

**IMPLEMENTASI  
ALGORITMA DIJKSTRA PADA ROBOT PEMADAM API**

**(Tesis)**

**Oleh  
SONIA WAHYUNI  
NPM 2225031009**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2026**

**IMPLEMENTASI  
ALGORITMA DIJKSTRA PADA ROBOT PEMADAM API**

**Oleh  
SONIA WAHYUNI**

**Tesis**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
MAGISTER TEKNIK ELEKTRO**

**Pada**

**Program Pascasarjana Magister Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**PROGRAM PASCASARJANA MAGISTER TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2026**

**ABSTRACT**  
**IMPLEMENTASI**  
**ALGORITMA DIJKSTRA PADA ROBOT PEMADAM API**

**By**  
**Sonia Wahyuni**

Fire incidents remain a critical threat in Indonesia, causing substantial losses in safety, economic productivity, and environmental stability. To address these challenges, this research presents the design of an autonomous fire-fighting robot based on the ESP32 microcontroller, integrating the Dijkstra algorithm to determine the shortest path toward the fire source. The robot is equipped with a Sharp GP2Y0A21 distance sensor, a photodiode sensor for arena line tracking, and a *flame* sensor for fire detection. Sensor data are processed in real time and translated into motor control via a MOSFET driver, status visualization through an OLED LCD display, and automatic activation of the extinguishing fan upon reaching the fire point. Experimental testing was conducted in a maze-like arena measuring  $2000 \times 1600 \times 400$  mm, with seven repeated trials for each navigation scenario. In the scenario toward Room A, the path with a weight of 11 achieved an average travel time of  $7.25 \pm 0.11$  seconds, while the alternative weighted 19 required  $16.23 \pm 0.28$  seconds, demonstrating a statistically significant difference ( $p < 0.01$ ). Similar performance trends were observed in Rooms B, C, and D. The results confirm that the adapted Dijkstra implementation operates efficiently and reliably in a controlled static laboratory environment. Further evaluation is recommended for dynamic and real-fire conditions.

**Keywords :** Dijkstra Algorithm, Robot Navigation, Fire-Fighting Robot, ESP32, Experimental Validation.

**ABSTRAK**

**IMPLEMENTASI**

**ALGORITMA DJIKSTRA PADA ROBOT PEMADAM API**

**Oleh**  
**Sonia Wahyuni**

Kebakaran merupakan permasalahan serius di Indonesia yang berdampak pada aspek keselamatan, ekonomi, dan lingkungan, sehingga diperlukan sistem mitigasi berbasis teknologi yang mampu beroperasi secara cepat dan presisi. Penelitian ini merancang robot pemadam api otonom berbasis mikrokontroler ESP32 yang mengintegrasikan algoritma djikstra untuk menentukan jalur terpendek menuju titik sumber api. Robot dilengkapi dengan sensor Sharp GP2Y0A21 sebagai pendeteksi jarak, sensor fotodioda sebagai pembaca garis arena, serta sensor *flame* untuk deteksi nyala api. Data sensor diproses secara real-time dan diterjemahkan menjadi kendali motor melalui driver MOSFET, tampilan status pada LCD OLED, serta aktivasi kipas pemadam saat robot mencapai target. Pengujian dilakukan pada arena labirin berukuran  $2000 \times 1600 \times 400$  mm dengan pengulangan sebanyak tujuh kali pada setiap skenario jalur. Pada skenario menuju Ruangan A, jalur berbobot 11 memiliki waktu tempuh rata-rata  $7.25 \pm 0.11$ , sedangkan jalur alternatif berbobot 19 membutuhkan  $16.23 \pm 0.28$  detik, dengan perbedaan signifikan ( $p < 0,01$ ). Pola serupa ditemukan pada Ruangan B, C, dan D, menunjukkan bahwa jalur dengan bobot minimum menghasilkan navigasi yang lebih efisien. Hasil penelitian ini membuktikan bahwa implementasi Dijkstra pada ESP32 mampu bekerja efektif pada lingkungan laboratorium statis. Evaluasi pada kondisi dinamis dan kebakaran nyata menjadi arah penelitian berikutnya.

**Kata kunci :** Algoritma Dijkstra, Navigasi Robot, Robot Pemadam Api, ESP32, Pengujian Eksperimental.

Judul Tesis : Implementasi Algoritma Dijkstra Pada Robot Pemadam Api

Nama Mahasiswa : Sonia Wahyuni

Nomor Pokok Mahasiswa : 2225031009

Program Studi : Magister Teknik Elektro

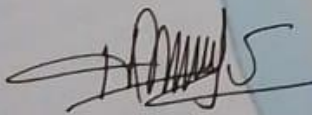
Fakultas : Teknik

**MENYETUJUI**

1. **Komisi Pembimbing**

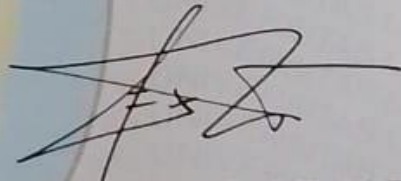
Pembimbing I

Pembimbing II



**Dr. Sri Purwiyanti, S.T., M.T., Ph.D**

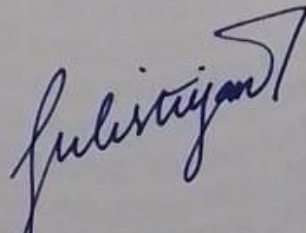
NIP 197310 04199803 2 001



**Dr. Eng. F.X. Arinto S., S.T., M.T.**

NIP 19691219 19990 3 1002

2. **Ketua Program Studi Magister Teknik Elektro**



**Prof. Dr. Ir. Sri Ratna Sulistiyanti, M.T.**

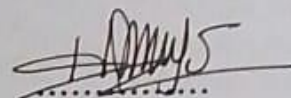
NIP 19651021 199512 2 001

## MENGESAHKAN

### 1. Komisi Penguji 1

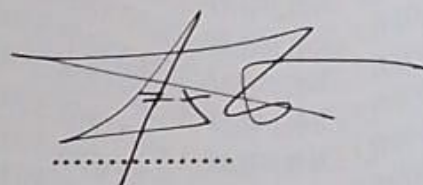
Ketua Komisi Penguji

(Pembimbing I) : **Dr. Sri Purwiyanti, S.T., M.T., Ph.D**



Sekretaris Komisi Penguji

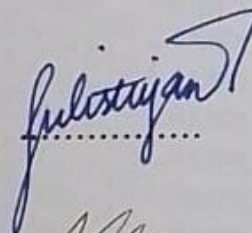
(Pembimbing II) : **Dr. Eng. F.X. Arinto S., S.T., M.T.**



Anggota Komisi Penguji

(Penguji I)

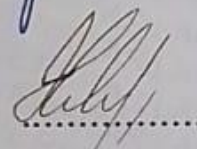
: **Prof. Dr. Ir. Sri Ratna Sulistiyanti, M.T.**



Anggota Komisi Penguji

(Penguji II)

: **Dr. Eng. Helmy Fitriawan, S.T., M.Sc.**



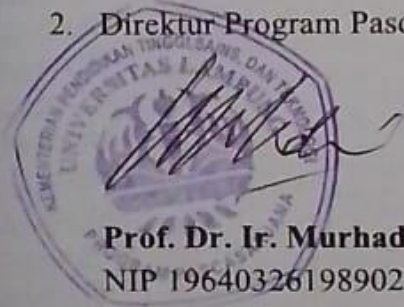
### 1. Dekan Fakultas Teknik



**Dr. Ahmad Herison, S.T., M.T.**

NIP 196910302000031001

### 2. Direktur Program Pascasarjana



**Prof. Dr. Ir. Murhadi, M.Si.**

NIP 196403261989021001

Tanggal Lulus Ujian Tesis : **19 Januari 2026**

## LEMBAR PERNYATAAN

Saya menyatakan dengan sesungguhnya bahwa tesis yang saya susun sebagai syarat untuk memperoleh gelar Magister Teknik dari Program Pascasarjana Magister Teknik Elektro seluruhnya merupakan hasil karya sendiri.

Adapun bagian-bagian tertentu dalam penulisan tesis yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah.

Tesis dengan judul “Implementasi Algoritma Dijkstra Pada Robot Pemadam Api” dapat diselesaikan berkat bimbingan dan motivasi dari pembimbing-pembimbing saya, yaitu :

1. **Dr. Sri Purwiyanti, S.T., M.T., Ph.D**
2. **Dr. Eng. F.X. Arinto S., S.T., M.T.**

Saya Ucapkan terima kasih yang sebesar besarnya kepada semua pihak, khususnya kedua dosen pembimbing dan Bapak/Ibu Dosen Program Studi Magister Teknik Elektro Universitas Lampung yang banyak memberikan ilmu pengetahuan, bimbingan dan motivasi.

Apabila dikemudian hari ditemukan seluruh atau sebagian tesis ini bukan hasil karya sendiri atau adanya plagiat dalam bagian-bagian tertentu, saya bersedia menerima sanksi pencabutan gelar akademik yang saya sandang dan sanksi-sanksi lainnya sesuai dengan peraturan perundangan yang berlaku.

Bandar Lampung, Januari 2026



**Sonia Wahyuni**

NPM. 2225031009

## RIWAYAT HIDUP



Penulis dilahirkan di Sekayu pada tanggal 20 Maret 1999. Penulis merupakan anak ke-5 dari 5 bersaudara dari pasangan Bapak Zuribnu A. Hamid dan Ibu Nur Asmala. Penulis juga mempunyai satu orang putra yang bernama Rayyan Ghani Alfatih. Pendidikan dasar ditempuh di MI Istiqomah dan lulus pada tahun 2011. Pendidikan menengah pertama diselesaikan di SMP Negeri 1 Sekayu,

Kecamatan Sekayu, dan lulus pada tahun 2014. Pendidikan menengah kejuruan diselesaikan di SMA Negeri 1 Sekayu lulus pada tahun 2017.

Pada tahun 2017, penulis melanjutkan pendidikan ke jenjang perguruan tinggi di Universitas Pendidikan Indonesia pada Program Studi S-1 Pendidikan Teknik Elektro dengan konsentrasi keahlian Elektronika Industri.

Pada tahun 2021, penulis menyelesaikan Program Studi Studi S-1 Pendidikan Teknik Elektro dengan judul tugas akhir “Rancang Bangun Dispenser Sabun cuci tangan Otomatis dilengkapi Timer” dan skripsi dengan judul “*Project Based Learning* Mata Pelajaran Rangkaian Elektronika dan Kontroller Berbasis Proteus Selama Masa Pandemi Covid-19 di SMKN 12 Bandung”.

Pada tahun 2022, penulis terdaftar sebagai Mahasiswa Program Pascasarjana Magister Teknik Elektro Universitas Lampung. Pada tahun 2024, penulis melakukan penelitian pada bidang robotika dengan judul tesis “Implementasi Algoritma Dijkstra Pada Robot Pemadam Api” di bawah bimbingan Ibu Dr. Sri Purwiyanti, S.T., M.T. dan Bapak Dr. Eng. F.X. Arinto S., S.T., M.T.

Bandar Lampung, Januari 2026

Penulis

**Sonia Wahyuni**



### **PERSEMBAHAN**

Dengan Ridho Allah SWT, teriring shalawat kepada Nabi Muhammad SAW

Karya Tulis ini kupersembahkan untuk:

#### **Ayah dan Ibu Tercinta**

Bapak Zuribnu dan Ibu Nur Asmala.

#### **Keluarga Kecil Tersayang**

Suamiku Rici Riansyah S.Tr,T. dan Anakku tersayang Rayyan Ghani Alfatih.

#### **Dosen Teknik Elektro**

Yang selalu membimbing, mengajarkan, memberikan saran, baik secara akademis maupun non akademis.

#### **Teman- teman kebanggaanku**

Rekan – rekan Jurusan Teknik Elektro.

#### **Sahabat-sahabatku**

Yang selalu membantu, memberikan semangat, mendukung menuju keberhasilan, serta berbagi cerita suka duka dalam berkeluh kesah.

#### **Keluarga Besar Magister Teknik Elektro 2022**

Yang selalu memberi semangat, dukungan dalam proses yang sangat panjang, dan selalu berdiri bersama dalam perjuangan menuju kesuksesan.

#### **Almamaterku**

Universitas Lampung

#### **Bangsa dan Negaraku**

Republik Indonesia

Terima kasih untuk semua yang telah diberikan kepadaku. *Jazakallah Khairan*



## MOTTO

*“Setiap proses memiliki waktu terbaiknya, tugas manusia adalah berusaha dan berserah”*

*“Kesalahan bukan kegagalan, tetapi data untuk perbaikan”*

*”Sebaik Apapun Perkataan, Jika Tidak Ada Aksi, Maka Tidak Akan ada Artinya. Maka Bangun dan Bergegaslah Merangkai Mimpi yang Didapat dari Tidur Semalam”*

*-Sonia Wahyuni-*

## SAN WACANA

### **Assalamu'alaikum Wr. Wb.**

Syukur **Alhamdulillahirabbil'alamin**, penulis haturkan puji syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat, hidayah, serta inayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan laporan Tesis ini dengan judul **“Implementasi Algoritma Dijkstra Pada Robot Pemadam Api ”** dengan sebaik-baiknya.

Shalawat beriring salam senantiasa tercurah kepada junjungan alam, Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya yang istiqamah menegakkan agama Islam hingga akhir zaman. Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister, sekaligus sebagai wadah pembelajaran dan pengembangan ilmu di bidang teknik kendali dan kecerdasan buatan. Dalam proses penyusunan tesis ini, penulis menghadapi berbagai tantangan dan hambatan, namun berkat bimbingan, doa, serta dukungan dari berbagai pihak, akhirnya penulis dapat menyelesaikannya dengan baik.

Dalam penyusunan tugas akhir ini penulis banyak mendapatkan bimbingan, motivasi, serta bantuan baik moral maupun material dari berbagai pihak. Oleh karena itu, dengan sepuh ketulusan hati, penulis menyampaikan **ucapan terima kasih** kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Prof. Dr. Ir. Murhadi, M.Si., selaku Direktur Program Pascasarjana Fakultas Teknik Universitas Lampung.
3. Bapak Dr. Ahmad Herison, S.T., M.T. selaku Dekan Fakultas Teknik, Universitas Lampung.
4. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
5. Ibu Dr. Sri Purwiyanti, S.T., M.T., selaku dosen pembimbing utama tesis yang banyak memberikan waktu, ide pemikiran dan semangat serta motivasi bagi penulis.

6. Bapak Dr. Eng. F.X. Arinto Setyawan, S.T., M.T., selaku pembimbing kedua tesis, yang telah banyak memberikan waktu, pengalaman, motivasi dan pemikiran bagi penulis.
7. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku dosen penguji kedua yang telah banyak memberikan kritik dan saran yang bermanfaat bagi penulis.
8. Seluruh Dosen Program Studi Magister Teknik Elektro Universitas Lampung, berkat ilmu yang telah diajarkan kepada penulis selama penulis menjalani masa studi di perkuliahan.
9. Seluruh Tenaga Pendidik Program Studi Magister Teknik Elektro Fakultas Teknik Universitas Lampung yang telah banyak membantu kepada penulis, sehingga dapat menyelesaikan tugas akhir ini.
10. Seluruh teman-teman Program Studi Magister Teknik Elektro UNILA angkatan 2022 untuk kebersamaan yang telah dijalani. Tiada kata yang dapat penulis utarakan untuk mengungkapkan perasaan senang dan bangga menjadi bagian dari angkatan 2022.
11. Kedua orang tua, ayah dan ibu yang selalu menyayangi, mendidik, membimbing, berkorban serta mendoakan penulis.
12. Suamiku Rici Riansyah yang selalu berdoa, memberi dukungan moral, berjuang bersama-sama dan materil mengenai kebutuhan terkait perkuliahan hingga kehidupan agar semua berjalan lancar.
13. Anakku Rayyan Ghani Alfatih yang tanpa rewel ditinggal-tinggal, ikut menyemangati dan membersamai menyelesaikan perkuliahan ini.
14. Serta semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak bisa penulis sebutkan satu-persatu.

Akhir kata, Penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan, baik dari segi isi maupun cara penyajiannya. Oleh karena itu, Penulis sangat mengharapkan saran serta kritik yang bersifat membangun dari

pembaca. Akhir kata semoga karya sederhana ini dapat berguna dan bermanfaat bagi kita semua. Aamiin Allahumma Aamiin.

**Wassalamu'alaikum Wr. Wb.**

Bandar Lampung, Januari 2026

Penulis,

Sonia Wahyuni

## DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
ABSTRACT.....	iii
ABSTRAK.....	iv
LEMBAR PENGESAHAN .....	vi
SAN WACANA.....	xi
DAFTAR ISI.....	xiv
DAFTAR GAMBAR .....	xvi
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Tujuan .....	2
1.3    Manfaat Penelitian .....	3
1.4    Rumusan Masalah .....	3
1.5    Batasan Masalah.....	3
1.6    Hipotesis.....	3
1.7    Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1    Pengertian Robot.....	6
2.2    Graf .....	8
2.3 <i>Shortest Path</i> .....	9
2.4    Algoritma Dijkstra .....	10
2.5    Penelitian Terdahulu .....	13
BAB III METODE PENELITIAN.....	16
3.1    Waktu dan Tempat Penelitian .....	16
3.2    Alat dan Bahan Penelitian.....	16
3.3    Prosedur Penelitian.....	17
3.4    Perancangan Robot.....	19
3.5    Desain Perangkat Keras .....	21

3.6	Perancangan Algoritma Dijkstra .....	23
3.7	Pembentukan Graf Arena dan Penentuan Bobot.....	26
BAB IV HASIL DAN PEMBAHASAN .....		30
4.1	Pengujian Sistem.....	30
4.2	Hasil Rancangan Denah Arena .....	31
4.3	Hasil Perancangan Elektronik .....	32
4.4	Hasil Pengujian Algoritma Dijkstra .....	34
4.4.1	Pengujian Ruang A .....	34
4.4.2	Pengujian Ruang B.....	37
4.4.3	Pengujian Ruang C.....	41
4.4.4	Pengujian Ruang D .....	44
4.4.5	Pengujian Akurasi Titik Berhenti Robot terhadap Titik Api .....	47
4.4.6	Analisis <i>True-False</i> Pemilihan Jalur Terpendek .....	49
4.4.7	Analisis <i>True-False</i> Deteksi Api dan Aktivasi Kipas .....	51
4.5	Analisis Perbandingan dengan Algoritma Baseline .....	53
BAB V KESIMPULAN DAN SARAN.....		57
5.1	Kesimpulan .....	57
5.2	Saran.....	57
DAFTAR PUSTAKA .....		60
LAMPIRAN.....		63

## DAFTAR GAMBAR

<b>Gambar</b>	<b>Halaman</b>
<b>Gambar 2. 1</b> Karakteristik Robot.....	8
<b>Gambar 2. 2</b> Graf Serderhana .....	9
<b>Gambar 2. 3</b> Graf umum.....	9
<b>Gambar 2. 4</b> Graf Algoritma Djikstra.....	11
<b>Gambar 3. 1</b> Flowchart .....	18
<b>Gambar 3. 2</b> Blok diagram .....	20
<b>Gambar 3. 3</b> Perancangan Kerangka Robot Pemadam Api.....	22
<b>Gambar 3. 4</b> Skema Rangkaian Robot Pemadam Api.....	23
<b>Gambar 3. 5</b> Labirin Arena Percobaan Robot .....	24
<b>Gambar 3. 6</b> Representasi graf berbobot .....	25
<b>Gambar 3. 7</b> Flowchart implementasi sistem navigasi robot pemadam api berbasis algoritma djikstra .....	29
<b>Gambar 4. 1</b> Denah Arena Pengujian Robot Pemadam Api .....	31
<b>Gambar 4. 2</b> Hasil Perancangan Robot Pemadam Api.....	33
<b>Gambar 4. 3</b> Potongan program perhitungan jalur terpendek menggunakan algoritma djikstra .....	34
<b>Gambar 4. 4</b> Penentuan Rute Terpendek dari Titik Home (H) ke Ruangan A....	35
<b>Gambar 4. 5</b> Penentuan Rute Terpendek dari Titik Home (H) ke Ruangan B ....	38
<b>Gambar 4. 6</b> Penentuan Rute Terpendek dari Titik Home ke Ruangan C.....	41
<b>Gambar 4. 7</b> Penentuan Rute Terpendek dari Titik Home ke Ruangan D .....	45
<b>Gambar 4. 8</b> kondisi FALSE pada konfigurasi arena pengujian .....	50
<b>Gambar 4. 9</b> kondisi FALSE pada konfigurasi deteksi api .....	52
<b>Gambar 4. 10</b> Potongan program BFS .....	54

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
<b>Tabel 2. 1</b> Penjelasan Graf Algoritma Dijkstra .....	11
<b>Tabel 3. 1</b> Spesifikasi Robot dan Arena Robot .....	21
<b>Tabel 3. 2</b> Bobot Jalur Antar Simpul.....	25
<b>Tabel 4. 1</b> Penentuan Bobot Jalur.....	35
<b>Tabel 4. 2</b> Hasil Pengujian Pemilihan Jalur Algoritma Dijkstra Menuju Ruang A .....	36
<b>Tabel 4. 3</b> Hasil Pengujian Waktu Tempuh Jalur Alternatif (Pemilihan Manual).....	37
<b>Tabel 4. 4</b> Penentuan Bobot Jalur Menuju Ruangan B .....	38
<b>Tabel 4. 5</b> Hasil Pengujian Pemilihan Jalur Algoritma Dijkstra Menuju Ruang B .....	39
<b>Tabel 4. 6</b> Hasil Pengujian Waktu Tempuh Jalur Alternatif (Pemilihan Manual).....	40
<b>Tabel 4. 7</b> Penentuan Bobot Jalur Menuju Ruangan C .....	42
<b>Tabel 4. 8</b> Hasil Pengujian Pemilihan Jalur Algoritma Dijkstra Menuju Ruang C .....	43
<b>Tabel 4. 9</b> Hasil Pengujian Waktu Tempuh Jalur Alternatif Menuju Ruang C....	43
<b>Tabel 4. 10</b> Perhitungan Bobot Jalur Menuju Ruangan D .....	45
<b>Tabel 4. 11</b> Hasil Pengujian Pemilihan Jalur Algoritma Dijkstra Menuju Ruang D .....	46
<b>Tabel 4. 12</b> Hasil Pengujian Waktu Tempuh Jalur Alternatif (Pemilihan Manual) .....	47
<b>Tabel 4. 13</b> Hasil Pengujian Jarak Titik Berhenti Robot terhadap Titik Api .....	48
<b>Tabel 4.14</b> Perbandingan Hasil Perhitungan Jalur Dijkstra dan BFS.....	55
<b>Tabel 4. 15</b> Pengujian Waktu Tempuh Berdasarkan Jalur Algoritmis.....	56

**DAFTAR LAMPIRAN**

<b>Lampiran</b>	<b>Halaman</b>
Lampiran 1. Pengujian Robot & Dokumentasi Robot .....	64
Lampiran 2. Publikasi Jurnal.....	67

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Fenomena kebakaran masih menjadi salah satu ancaman serius di Indonesia, baik di kawasan perkotaan maupun di wilayah hutan dan lahan. Dampaknya tidak hanya menimbulkan korban jiwa, tetapi juga kerugian ekonomi serta degradasi lingkungan dalam skala besar. Data BPBD DKI Jakarta mencatat pada tahun 2024 terdapat 1.888 kejadian kebakaran dan lebih dari 1.100 kasus di antaranya dipicu oleh korsleting listrik [1].

Pada tingkat nasional, Pusat Informasi Kriminal Nasional (Pusiknas) Polri melaporkan 935 insiden kebakaran hingga Oktober 2024, dengan 75,29% di antaranya menyebabkan kerusakan signifikan pada rumah penduduk [2]. Sementara itu, BNPB melaporkan 629 kasus kebakaran hutan dan lahan sepanjang 2024, dengan dampak terbesar terjadi di Sumatera Selatan dengan luas lahan terdampak mencapai 9.697 hektare. Data tersebut memperlihatkan bahwa mitigasi kebakaran memerlukan pendekatan yang lebih responsif, presisi tinggi, dan mampu beroperasi pada situasi berbahaya yang sulit dijangkau oleh personel pemadam konvensional.

Seiring perkembangan teknologi, paradigma penanganan kebakaran telah bergeser dari sistem manual menuju sistem otomatis berbasis sensor, kendali cerdas, dan *decision-making* berbantuan algoritma [3]. Robot pemadam api merupakan salah satu bentuk inovasi strategis, karena memiliki kemampuan beroperasi secara otonom pada wilayah berisiko tinggi, seperti gedung bertingkat, area industri bahan kimia, dan ruang sempit yang membahayakan keselamatan manusia. Salah satu aspek krusial pada navigasi robot pemadam api adalah *path planning*, yang menentukan jalur optimal berdasarkan waktu tempuh, jarak, dan kondisi spasial lingkungan [4], [5].

Berbagai metode *path planning* telah diteliti dan dikembangkan, di antaranya algoritma genetika [6], algoritma A\* (*A-star*) yang mengombinasikan biaya lintasan yang telah ditempuh (*g-cost*) dan estimasi jarak ke tujuan (*h-cost*) melalui fungsi heuristik untuk memperoleh jalur yang optimal dan efisien [7], metode berbasis potensial [8], serta Triangulasi Delaunay untuk pemetaan spasial.

Namun, sebagian besar studi sebelumnya masih berfokus pada simulasi atau implementasi terbatas tanpa integrasi penuh dengan sistem sensor dan aktuator robot. Misalnya, Jijesh merancang robot pemadam api berbasis ESP32 namun belum menerapkan algoritma pencarian jalur otomatis [3], sementara Ranaweera mengembangkan *shortest path planning* berbasis PSO [4], tetapi hanya dievaluasi melalui simulasi matematika tanpa pengujian perangkat fisik. Celah penelitian ini menunjukkan urgensi untuk menghadirkan pendekatan *path planning* yang terintegrasi dengan sistem sensor dan aktuator secara nyata pada robot fisik agar dapat divalidasi dalam situasi laboratoris yang representatif.

Penelitian ini mengusulkan implementasi algoritma djikstra yang diadaptasikan secara khusus untuk dijalankan secara efisien pada mikrokontroler ESP32 yang memiliki keterbatasan memori dan kebutuhan pemrosesan *real-time*. Selain itu, penelitian ini mengembangkan mekanisme pembentukan graf navigasi semi-otomatis berdasarkan hasil pelacakan awal robot dalam arena uji, serta merancang metode translasi keluaran jalur terpendek menjadi perintah gerak aktual melalui integrasi sensor Sharp GP2Y0A21, fotodioda, dan sensor *flame*. Implementasi pada robot fisik di arena labirin berukuran laboratorium menjadi pembeda penting dibanding penelitian terdahulu yang masih berorientasi pada simulasi dan belum mencakup integrasi kontrol aktuator serta validasi eksperimental.

Dengan demikian, penelitian ini memberikan kontribusi nyata terhadap pengembangan navigasi robot pemadam api otonom, meliputi optimasi implementasi algoritma djikstra pada platform *embedded*, integrasi multi-sensor untuk koreksi lintasan dinamis, serta pembuktian eksperimental melalui pengujian berulang. Lingkup pengujian berfokus pada arena statis berbentuk labirin sebagai tahap awal sebelum dikembangkan pada lingkungan dinamis dan kondisi kebakaran sesungguhnya.

## 1.2 Tujuan

Adapun tujuan yang diharapkan pada penelitian yang dilakukan adalah sebagai berikut:

1. Untuk menemukan jalur terpendek yang harus diambil oleh robot dari titik awal ke tujuan.
2. Untuk mengoptimalkan waktu mencapai target.
3. Mengimplementasikan Algoritma Dijkstra pada robot pemandam api.

### **1.3 Manfaat Penelitian**

Penelitian dengan menerapkan algoritma djikstra, robot pemadam api dapat dengan cepat menentukan jalur terpendek menuju sumber api. Kecepatan respon yang lebih baik dapat membantu dalam meminimalkan kerugian dan risiko yang terkait dengan kebakaran.

### **1.4 Rumusan Masalah**

Rumusan masalah yang akan dibahas dalam penelitian tugas akhir ini adalah:

1. Bagaimana menemukan jalur terpendek antara titik awal sampai ke tujuan?
2. Bagaimana mengoptimalkan waktu mencapai target.
3. Bagaimana cara mengimplementasikan Algoritma Dijkstra pada robot pemandam api.

### **1.5 Batasan Masalah**

Hal yang menjadi batasan masalah dalam penelitian tugas akhir ini adalah:

1. Penelitian menggunakan prototipe mobile robot
2. Simulasi robot menggunakan labirin persegi

### **1.6 Hipotesis**

Hipotesis dalam penelitian ini menyatakan bahwa penerapan algoritma djikstra pada robot pemadam api akan memberikan dampak positif dalam meningkatkan efisiensi operasional. Dengan mengimplementasikan algoritma djikstra, robot diperkirakan mampu menemukan jalur terpendek secara optimal menuju titik api, sehingga dapat mengurangi waktu respons dan meminimalkan penggunaan sumber daya seperti energi.

Hipotesis ini juga menyiratkan bahwa efisiensi pergerakan robot akan meningkat dan waktu yang dibutuhkan untuk mencapai sumber kebakaran akan lebih singkat. Selain itu, algoritma djikstra diharapkan mampu mendukung adaptasi dinamis terhadap perubahan lingkungan serta mengoptimalkan rute pergerakan secara *real-time*. Efektivitas komunikasi jarak jauh pun diharapkan mengalami peningkatan, sehingga proses pemantauan dan pengendalian dapat dilakukan dengan lebih efisien dari jarak jauh.

Secara keseluruhan, implementasi algoritma djikstra dalam sistem robot pemadam api diyakini akan menghasilkan peningkatan kinerja yang signifikan, mempercepat respons terhadap situasi kebakaran dan membantu mengurangi dampak negatif yang mungkin terjadi.

## **1.7 Sistematika Penulisan**

Penulisan tesis ini disusun menurut sistematika secara runtut sebagai berikut:

### **BAB I. PENDAHULUAN**

Pada bab ini menjelaskan tentang latar belakang, tujuan, manfaat, rumusan masalah, batasan masalah, hipotesis, dan sistematika penulisan. Uraian di atas dimaksudkan untuk menguraikan hal-hal yang bersifat menjelaskan hal yang akan diteliti.

### **BAB II. TINJAUAN PUSTAKA**

Pada bab ini memaparkan landasan teori terhadap pengertian umum atau hal teoritis untuk menjelaskan pokok bahasan pada penelitian yang dilakukan.

### **BAB III. METODE PENELITIAN**

Dalam bab ini menguraikan tata cara atau teknik yang digunakan dalam mengumpulkan data parameter objek penelitian, serta pengumpulan data penelitian untuk menunjang keberlangsungan penelitian.

### **BAB IV. HASIL DAN PEMBAHASAN**

Pada bab ini berisi mengenai hasil pengujian dan pembahasan terhadap hasil penelitian yang diperoleh.

### **BAB V. KESIMPULAN DAN SARAN**

Pada bab ini berisi tentang kesimpulan dari hasil penelitian serta saran-saran untuk penelitian selanjutnya.

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN**

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Pengertian Robot

Menurut KBBI (Kamus Besar Bahasa Indonesia), robot alat berupa orang-orangan dan sebagainya yang dapat bergerak (berbuat seperti manusia) yang dikendalikan oleh mesin [6]. Robot adalah alat otomatis yang dapat diprogram untuk melakukan tugas-tugas berulang tanpa bantuan manusia [7].

Menurut definisi dari kamus Meriam-Webster, robot adalah mesin yang terlihat seperti manusia dan melakukan berbagai tindakan yang kompleks dari manusia seperti berjalan atau berbicara, atau suatu peralatan yang bekerja secara otomatis. Robot biasanya diprogram untuk melakukan pekerjaan berulang kali dan memiliki mekanisme yang dipandu oleh kontrol otomatis. Sedangkan robotika adalah cabang teknologi yang berkaitan dengan desain, konstruksi, operasi, dan aplikasi dari robot. Robotika merupakan cabang ilmu pengetahuan yang mempelajari tentang robot. Cabang ilmu tersebut mencakup desain mesin robot, elektronika, pengontrolan, pemrograman komputer, kecerdasan buatan, dan lain sebagainya.

Terdapat pendapat lain mengenai definisi robot, seperti diutarakan oleh *Robotic Institute of America* sebuah institusi robot yang terdapat 18 di Universitas Carnegie Mellon di kota Pittsburgh, negara bagian Pensilvania, Amerika Serikat, pada tahun 1979 mendefinisikan secara formal yang dimaksud dengan robot adalah "manipulator multi fungsi dan dapat diprogram ulang yang dirancang untuk menggerakkan material, alat, atau perangkat khusus melalui sejumlah gerakan terprogram untuk melakukan aktifitas tertentu".

Robot merupakan suatu entitas mekanik atau elektronik yang dapat diatur oleh program, dapat bergerak dan sering kali dirancang untuk melakukan tugas-tugas tertentu. Robot dapat berupa sistem sederhana yang digunakan untuk tugas-tugas repetitif hingga sistem kompleks yang dilengkapi dengan kecerdasan buatan dan kemampuan untuk berinteraksi dengan lingkungan sekitarnya. Umumnya, robot memiliki beberapa komponen utama, seperti sensor untuk mendeteksi lingkungan, aktuator untuk melakukan tindakan fisik, dan otak atau unit

pemrosesan untuk mengambil keputusan berdasarkan data yang diterima [8]. Terdapat empat karakteristik dasar yang harus dimiliki oleh setiap robot modern. **Gambar 2.1** merupakan bagan karakteristik pada robot. Karakteristik dasar tersebut adalah sebagai berikut.

1. Memiliki sensor.

Sensor merupakan peralatan yang berguna untuk mengukur ataupun merasakan sesuatu pada lingkungan di luar robot, layaknya indera pada makhluk hidup, dan memberi laporan hasilnya kepada robot. Dengan adanya sensor, robot bisa memiliki suatu pertimbangan dalam mengambil keputusan. Contoh dari sensor adalah sensor cahaya untuk mendeteksi adanya cahaya dan temperatur sensor untuk mengukur suhu.

2. Memiliki sistem kecerdasan (Kontrol).

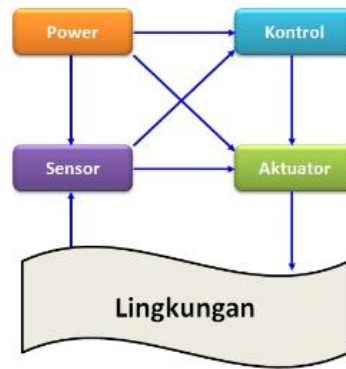
Sistem kecerdasan bekerja dengan memproses data masukan berupa keadaan ataupun kejadian yang sedang terjadi dari luar lingkungan. Selanjutnya sistem menghasilkan keluaran berupa instruksi ataupun keputusan pada robot untuk melakukan suatu tindakan tertentu. Sistem ini secara umum memiliki prinsip kerja seperti otak pada makhluk hidup, yang berfungsi untuk berpikir dan memutuskan tindakan apa yang perlu diambil pada suatu waktu tertentu.

3. Memiliki peralatan mekanik (Aktuator).

Peralatan mekanik berfungsi untuk membuat robot dapat melakukan suatu tindakan tertentu dan berinteraksi dengan lingkungannya. Contohnya seperti adanya roda bermotor untuk bergerak, lengan untuk mengambil objek, dan lain-lain.

4. Memiliki sumber daya (*Power*).

Sebagaimana pada organisme kehidupan yang membutuhkan makanan untuk hidup, robot juga memerlukan sumber tenaga untuk menggerakkan komponen elektrik dan mekanika yang terpasang. Sumber energi pada robot mencakup penyedia tenaga listrik seperti baterai, dan sistem pengatur transmisi yang bertugas mengonversi tenaga listrik sesuai kebutuhan setiap komponen [9].

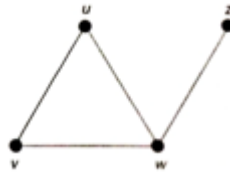


**Gambar 2. 1** Karakteristik Robot

Robot digunakan dalam berbagai konteks, termasuk industri manufaktur, layanan kesehatan, eksplorasi luar angkasa, militer, dan rumah tangga. Penggunaan robot telah memberikan kontribusi signifikan dalam meningkatkan efisiensi, keamanan, dan kemampuan manusia untuk menangani tugas-tugas yang sulit atau berbahaya [10].

## 2.2 Graf

Graf adalah struktur matematika yang terdiri dari dua komponen utama yaitu simpul (*vertex*) dan sisi (*edge*). Graf digunakan untuk merepresentasikan hubungan atau koneksi antara objek-objek tertentu [11]. Sebuah diagram graf sederhana (*simple graph*)  $G$  dibentuk dari sebuah himpunan  $V(G)$  yang memuat elemen-elemen yang disebut verteks (atau simpul [*node*] atau titik [*point*]) dalam jumlah yang terhingga, dan sebuah himpunan terhingga  $E(G)$  yang memuat pasangan-pasangan elemen  $V(G)$  [yaitu, pasangan-pasangan simpul/titik yang ada di dalam  $V(G)$ ] yang berbeda dan tak berurutan, disebut rusuk (*edge*) (*lines* atau garis).  $V(G)$  didefinisikan sebagai himpunan simpul (*vertex-set*) dan  $E(G)$  sebagai himpunan rusuk (*edge-set*) dari  $G$ . Sebuah rusuk  $\{v, w\}$  dikatakan menghubungkan (*join*) simpul  $v$  dan  $w$ , dan biasanya dapat dituliskan dalam bentuk singkat  $vw$ . Sebagai contoh, **Gambar 2.2** memperlihatkan sebuah diagram graf sederhana  $G$  yang memiliki himpunan simpul  $V(G) = \{u, v, w\}$  dan himpunan rusuk  $E(G)$  yang terdiri dari garis- garis  $uv, uw, vw$ , dan  $wz$ .



**Gambar 2. 2** Graf Sederhana

Pada sebuah diagram graf sederhana, terdapat paling tidak satu buah rusuk yang menggabungkan sepasang simpul. Akan tetapi, banyak fitur dari diagram graf sederhana yang berlaku pula pada bangun-bangun yang lebih kompleks, di mana dua buah simpul boleh jadi dihubungkan oleh lebih dari satu rusuk; rusuk-rusuk penghubung semacam ini disebut rusuk berganda (*multiple edges*). Selain itu, dapat juga menghilangkan batasan bahwa sebuah rusuk harus menggabungkan dua buah simpul yang berbeda, dan membolehkan adanya *loop* yaitu, rusuk yang menghubungkan sebuah simpul dengan dirinya (simpul itu) sendiri. Bangun yang terbentuk, dengan diperbolehkannya penggunaan rusuk berganda dan loop, disebut diagram graf umum (*general graph*). Setiap graf sederhana adalah sebuah graf, namun tidak setiap graf adalah sebuah graf sederhana [12] seperti yang terlihat pada **Gambar 2.3**.



**Gambar 2. 3** Graf umum

Oleh karena itu, sebuah graf adalah representasi dari gabungan titik-titik dan bagaimana titik-titik tersebut bergabung, dan properti metrik apapun tidak relevan.

### 2.3 *Shortest Path*

Jalur terpendek atau *shortest path* adalah rute atau lintasan yang membutuhkan waktu atau jarak paling sedikit untuk mencapai tujuan dari suatu lokasi atau titik awal [13]. Konsep ini banyak digunakan dalam berbagai konteks, termasuk dalam pemrograman linier, graf teori, dan masalah optimasi lainnya.

Dalam konteks graf teori, *shortest path* sering kali diukur dalam jumlah edge atau bobot yang mewakili jarak atau biaya antar simpul (*node*) di graf. Algoritma terkenal seperti Algoritma Dijkstra dan Algoritma Bellman-Ford digunakan untuk menemukan jalur terpendek dalam graf.

Pada dasarnya, tujuan dari mencari jalur terpendek adalah untuk mengoptimalkan suatu parameter tertentu, seperti waktu, biaya, atau jarak, dalam perjalanan dari satu titik ke titik lainnya. Algoritma-algoritma ini memiliki berbagai kegunaan praktis, seperti perencanaan rute dalam sistem navigasi, manajemen jaringan komputer, dan logistik transportasi [14].

Dalam konteks robot pemadam api, masalah jalur terpendek (*shortest path*) sangat penting untuk menemukan rute tercepat menuju lokasi kebakaran atau evakuasi [15]. Robot pemadam api biasanya beroperasi di lingkungan yang bisa berupa bangunan, area terbuka, atau jaringan pipa, di mana jalur yang tersedia antara titik-titik (*node*) dipengaruhi oleh kondisi seperti panjang jarak, waktu tempuh, hambatan fisik, atau bahkan kondisi berbahaya seperti api dan asap.

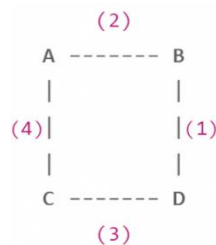
## 2.4 Algoritma Dijkstra

Algoritma Dijkstra adalah salah satu algoritma pencarian jalur terpendek yang ditemukan oleh Edsger W. Dijkstra, seorang ilmuwan komputer asal Belanda, pada tahun 1956 dan diterbitkan secara resmi pada tahun 1959. Algoritma ini digunakan untuk menemukan jalur terpendek dari satu titik (*node*) ke titik lainnya dalam graf berbobot dengan bobot non-negatif.

Edsger Dijkstra menemukan algoritma ini saat bekerja di *Mathematisch Centrum di Amsterdam*, Belanda. Menurut pengakuannya, ide algoritma ini muncul dalam waktu sekitar 20 menit ketika ia sedang mencari cara untuk menentukan jalur terpendek dalam jaringan transportasi. Pada awalnya, Dijkstra merancang algoritma ini untuk menemukan jalur terpendek dalam sistem transportasi, seperti rute jalan antara dua kota. Namun, seiring perkembangan teknologi, algoritma ini menjadi sangat berguna dalam berbagai bidang, termasuk jaringan komputer, perutean data, dan kecerdasan buatan. Algoritma Dijkstra bekerja dengan prinsip greedy, yaitu pada setiap langkah memilih solusi lokal terbaik (simpul dengan jarak terkecil) dengan harapan solusi global terbaik akan tercapai.

Algoritma Dijkstra adalah algoritma yang digunakan untuk menemukan jalur terpendek dari satu titik asal (*source*) ke semua titik lainnya dalam sebuah graf berbobot. Algoritma ini sangat populer dalam pemrograman graf karena efisiensinya dalam menyelesaikan masalah rute terpendek, terutama di bidang seperti jaringan komputer, transportasi, dan sistem navigasi. Tujuan utama dari algoritma ini adalah untuk menemukan jalur terpendek dari satu titik awal ke semua titik lainnya dalam graf [16]. **Gambar 2.4** merupakan ilustrasi Graf Algoritma Dijkstra beserta penjelasannya bisa di lihat pada **Tabel 2.1**. Konsep utamanya adalah:

1. Graf (*Graph*): Kumpulan simpul (*nodes/vertices*) dan sisi (*edges*).
2. Simpul (*Vertex/V*): Titik dalam graf, mewakili objek atau lokasi.
3. Sisi (*Edge/E*): Jalur penghubung antara dua simpul, memiliki bobot tertentu.
4. Bobot (*Weight*): Nilai jarak atau biaya yang diasosiasikan dengan suatu sisi dalam graf.
5. Sumber (*Source*): Titik awal dari mana jalur terpendek dicari.
6. Fungsi jarak ( $d[v]$ ): Menyimpan jarak minimum yang diketahui dari sumber ke simpul  $v$ .



**Gambar 2. 4** Graf Algoritma Dijkstra

**Tabel 2. 1** Penjelasan Graf Algoritma Dijkstra

Elemen	Penjelasan
Simpul	A, B, C, D adalah simpul ( <i>vertex/node</i> )
Sisi	Garis penghubung antar simpul: A-B, A-C, B-D, C-D
Bobot	Angka dalam tanda kurung pada sisi: misal sisi A-B memiliki bobot 2
Sumber	Jika mulai dari A, maka A adalah simpul sumber ( <i>source node</i> )
Jalur	Misalnya $A \rightarrow C \rightarrow D$ adalah jalur dengan total bobot = $4 (A-C) + 3 (C-D) = 7$
Jarak	Jarak dari A ke D bisa melalui beberapa jalur, algoritma akan cari yang paling minimum

Algoritma Dijkstra bekerja pada graf berbobot dan berarah (*directed weighted graph*). Algoritma ini hanya bekerja pada graf yang memiliki bobot tepi non-negatif. Berikut adalah pendekatan matematisnya:

### 1. Representasi Graf

Graf berbobot direpresentasikan sebagai  $G = (V, E)$  di mana:

- a.  $V$  adalah himpunan simpul (*vertices*).
- b.  $E$  adalah himpunan sisi (*edges*) dengan bobot non-negatif  $w(u, v)$  yang menunjukkan biaya (berat) antara simpul  $u$  dan  $v$ .

Bobot diberikan dalam bentuk matriks bobot (*adjacency matrix*) mengacu pada representasi graf dengan bobot antar simpul atau daftar ketetanggaan (*adjacency list*) daftar yang menyimpan daftar tetangga dari setiap simpul dalam graf.

### 2. Inisialisasi

Untuk setiap simpul  $v \in V$ , didefinisikan:

- a. Jarak awal dari sumber:  
 $d(s) = 0$   
 untuk simpul sumber  $s$ , dan  $d(v) = \infty$   
 infinity untuk semua simpul lainnya.
- b. Himpunan yang belum dikunjungi:  $Q = V$ , yaitu semua simpul dalam graf.

### 3. Iterasi Proses

Selama  $Q$  tidak kosong:

1. Pilih simpul  $u$  dengan nilai  $d(u)$  terkecil dari  $Q$ .
2. Hapus  $u$  dari  $Q$ .
3. Perbarui jarak untuk setiap tetangga  $v$  dari  $u$  yang masih ada di  $Q$  dengan rumus:  $d(v) = \min(d(v), d(u) + w(u, v))$   
 Jika  $d(u) + w(u, v)$  lebih kecil dari nilai sebelumnya.

di mana:

1.  $d(v)$  adalah jarak terpendek ke simpul  $v$ .
2.  $w(v, u)$  adalah bobot sisi antara  $v$  dan  $u$ .

Setelah semua simpul telah dikunjungi, maka  $d(v)$  menyimpan jarak terpendek dari simpul sumber ke simpul  $v$ . Setelah semua simpul telah dikunjungi, maka  $d(v)$  menyimpan jarak terpendek dari simpul sumber ke simpul  $v$ .

#### 4. *Termination* (Hasil Akhir)

Setelah semua simpul telah dikunjungi, maka  $d(v)$  untuk setiap  $v$  berisi jarak terpendek dari sumber ke simpul tersebut.

Algoritma Dijkstra digunakan untuk membantu robot pemadam api menemukan jalur terpendek menuju lokasi api dalam lingkungan yang direpresentasikan sebagai graf. Jalur terpendek memungkinkan robot untuk mencapai tujuan dengan efisiensi waktu dan energi, yang sangat penting dalam situasi darurat [17]. Berikut ini langkah-langkah mengerjakan algoritma djikstra:

1. Inisialisasi
  - a. Tetapkan jarak simpul awal ke 0, dan semua simpul lain ke  $\infty$ .
  - b. Gunakan struktur data untuk menyimpan jarak dan simpul yang belum dikunjungi.
2. Pilih Simpul dengan Jarak Minimum
  - a. Pilih simpul dengan jarak terkecil yang belum dikunjungi.
  - b. Tandai simpul tersebut sebagai dikunjungi.
3. Perbarui Jarak Tetangga
  - a. Untuk setiap tetangga simpul yang diproses, perbarui jaraknya jika ditemukan jarak lebih pendek.
4. Ulangi Proses
  - a. Ulangi langkah 2 dan 3 hingga semua simpul telah dikunjungi atau tidak ada simpul yang bisa dijangkau.
5. Hasil Akhir
  - a. Diperoleh jarak terpendek dari simpul awal ke semua simpul lainnya.
  - b. Jika diperlukan, rekonstruksi jalur menggunakan simpul sebelumnya.

## 2.5 Penelitian Terdahulu

Dalam pelaksanaan penelitian ini, dilakukan penelusuran terhadap sejumlah penelitian terdahulu untuk memperoleh referensi yang relevan mengenai implementasi algoritma djikstra pada robot pemadam api. Berikut disajikan uraian beberapa penelitian terdahulu yang relevan dan memiliki keterkaitan dengan topik penelitian ini. Tinjauan terhadap penelitian-penelitian tersebut digunakan sebagai

dasar untuk memahami perkembangan metode yang telah ada serta sebagai acuan dalam menentukan posisi dan kontribusi penelitian yang dilakukan.

Dalam penelitian yang dilakukan oleh Ren *et al.* [3] berjudul “*Path Planning of Mobile Robots Based on Improved A Algorithm*” , diusulkan pengembangan algoritma A\* yang telah ditingkatkan guna meningkatkan efisiensi perencanaan jalur dan keamanan navigasi robot mobile. Penelitian tersebut dilakukan secara simulatif dan menunjukkan kecepatan respons perencanaan jalur yang tinggi, namun belum diimplementasikan pada perangkat keras nyata seperti mikrokontroler.

Selanjutnya, Jijesh *et al.* [4] dalam penelitian “*Design and Implementation of Automated Fire Fighting and Rescuing Robot*” merancang robot pemadam api otomatis yang dikendalikan oleh manusia dan berbasis mikrokontroler ESP32. Penelitian ini lebih menitikberatkan pada upaya penyelamatan korban kebakaran secara dini, namun belum mengimplementasikan algoritma pencarian jalur otomatis seperti Dijkstra atau algoritma shortest path lainnya.

Sementara itu, Ranaweera *et al.* [5] dalam penelitian berjudul “*A Shortest Path Planning Algorithm for PSO-Based Firefighting Robots*” (2018) mengembangkan metode perencanaan jalur terpendek menggunakan algoritma Particle Swarm Optimization (PSO) pada robot pemadam kebakaran. Penelitian ini difokuskan pada pencarian jalur optimal dalam skenario multi-robot dan dilakukan sepenuhnya dalam lingkungan simulasi menggunakan perangkat lunak *Mathematica*. Meskipun pendekatan PSO menunjukkan potensi yang baik dalam optimasi global, penelitian tersebut belum menguji implementasi algoritma pada perangkat keras nyata sehingga belum merepresentasikan performa sistem pada kondisi fisik sebenarnya.

Berdasarkan tinjauan terhadap penelitian-penelitian terdahulu tersebut, penelitian ini menawarkan pendekatan yang berbeda dengan mengimplementasikan algoritma Dijkstra secara langsung pada mikrokontroler ESP32 yang dikenal hemat energi dan mendukung integrasi Internet of Things (IoT). Selain itu, sistem dilengkapi dengan sensor *flame* untuk deteksi api dan sensor jarak Sharp

GP2Y0A21 sebagai pendukung navigasi serta penghindaran rintangan. Penelitian ini menekankan penerapan di dunia nyata dengan sistem yang ringan dan efisien, tidak hanya terbatas pada simulasi. Berbeda dengan penelitian Ren *et al.* [2] yang berfokus pada pengembangan algoritma A\* dalam lingkungan simulasi, serta penelitian Jijesh *et al.* [3] yang menitikberatkan pada pengendalian manual tanpa perencanaan jalur otomatis, penelitian ini lebih menekankan pada navigasi mandiri dan perencanaan jalur terpendek secara *real-time*. Selain itu, pendekatan yang diusulkan dinilai lebih praktis dibandingkan penelitian Ranaweera *et al.* [4] yang masih terbatas pada simulasi matematis tanpa implementasi perangkat keras.

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Penelitian ini dilakukan pada periode bulan Agustus hingga Januari 2026 di Laboratorium Teknik Elektronika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

#### **3.2 Alat dan Bahan Penelitian**

Guna menyelesaikan penelitian dengan baik, perlu melengkapi alat dan komponen menjadi satu kesatuan dengan baik. Alat dan komponen yang digunakan sebagai berikut.

1. Sensor Sharp GP fungsi sensor ini akan digunakan untuk mendeteksi jarak antara robot dan objek di depannya. Informasi ini bisa digunakan untuk navigasi robot agar tidak menabrak objek.
2. Sensor fotodiode digunakan untuk mendeteksi cahaya, bisa berfungsi sebagai deteksi api atau untuk navigasi berdasarkan intensitas cahaya.
3. Sensor *flame* atau sensor api berfungsi untuk mendeteksi adanya nyala api. Ini adalah sensor utama yang mendeteksi api untuk dipadamkan. Sensor *flame* yang digunakan disini adalah sensor *flame* 360°, sensor ini mampu mendeteksi keberadaan api dari segala arah (360 derajat), sehingga robot dapat menentukan lokasi api secara lebih akurat tanpa perlu memutar tubuhnya terlebih dahulu.
4. Mikrokontroler ESP32 berfungsi sebagai otak robot yang akan mengontrol semua sensor, aktuator, dan logika pemadam api.
5. Driver MOSFET digunakan untuk mengontrol daya yang dikirimkan ke motor dan kipas dengan menggunakan sinyal dari ESP32.
6. Motor DC berfungsi menggerakkan roda robot agar dapat bergerak ke arah yang ditentukan.
7. LCD OLED untuk menampilkan informasi seperti status sensor, kondisi robot, atau data lainnya.

8. Driver kipas dan kipas berfungsi untuk menggerakkan kipas untuk memadamkan api setelah terdeteksi oleh sensor *flame*.
9. Baterai LiPo (*Lithium Polymer*) 3-cell (3S) berfungsi sebagai sumber daya utama pada robot yang membutuhkan tegangan dan arus tinggi untuk pengoperasian komponen seperti motor dc dan mikrokontroler.
10. *Wall Following* terbuat dari triplek yang membantu robot bergerak otomatis di sepanjang dinding atau batas area, baik di koridor, lorong, atau ruangan tertutup. Serta area tempat kebakaran dengan 4 ruangan.

### 3.3 Prosedur Penelitian

Prosedur penelitian ini diawali dengan tahap studi pustaka, yaitu mengumpulkan dan mempelajari literatur berupa jurnal ilmiah, buku, serta referensi lain yang berkaitan dengan robot pemadam api dan algoritma djikstra dalam penentuan jalur terpendek. Tahap ini bertujuan untuk memperoleh dasar teori yang kuat sebagai acuan dalam perancangan dan pengembangan sistem yang akan dibuat.

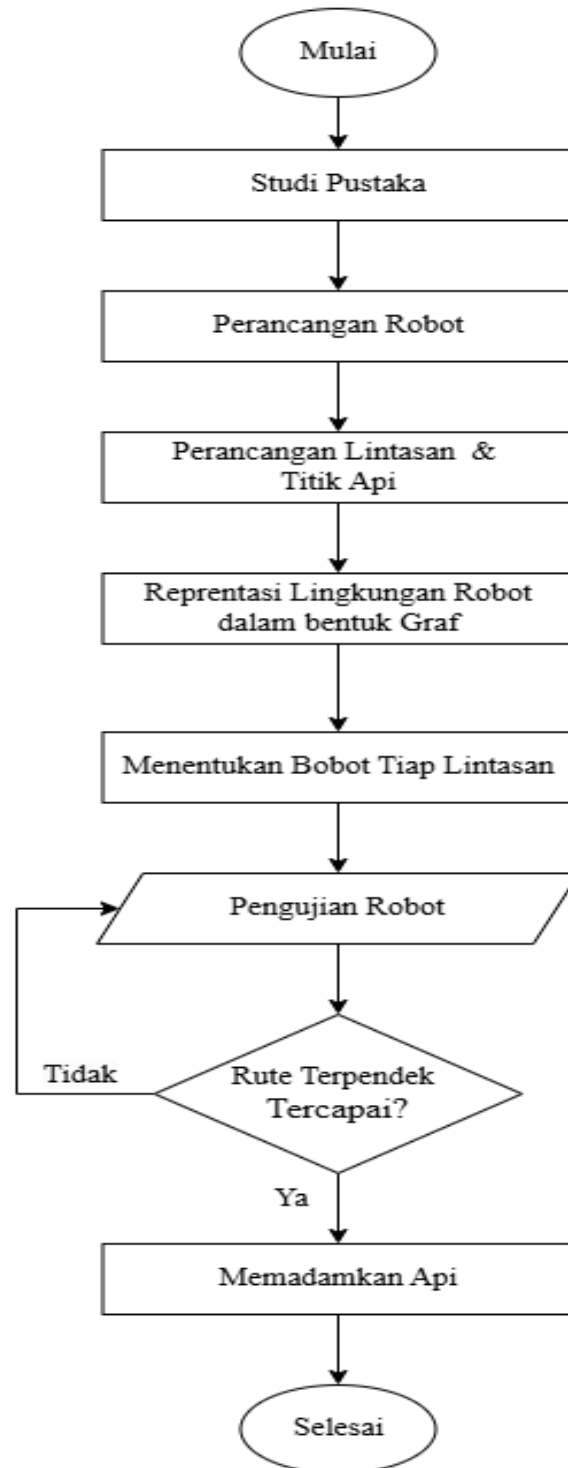
Tahap selanjutnya adalah perancangan robot pemadam api, yang meliputi perancangan mekanik, sistem elektronik, serta sistem kendali robot. Setelah robot dirancang, dilakukan perancangan lintasan dan penentuan posisi titik api sebagai area pengujian. Pada tahap ini juga dirancang metode *wall following* untuk membantu robot dalam bernavigasi mengikuti dinding lingkungan.

Lingkungan operasi robot kemudian direpresentasikan dalam bentuk graf, di mana setiap node merepresentasikan lokasi tertentu dan setiap *edge* merepresentasikan hubungan antar lokasi berupa jarak atau lintasan. Berdasarkan representasi graf tersebut, selanjutnya ditentukan bobot pada setiap lintasan yang akan digunakan dalam proses perhitungan rute terpendek.

Tahap berikutnya adalah pengujian robot untuk menentukan rute terpendek menuju titik api menggunakan algoritma djikstra. Pengujian dilakukan secara berulang hingga rute terpendek berhasil dicapai sesuai dengan kriteria yang telah ditentukan. Apabila rute terpendek belum tercapai, maka proses pengujian akan diulangi kembali.

Setelah rute terpendek berhasil diperoleh, robot diarahkan menuju titik api melalui lintasan tersebut dan melakukan proses pemadaman api. Prosedur

penelitian diakhiri setelah robot berhasil mencapai titik api dan memadamkan api sesuai dengan tujuan penelitian. Alur keseluruhan prosedur penelitian ini ditunjukkan pada **Gambar 3.1**.



**Gambar 3. 1** *Flowchart*

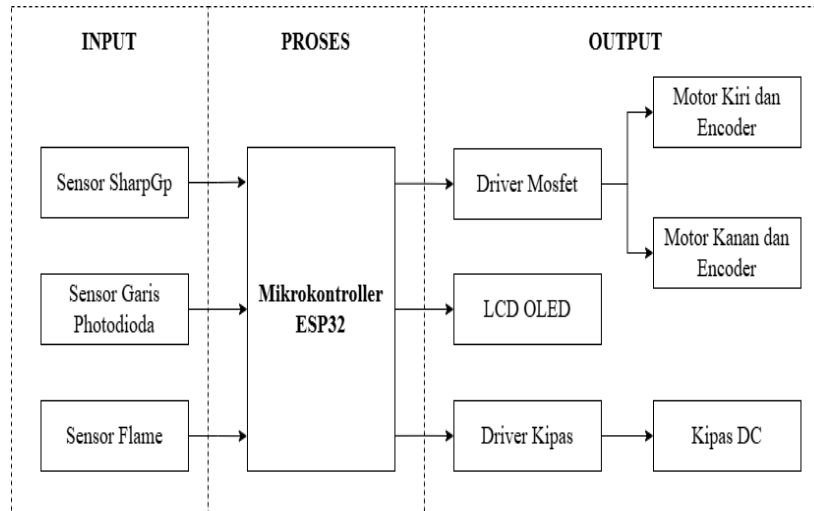
### 3.4 Perancangan Robot

Robot yang akan dirancang yaitu robot beroda dua yang dirancang untuk memadamkan api. Robot beroda merupakan robot yang dapat berpindah menggunakan roda bermotor untuk bergerak. Dibandingkan dengan robot jenis lain robot ini lebih mudah dibangun, dikontrol, dan digunakan pada bidang datar. Sensor yang digunakan yaitu sensor sharp GP2Y0A21 yang akan digunakan untuk mendeteksi jarak antara robot dan objek di depannya. Informasi ini bisa digunakan untuk navigasi robot agar tidak menabrak objek. Kemudian sensor fotodiode berfungsi sebagai deteksi api atau untuk navigasi berdasarkan intensitas cahaya. Kemudian terakhir sensor *flame* berfungsi untuk mendeteksi api untuk dipadamkan. Sensor *flame* yang digunakan disini adalah sensor *flame* 360° Sensor ini mampu mendeteksi keberadaan api dari segala arah (360 derajat), sehingga robot dapat menentukan lokasi api secara lebih akurat tanpa perlu memutar tubuhnya terlebih dahulu.

Robot ini memiliki sistem kecerdasan (*control*) berupa mikrokontroler ESP32 sebagai otak robot yang akan mengontrol semua sensor, aktuator, dan logika pemadam api. Pemilihan ESP32 disini karena memiliki kelebihan signifikan dibandingkan mikrokontroler lain, seperti Arduino Uno atau STM32, terutama karena konektivitas nirkabel bawaan berupa Wi-Fi dan Bluetooth yang memungkinkan komunikasi jarak jauh tanpa modul tambahan. Dengan prosesor dual-core 32-bit berkecepatan hingga 240 MHz, ESP32 unggul dalam multitasking dan mampu menangani proses kompleks, seperti pengolahan data sensor, kontrol motor, dan algoritma navigasi secara simultan. Selain itu, ESP32 memiliki kapasitas RAM yang besar (520 KB SRAM) dan penyimpanan flash yang memadai, memungkinkan penerapan algoritma yang lebih canggih, seperti Dijkstra atau *machine learning*, yang sulit dijalankan pada mikrokontroler dengan spesifikasi lebih rendah.

Kemudian Aktuator yang digunakan disini adalah yaitu motor DC yang berfungsi menggerakkan roda robot agar dapat bergerak ke arah yang ditentukan. Kemudian sumber daya (*power*) yang digunakan pada robot ini yaitu Baterai LiPo (*Lithium Polymer*) 3-cell (3S) berfungsi sebagai sumber daya utama pada robot yang membutuhkan tegangan dan arus tinggi untuk pengoperasian komponen

seperti motor dc dan mikrokontroler. Robot ini dilengkapi oleh LCD OLED untuk menampilkan informasi seperti status sensor, kondisi robot, atau data lainnya seperti pada **Gambar 3.2**.



**Gambar 3. 2** Blok diagram

**Tabel 3.1** memuat spesifikasi komponen yang digunakan pada robot dan arena pengujiannya. Robot ini terdiri dari dua buah Sensor Sharp GP berukuran  $29 \times 13 \times 13$  mm, satu set sensor fotodioda dengan ukuran  $20 \times 10 \times 8$  mm, serta satu set sensor *flame*  $360^\circ$  berukuran  $30 \times 30 \times 10$  mm. Sistem kendali robot menggunakan satu unit mikrokontroler ESP32 berukuran  $51 \times 25 \times 5$  mm yang terhubung dengan dua buah driver MOSFET berukuran  $20 \times 15 \times 10$  mm. Bagian penggerak robot terdiri dari dua buah motor DC yang dilengkapi roda berukuran  $\emptyset 65 \times 22$  mm. Untuk tampilan informasi, robot menggunakan satu buah LCD OLED berukuran  $27 \times 27 \times 4$  mm. Selain itu, terdapat satu driver kipas berukuran  $25 \times 20 \times 10$  mm dan satu unit kipas pemadam berukuran  $40 \times 25$  mm. Seluruh komponen mendapatkan pasokan daya dari baterai LiPo 3S 11.1V 450mAh dengan dimensi  $55 \times 30 \times 12$  mm. Komponen-komponen tersebut dipasang pada rangka robot berukuran  $180 \times 180 \times 170$  mm dan dihubungkan menggunakan delapan buah kabel jumper sepanjang 100 mm. Pengujian robot dilakukan di dalam arena labirin berukuran  $2000 \times 1600 \times 400$  mm yang menjadi tempat simulasi pergerakan dan pengujian robot secara keseluruhan.

**Tabel 3. 1** Spesifikasi Robot dan Arena Robot

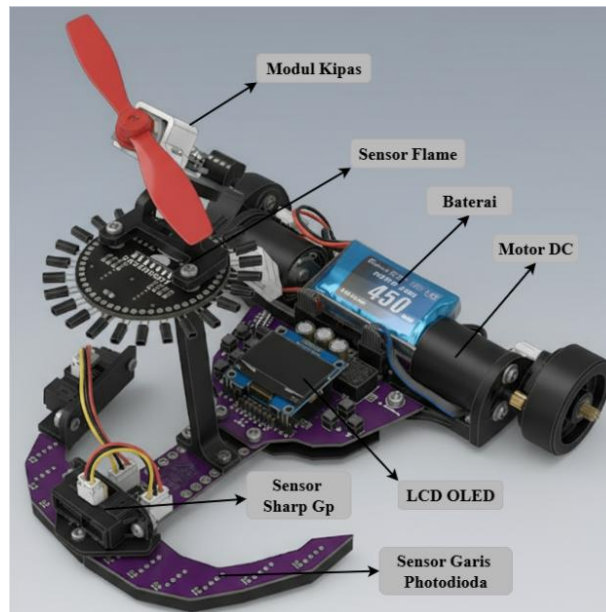
<b>Komponen</b>	<b>Jumlah</b>	<b>Dimensi (mm)</b>
Sensor <i>SharpGp</i>	2 Buah	$29 \times 13 \times 13$
Sensor <i>Fotodioda</i>	1 Set	$20 \times 10 \times 8$
Sensor <i>Flame 360°</i>	1 Set	$30 \times 30 \times 10$
Mikrokontroler ESP32	1 Buah	$51 \times 25 \times 5$
<i>Driver</i> MOSFET	2 Buah	$20 \times 15 \times 10$
Motor DC + Roda	2 Buah	$\varnothing 65 \times 22$
LCD OLED	1 Buah	$27 \times 27 \times 4$
<i>Driver</i> Kipas	1 Buah	$25 \times 20 \times 10$
Kipas Pemadam	1 Buah	$40 \times 25$
Baterai LiPo 3S 11.1V 450mAh	1 Buah	$55 \times 30 \times 12$
Rangka Robot	1 Buah	$180 \times 180 \times 170$
Kabel Jumper	8 Buah	100
Labirin Arena Robot	1 1 Buah	$2000 \times 1600 \times 400$

### 3.5 Desain Perangkat Keras

Desain perangkat keras pada robot pemadam api ini dilakukan untuk memastikan integrasi antara komponen input, proses, dan output dapat berjalan secara optimal sesuai dengan fungsi operasional robot [9]. Perancangan mekanik dan elektronik disusun dengan mempertimbangkan efisiensi ruang, stabilitas gerak, pusat gravitasi, serta kemudahan pemeliharaan sistem. Kerangka robot berfungsi sebagai struktur utama yang menopang seluruh komponen, termasuk sensor, aktuator, dan mikrokontroler, sekaligus menjaga keseimbangan robot saat melakukan manuver pada lintasan labirin.

Komponen sensor ditempatkan secara strategis untuk mengoptimalkan fungsi deteksi. Sensor Sharp GP2Y0A21 dipasang pada bagian depan robot untuk mendeteksi jarak terhadap objek penghalang, sementara sensor fotodioda ditempatkan di bawah untuk membaca garis arena sebagai panduan navigasi. Sensor *flame 360°* terpasang pada bagian atas untuk memungkinkan deteksi sumber api dari berbagai arah tanpa perlu memutar arah robot [10]. Integrasi tata letak ini dirancang agar seluruh input sensor dapat memberikan informasi lingkungan yang

tepat bagi sistem kendali. Bentuk rancangan fisik robot pemadam api ditunjukkan pada **Gambar 3.3**.

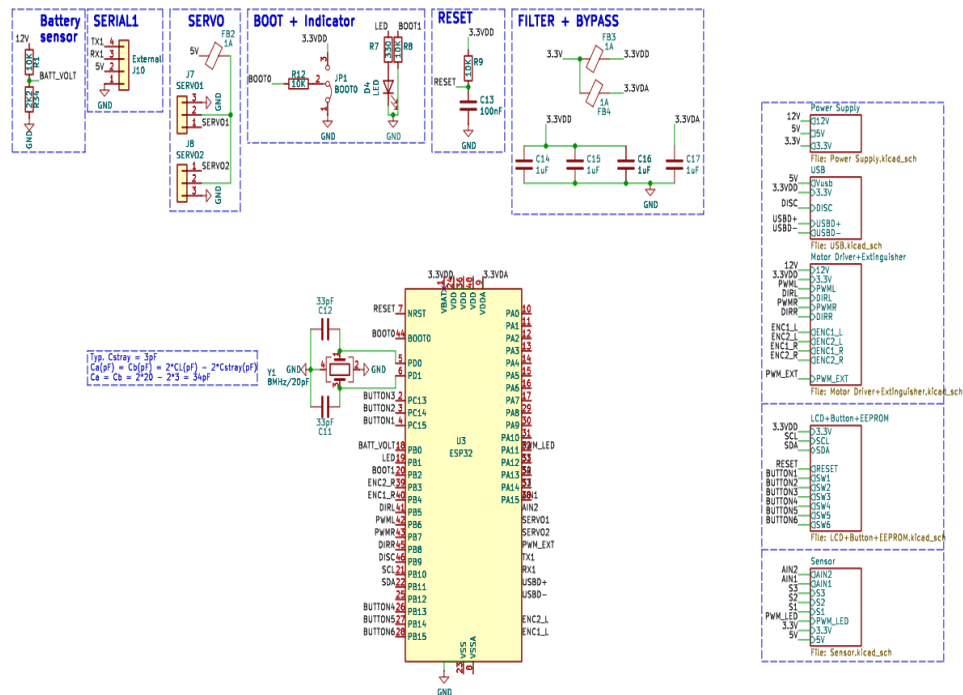


**Gambar 3.3** Perancangan Kerangka Robot Pemadam Api

Bagian proses dikendalikan oleh mikrokontroler ESP32, yang bertugas mengolah data sensor dan menjalankan algoritma navigasi berbasis djikstra. Pemilihan ESP32 didasarkan pada kemampuan pemrosesan *dual-core* 32-bit hingga 240 MHz, kapasitas memori besar (520 KB SRAM), dan konektivitas Wi-Fi/Bluetooth yang memberikan fleksibilitas ekspansi sistem. Output kendali diberikan kepada dua motor DC melalui driver MOSFET untuk menghasilkan gerakan diferensial (*differential steering*) sebagai dasar translasi jalur ke arah gerak robot. Modul kipas pemadam dikendalikan melalui driver khusus dan diaktifkan ketika sensor *flame* mendeteksi sumber api. Selain itu, LCD OLED digunakan sebagai perangkat antarmuka untuk menampilkan status sistem dan informasi sensor secara *real-time*.

Seluruh komponen mendapat suplai daya dari baterai Li-Po 3S 11,1 V yang mampu menyediakan arus tinggi untuk motor dan modul kendali. Dengan integrasi perangkat keras ini, sistem robot mampu mendeteksi api, menavigasi rintangan, dan melakukan pemadaman secara mandiri dengan dukungan algoritma djikstra secara

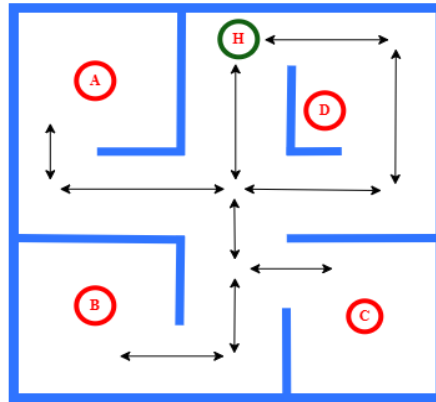
*real-time*. Rancangan fisik robot ditunjukkan pada **Gambar 3.3** sedangkan skematik rangkaian elektronik ditunjukkan pada **Gambar 3.4**.



**Gambar 3. 4** Skema Rangkaian Robot Pemadam Api

### 3.6 Perancangan Algoritma Dijkstra

Algoritma Dijkstra digunakan dalam penelitian ini untuk menentukan lintasan terpendek yang harus ditempuh robot pemadam api dari titik awal (*home*) menuju lokasi sumber api, kemudian kembali ke titik home setelah proses pemadaman selesai [8], [11]. Arena pengujian dirancang berbentuk labirin dan direpresentasikan sebagai sebuah graf berbobot yang terdiri atas simpul (*nodes*) sebagai titik navigasi serta sisi (*edges*) sebagai jalur penghubung antar simpul. Representasi graf ini memungkinkan proses perhitungan matematis terhadap jalur optimal berdasarkan pembobotan jarak aktual pada arena fisik. Konfigurasi arena percobaan robot pemadam api ditunjukkan pada **Gambar 3.5**.



**Gambar 3. 5** Labirin Arena Percobaan Robot

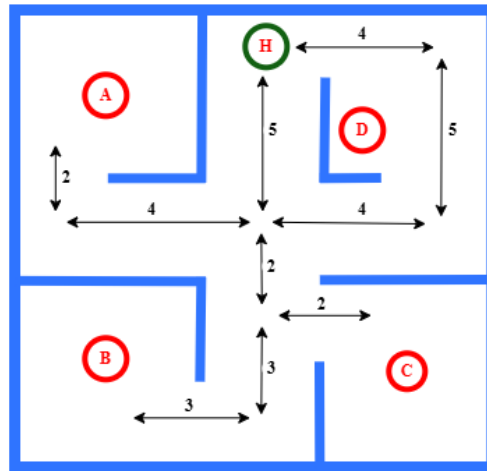
Arena uji memiliki satu simpul awal yaitu titik *home* (H), serta empat simpul target pemadaman api yang diberi label A, B, C, dan D. Jalur navigasi yang dapat dilalui robot ditandai dengan arah panah yang menggambarkan arah pergerakan yang mungkin dilakukan. Sebelum proses pencarian jalur menggunakan algoritma djikstra, robot terlebih dahulu melakukan proses pelacakan lintasan (*initial tracking*) untuk memetakan struktur arena dan menyimpan koordinat navigasi ke dalam basis data internal. Proses pemetaan dilakukan berdasarkan pembacaan sensor Sharp GP2Y0A21 sebagai sensor jarak dan sensor fotodioda sebagai pembaca garis arena dan berdasarkan jumlah putaran encoder yang terdapat pada motor dc. Informasi ini kemudian dikonversi menjadi struktur graf yang digunakan untuk perhitungan *shortest path*. Setiap simpul navigasi pada arena direpresentasikan sebagai himpunan sebagai berikut:

$$V = \{ H, A, B, C, D \} \dots\dots\dots (1)$$

Sementara sisi yang menghubungkan antar simpul direpresentasikan sebagai jalur berbobot sesuai dengan jarak lintasan sebenarnya. Nilai bobot diperoleh dari hasil pengukuran jarak linear lintasan robot berdasarkan satuan grid pada arena, sebagaimana ditunjukkan pada **Gambar 3.6** dan **Tabel 3.2** Representasi graf berbobot ini diformulasikan sebagai:

$$G = (V, E, W) \dots\dots\dots (2)$$

Di mana  $V$  adalah himpunan simpul navigasi,  $E$  adalah jalur yang tersedia, dan  $W$  merupakan bobot jarak antar simpul.



**Gambar 3. 6** Representasi graf berbobot

Pembobotan pada graf diperoleh dari hasil pengukuran jarak fisik antar simpul pada arena dan digunakan sebagai parameter optimasi lintasan pada algoritma djikstra. Nilai bobot antar simpul lengkap ditampilkan pada **Tabel 3.2**.

**Tabel 3. 2** Bobot Jalur Antar Simpul

Jalur	Bobot (Jarak)
$H \leftrightarrow A$	11
$H \leftrightarrow D$	4
$A \leftrightarrow B$	14
$B \leftrightarrow C$	8
$C \leftrightarrow D$	13

Proses penentuan jalur optimal dimulai dengan menginisialisasi simpul *home* (H) sebagai simpul sumber dengan jarak awal bernilai 0, sedangkan jarak awal simpul lainnya diatur bernilai tak hingga. Algoritma djikstra kemudian melakukan proses relaksasi bobot secara iteratif sampai jarak minimum menuju simpul tujuan diperoleh. Hasil perhitungan berupa urutan simpul jalur optimal, misalnya  $H \rightarrow D \rightarrow C$ , kemudian diterjemahkan ke dalam perintah gerak aktual seperti maju, belok kanan atau kiri, dan berhenti melalui modul kendali motor pada

mikrokontroler ESP32. Proses translasi jalur dikombinasikan dengan pembacaan sensor secara *real-time* untuk memastikan pergerakan fisik robot tetap sesuai jalur meskipun terdapat deviasi akibat slip roda atau gangguan permukaan lintasan.

Dengan pendekatan ini, algoritma djikstra tidak hanya menghasilkan optimasi matematis terhadap lintasan terpendek, tetapi juga menjamin eksekusi navigasi nyata secara konsisten dan adaptif. Integrasi algoritma dengan sistem sensor dan aktuator memungkinkan robot mencapai titik api secara efisien, memadamkannya, dan kembali ke titik *home* secara mandiri.

### 3.7 Pembentukan Graf Arena dan Penentuan Bobot

Pembentukan graf arena merupakan tahapan fundamental dalam proses perencanaan jalur robot, karena struktur navigasi lintasan perlu direpresentasikan dalam bentuk graf berbobot untuk mendukung perhitungan algoritma djikstra. Arena uji berukuran 2000 mm × 1600 mm × 400 mm dipetakan menjadi graf yang terdiri dari simpul (*nodes*) dan sisi (*edges*) berdasarkan hasil pemetaan lintasan oleh robot menggunakan sensor Sharp GP2Y0A21 dan sensor fotodiode selama tahap *initial tracking*. Proses pemetaan ini memungkinkan robot memperoleh struktur topologi arena secara menyeluruh melalui pendeteksian posisi belokan dan persimpangan.

Setiap simpul pada graf direpresentasikan sebagai titik navigasi, yaitu simpul awal H (*home*) dan simpul tujuan A, B, C, serta D yang mewakili lokasi sumber api. Jalur antar simpul direpresentasikan sebagai sisi graf dan diberi bobot sesuai jarak tempuh aktual yang harus dilalui robot. Pembobotan jarak diperoleh berdasarkan pengukuran fisik lintasan menggunakan sensor encoder bawaan pada motor DC. Encoder menghitung jumlah pulse putaran roda, di mana jarak tempuh dihitung menggunakan konversi jumlah pulse terhadap keliling roda. Dengan diameter roda 65 mm, keliling roda adalah 204,2 mm. Jika encoder menghasilkan 20 pulse per satu putaran penuh, maka jarak per pulse dapat dihitung sebagai:

$$\text{Jarak per pulse} = \frac{204.2 \text{ mm}}{20} = 10.21 \text{ mm} \dots\dots\dots(3)$$

Berdasarkan nilai tersebut, dilakukan proses diskretisasi jarak untuk mempermudah representasi lintasan dalam bentuk graf navigasi. Pada penelitian

ini, satu satuan bobot didefinisikan sebagai akumulasi sekitar 16 pulsa encoder, sehingga satu bobot merepresentasikan jarak linear sebesar  $\pm 16$  cm, yang diperoleh dari hasil konversi  $16 \times 10,21 \text{ mm} \approx 163,36 \text{ mm}$  dan dibulatkan menjadi 16 cm. Pendefinisian satu bobot sebesar 16 cm dipilih untuk menjaga keseimbangan antara ketelitian pengukuran jarak dan kesederhanaan pemodelan graf.

Dengan dasar tersebut, bobot yang ditampilkan pada **Gambar 3.6**, seperti angka 5, 4, 3, 2, dan 1, merupakan nilai jarak relatif antar simpul dalam satuan unit jarak hasil proses kalibrasi dan *training phase*. Bobot 1 merepresentasikan jarak 16 cm, bobot 2 setara dengan 32 cm, bobot 3 setara 48 cm, bobot 4 setara 64 cm, dan bobot 5 merepresentasikan jarak 80 cm yang dihasilkan dari perpindahan robot.

Sebagai contoh, nilai bobot 5 menunjukkan bahwa lintasan antar simpul membutuhkan jarak tempuh sekitar 80 cm, yang diperoleh dari akumulasi pulsa encoder selama robot bergerak mengikuti jalur tersebut. Sementara itu, bobot 2 menunjukkan lintasan dengan jarak 32 cm, yang juga diperoleh dari hasil pencatatan pulsa encoder saat robot bergerak dari satu simpul ke simpul lainnya. Penentuan bobot ini dilakukan dengan menjalankan robot secara manual mengikuti jalur antar simpul dan mencatat total pulsa encoder selama proses pergerakan.

Nilai pulsa encoder yang diperoleh kemudian dikonversi ke dalam satuan jarak berdasarkan hasil kalibrasi roda, di mana setiap 1 bobot ditetapkan setara dengan 16 cm. Apabila jarak aktual antar simpul berada di antara dua nilai bobot, maka dilakukan pembulatan ke nilai bobot terdekat untuk menjaga konsistensi representasi graf navigasi. Dengan pendekatan ini, bobot pada graf bukan merupakan asumsi visual berdasarkan panjang panah pada gambar, melainkan nilai objektif yang merepresentasikan jarak fisik aktual yang diukur langsung oleh robot melalui encoder.

Dengan demikian, bobot pada graf bukan merupakan asumsi visual berdasarkan jarak panah pada gambar, tetapi merupakan nilai objektif yang merepresentasikan jarak fisik aktual yang diukur secara langsung oleh robot melalui encoder di mana  $V = \{H, A, B, C, D\}$  adalah himpunan simpul navigasi,  $E$  adalah jalur penghubung antar simpul dan  $W$  adalah bobot jarak real dalam bentuk konversi pulse encoder. Nilai bobot tersebut digunakan oleh algoritma djikstra untuk menentukan lintasan optimal yang kemudian dieksekusi robot secara *real-*

*time*. Pendekatan ini memastikan bahwa lintasan yang dihitung secara matematis sesuai dengan kondisi dunia nyata dan dapat dijalankan secara konsisten tanpa deviasi signifikan akibat slip roda atau rintangan permukaan.

Integrasi antara pemetaan sensor awal, pelatihan jarak berbasis encoder dan representasi graf berbobot memungkinkan robot mengeksekusi jalur secara efisien, adaptif dan terukur, sehingga dapat mencapai titik api dan kembali ke titik *home* dengan akurasi navigasi yang tinggi.

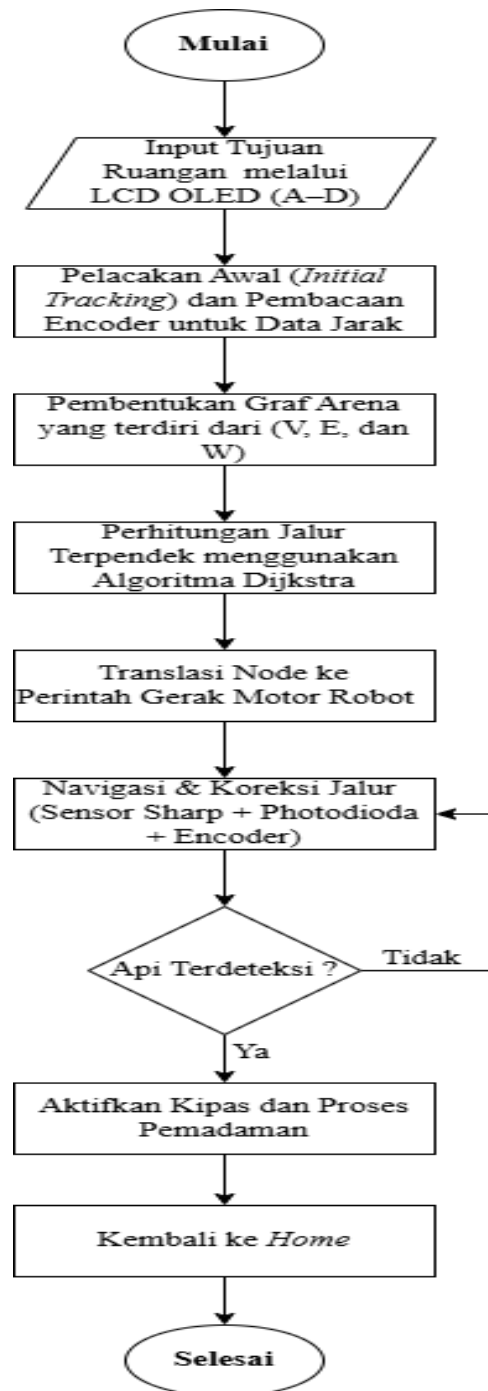
### 3.8 *Flowchart* dan Implementasi Algoritma djikstra

Implementasi algoritma djikstra pada sistem navigasi robot pemadam api digunakan untuk menentukan jalur terpendek dari titik awal (*home*) menuju lokasi sumber api pada salah satu ruangan tujuan [12]. Arena navigasi direpresentasikan sebagai graf berbobot, di mana simpul (*node*) menunjukkan titik-titik persimpangan dan sisi (*edge*) merupakan jalur penghubung yang memiliki bobot berdasarkan jarak lintasan fisik. Pembentukan graf dilakukan secara semi-otomatis, dimulai dari tahapan *initial tracking* ketika robot mengelilingi arena dan mencatat jarak antar simpul menggunakan pembacaan encoder motor DC sebagai dasar pengukuran jarak linier, serta sensor fotodioda dan Sharp GP2Y0A21 untuk validasi lintasan. Hasil akuisisi jarak kemudian dikonversi ke dalam bentuk matriks bobot sehingga dapat diproses oleh algoritma djikstra pada mikrokontroler ESP32.

Setelah graf terbentuk, algoritma djikstra menghitung jarak minimum dari simpul awal menuju simpul tujuan. Proses ini dilakukan melalui pembaruan bobot jarak secara iteratif berdasarkan jalur yang paling efisien hingga diperoleh urutan simpul optimal. Hasil perhitungan tersebut direkonstruksi menjadi jalur aktual melalui *previous node tracing*, yang kemudian diterjemahkan menjadi perintah gerak robot, seperti maju, berbelok kanan, berbelok kiri, dan berhenti, sesuai dengan jarak yang dibaca encoder [9].

Alur proses komputasi dan eksekusi algoritma ditunjukkan pada *flowchart* implementasi sistem yang menggambarkan tahapan mulai dari input tujuan ruangan, perhitungan jalur terpendek, navigasi lintasan, deteksi api, proses pemadaman, hingga kembali ke titik awal. **Gambar 3.7** merupakan menunjukkan alur kerja sistem navigasi dan pemadaman api pada robot berbasis algoritma

dijkstra, mulai dari pemilihan tujuan, perhitungan jalur terpendek, navigasi, deteksi api, hingga kembali ke titik awal.



**Gambar 3. 7** Flowchart implementasi sistem navigasi robot pemadam api berbasis algoritma djikstra

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil perancangan, implementasi serta pengujian sistem yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Algoritma Dijkstra berhasil diterapkan pada sistem navigasi robot pemadam api dan secara konsisten memilih jalur terpendek pada empat skenario ruangan dengan bobot minimum, yaitu 11 (Ruang A), 13 (Ruang B), 9 (Ruang C), dan 4 (Ruang D), dengan tingkat akurasi pemilihan jalur mencapai 100%.
2. Waktu tempuh robot menuju titik api menggunakan algoritma Dijkstra menunjukkan kinerja paling efisien dibandingkan BFS. Waktu tempuh rata-rata Dijkstra masing-masing sebesar 7,2 detik (Ruang A), 9,6 detik (Ruang B), 8,1 detik (Ruang C), dan 3,1 detik (Ruang D), sedangkan BFS membutuhkan 12,4 detik, 11,1 detik, 11,2 detik, dan 14,6 detik pada ruangan yang sama.
3. Implementasi algoritma Dijkstra pada robot pemadam api berbasis ESP32 menunjukkan kinerja yang stabil dan andal di lingkungan statis, dengan variasi waktu tempuh yang rendah serta integrasi sensor dan aktuator yang berjalan konsisten sesuai perancangan.

#### **5.2 Saran**

Berdasarkan keterbatasan penelitian dan hasil yang diperoleh, maka beberapa saran yang dapat diberikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Pengembangan selanjutnya disarankan untuk mengintegrasikan algoritma djikstra dan *Breadth First Search* (BFS) dalam satu sistem navigasi terpadu. Penggabungan kedua algoritma ini diharapkan mampu menghasilkan sistem navigasi yang adaptif dan efisien secara komputasi, serta tetap optimal pada lingkungan statis maupun dinamis.
2. Pengembangan metode navigasi adaptif dapat dilakukan dengan membandingkan algoritma djikstra dengan algoritma lain, seperti A\*, D\*,

atau D\* Lite, guna meningkatkan kemampuan robot dalam menyesuaikan jalur secara *real-time* terhadap perubahan kondisi lingkungan.

3. Pada pengembangan selanjutnya, disarankan untuk melakukan pengujian dan standarisasi terhadap tinggi api yang digunakan dalam proses pemadaman. Berdasarkan hasil pengamatan selama pengujian, tinggi api sebesar 15 cm dinilai paling representatif karena cukup stabil untuk dideteksi sensor api dan pemadam api menggunakan kipas, namun masih aman untuk pengujian laboratorium. Penetapan tinggi api yang konsisten diharapkan dapat meningkatkan validitas dan keterbandingan hasil pengujian sistem pemadam api.

## **DAFTAR PUSTAKA**

## DAFTAR PUSTAKA

- [1] Antara News, “Merefleksikan kembali kebakaran di Jakarta selama 2024.” Accessed: Jan. 30, 2025. [Online]. Available: <https://www.antaranews.com/berita/4551202>
- [2] BNPB, “Data kebakaran hutan dan lahan di Indonesia tahun 2024,” 2024. [Online]. Available: <https://bnpb.go.id/>
- [3] H. Ren, Y. Shi, and Y. Qiao, “A multi-objective optimization model of robot path planning under different scenarios,” in Proc. 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, Jan. 2023, pp. 1391–1394. :contentReference[oaicite:1]{index=1}
- [4] G. Jijesh, J. J.; Palle, S. S.; Bolla, D. R.; Penna, M.; Sruthi, V. P.; Alla, “Design and Implementation of Automated Fire Fighting and Rescuing Robot,” in *2020 International Conference on RTEICT*, 2020, pp. 320–323. doi: 10.1109/RTEICT49044.2020.9315552.
- [5] P. Ranaweera, D. M.; Hemapala, K. T. M. U.; Buddhika, A. G.; Jayasekara, “A Shortest Path Planning Algorithm for PSO-Based Firefighting Robots,” in *AEEICB*, 2018, pp. 1–5. doi: 10.1109/AEEICB.2018.8480971.
- [6] F. W. and X. Zhang, “Real-Time Path Planning of Mobile Robots Based on Multi-Sensor Fusion,” *Sensors*, vol. 21, no. 3, pp. 1–14, 2021, doi: 10.3390/s21030877.
- [7] R. Burchardt, H.; Salomon, “Implementation of Path Planning Using Genetic Algorithms on Mobile Robots,” *IEEE Int. Conf. Autom. Sci. Eng.*, pp. 1–6, 2016.
- [8] M. Lin et al, “Path Planning of Mobile Robot Based on Improved A\* Algorithm,” *IEEE Int. Conf. Autom. Sci. Eng.*, pp. 1–5, 2017.
- [9] U. & Kranthi, “Optimizing the Robot’s Path Using Dijkstra Algorithm,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 4, no. 6, 2016.
- [10] R. Mohan and A. P. Nair, “IoT-Based Intelligent Fire Detection and Monitoring System,” *IEEE Sensors Lett.*, vol. 5, no. 2, p. 2021, 2021, doi: 10.1109/LSSENS.2021.3054982.

- [11] A. Kumar, "Firefighting Mobile Robot with IR-Flame Sensor Navigation," *Int. J. Robot. Autom.*, vol. 9, no. 4, pp. 112–118, 2021.
- [12] S. L. et Al, "Shortest-Path Navigation for Autonomous Mobile Robots Using Dijkstra-Based Grid Mapping," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4114–4123, 2021.
- [13] Y. Zhao, L. Sun, and H. Liu, "Autonomous Firefighting Robot Navigation Using Hybrid Dijkstra–A\* Algorithm," *IEEE Access*, vol. 9, pp. 112345–112356, 2021, doi: 10.1109/ACCESS.2021.3104567.
- [14] A. M. Hasan, M. R. Islam, and S. A. Rahman, "Development of an Intelligent Firefighting Robot Using Multi-Sensor Integration," *Sensors*, vol. 22, no. 5, pp. 1–18, 2022, doi: 10.3390/s22051847.
- [15] K. Wang, J. Li, and X. Chen, "Shortest Path Planning for Mobile Robots in Dynamic Environments Based on Improved Dijkstra Algorithm," *International Journal of Advanced Robotic Systems*, vol. 18, no. 4, pp. 1–12, 2021, doi: 10.1177/17298814211031245.
- [16] M. Al-Khedher and A. Al-Shorman, "IoT-Based Fire Detection and Autonomous Firefighting Robot System," *Journal of King Saud University – Engineering Sciences*, vol. 34, no. 2, pp. 123–132, 2022, doi: 10.1016/j.jksues.2021.03.006.
- [17] S. Patel and R. Gupta, "Grid-Based Path Planning of Autonomous Mobile Robots Using BFS and Dijkstra Algorithms," *International Journal of Intelligent*.
- [18] A. M. Hasan, M. R. Islam, and S. A. Rahman, "Development of an Intelligent Firefighting Robot Using Multi-Sensor Integration," *Sensors*, vol. 22, no. 5, pp. 1–18, 2022, doi: 10.3390/s22051847.
- [19] *Robotics and Applications*, vol. 6, no. 3, pp. 345–356, 2022, doi: 10.1007/s41315-022-00215-8.
- [20] J. Kim, H. Park, and S. Lee, "Real-Time Navigation of Firefighting Robots Using Sensor Fusion and Shortest Path Algorithms," *Robotics and Autonomous Systems*, vol. 146, pp. 103884, 2022, doi: 10.1016/j.robot.2021.103884.
- [21] A. Elkhateeb, M. Hassan, and A. Riad, "Design and Implementation of an

- Autonomous Firefighting Robot with Intelligent Navigation,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 512–520, 2022, doi: 10.11591/ijece.v12i1.23456.
- [22] T. Nguyen and D. Tran, “Autonomous Mobile Robot Path Planning in Indoor Fire Scenarios Using Graph-Based Algorithms,” *Applied Sciences*, vol. 11, no. 19, pp. 1–16, 2021, doi: 10.3390/app11199123.
- [23] M. R. Kadhim, A. H. Ali, and S. A. Abbas, “Autonomous Firefighting Robot with Intelligent Path Planning Based on Dijkstra Algorithm,” *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 210–219, 2022, doi: 10.22266/ijies2022.0630.19.
- [24] L. Chen, Y. Zhou, and Q. Huang, “Graph-Based Navigation and Obstacle Avoidance for Autonomous Mobile Robots in Fire Emergency Environments,” *Journal of Robotics*, vol. 2021, pp. 1–11, 2021, doi: 10.1155/2021/8893476.
- [25] H. Zhang, Y. Liu, and J. Wang, “Autonomous Mobile Robot Navigation for Firefighting Applications Using Improved Graph Search Algorithms,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 8123–8132, 2022, doi: 10.1109/TIE.2021.3109874.
- [26] S. Ahmed, M. S. Hossain, and M. A. Rahman, “Design and Path Optimization of an Autonomous Firefighting Robot in Indoor Environments,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 456–464, 2022, doi: 10.14569/IJACSA.2022.0130653.