

KONTROL MULTI MOTOR SERVO DENGAN *JOYSTICK* *VIRTUAL* BERBASIS *ANDROID*

I Gst. Agung Mahendra Sedana Yoga^{1*}, I Nyoman Piarsa², Putu Wira Buana³

Teknologi Informasi, Fakultas Teknik, Universitas Udayana¹

Email*: agung_mahendra079@student.unud.ac.id

Teknologi Informasi, Fakultas Teknik, Universitas Udayana²

Email: manpits@unud.ac.id

Teknologi Informasi, Fakultas Teknik, Universitas Udayana³

Email: wbhuana@unud.ac.id

(*) *Corresponding Author*

ABSTRAK

Penelitian ini berfokus pada perancangan sistem kontrol berbasis *Internet of Things* (IoT) untuk beberapa motor servo. Sistem ini dilengkapi dengan pengontrol berbasis *Android* yang menggunakan *joystick virtual*, memungkinkan kontrol secara *online* dan *real-time*. Penggunaan *joystick virtual* pada aplikasi *smartphone Android* memungkinkan operator untuk mengendalikan perangkat dari jarak jauh, sehingga dapat meningkatkan keselamatan dan efisiensi kerja, terutama di area yang berbahaya atau sulit dijangkau. Tujuan dari penelitian ini adalah untuk mengevaluasi keberhasilan dan akurasi pergerakan servo guna memastikan performa yang optimal. Hasil penelitian ini bertujuan untuk membuktikan bahwa teknologi IoT dapat menjadi referensi dalam menerapkan kontrol *wireless* pada berbagai sistem otomatisasi, sehingga dapat meningkatkan kualitas operasional dalam berbagai aplikasi. Hasil pengujian menunjukkan bahwa kontrol *joystick* melalui perangkat *Android* dapat membantu proses pengendalian secara daring dan dapat dioperasikan secara efektif. Namun, keterbatasan pada perangkat keras, seperti motor servo dan bracket, menyebabkan ketepatan pemindahan barang berkurang, yang pada akhirnya mengakibatkan kegagalan pemindahan barang dalam beberapa percobaan dengan tingkat keberhasilan mencapai 70 persen. Meskipun demikian, penelitian ini dapat menjadi acuan bagi pengembangan lebih lanjut dalam bidang terkait, khususnya dalam meningkatkan akurasi dan stabilitas sistem kontrol berbasis IoT.

Kata kunci: *Internet of Things* (IoT), motor servo, kontrol jarak jauh, *joystick virtual*, *Android*

ABSTRACT

This research focuses on designing an Internet of Things (IoT)-based control system for multiple servo motors. The system is equipped with an Android-based controller using a virtual joystick, enabling online and real-time control. The use of a virtual joystick on an Android smartphone application allows operators to control the device remotely, enhancing safety and work efficiency, especially in hazardous or hard-to-reach areas. The objective of this research is to evaluate the success and accuracy of servo movements to ensure optimal performance. The findings aim to demonstrate that IoT technology can serve as a reference for implementing wireless control in various automated systems, improving operational quality in different applications. Experimental results show that joystick control via an Android device facilitates online operation and can be effectively

utilized. However, hardware limitations, such as servo motors and brackets, reduce the accuracy of object transfer, leading to failure in some trials, with a success rate of 70 percent. Nevertheless, this research can serve as a reference for future developments in the field, particularly in enhancing the accuracy and stability of IoT-based control systems.

Keywords: Internet of Things (IoT), servo motor, remote control, virtual joystick, Android

1. PENDAHULUAN

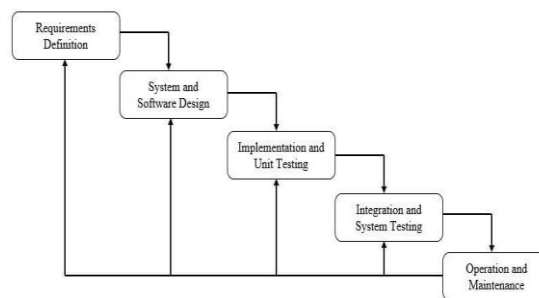
Perkembangan teknologi yang digunakan pada masa kini membuat pekerjaan-pekerjaan yang menggunakan sistem konvensional secara bertahap digantikan dengan sistem yang lebih canggih [1]. Tidak hanya dalam bidang komunikasi, teknologi mulai diterapkan pada berbagai macam bidang seperti pendidikan, transportasi, kesehatan hingga kebutuhan sosial masyarakat. Salah satu tantangan dalam penggunaan mesin dan perangkat otomatis adalah ketergantungan pada operator manusia yang harus mengoperasikan perangkat secara langsung. Hal ini dapat menjadi kurang efisien dan berisiko, terutama jika terjadi kesalahan manusia, kondisi lingkungan yang sulit, atau kegagalan mekanis. Selain itu, operator juga dapat terpapar pada getaran, kebisingan, dan debu yang berdampak negatif pada kesehatan dalam jangka panjang [2]. Oleh karena itu, sistem otomatisasi dengan kontrol jarak jauh menjadi solusi yang lebih praktis dan efisien.

Permasalahan ini dapat diatasi dengan penerapan *Internet of Things* (IoT), yang merupakan sebuah konsep di mana perangkat tertentu memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke komputer [3]. Proses pengumpulan data dalam sistem IoT memerlukan antarmuka yang berfungsi sebagai pengumpul data, yang kemudian dapat diakses kembali ke dalam perangkat lunak untuk dianalisis [4]. IoT memungkinkan perangkat untuk berkomunikasi dengan sensor, aktuator, sistem operasi, mikrokontroler, dan alat analitis dengan konsep utama yaitu saling terhubung dan bekerja untuk melakukan tugas tertentu. Integrasi IoT dengan mikrokontroler seperti *Arduino* dapat membantu dalam mengatasi berbagai permasalahan [5].

Alternatif yang dapat dilakukan untuk menemukan solusi adalah dengan perancangan sistem kontrol IoT menggunakan beberapa motor servo yang memiliki sistem pengontrol *Android* berbentuk *joystick virtual*. Sistem ini memungkinkan kontrol secara *online* dan *real-time* dalam proses penggunaannya. Konsep komunikasi IoT membantu menghubungkan sensor dan perangkat elektronik serta objek fisik lainnya agar dapat diakses melalui internet, memungkinkan pertukaran informasi dan kontrol antara aplikasi dan perangkat [6]. Kontrol akan diterapkan pada aplikasi smartphone berbasis *Android* yang menggunakan *joystick virtual* sebagai input kontrol pada susunan motor servo, sehingga teknologi ini memungkinkan operator untuk mengoperasikan perangkat dari jarak jauh, meningkatkan keselamatan dan efisiensi kerja, terutama di area berbahaya atau sulit dijangkau. Penelitian ini diharapkan dapat digunakan sebagai acuan maupun pengembangan pada penelitian selanjutnya untuk menyelesaikan permasalahan terkait otomatisasi sistem kontrol berbasis IoT. Penggunaan IoT dalam penelitian ini memberikan keuntungan dalam pengumpulan data yang akurat, real-time, dan terus-menerus [7].

2. METODE

Penelitian ini akan menerapkan metode *Software Development Life Cycle* (SDLC) dengan menggunakan model *Waterfall*. Model *Waterfall* adalah suatu pendekatan pengembangan perangkat lunak yang mengikuti urutan tahapan, serupa dengan aliran air terjun [8]. Dalam model *Waterfall*, tahapan-tahapan yang berbeda harus diselesaikan secara berurutan, dan hanya dilanjutkan ke tahapan berikutnya setelah menyelesaikan tahapan sebelumnya dengan baik. Model *Waterfall* memiliki sifat rekursif, yang berarti bahwa setiap tahapan dapat dikerjakan berulang sampai mencapai tingkat kesempurnaan yang diinginkan. Pada tahap perancangan alat ini memiliki tujuan agar pada saat proses pembuatan alat bisa berjalan dengan baik sesuai dengan apa yang diharapkan sampai akhir hingga alat tersebut bisa digunakan secara sempurna sesuai dengan keinginan [9]. Berikut adalah gambaran tentang tahapan-tahapan dalam model *Waterfall*.



Gambar 1. Metode penelitian

Model *Waterfall* pada Gambar 1. adalah model pengembangan perangkat lunak yang mengikuti serangkaian tahapan yang ditentukan secara berurutan, di mana setiap tahapannya bergantung pada penyelesaian tahapan sebelumnya. Dalam model *Waterfall*, ada beberapa tahapan utama, yaitu:

- 1) *Requirement Definition* (Definisi Kebutuhan) adalah definisi kebutuhan, proses melibatkan analisis kebutuhan pengguna, kebutuhan fungsional, dan kebutuhan non-fungsional. Hasil dari tahap ini adalah dokumen spesifikasi kebutuhan seperti alat dan bahan, perangkat lunak, dan bahasa pemrograman.
- 2) *System and Software Design* (Desain Sistem dan Perangkat Lunak) Setelah kebutuhan didefinisikan, tahap berikutnya adalah merancang sistem dan perangkat lunak. Ini mencakup perencanaan arsitektur sistem, rancangan perangkat lunak, dan perancangan basis data jika relevan. Hasil dari tahap ini adalah dokumen desain yang merinci bagaimana sistem akan dibangun seperti gambaran umum dan desain *mockup* aplikasi.
- 3) *Implementation and Unit Testing* (Implementasi dan Pengujian Unit) Tahap ini melibatkan implementasi sebenarnya dari perangkat lunak berdasarkan desain yang telah dibuat. Proses meliputi pembuatan kode dan sistem, menguji unit-unit perangkat lunak secara terpisah, dan mengintegrasikan unit-unit tersebut.
- 4) *Integration System* (Integrasi Sistem) Setelah unit-unit perangkat lunak diuji secara terpisah, seluruh perangkat diintegrasikan menjadi sistem yang lengkap. Tahap ini menguji bagaimana komponen-komponen sistem berinteraksi satu sama lain. Inti dari tahap ini adalah memastikan bahwa sistem berfungsi sebagai keseluruhan dan bahwa komunikasi antar komponen berjalan dengan baik. Penelitian ini juga mengintegrasikan *Broker* MQTT HiveMQ sebagai protokol komunikasi pada perangkat IoT

- 5) *Operation and Maintenance* (Operasi dan Pemeliharaan) Tahap terakhir adalah operasi dan pemeliharaan sistem. Proses melibatkan peluncuran sistem ke lingkungan untuk memastikan bahwa sistem tetap berjalan dengan baik. Pemeliharaan dapat mencakup pembaruan perangkat lunak, perbaikan kesalahan, dan peningkatan fitur.

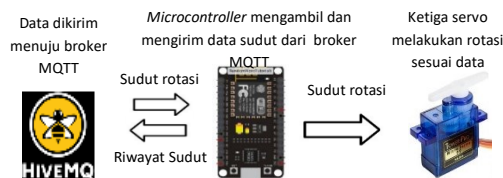
Gambaran Umum

Penelitian ini menggunakan beberapa macam perangkat yang saling diintegrasikan. Berikut ini merupakan gambaran umum mengenai perancangan sistem kontrol multi motor servo dengan *joystick virtual Internet of Things* (IOT).



Gambar 2. Gambaran umum kontrol

Gambar 2. merupakan sistem kontrol multi motor servo menggunakan input dari user untuk mengontrol motor servo secara langsung menggunakan *joystick virtual* yang tersedia pada aplikasi *Android*. Aplikasi *Android* akan menerima input dan mengirimkan data ke mikrokontroler melalui *broker* MQTT HiveMQ. MQTT *broker* berfungsi sebagai media yang menghubungkan antara *node publisher* dan *node subscriber* [10].



Gambar 3. Gambaran umum kontrol

Gambar 3. merupakan gambaran proses dari mikrokontroler yang akan mengambil data dari MQTT dan akan memproses data tersebut lalu mengeksekusi perintah untuk menggerakkan motor servo. Hasil pergerakan kinerja motor servo dapat dimatikan atau dihidupkan dengan menggunakan *handphone Android* [11].

Protokol Komunikasi

Pada penelitian ini, konsep dari penerapan *internet of things* adalah untuk memungkinkan kontrol beberapa motor servo sekaligus melalui aplikasi *Android* yang dapat dilakukan secara *wireless* tanpa batasan jarak tertentu atau dapat dikatakan secara *online*. Pada percobaan ini, pengguna akan mengubah kontrol arah gerak motor servo melalui aplikasi *Android* yang sudah dirancang, dan aplikasi *Android* tersebut akan melakukan komunikasi dengan protokol MQTT dari HiveMQ untuk mengirimkan informasi perintah gerak. Selanjutnya *microcontroller* akan melakukan pengambilan data dari *broker* MQTT dan akan melakukan perintah gerak motor servo sesuai data yang diterima.

Pengguna akan melakukan kontrol dengan *joystick virtual* pada aplikasi *Android* dan aplikasi tersebut akan otomatis mengirimkan data kepada HiveMQ yang nantinya akan diakses oleh NODEMCU ESP8266 sebagai perintah untuk mengontrol motor servo.



Gambar 4. Protokol komunikasi

Gambar 4. merupakan gambaran mengenai proses transfer data antara aplikasi *Android* dan *microcontroller*. Pada proses transfer data, *microcontroller* NodeMCU berperan sebagai *subscriber* dan *publisher* serta sebagai *gateway* untuk berkomunikasi dengan HiveMQ [12], begitu juga dengan aplikasi pengguna yang menjadi *subscriber* sekaligus *publisher* terhadap HiveMQ.

Kebutuhan *Hardware*

Hardware adalah semua komponen fisik yang membentuk komputer atau perangkat elektronik. Ini mencakup semua perangkat keras fisik yang memiliki tugas dan fungsi tertentu. Berikut merupakan tabel perangkat keras yang digunakan pada penelitian ini.

Tabel 1. Perangkat keras

Alat	Fungsi
NodeMCU ESP8266	Mikrokontroler yang berfungsi untuk menerima dan mengolah data dan melakukan eksekusi perintah untuk menggerakkan motor servo. Koneksi sinyal yang didapat oleh ESP8266 mempengaruhi dalam mendapatkan sinyal dari satelit sehingga waktu tunggu untuk mendapatkan sinyal tidak menentu [13].
<i>Smartphone</i>	<i>Smartphone</i> digunakan sebagai media untuk melakukan kontrol pada aplikasi. Aplikasi akan menyediakan sebuah <i>joystick virtual</i> untuk menerima <i>input</i> dari <i>user</i> .
Kabel Jumper	Kabel berfungsi untuk menghubungkan komponen elektronik, seperti mikrokontroler, sensor, dan motor servo.
Motor Servo (SG90)	Motor servo berfungsi untuk menggerakkan sendi-sendi lengan robot dengan presisi tinggi sesuai dengan perintah yang diterima, memungkinkan kontrol gerakan yang akurat. Servo biasa berfungsi untuk mendorong atau memutar objek yang membutuhkan kontrol dengan presisi tinggi dalam hal posisi sudut, akselerasi, dan kecepatan [14].
<i>Bracket</i>	<i>Bracket</i> motor servo akan digunakan sebagai bahan untuk membuat <i>prototype</i> media yang digunakan sebagai wadah untuk semua perangkat sistem kontrol multi motor servo
Power Supply Breadboard	Modul ini memiliki banyak pin <i>output</i> yang digunakan untuk memberikan daya 5v pada ketiga buah servo.

Tabel 1. merupakan kebutuhan *hardware* pada penelitian kontrol multi motor servo dengan *joystick virtual*. Terdapat beberapa kebutuhan *hardware* pada penelitian ini yaitu, NodeMCU ESP8266, *Smartphone*, kabel jumper, Motor Servo, dan *Bracket*.

Kebutuhan Software

Software merupakan program komputer yang mengontrol operasi perangkat keras. Perangkat mencakup sistem operasi, aplikasi, game, perangkat lunak utilitas, dan kode program yang menginstruksikan perangkat keras untuk melakukan tugas-tugas tertentu. Berikut merupakan tabel perangkat lunak yang digunakan pada penelitian ini.

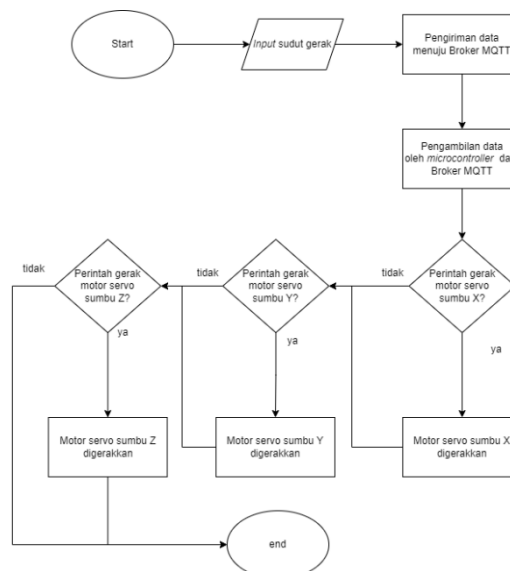
Tabel 2. Perangkat lunak

<i>Perangkat Lunak</i>	<i>Fungsi</i>
<i>Android Studio</i>	<i>Android Studio</i> merupakan IDE yang digunakan untuk mengembangkan aplikasi berbasis <i>Android</i> . <i>Android studio</i> juga digunakan sebagai alat kontrol bagi <i>user</i> untuk melakukan kontrol multi motor servo.
<i>Arduino IDE</i>	<i>Arduino IDE</i> merupakan IDE yang digunakan untuk mengembangkan program atau perintah pada <i>microcontroller</i> . Program yang dirancang akan difungsikan untuk menggerakkan motor servo tertentu. Bahasa pemrograman <i>arduino</i> merupakan bahasa C yang sudah disederhanakan <i>syntax</i> bahasa pemrogramannya sehingga mempermudah dalam mempelajari dan mendalami mikrokontroler [15].
HiveMQ	HiveMQ berfungsi media komunikasi antara aplikasi android dan <i>microcontroller</i> . HiveMQ memungkinkan aplikasi dan <i>microcontroller</i> saling bertukar data atau informasi terkait kontrol dan pergerakan motor servo.

Tabel 2. merupakan kebutuhan *hardware* pada penelitian kontrol multi motor servo dengan *joystick virtual*. Terdapat beberapa kebutuhan *software* pada penelitian ini yaitu *Android Studio*, *Arduino IDE*, dan HiveMQ.

Flowchart

Flowchart adalah representasi visual dari alur atau algoritma suatu proses atau tugas menggunakan simbol-simbol grafis untuk menggambarkan langkah-langkah, keputusan, dan aliran proses. Berikut merupakan *flowchart* dari sistem kontrol multi motor servo.

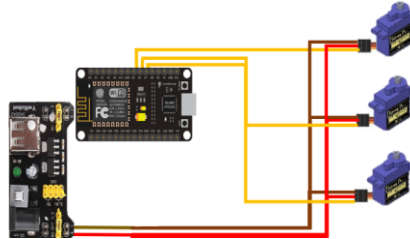


Gambar 5. *Flowchart* sistem kontrol

Gambar 5 merupakan *flowchart* sistem kontrol multi motor servo. Proses sistem meliputi proses *input*, komunikasi perintah, dan operasi gerak motor servo. Proses *input* oleh pengguna dilakukan dengan *joystick virtual* pada aplikasi *Android* lalu aplikasi akan mengirimkan data menuju broker MQTT sehingga *microcontroller* dapat mengambil informasi perintah gerak pada masing-masing servo.

Rancangan Perangkat

Pada penelitian ini menggunakan konsep *internet of things* yang terdiri dari beberapa komponen. Komponen tersebut akan dirakit sedemikian rupa sehingga dapat bergerak dengan baik. Rangkaian motor servo diletakkan pada *bracket* yang sudah dibentuk dan komponennya terhubung dengan *microcontroller* menggunakan kabel *jumper*. Berikut merupakan konfigurasi seluruh komponen menggunakan kabel *jumper*.



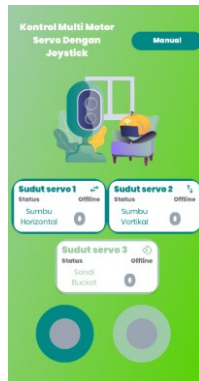
Gambar 6. Konfigurasi motor servo

Gambar 6. merupakan *wiring diagram* yang menggambarkan konfigurasi kabel yang menghubungkan seluruh alat. Setiap motor servo dihubungkan dengan 3 jenis pin yaitu pin *ground*, pin *VCC*, dan pin *In*. Pin *ground* berfungsi sebagai pengantar arus potensial nol, pin *VCC* sebagai pengantar arus bertegangan, dan pin *In* sebagai pin transfer data *input*. Ketiga buah servo akan menggunakan daya yang berasal dari *power supply breadboard* yang memiliki tegangan 5v.

3. HASIL DAN PEMBAHASAN

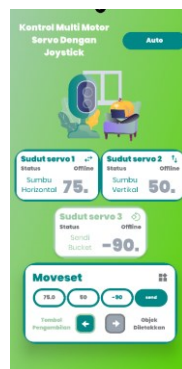
Rancangan aplikasi *Android* dalam penelitian ini difokuskan pada pengembangan antarmuka pengguna yang intuitif dan responsif untuk mengontrol motor servo yang disusun menggunakan *bracket*. Aplikasi ini dirancang dengan tujuan utama menyediakan *joystick virtual* yang memungkinkan operator untuk mengendalikan pergerakan motor servo secara *real-time* dari jarak jauh. Aplikasi *Android* dirancang menggunakan *Android studio* yang menggunakan bahasa pemrograman XML sebagai *user interface* dan bahasa *Java* sebagai bahasa pemrosesan data dan logika. *User Interface* (UI) pada aplikasi kontrol dibuat dengan tampilan yang menarik agar pengguna dapat dengan mudah mengerti tata cara pengoperasian aplikasi [16].

Aplikasi terdiri menu *login* dan menu kontrol. Menu *login* digunakan untuk memverifikasi pengguna yang sedang mengontrol servo. Ketika terdapat pengguna lain yang mencoba mengakses aplikasi, maka aplikasi pengguna tersebut akan *force close* selagi pengguna aplikasi pertama masih melakukan kontrol. Dapat dilihat pada Gambar 7.



Gambar 7. UI halaman kontrol manual

Menu kontrol terdiri dari 2 buah *joystick* dan status keaktifan serta sudut dari ketiga motor servo. *Joystick* sebelah kiri digunakan untuk mengontrol motor servo horizontal dan vertikal, sedangkan *joystick* sebelah kanan untuk mengontrol motor servo sendi *bucket* (vertikal).

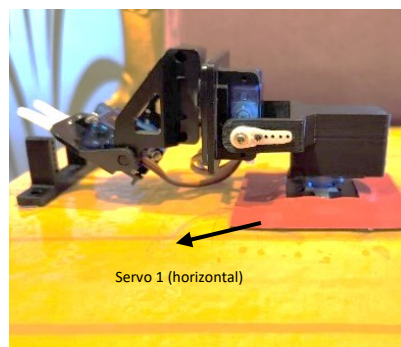


Gambar 8. UI halaman kontrol otomatis

Aplikasi juga menyediakan mode kontrol otomatis yang dapat digunakan dengan memberikan *input* letak objek yang akan dipindahkan. *Input* terdiri dari 3 buah titik sudut sumbu X, Y, dan *Bucket* serta tombol pengambilan dan peletakkan.

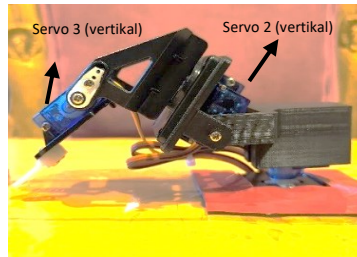
Pengujian Manual

Pengujian manual meliputi percobaan menggerakkan *joystick* pada aplikasi android untuk menggerakkan ketiga buah servo. Servo 1 dan servo 2 dapat dikontrol melalui *joystick* sebelah kiri yang digunakan untuk bergerak secara horizontal dan vertikal. Servo 3 dapat dikontrol melalui *joystick* sebelah kanan yang digunakan untuk bergerak secara vertikal.



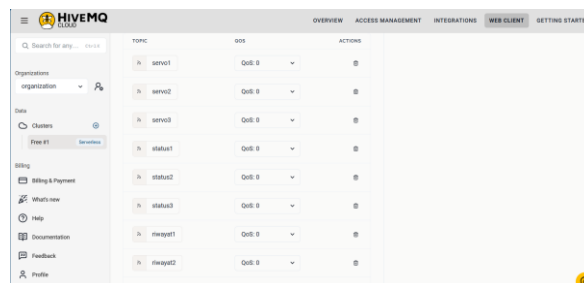
Gambar 9. Percobaan motor servo 1

Percobaan dilakukan untuk menguji pergerakan tiga motor servo menggunakan *joystick virtual*. Gambar 9 menunjukkan percobaan pada motor servo 1, yang terletak di bagian bawah, dengan pergerakan horizontal menggunakan *joystick* kiri pada sudut -90° . Sementara itu, Gambar 10 menunjukkan percobaan pada motor servo 3, yang terletak di bagian ujung (*bucket*), dengan pergerakan vertikal menggunakan *joystick* kanan pada sudut -90° .



Gambar 10. Percobaan motor servo 2 dan 3

Selain itu, pengujian juga dilakukan pada motor servo 1 dengan pergerakan horizontal ke sudut 90° , motor servo 2 yang berada di bagian tengah dengan pergerakan vertikal ke sudut -58° dan 59° , serta motor servo 3 dengan pergerakan vertikal ke sudut 90° . Percobaan pada motor servo 2 memiliki keterbatasan ruang gerak pada *bracket*, sehingga sudut maksimal tidak dapat mencapai -90° atau 90° .

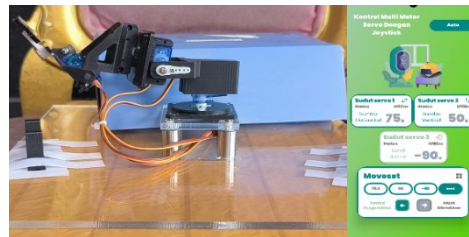


Gambar 11. *Dashboard* MQTT

Gambar 11 merupakan screenshot dari *dashboard broker* MQTT HiveMQ. Gambar menampilkan *topic* komunikasi data dari lalu lintas informasi pada *broker* yang dikirimkan oleh aplikasi menuju perangkat IoT. Data yang dikirimkan terbagi menjadi beberapa jenis yang terdiri dari sudut servo yang berisikan nilai dari sudut rotasi yang akan dijalankan, status keaktifan servo, dan riwayat sudut servo.

Pengujian Otomatis

Pengujian meliputi percobaan menggerakkan *joystick* pada aplikasi android secara otomatis menggerakkan ketiga buah servo untuk mengambil sebuah objek pada sudut tertentu lalu meletakkannya pada sudut tertentu yang diberikan pengguna. Contoh percobaan dilakukan dengan mengambil objek pada sudut -75° dan meletakkan objek tersebut pada sudut 75° . Dapat dilihat pada gambar 12.



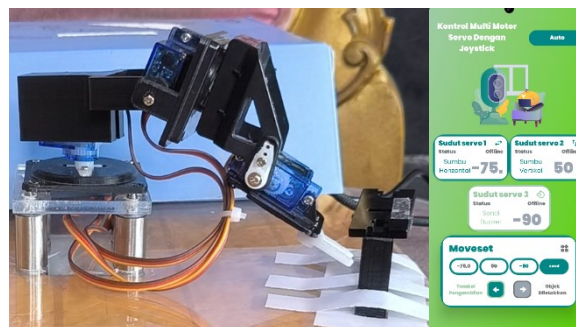
Gambar 12. Pengambilan objek

Pengujian otomatis memerlukan 3 buah input yang terdiri dari sudut sumbu x, y dan sumbu *bucket*. Seperti percobaan yang dilakukan diatas, ketiga servo akan bergerak untuk mengambil objek pada sudut sumbu x -75° dengan menekan tombol pengambilan^o.



Gambar 13. Hasil pengambilan

Gambar 13. Setelah pengguna menekan tombol ambil, objek akan diambil dan ketiga buah servo akan membawa objek tersebut ke sudut selanjutnya yang diberikan pengguna. Seperti percobaan yang dilakukan diatas, ketiga servo akan bergerak untuk membawa objek pada sudut sumbu x 0° .



Gambar 14. Peletakkan objek

Setelah pengguna menekan tombol jalan, objek akan diletakkan pada sudut 75° . Seperti percobaan yang dilakukan diatas, ketiga servo akan bergerak untuk meletakkan objek pada sudut 75° dan akan kembali bergerak menuju titik tengah. Aplikasi android akan memunculkan tombol reset untuk memungkinkan pengguna untuk melakukan pemindahan objek lagi sesuai sudut yang diberikan.

Pengujian Pemindahan Barang

Pengujian meliputi percobaan kontrol otomatis yang dilakukan berulang sebanyak 30 kali. Pengujian terdiri dari gerakan otomatis memindahkan objek dari sudut 0° menuju sudut 75° , lalu dilanjutkan dengan pemindahan barang dari sudut 75° dan sudut -75° , dan barang akan dikembalikan dengan pemindahan barang dari -75° menuju sudut 0° . Jika

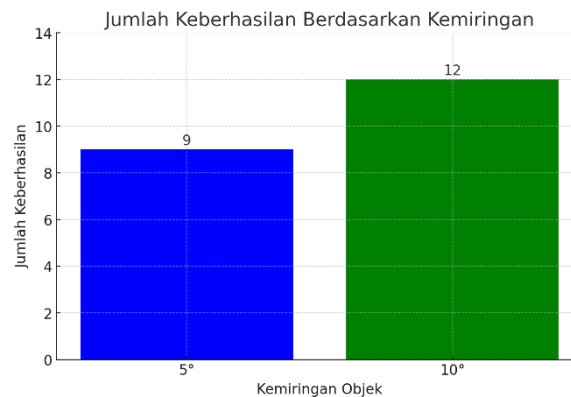
barang yang dipindahkan jatuh, maka pengujian dianggap gagal dan barang akan ditaruh kembali di sudut tujuan. Hasil dari 30 percobaan tersebut dijabarkan pada tabel 3.

Tabel 3. Hasil pengujian ketepatan

Percobaan ke-	Sudut		Kemiringan	Hasil
	Ambil	Letak		
1	Tengah (0°)	Kanan (75°)	~5°	Berhasil
2	Kanan (75°)	Kiri (-75°)	~10°	Berhasil
3	Kiri (-75°)	Tengah (0°)	-	Gagal (Objek jatuh)
4	Tengah (0°)	Kanan (75°)	~10°	Berhasil
5	Kanan (75°)	Kiri (-75°)	~5°	Berhasil
6	Kiri (-75°)	Tengah (0°)	~10°	Berhasil
7	Tengah (0°)	Kanan (75°)	-	Gagal (Objek jatuh)
8	Kanan (75°)	Kiri (-75°)	~10°	Berhasil
9	Kiri (-75°)	Tengah (0°)	~5°	Berhasil
10	Tengah (0°)	Kanan (75°)	~10°	Berhasil
11	Kanan (75°)	Kiri (-75°)	-	Gagal (Objek jatuh)
12	Kiri (-75°)	Tengah (0°)	~5°	Berhasil
13	Tengah (0°)	Kanan (75°)	~5°	Berhasil
14	Kanan (75°)	Kiri (-75°)	-	Gagal (Objek jatuh)
15	Kiri (-75°)	Tengah (0°)	~10°	Berhasil
16	Tengah (0°)	Kanan (75°)	-	Gagal (Objek jatuh)
17	Kanan (75°)	Kiri (-75°)	~10°	Berhasil
18	Kiri (-75°)	Tengah (0°)	~5°	Berhasil
19	Tengah (0°)	Kanan (75°)	~10°	Berhasil
20	Kanan (75°)	Kiri (-75°)	-	Gagal (Objek jatuh)
21	Kiri (-75°)	Tengah (0°)	~10°	Berhasil
22	Tengah (0°)	Kanan (75°)	-	Gagal (Objek jatuh)
23	Kanan (75°)	Kiri (-75°)	~5°	Berhasil
24	Kiri (-75°)	Tengah (0°)	-	Gagal (Objek jatuh)
25	Tengah (0°)	Kanan (75°)	~5°	Berhasil
26	Kanan (75°)	Kiri (-75°)	~10°	Berhasil
27	Kiri (-75°)	Tengah (0°)	-	Gagal (Objek jatuh)
28	Tengah (0°)	Kanan (75°)	~10°	Berhasil
29	Kanan (75°)	Kiri (-75°)	~5°	Berhasil
30	Kiri (-75°)	Tengah (0°)	~10°	Berhasil

Tabel 3 merupakan hasil dari percobaan gerakan otomatis untuk mengambil dan meletakkan objek pada sudut 0°, 75° dan -75°. Pengujian menunjukkan bahwa perpindahan objek sering kali mengalami kegagalan baik pada saat proses pengambilan objek maupun peletakkan objek. Pemindahan objek sangat bergantung pada posisi objek saat persiapan untuk proses pengambilan, hal ini disebabkan karena bentuk objek yang tidak memiliki pengait dan sangat mudah untuk bergeser pada *bracket* pondasi. Ketepatan pengujian otomatis diuji sebanyak 30 kali yang terdiri dari 21 kali proses pemindahan berhasil dan 9 kali proses mengalami kegagalan dan hasil tersebut menunjukkan bahwa percobaan ini memiliki nilai tingkat ketepatan sebesar 70%. Keberhasilan pemindahan objek juga dipengaruhi oleh kemiringan posisi objek yang dijabarkan pada bagian selanjutnya yaitu tingkat presisi objek.

Tingkat Presisi Objek



Gambar 15. Presisi objek

Gambar 15 merupakan grafik presisi objek berdasarkan hasil pengujian, tingkat keberhasilan pemindahan objek dipengaruhi oleh sudut kemiringan objek saat diletakkan. Dari 30 percobaan yang dilakukan, kemiringan sekitar 5° dan 10° menunjukkan tingkat keberhasilan yang lebih tinggi dibandingkan kondisi kemiringan lainnya yang dapat mengakibatkan objek mengalami ketidakseimbangan.

4. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diperoleh dari penelitian ini adalah berhasilnya perancangan dan pengujian sistem kontrol multi-motor servo yang dikendalikan melalui *joystick virtual* berbasis perangkat *Android* menggunakan protokol MQTT. Penerapan teknologi *Internet of Things* (IoT) memungkinkan transfer data secara efisien tanpa interaksi langsung antara manusia dan komputer. Sistem ini mengintegrasikan mikrokontroler *Arduino* dengan protokol MQTT, sensor, dan motor servo untuk menciptakan solusi kendali jarak jauh secara *online* dan *real-time*.

Hasil pengujian menunjukkan bahwa kontrol *joystick* melalui perangkat *Android* dapat membantu proses pengendalian secara daring dan dioperasikan secara efektif. Namun, keterbatasan pada perangkat keras, seperti motor servo dan *bracket*, menyebabkan ketepatan pemindahan barang berkurang, yang berdampak pada kegagalan pemindahan barang dalam beberapa percobaan, dengan tingkat keberhasilan mencapai 70 persen. Penelitian ini diharapkan menjadi langkah awal untuk pengembangan lebih lanjut dalam sistem kontrol jarak jauh berbasis IoT, khususnya dalam meningkatkan efisiensi dan keandalan pengoperasian perangkat yang menggunakan motor servo.

5. DAFTAR PUSTAKA

- [1] Abadi, S. C., Prafanto, A., Eriyadi, M., Suryadi, A., & Ramadi, G. D. (2020). Penerapan Protokol Publish / Subscribe Messaging System Pada Sistem Wireless Building Network Berbasis Nrf24L01. *Jurnal Teknologi Terapan*, 6(2), 172.
- [2] Effendi, R., Salahuddin, & Usmardi. (2020). Kendali Senapan Menggunakan Joystick Berbasis Mikrokontroler ATMEGA32 dengan Modul NRF24L01. *JURNAL LITEK : Jurnal Listrik Telekomunikasi Elektronika*, 17(2), 61–68. http://www.slagcoin.com/joystick/pcb_diagrams/ps1_
- [3] Handayani, S., Nuzuluddin, M., & 'Alimuddin, 'Alimuddin. (2024). Rancang Bangun Smart Hidroponik Berbasis IoT dengan Aplikasi Android. *Jurnal*

- PRINTER: Jurnal Pengembangan Rekayasa Informatika Dan Komputer*, 2(1), 46–59. <https://doi.org/10.29408/jprinter.v2i1.23799>
- [4] Hidayat, A. F., & Safitri, A. E. (2024). Rancang Bangun Tempat Sampah Pintar pada Rumah Tangga Berbasis IoT dan Android. *JORAPI : Journal of Research and Publication Innovation*, 2(1), 1414–1418.
- [5] Jamil, M., & Said, M. (2018). The Utilization of Internet of Things for Multi Sensor Data Acquisition Using Thingspeak. *Jurnal Ilmiah Pendidikan Teknik Elektro*, 3(1), 13–18.
- [6] Natsir, M., Bayu Rendra, D., & Derby Yudha Anggara, A. (2019). Implementasi IoT untuk Sistem Kendali AC Otomatis pada Ruang Kelas di Universitas Serang Raya. *Jurnal PROSISKO*, 6(1), 69–72. <https://www.arduino.cc/en/Products/Counterfeit>
- [7] Pamungkas, I. K. K. A., Hartati, R. S., & Kumara, I. N. S. (2021). Literature Review Of The Internet Of Things On The Drinking Water Management System. *International Journal of Engineering and Emerging Technology*, 6(2), 1–9.
- [8] Devanugraha, A., Dharma, E. M., & Putri, I. Gst. A. P. D. (2024). Rancang Bangun Aplikasi Keuangan UMKM Berbasis Mobile dengan Metode Waterfall. *Jurnal Teknologi Informasi Dan Komputer*, 10(1), 17–27.
- [9] Ramdan, R., Lasmadi, L., & Setiawan, P. (2022). Sistem Pengendali On-Off Lampu dan Motor Servo sebagai Penggerak Gerendel Pintu Berbasis Internet Of Things (IoT). *AVITEC*, 4(2), 211. <https://doi.org/10.28989/avitec.v4i2.1317>
- [10] Rifa'i, A., Rasyid, M. U. H. Al, & Gunawan, A. I. (2021). Sistem Pemantauan dan Kontrol Otomatis Kualitas Air Berbasis IoT Menggunakan Platform Node-Red untuk Budidaya Udang. *Jurnal Teknologi Terapan*, 7(1), 19–26.
- [11] Rizqy, M. D. A., & Paramytha, N. (2024). Rancang Bangun System Keamanan pada Smarthome Menggunakan Aplikasi Android Berbasis IoT. *Ensiklopedia Of Journal*, 7(1), 232–238.
- [12] Somantri, Insany, G. P., & Novianti, A. (2023). Rancang Bangun Aplikasi Pendeteksi Tingkat Kepekatan Asap Hasil Pembakaran Berbasis Internet of Things. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 7(2), 721–730.
- [13] Somantri, Insany, G. P., & Novianti, A. (2024). Rancang Bangun Sistem Monitoring Rental Sepeda Berbasis Iot Dan Android. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, 9(2), 720–729.
- [14] Widasari, C., & Zulkarnain, L. A. (2021). Jurnal Politeknik Caltex Riau Rancang Bangun Sistem Monitoring Penggunaan Air PDAM Berbasis IoT. *Jurnal Komputer Terapan*, 7(2), 153–162. <https://jurnal.pcr.ac.id/index.php/jkt/>
- [15] Yufrida, A. A., Rahayu, L. P., & Syahbana, D. F. (2021). Implementasi Kontrol Torsi Motor Servo Menggunakan Metode PI pada Sistem Automatic Pallet Dispenser. *Jurnal Teknik ITS*, 10(2), B244–B250.
- [16] Yuniarti, E., Hidayat, R., Ardian, E., & Studi Teknik Elektro, P. (2022). Rancang bangun Sistem Monitoring Motor Servo dan Jumlah Pakan Ikan Berbasis Internet of Things. 7(1). <https://doi.org/10.31851/ampere>