

Implementation of the Internet of Things in a Disinfectant Robot Remote Control System

Implementasi *Internet Of Things* Pada Sistem Pengendali Jarak Jauh Robot Disinfektan

Christine J. Kambey, Sherwin R. U. Sompie, Agustinus Jacobus

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails: kambeychristine@gmail.com, aldo@unsrat.ac.id, a.jacobus@unsrat.ac.id

Received: 27 February 2024; revised: 03 December 2024; accepted: 03 March 2025

Abstract — Disinfectant robot is an omni-wheeled medium size robot equipped with UVC LED. The concept of Internet Of Things can be used in designing an android-based room disinfectant robot controller. The implementation of the IoT in this research that is IoT platform, ThingSpeak. ThingSpeak was used in collecting data, displaying graphs, and analyzing the data flow of the disinfectant robot. This research aims to develop an IoT-based remote control system for disinfectant robots via android. The methods used in this research are literature study, location observation, developing and construction of disinfectant robots, and system trials. Experimentation has 3 different stages, hardware testing, software testing, and then over-all system testing. The results of this study such as data received by ThingSpeak which has a percentage of success 94.85% for 45 minutes testing. The success rate of sending maneuver data is 90%. The percentage of successful data transmission from the android application to the disinfectant robot has an average of 85%. In 9 minutes 16 seconds the voltage on the battery on the robot decreased from 24,7 Volts to 24,17 Volts. The distance between disinfectant robot and the internet connection source greatly affects the time delay in sending data to the cloud.

Key words—Android; Arduino; Internet of Things; Robot; ThingSpeak.

Abstrak — Robot disinfektan adalah robot beroda berukuran sedang dengan manuver omni-wheel yang dilengkapi dengan LED UVC. Konsep IoT dapat digunakan dalam perancangan aplikasi pengendali robot disinfektan ruangan berbasis android melalui smartphone. Implementasi *Internet Of Things* dalam penelitian ini dapat dilihat dalam penggunaan platform IoT yaitu ThingSpeak. ThingSpeak memiliki peran dalam menampung data, menampilkan grafik, dan menganalisa aliran data dari robot disinfektan. Penelitian ini bertujuan mengembangkan sebuah sistem pengendali jarak jauh berbasis *Internet Of Things* untuk robot disinfektan menggunakan aplikasi *mobile* android. Metode yang digunakan dalam penelitian ini adalah studi literatur, observasi lokasi penelitian, perancangan dan pembuatan robot disinfektan, dan uji coba sistem. Pengujian dibagi menjadi 3 tahap, yaitu pengujian *hardware*, pengujian *software*, kemudian pengujian sistem. Hasil dari penelitian ini mencakup data yang diterima ThingSpeak memiliki persentase keberhasilan yaitu 94,85% dalam waktu pengujian 45 menit. Tingkat keberhasilan pengiriman data manuver memiliki tingkat keberhasilan 90%. Persentase keberhasilan pengiriman data dari aplikasi android ke robot disinfektan memiliki rata-rata 85%. Dalam pengujian selama 9 menit 16 detik tegangan pada baterai yang ada pada robot menurun dari 24,7 Volt ke 24,17 Volt. Jarak antara robot

disinfektan ke sumber koneksi internet sangat mempengaruhi time delay pada pengiriman data ke cloud.

Kata kunci —Android; Arduino; Internet of Things; Robot; ThingSpeak.

I. PENDAHULUAN

Pandemi *Coronavirus Disease 2019* (COVID-19) yang terjadi sejak awal tahun 2020 tidak hanya berdampak di dunia kesehatan, melainkan pada perekonomian, pariwisata, pendidikan, dan seterusnya. Di era *new normal*, salah satu tindakan pencegahan penyebaran COVID-19 yang dapat dilakukan yaitu dengan mendesinfeksi ruangan kerja secara berkala. Sinar ultraviolet C (UVC) merupakan sinar UV yang paling ampuh dalam merusak DNA mikroba dan virus sehingga dapat digunakan untuk disinfeksi ruangan. Namun, manusia tidak seharusnya terpapar langsung dengan sinar UVC apalagi dalam waktu yang lama karena dapat merusak DNA dan memberikan efek terbakar pada kulit [1].

Dengan perkembangan teknologi yang semakin pesat, robot semakin banyak digunakan untuk membantu pekerjaan manusia contohnya robot *vacuum cleaner*. Robot merupakan sebuah alat yang terdiri dari 3 komponen utama yaitu komponen penggerak, komponen elektronika, dan komponen yang bisa diprogram berulang-ulang. Robot bukanlah makhluk hidup yang memiliki DNA dan rentan terhadap sinar UV sehingga dapat digunakan sebagai alat disinfeksi ruangan menggunakan sinar ultraviolet.

Saat ini, *Internet of Things* (IoT) sedang marak dikembangkan. IoT merupakan sebuah konsep konektivitas antara suatu objek dengan internet. Android merupakan sistem operasi *mobile* berbasis linux yang digunakan dalam *smartphone* dan *tablet*. Koneksi IoT dapat digunakan dalam perancangan aplikasi pengendali robot disinfektan ruangan berbasis android melalui *smartphone*. Dengan demikian operator dapat mengendalikan robot disinfektan secara jarak jauh dan tidak akan terpapar sinar UVC sama sekali.

A. Penelitian Terkait

Beberapa penelitian lainnya yang berkaitan dengan penelitian ini diantaranya dalam sebuah studi di *University of Pisa*, dikatakan telah banyak ketertarikan terhadap pengembangan strategi yang lebih efektif dan lebih

komprehensif untuk meningkatkan teknologi disinfeksi tanpa kontak langsung manusia[2]. Sistem disinfeksi ruangan yang dimaksud menggunakan teknologi yang tidak memerlukan kontak langsung manusia menggunakan sinar ultraviolet C. Sinar UVC ini mampu mendesinfeksi ruangan dan lebih efektif dari sistem disinfeksi ruangan manual.

Pandemi COVID-19 telah menyoroti potensi teknologi disinfeksi UV baik untuk udara dan permukaan benda yang merupakan media utama penularan penyakit. Peraturan publik di seluruh dunia sedang mempertimbangkan perangkat disinfeksi UV untuk permukaan yang sering disentuh dan sirkulasi aliran udara. Namun, pemahaman yang terbatas tentang aspek disinfeksi UV dapat menyebabkan penggunaan yang tidak tepat. Studi yang ada membahas dasar-dasar fenomena disinfeksi UV, parameter penting, dan protokol untuk sterilisasi UV yang aman bagi manusia[3]. Perangkat UV dapat digunakan sebagai disinfeksi, namun untuk penggunaan secara umum perlu diadakan sosialisasi kepada publik.

Internet of Things (IoT) dapat dianggap sebagai layanan yang dapat mencakup berbagai jenis permintaan sesuai dengan sistem yang dirancang. Diharapkan penerapan IoT meningkat pesat dan meluas ke banyak sektor. IoT memiliki masa depan yang menjanjikan dalam aktivitas sehari-hari, kebutuhan medis, dan pencegahan bahaya[4]. *Internet of Things* adalah teknologi yang dapat mengatasi tantangan dalam hal keamanan, kapasitas pemrosesan, dan mobilitas data.

Modul ESP32 adalah mikrokontroler dengan modul Wi-Fi yang banyak digunakan dalam penelitian *bertemakan Internet of Things* di beberapa tahun terakhir. ESP32 dikatakan lebih canggih dibandingkan dengan ESP6288 karena ESP32 memiliki spesifikasi lebih tinggi. ESP32 dapat menggunakan koneksi Wi-Fi ataupun *Bluetooth* untuk memproses data[5].

Mikrokontroler ESP32 banyak kali digunakan pada sebuah sistem *monitoring*. Seperti pada penelitian di Da-Yeh University, ESP32 dijadikan sebagai modul WiFi yang terintegrasi pada sistem *monitoring* kualitas air pada budidaya perairan atau perikanan[6]. ESP32 dihubungkan ke beberapa sensor yaitu sensor suhu, sensor pH, sensor pencemaran oksigen, dan sensor konduktivitas elektronik. Data dari sensor-sensor tersebut dapat dilihat pada webserver.

Penelitian lainnya menunjukkan bahwa sistem kualitas udara dapat diimplementasikan dengan menggunakan perangkat dengan harga terjangkau yaitu Arduino dan *ThingSpeak*. Dengan melalui berbagai sistem pengujian, aplikasi *mobile* yang dapat dengan mudah digunakan untuk memantau kualitas udara di seluruh spektrum dapat dikembangkan, sehingga dapat meningkatkan kesehatan penduduk[7].

Teknologi komunikasi saat ini memiliki potensi untuk membawa peningkatan besar dalam bidang SG (*Smart-Grid monitoring and control system*). Prosedur pemodelan langkah demi langkah dari dua pendekatan komunikasi yang diusulkan mengungkapkan kemungkinan penggunaan model komputasi *Fog* berbasis MATLAB / Simulink dalam interaksi real-time dengan model komputasi cloud berbasis platform *open-source ThingSpeak IoT*. Pendekatan berbasis *Cloud-Fog* telah menunjukkan pengurangan yang substansial dari total waktu simulasi dan jumlah data yang dikirim ke *Cloud*[8].

Konsep IoT dapat digunakan dalam perancangan arsitektur perangkat lunak untuk mengelola *smart-home*. Solusi yang diusulkan menyediakan modul yang diperlukan untuk mengelola sensor dan aktuator dari jarak jauh. Sistem *log-in* dapat diintegrasikan ke dalam arsitektur perangkat lunak rumah pintar. Modul pengenalan wajah dan modul keaktifan juga diintegrasikan ke dalam sistem rumah pintar melalui modul *log-in* sebagai metode yang disarankan sebelumnya dalam mengotorisasi dan mengautentikasi pengguna ke dalam sistem rumah pintar[9].

B. Robot

Kata robot berasal dari sebuah kata dalam Bahasa Ceko yaitu “*robota*” yang artinya “pekerja”. Pada dasarnya robot diciptakan untuk melakukan pekerjaan yang tidak ingin atau tidak dapat dilakukan oleh manusia. Setiap ahli memiliki pemahaman sendiri mengenai definisi robot. Secara umum dapat disimpulkan bahwa robot merupakan sebuah alat yang terdiri dari 3 komponen utama dengan 3 fungsi utama yaitu memahami dan mengumpulkan informasi di lingkungan sekitarnya, mengelola informasi tersebut, dan melakukan pergerakan secara fisik. Sebuah alat dapat diklasifikasikan sebagai robot apabila dapat melakukan salah satu atau lebih perilaku manusia, tidak harus meniru keseluruhannya.

C. Mikrokontroler Arduino Mega 2560

Arduino merupakan board elektronik bersifat *open-source* yang di dalamnya terdapat chip mikrokontroler[10]. Arduino Mega 2560 menggunakan chip mikrokontroler ATmega2560. Arduino Mega 2560 sering digunakan sebagai mikrokontroler pada robot karena banyaknya pin yang tersedia di *board* tersebut. Arduino Mega 2560 memiliki 54 pin digital I/O dimana 15 diantaranya dapat digunakan sebagai pin PWM dan 16 pin analog dan beberapa pin dengan fungsi khusus yaitu 4 port UART, 6 pin *external interrupt*, pin SPI, pin I2C, dan pin LED.

D. ESP32

ESP32 adalah modul mikrokontroler berukuran kecil dari Espressif yang dikatakan sebagai penerus dari ESP8266. ESP32 memiliki banyak variasi model yang bisa disesuaikan dengan kebutuhan[11]. Salah satunya yaitu ESP32-DevKitC, pada *board* ini semua pin ESP32 terbuka dan terlihat jelas sehingga mudah digunakan. Pengguna dapat langsung menyambungkan kabel *jumper* ke *header* pin ataupun meletakkannya pada *breadboard*.

E. PING))) Parallax

Sensor PING))) *Ultrasonic* merupakan sensor yang mengukur jarak benda. Sensor ini bekerja dengan melepaskan sinyal berfrekuensi tinggi kemudian mengukur berapa lama hingga diterima *echo*. Pengukurannya bergantung pada kecepatan suara di udara[12]. Kecepatan suara dapat berubah-ubah sesuai dengan faktor suhu, kelembapan, terlebih kualitas udara.

F. Arduino IDE

Arduino IDE merupakan *software* yang digunakan untuk melakukan pemrograman pada semua jenis *board* Arduino.



Gambar 1. Robot Disinfektan

Arduino IDE berfungsi layaknya *text editor* dimana kita dapat menulis, mengubah, memeriksa, dan melakukan verifikasi terhadap suatu kode. Arduino IDE digunakan dalam bahasa pemrograman C/C++[13]. Kode yang dibuat menggunakan Arduino IDE disebut dengan istilah *sketch* dan disimpan dalam ekstensi file .ino. Arduino IDE bersifat *open-source* artinya *library* Arduino dapat dikembangkan oleh siapa saja. Selain itu, Arduino IDE dapat digunakan untuk melakukan pemrograman pada *board* mikrokontroler selain Arduino.

G. Liquid Crystal Display 4x20 Character

Liquid Crystal Display 4x20 berarti ukuran karakter yang bisa ditampilkan yaitu 4 baris dan 20 kolom. LCD 4x20 ini menunjukkan karakter berwarna *pixel* putih pada *background* biru. LCD ini menggunakan *interface* I2C dengan 4 pin untuk mengendalikannya, yaitu pin SDA, pin SCL, pin VCC, dan pin GND. LCD 4x20 dapat bekerja pada tegangan 5V.

H. Motor DC PG-45

Motor DC PG-45 merupakan aktuator penggerak yang banyak digunakan pada robot berukuran besar. Motor DC berarti motor tersebut bekerja pada input tegangan DC 24V. Pada voltase 24 Volt, motor PG-45 dapat mencapai kecepatan putaran 3250 rpm. Tenaga yang dihasilkan motor PG-45 ketika diberi tegangan 24 Volt mencapai 22,5 Watt dan arus yang dihasilkan adalah 0.2 Ampere.

I. Disinfektan

Disinfektan adalah bahan yang digunakan untuk melakukan disinfeksi. Disinfeksi adalah pemusnahan bakteri patogen, biasanya dengan bahan kimia antiseptik[14]. Menurut *Encyclopedia Britannica*, disinfektan diartikan sebagai “bahan apa saja, seperti kreosot, alkohol, diaplikasikan pada benda mati untuk membunuh mikroorganisme”. Selain menggunakan alkohol, banyak orang sudah menggunakan sinar ultraviolet untuk melakukan disinfeksi. Tidak hanya di lingkungan layanan kesehatan, penggunaan disinfektan sangatlah penting untuk mengurangi kemungkinan kontaminasi COVID-19 di rumah, perkantoran, sekolah, gym, bangunan dengan akses publik, tempat ibadah, pertokoan, sarana transportasi dan restoran[15].

J. Ultraviolet

Sinar ultraviolet adalah gelombang elektromagnetik yang memiliki panjang pita 100-400 nm. Sinar ultraviolet terbagi atas 3 yaitu ultraviolet A (UVA), ultraviolet B (UVB), ultraviolet C (UVC) dengan rincian panjang gelombang secara berturut-turut 315-400 nm, 280-325 nm, dan 100-280 nm. Sinar ultraviolet C merupakan tipe sinar ultraviolet dengan panjang gelombang paling pendek dan sudah sangat dikenal sebagai pembasmi kuman penyakit atau mikroorganisme. Sinar UVC telah digunakan untuk mendesinfeksi ruangan di layanan kesehatan.

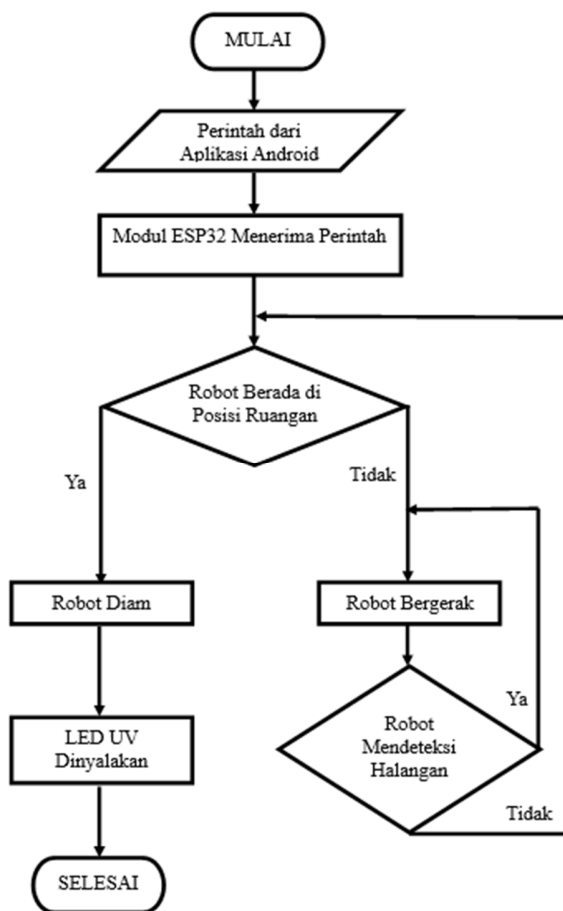
K. Internet of Things

Konsep IoT telah dibahas sejak awal tahun 1990an, meskipun belum ditemukan pada saat itu. Konsep ini dipopulerkan menggunakannya dalam riset pasar di *Auto-ID Center MIT*. Awalnya, jaringan ini didasarkan pada produk dengan chip *Radio Frequency ID* (RFID), sebuah teknologi yang telah ada selama setengah dekade. Penerapan pertama chip RFID adalah dalam lingkup inventaris, mulai dari memfasilitasi routing sampai loss prevention. Seiring waktu, teknologi dan metode lain dikembangkan untuk menghubungkan objek dalam jaringan, seperti *bar-code*, *Quick Response* (QR) *code*, *digital watermarking* dan *Near Field Communication* (NFC). Ketika biaya pengembangan teknologi *smart object* semakin menurun, jumlah aplikasi untuk objek jaringan meningkat. Penerapannya termasuk menghubungkan objek untuk pengawasan dan keamanan. IoT juga telah diterapkan untuk keperluan industri lainnya seperti transportasi, keamanan pangan dan manajemen dokumen[16].

Pada dasarnya *Internet of Things* (IoT) adalah suatu konsep yang menghubungkan antara internet dengan benda atau perangkat elektronika. *Internet of Things* menggunakan sistem komunikasi antar perangkat tanpa kabel. Dengan perkembangan teknologi saat ini, internet menjadi semakin mudah diakses. Hal ini mendukung efisiensi dan performa dari perangkat berbasis *Internet of Things* yang memungkinkan komunikasi jarak jauh. Ketika beberapa perangkat dapat melakukan komunikasi dengan internet tanpa bantuan manusia maka sistem tersebut dapat disebut sebagai IoT. Komunikasi antar alat tersebut dikenal dengan *Machine-Type-Communication* (MTC). Beberapa ide penerapan IoT yang populer saat ini seperti *smart house*, *smart health*, dan *smart city*.

L. ThingSpeak

ThingSpeak adalah salah satu platform *Internet of Things* yang menyediakan layanan secara gratis. *ThingSpeak* membolehkan pengguna mengumpulkan, menampilkan, dan menganalisa aliran data dalam *cloud*. Platform ini merekam data, memberikan *timestamp* dan memberikan *output* berupa grafik visual yang mudah dilihat pengguna[17]. *ThingSpeak* dapat diakses melalui alamat website *thingspeak.com* dan grafik *output* dapat dilihat melalui aplikasi android gratis. Pengguna dapat menggunakan akun *MathWorks* untuk masuk ke dalam website *ThingSpeak*. Ketika sudah membuat konfigurasi *channel*, pengguna dapat menuju ke *Private View* yang akan menunjukkan grafik data setiap *field*.



Gambar 2. Prinsip Kerja Sistem

M. Android

Android adalah sistem operasi berbasis Linux yang digunakan pada perangkat mobile seperti smartphone dan tablet. Android bersifat *open-source* yang artinya bebas digunakan oleh developer, desainer, dan pembuat perangkat secara gratis. Android Studio adalah *software IDE* yang dikembangkan untuk pengembangan aplikasi Android menggunakan bahasa pemrograman Java. Android Studio berfungsi sebagai *text editor* untuk menulis, mengubah, menyimpan kode. Selain itu, Android Studio menyediakan banyak fitur yang dapat mendukung produktivitas dalam pengembangan aplikasi[18].

N. MIT App Inventor

MIT App Inventor adalah platform online yang dapat digunakan untuk mengembangkan aplikasi android. Pengguna dapat membuat desain antar muka aplikasi dengan metode *drag-and-drop* dan menggunakan *block* untuk memprogram aplikasi. Antar muka MIT App Inventor terdiri dari 2 editor utama yaitu *design editor* dan *block editor*. *Design editor* atau *designer* menggunakan *interface drag-and-drop* untuk menempatkan elemen-elemen dari antar muka (UI) aplikasi yang sedang dikembangkan. *Block editor* atau *blocks* adalah tempat untuk mengatur logika dari aplikasi dengan menggunakan blok-blok berwarna yang dapat tersambung seperti puzzle untuk mendeskripsikan program aplikasi.

II. METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian dilakukan di lingkungan Program Studi Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi dan uji coba dilakukan di Lab Biomolekuler Universitas Sam Ratulangi. Waktu pelaksanaan dimulai dari Maret 2021 sampai Juni 2023.

B. Perangkat Keras

1) Robot Disinfektan

Robot desinfektan yang digunakan dalam penelitian ini adalah robot beroda berukuran sedang dengan manuver *omni-wheel*, seperti terlihat pada Gambar 1. Robot ini dilengkapi dengan LED UVC. Sinar UVC sering digunakan untuk mendesinfeksi ruangan di layanan kesehatan. Selain itu, sinar UVC sudah banyak digunakan sebagai disinfektan untuk peralatan medis seperti peralatan yang terpapar dengan virus COVID-19.

2) Perancangan Komunikasi Arduino Mega dan ESP32

Arduino Mega dan ESP32 dapat berkomunikasi melalui komunikasi serial. Baik dari ESP32 ke Arduino maupun dari Arduino ke ESP32. ESP32 akan disambungkan ke Arduino menggunakan kabel *jumper*. GND ESP32 ke GND Arduino menggunakan kabel hitam. 3V3 ESP32 disambungkan ke 3.3V Arduino menggunakan kabel merah. Pin TX dan RX disambungkan menggunakan pin digital masing-masing pin TX disambungkan ke pin RX.

3) Perancangan LCD

LCD yang digunakan sudah termasuk dengan modul I2C sehingga memudahkan wiring dengan Arduino. GND LCD disambungkan ke GND Arduino menggunakan kabel hitam. VCC LCD disambungkan ke 5V Arduino menggunakan kabel merah. Pin SCL dan pin SDA bisa disambungkan ke pin analog Arduino.

4) Perancangan Aktuator

Aktuator yang digunakan yaitu motor DC PG-45. Motor ini membutuhkan *power supply* seperti baterai agar dapat bergerak. Motor PG-45 membutuhkan *driver motor* yang disambungkan ke Arduino agar dapat diprogram melalui Arduino IDE. Motor PG-45 disambungkan ke *driver motor*, *driver motor* disambungkan ke *power supply* dan Arduino. Untuk memprogram motor DC, digunakan pin digital PWM pada Arduino.

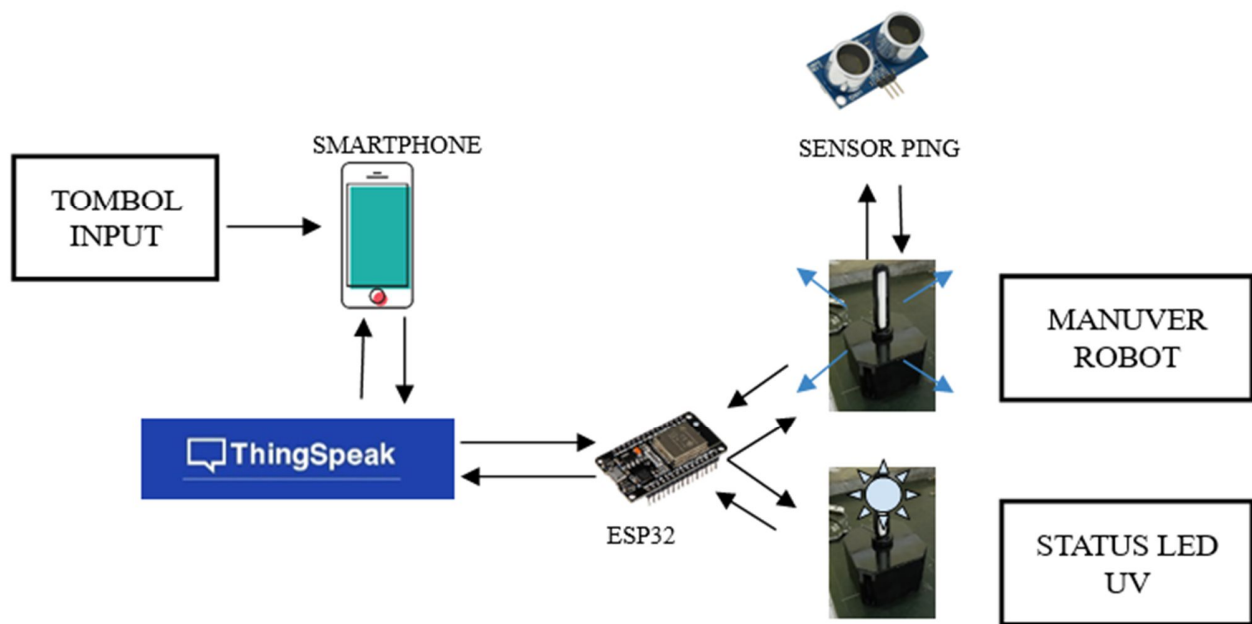
5) Perancangan Sensor PING

Sensor PING terdiri dari 3 pin, GND, 5V, dan SIG yang digunakan sebagai input/output. Pin GND PING disambungkan ke GND Arduino menggunakan kabel hitam. Pin 5V PING disambungkan ke 5V Arduino menggunakan kabel merah. Pin SIG dari PING disambungkan ke pin digital Arduino.

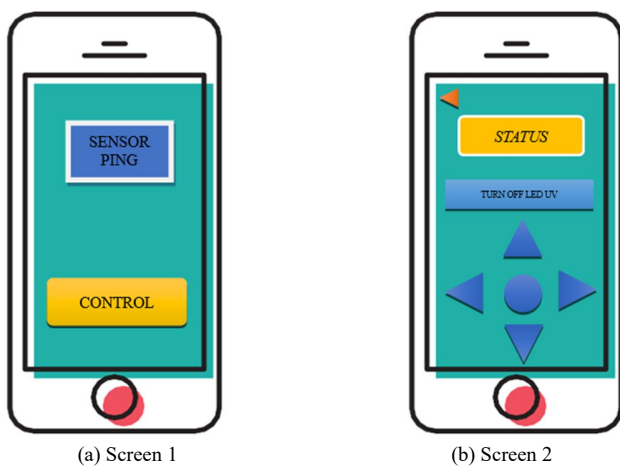
C. Desain Sistem

1) Prinsip Kerja Sistem

Prinsip kerja sistem dapat dilihat pada Gambar 2. Sistem pengendalian robot disinfektan terdiri dari 2 bagian yaitu aplikasi android dan modul Wi-Fi. Aplikasi android diakses melalui smartphone operator. Sedangkan modul Wi-Fi yang



Gambar 3. Desain Proses



(a) Screen 1

(b) Screen 2

Gambar 4. Desain Interface

Kemudian perintah tersebut akan ditransmisikan ke ESP32 dan ESP32 akan memberikan informasi kembali ke aplikasi untuk menunjukkan status robot. Operator dapat melakukan 2 tugas utama yaitu memerintahkan manuver robot dan pemasangan atau pemadaman LED UV. Sensor ping akan memberikan informasi jarak robot dengan lingkungan sekitarnya terutama ketika robot sedang berpindah tempat dan hasil pembacaan sensor akan ditampilkan di *smartphone*.

3) Desain Interface

Desain *User Interface* terdiri dari 2 *screen*. Gambar 4.a, *screen* pertama, menampilkan nilai sensor PING. Gambar 4.b, *screen* kedua menampilkan keadaan atau status robot dan menampilkan tombol-tombol pengaturan lampu LED dan manuver robot.

III. HASIL DAN PEMBAHASAN

digunakan yaitu ESP32 sudah dipasang pada robot disinfektan.

Cara kerja sistem pengendali robot disinfektan dimulai dari aplikasi pengendali berbasis android. Dalam *flowchart* digambarkan bahwa ketika operator memasukkan sebuah perintah melalui aplikasi android, maka modul ESP32 akan menerima perintah tersebut kemudian menjalankannya.

Apabila robot berada pada posisi yang dikehendaki operator, maka robot tidak akan digerakkan dan hanya diam. Selanjutnya, operator dapat menyalakan LED UV melalui aplikasi pengendali. Namun apabila robot belum berada di posisi, operator dapat menjalankan manuver robot agar sampai pada posisi. Sementara itu, jika robot mendeteksi halangan maka robot akan menghindar.

2) Desain Proses

Desain proses seperti terlihat pada Gambar 3. Pertama-tama, operator memasukkan perintah dengan menekan tombol *input* pada aplikasi pengendali di *smartphone* android.

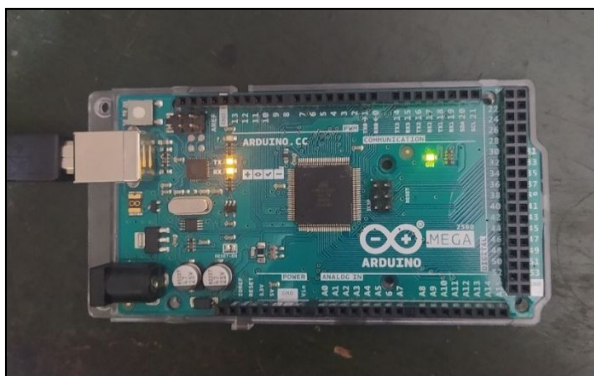
A. Pengujian Perangkat Keras

1) Pengujian Mikrokontroler Arduino Mega

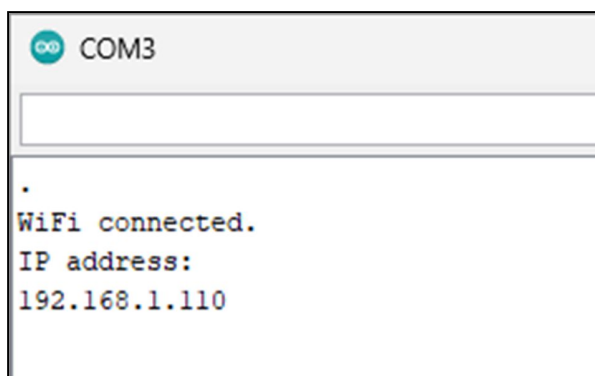
Pengujian ini dilakukan dengan mengunggah program BLINK ke *board* mikrokontroler Arduino Mega. LED yang digunakan adalah LED_BUILTIN yang tersambung pada pin 13 di *board* Arduino Mega. Hasil dari pengujian ini adalah LED 'ON' pada *board* Arduino Mega akan menyala selama 1 detik lalu redup 1 detik atau berkedap-kedip, seperti terlihat pada Gambar 5. Arduino Mega dapat menjalankan program berarti *chip* dalam Arduino Mega berfungsi dengan baik.

2) Pengujian ESP32

Pengujian pada tahap ini dilakukan dengan koneksi ESP32 ke jaringan WiFi. Pemrograman ESP32 menggunakan Arduino IDE dengan tambahan *board manager URL* pada opsi. Dengan ini *board manager* ESP32 Arduino akan ditambahkan pada *Tools* Arduino IDE. Gambar 6 menunjukkan hasil koneksi ESP32 ke WiFi berupa *IP Address*.



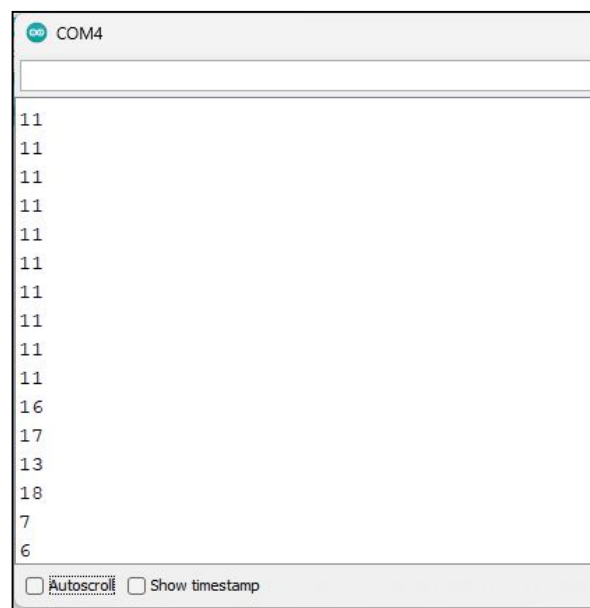
Gambar 5. LED ON pada Arduino Mega



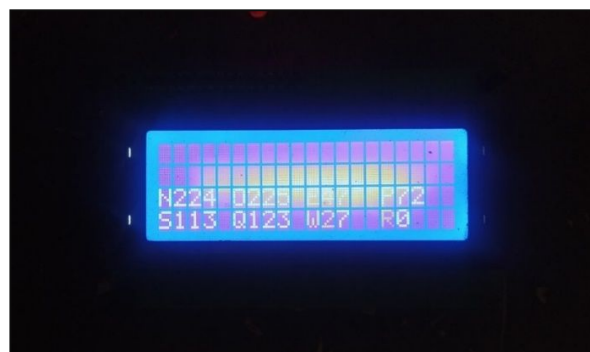
Gambar 6. Hasil koneksi ESP32 ke WiFi



Gambar 7. Tampilan Hasil testLCD



Gambar 8. Hasil Baca Sensor PING))



Gambar 9. Tampilan LCD pada robot

3) Pengujian LCD

Pengujian LCD dilakukan dengan menampilkan karakter pada LCD yang sudah dipasangkan pada robot disinfektan (Gambar 7). Penempatan karakter ada pada baris ke 4, kolom ke 2. Hasil dari program ini yaitu layar LCD diisi karakter pada kolom ke 2, dari baris ke 4 sampai baris ke 15. Kata yang ditampilkan pada layar adalah "HELLO THERE" berjumlah 11 karakter.

4) Pengujian Sensor PING

Sensor jarak yang digunakan pada robot disinfektan yaitu sensor PING. Dalam pengujian ini, sensor PING disambungkan ke *board* mikrokontroler dan mengambil daya dari *port* komputer. Hasil baca sensor untuk jarak objek yang berdiam yaitu 11 cm, kemudian bervariasi ketika objek dipindahkan semakin mendekati sensor. Gambar 8 menampilkan hasil pada *Serial Monitor*.

5) Pengujian Motor Penggerak pada Robot Disinfektan

Robot disinfektan menggunakan 2 roda yang digerakkan dengan motor PG45. Motor diprogram pada interval 0-255 *pulse-width modulation* (PWM). Arah putaran motor dibagi 2, yaitu, L-PWM dan R-PWM. Pengujian ini dilakukan dengan

melihat arah putaran roda pada robot dengan memberikan nilai PWM pada satu satu pin pada roda. Hasil dari pengujian ini adalah logika gerakan robot yang dilihat seperti pada Tabel I.

6) Pengujian Sensor PING pada Robot Disinfektan

Rancangan robot memiliki 8 sisi yang dipasangkan sensor tiap sisi. Pada robot dipasangkan Liquid Crystal Display berukuran 20x4 *cell*. Data dari sensor ping diproses satu per satu setiap sensor. Data yang diterima dikonversi ke dalam satuan sentimeter. Hasil bacaan sensor ping kemudian ditampilkan pada layar LCD.

7) Pengujian LED UV

Pengujian LED UV dilakukan dengan menggunggah program ke robot disinfektan. LED UV disambungkan ke RELAY dalam robot. Logika LOW menghasilkan LED OFF pada robot, dapat dilihat pada Gambar 10. Logika HIGH menghasilkan LED ON pada robot, dapat dilihat pada Gambar 11. Hasil dari pengujian ini menunjukkan lampu LED UVC yang dipasang pada robot dapat berfungsi dengan baik.

B. Pengujian Perangkat Lunak

Aplikasi Android yang dikembangkan terdiri dari 2 *Screen*.



Gambar 10. Robot UV dengan LED OFF



Gambar 11. Robot UV dengan LED ON

Tabel I
Logika Maneuver 2 Roda

MANEUVER	LPWM	RPWM	LPWM	RPWM
	RODA 1	RODA 1	RODA 2	RODA 2
Maju	PWM	0	PWM	0
Mundur	0	PWM	0	PWM
Belok Kanan	PWM	0	0	PWM
Belok Kiri	0	PWM	PWM	0
Berhenti	0	0	0	0

Screen 1, seperti pada Gambar 12.a, menampilkan layout PING berisi nama setiap PING dan angka dibawahnya, tombol ON dan OFF untuk lampu UVC, dan tombol MOVE ROBOT. Screen 2, seperti pada Gambar 12.b, menampilkan tombol 4 arah dan tombol O, tombol ON dan OFF untuk lampu UVC, dan tombol BACK. Hasil dari pengujian ini dilihat dari keberhasilan tampilan aplikasi pada smartphone.

C. Pengujian Sistem

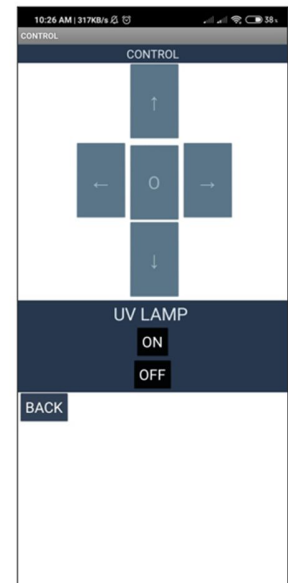
1) Pengujian Komunikasi ESP32 dan Arduino Mega

Pengiriman data antara ESP32 dan Arduino Mega menggunakan komunikasi serial. Untuk mode otomatis atau wall-following, Arduino mengirim data hasil bacaan jarak dari sensor PING dan diterima oleh ESP32.

Pada Gambar 13, Serial Monitor COM4 adalah serial monitor dari Arduino Mega yang memproses data dari PING N yaitu 14 sentimeter. Data tersebut diterima oleh ESP32 melalui komunikasi serial, seperti pada Gambar 14 dimana nilai N adalah 14.



(a) Screen 1



(b) Screen 2

Gambar 12. Screenshot Aplikasi Android

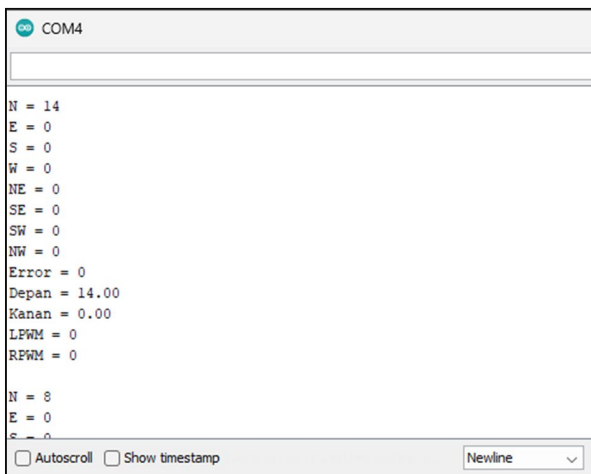
2) Pengujian Transfer Data PING dari Robot ke Aplikasi Android

Pada pengujian ini, digunakan sensor ping pada robot disinfektan sebagai data yang akan ditunjukkan pada aplikasi android. Untuk setiap ping digunakan konfigurasi *Software Serial* masing-masing, dimana pin yang digunakan sebagai *Receiver* (Rx) dan *Transmitter* (Tx).

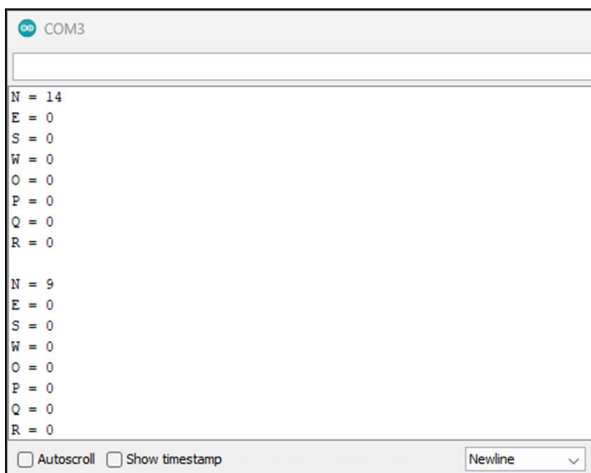
Komunikasi “serial” dan “comm” dimulai dengan *baud rate* 9600. Koneksi ke *ThingSpeak* dimulai sebagai *client*. ESP32 melakukan koneksi ke WiFi menggunakan *ssid* dan *password* yang sudah ditentukan. Data yang masuk pada mikrokontroler ESP32 akan dikonversi ke integer kemudian dikirim ke *field* pada *channel* “PING SENSORS” di *ThingSpeak*. Data hasil bacaan dari Arduino yang sudah diterima ESP32 ditampilkan ke *platform ThingSpeak*. Pada *platform ThingSpeak*, *field* 1 sampai *field* 8 digunakan untuk menampung hasil bacaan sensor. Hasil bacaan data 8 sensor PING diterima oleh *ThingSpeak* dan ditampilkan dalam grafik pada setiap *field* untuk setiap sensor PING. Gambar 15 menampilkan hasil sensor ping N. Gambar 16 menampilkan hasil sensor ping NE dan PING E. Gambar 17 menampilkan hasil sensor ping SE dan PING S. Gambar 18 menampilkan hasil sensor ping SW dan PING W. Gambar 19 menampilkan hasil sensor ping NW.

Aplikasi Android mengambil data *ThingSpeak* menggunakan *json* dan menampilkannya sesuai dengan nomor *field* dan singkatan untuk masing-masing. Hasil data baca sensor PING pada robot dapat dilihat di aplikasi android. Data dari sensor PING disimpan di *ThingSpeak* dan dapat di-export dalam format file.csv. Gambar 20 menunjukkan hasil bacaan sensor PING pada aplikasi *controller*. Persamaan (1) digunakan untuk menghitung hasil dari data yang ditampilkan dalam format persen.

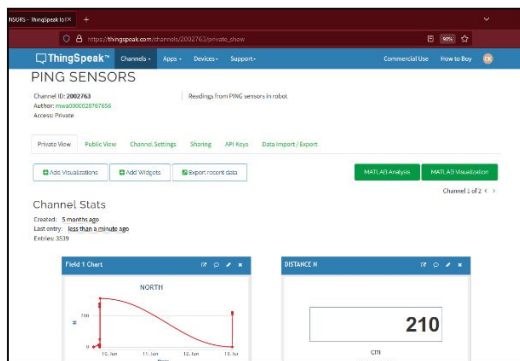
$$\text{Persentase} = \frac{\text{Data}}{\text{Jumlah Data}} \times 100 \% \quad (1)$$



Gambar 13. Hasil bacaan sensor PING N di Arduino

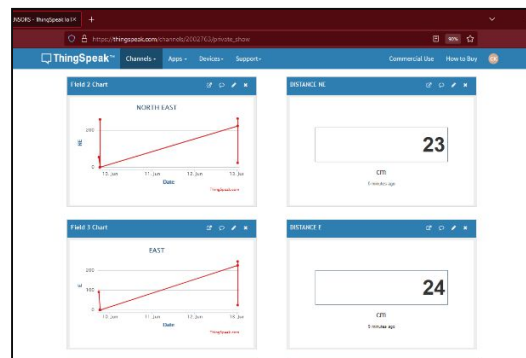


Gambar 14. Hasil bacaan sensor PING N di ESP32

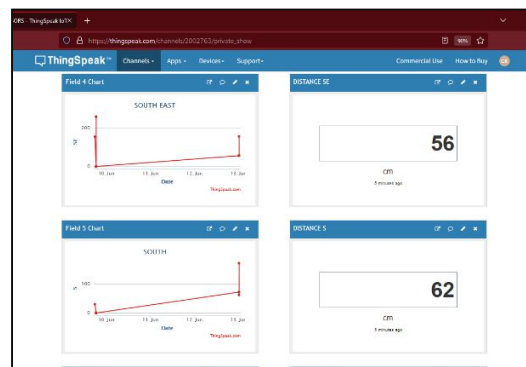


Gambar 15. Tampilan Web ThingSpeak untuk PING SENSORS

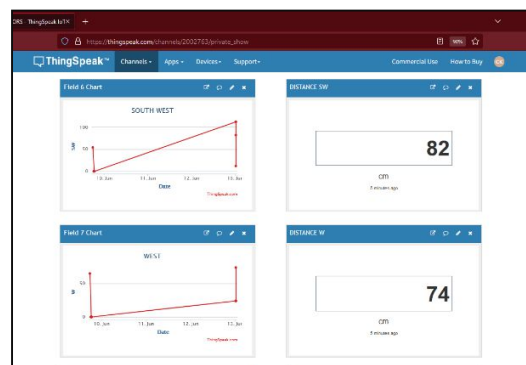
Dalam waktu 45 menit, data yang disimpan *ThingSpeak* berjumlah 148 *data entry*. Dari data tersebut waktu yang dibutuhkan untuk merekam data paling cepat ada 15 detik dan paling lama adalah 2 menit 42 detik. Gambar 21 memperlihatkan grafik interval waktu ketika data dikirimkan ke *ThingSpeak* dari ESP32. Dengan memperhatikan batasan fitur *ThingSpeak* 15 detik, maka dapat disimpulkan bahwa *time delay* paling lama dalam pengujian selama 45 menit adalah 2 menit 27 detik.



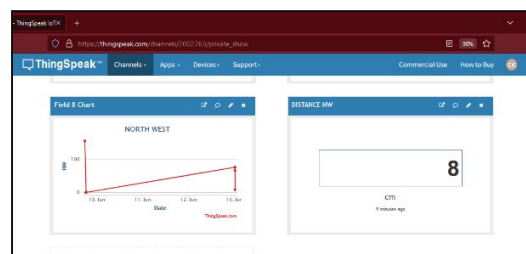
Gambar 16. Tampilan Web ThingSpeak untuk PING SENSORS



Gambar 17. Tampilan Web ThingSpeak untuk PING SENSORS



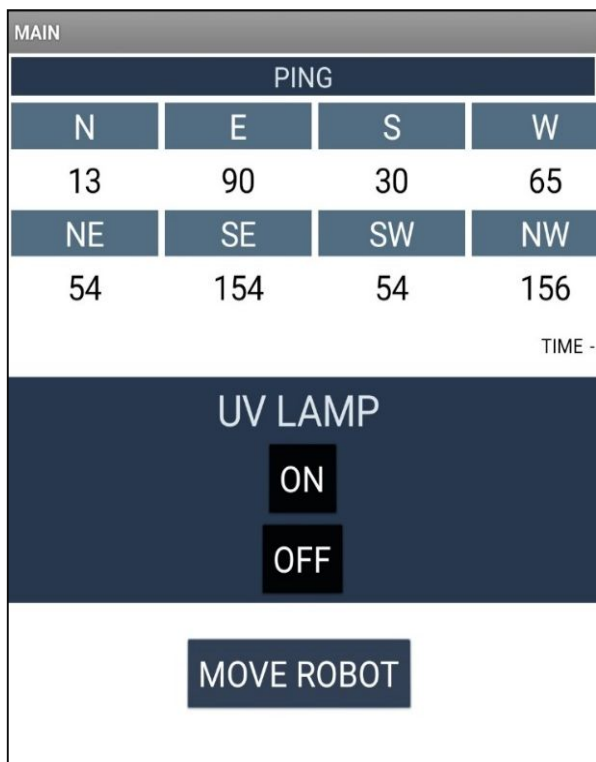
Gambar 18. Tampilan Web ThingSpeak untuk PING SENSORS



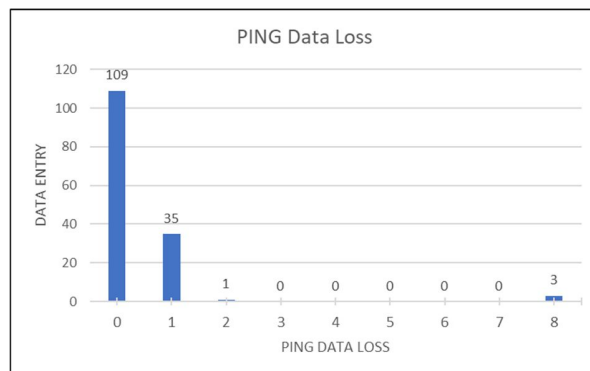
Gambar 19. Tampilan Web ThingSpeak untuk PING SENSORS

Dalam pengujian selama 45 menit, data yang masuk ada 148 *data entry* berarti total data yang masuk adalah 1184 data sensor ping. Gambar 22 memperlihatkan grafik jumlah data loss dalam setiap 148 *data entry*.

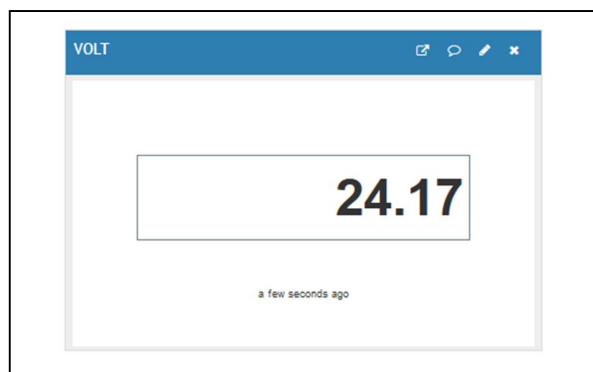
Gambar 23 menunjukkan bahwa dalam 107 *data entry*, data yang berhasil dikirim tanpa *data loss* adalah 84 *data entry*.



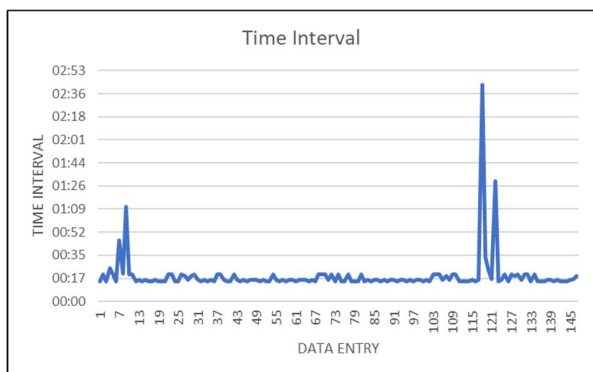
Gambar 20. Tampilan Aplikasi



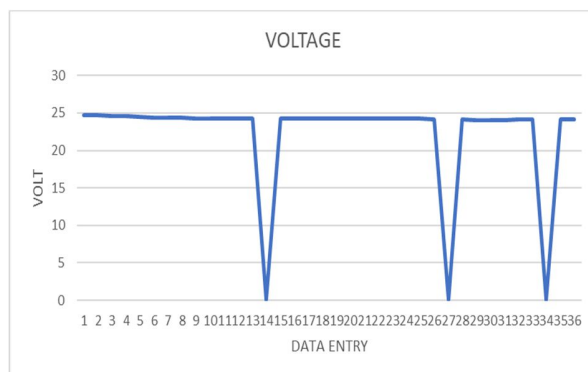
Gambar 19. Diagram PING Data Loss Pengujian 45 Menit



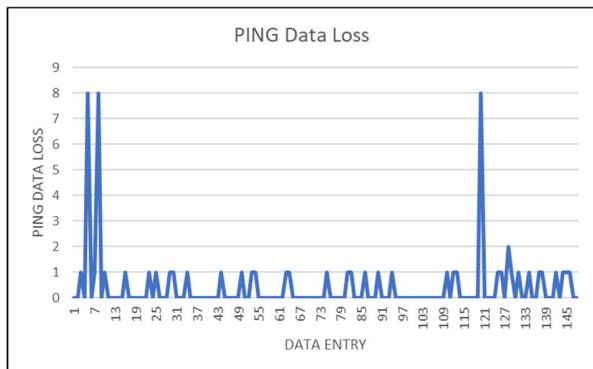
Gambar 23. Halaman ThingSpeak



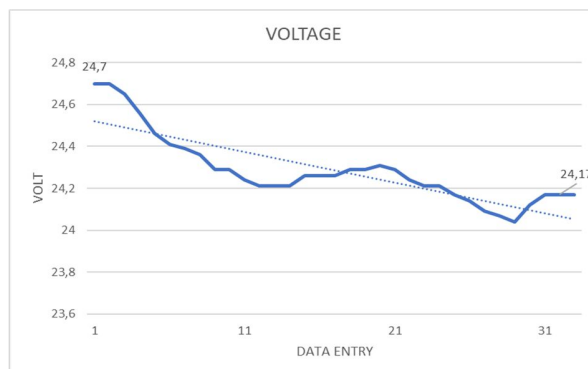
Gambar 21. Grafik Time Interval Pengujian 45 Menit



Gambar 24. Grafik Data Tegangan dari ThingSpeak



Gambar 22. Grafik PING Data Loss Pengujian 45 Menit



Gambar 25. Grafik Data Tegangan

Data loss sebanyak 1 data dalam 1 entry terjadi 35 kali. Data loss sebanyak 2 data dalam 1 entry terjadi 1 kali. Data loss sebanyak 8 data dalam 1 entry terjadi 3 kali. Total data loss adalah 61 dalam 45 menit.

$$\frac{\text{Data Loss}}{\text{Jumlah Data Diterima}} \times 100 \% = \frac{61}{1184} \times 100 \% = 5,15 \%$$

Tabel II
Tabel Data Lampu UV dari Aplikasi Android ke ThingSpeak

Button Used	Data Sent	Time Delay	OUTPUT
OFF	0	00:45	SUCCESS
ON			FAILED
ON			FAILED
ON	1	00:09	SUCCESS
OFF	0	00:03	SUCCESS
ON	1	00:03	SUCCESS
OFF	0	00:41	SUCCESS
ON			FAILED
OFF	0	01:29	SUCCESS
OFF	0	00:01	SUCCESS
OFF	0	00:22	SUCCESS
OFF			FAILED
ON			FAILED
ON	1	01:05	SUCCESS
OFF	0	01:27	SUCCESS
OFF	0	00:01	SUCCESS
ON	1	00:16	SUCCESS
OFF	0	00:47	SUCCESS
OFF	0	00:44	SUCCESS
ON	1	00:13	SUCCESS
OFF	0	01:31	SUCCESS
ON	1	00:21	SUCCESS
OFF	0	00:17	SUCCESS
OFF	0	00:36	SUCCESS
ON			FAILED
OFF	0	00:06	SUCCESS
ON	1	00:01	SUCCESS
OFF	0	00:09	SUCCESS
OFF	0	00:24	SUCCESS
ON	1	00:10	SUCCESS

$$\frac{\text{Data PING}}{\text{Jumlah Data Diterima}} \times 100 \% = \frac{1123}{1184} \times 100 \% = 94,85 \%$$

Berdasarkan perhitungan, dapat disimpulkan bahwa data yang diterima *ThingSpeak* memiliki persentase keberhasilan yaitu 94,85% dalam waktu pengujian 45 menit.

3) Pengujian Transfer Data VOLT dari Robot ke Aplikasi Android

Pada pengujian ini, digunakan sensor tegangan DC yang terpasang di robot. Program yang digunakan adalah program transfer data dari robot ke aplikasi android dengan modifikasi penambahan program untuk perhitungan data tegangan. Robot dipasang dan beroperasi melakukan pergerakan menggunakan motor DC dan sensor PING. Gambar 24 menunjukkan

Tabel III
Tabel Data Maneuver dari Aplikasi Android ke ThingSpeak

Button Used	Data Sent	Time Delay	OUTPUT
FORWARD	1	00:04	SUCCESS
BREAK	0	00:06	SUCCESS
RIGHT			FAILED
RIGHT	4	00:06	SUCCESS
LEFT	3	00:01	SUCCESS
BREAK	0	00:08	SUCCESS
LEFT	3	00:20	SUCCESS
FORWARD	1	00:32	SUCCESS
BREAK	0	00:20	SUCCESS
FORWARD	1	00:21	SUCCESS
BREAK	0	00:03	SUCCESS
FORWARD	1	00:03	SUCCESS
BREAK	0	00:05	SUCCESS
REVERSE			FAILED
REVERSE	2	00:28	SUCCESS
LEFT	3	00:05	SUCCESS
BREAK	0	00:07	SUCCESS
LEFT	3	00:12	SUCCESS
RIGHT	4	00:24	SUCCESS
FORWARD	1	00:03	SUCCESS
BREAK	0	00:03	SUCCESS
LEFT	3	00:12	SUCCESS
BREAK	0	00:15	SUCCESS
FORWARD	1	00:23	SUCCESS
LEFT	3	00:10	SUCCESS
RIGHT	4	00:18	SUCCESS
BREAK	0	00:06	SUCCESS
BREAK	0	00:02	SUCCESS
LEFT	3	00:00	SUCCESS
RIGHT			FAILED

tampilan *field* dalam halaman web *ThingSpeak* untuk data tegangan yang masuk.

Pada pengujian ini, robot bergerak mengelilingi ruangan di Lt.3 Lab Biomolekuler UNSRAT. Waktu yang dibutuhkan robot untuk mengelilingi ruangan adalah 9 menit 16 detik. Dapat dilihat pada Gambar 25, grafik menampilkan hasil baca data tegangan robot disinfeksi dari *ThingSpeak* selama 9 menit 16 detik. Data “VOLT” pada angka 0 memiliki arti yaitu data tidak diterima oleh *ThingSpeak* atau data hilang (*data loss*).

$$\frac{\text{Data Tegangan}}{\text{Jumlah Data Diterima}} \times 100 \% = \frac{33}{36} \times 100 \% = 91,67 \%$$

$$\frac{\text{Data Loss}}{\text{Jumlah Data Diterima}} \times 100 \% = \frac{3}{36} \times 100 \% = 8,33 \%$$

Berdasarkan perhitungan persentase, hasil dari pengiriman data tegangan dari robot disinfektan ke *ThingSpeak* selama 9 menit 16 detik yaitu data diterima sebanyak 91,67% dan *data loss* sebanyak 8,33%.

Gambar 22 menunjukkan grafik data yang diterima dengan menghilangkan *data loss*. Dari gambar ini dapat dilihat adanya penurunan tegangan pada robot. Berdasarkan data tersebut, setelah waktu 9 menit 16 detik tegangan pada baterai yang ada pada robot menurun dari 24,7 Volt ke 24,17 Volt.

4) Pengujian Transfer Data Aplikasi Android ke Robot

Pengujian dilakukan dengan menggunakan aplikasi android pada menu *control* yang terdiri dari manuver maju, mundur, belok kiri, belok kanan, diam, serta tombol untuk menyalakan lampu UV dan mematikan lampu UV.

$$\frac{\text{Data Button ON OFF}}{\text{Jumlah Button Ditekan}} \times 100 \% = \frac{24}{30} \times 100 \% = 80 \%$$

Dalam pengujian lampu UV, tombol ON dan OFF ditekan sebanyak 30 kali. Tabel II menampilkan data keberhasilan data yang masuk. Waktu yang dihitung adalah ketika tombol ditekan, data dapat dilihat pada halaman *ThingSpeak*, dan lampu UV menyala atau padam sesuai dengan perintah dari button. Berdasarkan data dari tabel, dapat disimpulkan bahwa tingkat keberhasilan pengiriman data lampu UV memiliki tingkat keberhasilan 80%.

$$\frac{\text{Data Button Maneuver}}{\text{Jumlah Button Ditekan}} \times 100 \% = \frac{27}{30} \times 100 \% = 90 \%$$

Data pada Tabel III menunjukkan hasil dari pengoperasian robot disinfektan dengan 30 kali percobaan menggunakan tombol-tombol pada aplikasi android. Berdasarkan data dari tabel, dapat disimpulkan bahwa tingkat keberhasilan pengiriman data manuever memiliki tingkat keberhasilan 90%. Maka dapat disimpulkan bahwa persentase keberhasilan pengiriman data dari aplikasi android ke robot disinfektan memiliki rata-rata 85%. Dengan melihat data pada tabel II dan tabel III, *time delay* pengiriman data paling lama adalah 01:31 (1 menit 32 detik).

IV. PENUTUP

A. Kesimpulan

Kesimpulan yang dapat diambil pada penelitian ini adalah keberhasilan pengiriman data sensor ping dari robot disinfektan ke aplikasi android adalah 94,85 % dengan *time delay* paling lama adalah 2 menit 27 detik. Ketika robot bergerak mengelilingi ruangan di Lt.3 Lab Biomolekuler UNSRAT terjadi penurunan tegangan baterai dari 24,7 Volt ke 24,17 Volt. Pengiriman data tegangan dari robot disinfektan ke *ThingSpeak* dalam waktu 9 menit 16 detik yaitu data diterima sebanyak 91,67% dan *data loss* sebanyak 8,33%.

Persentase keberhasilan pengiriman data dari aplikasi android ke robot disinfektan adalah 85% dengan *time delay* pengiriman data paling lama adalah 01 menit 31 detik. Jarak antara robot disinfektan dan sumber koneksi internet sangat mempengaruhi *time delay* pada pengiriman data ke *cloud*.

B. Saran

Saran dalam penelitian ini tentunya masih memiliki banyak kekurangan sehingga kedepannya ketika penelitian ini dikembangkan diharapkan akan ditambahkan fitur-fitur yang dapat mempermudah operator dalam menggunakan aplikasi pengendali jarak jauh. Fitur-fitur seperti penjadwalan sesi robot disinfeksi, kontrol aktuator lebih cepat, dan penggunaan platform *Internet of Things* lainnya.

V. KUTIPAN

- [1] Rajesh Kowti *et al.*, "UV rays induced DNA damage: Protection by polyphenols enriched extract of mint leaves," *World Journal of Advanced Research and Reviews*, vol. 8, no. 2, 2020, doi: 10.30574/wjarr.2020.8.2.0405.
- [2] B. Casini *et al.*, "Evaluation of an ultraviolet C (UVC) light-emitting device for disinfection of high touch surfaces in hospital critical areas," *Int J Environ Res Public Health*, vol. 16, no. 19, 2019, doi: 10.3390/ijerph16193572.
- [3] M. Raeiszadeh and B. Adeli, "A Critical Review on Ultraviolet Disinfection Systems against COVID-19 Outbreak: Applicability, Validation, and Safety Considerations," *ACS Photonics*, vol. 7, no. 11. American Chemical Society, pp. 2941–2951, Nov. 18, 2020. doi: 10.1021/acsp Photonics.0c01245.
- [4] S. Villamil, C. Hernández, and G. Tarazona, "An overview of internet of things," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, 2020, doi: 10.12928/TELKOMNIKA.v18i5.15911.
- [5] M. Babiuch, P. Foltyniek, and P. Smutny, "Using the ESP32 microcontroller for data processing," in *Proceedings of the 2019 20th International Carpathian Control Conference, ICCCC 2019*, 2019. doi: 10.1109/CarpathianCC.2019.8765944.
- [6] J. Y. Lin, H. L. Tsai, and W. H. Lyu, "An integrated wireless multi-sensor system for monitoring the water quality of aquaculture," *Sensors*, vol. 21, no. 24, Dec. 2021, doi: 10.3390/s21248179.
- [7] A. H. Kelechi *et al.*, "Design of a low-cost air quality monitoring system using arduino and thingspeak," *Computers, Materials and Continua*, vol. 70, no. 1, 2021, doi: 10.32604/cmc.2022.019431.
- [8] M. Forcan and M. Maksimović, "Cloud-Fog-based approach for Smart Grid monitoring," *Simul Model Pract Theory*, vol. 101, 2020, doi: 10.1016/j.simpat.2019.101988.
- [9] R. F. Al-Mutawa and F. A. Eassa, "A smart home system based on internet of things," *International Journal of Advanced Computer Science and Applications*, no. 2, 2020, doi: 10.14569/ijacsa.2020.0110234.
- [10] S. F. Barrett, "Arduino I: Getting Started," in *Synthesis Lectures on Digital Circuits and Systems*, vol. 15, no. 1, 2020. doi: 10.2200/S01001ED1V01Y202003DCS058.
- [11] Espressif System, "ESP32 Series Datasheet," *Espressif Systems*, 2019.
- [12] Parallax, "PING™ Ultrasonic Distance Sensor," *Parallax*, no. 916, 2013.

- [13] M. C. Ramon, “Arduino IDE and Wiring Language,” in *Intel® Galileo and Intel® Galileo Gen 2*, 2014. doi: 10.1007/978-1-4302-6838-3_3.
- [14] E. Setiawan, “KBBI - Kamus Besar Bahasa Indonesia,” *kamus besar bahasa indonesia*, 2019.
- [15] WHO, “WHO COVID-19,” World Health Organization.
- [16] K. Schwab, *The Fourth Industrial Revolution*. 2016. [Online]. Available: www.weforum.org
- [17] M. A. G. Maureira, D. Oldenhof, and L. Teernstra, “ThingSpeak-an API and Web Service for the Internet of Things.”
- [18] I. Krajci and D. Cummings, “History and Evolution of the Android OS,” in *Android on x86*, 2013. doi: 10.1007/978-1-4302-6131-5_1.



Christine J. Kambey, lahir di Manado pada tanggal 17 Juni 1999. Penulis adalah anak pertama dari pasangan Jusuf I. Kambey dan Grace Kembuan, penulis memiliki saudara kandung adik David Z. Kambey. Penulis menempuh pendidikan secara berturut-turut di TK Dorkas Tomohon (2005), SD GMIM 6 TOMOHON (2005-2011), SMP NEGERI 1 TOMOHON (2011-2014), SMA

NEGERI 1 TOMOHON (2014-2017).

Pada tahun 2017, penulis melanjutkan studi di Program Studi S1 Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sam Ratulangi Manado. Selama studi penulis turut serta dalam Kontes Robot ABU Indonesia (KRAI) yang diselenggarakan di Universitas Tarumanegara tahun 2018, Universitas Muhammadiyah Yogyakarta tahun 2018, dan Universitas Mataram tahun 2019. Selama perkuliahan penulis aktif sebagai pengurus organisasi UKM EDUKASI ROBOTIKA (EURO) tahun 2019-2021, anggota Himpunan Mahasiswa Elektro (HME), dan pengurus komunitas Unsrat IT Community (UNITY) tahun 2020-2021.