktikan bahwa PCB berfungsi secara efisien, stabil, dan mampu memenuhi kebutuhan sistem robot dengan baik.
- Perangkat PCB mampu menunjang performa yang optimal pada robot, menjaga presisi gerakan robot, dan memastikan stabilitas dalam setiap operasi.
- Pengolahan citra digital seperti *thresholding*, *edge detection* dan pengurangan *noise* sangat membantu dalam pendeteksian seperti deteksi objek, warna dan lokasi.
- Algoritma kinematika terbalik memerlukan lebih banyak pengujian dibandingkan dengan pengujian yang telah dilakukan sebelumnya.
- Pencahayaan pada lingkungan percobaan sangat berpengaruh dengan kemampuan robot dalam mendeteksi objek dan warnanya.

### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Negeri Yogyakarta, khususnya Departemen Teknik Elektro dan Elektronika, Fakultas Vokasi, atas dukungan finansial yang diberikan dalam penelitian. Penulis juga ingin menyampaikan terimakasih kepada Ir. Ardy Seto Priambodo S. T., M. Eng., selaku dosen pengampu mata kuliah praktikum robotika yang telah memberikan saran dan arahan terhadap proses penelitian. Ucapan terima kasih juga penulis sampaikan kepada tim penelitian yang telah bekerja dengan penuh semangat dan tetap solid dalam menghadapi berbagai tantangan selama proses penelitian. Dukungan dan kerja sama dari seluruh anggota tim menjadi faktor penting dalam keberhasilan penelitian.

### DAFTAR PUSTAKA

- [1] A. P. P. Prasetyo, K. Exaudi, S. Sembiring, B. Alfaretz and M. A. Amaria, "Pergerakan Robot Lengan Pengambil Objek Dengan Sistem Perekam Gerak Berbasis Komputer," *JTEV (Jurnal Teknik Elektro dan Vokasional)*, pp. 230-231, 2022.
- [2] V. D. Cong, D. A. Duy and L. H. Phuong, "Development of Multi-Robotic Arm System for Sorting System Using Computer Vision," *Journal of Robotics and Control (JRC)*, vol. 3, no. 5, pp. 690-698, 2022.
- [3] S. S. B. Krishna, S. Kusumastuti, B. Yunanto, F. Maulana and H. T. Sasti, "Robotic Arm 6 Degree of Freedom (DoF) on SCADA-based Modular Production System (MPS)," *NST Proceedings*, pp. 21-27, 2022.
- [4] A. Roshanianfard, N. Noguchi and T. Kamata, "Design and performance of a robotic arm for farm use," *IJABE*, vol. 12, pp. 146-158, 2019.
- [5] Ge, T. Kang, "Solving Inverse Kinematics Constraint Problems for Highly Articulated Models", Thesis, Master of Mathematics in Computer Science, University of Waterloo, Canada, 2000.
- [6] D. Serap, "Forward and Inverse Kinematics Solution of 6-DOF Robots Those Have Offset and Spherical Wrist", *Eurasian J. Sci. Eng. Tech.*, 2022; 3(1): 014 - 028.
- [7] S. Luo, D. Chu, Q. Li, and Y. He, "Inverse kinematics solution of 6-DOF manipulator based on multi-objective full-parameter optimization PSO algorithm," *Frontiers in Neurorobotics*, vol. 16, Mar. 2022. doi:10.3389/fnbot.2022.791796
- [8] Nizam, M. N., Yuana, N. H., & Wulansari, N. Z. (2022). Mikrokontroler ESP 32 Sebagai Alat Monitoring Pintu Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 767–772.
- [9] Kusuma, H. A., Wijaya, S. B., & Nusyirwan, D. (2023). Sistem Keamanan Rumah Berbasis ESP32-Cam dan Telegram Sebagai Notifikasi. *Infotronik Jurnal Teknologi Informasi Dan Elektronika*, 8(1), 30.
- [10] Prasetyo, T. F., Sujadi, H., & Azizi, R. M. (2020). Desain Dan Pengembangan Peralatan Rekayasa Otomatis Pada Papan Tulis Menggunakan Arduino Uno R3 Terintegrasi Dengan Android: Indonesia. *INFOTECH Journal*, 6(2), 57–64.
- [11] Kusuma, Johan Wijaya, dkk. 2013. Penerapan Invers Kinematik Terhadap Pergerakan Kaki Pada Robot Hexapod. Jurusan Teknik Informatika. STMIK GI MDP.
- [12] E. K. Frisyras, V. C. Moulianitis, and N. A. Aspragathos, "Anns to approximate all the inverse kinematic solutions of non-cuspidal manipulators," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 418–423, 2018. doi:10.1016/j.ifacol.2018.11.586
- [13] J. V. Patel. (2021). Robotic Manipulator (Robotic Arm). *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 9, pp. 218-220.
- [14] S. Sriani, A. Nabila, "Implementasi deep learning untuk mengidentifikasi Umur Manusia menggunakan Convolutional Neural Network (CNN)," *Jurnal Informatika dan Teknik Elektro*

*Terapan*, vol. 12, no. 3, Aug. 2024.  
doi:10.23960/jitet.v12i3.4457