

PERANCANGAN ROBOT LENGAN BERBASIS OPENCV DAN GOOGLE ASSISTANT

Gde Ilham Romadhony¹, Agung Surya Wibowo², Dien Rahmawati³

^{1, 2, 3}Fakultas Teknik Elektro, Universitas Telkom

¹gdeilham@student.telkomuniversity.ac.id, ²agungsw@telkomuniversity.ac.id,

³dienrahmawati@telkomuniversity.ac.id

Abstrak

Robot lengan merupakan robot yang memiliki fungsi seperti tangan manusia yang dapat membantu mobilisasi khususnya dalam mengambil dan memposisikan barang. Hal ini dapat membantu aktivitas manusia baik dalam dunia industri maupun kehidupan sehari-hari. Agar dapat diakses oleh masyarakat luas maka diperlukan perancangan robot lengan menggunakan Google assistant untuk memberikan perintah suara, adanya segmentasi warna untuk mendeteksi warna objek, melacak objek, dan *inverse kinematics*. Robot telah mampu mencapai tingkat keberhasilan 100% dengan waktu rata-rata yang dibutuhkan untuk menyelesaikan misi 28.675 detik. Penggunaan Google assistant mampu menerima perintah dengan tingkat keberhasilan sebesar 96.97% dan *error servo 1* sebesar 3.18° dengan akurasi sebesar 97.69%, *error servo 2* sebesar 11.99° dengan akurasi sebesar 89.57%, dan *error servo 3* sebesar 18.81° dengan akurasi sebesar 88%. Supaya deteksi dapat berjalan dengan lancar maka harus didukung dengan pencahayaan minimum sebesar 100 lux.

Kata Kunci: *Google assistant, robot lengan, segmentasi warna, inverse kinematics*

Abstract

Robot arm is a robot that has a function like a human hand that can help mobilize, especially in picking up and positioning items. This can help human activities both in the industrial world and in everyday life. In order to be accessible to the wider community, it is necessary to design a robot arm using Google assistant to give voice commands, color segmentation to detect object colors, track objects, and inverse kinematics. The robot has been able to achieve a 100% success rate with an average time required to complete a mission of 28.675 seconds. The use of the Google assistant is able to receive commands with a success rate of 96.97% and servo 1 error of 3.18° with an accuracy of 97.69%, servo 2 error of 11.99° with an accuracy of 89.57%, and servo 3 error of 18.81° with an accuracy of 88%. In order for detection to run smoothly, it must be supported by a minimum lighting of 100 lux.

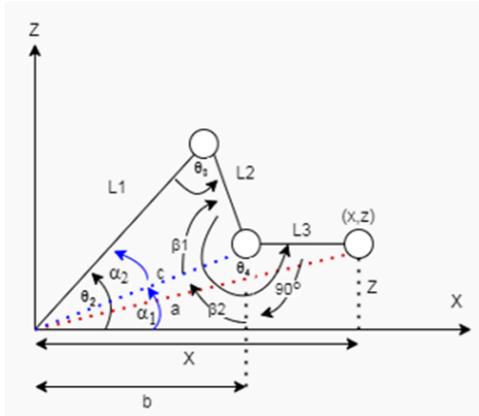
Key Words: *Google assistant, arm robot, color segmentation, inverse kinematics*

1. Pendahuluan

Peran robot sangat meningkat pada beberapa dekade terakhir dalam dunia industri maupun kehidupan sehari-hari, salah satunya adalah robot lengan. Hampir seluruh cabang industri teknologi menggunakan robot lengan untuk membantu dan mempermudah aktivitas manusia seperti memindahkan barang dan memposisikan benda. Selain itu dalam kehidupan sehari-hari, robot lengan digunakan untuk membantu pekerjaan rumah seperti robot pelayan. Kontrol pada robot harus diperhatikan agar dapat dikendalikan oleh masyarakat luas. Penggunaan kendali robot dengan perintah suara merupakan salah satu solusi yang memberikan kemudahan dalam mengendalikan robot. Hal ini memungkinkan adanya interaksi antara robot dengan manusia menggunakan suara tanpa perlu melakukan gerakan seperti menekan tombol untuk memberikan

perintah.

Pada penelitian yang dilakukan oleh W.S. Mada Sanjaya dan Dyah Anggraeni [1] mengenai pengendalian robot lengan dengan suara menggunakan metode proses suara. Metode yang digunakan adalah MFCC (*Mel-Frequency Cepstrum Coefficients*) dalam mengekstrasi ciri suara dan menggunakan metode ANFIS (*Adaptive Neuro-Fuzzy Inference System*) yang digunakan sebagai metode pembelajaran untuk pengenalan pola suara. Dari penelitian tersebut didapatkan tingkat keberhasilan mengenali data suara terlatih adalah 87.7%. Sedangkan penelitian yang dilakukan Agus Nurcahyo dan Heru Susanto [2] mengenai pengendalian putaran motor menggunakan Google assistant, didapatkan akurasi keberhasilan perintah suara sebesar 92.5%. Google assistant sendiri memiliki keunggulan yang lebih seperti bersifat global karena telah banyak digunakan diberbagai negara dan



Gambar 1. Rumus *Inverse kinematic*

dirancang untuk mengontrol berbagai perangkat.

Berdasarkan pada kemudahan dalam mengakses robot lengan agar dapat memindahkan dan memposisikan benda, maka dilakukan penelitian untuk merancang robot lengan dengan perintah suara dengan menggunakan *speech recognition* dengan memanfaatkan fitur Google assistant, menggunakan metode segmentasi warna untuk mendeteksi warna dan metode *inverse kinematics* untuk mengubah jarak benda menjadi sudut servo. Robot menggunakan Raspberry Pi 3 sebagai mikrokomputer yang dapat melakukan proses citra yang dihubungkan dengan PCA 9685 untuk mengendalikan aktuator robot.

2. Dasar Teori

2.1 *Inverse kinematic*

Inverse kinematic merupakan suatu gambaran dari gerak translasi dan rotasi dari titik, objek, atau kumpulan tubuh (kelompok objek) yang tidak mempertimbangkan adanya referensi yang menyebabkan gerak seperti massa, torsi, dan gaya. *Inverse kinematic* merupakan suatu metode yang menentukan titik lokasi dari *end effector*, sehingga pada *inverse kinematics* perlu melakukan perhitungan untuk mendapatkan sudut dari sambungan (*joint*) robot sehingga *end effector* dapat menjangkau titik lokasi [3].

$$b = x - l_3 \tag{1}$$

Nilai b dicari untuk mendapatkan nilai alas dari segitiga *cxz*, dengan nilai b dan z diketahui nilainya.

$$c = \sqrt{b^2 + z^2} \tag{2}$$

Untuk mendapatkan nilai sisi miring segitiga *cxz*, maka menggunakan rumus pythagoras.

$$a_1 = \tan^{-1} \left(\frac{z}{b} \right) \tag{3}$$

Nilai sudut a_1 didapatkan dengan memanfaatkan segitiga *cxz*, dengan menggunakan rumus *arctan* dari nilai sisi didepan sudut dibagi alas segitiga *cxz*.

$$a_2 = \cos^{-1} \left(\frac{l_1^2 + c^2 + l_2^2}{2l_1c} \right) \tag{4}$$

Sudut a^2 didapatkan dengan memanfaatkan segitiga *c,l1,l2*. Merupakan nilai *arcos* dari panjang sisi segitiga yang mengapit sudut dikurang sisi dihadapan sudut a^2 , kemudian dibagi dengan 2 kali sisi yang mengapit sudut yaitu c dan l1. Sudut a^2 didapatkan dengan memanfaatkan segitiga *c,l1,l2*. Merupakan nilai *arcos* dari panjang sisi segitiga yang mengapit sudut dikurang sisi dihadapan sudut a^2 , kemudian dibagi dengan 2 kali sisi yang mengapit sudut yaitu c dan l1.

$$\theta_2 = a_1 + a_2 \tag{5}$$

$$\theta_3 = \cos^{-1} \left(\frac{l_1^2 - c^2 + l_2^2}{2l_1l_2} \right) \tag{6}$$

Sudut θ_3 didapatkan dengan memanfaatkan segitiga *c,l1,l2*. Merupakan nilai *arcos* dari panjang sisi segitiga yang mengapit sudut dikurang sisi dihadapan sudut θ_3 , kemudian dibagi dengan 2 kali sisi yang mengapit sudut yaitu l1 dan l2.

$$\beta_1 = \cos^{-1} \left(\frac{l_2^2 - c^2 - l_1^2}{2cl_2} \right) \tag{7}$$

Sudut β_1 didapatkan dengan memanfaatkan segitiga *c,l1,l2*. Merupakan nilai *arcos* dari panjang sisi segitiga yang mengapit sudut dikurang sisi dihadapan sudut β_1 , kemudian dibagi dengan 2 kali sisi yang mengapit sudut yaitu c dan l2.

$$\beta_2 = \tan^{-1} \left(\frac{b}{z} \right) \tag{8}$$

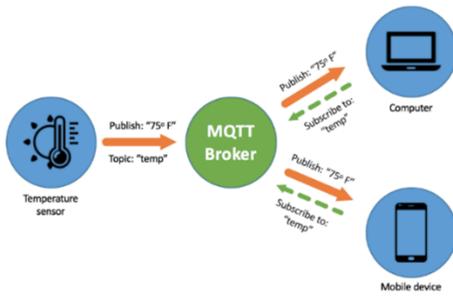
Nilai sudut β_2 didapatkan dengan memanfaatkan segitiga *bcz*, dengan menggunakan rumus *arctan* dari nilai sisi didepan sudut dibagi alas segitiga *bcz*.

$$\theta_4 = \beta_1 + \beta_2 + 90^\circ \tag{9}$$

2.2 Segmentasi Warna

Segmentasi merupakan tahap pemrosesan data untuk memisahkan antara objek yang dikehendaki (*foreground*) dengan objek yang tidak dikehendaki (*background*) [4]. Segmentasi warna dilakukan dengan menggunakan seleksi berdasarkan model warna HSV yang diinginkan serta toleransi yang sesuai pada setiap dimensi warna HSV.

Pada dasarnya citra digital menggunakan acuan warna berformat RGB sebagai standar acuan warna. Agar mempermudah pemrosesan dalam mendeteksi warna, maka gambar yang semula berformat RGB harus dikonversi menjadi HSV. Diperlukan nilai toleransi pada



Gambar 2. Mekanisme MQTT

setiap dimensi HSV untuk membentuk segmen yang sesuai. Pada opencv telah menyediakan sebuah fungsi untuk melakukan konversi RGB (Red, Green, Blue) menjadi HSV (Hue, Saturation, Value) dengan menggunakan fungsi `cv.CvtColor` dan ketentuan parameter yang digunakan adalah `CV_BGR2HSV`. Berikut merupakan baris perintah yang digunakan untuk mengkonversi RGB ke HSV [5].

2.3 Google assistant

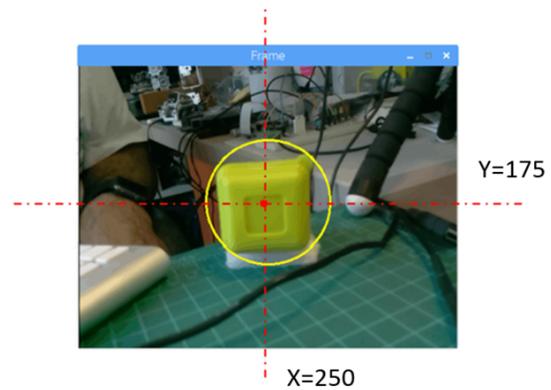
Google assistant adalah asisten pribadi yang dikembangkan oleh google dan tersedia dalam aplikasi pencarian google untuk gawai dan komputer pribadi. Google assistant menggunakan bahasa antarmuka sesuai dengan bahasa alami pengguna untuk menjawab pertanyaan, memberi rekomendasi, dan memproses suatu tindakan dengan mengintegrasikan ke layanan web [6].

2.4 Protokol MQTT

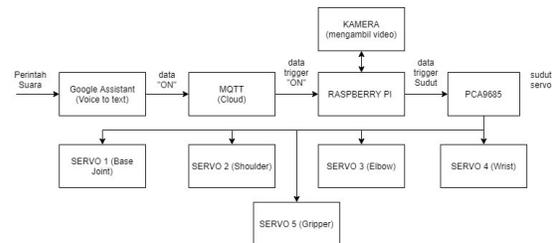
MQTT (*Message Queue Telemetry Transport*) merupakan suatu protokol untuk melakukan komunikasi yang bersifat M2M atau Machine to machine. Jika dalam pengiriman terjadi koneksi yang terputus, protokol MQTT menjamin seluruh pesan yang terkirim. Metode komunikasi yang digunakan oleh protokol MQTT adalah metode komunikasi publish/subscribe. Pesan dari MQTT akan dikirim oleh publisher yang berisikan topik ke broker, setelah topik diolah maka akan diteruskan ke subscriber sesuai dengan permintaan dari pengguna seperti pada pada Gambar 2 [7].

2.5 Sistem Melacak Objek

Sistem pelacakan adalah sistem yang dapat mendeteksi objek dalam keadaan diam maupun dalam keadaan bergerak [8]. Pelacakan banyak dibutuhkan dalam berbagai pengaplikasian seperti human computer interface, komunikasi/kompresi video, dan keamanan sistem [9]. Untuk kali ini pelacakan memanfaatkan kamera atau video secara realtime untuk mendeteksi letak benda berwarna. Peletakan kamera pada robot menyebabkan adanya pergerakan pada kamera, sehingga dibutuhkan algoritma yang dapat melacak kordinat pada



Gambar 3. Koordinat Titik Tengah Objek



Gambar 4. Blok Diagram Sistem Keseluruhan

terminal video saat kamera menyala. Koordinat direpresentasikan dalam sumbu X dan Y, dengan koordinat dari X dimulai dari 0-500 dan koordinat Y dimulai dari 0-350. Dengan nilai titik tengah dari sumbu X adalah 250 dan sumbu Y adalah 175. Titik tengah akan dijadikan sebagai titik utama agar pelacakan objek selalu memposisikan kamera tepat didepan dan ditengah objek yang dilacak seperti pada Gambar 3.

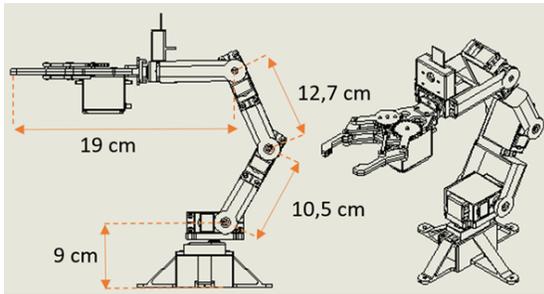
3. Perancangan Sistem

3.1 Desain Sistem

Pada Gambar 4 dapat dilihat hubungan antara komponen yang akan digunakan. Cara kerja sistem robot lengan adalah dengan memberikan perintah suara melalui Google assistant yang terhubung dengan MQTT. MQTT akan mengirim pesan ke Raspberry Pi 3 untuk mengaktifkan proses segmentasi warna, *inverse kinematic*, dan tracking objek. Kamera akan membantu proses segmentasi warna dan PCA9685 akan mengatur sudut yang telah diproses di mikrokomputer ke 5 servo pada base joint, shoulder, elbow, wrist, dan gripper.

3.2 Desain Perangkat Keras

Pada Gambar 5 desain mekanik untuk 5 servo yang digunakan yaitu base joint, shoulder, elbow, wrist, dan gripper. Untuk menggerakkan aktuator robot lengan, kelima servo dihubungkan ke PCA9685 untuk mengatur masing masing sudut servo. Data servo didapatkan saat program *Inverse kinematic* diproses di Raspberry Pi 3



Gambar 5. Desain 3D Robot Lengan

dan dikirimkan melalui pin GPIO 2 dan GPIO 3 menuju pin SDA dan SCL PCA9685.

3.3 Desain Perangkat Lunak

Pada Gambar 6 terlihat diagram alir yang terdapat pada mikrokontroler. Terdapat beberapa tahapan yang merujuk pada Gambar 6. Dimulai dengan inisialisasi awal sudut untuk 5 servo sebelum menerima perintah. Saat perintah diterima robot lengan mulai aktif melakukan tracking objek dan segmentasi warna. Terdapat banyak kondisi yang digunakan untuk setiap proses untuk bisa mengelola data ke program *inverse kinematic*. Kemudian data tersebut dikirim ke masing-masing servo untuk menggerakkan robot lengan.

4. Hasil Percobaan dan Analisis

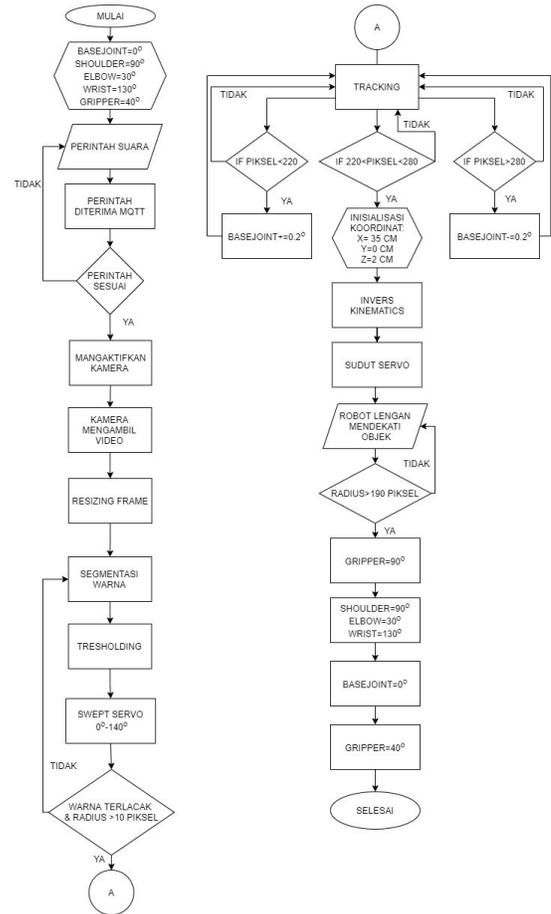
4.1 Pengujian Inverse Kinematic Robot Lengan

Dalam pengujian kali ini, parameter yang akan ditinjau adalah sudut servo yang dihasilkan berdasarkan perhitungan dengan program saat robot bergerak dan perhitungan manual dengan menggunakan busur derajat. Terdapat 3 kondisi peletakan objek terhadap robot sesuai dengan Gambar 7. Hal tersebut dilakukan untuk mengetahui akurasi dan keberhasilan robot dalam menyelesaikan misi untuk mengambil objek yang diperintahkan di berbagai kondisi peletakan.

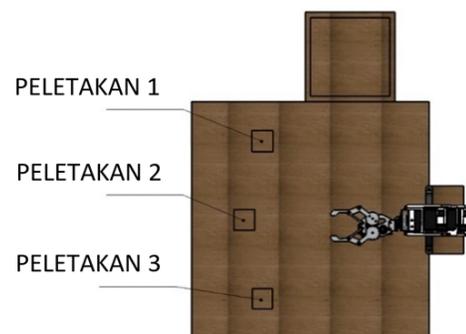
Sebelum dilakukan pengujian telah dilakukan kalibrasi servo dengan menggunakan sudut 90° untuk mempermudah mengatur posisi lengan servo. Penggunaan 90° diambil karena sulitnya mengatur robot lengan dalam mengatur ke 0°, hal ini disebabkan karena adanya frame robot yang tidak memungkinkan untuk melakukan pergerakan di 0°. Pengukuran dilakukan menggunakan alat ukur berupa busur derajat.

Dalam program *Inverse kinematic* dihasilkan sudut servo 1 bernilai 138.1 derajat, servo 2 bernilai 114.9 derajat, dan servo 3 bernilai 156.81 derajat. Seluruh peletakan memiliki nilai ketiga sudut yang sama, hal tersebut dikarenakan jarak dari ketiga objek sama yaitu 35 cm.

Untuk di seluruh peletakan didapatkan rata-rata error servo 1 sebesar 3.18° dengan akurasi sebesar 97.69%, error servo 2 sebesar 11.99° dengan akurasi



Gambar 6. Diagram Alir Dalam Mikrokomputer



Gambar 7. Peletakan Objek Robot Lengan

Tabel 1. Tabel Sudut Servo Dengan Pengukuran Busur Derajat

POSISI OBJEK	BERDASARKAN PROGRAM (Derajat)			BERDASARKAN BUSUR DERAJAT (Derajat)		
	Servo 1	Servo 2	Servo 3	Servo 1	Servo 2	Servo 3
Peletakan 1	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
Peletakan 2	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
Peletakan 3	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138
	138.18	114.99	156.81	135	103	138

Tabel 2. Tabel Rata-Rata Error Sudut Servo

POSISI OBJEK	ERROR (Derajat)		
	Servo 1	Servo 2	Servo 3
Peletakan 1	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
Peletakan 2	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
Peletakan 3	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
	3.18	11.99	18.81
Rata-rata Error	3.18	11.99	18.81

Tabel 3. Tabel Rata-Rata Akurasi Sudut Servo

POSISI OBJEK	AKURASI (%)		
	Servo 1	Servo 2	Servo 3
Peletakan 1	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
Peletakan 2	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
Peletakan 3	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
	97.69	89.57	88
Rata-Rata Akurasi	97.69	89.57	88

Tabel 4. Tabel Data Uji Coba Perintah Warna

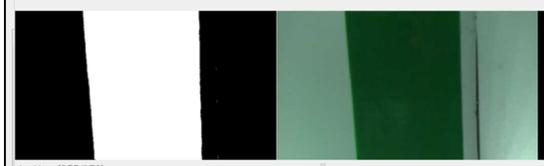
No	Perintah Warna	Uji Coba		Keberhasilan Menerima Perintah (%)
		Berhasil	Gagal	
1	Kuning	29	1	96.67
2	Hijau	28	2	93.33
-3	Biru	30	0	100
Rata-Rata Keberhasilan				96.67

sebesar 89.57%, dan *error* servo 3 sebesar 18.81° dengan akurasi sebesar 88%. Dengan rata-rata *error* yang didapat dari seluruh peletakan tidak memberikan pengaruh yang signifikan terhadap pergerakan robot lengan mendekati objek yang diinginkan. Hal tersebut dibuktikan dengan berhasilnya robot lengan menyelesaikan misi.

4.2 Pengujian Perintah Deteksi Warna Pada Google Assistant

Dalam pengujian kali ini, parameter yang akan ditinjau adalah perintah yang diterima robot sesuai dengan warna objek yang diinginkan. Data yang diambil berdasarkan percobaan sebanyak 30 kali percobaan untuk masing-masing perintah warna dengan jarak antara pengucap dan gawai adalah kurang dari 40 cm.

Berdasarkan Tabel 4 terdapat beberapa kali kegagalan yang disebabkan oleh ketidakjelasan dalam pengucapan perintah, sehingga tidak dapat dikenali maupun menjalankan perintah yang diinginkan. Jarak antara pengucap dan gawai harus diperhatikan, semakin dekat jarak maka akan mempermudah Google assistant untuk mengenali perintah yang diucapkan. Dari 90 kali percobaan dengan masing-masing warna 30 kali percobaan, didapatkan tingkat keberhasilan Google assistant mengenali perintah sebesar 96.67% dengan hanya menggunakan 3 perintah suara.



Gambar 8. Thresholding



Gambar 9. Pembacaan Luxmeter

4.3 Pengujian Pencahayaan Robot Lengan

Untuk menjalankan deteksi warna yang baik perlu pencahayaan yang sesuai. Oleh karena itu dilakukan pengujian pada pencahayaan yang bertujuan untuk mengetahui nilai minimum cahaya yang dibutuhkan untuk bisa memberikan tingkat keberhasilan tinggi pada saat deteksi warna berjalan.

Pada Gambar 8 menunjukkan hasil thresholding dari objek berwarna hijau dengan objek bernilai 1 atau berwarna putih. Gambar 8 merupakan thresholding paling sempurna dan baik tanpa adanya gangguan pada objek yang dideteksi kamera. Hal tersebut didukung oleh pencahayaan sebesar 100 lux pada Gambar 9, sehingga minimum besar cahaya yang harus digunakan sebesar 100 lux agar warna dapat dideteksi dengan sempurna.

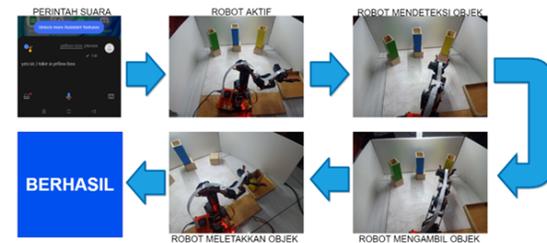
4.4 Pengujian Waktu Robot Lengan Menyelesaikan Misi

Dalam pengujian kali ini, parameter yang akan ditinjau dari mulai robot bergerak hingga menyelesaikan misi. Untuk setiap penempatan dan setiap warna akan diambil data waktu sebanyak 10 kali dengan total keseluruhan uji coba sebanyak 90 kali.

Didapatkan rata rata waktu untuk menyelesaikan misi di peletakan 1 membutuhkan waktu 28.469 detik, pada peletakan 2 membutuhkan waktu 27.358 detik, dan pada peletakan 3 membutuhkan waktu 30.197 detik untuk menyelesaikan misi. Waktu yang dihasilkan tidak selalu konstan karena proses deteksi dan pergerakan servo yang dilakukan dengan delay waktu. Pencahayaan terhadap objek mempengaruhi proses deteksi dan

Tabel 5. Tabel Data Rata-Rata Waktu

No	Letak Objek	Banyak Percobaan	Rata-Rata Waktu (detik)
1	Peletakan 1	30	28.469
2	Peletakan 2	30	27.358
3	Peletakan 3	30	30.197
Rata-Rata Keseluruhan Waktu			28.675



Gambar 10. Pengujian Misi Robot Lengan

penggunaan delay terhadap servo berguna untuk mendeteksi secara perlahan sehingga menghasilkan akurasi robot dalam menjalankan misi meningkat meskipun waktu yang dicapai masih relatif lama yaitu 28.675 detik untuk seluruh warna di seluruh penempatan objek.

4.5 Pengujian Misi Robot Lengan

Pengujian dilakukan di setiap peletakan sebanyak 10 kali untuk setiap warna, sehingga dalam suatu peletakan akan diuji 3 warna dengan total uji coba sebanyak 30 kali. Setelah itu akan dihitung tingkat keberhasilan robot dalam menyelesaikan misi dengan baik. Dengan acuan tingkat keberhasilan robot lengan harus lebih besar dari 78%.

Beraskan data Tabel 6 didapatkan tingkat keberhasilan robot lengan menjalankan misi hingga selesai sebesar 100%. Tingkat keberhasilan yang didapatkan karena pendeteksian objek secara perlahan sehingga caput mampu mencaput objek yang berada didepannya. Jika dilakukan secara cepat pergerakan servo menggapai objek, dikhawatirkan caput tidak mampu menempatkan objek ditengah-tengah caput sehingga robot akan kesulitan untuk mengambil benda yang diinginkan.

Tabel 6. Tabel Uji Coba Misi

No	Misi	Uji Coba		Keberhasilan Menyelesaikan Misi (%)
		Berhasil	Gagal	
1	Kuning	30	0	100.00
2	Hijau	30	0	100.00
3	Biru	30	0	100.00
Rata-Rata				100.00

5. Kesimpulan

Berdasarkan dari perancangan dan percobaan yang telah dilakukan pada robot lengan, dapat ditarik kesimpulan bahwa Google assistant mampu menerima perintah dengan tingkat keberhasilan sebesar 96.97%. Robot lengan membutuhkan waktu untuk menyelesaikan misi di peletakan 1 membutuhkan waktu 28.469 detik, pada peletakan 2 membutuhkan waktu 27.358 detik, dan pada peletakan 3 membutuhkan waktu 30.197 detik untuk menyelesaikan misi. Dengan rata-rata waktu untuk keseluruhan membutuhkan 28.675 detik untuk menyelesaikan misi. Radius minimum yang dihasilkan sebesar 290 piksel dan agar deteksi dapat berjalan dengan lancar maka harus didukung dengan pencahayaan minimum sebesar 100 lux. *Error* servo 1 sebesar 3.18° dengan akurasi sebesar 97.69%, *error* servo 2 sebesar 11.99° dengan akurasi sebesar 89.57%, dan *error* servo 3 sebesar 18.81° dengan akurasi sebesar 88%. Untuk tingkat keberhasilan robot lengan menyelesaikan misi sebesar 100% untuk seluruh kondisi peletakan dan seluruh warna.

Daftar Pustaka

- [1] W. M. Sanjaya and D. Anggraeni, "Sistem Kontrol Robot Arm 5 DOF Berbasis Pengenalan Pola Suara Menggunakan Mel-Frequency Cepstrum Coefficients (MFCC) dan Adaptive Neuro-Fuzzy Inference System (ANFIS)," *Wahana Fis.*, 2016, doi: 10.17509/wafi.v1i2.4277.
- [2] A. Nurcahyo and H. Susanto, "Desain Dan Implementasi Pengendali Putaran Motor Dc Menggunakan Voice Command Berbasis Internet of Things," vol. 6, no. 02, pp. 49–59, 2020.
- [3] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse Kinematics Techniques in Computer Graphics: A Survey," *Comput. Graph. Forum*, vol. 37, no. 6, pp. 35–58, 2018, doi: 10.1111/cgf.13310.
- [4] T. and others Andono, Pulung Nurtantio and Sutojo, *Pengolahan citra digital*. Penerbit Andi, 2017.
- [5] I. Nurdin, "Pengolahan Citra Mendeteksi Objek Berdasarkan Warna Menggunakan Metode Blobs Detection," *Univ. Telkom*, 2018.
- [6] A. Purwanto and S. Lutfi, "Pengendalian Lampu Rumah Berbasis Google Asisstant Melalui Smartphone Menggunakan Nodemcu-12e Esp8266 Di Nuke Komputer Service," *J. Himsya Tech*, vol. 20, no. 2, pp. 1–6, 2019.
- [7] Z. B. Abilovani, W. Yahya, and F. A. Bakhtiar, "Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, 2018.
- [8] A. Fahriannur, R. Mardiyanto, and M. Siswanto, "Sistem Pelacakan Objek Menggunakan Kombinasi Algoritma Optical Flow dan Template Matching," *J. Teknol. dan Sist. Komput.*, 2018, doi: 10.14710/jtsiskom.6.1.2018.13-17.
- [9] D. A. Prabowo and D. Abdullah, "Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking," *Pseudocode*, 2018, doi: 10.33369/pseudocode.5.2.85-91.