

Implementasi *Inverse Kinematic* Pada Lengan Robot Untuk Proses Pemilahan Barang Berdasarkan Objek Menggunakan *Computer Vision*

Reiyan Prayoga^a, Radimas Putra Muhammad Davi Labib^b, Aryuanto Soetedjo^c

^{abc}Program Studi Teknik Elektro S1, Institut Teknologi Nasional Malang, Indonesia

ARTICLE INFO

Article history:

Received February 27, 2025

Received in revised form
October 10, 2025

Accepted October 20, 2025

Available online November 29, 2025

Keywords:

Inverse Kinematics

Robotic Arm

Object Detection and Sorting

ABSTRACT

This research applies the inverse kinematics method to control the motion of a robotic arm designed to sort objects based on their color and shape using a computer vision approach. The developed system consists of several components, including a conveyor, an external camera, a robotic arm, an ESP32 microcontroller, and a SCADA platform for system monitoring and control. The external camera captures images of objects moving along the conveyor, which are processed using the HSV algorithm to detect colors—red, yellow, and blue—and the YOLOv8 model to classify object shapes such as cubes and cylinders. The detection results are used as references by the inverse kinematics algorithm to calculate the joint angles of the robotic arm in order to reach the specified target coordinates, allowing the objects to be moved to their respective positions. Experimental results show that the image processing subsystem performs reliably under various lighting conditions, including low illumination and environments with multiple surrounding objects. In addition, the inverse kinematics test showed an average positional deviation of about 1–2 cm from the desired coordinates. Overall, the system is capable of performing object sorting based on color and shape.

1 Pendahuluan

Era Revolusi Industri 4.0 tidak hanya menitikberatkan pada pengembangan Internet of Things (IoT), tetapi juga pada inovasi perangkat otomatis yang memiliki tingkat integrasi tinggi, fleksibilitas, dan kemudahan dalam pengoperasian [1][2][3][4]. Salah satu teknologi yang berperan penting dalam era ini adalah lengan robot yang digunakan untuk melakukan penyortiran barang[2][5][6]. Dalam konteks otomasi, penyortiran berarti mengelompokkan objek berdasarkan ciri fisik seperti warna, bentuk, atau ukuran sesuai kebutuhan aplikasi [7][8].

Pergerakan lengan robot dirancang dengan meniru cara kerja serta struktur tangan manusia [2][9]. Secara umum, lengan robot terdiri dari beberapa bagian utama seperti pergelangan dan lengan yang tersusun atas sejumlah *link*. Setiap *link* dihubungkan oleh *joint*, di mana setiap sambungan tersebut mewakili satu derajat kebebasan (*degree of freedom*) yang memungkinkan robot untuk bergerak secara dinamis[1].

Lengan robot memiliki beberapa servo yang berfungsi menggerakkan setiap *joint*. Untuk memastikan pergerakan setiap *joint* sesuai dengan posisi target, diperlukan perhitungan sudut gerak yang akurat. Perhitungan ini dilakukan menggunakan metode *inverse kinematics*, yaitu pendekatan matematis untuk menentukan konfigurasi sendi agar ujung lengan dapat mencapai posisi yang diinginkan dengan tepat [10].

Metode *inverse kinematics* (IK) merupakan pendekatan matematis yang digunakan untuk menghitung sudut pada setiap sambungan lengan robot berdasarkan posisi target yang ingin dicapai [11]. Dengan menggunakan metode ini, operator tidak harus menentukan secara manual sudut gerakan, karena sistem secara otomatis menghitung nilai sudut yang diperlukan untuk ujung lengan mencapai titik tujuan [10].

Sebelum proses penyortiran, sistem harus mengenali objek yang akan dipilah. Proses ini dilakukan melalui pengolahan citra digital yang memungkinkan identifikasi objek berdasarkan warna, bentuk, atau ukuran, dan menjadi bagian penting dari otomatisasi industri 4.0[12].

Penelitian pada jurnal [13], merancang sistem lengan robot yang mampu memilih objek berdasarkan warna dan ketinggian menggunakan kamera CMUCam Pixy. Setiap objek yang terdeteksi kemudian dipindahkan ke lokasi yang telah ditentukan sesuai dengan parameter warnanya.

Sementara itu, pada jurnal [14], sistem serupa juga dirancang dengan fokus pada penyortiran warna, di mana CMUCam Pixy berfungsi mendeteksi warna objek, lalu robot

memindahkannya ke wadah yang sesuai dengan kategori warna tersebut.

Adapun penelitian pada jurnal [15], merancang lengan robot yang dapat memilah barang berdasarkan berat objek dengan bantuan sensor load cell, serta mengintegrasikan *sistem Internet of Things* (IoT) untuk pemantauan dan kendali jarak jauh.

Berdasarkan penelitian-penelitian sebelumnya, penelitian ini merancang robot pemilah objek yang memanfaatkan metode *inverse kinematic* sebagai sistem kendali pergerakan lengan robot. Mekanisme alat ini melibatkan konveyor untuk membawa objek menuju posisi deteksi kamera. Setelah objek terdeteksi, pengolahan citra dilakukan untuk mengidentifikasi warna dan bentuk objek. Hasil identifikasi tersebut digunakan untuk mengarahkan lengan robot agar memindahkan objek ke lokasi yang sesuai. Kebaruan dari penelitian ini terletak pada integrasi metode *inverse kinematic* dengan sistem visi komputer berbasis algoritma HSV untuk deteksi warna dan model YOLOv8 untuk deteksi bentuk. Pendekatan ini memungkinkan sistem melakukan identifikasi dua parameter sekaligus yaitu warna dan bentuk dalam berbagai kondisi pencahayaan. Selain itu, penggunaan *webcam* konvensional menggantikan CMUCam Pixy memberikan fleksibilitas dalam modifikasi algoritma pemrosesan citra, sehingga sistem dapat disesuaikan dan dikembangkan lebih lanjut sesuai kebutuhan. Keunggulan lain dari penelitian ini adalah integrasinya dengan platform SCADA (*Supervisory Control and Data Acquisition*), yang memungkinkan proses pemantauan dan kendali secara real time melalui antarmuka HMI.

2 Studi Literatur

Penelitian pada jurnal [13], sistem robot lengan yang dirancang berhasil memilah objek berdasarkan warna dengan integrasi sensor dan motor yang dikendalikan oleh Arduino Mega 2560. Sistem ini dapat memindahkan objek di atas konveyor, dan menempatkannya di wadah yang sesuai dengan tingkat akurasi yang tinggi. Kamera CMUCam5 Pixy juga dapat mengidentifikasi barang berdasarkan warna dengan tepat.

Penelitian pada jurnal [14], hasil penelitian menunjukkan bahwa lengan robot dapat mengambil dan menempatkan objek berwarna merah, hijau, dan biru di lokasi yang sesuai. Metode *inverse kinematic* yang diterapkan memungkinkan sistem bergerak mulai proses pengambilan hingga penempatan objek.

Selain itu, kamera CMUCam5 Pixy mampu mendeteksi perbedaan warna dan tinggi objek.

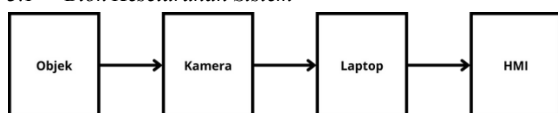
Dari penelitian kedua jurnal diatas sama-sama menggunakan CMUCam5 Pixy. CMUCam Pixy memiliki kelebihan yaitu CMUCam5 Pixy memiliki antarmuka yang sederhana dan mudah dipahami. Pengguna dapat "mengajari" Pixy mengenali objek dengan warna dan bentuk tertentu. Warna dan bentuk objek akan disimpan, dan pengguna dapat menggunakan tombol "teaching" atau aplikasi PixyMon untuk mengajari Pixy. Namun terdapat kekurangan pada CMUCam Pixy yaitu sensitif terhadap cahaya. Pixy dapat sangat sensitif terhadap perubahan kondisi pencahayaan. Terkadang, bayangan saja dapat membuatnya gagal mengenali warna dan yang diatur. Selain itu, user tidak dapat memodifikasi algoritma pemrosesan gambar karena CMUCam pixy di buat untuk memudahkan para pengguna saja.

Penelitian pada jurnal [15], lengan robot mampu memindahkan objek sesuai dengan beratnya dalam rentang waktu sekitar 10 detik dari posisi awal di palet hingga ke lokasi tujuan. Sensor Cell yang digunakan menunjukkan tingkat akurasi pengukuran mencapai 95,52%, kamera ESP 32 dapat menampilkan video streaming secara langsung. Kekurangan pada penelitian ini, tidak menerapkan metode kinematika robot baik itu *inverse kinematic* maupun *forward kinematic*. Pergerakan lengan robot masih menggunakan *trial and error*.

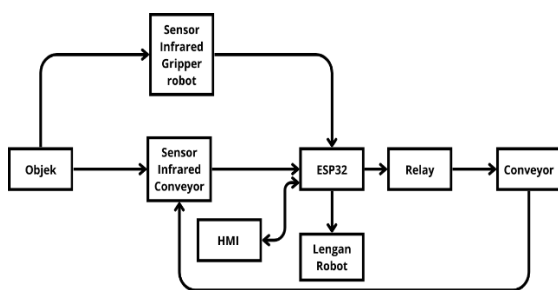
Pada penelitian ini penulis ingin mengembangkan penelitian dari jurnal [15], dimana penulis akan menerapkan *inverse kinematic* untuk metode pergerakan lengan robot dan juga mengembangkan dari segi *computer vision* karena dalam artikel jurnal [13] dan [14] algoritma dari pemrosesan gambar tidak dapat dimodifikasi, maka dari itu, sistem nantinya menggunakan *webcam* untuk mendeteksi barang sehingga algoritma pemrosesan gambar dapat dikembangkan sesuai dengan keinginan.

3 Metodologi

3.1 Blok Keseluruhan Sistem



Gambar 1. Alur Kerja Computer Vision

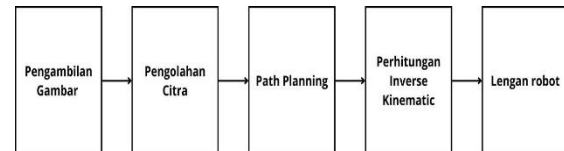


Gambar 2. Alur Kerja Lengan Robot

Berdasarkan Gambar 1 dan 2, cara kerja sistem ini, ketika conveyor berjalan objek juga ikut berjalan, kamera akan mendeteksi objek tersebut. Kamera akan mengirimkan data gambar tersebut ke laptop, laptop akan memproses data gambar tersebut menggunakan algoritma warna dan bentuk. Hasil dari proses ini, laptop akan mengirimkan kode warna dan objek. Selanjutnya HMI (Human Machine Interface) berkomunikasi dengan ESP32, ESP32 menerima data dari HMI. Ketika Objek bergerak diatas conveyor dan mencapai titik akhir, sensor infrared pada conveyor mendeteksi keberadaan objek. Sinyal dari sensor ini dikirimkan ke ESP32, lalu conveyor akan berhenti. Setelah

conveyor berhenti, lengan robot mulai bergerak menuju objek. Pada gripper lengan robot terdapat sensor infrared untuk mendeteksi keberadaan objek di area cengkraman. Jika sensor mendeteksi objek, gripper mencengkram objek dengan aman, lalu lengan robot memindahkannya ke lokasi yang telah ditentukan. Setelah lengan robot selesai memindahkan objek, ESP32 mengirimkan perintah ke relay untuk menyalakan kembali conveyor. Conveyor kembali bergerak untuk membawa objek berikutnya ke posisi yang sesuai.

3.2 Mekanisme Kerja Sistem Secara Keseluruhan



Gambar 3. Gambaran Umum Cara Kerja Sistem

Berdasarkan Gambar 3, alur kerja sistem, yaitu proses pengendalian robot dimulai dengan pengambilan gambar, dimana kamera akan mendeteksi objek yang ada dalam area kerja. Setelah itu, sistem melakukan pengolahan citra untuk menganalisis gambar dan menentukan warna, bentuk, serta koordinat objek yang terdeteksi. Setelah objek berhasil dikenali, tahap selanjutnya adalah perencanaan lintasan atau *trajectory*, yang menentukan jalur yang akan dilalui oleh lengan robot untuk mencapai posisi objek. Kemudian, pada tahap *inverse kinematic*, Sistem melakukan perhitungan sudut pada setiap sendi lengan robot dengan mengacu pada posisi target yang telah ditentukan melalui *path planning*. Terakhir, lengan robot dikendalikan untuk menggerakkan motor sesuai dengan perhitungan sebelumnya, sehingga dapat mengambil dan memindahkan objek berdasarkan warna dan bentuk yang telah ditentukan.

3.3 Perancangan Pendeteksian Warna

Pendeteksian merupakan teknik proses untuk mengenali dan membedakan objek berdasarkan karakteristik warnanya. lam penelitian ini, sistem pendeteksian warna digunakan untuk mengenali objek berdasarkan tiga kategori warna utama, yaitu merah, kuning, dan biru, yang terdeteksi secara langsung melalui kamera. Berikut adalah alur untuk pendeteksian warna:

1. Menambahkan Pustaka

Perancangan pendeteksian warna pada penelitian ini, menggunakan pustaka *OpenCV* untuk pengolahan gambar. Editor untuk menuliskan kode adalah *Visual Studio Code*.

2. Pengambilan Gambar

Gambar ditangkap menggunakan kamera dengan resolusi yang telah ditentukan.

3. Konversi Gambar ke Ruang Warna HSV

Gambar yang diambil dari kamera dalam format BGR (Blue, Green, Red) dikonversi ke HSV (Hue, Saturation, Value). Alasan menggunakan HSV adalah untuk mempermudah identifikasi warna karena memisahkan informasi warna (Hue) dari kecerahan (Value).

4. Penentuan Rentang Warna

Setiap warna ditentukan dengan nilai berikut:

- Merah: [0, 100, 100] hingga [10, 255, 255]
- Biru: [80, 100, 100] hingga [140, 255, 255]
- Kuning: [20, 100, 100] hingga [35, 255,255]

5. Buat Mask untuk Setiap Warna

Membuat mask untuk setiap warna menggunakan fungsi *cv2.inRange()*, mask untuk setiap warna berdasarkan rentang yang telah ditentukan.

6. Menentukan Jumlah Piksel pada Mask

Menentukan jumlah piksel yang terdeteksi dalam masing-masing warna menggunakan fungsi *cv2.countNonZero(mask)*.

7. Menentukan Warna Dominan

Jumlah piksel untuk setiap warna dihitung dan dibandingkan. Jika jumlah piksel lebih dari 500, maka warna tersebut dianggap dominan. Nilai 500 digunakan sebagai ambang batas untuk menentukan apakah warna tersebut cukup signifikan dalam gambar atau tidak.

8. Menampilkan Hasil pada *Frame*

Jika warna terdeteksi, maka akan menampilkan warna ('Merah', Hijau', atau 'Kuning') pada *frame*. Namun, jika objek yang terdeteksi tidak memenuhi kriteria maka akan mengembalikan fungsi 'Tidak ada warna'.

9. Pengujian

Pengujian bertujuan untuk memvalidasi akurasi sistem dalam mendeteksi warna seperti merah, kuning, biru pada berbagai kondisi dan parameter.

3.4 Perancangan Pendeteksian Bentuk

Pendeteksian bentuk merupakan teknik atau proses untuk mengenali objek dimana pada penelitian ini menggunakan objek berbentuk silinder dan kubus. Berikut adalah alur perancangan pendeteksian bentuk:

1. Pengumpulan Dataset

Pada tahap ini, dataset dikumpulkan sebagai bahan utama untuk pelatihan mode. Dataset berupa gambar-gambar yang berisi objek yang akan dideteksi, yakni kubus dan silinder. Gambar diperoleh dari kamera, unduhan dari internet. *Dataset* mencakup berbagai kondisi sudut pandang.

2. Labeling

Gambar yang terkumpul, selanjutnya diberi label. Untuk pemberian label menggunakan *Roboflow*. *Roboflow* merupakan platform berbasis web yang digunakan untuk mengelola dataset, melakukan pelabelan objek, serta menyiapkan data pelatihan dalam proyek *computer vision*, seperti deteksi objek, klasifikasi gambar, dan segmentasi gambar. Pada tahap ini, setiap objek dalam gambar diberi *bounding box* dan label yaitu "silinder" dan "kubus".

3. Training Data

Dataset yang sudah dilabeli digunakan untuk melatih model YOLOv8. Proses pelatihan data dilakukan di *Google Colab*. *Google Colab* merupakan layanan berbasis *cloud* dari *Google* yang memungkinkan pengguna menjalankan kode *Python* secara langsung melalui peramban tanpa harus menginstal perangkat lunak tambahan di komputer lokal. Untuk melakukan training data, gambar yang sudah diberi label tadi kemudian di unduh dan saat pengunduhan terdapat kode unik yang dimasukkan ke *google colab*. Model dilatih dengan konfigurasi tertentu (jumlah *epoch*, ukuran gambar, dan lain lain) hingga mencapai akurasi yang optimal. Setelah data ditraining maka akan disimpan dalam bentuk file *best.pt*.

4. Implementasi YoLov8

Tahap implementasi dilakukan menggunakan bahasa pemrograman *Python*, sedangkan proses pengembangan aplikasinya dijalankan melalui *Visual Studio Code* sebagai lingkungan pemrograman utama.

5. Pengujian

Pada tahap ini, dilakukan pengujian dalam berbagai kondisi untuk memastikan model dapat mendeteksi objek dengan baik.

3.5 Path Planning

Path planning atau *trajectory planning* adalah proses perencanaan lintasan yang harus dilewati oleh robot untuk bergerak dari titik awal ke titik tujuan. Dalam penelitian ini, untuk menentukan lintasan dasar menggunakan persamaan berikut:

Persamaan untuk sumbu x, merujuk pada persamaan (1):

$$x = x_1 + t \times (x_2 - x_1) \quad (1)$$

Persamaan untuk sumbu y, merujuk pada persamaan (2) :

$$y = y_1 + t \times (y_2 - y_1) \quad (2)$$

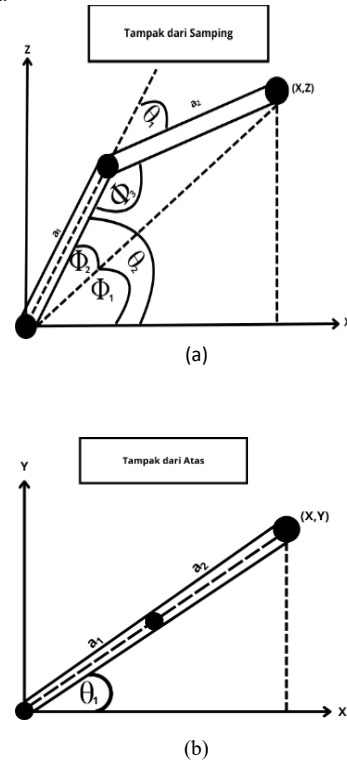
Persamaan untuk Z, merujuk pada persamaan (3):

$$z = z_{max} + \sin \times (\pi \cdot t) \quad (3)$$

3.6 Inverse Kinematic

Inverse Kinematic (IK) adalah metode dalam robotika yang digunakan untuk menghitung sudut-sidit pada sendi robot (*joint angles*) yang diperlukan agar ujung efektor (*end-effector*). Dalam *inverse kinematic*, utamanya adalah posisi dan orientasi target (x,

y, z), sedangkan outputnya adalah sudut (*joint angles*) (α, β, γ) yang akan mengatur posisi robot. Pada Penelitian ini *inverse kinematic* dirancang untuk pergerakan untuk memindahkan objek ke tempat masing-masing objek yang sesuai dengan bentuk dan warnanya.



Gambar 4. Representasi model *inverse kinematic* pada robot 3 DOF: (a) tampilan depan dan (b) tampilan samping

Mengacu pada Gambar 4 (a), mencari θ_1 , merujuk pada persamaan (4):

$$\theta_1 = \arctan \frac{y}{x} \quad (4)$$

Berdasarkan Gambar 4 (a) dan (b), mencari θ_2 :

- Untuk mencari r_1 dan r_2 , merujuk pada persamaan (5) dan (6):

$$r_1 = \sqrt{x^2 + y^2} \quad (5)$$

$$r_2 = \sqrt{r_1^2 + y^2} \quad (6)$$

- Untuk mencari ϕ_1 dan ϕ_2 , merujuk pada persamaan (7) dan (8):

$$\phi_1 = \arccos \frac{r_1}{r_2} \quad (7)$$

$$\phi_2 = \arccos \frac{a_1^2 + r_2^2 - a_2^2}{2 \times a_1 \times r_2} \quad (8)$$

- Maka nilai dari θ_2 , merujuk pada persamaan (9):

$$\theta_2 = \phi_1 + \phi_2 \quad (9)$$

- Mencari θ_3 , merujuk pada persamaan (10) dan (11):

$$\phi_3 = \arccos \frac{a_1^2 + a_2^2 - r_2^2}{2 \times a_1 \times a_2} \quad (10)$$

$$\theta_3 = 180 - \phi_3 \quad (11)$$

4 Pembahasan

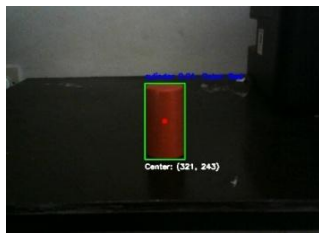
4.1 Pengujian Pengolahan Citra

Pengujian proses pengolahan citra dilakukan untuk memastikan bahwa algoritma yang diterapkan mampu mengenali bentuk dan warna objek secara akurat. Dalam penelitian ini, dilakukan tiga jenis pengujian, dan hasil dari masing-masing pengujian dijelaskan pada bagian berikut.

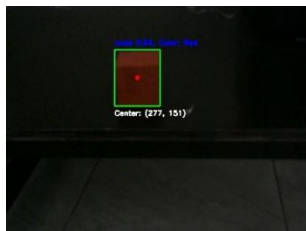
a. Pengujian dengan Kurang Cahaya dan Tanpa Objek
Tabel 1. Pengujian dengan Kurang Cahaya dan Tanpa Objek

No	Jenis Objek	Keberhasilan Deteksi dari Tiga Percobaan (3/3)	Keterangan
1.	Silinder Merah	3/3	Sukses
2.	Silinder Biru	3/3	Sukses
3.	Silinder Kuning	3/3	Sukses
4.	Silinder Tanpa Warna	3/3	Sukses
5.	Kubus Merah	3/3	Sukses
6.	Kubus Biru	3/3	Sukses
7.	Kubus Kuning	3/3	Sukses
8.	Kubus Tanpa Warna	3/3	Sukses

Pengujian ini dilakukan sebanyak 3 kali, dari Tabel 1, hasil menunjukkan bahwa sistem mampu mendeteksi seluruh objek dengan tingkat keberhasilan 3/3 atau 100% pada setiap percobaan meskipun dalam kondisi pencahayaan rendah. Hal ini menunjukkan bahwa algoritma HSV bekerja stabil karena mampu memisahkan komponen warna dan intensitas cahaya, sehingga perubahan pencahayaan tidak memengaruhi hasil deteksi. Model YOLOv8 juga tetap dapat mengenali bentuk kubus dan silinder dengan baik melalui kemampuannya mendeteksi fitur visual seperti tepi dan kontur. Hasil ini menunjukkan bahwa sistem deteksi memiliki performa yang andal pada kondisi kurang cahaya. Berikut merupakan hasil penunjang dari pengujian:



Gambar 5. Hasil Pengujian Pengolahan Citra pada Silinder Merah



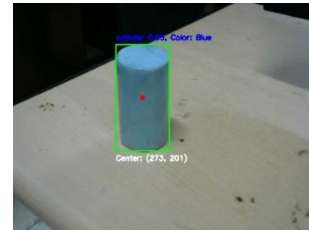
Gambar 6. Hasil Pengujian Pengolahan Citra pada Kubus Merah

Gambar 5 dan 6 di atas menunjukkan hasil pengujian dengan kurang cahaya dan tanpa objek.

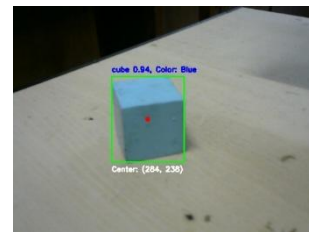
b. Pengujian dengan Cukup Cahaya dan Tanpa Objek
Tabel 2. Pengujian dengan Cukup Cahaya dan Tanpa Barang

No	Jenis Objek	Keberhasilan Deteksi dari Tiga Percobaan (3/3)	Keterangan
1.	Silinder Merah	3/3	Sukses
2.	Silinder Biru	3/3	Sukses
3.	Silinder Kuning	3/3	Sukses
4.	Silinder Tanpa Warna	3/3	Sukses
5.	Kubus Merah	3/3	Sukses
6.	Kubus Biru	3/3	Sukses
7.	Kubus Kuning	3/3	Sukses
8.	Kubus Tanpa Warna	3/3	Sukses

Pada pengujian yang kedua ini, dilakukan sebanyak 3 kali percobaan. Berdasarkan Tabel 2, hasil pengujian dengan pencahayaan yang cukup dan tanpa objek, semua warna dan bentuk dari objek berhasil di deteksi dengan akurasi 100%. Berikut merupakan hasil penunjang dari pengujian:



Gambar 7. Hasil Pengujian Pengolahan Citra pada Silinder Biru



Gambar 8. Hasil Pengujian Pengolahan Citra pada Kubus Biru

Gambar 7 dan 8 menunjukkan hasil pengujian dengan cukup cahaya dan tanpa objek.

c. Pengujian dengan Cukup Cahaya dan Banyak Objek
Tabel 3. Pengujian Cukup Cahaya dengan Banyak Objek

No	Jenis Objek	Keberhasilan Deteksi 3 Percobaan	Keterangan
1.	Silinder Merah	3/3	Semua Percobaan Sukses
2.	Silinder Biru	3/3	Semua Percobaan Sukses
3.	Silinder Kuning	3/3	Semua Percobaan Sukses
4.	Silinder Tanpa Warna	3/3	Semua Percobaan Sukses
5.	Kubus Merah	3/3	Semua Percobaan Sukses
6.	Kubus Biru	3/3	Semua Percobaan Sukses
7.	Kubus Kuning	3/3	Semua Percobaan Sukses
8.	Kubus Tanpa Warna	3/3	Semua Percobaan Sukses

Pada pengujian yang ketiga, sama seperti pengujian sebelumnya, dilakukan 3 kali percobaan, hasil pada Tabel 3 menunjukkan bahwa seluruh objek berhasil terdeteksi dengan tingkat keberhasilan 100%. Hasil ini menunjukkan bahwa sistem mampu mempertahankan kinerja deteksi meskipun jumlah objek yang diamati meningkat. Algoritma HSV tetap dapat membedakan warna secara konsisten karena distribusi cahaya yang merata, sedangkan YOLOv8 mampu mengenali bentuk objek secara tepat tanpa terjadi kesalahan klasifikasi. Kondisi pencahayaan yang stabil juga membantu sistem dalam memisahkan setiap objek dengan jelas, sehingga tidak terjadi tumpang tindih dalam proses identifikasi. Hasil ini menandakan bahwa sistem memiliki stabilitas dan reliabilitas yang baik ketika bekerja pada lingkungan dengan tingkat kompleksitas lebih tinggi. Berikut merupakan hasil penunjang dari pengujian:



Gambar 9. Hasil Pengujian Pengolahan Citra pada Silinder Kuning



Gambar 10. Hasil Pengujian Pengolahan Citra pada Kubus Tanpa Warna

Gambar 9 dan 10 menunjukkan hasil pengujian dengan cukup cahaya dan tanpa objek.

Secara keseluruhan, sistem dapat mendeteksi objek dengan baik pada berbagai kondisi yaitu saat cahaya yang kurang, pencahayaan yang cukup dan kondisi banyak objek. Tingkat akurasi pendeteksian objek yaitu 100% karena tidak ada percobaan yang gagal dari ketiga pengujian yang dilakukan.

4.2 Pengujian Inverse Kinematic

Bagian ini, dilakukan pengujian untuk mengevaluasi kinerja algoritma *inverse kinematic* dimana pada pengujian dilakukan dengan memasukkan koordinat X, Y, dan Z dari setiap kotak penempatan yang berbeda-beda. Tujuan pengujian ini yaitu untuk menguji akurasi pergerakan lengan robot dalam mencapai posisi yang dituju pada koordinat X, Y, dan Z. Berikut adalah data hasil pengujian *inverse kinematic*.

- a. Pengujian *Inverse Kinematic* ke Koordinat Kotak Merah
Tabel 4. Pengujian Inverse Kinematic (1)

No	Posisi Tujuan (Sumbu X , Y, Z) [cm]	Posisi Aktual (Sumbu X, Y, Z) [cm]	Selisih Posisi (Sumbu X, Y, Z) [cm]
1.	(14, 14, 9)	(16, 14, 11)	(2, 0, 2)
2.	(14, 14, 9)	(16, 14, 11)	(2, 0, 2)
3.	(14, 14, 9)	(16, 14, 11)	(2, 0, 2)
Rata-Rata Deviasi [cm]			(2, 0, 2)

Mengacu pada Tabel 4, target koordinatnya adalah posisi 14 cm, 14 cm, 9 cm. Hasil menunjukkan bahwa rata-rata titik koordinat aktual yang di dapat dari 3 percobaan yaitu posisi 16 cm, 14 cm, 11 cm. Rata-rata deviasi yang didapatkan yaitu untuk sumbu X dan Z adalah 2 cm sedangkan deviasi sumbu Y yaitu 0 cm.

- b. Pengujian *Inverse Kinematic* ke Koordinat Kotak Biru
Tabel 5. Pengujian Inverse Kinematic (2)

No	Posisi Tujuan (Sumbu X , Y, Z) [cm]	Posisi Aktual (Sumbu X, Y, Z) [cm]	Selisih Posisi (Sumbu X, Y, Z) [cm]
1.	(14, 4, 9)	(16, 4, 11)	(2, 0, 2)
2.	(14, 4, 9)	(16, 4, 11)	(2, 0, 2)
3.	(14, 4, 9)	(16, 4, 11)	(2, 0, 2)
Rata-Rata Deviasi [cm]			(2, 0, 2)

Mengacu pada Tabel 5, target koordinatnya adalah posisi 14 cm, 4 cm, 9 cm. Hasil menunjukkan bahwa rata-rata titik koordinat aktual yang di dapat dari 3 percobaan yaitu posisi 16 cm, 4 cm, 11 cm rata-rata deviasi yang diperoleh menunjukkan

bahwa pada sumbu X sebesar 2 cm, sumbu Y sebesar 0 cm, dan pada sumbu Z sebesar 2 cm.

- c. Pengujian *Inverse Kinematic* ke Koordinat Kotak Kuning
Tabel 6. Pengujian Inverse Kinematic (3)

No	Posisi Tujuan (Sumbu X , Y, Z) [cm]	Posisi Aktual (Sumbu X, Y, Z) [cm]	Selisih Posisi (Sumbu X, Y, Z) [cm]
1.	(25, 4, 9)	(26, 4, 11)	(1, 0, 2)
2.	(25, 4, 9)	(26, 4, 11)	(1, 0, 2)
3.	(25, 4, 9)	(26, 4, 11)	(1, 0, 2)
Rata-Rata Deviasi [cm]			(1, 0, 2)

Mengacu pada Tabel 6, target koordinatnya adalah posisi 14 cm, 14 cm, 9 cm. Hasil menunjukkan bahwa rata-rata titik koordinat aktual yang di dapat dari 3 percobaan yaitu posisi 16 cm, 14 cm, 11 cm. Rata-rata deviasi yang diperoleh menunjukkan bahwa pada sumbu X sebesar 1 cm, sumbu Y sebesar 0 cm, dan pada sumbu Z sebesar 2 cm.

- d. Pengujian *Inverse Kinematic* ke Koordinat Kotak Tanpa Warna
Tabel 7. Pengujian Inverse Kinematic (4)

No	Posisi Tujuan (Sumbu X , Y, Z) [cm]	Posisi Aktual (Sumbu X, Y, Z) [cm]	Selisih Posisi (Sumbu X, Y, Z) [cm]
1.	(25, 14, 9)	(26, 14, 11)	(1, 0, 2)
2.	(25, 14, 9)	(26, 14, 11)	(1, 0, 2)
3.	(25, 14, 9)	(26, 14, 11)	(1, 0, 2)
Rata-Rata Deviasi [cm]			(1, 0, 2)

Berdasarkan Tabel 7 target koordinatnya adalah posisi 25 cm, 4 cm, 9 cm. Hasil menunjukkan bahwa rata-rata titik koordinat aktual yang di dapat dari 3 percobaan yaitu posisi 26 cm, 14 cm, 11 cm. Rata-rata deviasi yang diperoleh menunjukkan bahwa pada sumbu X sebesar 1 cm, sumbu Y sebesar 0 cm, dan pada sumbu Z sebesar 2 cm.

Keseluruhan pengujian *inverse kinematic* menunjukkan adanya deviasi rata-rata sebesar 1–2 cm dari koordinat target yang diinginkan. Penyimpangan ini terutama disebabkan oleh faktor mekanis, seperti kelonggaran pada sambungan antar link serta keterbatasan presisi servo Hitec HS-625MG yang digunakan. Servo analog tersebut memiliki akurasi sudut sekitar $\pm 1-2^\circ$, sehingga kesalahan kecil pada sudut setiap sendi dapat terakumulasi menjadi deviasi posisi beberapa sentimeter pada ujung lengan. Selain itu, faktor lain seperti ketidaktepatan perhitungan dimensi link, mempengaruhi hasil akhir. Meskipun demikian, nilai deviasi tersebut masih berada dalam batas toleransi sistem dan menunjukkan bahwa metode *inverse kinematic* telah diterapkan dengan baik.

4.2 Pengujian Keseluruhan Sistem

Tabel 8. Tabel Pengujian Keseluruhan

No	Warna Dan Objek	Hasil Deteksi Bentuk (Berhasil/Tidak)	Hasil Deteksi Warna (Berhasil/Tidak)	Keberhasilan Mengambil dan Meletakkan Objek (Berhasil/Tidak)
1.	Silinder Merah	Ya	Ya	Ya
2.	Silinder Kuning	Ya	Ya	Ya
3.	Silinder Biru	Ya	Ya	Ya
4.	Silinder Tanpa Warna	Ya	Ya	Ya
5.	Kubus Merah	Ya	Ya	Ya
6.	Kubus Kuning	Ya	Ya	Ya
7.	Kubus Biru	Ya	Ya	Ya
8.	Kubus Tanpa Warna	Ya	Ya	Ya

Berdasarkan Tabel 8, seluruh objek dengan berbagai warna dan bentuk berhasil terdeteksi, diambil, dan ditempatkan dengan benar. Hasil ini menunjukkan bahwa integrasi antara algoritma HSV, YOLOv8, dan metode *inverse kinematic* telah berjalan dengan baik, sehingga setiap tahapan proses mulai dari deteksi, pengambilan, hingga penempatan objek dapat dilakukan secara berurutan tanpa kesalahan. Keberhasilan ini juga menandakan bahwa komunikasi antara kamera, mikrokontroler, dan sistem SCADA bekerja secara sinkron.

5 Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sistem lengan robot yang mampu melakukan proses klasifikasi dan pemindahan objek berdasarkan warna dan bentuk dengan memanfaatkan pendekatan computer vision serta metode *inverse kinematic*. Hasil pengujian menunjukkan bahwa sistem pengolahan citra berbasis algoritma HSV dan model YOLOv8 mampu mendeteksi objek dengan tingkat keberhasilan 100% pada berbagai kondisi pencahayaan dan variasi jumlah objek. Pengujian pergerakan lengan robot menggunakan metode *inverse kinematic* menunjukkan adanya deviasi rata-rata sebesar 1–2 cm dari koordinat target yang diinginkan. Penyimpangan tersebut disebabkan oleh faktor mekanis seperti kelonggaran pada sambungan antar link, keterbatasan akurasi servo dalam mempertahankan sudut gerak, serta ketidaktepatan pada perhitungan dimensi *link* yang digunakan dalam model kinematik. Ketepatan nilai panjang setiap link sangat berpengaruh terhadap hasil perhitungan posisi akhir, karena kesalahan kecil pada dimensi dapat menghasilkan perbedaan posisi yang terakumulasi di ujung lengan. Meskipun terjadi penyimpangan kecil, nilai tersebut masih berada dalam batas toleransi sistem dan tidak memengaruhi keberhasilan proses pemindahan objek. Secara keseluruhan, sistem yang dikembangkan mampu mengenali, mengambil, dan menempatkan objek dengan tepat sesuai warna dan bentuknya.

Untuk pengembangan selanjutnya, sistem dapat disempurnakan dengan melakukan kalibrasi yang lebih presisi terhadap panjang dimensi setiap link agar hasil perhitungan *inverse kinematic* semakin akurat. Penggunaan servo dengan tingkat presisi lebih tinggi juga disarankan untuk mengurangi deviasi sudut gerak yang disebabkan oleh faktor mekanis seperti kelonggaran sambungan antar link.

1. Kontribusi Penulis

Dalam penelitian ini, perancangan penelitian, pengumpulan data, serta analisis utama terhadap hasil yang diperoleh dikerjakan oleh seluruh tim yang terlibat.

2. Ucapan Terima Kasih

Penulis berterima kasih kepada Program Studi Teknik Elektro S1 ITN Malang yang telah memberi dukungan penuh atas penelitian yang dilakukan.

3. Referensi

- [1] L. Febrimadhan and A. Triwiyatno, "Perancangan Arm Manipulator Pemilah Barang Inverse Kinematics," vol. 5, pp. 1–8, 2016.
- [2] S. G. Ludony, M. Mulyadi, and K. Indriati, "Rancang Bangun Purwarupa Lengan Robot Berbantuan Raspberry Pi," J. Elektro, vol. 13, no. 2, pp. 115–124, 2021, doi: 10.25170/jurnalelektro.v13i2.1979.
- [3] W. T. Setiawan, E. D. M., and Syaifurrahman, "Rancang Bangun Robot Lengan 5 Dof Pemindah Barang Menggunakan Sensor Kamera Berbasis Arduino Due," J. Tek. Elektro Univ. Tanjungpura, vol. Vol 2, No. 2018.
- [4] S. Alam, G. Tjahjadi, N. R. Yenita, and S. Supriyadi, "Rancang Bangun Prototype Pengendalian Lengan Robot (Robotic Arm) Sebagai Pemindah Barang Berbasis Internet of Things," Fakt. Exacta, vol. 14, no. 3, p. 140, 2021, doi: 10.30998/faktorexacta.v14i3.9807.

- [5] R. Rendyansyah, A. P. P. Prasetyo, K. Exaudi, S. Sembiring, B. A. Tarigan, and M. A. Amaria, "Pergerakan Robot Lengan Pengambil Objek Dengan Sistem Perakam Gerak Berbasis Komputer," JTEV (Jurnal Tek. Elektro dan Vokasional), vol. 8, no. 2, p. 230, 2022, doi: 10.24036/jtev.v8i2.113147.
- [6] A. A. Dalimunthe, N. A. Adli, and Taryudi, "Prototipe Robot Lengan 3 Degree of Freedom Sebagai Alat Sorting Barang Berdasarkan Warna," J. Autocracy, vol. Vol.6, pp. 62–68, 2019, doi: 10.21009/autocracy.06.2.2.
- [7] I. Al Ashfahani, Y. Sofyan, S. Yahya, and K. Kunci, "Rancang Bangun Lengan Robot Penyortir Benda Kerja Berdasarkan Warna Dengan Pengendalian Kestabilan Cahaya Menggunakan Fuzzy logic," pp. 13–14, 2022.
- [8] R. H. Hutabarat, S. R. Sulistiyanti, E. Nasrullah, and A. M. Avr, "111-Article Text-144-1-10-20140704",
- [9] M. H. Maftuh, "Lengan Robot Pemindah Barang Berdasarkan Ukuran Tinggi," E-JOINT (Electronica Electr. J. Innov. Technol., vol. 1, no. 2, pp. 36–41, 2020, doi: 10.35970/e-joint.v1i2.431.
- [10] M. A. H. K. Sinaga and D. S. Pamungkas, "Invers Kinematik Robot Arm 4 Dof Menggunakan Sensor Leap Motion," Semin. Nas. Terap. Ris. Inov. Ke-6 ISAS Publ. Ser. Eng. Sci., vol. 6, no. 1, pp. 363–371, 2020.
- [11] T. R. Putra, R. Maulana, and W. Kurniawan, "Implementasi Komputasi Paralel dalam Pergerakan Lengan Robot Menggunakan Inverse Kinematic," vol. 2, no. 9, pp. 2516–2521, 2018.
- [12] I. Sulaeman, A. W. Dani, and T. Pangaribowo, "Analisa Inverse Kinematics Pada Prototype 3-DoF Arm Robot Dengan Metode Anfis," J. Teknol. Elektro, vol. 13, no. 1, p. 14, 2022, doi: 10.22441/jte.2022.v13i1.003.
- [13] M. KURNIA, Rancang bangun lengan robot penyeleksi dan pemindah barang berdasarkan warna dan ketinggian menggunakan kamera. 2017. [Online]. Available: <https://repository.unej.ac.id/handle/123456789/84228>
- [14] E. Ramadhani, "Perancangan Robot Lengan Pemilah Obyek Sesuai Warna Berbasis Arduino Mega 2560," Jurnal, vol. 2, no. 2, pp. 432–439, 2019.
- [15] D. Didit Wahyu, S. Aryuanto, and S. I Komang, "Rancang Bangun Lengan Robot Pemilah Barang Berdasarkan Berat dengan Pemanfaatan Internet Of Things (IoT) Sebagai Kontrol Dan Monitoring Jarak Jauh," Semin. Has. Elektro S1 ITN Malang, vol. 21, no. 1, pp. 1–2, 2020.