Jurnal Teknik Elektro, Vol. 15, No. 1, Maret 2022, 13-19
ISSN 1411-870X

Perancangan Sistem Kontrol Robot Desinfeksi Ruangan Dengan Lampu UV-C

Jeremy Winston[1], Indar Sugiarto[2], Handy Wicaksono[3]
Program Studi Teknik Elektro, Universitas Kristen Petra
Jl.Siwalankerto 121-131, Surabaya 60236, Indonesia
Email: c11170013@john.petra.ac.id[1], indi@petra.ac.id[2], handy@petra.ac.id[3]

***Abstrak* — Pandemi Covid-19, yang disebabkan oleh coronavirus jenis baru bernama SARS-CoV-2, telah memakan banyak sekali korban jiwa dan sampai jurnal ini ditulis virus ini terus menyebar ke seluruh penjuru dunia. Salah satu langkah pencegahan penularan Covid-19 adalah dengan melakukan desinfeksi ruangan secara rutin.**

Oleh karena itu penelitian ini bertujuan untuk membuat sebuah robot yang dapat mendesinfeksi ruangan dengan lampu UV-C secara semi-otomatis menggunakan metode magnetic *line following*. Lampu UV-C digunakan sebagai agen desinfeksi karena tidak mengandung bahan kimia dan metode magnetic *line following* digunakan karena dapat digunakan dalam kondisi lingkungan yang kotor dan gelap sehingga robot ini cocok digunakan di berbagai macam kondisi.

Kata Kunci — robot desinfeksi ruangan, *line following*, raspberry pi 4, ROS

I. PENDAHULUAN

Wabah SARS-CoV-2 atau yang dikenal dengan COVID-19, yang muncul pada akhir tahun 2019 lalu telah ditetapkan oleh World Health Organization (WHO) sebagai sebuah pandemi pada tanggal 11 Maret 2020. Setelah setahun lebih pandemi ini dimulai, jumlah korban jiwa yang disebabkan COVID-19 di Indonesia masih terus meningkat. Mengingat bahwa sampai saat ini metode perawatan yang tersedia untuk menangani pasien yang positif COVID-19 masih dalam tahap penelitian, maka langkah-langkah pencegahan penularan virus COVID-19 menjadi prioritas utama terutama di tempat umum seperti di perkantoran, rumah sakit, mall, dan sebagainya.

Centers for Disease Control and Prevention (CDC) telah menyarankan beberapa metode untuk sterilisasi dan disinfeksi virus dan bakteri salah satunya dengan menggunakan radiasi sinar UV-C [1].Kelebihan radiasi sinar UV-C di antara metode desinfeksi lain adalah sinar UV-C bebas dari bahan kimia, tidak beracun dan natural. Selain itu, sinar UV-C tidak dapat merusak dokumen kertas maupun alat elektronik.

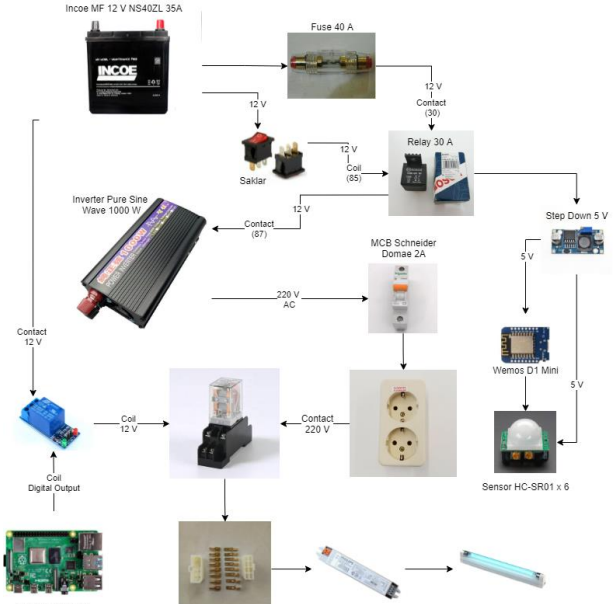
Namun paparan sinar UV-C secara langsung dapat menyebabkan masalah kesehatan serius bagi manusia terutama bagi kesehatan kulit dan mata, Maka dari itu dalam penelitian ini dikembangkan sebuah robot desinfeksi UV-C yang dapat bergerak secara otomatis. Dengan robot ini proses desinfeksi akan menjadi lebih aman dan mudah karena tidak perlu melibatkan manusia secara langsung dan proses desinfeksi dapat dilakukan di tempat umum saat malam hari ketika tidak ada orang.

II. PERENCANAAN DAN IMPLEMENTASI

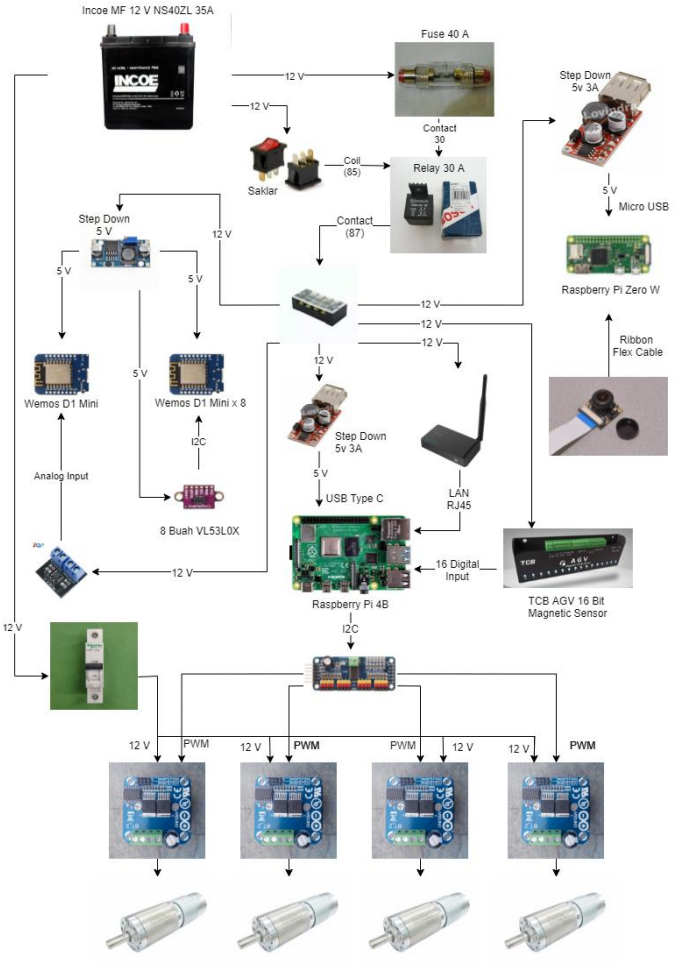
Robot desinfeksi ini dikerjakan oleh dua mahasiswa yaitu penulis dan Rio Alfandy. Penulis akan fokus pada pembuatan modul desinfeksi UV-C, sedangkan Rio akan fokus pada pembuatan *base* atau penggerak robot. Penggerak robot memiliki bentuk balok dengan ukuran panjang, lebar, dan tinggi yaitu 50cm x 50cm x 35cm, sedangkan modul desinfeksi UV-C berbentuk prisma segi enam dengan ukuran panjang, lebar, dan tinggi yaitu 15cm x 15cm x 135cm. Penggerak robot memiliki 4 buah roda yang dihubungkan dengan 4 buah motor yang dapat membuat robot bergerak. Robot ini didesain untuk bersifat modular sehingga penggerak robot ini dapat dipasang modul lain selain modul UV-C.

Dalam robot ini terdapat sebuah komputer Raspberry Pi 4B sebagai otak dari robot. Di dalam komputer ini dibangun suatu perangkat lunak berbasis *framework Robot Operating system* (ROS) Noetic. Robot ini memiliki 2 macam jenis kontrol yaitu control secara otomatis menggunakan metode *magnetic line following* dan kontrol secara manual menggunakan *joystick.* Metode *magnetic line following* digunakan agar robot tetap dapat bekerja dengan baik meskipun dalam kondisi lingkungan yang gelap dan kotor.

A. Rancangan Perangkat keras



Gambar 1. Diagram blok perangkat keras modul UV-C

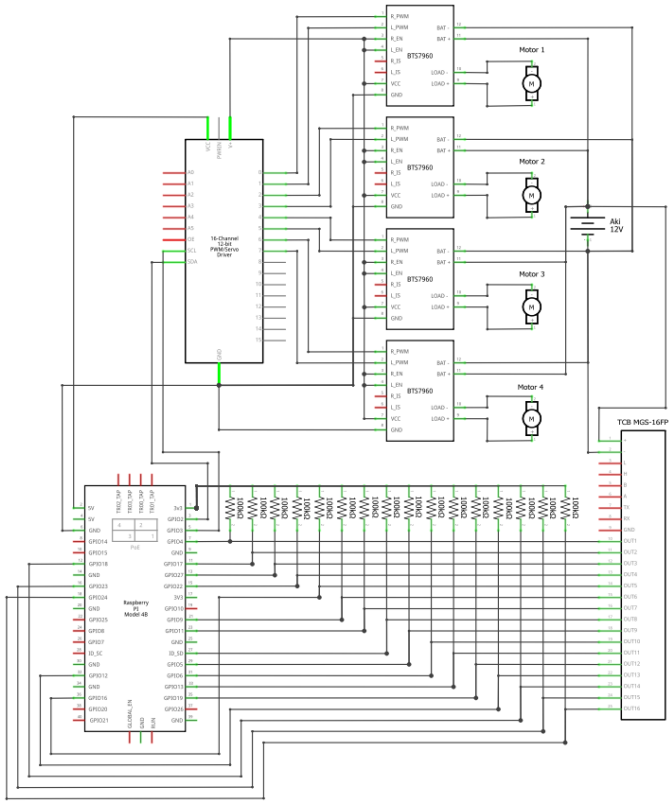


Gambar 2. Diagram blok perangkat keras penggerak robot

Robot ini tersusun dari beberapa komponen seperti yang dapat dilihat pada Gambar 1 yang merupakan diagram blok rancangan modul UV-C dari bagian penggerak robot dan Gambar 2 yang merupakan diagram blok bagian penggerak robot.

Sumber energi listrik dari robot ini berasal dari aki Incoe Maintenance Free dengan kapasitas 35Ah. Untuk *power supply* Raspberry Pi 4, digunakan modul *voltage step down* 5V dengan arus maksimal 3A, yang memiliki *output interface* USB Type A. Untuk *power supply* Wemos D1 Mini digunakan modul *voltage step down* LM2596.

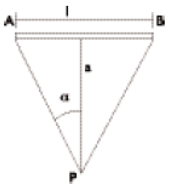
Empat motor pada robot ini dapat dikendalikan oleh Raspberry Pi 4 dengan menambahkan sebuah modul PCA9685 *16 Channel 12 bit PWM* dan 4 buah *driver* motor BTS7960 dengan rangkaian yang dapat dilihat pada Gambar 3. PCA9685 berfungsi untuk menghasilkan sinyal PWM (*Pulse Width Modulation*) yang dibutuhkan untuk mengatur kecepatan motor, sedangkan *driver* motor BTS7960 berfungsi untuk mengubah sinyal PWM 5V dari PCA6985 menjadi sinyal PWM 12V dengan arus maksimal 43 A. Robot ini akan menggunakan 8 *output* PWM dari PCA9685 karena setiap *driver* motor BTS7960 yang memerlukan 2 *input* PWM. Satu PWM untuk mengatur putaran motor secara CW (Clockwise) dan satu PWM untuk mengatur putaran motor CCW (Counter Clockwise).



Gambar 3. Diagram rangkaian perangkat keras dengan Raspberry Pi 4

Sensor magnet yang digunakan pada robot ini adalah *magnetic guide sensor* TCB MGS-16FP. Sensor ini terdiri dari 16 buah sensor magnet yang terbentang sepanjang bodi sensor yang menghasilkan *output* digital secara *open collector*. 16 *output* digital ini dihubungkan ke 16 pin GPIO raspberry Pi 4 dengan menambahkan resistor *pull up*. Rangkaian sensor magnet dengan Raspberry Pi 4 dapat dilihat pada Gambar 3. Sensor jarak VL53L0X dipasang pada 8 titik robot yaitu pada empat sudut bodi robot dan empat sisi bodi robot. Setiap sensor akan dikontrol menggunakan sebuah Wemos D1 Mini. Selain itu juga terdapat 6 buah Sensor PIR HC-SR01 yang terletak di bagian atas robot. 6 buah sensor PIR ini dikontrol menggunakan sebuah Wemos D1 Mini.

Untuk memastikan proses desinfeksi berhasil maka perlu dihitung dosis radiasi UV C dengan menggunakan persamaan dibawah ini:



$$E = \frac{\varphi}{2\pi^2 la}(2\alpha + sin2\alpha) \quad (2\text{-}1)$$

E = Radiasi UV-C pada titik P (W/m²)
φ = Daya radiasi sinar UV-C dari lampu UV-C (W)
π = konstanta phi (3.14)
l = Panjang lampu UV-C (m)
a = jarak lampu UV-C dengan permukaan (m)
α = Sudut yang terbentuk pada titik P

$$H = Et \quad (2\text{-}2)$$

H : Dosis (J/m²)
E : Radiasi UV-C (W/m²)
T = waktu pemaparan (detik)

Modul UV-C dilengkapi dengan 6 buah Lampu PHILIPS TUV 36W G36 T8 yang memiliki panjang 1200 mm dan mampu menghasilkan daya radiasi UV-C sebesar 15 watt. Dengan menggunakan rumus (2-1) maka dapat diketahui daya radiasi yang didapat setiap sisi ruangan dalam radius 5 meter, yaitu sebesar 0,136 W/m². Untuk dapat mendesinfeksi virus SARS-Cov-2, dibutuhkan dosis radiasi UV-C sebesar 27 J/m²[2]. Dengan demikian, dengan menggunakan rumus (2-2), untuk dapat mendesinfeksi virus SARS-Cov-2 dalam radius 5 meter dari robot dibutuhkan waktu 198 detik atau 3 menit 18 detik. Dengan demikian dalam waktu 3 menit 18 detik robot ini dapat mendesinfeksi area seluas 100 m².

B. Perancangan Perangkat Lunak Robot

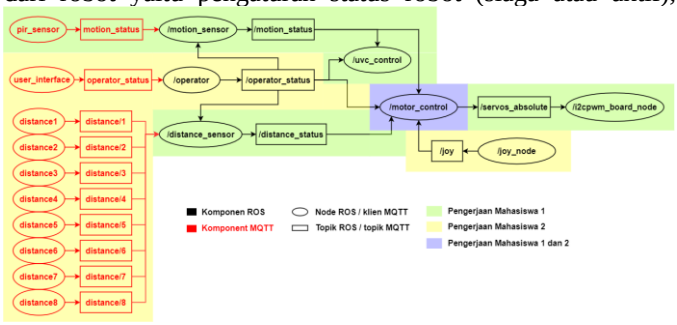
Agar robot ini dapat berjalan dengan baik dibutuhkan beberapa paket ROS antara lain:

a. Paket ROS i2c_pwm_board, yang diambil dari github dengan link: https://gitlab.com/bradanlane/ros-i2cpwmboard/
b. Paket ROS joy, yang diambil dari github dengan link: https://github.com/ros-*drivers*/joystick_*drivers*.git
c. Paket ROS common_msg, yang diambil dari github dengan link: https://github.com/ros/common_msgs
d. Paket ROS SPERO, yang dibuat oleh penulis, yang juga bisa diambil dengan link: https://github.com/Jeremywinst/spero.git

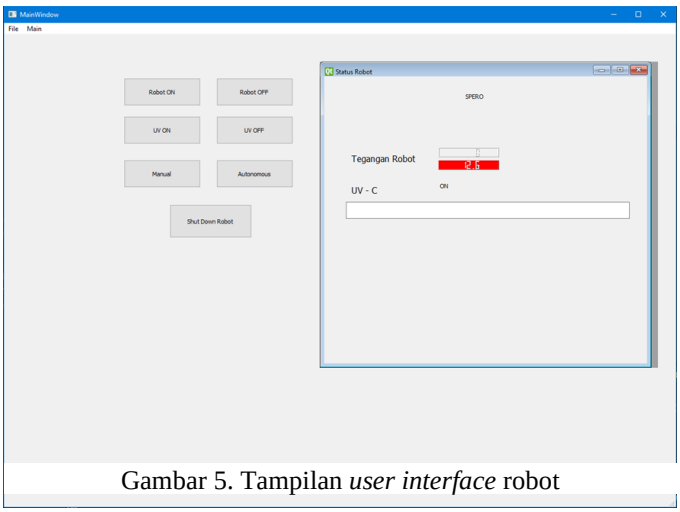
Selain ROS, robot ini juga menggunakan protokol komunikasi MQTT Kerangka kerja perangkat lunak robot beserta pembagian tugas pengerjaannya dapat dilihat pada Gambar 4.

1) User interface operator robot

Desain UI (*User interface*) dikerjakan oleh Rio Alfandy. UI ini dibuat menggunakan QT Designer 5 dengan desain sederhana agar robot dapat dikendalikan dengan mudah oleh operator. UI dari robot dapat dilihat pada Pada Gambar 5. UI ini berfungsi untuk melakukan pengaturan dan mendapatkan informasi tentang robot. Pada UI terdapat 3 macam pengaturan dari robot yaitu pengaturan status robot (siaga atau aktif),



pengaturan mode robot (manual atau otomatis), dan pengaturan lampu UVC (mati atau nyala). Pada UI ini operator juga dapat mengetahui status terkini pada robot yaitu tegangan baterai pada robot dan status lampu UV sedang aktif atau mati.



Gambar 5. Tampilan *user interface* robot

2) Program kontrol operator

Program ini merupakan sebuah node ROS bernama operator yang dikerjakan oleh Rio Alfandy. Node ini berfungsi untuk robot menerima *input* yang diberikan oleh operator melalui *user interface* menggunakan komunikasi MQTT, kemudian mem-*publish* data tersebut ke sebuah topik di dalam jaringan ROS. Node ini menerima *input* dari *user interface* dengan cara *subscribe* ke topik MQTT /operator_status_MQTT yang mana *publisher* dari topik ini adalah *user interface* itu sendiri.

3) Program pembacaan distance sensor

Program ini merupakan sebuah node ROS bernama distance_status yang dikerjakan oleh Rio Alfandy. Node ini berfungsi untuk mendapatkan *input* pembacaan jarak dari 8 sensor VL53L0X melalui *subscribe* ke topik MQTT distance_status. Selanjutnya semua nilai dari pembacaan sensor akan diolah menjadi 3 pengaturan kecepatan. Jika terdapat objek dalam radius < 15 cm dari robot maka robot akan berhenti. Jika tidak ada objek di antara radius 15 - 25 cm dari robot maka kecepatan robot akan dikurangi menjadi setengah. Selain dari itu maka robot dapat berjalan dengan kecepatan penuh.

Gambar 4. Pembagian tugas dalam konteks kerangka kerja Perangkat lunak robot

4) Program pembacaan PIR sensor

Program ini merupakan sebuah node ROS bernama motion_status yang dikerjakan oleh mahasiswa pertama. Node ini berfungsi untuk mendapatkan *input* pembacaan dari 6 sensor yang berasal dari sebuah WEMOS D1 Mini dengan komunikasi MQTT. Program ini bekerja dengan cara *subscribe* topik MQTT motion_status_MQTT. Jika terdapat satu sensor PIR yang mendeteksi adanya pergerakan manusia, node ini akan lampu UV-C akan dimatikan. Bila tidak ada satupun sensor PIR yang mendeteksi gerakan maka lampu UV-C akan dinyalakan.

5) Program control I2C PWM board

Program ini merupakan sebuah node ROS bernama i2cpwm_board_node. Node ini merupakan node yang berfungsi untuk mengatur *output* PWM dari modul PCA9685. Terdapat 2 komponen utama yang akan digunakan dalam node ini yaitu *service* ROS set_pwm_frequency() dan *subscriber* servos_absolute().

Service ROS set_pwm_frequency() ini berfungsi untuk mengubah frekuensi dari PWM yang dihasilkan. Secara *default* PCA9685 frekuensi PWM yang dihasilkan adalah 50 Hz. Namun untuk dapat menggerakkan motor dengan halus dibutuhkan frekuensi yang tinggi. *Subscriber* servos_absolute() merupakan node yang berfungsi untuk *subscribe* ke topik /servos_absolute yang nantinya dapat mengubah *duty cycle* untuk setiap *channell* PWM yang diinginkan.

6) Program pengendalian dengan joystick

Program ini merupakan sebuah node ROS yang bernama joy_node. Node ini berfungsi untuk membaca *input* dari setiap tombol yang ada di *joystick* dan *publish* semua data itu ke topik /joy. *Joystick* yang digunakan untuk robot ini adalah *joystick* Logitech F710.

7) Program kontrol motor

Program ini merupakan sebuah node ROS bernama motor_control. Node ini adalah program yang bertanggung jawab atas pengaturan pergerakan empat motor yang berada pada robot. Dalam node ini terdapat 2 class utama yaitu MagnetControl() dan MotorControl(). Class MagnetControl() berfungsi untuk mendapatkan 16 *input* digital dari sensor magnet dan mengkonversikannya menjadi 2 buah variabel kecepatan motor sebelah kanan dan sebelah kiri. Class MotorControl() merupakan class yang berisi program utama untuk pengontrolan motor pada robot. Class ini berisi program yang akan *subscribe* ke topik /servos_absolute dengan memberikan id PWM dan nilai *duty cycle* yang diinginkan

Selanjutnya untuk dapat berkomunikasi dengan node yang lain, node ini akan *publish* ke topik /servos_absolute serta *subscribe* ke 5 buah topik yaitu /operator, /joy, /follow_line, /distance_status, dan /motion_status.

8) Program kontrol lampu UV-C

Program ini merupakan sebuah node ROS bernama uvc_control. Node ini adalah program yang bertanggung jawab atas pengaturan semua lampu UV-C pada robot. Lampu UV-C ini dapat dikendalikan oleh 2 node yaitu note operator dan node node motion_status. Oleh karena itu Node ini akan melakukan *subscribe* ke 2 topik yaitu topik operator_status dan motion_status.

C. Konfigurasi jaringan

Untuk komunikasi antar *microprocessor* seperti Raspberry Pi 4 dengan Wemos D1 mini maka dibutuhkan konfigurasi jaringan agar dapat berkomunikasi. Pengiriman data menggunakan protokol MQTT sehingga data sensor yang terpasang pada Wemos D1 mini dapat diterima oleh Raspberry Pi 4 untuk memproses seluruh proses robot.

Untuk komunikasi Raspberry Pi 4 dengan Raspberry Pi 400 untuk operator digunakan access point yang berada pada luar robot dan untuk komunikasi antara Raspberry Pi dengan Wemos D1 mini untuk menerima data sensor digunakan Router Prolink PRN 2001 yang berada pada dalam robot.

Gambar 6. Konfigurasi jaringan

III. PENGUJIAN DAN HASIL

A. Pengujian sensor magnet

Pengujian ini dilakukan untuk mengetahui medan magnet yang dihasilkan oleh lintasan magnet dan jangkauan jarak pendeteksian lintasan magnet oleh sensor magnet TCB MGS-16FP yang digunakan pada robot ini. Pengukuran medan magnet dilakukan menggunakan tesla meter. Adapun 2 jenis magnet yang digunakan dalam pengujian ini yaitu 2 jenis magnet yaitu *magnetic tape* 3M dengan lebar 10 mm tebal 2.2 mm dan magnet ferit berdiameter 18 mm dengan tebal 3 mm.

Tabel 1. Data pengujian sensor magnet1

Jarak	Magnet Terdeteksi		Medan Magnet	
	Magnet Ferit	Magnet Strip	Magnet Ferit	Magnet Strip
0 mm	Ya	Ya	35.8 mT	32.6 mT
5 mm	Ya	Tidak	13.8 mT	-1.4 mT
10 mm	Ya	Tidak	4.5 mT	-0.4 mT
15 mm	Ya	Tidak	2.3 mT	-0.3 mt
20 mm	Ya	Tidak	1.6 mT	-0.2 mT
25 mm	Ya	Tidak	1 mT	-0.1mT

30 mm	Ya	Tidak	0.7 mT	0 mT
35 mm	Tidak	Tidak	0.4 mT	0 mT
40 mm	Tidak	Tidak	0.3 mT	0 mT

Dari hasil pengujian pada Tabel 1, dapat diketahui bahwa sensor magnet TCB MGS-16FP hanya mampu mendeteksi sebuah medan magnet diatas 0.7 mT. Oleh karena itu lintasan magnet yang dapat digunakan adalah lintasan magnet ferit dengan jarak antara sensor magnet dan lintasan magnet adalah diantara 5 – 30 mm.

B. Pengujian respon motor terhadap sensor VL53L0X

Pengujian ini dilakukan untuk mengetahui respon motor terhadap jarak yang dibaca oleh sensor VL53l0X. Sebelum itu perlu dilakukan pengujian terhadap sensor VL53L0X untuk mengetahui jangkauan jarak dari objek yang dapat dideteksi oleh sensor VL53L0X. Dari hasil pengujian jangkauan jarak, dapat diketahui bahwa sensor VL53L0X memiliki jangkauan pembacaan jarak hingga 120 cm, yang mana jarak ini lebih dari cukup untuk digunakan dalam robot ini.

Selanjutnya dilakukan pengujian respon motor terhadap pembacaan sensor ini dengan hasil yang ditampilkan pada Tabel 2 Dari hasil pengujian pada Tabel 2, dapat disimpulkan bahwa motor dapat merespon dengan baik sesuai dengan program yang dibuat.

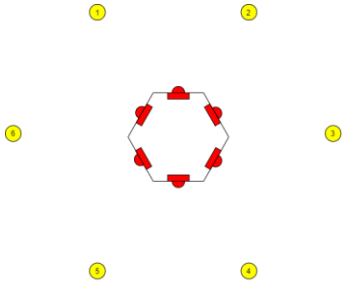
Tabel 2. Respon motor terhadap sensor VL53L0X

Jarak Objek	Respon Motor
2 cm	Berhenti
4 cm	Berhenti
6 cm	Berhenti
8 cm	Berhenti
10 cm	Berhenti
12 cm	Berhenti
14 cm	Berhenti
16 cm	Pelan
18 cm	Pelan
20 cm	Pelan
22 cm	Pelan
24 cm	Pelan
26 cm	Cepat
28 cm	Cepat
30 cm	Cepat

C. Pengujian respon lampu UV-C terhadap sensor PIR HC-SR01

Pengujian ini dilakukan untuk mengetahui respon motor terhadap pendeteksian gerakan manusia oleh sensor PIR HC-SR01 dan mengetahui *blind spot* pendeteksian gerakan. Namun sebelum itu perlu diketahui jangkauan dari sensor PIR HC-SR01 dalam mendeteksi gerakan manusia. Hasil dari pengujian jangkauan sensor PIR HC-SR01 dapat diketahui bahwa sensor HC-SR01 memiliki jangkauan sebesar 600 cm.

Selanjutnya dilakukan pengujian *blind spot* pada robot baik secara horizontal dan vertikal. Pengujian secara horizontal di dilakukan pada 6 buah titik yang terletak diantara 2 buah sensor PIR HC-SR01, seperti pada Gambar 7, sedangkan pengujian vertikal dilakukan pada daerah yang di tunjukkan oleh gambar 8. Hasil dari pengujian *blind spot* secara horizontal dapat dilihat pada Tabel 3 dan pengujian *blind spot* secara vertikal dapat

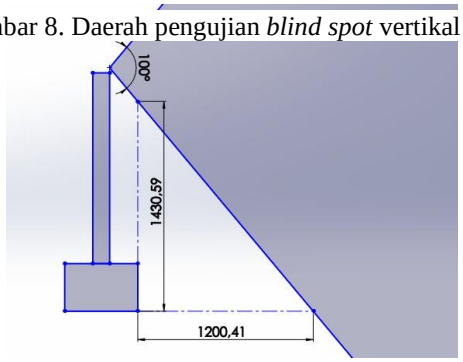


Gambar 7. Titik-titik pengujian *blind spot* horizontal dilihat pada Tabel 4.

Tabel 3. Data pengujian *blind spot* horizontal

Posisi	Jarak	Deteksi gerakan
1	600 cm	Ya
2	600 cm	Ya
3	600 cm	Ya
4	600 cm	Ya
5	600 cm	Ya
6	600 cm	Ya

Gambar 8. Daerah pengujian *blind spot* vertikal



Tabel 4. Data pengujian *blind spot* vertikal

Sumbu X	Sumbu Y	Deteksi Gerakan
0 mm	1430 mm	Tidak
200 mm	1192 mm	Tidak
400 mm	954 mm	Tidak
600 mm	715 mm	Tidak
800 mm	477 mm	Tidak
1000 mm	239 mm	Tidak

Dari kedua tabel ini dapat disimpulkan bahwa pada jarak lebih dari 100 cm robot tidak memiliki *blind spot* baik secara horizontal maupun vertikal sedangkan pada jarak di bawah 100 cm dari robot, terdapat sebuah area *blind spot*. yang membentuk sebuah kerucut dengan tinggi 170 cm (sama dengan tinggi robot) dan jari jari 125 cm dari titik tengah robot.

Selanjutnya dilakukan pengujian respon lampu UV-C terhadap pembacaan sensor HC-SR01 dengan hasil yang ditampilkan pada Tabel 5. Dari hasil pengujian pada Tabel 5, dapat disimpulkan bahwa lampu UV-C dapat merespon dengan baik sesuai dengan program yang dibuat.

Tabel 5. Respon lampu UV-C terhadap sensor PIR HC-SR01

Posisi	Respon Lampu UV-C	
	Ada Gerakan	Tidak Ada Gerakan
1	Mati	Nyala
2	Mati	Nyala
3	Mati	Nyala
4	Mati	Nyala
5	Mati	Nyala
6	Mati	Nyala
7	Mati	Nyala
8	Mati	Nyala
9	Mati	Nyala
10	Mati	Nyala
11	Mati	Nyala
12	Mati	Nyala

D. Pengujian magnetic line following

Pengujian ini dilakukan untuk menguji fungsi *line following* dari robot dan melakukan *tuning* PID secara manual, sehingga robot dapat menyelesaikan berbagai macam lintasan dengan handal. Pertama-tama dilakukan eksperimen menentukan nilai Kp untuk mengurangi rise time dari sistem. Hasil eksperimen nilai Kp dapat dilihat pada Tabel 6.

Tabel 6. Percobaan nilai Kp

Kp	Performa
1.0	Overdamped
1.2	Overdamped
1.4	Overdamped
1.6	Critically Damped
1.8	Underdamped
2.0	Underdamped
2.2	Underdamped
2.4	Underdamped
2.6	Underdamped
2.8	Underdamped
3.0	Underdamped

Dari hasil eksperimen *tuning* nilai Kp dapat diamati saat Kp bernilai 1.0-1.4, robot membutuhkan waktu yang relatif lambat untuk bisa kembali mengikuti garis ketika terdapat belokan jika dibandingkan dengan percobaan nilai Kp yang lain. Hal ini membuat robot mengalami *overdamped* yang dapat menyebabkan robot bergerak ke luar garis ketika menghadapi jalur yang berbelok. Saat Kp bernilai 1.8-3.0, robot memberikan respon yang agresif terhadap perubahan error yang kecil. Hal ini membuat robot mengalami *underdamped* yang menyebabkan sistem kontrol robot menjadi seperti sistem kontrol *on* dan *off*.

Namun, Saat Kp bernilai 1.6, dapat diamati bahwa robot membutuhkan waktu yang relatif cepat untuk bisa kembali mengikuti garis ketika terdapat belokan jika dibandingkan dengan percobaan nilai Kp yang lain tanpa menimbulkan respon yang agresif. Hal ini membuat sistem kontrol robot yang *critical damped*. Selanjutnya dengan nilai Kp = 1.6 tidak tampak sistem yang berosilasi dan juga tidak tampak *steady state error* sehingga penggunaan parameter proporsional saja sudah cukup untuk sistem kontrol robot ini.

E. Pengujian daya tahan baterai robot

Pengujian ini dilakukan untuk menguji daya tahan baterai robot, untuk mengetahui berapa luas daerah yang dapat didesinfeksi oleh robot dari baterai dalam sekali *charge.*

Dari hasil tabel 7 dapat diketahui robot dapat melakukan proses desinfeksi secara otomatis selama 52 menit. Mengingat bahwa satu siklus desinfeksi berdurasi 3 menit 18 detik mampu mendesinfeksi area seluas 100 m², maka dapat disimpulkan bahwa dalam sekali *charge* robot ini dapat mendesinfeksi area seluas 1575 m² dengan waktu 52 menit.

Tabel 7. Data percobaan daya tahan baterai

Percobaan	Waktu	Keterangan
1	55 menit	Tegangan Awal aki = 13.2 V Tegangan terakhir aki = 10.2 V
2	50 menit	Tegangan Awal aki = 13.2 V Tegangan terakhir aki = 10.2 V
3	52 menit	Tegangan Awal aki = 13.2 V Tegangan terakhir aki = 10.2 V

IV. KESIMPULAN

Dari seluruh perencanaan, implementasi, dan pengujian dari robot desinfeksi ruangan ini didapatkan kesimpulan sebagai berikut:

1. Dengan baterai penuh robot ini mampu mendesinfeksi area seluas 1575 m² dengan waktu 52 menit.
2. Sensor magnet yang digunakan dalam robot ini hanya mampu mendeteksi medan magnet lebih dari 0.7 mT. Dengan demikian dibutuhkan sebuah jalur magnet yang dapat menghasilkan medan magnet lebih besar dari 0.7mT. Lintasan untuk robot desinfeksi ini menggunakan magnet ferit dengan diameter 18mm dan tebal 3 mm, dengan jarak maksimal sensor magnet dengan lintasan magnet adalah 30 mm.
3. Konfigurasi dari sensor PIR HC-SR01, membuat robot ini mampu mendeteksi gerakan manusia dalam radius 6 meter dari robot. Namun terdapat sebuah *blind spot* yang terletak di 100 cm pertama di depan robot, membentuk sebuah kerucut dengan tinggi 170 cm (sama dengan tinggi robot) dan jari jari 125 cm dari titik tengah robot. Untuk mengatasi

masalah ini operator robot dapat mengawasi robot melalui *user interface* yang sudah disediakan.

V. DAFTAR PUSTAKA

[1] W. A. Rutala and D. J. Weber, "Disinfection and sterilization in healthcare facilities, 2008," *Bennett Brachman's Hosp. Infect. Sixth Ed.*, no. May, pp. 55–56, 2013, doi: 10.1017/9781107153165.009.

[2] W. J. Kowalski, "2020 COVID-19 Coronavirus Ultraviolet Susceptibility 2020 COVID-19 Coronavirus Ultraviolet Susceptibility," *Purple Sun*, no. March, pp. 1–4, 2020.