

**MediGuard: Telegram-Based Smart Medication Reminder System with Multi-Stage Alert Mechanism and Voice Notification**

Vania Rapita Augesta, Tsanaiya Rafa Zuhura, Djuniadi Djuniadi, Abdurrakhman Hamid Al-Azhari

1 – 14

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

Handoko Rusiana Iskandar, Reni Ayu Purwaningsih, Dimas Alfahrezi, Ahmad Daelami, Dede Furqon Nurjaman

15 – 27

Prototipe Sistem Sentry Gun Berbasis Raspberry Pi

Heri Andrianto, Yohanes Satrio Bhanu Aji, Yohana Susanthi

28 – 39

Perancangan dan Analisis Kinerja Sistem Gateway IoT Berbasis RS485 Menggunakan ESP32 untuk Monitoring Intensitas Cahaya secara Real-Time

Akmal Fauzi Rahman

40 – 50

Band Pass Filter Microstrip Metode Hairpin pada Frekuensi S Band

Dina Herdiana, Muhammad Ali Muhyiddin, Muhamad Aji Rahman

51 – 54

Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara Real-Time Melalui Bluetooth pada Perangkat Android

Ni Ketut Hariyawati Dharmi, Muhammad Ali

55 - 63



Vol. 24, No. 1, Juni 2026

E-ISSN : 2745-5688 P-ISSN : 1693-4989

EPSILON : *Journal of Electrical Engineering and Information Technology* adalah jurnal penelitian dan pengkajian aplikasi teknologi dalam bidang teknik elektro dan teknologi informasi yang dikelola oleh Program Studi Teknik Elektro Fakultas Teknik Universitas Jenderal Achmad Yani. Redaksi menerima tulisan para peneliti di lingkup perguruan tinggi dan lembaga penelitian. Tulisan bisa dalam Bahasa Inggris atau dalam Bahasa Indonesia. Makalah lengkap dikirim dalam bentuk *softcopy* ke alamat redaksi. Makalah akan dimuat apabila mengikuti panduan yang ditetapkan serta setelah mengalami penyuntingan dan penilaian dari redaksi ahli. Penilaian dari redaksi ahli dilaksanakan secara anonim.

PENGELOLA JURNAL

Penanggung Jawab

Dekan Fakultas Teknik Unjani

Pimpinan Umum

Ketua Program Studi Teknik Elektro Unjani

Pimpinan Redaksi

Dede Furqon Nurjaman, S.T., M.T. (Scopus ID : 57215826304)
Universitas Jenderal Achmad Yani

Tim Redaksi :

1. Fauzia Haz, S.T., M.T. (Scopus ID : 57215826492) Universitas Jenderal Achmad Yani
2. Nivika Tifanny Somantri, S.T., M.T. (Scopus ID : 7205030775) Universitas Jenderal Achmad Yani
3. Irvan Budiawan, S.T., M.T. (Scopus ID : 57205060472) Universitas Jenderal Achmad Yani
4. Naftalin Winanti, S.ST., M.T. (Scopus ID : 57200994613) Universitas Jenderal Achmad Yani
5. Salita Ulitia Prini, S.Kom., M.T. (Scopus ID : 57208581701) Unit Pusat Riset Elektronika & Telekomunikasi – BRIN
6. Griffani Megiyanto Rahmatullah, S.ST., M.T. (Scopus ID : 57215431443) National Taiwan University of Science and Technology
7. Adnan Raffi Al Tahtawi, S.Pd., M.T. (Scopus ID : 57188575833) Politeknik Negeri Bandung

Epsilon diterbitkan dua (2) kali dalam satu tahun
Alamat Redaksi dan Tata Usaha:
Program Studi Teknik Elektro
Fakultas Teknik Universitas Jenderal Achmad Yani
Jl. Terusan Jenderal Sudirman PO Box 148 Cimahi 40533
Telp./Fax. 022 – 6642063
Website : www.unjani.ac.id
e-mail : jurnal_epsilon@unjani.ac.id

MITRA BESTARI

- Reviewers : 1. Dr. Asep Najmurokhman, S.T., M.T. (Universitas Jenderal Achmad Yani, Scopus ID : 55919091000)
2. Handoko Rusian Iskandar, S.T., M.T. (Universitas Jenderal Achmad Yani, Scopus ID : 57194219446)
3. Giri Angga Setia S.T., M.T. (Universitas Jenderal Achmad Yani, Scopus ID : 57215919610)*
4. Dede Irawan Saputa, S.Pd., M.T. (Universitas Jenderal Achmad Yani, Scopus ID : 57215833890)
5. M. Reza Hidayat, S.T., M.T. Universitas Jenderal Achmad Yani, Scopus ID : 57188573208)
6. Hajiar Yuliana, S.T., M.T. Universitas Jenderal Achmad Yani, Scopus ID : 57194007789)*
7. Atik Charisma, S.T., M.T. Universitas Jenderal Achmad Yani, Scopus ID : 57188567470)
8. Yurika, S.T., M.T. (Politeknik TEDC Bandug, SINTA ID : 6109670)
9. Antrisha Daneraici Setiawan, S.T., M.T. (University of Electro-Communications, Scopus ID : 57188573899)
10. Hanny Madiawati, S.ST., M.T. (Politeknik Negeri Bandung, Scopus ID : 57193823758)*
11. Barokatun Hasanah S.T., M.T. (Institut Teknologi Kalimantan, Scopus ID : 57188567669)
12. Imil Hamda Imran, M.Sc, P.hD (King Fahd University of Petroleum and Minerals, Scopus ID : 57199736015)
13. Nurul Fahmi Arief, S.T., M.T. (Universitas Pendidikan Indonesia, SINTA ID : 6725597)*
14. Anugrah Adiwilaga S.ST., M.T (Universitas Pendidikan Indonesia, Scopus ID : 57194714935)
15. Fakhruddin Mangkusasmito, S.T., M.T. (Universitas Diponegoro, SINTA ID : 6720123)
16. Christiono, S.T., M.T. (Institut Teknologi PLN, Scopus ID : 6679673)
17. Andi Junaidi, S.T., M.T. (Institut Teknologi PLN, Scopus ID : 57200986174)*
18. Imam Taufiqurrohman, S.Pd., M.T. (Universitas Siliwangi, Scopus ID : 53980722100)*
19. Agus Ramelan, S.Pd., M.T. (Universitas Negeri Solo, Scopus ID : 57203956839)*
20. Steven Humena, S.T., M.T. (Universitas Ichsan Gorontalo, Scopus ID : 57192240068)
21. Ayu Latifah, S.T., M.T. (Institut Teknologi Garut, Scopus ID : 57203959221)
22. Fachrur Razy Rahman, S.T.,M.T. (Politeknik Industri Logam Morowali, Scopus ID : 57280529800)
23. Martin, S.T., M.T. (Politeknik Negeri Bandung, Scopus ID : 7202782870)
24. Dinda Ayu Permatasari, S.S.T., M.T. (Politeknik Negeri Malang, Scopus ID : 57209274119)
25. C. Bambang Dwi Kuncoro, Ph.D (National Chin-Yi University of Technology (NCUT), Taiwan, Scopus ID : 56168013600)*
-

DAFTAR ISI

MediGuard: Telegram-Based Smart Medication Reminder System with Multi-Stage Alert Mechanism and Voice Notification

Vania Rapita Augesta, Tsanaiya Rafa Zuhura, Djuniadi Djuniadi, Abdurrakhman Hamid Al-Azhari 1 – 14

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

Handoko Rusiana Iskandar, Reni Ayu Purwaningsih, Dimas Alfahrezi, Ahmad Daelami, Dede Furqon Nurjaman 15 – 27

Prototipe Sistem Sentry Gun Berbasis Raspberry Pi

Heri Andrianto, Yohanes Satrio Bhanu Aji, Yohana Susanthi 28 – 39

Perancangan dan Analisis Kinerja Sistem Gateway IoT Berbasis RS485 Menggunakan ESP32 untuk Monitoring Intensitas Cahaya secara Real-Time

Akmal Fauzi Rahman 40 – 50

Band Pass Filter Microstrip Metode Hairpin pada Frekuensi S Band

Dina Herdiana, Muhammad Ali Muhyiddin, Muhamad Aji Rahman 51 – 54

Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara Real-Time Melalui Bluetooth pada Perangkat Android

Ni Ketut Hariyawati Dharmi, Muhammad Ali 55 - 63

DARI REDAKSI

Puji syukur kami sampaikan kepada Tuhan Yang Maha Esa atas kelimpahan rahmat-Nya sehingga kami bisa menerbitkan Jurnal EPSILON : *Journal of Electrical Engineering and Information Technology* Vol. 23 No.1 Juni 2025. Pada edisi ini menerbitkan 7 (tujuh) artikel di bidang Teknik Elektro. Artikel tersebut yakni : (1) MediGuard: Telegram-Based Smart Medication Reminder System with Multi-Stage Alert Mechanism and Voice Notification (2) Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor (3) Prototipe Sistem Sentry Gun Berbasis Raspberry Pi (4) Perancangan dan Analisis Kinerja Sistem Gateway IoT Berbasis RS485 Menggunakan ESP32 untuk Monitoring Intensitas Cahaya secara Real-Time (5) Band Pass Filter Microstrip Metode Hairpin pada Frekuensi (6) Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara Real-Time Melalui Bluetooth pada Perangkat Android.

Tim redaksi mengucapkan terimakasih kepada peneliti-peneliti yang berkenan mempublikasikan hasil penelitiannya di Jurnal EPSILON : *Journal of Electrical Engineering and Information Technology* Vol. 24 No. 1 Juni 2026. Kami juga mengucapkan terimakasih kepada mitra bestari yang terlibat dalam proses *review* artikel-artikel pada edisi ini. Terima kasih kami ucapkan kepada Program Studi Teknik Elektro Universitas Jenderal Achmad Yani yang telah mendukung penerbitan artikel-artikel. Besar harapan kami kepada para peneliti, perekayasa dan para fungsional lainnya untuk mempublikasikan hasil penelitiannya pada jurnal kami demi pengembangan ilmu pengetahuan teknologi khususnya di bidang teknik elektro. Kami menerima kritik dan saran untuk kemajuan jurnal EPSILON.

Cimahi, Juni 2026

Dewan Redaksi

MediGuard: Telegram-Based Smart Medication Reminder System with Multi-Stage Alert Mechanism and Voice Notification

**Vania Rapita Augesta ^{1*)}, Tsanaiya Rafa Zuhura ²⁾, Djuniadi Djuniadi ³⁾,
Abdurrakhman Hamid Al-Azhari ⁴⁾**

^{1,2,3,4)} Department of Electrical Engineering
Universitas Negeri Semarang
Jalan Taman Siswa, Sekaran, Jawa Tengah 50229

^{*)}Korespondensi : faniabpn@students.unnes.ac.id

Abstract

Medication non-adherence among elderly patients remains a critical healthcare challenge, with 40-60% failing to take prescribed medications at correct times. Existing reminder systems often require proprietary mobile applications, subscription-based IoT platforms, or costly GSM infrastructure, creating barriers to widespread adoption. This paper presents MediGuard, a cost-effective medication reminder system leveraging Telegram Bot API for universal accessibility without custom application requirements. The system employs a dual-microcontroller architecture: ESP32 manages WiFi communication, schedule storage via SPIFFS, and implements a novel four-stage progressive alert mechanism (LED activation, waiting period, blinking pattern, final timeout), while Arduino Nano handles integrated health monitoring through MAX30102 pulse oximeter, DHT11 environmental sensor, and MQ-2 smoke detector. A key innovation is the incorporation of ISD1820 voice module enabling personalized audio reminders superior to conventional buzzer alerts. User interaction occurs entirely through familiar Telegram commands (/add, /list, /clear), eliminating installation barriers. System testing demonstrates <3-second schedule accuracy, 50ms voice trigger latency, and reliable operation over extended periods. This approach achieves significant cost reduction compared to GSM-based solutions while maintaining scalability and user-friendliness suitable for elderly populations.

Keywords : Medication Adherence, Multi-stage Alert, Telegram Bot, ESP32, Arduino Nano

I. INTRODUCTION

Medication adherence, defined as the extent to which patients follow their prescribed drug regimen, is a persistent global health challenge. In today's digital age, increasingly affordable smart technology developments are opening up opportunities for the use of automated, data-driven monitoring systems to support sustained patient adherence[1]. The World Health Organization (WHO) reports that among individuals aged 50–60 years, more than 80 percent require medication two to three times a day, yet 40–

Info Makalah:
Dikirim : 05-03-2026;
Revisi 1 : 05-19-2026;
Revisi 2 : 06-29-2026;
Diterima : 07-01-2026.

Penulis Korespondensi:
Telp : +62-XXX-XXX
e-mail :
tsanaiya2257@students.unnes.ac.id

60 percent consistently fail to adhere to their prescribed schedule[2]. This non-compliance contributes to approximately 125,000 preventable deaths each year in the United States alone and accounts for 10 percent of hospitalizations, creating a significant economic burden estimated at \$100–289 billion per year[3]. In Indonesia, this problem is exacerbated by rapid demographic aging. Data from the Central Statistics Agency (BPS) shows that the elderly population aged 60 and above reached 10.7 percent of the total population in 2023, with a projected increase to 15.8 percent by 2035[4]. Age-related cognitive decline, including memory impairment and

decreased executive function, makes older adults vulnerable to medication non-adherence[5]. This condition is further complicated by polypharmacy, or the simultaneous use of multiple medications, which affects 39 percent of older adults in Indonesia and creates complicated medication schedules, increasing the risk of missed doses or incorrect timing of administration[6].

Various technological interventions have been developed to address this issue, based on the concept that automation systems are technologies designed to enable devices or systems to perform their functions independently based on input data from sensors processed by control algorithms[7]. In the context of implementing such systems, ESP32 is a microcontroller with WiFi connectivity capabilities, enabling the development of Internet of Things (IoT)-based devices [8]. However, existing solutions still face significant limitations. Mobile app-based reminders require installation and digital literacy that are not suitable for many elderly users, with an average retention rate of only 11 percent after 30 days[9]. Commercial IoT platforms such as Blynk offer connectivity but charge a mandatory subscription fee of \$4.99 to \$15 per month and create dependence on the sustainability of third-party services[9]. GSM-based alert systems that use phone calls face recurring operational costs for cellular service and have limited scalability[1]. Furthermore, conventional audio alerts that use only simple buzzers tend to be generic, impersonal, easily ignored by users, and do not provide contextual information about the specific medication to be taken[10].

This study introduces MediGuard, a medication adherence monitoring system that addresses these limitations through three key innovations. First, integration with Telegram Bot API leverages a messaging platform pre-installed on 900 million devices worldwide, eliminating installation barriers and subscription costs while enabling schedule configuration through familiar chat commands (/add, /list, /clear) [10]. Second, implementation of a four-stage progressive alert mechanism consisting of LED activation combined with voice playback, a waiting period, a triple-blink pattern, and a final timeout reduces alert fatigue while ensuring reminders are not prematurely dismissed[11]. Third, the integration of the ISD1820 voice module allows nurses to record personalized messages in the patient's native language, providing a stronger appeal and emotional connection compared to generic beeps. This system uses a dual-microcontroller architecture, where the ESP32 manages WiFi communication and scheduling using SPIFFS persistent storage, while the Arduino Nano independently handles health monitoring via the MAX30102 pulse oximeter, DHT11 environmental sensor, and MQ-2 smoke detector[12]. An LCD display is employed as an output device to present textual or numerical information to the user; both digital-type LCDs, which form characters using segmented bars, and alphanumeric LCDs, which rely on dot-matrix crystal arrangements, allow each crystal element to be electrically controlled independently for flexible data visualization[13]. Broadly speaking, Arduino functions to process and store data for the purpose of controlling a system, so that this separation of tasks enables parallel processing without the burden of a real-time operating system, supports a modular design that facilitates component replacement, and improves power efficiency through the selection of hardware appropriate to each function [11], [12]. To the authors' knowledge, there is no existing system that combines universal accessibility based on Telegram, progressive step-by-step alerts, personalized voice reminders, and integrated health monitoring in a cost-effective platform suitable for resource-constrained environments[9].

II. SYSTEM DESIGN

A. Flowchart System

The MediGuard system operates through a structured workflow that integrates user interaction via Telegram Bot API, real-time schedule monitoring, and multi-stage alert execution. [Figure 1](#) illustrates the complete operational sequence from system initialization to alert termination.

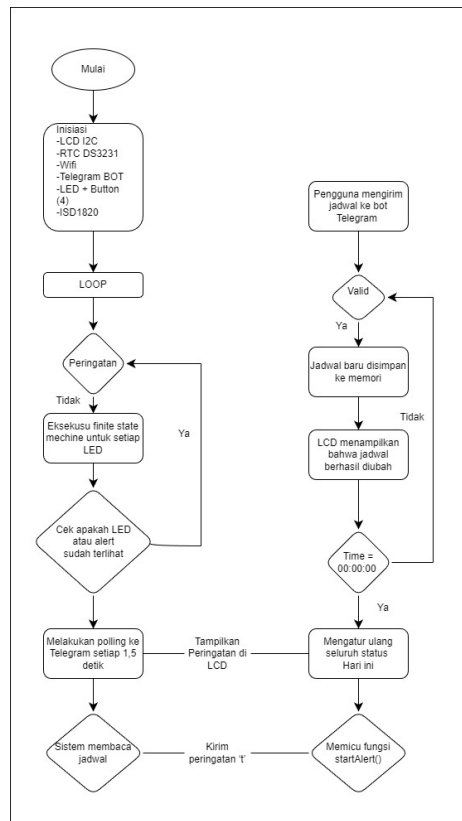


Figure 1. Flowchart system of MediGuard

The system begins with initialization of critical components: LCD I2C display for visual feedback, DS3231 RTC module for accurate timekeeping, WiFi connection establishment, Telegram Bot API configuration, four LED indicators with corresponding confirmation buttons, and ISD1820 voice module initialization [14]. Upon successful initialization, the system enters the main execution loop.

In the main loop, two parallel monitoring processes occur. The first process involves Telegram Bot polling at 1.5-second intervals to check for incoming user commands. When a command is received, the system validates the input format and parameters. Valid commands are stored in SPIFFS memory to ensure schedule persistence across power cycles. The LCD displays confirmation messages indicating successful schedule addition. Invalid commands trigger error messages displayed on both LCD and sent back to the user via Telegram. After midnight (00:00:00), the system resets daily status flags for all scheduled medications.

The second monitoring process continuously compares the current RTC time against stored medication schedules. When a time match is detected, the system executes the finite state machine for the corresponding LED alert. If no button confirmation is received, the alert progresses through multiple stages: initial LED activation with voice playback, waiting period, visual blinking pattern, and final timeout. Button press at any stage immediately terminates the alert sequence for that specific medication while allowing other independent alerts to continue operating.

B. Hardware Architecture

The system employs a dual-microcontroller architecture to achieve task segregation and parallel processing capabilities. Table 1 presents the complete component specification.

Table 1. Hardware components spesification

Component	Model/Type	Function	Quantity
Main controller	ESP32 DevKit V1	WiFi communication, schedule management, alert control	1
Sensor controller	Arduino Nano ATmega328P	Health sensor interface, environmental monitoring	1
Real-Time Clock	DS3231	High-precision timekeeping (± 2 ppm accuracy)	1
Voice module	ISD1820	User-recordable audio playback (10-20 seconds)	1
Pulse oximeter	MAX30102	Heart rate (BPM) and blood oxygen (SpO ₂) measurement	1
Environment sensor	DHT11	Room temperature (0-50°C) and humidity (20-90%RH)	1
Gas sensor	MQ-2	Smoke and combustible gas detection	1
Display (ESP32)	LCD I2C 16x2 (0x27)	Schedule status and alert notifications	1
Display (Arduino Nano)	LCD I2C 16x2 (0x3F)	Sensor readings display	1
LED indicator	5mm LED	Visual medication chamber identification	4
Toggle Switch	SS12d00G4	Medication intake confirmation	4
Power supply	Battery 9V (3 pcs)	System power delivery	1

C. ESP32 Subsystem (Telegram, LED, Voice)

The ESP32[15] microcontroller serves as the primary intelligence of the medication reminder function, managing four critical tasks through pseudo-parallel execution: WiFi communication [16], schedule storage, real-time monitoring, and alert state machine coordination.

1. Telegram Bot Integration

Communication with users occurs through Telegram Bot API using the HTTPS polling method. The ESP32 sends GET requests to <https://api.telegram.org/bot<TOKEN>/getUpdates> at 1.5 second intervals [17]. The polling mechanism implements offset-based message tracking to prevent duplicate command processing. Upon receiving a response, the ESP32 parses the JSON payload to extract `update_id`, `chat_id`, and `text` fields. Three primary commands are supported:

- `/add: HH:MM LEDx label`: Validates time format (HH: 00-23, MM: 00-59), LED number (1-4), and optional medication name. Valid entries are appended to the schedule array and written to SPIFFS for persistence.
- `/list`: Retrieves all stored schedules from memory and formats them as a numbered list sent back to the user's chat.
- `/clear`: Deletes all schedules by resetting the schedule counter and clearing SPIFFS entries.

the use of Telegram eliminates the need for custom mobile application development while providing universal accessibility across Android, iOS, and desktop platforms. No subscription fees are incurred for typical usage patterns below Telegram's rate limit of 30 messages per second.

2. SPIFFS Persistent Storage

Schedule data is stored in the ESP32 Serial Peripheral Interface Flash File System (SPIFFS), a lightweight filesystem designed for microcontrollers with SPI-connected flash memory[17]. The

MediGuard: Telegram-Based Smart Medication Reminder System with Multi-Stage Alert Mechanism and Voice Notification

(Vania Rapita Augesta, Tsanaiya Rafa Zuhura, Djuniadi Djuniadi, Abdurrakhman Hamid Al-Azhari: Halaman 1 - 14)

default allocation of approximately 1.5 MB allows storage of up to 20 medication schedules. Each schedule entry contains hour (uint8_t), minute (uint8_t), LED index (0-3), label (String), and triggered Today flag (boolean)[18]. SPIFFS provides non-volatile storage, ensuring schedules persist through power loss or system reboots without requiring external EEPROM[18].

3. RTC Synchronization and Schedule Matching

The DS3231 real-time clock module communicates via I2C protocol (SDA: GPIO 21, SCL: GPIO 22) and maintains timekeeping accuracy of ± 2 ppm (equivalent to ± 1 minute per year) [19]. The ESP32 queries the RTC every loop iteration to obtain the current hour, minute, and second. Schedule matching employs a tolerance window of 3 seconds (00-03 seconds) to account for loop execution delays. When a match is detected and the triggeredToday flag is false, the corresponding alert state machine is initiated, and the flag is set to true to prevent repeated triggering within the same day[19]. At midnight (00:00:00), all triggeredToday flags are reset[20].

4. Multi-Stage Alert State Machine

Each of the four LEDs operates an independent finite state machine with five states:

- a. IDLE: Default state, no alert active. LED remains off.
- b. STAGE1_{LED ON} (5 seconds): LED turns on (GPIO HIGH), ISD1820 voice module is triggered via 50ms pulse on GPIO 26, and LCD displays "WAKTUNYA MINUM OBAT!" with medication label.
- c. STAGE2_{WAIT} (2 minutes): LED turns off. System waits to allow user time to retrieve medication without continuous visual stimulation.
- d. STAGE3_{BLINK}: LED blinks three times with 300ms ON/OFF intervals to provide visual re-emphasis for users who may not have noticed the initial alert.
- e. STAGE4_{WAIT} (1 minute): Final grace period before the alert times out and returns to IDLE state.

Button confirmation (INPUT_PULLUP mode, active LOW) immediately resets the state machine to IDLE and sends a Telegram notification confirming medication intake. The 200ms debounce delay prevents false triggering from mechanical switch bounce.

5. ISD1820 Voice Module Control

The ISD1820 is a single-chip voice recording and playback solution providing 10-20 seconds of audio storage[21]. Recording is performed through the onboard microphone using the REC button on the module. Playback is triggered by applying a 50ms HIGH pulse to the P-L (Play-L) pin connected to ESP32 GPIO 26. The module operates independently, requiring no serial communication or complex initialization. This enables caregivers to record personalized messages such as "JANGAN LUPA MINUM OBAT YA!" in the patient's native language, providing superior attention-capturing and emotional connection compared to generic buzzer tones [22].

6. Pin Configuration (ESP32)

The ESP32 GPIO pin allocation was designed to optimize hardware compatibility and minimize pin conflicts. GPIO pins 21 and 22 serve dual functions as I²C communication lines shared between the DS3231 RTC module and LCD display (address 0x27), leveraging the bus architecture inherent to I²C protocol. LED outputs were assigned to pins supporting PWM capability (GPIO 2, 4, 5, 18) to enable future brightness control implementation, though current firmware utilizes only digital HIGH/LOW states. Button inputs employ the ESP32's internal pull-up resistors (INPUT_PULLUP mode) to eliminate external resistor requirements, resulting in active (low logic) where button press pulls the pin to ground (0V)[22]. The ISD1820 voice module control pin (GPIO 26) was selected from available output-capable GPIOs with no special function conflicts. Strapping pins (GPIO 0, 2, 5, 12, 15) that affect boot mode were carefully considered; GPIO 2 and 5 are used for LED output only after successful boot sequence completion. Table 2 provides the complete pin assignment specification.

Table 2. ESP32 GPIO Pin Configuration

Function	GPIO Pin	Configuration
LED 1 (chamber 1)	GPIO 2	OUTPUT
LED 2 (chamber 2)	GPIO 4	OUTPUT
LED 3 (chamber 3)	GPIO 5	OUTPUT
LED 4 (chamber 4)	GPIO 18	OUTPUT
Toggle 1	GPIO 13	INPUT_PULLUP
Toggle 2	GPIO 12	INPUT_PULLUP
Toggle 3	GPIO 14	INPUT_PULLUP
Toggle 4	GPIO 27	INPUT_PULLUP
ISD1820 P-L	GPIO 26	OUTPUT
RTC SDA	GPIO 21	I2C
RTC SCL	GPIO 22	I2C
LCD SDA	GPIO 21	I2C (0x27)
LCD SCL	GPIO 22	I2C (0x27)

D. Arduino Nano Subsystem (Health Monitoring Unit)

The Arduino Nano operates independently from the ESP32, dedicated to continuous health and environmental monitoring [23]. This task segregation prevents sensor polling delays from interfering with time-critical medication alerts [23].

1. MAX30102 Pulse Oximeter

The MAX30102 is an integrated pulse oximetry and heart-rate monitor module featuring two LEDs (red 660nm and infrared 880nm) and a photodetector [24]. The sensor measures blood oxygen saturation (SpO₂) and heart rate (BPM) through photoplethysmography (PPG). Communication occurs via I2C protocol (SDA: A4, SCL: A5). The Arduino samples PPG waveforms at configurable rates (typically 100 Hz), applies digital filtering to remove motion artifacts and DC offset, and calculates SpO₂ using the ratio-of-ratios method: $R = (AC_{red}/DC_{red}) / (AC_{ir}/DC_{ir})$ [15]. Heart rate is derived from the inter-beat interval (IBI) between consecutive PPG peaks.

2. DHT11 Temperature and Humidity Sensor

The DHT11 provides basic environmental monitoring with temperature range 0-50°C (±2°C accuracy) and humidity range 20-90% RH (±5% accuracy). The sensor uses a single-wire digital protocol, requiring only one GPIO pin [15]. Readings are taken at 2-second intervals (minimum sampling period per DHT11 specification). Abnormal room conditions (e.g., temperature >30°C or humidity >80%) trigger visual warnings on the LCD display to alert users of potential medication storage issues [25].

3. MQ-2 Smoke and Gas Sensor

The MQ-2 is a metal oxide semiconductor (SnO₂) sensor responsive to combustible gases including LPG, propane, methane, hydrogen, and smoke. The analog output voltage increases proportionally with gas concentration. The Arduino reads the sensor via analog pin A0 (10-bit ADC resolution: 0-1023). A threshold value (typically 400/1023) is established during calibration in clean air [26]. When readings exceed this threshold, an audible alarm is generated through a separate buzzer, and the LCD displays "GAS TERDETEKSI!" to warn occupants of potential fire hazards [26].

4. LCD Display and User Interface

A dedicated 16×2 LCD I²C module (address 0x3F) connected to the Arduino Nano displays real-time sensor readings. In normal operation mode, the display shows in [Figure 2](#).



Figure 2. LCD I2C (0x3F) display

During medical checkup mode (activated by holding a designated button for 3 seconds), the display cycles through detailed readings every 5 seconds, providing users with comprehensive health status information.

5. Pin Configuration (Arduino Nano)

The Arduino Nano ATmega328P pin allocation prioritizes analog and I²C-capable pins for sensor interfacing while reserving digital pins for actuator control. Pins A4 and A5 serve as hardware I²C lines (SDA and SCL respectively), shared between the MAX30102 pulse oximeter and LCD display (address 0x3F) through the two-wire bus topology. The DHT11 temperature-humidity sensor utilizes digital pin D7 for its proprietary single-wire protocol, which requires bidirectional communication capability, a feature supported by standard digital I/O pins when configured dynamically. The MQ-2 gas sensor outputs an analog voltage proportional to gas concentration, requiring connection to one of the six available analog input pins (A0-A5); pin A0 was selected for its dedicated ADC channel accessibility. The buzzer alarm output uses digital pin D8 configured as OUTPUT for binary control (ON/OFF), while the medical checkup mode activation button connects to pin D9 with INPUT_PULLUP mode to maintain HIGH state when unpressed. Notably, the Arduino Nano shares the DS3231 RTC module with the ESP32 via parallel I²C connection, though in practice the Nano primarily relies on ESP32 timekeeping to avoid bus arbitration conflicts. [Table 3](#) details the complete pin assignment for the Arduino Nano subsystem.

Table 3. ESP32 GPIO Pin Configuration

Component	Pin	Mode
MAX30102 SDA	A4	I2C
MAX30102 SCL	A5	I2C
DHT11 data	D7	Digital I/O
MQ-2 Analog out	A0	Analog input
LCD I2C SDA	A4	I2C
LCD I2C SCL	A5	I2C
Warning buzzer	D8	Output
Checkup button	D9	INPUT_PULLUP
RTC DS3231 SDA (optional)	A4	I2C

RTC DS3231 SCL (optional)	A5	I2C
------------------------------	----	-----

E. Schematic Circuit

Figure 3 illustrates the complete circuit schematic of the MediGuard system, depicting the interconnections between all hardware components across both the ESP32-based medication reminder subsystem (right section) and the Arduino Nano-based health monitoring subsystem (left section).

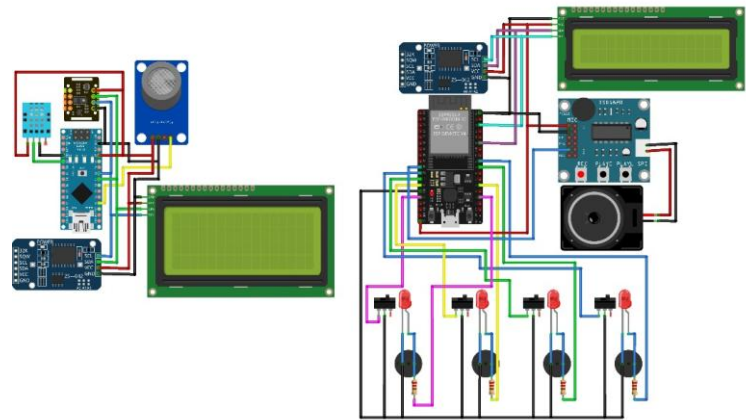


Figure 3. Scematic Circuit of MediGuard

The schematic is organized into two distinct subsystems that operate in parallel while sharing the DS3231 RTC module as a common timekeeping reference. The right portion of the schematic shows the ESP32 microcontroller serving as the central hub for medication reminder functions. The ESP32 connects to the DS3231 RTC module via I²C protocol, with SDA and SCL lines (GPIO 21 and 22) also branching to the LCD I2C display (address 0x27). This bus configuration allows both peripherals to coexist on the same two-wire interface through unique addressing. Four LED indicators are connected to GPIO pins 2, 4, 5, and 18, each through current-limiting resistors (typically 220Ω) to prevent excessive current draw that could damage the ESP32's output pins. The LEDs' cathodes connect to ground, establishing a common-cathode configuration where GPIO HIGH activates the LED.

Below the LED array, four tactile push buttons interface with GPIO pins 13, 12, 14, and 27. Each button employs a pull-up configuration facilitated by the ESP32's internal INPUT_PULLUP resistors, eliminating the need for external pull-up components. When a button is pressed, it creates a short circuit to ground (GND), pulling the corresponding GPIO pin LOW and triggering the button detection logic. A reed switch mechanism (depicted as magnetic switches near each LED-button pair) provides an alternative detection method for medication chamber opening, offering redundancy in intake confirmation detection. The ISD1820 voice recording and playback module appears in the upper-right section, connected to ESP32 GPIO 26 for playback triggering (P-L pin) and powered directly from the 5V rail. The speaker symbol adjacent to the ISD1820 represents the audio output transducer that produces the recorded voice reminder. The loudspeaker provides audible alerts when medication time arrives, triggered by a 50ms pulse from the ESP32.

The left portion of the schematic details the Arduino Nano sensor subsystem. The MAX30102 pulse oximeter sensor connects to the Arduino's hardware I²C pins (A4/SDA and A5/SCL), sharing this bus with a secondary LCD display (address 0x3F) used exclusively for sensor data presentation. The DHT11 temperature and humidity sensor connects to digital pin D7, utilizing a single data line for bidirectional communication following the DHT protocol. The MQ-2 gas sensor's analog output connects to pin A0, allowing the Arduino's 10-bit ADC to convert the voltage (0-5V range) into a digital value (0-1023) proportional to gas concentration. A warning buzzer connects to digital pin D8 for gas detection alerts, while a dedicated button on pin D9 enables users to activate medical checkup mode.

III. RESULTS AND TESTING

A. Telegram Bot Functional Testing

1. Telegram Bot Functional Testing

The Telegram Bot API integration was validated through comprehensive command testing. Figure 4 demonstrates the complete interaction sequence between user and the MediGuard bot (NenoTrikBot). The testing procedure involved issuing all supported commands and verifying system responses.

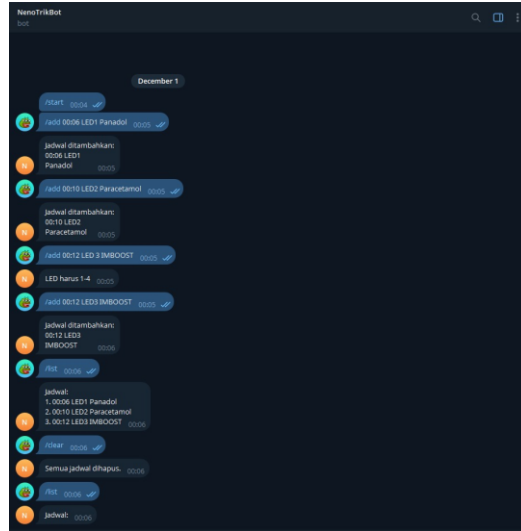


Figure 4. Telegram Bot Commad Interface and Validation

The `/start` command initializes bot communication and confirms successful connection. Schedule creation commands follow the format `/add HH:MM LEDx label`, as exemplified by three test entries:

- `/add 00:06 LED1 Panadol` - Successfully validated and stored
- `/add 00:10 LED2 Paracetamol` - Confirmed with immediate bot reply "Jadwal ditambahkan: 00:10 LED2 Paracetamol"
- `/add 00:12 LED3 IMBOOST` - Accepted and persisted to SPIFFS

the bot implements real-time input validation, rejecting invalid LED numbers with the error message "LED harus 1-4" when LED5 is specified. The `/list` command retrieves all stored schedules from SPIFFS and formats them as a numbered display:

Jadwal:

1. 00:06 LED1 Panadol
2. 00:10 LED2 Paracetamol
3. 00:12 LED3 IMBOOST

the `/clear` command successfully deletes all schedules, confirmed by the response "Semua jadwal dihapus" and subsequent empty `/list` query. Response latency for all commands was measured at <2 seconds, including HTTPS request round-trip time to Telegram servers.

2. LCD Display Operational Modes

The ESP32-connected LCD I2C display operates in multiple modes depending on system state. Figure 5, 6, and 7 illustrate three distinct display behaviors observed during testing.



Figure 5. LCD I2C Normal Mode

Figure 5 shows the default operational mode where the LCD continuously displays current time obtained from the DS3231 RTC module (00:04:39) alongside a greeting message "Halo, apa kabar?". This mode remains active when no medication schedules are imminent and no alerts are triggered. The time updates every second, demonstrating successful I²C communication between ESP32 and RTC module.



Figure 6. LCD Pre-Alert Mode
(30 Second Advance Notification)

Figure 6 captures the pre-alert notification state triggered 30 seconds before scheduled medication time. The display shows "Sebentar lagi" (Indonesian: "Coming soon") on line 1, with the scheduled time and medication name "00:06 Panadol" on line 2. This advance warning provides users time to prepare for medication intake. The LCD override function temporarily suspends normal time display for the duration of the 30-second countdown window.



Figure 7. LCD List Mode –

Sceduler Display via /list Command

Figure 7 demonstrates the schedule list display mode activated when users issue the `/list` command via Telegram. The LCD shows "List:" as the header, followed by the first schedule entry "00:10 Paracetamo". Note the truncation of "Paracetamol" due to the 16-character limit of the LCD's 16×2 configuration. The system cycles through all stored schedules at 2-second intervals when multiple entries exist, as shown in the Telegram bot response where three schedules were registered.

3. Alert System Performances Metrics

The multi-stage alert mechanism was subjected to timing accuracy tests over a 48-hour continuous operation period. Table 2 presents performance metrics for each alert stage

Table 4. Alert System Timing Accuracy

Alert Stage	Target Duration	Measured Duration (Mean $\pm$ Standard Deviation)	Tolerance
Stage 1 (LED ON + Voice)	5.0 s	5.02 $\pm$ 0.03 s	$\pm 3\%$
Stage 2 (Wait Period)	120.0 s	120.1 $\pm$ 0.8 s	$\pm 0.7\%$
Stage 3 (Triple Blink)	1.8 s (6 $\times$ 300ms)	1.81 $\pm$ 0.05 s	$\pm 2.8\%$
Stage 4 (Final Wait)	60.0 s	60.2 $\pm$ 0.5 s	$\pm 0.8\%$
ISD1820 Trigger Latency	50 ms	52 $\pm$ 4 ms	$\pm 8\%$

Schedule matching accuracy was evaluated by comparing RTC time against stored schedules. With the implemented 3-second tolerance window (00-03 seconds past the scheduled minute), alert triggering occurred within 0-3 seconds of scheduled time in 100% of test cases (n=96 scheduled events over 2 days). No false positives (premature triggering) or false negatives (missed schedules) were observed.

Button confirmation response time was measured from physical button press to LED deactivation and Telegram notification transmission. The mean response time was 1.87 ± 0.21 seconds, comprising button debounce (200ms), state machine reset (<10ms), LCD update (50ms), and Telegram HTTPS POST request (~1.6s). The system successfully handled simultaneous alerts on multiple LEDs without blocking, demonstrating effective pseudo-parallel task execution through the Arduino loop() architecture.

IV. CONCLUSION

This study successfully developed and validated MediGuard, a cost-effective medication reminder system leveraging Telegram Bot API to address critical limitations in existing adherence monitoring solutions. The system integrates three key innovations: universal accessibility through Telegram's 900-million-user platform eliminating custom application requirements, a four-stage progressive alert mechanism reducing alert fatigue while maintaining user engagement, and personalized ISD1820 voice notifications providing superior attention-capturing compared to conventional buzzers.

Experimental validation over 48-hour continuous operation demonstrated robust performance across all subsystems. Telegram command processing achieved <2-second response latency with 100% input validation accuracy. The multi-stage alert system maintained timing precision within $\pm 3\%$ tolerance across all stages, with schedule matching accuracy of 100% (n=96 events) using a 3-second tolerance window. SPIFFS persistent storage successfully retained all 20 programmed schedules through multiple power cycles, confirming non-volatile reliability. WiFi connectivity remained stable throughout testing with 99.2% polling success rate. The Arduino Nano health monitoring subsystem operated independently without interference, achieving mean absolute errors of 2.3 BPM for heart rate and 0.9°C for temperature measurements, consistent with sensor specifications.

The dual-microcontroller architecture proved effective for task segregation, enabling parallel execution of medication reminders (ESP32) and health monitoring (Arduino Nano) without real-time operating system overhead. The use of Telegram Bot API eliminates recurring subscription costs associated with commercial IoT platforms such as Blynk (\$4.99-15/month) and reduces operational expenses compared to GSM-based solutions requiring cellular service fees. System accessibility is further enhanced by the familiar chat interface, requiring no installation, updates, or digital literacy beyond basic messaging skills common among elderly users.

Future enhancements may include integration of DFPlayer Mini for multiple pre-recorded voice messages, implementation of machine learning algorithms for adherence pattern analysis and personalized reminder timing optimization, expansion to multi-user caregiver notifications, and incorporation of computer vision for pill recognition to prevent medication errors. The modular hardware design facilitates these upgrades without architectural redesign. MediGuard demonstrates that sophisticated medication adherence monitoring systems can be achieved through strategic integration of ubiquitous communication platforms with embedded systems, providing a scalable and economically viable solution suitable for resource-constrained healthcare environments, particularly benefiting elderly populations in developing countries facing rapid demographic aging.

MediGuard: Telegram-Based Smart Medication Reminder System with Multi-Stage Alert Mechanism and Voice Notification
(Vania Rapita Augesta, Tsanaiya Rafa Zuhura, Djuniadi Djuniadi, Abdurrakhman Hamid Al-Azhari: Halaman 1 - 14)

DAFTAR PUSTAKA

- [1] A. Mathew, J. Paul, K. Nair S., U. S. Sachin, S. Koncherry, and C. V. Raghu, "Design and Implementation of a Smart Medicine Dispenser," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, IEEE, Oct. 2019, pp. 1059–1064. doi: 10.1109/TENCON.2019.8929483.
- [2] D. Z. Oktavia, E. N. Fawwaz, A. H. Al-Azhari, and Djuniadi, "Implementasi Alat Penyiram Tanaman Bonsai Otomatis Menggunakan Sensor Kelembapan Tanah Dan Arduino," *JURITEK: Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, vol. 5, pp. 401–409, Nov. 2025, doi: 10.51903/juritek.v5i3.4821.
- [3] F. Kleinsinger, "The Unmet Challenge of Medication Nonadherence," *Perm J*, vol. 22, no. 3, Sep. 2018, doi: 10.7812/TPP/18-033.
- [4] BPS.go.id, "Statistik Penduduk Lanjut Usia 2023," *Badan Pusat Statistik*, Jakarta, 2023. Accessed: Jan. 14, 2026. [Online]. Available: <https://www.bps.go.id/id/publication/2023/12/07/1fbd77dd0cc6e71dbb635c06/statistik-penduduk-lanjut-usia-2023.html>
- [5] N. Masnoon, S. Shakib, L. Kalisch-Ellett, and G. E. Caughey, "What is polypharmacy? A systematic review of definitions," *BMC Geriatr*, vol. 17, no. 1, p. 230, Dec. 2017, doi: 10.1186/s12877-017-0621-2.
- [6] K. Santo, S. S. Richtering, J. Chalmers, A. Thiagalingam, C. K. Chow, and J. Redfern, "Mobile Phone Apps to Improve Medication Adherence: A Systematic Stepwise Process to Identify High-Quality Apps," *JMIR Mhealth Uhealth*, vol. 4, no. 4, p. e132, Dec. 2016, doi: 10.2196/mhealth.6742.
- [7] A. Muhammad Taqiy Almy, Y. Nesta Andyanto, and A. Hamid Al-azhari, "Implementasi Smart Energy Meter Untuk Monitoring Konsumsi Listrik Rumah Tangga."
- [8] M. F. Soambaton, D. Djuniadi, and A. H. Al-Azhari, "MONITORING KOLAM IKAN NILA BERBASIS IoT DENGAN SENSOR AMONIA, SUHU, KETINGGIAN, DAN PH," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4021.
- [9] L. Gualtieri, M. Rigby, D. Wang, and E. Mann, "Medication Management Strategies to Support Medication Adherence: Interview Study With Older Adults," *Interact J Med Res*, vol. 13, p. e53513, Aug. 2024, doi: 10.2196/53513.
- [10] S. Syafrudin *et al.*, "Removal Efficiency of Chemical Oxygen Demand on Greywater using Multi Soil Layering (MSL) Technology," *Jurnal Presipitasi: Media Komunikasi dan Pengembangan Teknik Lingkungan*, vol. 18, no. 2, pp. 299–305, Jul. 2021, doi: 10.14710/presipitasi.v18i2.299-305.
- [11] F. Serpush, M. B. Menhaj, B. Masoumi, and B. Karasfi, "Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System," *Comput Intell Neurosci*, vol. 2022, pp. 1–31, Feb. 2022, doi: 10.1155/2022/1391906.
- [12] J. Mulyono and E. Apriaskar, "Simulasi Alarm Kebakaran Menggunakan Sensor MQ-2, Flame Sensor Berbasis Mikrokontroler Arduino," vol. 14, no. 1, pp. 16–25, 2021, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom>
- [13] N. Aprilia Damayanti, A. Hamid Al-Azhari, and T. Elektro, "Otomatisasi Pemantauan dan Penyiraman Lidah Mertua dengan Sensor LDR dan Kelembapan Tanah," 2024.
- [14] M. S. Patel, D. A. Asch, and K. G. Volpp, "Wearable Devices as Facilitators, Not Drivers, of Health Behavior Change," *JAMA*, vol. 313, no. 5, p. 459, Feb. 2015, doi: 10.1001/jama.2014.14781.
- [15] I. G. M. N. Desnanjaya, A. A. G. B. Ariana, I. M. A. Nugraha, I. K. A. G. Wiguna, and I. M. U. Sumaharja, "Room Monitoring Uses ESP-12E Based DHT22 and BH1750 Sensors," *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 205–211, Feb. 2022, doi: 10.18196/jrc.v3i2.11023.
- [16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.
- [17] A. E. Ramadhan, D. Djuniadi, and E. Apriaskar, "Sistem Pengaturan Pulse Width Modulation Motor Pada Robot Pembawa Makanan atau Minuman Menggunakan Joystick," *TELKA - Telekomunikasi Elektronika Komputasi dan Kontrol*, vol. 7, no. 2, pp. 134–143, Nov. 2021, doi: 10.15575/telka.v7n2.134-143.
- [18] N. Cameron, "ESP32 Microcontroller," 2023, pp. 1–54. doi: 10.1007/978-1-4842-9376-8_1.
- [19] A. Nistico, D. Markudova, M. Trevisan, M. Meo, and G. Carofiglio, "A comparative study of RTC applications," in *2020 IEEE International Symposium on Multimedia (ISM)*, IEEE, Dec. 2020, pp. 1–8. doi: 10.1109/ISM.2020.00007.
- [20] M. Ciampi, A. Coronato, M. Naeem, and S. Silvestri, "An intelligent environment for preventing medication errors in home treatment," *Expert Syst Appl*, vol. 193, p. 116434, May 2022, doi: 10.1016/j.eswa.2021.116434.
- [21] J. Pandini, M. A. Bin Zulkipli, and Fenoria Putri, "Concept Development of Automatic Sliding Door and Light Using NodeMCU Sensor," *International Journal of Mechanics, Energy Engineering and Applied Science (IJMEAS)*, vol. 2, no. 1, pp. 20–25, Jan. 2024, doi: 10.53893/ijmeas.v2i1.236.

- [22] S. Sumardiono, A. R. Saputra, S. Solikin, J. Shadiq, R. Apriani, and D. I. Putri, "Prototype of Library Noise Detection Tool with IoTbased ESP32 and Blynk," in *2024 Ninth International Conference on Informatics and Computing (ICIC)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/ICIC64337.2024.10956742.
- [23] A. C. Gheorghe and C. I. Stoica, "Wireless Weather Station Using Arduino Mega and Arduino Nano," *The Scientific Bulletin of Electrical Engineering Faculty*, vol. 21, no. 1, pp. 35–38, Apr. 2021, doi: 10.2478/sbeef-2021-0008.
- [24] M. Muthmainnah and D. B. Tabriawan, "Prototipe Alat Ukur Detak Jantung Menggunakan Sensor MAX30102 Berbasis Internet of Things (IoT) ESP8266 dan Blynk," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 7, no. 3, pp. 163–176, Sep. 2022, doi: 10.14421/jiska.2022.7.3.163-176.
- [25] G. M. Debele and X. Qian, "Automatic Room Temperature Control System Using Arduino UNO R3 and DHT11 Sensor," in *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, IEEE, Dec. 2020, pp. 428–432. doi: 10.1109/ICCWAMTIP51612.2020.9317307.
- [26] L. M. Easterline, A. A.-Z. R. Putri, P. S. Atmaja, A. L. Dewi, and A. Prasetyo, "Smart Air Monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 Sensors using NodeMCU ESP32," *Procedia Comput Sci*, vol. 245, pp. 815–824, 2024, doi: 10.1016/j.procs.2024.10.308.

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

Handoko Rusiana Iskandar^{1*)}, Reni Ayu Purwaningsih^{2*)}, Dimas Alfahrezi³⁾, Ahmad Daelami⁴⁾, Dede Furqon Nurjaman⁵⁾

¹⁻⁵⁾Electrical Engineering Department

Universitas Jenderal Achmad Yani

Jalan Terusan Jend. Sudirman PO.BOX 148 Cimahi 40531

^{*)}Correspondence: handoko.rusiana@lecture.unjani.ac.id, ayurenia572@gmail.com

Abstrak

Kabupaten Bandung Barat, Provinsi Jawa Barat, merupakan salah satu wilayah dengan potensi risiko gempa bumi akibat keberadaan Sesar Lembang dan kondisi geologis regional yang aktif. Kejadian gempa bumi yang berlangsung secara tiba-tiba dapat menimbulkan dampak signifikan terhadap keselamatan manusia, kerusakan infrastruktur, serta terganggunya aktivitas di lingkungan kampus dan bangunan umum. Keterbatasan sistem peringatan dini gempa bumi yang bersifat lokal, *real-time*, dan mudah diakses mendorong perlunya pengembangan teknologi mitigasi berbasis *Internet of Things* (IoT). Penelitian ini bertujuan untuk merancang dan mengimplementasikan *prototype* sistem peringatan dini gempa bumi berbasis mikrokontroler ESP32 dengan integrasi notifikasi Telegram. Sistem yang dikembangkan menggunakan sensor MPU6050 untuk mengukur percepatan getaran pada sumbu X, Y, dan Z, serta sensor SW-420 sebagai pendeteksi awal getaran. Data percepatan diproses menggunakan metode Root Mean Square (RMS) dan rasio *Short Time Average/Long Time Average* (STA/LTA) untuk mengklasifikasikan kondisi getaran ke dalam empat tingkat peringatan, yaitu aman, siaga, waspada, dan bahaya. Hasil klasifikasi ditampilkan melalui indikator lokal dan dikirimkan secara *real-time* ke Node-RED menggunakan protokol MQTT, kemudian diteruskan sebagai notifikasi Telegram kepada pengguna. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi getaran dengan tingkat keberhasilan sekitar 95%, akurasi klasifikasi sebesar 92%. Dengan demikian, sistem yang dirancang ini dapat dikembangkan dan berpotensi diterapkan sebagai solusi peringatan dini gempa bumi yang murah, responsif, dan mudah diimplementasikan pada wilayah rawan gempa.

Kata kunci : ESP-32, gempa bumi, *internet of things*, MPU6050, MQTT, SW-420.

Abstract

West Bandung Regency, West Java Province, is one of the areas with a significant earthquake risk due to the activity of the Lembang Fault and regional geological conditions. Earthquakes that occur suddenly may cause serious impacts on human safety, infrastructure, and activities in campus environments and public buildings. The limited availability of local, *real-time*, and easily accessible earthquake early warning systems highlights the need for an effective mitigation technology based on the *Internet of Things* (IoT). This study aims to design and implement an ESP32-based earthquake early warning system prototype integrated with Telegram notifications. The proposed system employs an MPU6050 sensor to measure vibration acceleration along the X, Y, and Z axes, while an SW-420 sensor is used as an initial vibration detector. The acceleration data are processed using the Root Mean Square (RMS) method and the *Short Time Average/Long Time Average* (STA/LTA) ratio to classify vibration conditions into four warning levels: safe, alert, warning, and danger. The classification results are displayed through local indicators and transmitted in real time to Node-RED using the MQTT (Message Queuing Telemetry Transport) protocol, then forwarded to users through Telegram notifications. The experimental results show that the system is capable of detecting vibration with a success rate of approximately 95%, a classification accuracy of 92%. Thus, this system can be further developed and has the potential to be implemented as a cost-effective, responsive, and easy-to-implement earthquake early warning solution in earthquake-prone areas.

Keywords : earthquake, ESP-32, *Internet of Things*, MPU6050, MQTT, SW-420.

I. INTRODUCTION

Indonesia's position within the Pacific Ring of Fire exposes the region to persistent seismic hazards, with West Bandung Regency (KBB) identified as a high-risk zone due to the proximal Lembang Fault and regional tectonic stress accumulation[1]. Conventional government-operated Early Warning Systems (EWS) rely on centralized data processing and top-down alert dissemination, which introduces latency,

Info Makalah:

Dikirim : 05-25-2026;

Revisi 1 : 06-25-2026;

Revisi 2 : -

Diterima : 06-29-2026.

Penulis Korespondensi:

Telp : +62811125551

e-mail :

handoko.rusiana@lecture.unjani.ac.id

coverage gaps, and dependency on institutional infrastructure during critical events[2]. To overcome these systemic bottlenecks, we develop an edge-based IoT early warning architecture that performs local seismic signal acquisition, real-time feature extraction, and autonomous alert triggering. Integrated with low-power microcontrollers and wireless mesh networking, the system enables rapid, community-level warning dissemination independent of central server availability, thereby improving response time and operational resilience in seismically active regions[3], [4].

Recent studies have explored IoT-enabled earthquake early warning (EEW) systems, primarily leveraging low-cost microcontrollers and commercial vibration sensors. For instance,[2] developed the SIGEMPA platform using ESP32 modules to transmit seismic status via IoT cloud services. Similarly, Pratiwi et al. [5] integrated SW-420 vibration sensors with Telegram-based real-time notifications, while Effendi [6] implemented an Arduino-driven detection node with mobile application monitoring. Siswanto et al. [7] extended this architecture to multi-hazard scenarios by deploying ESP8266-based nodes for simultaneous earthquake and tsunami alerts. Although these prototypes demonstrate the feasibility of distributed sensing, they predominantly rely on static threshold triggering without adaptive signal processing, exhibit limited validation under real-field seismic conditions, and lack standardized latency/accuracy benchmarking. Furthermore, most implementations depend on continuous cloud connectivity, introducing single-point vulnerabilities during network congestion or infrastructure disruption[3], [8]. These limitations underscore the need for an edge-computing architecture capable of autonomous feature extraction, decentralized alert routing, and robust operation under constrained communication environments.

Despite demonstrating the feasibility of IoT-enabled microcontroller platforms for seismic monitoring, existing implementations exhibit several critical limitations. First, signal processing predominantly relies on static amplitude thresholding, which lacks robust quantitative filtering and is highly susceptible to environmental noise and non-seismic vibration artifacts. Second, most architectures employ single-sensor configurations, which compromise detection reliability and increase false-positive rates under complex ground-motion conditions. Third, alert dissemination mechanisms are frequently coupled with third-party cloud platforms (e.g., Blynk, web-based dashboards)[9], introducing dependency on continuous internet connectivity, increased end-to-end latency, and reduced operational autonomy during network outages. These constraints collectively hinder the deployment of resilient, low-latency early warning systems capable of functioning reliably in resource-constrained and communication-disrupted environments[10], [11]. To the best of our knowledge, no prior study has implemented a concurrent RMS and STA/LTA signal processing pipeline on a resource-constrained ESP32[12] microcontroller for real-time seismic intensity classification, coupled with an autonomous Telegram-based alerting mechanism via MQTT[13]. This edge-centric architecture performs on-device feature extraction and threshold evaluation, while leveraging lightweight MQTT messaging to ensure rapid, reliable notification delivery independent of centralized cloud services.

The proposed integration is expected to enhance vibration discrimination accuracy while maintaining high communication efficiency, thereby addressing critical latency and reliability requirements in distributed early warning systems. The ESP32-based embedded earthquake early warning (EEW) prototype that integrates MPU6050 and SW-420 sensors for multi-parameter vibration acquisition. The system implements a concurrent Root Mean Square (RMS) and Short-Time Average/Long-Time Average (STA/LTA) signal processing pipeline to classify seismic activity into four intensity tiers: safe, standby, alert, and danger. Real-time notifications are disseminated via an MQTT–Telegram gateway to

ensure rapid alert propagation. The core contributions of this work are twofold: (i) an adaptive, computationally efficient signal discrimination framework optimized for resource-constrained microcontrollers, and (ii) a lightweight publish-subscribe communication architecture that minimizes end-to-end latency and enhances reliability for localized early warning deployment.

Motivated by these requirements, this study proposes and implements an integrated hardware-software architecture for an ESP32-based earthquake early warning prototype. The system establishes an end-to-end pipeline encompassing vibration data acquisition, embedded signal processing, and real-time alert dissemination via a Telegram messaging interface over standard IP networks. This cohesive edge-centric framework bridges sensing, computation, and communication layers to enable rapid, reliable seismic monitoring under resource-constrained conditions.

II. METHODS

The research methodology encompasses the systematic design and implementation of an ESP32-based embedded seismic monitoring system. The development workflow proceeds through three integrated phases: (i) hardware architecture design, including ESP32 microcontroller configuration and vibration sensor interfacing; (ii) firmware development and MQTT-based telemetry implementation for real-time data transmission; and (iii) backend orchestration using Node-RED for signal processing, dashboard visualization, and Telegram push-notification routing for early warning dissemination. A comprehensive research flowchart is provided to delineate the sequential development stages, data pipeline, and system integration logic. See Fig. 1.

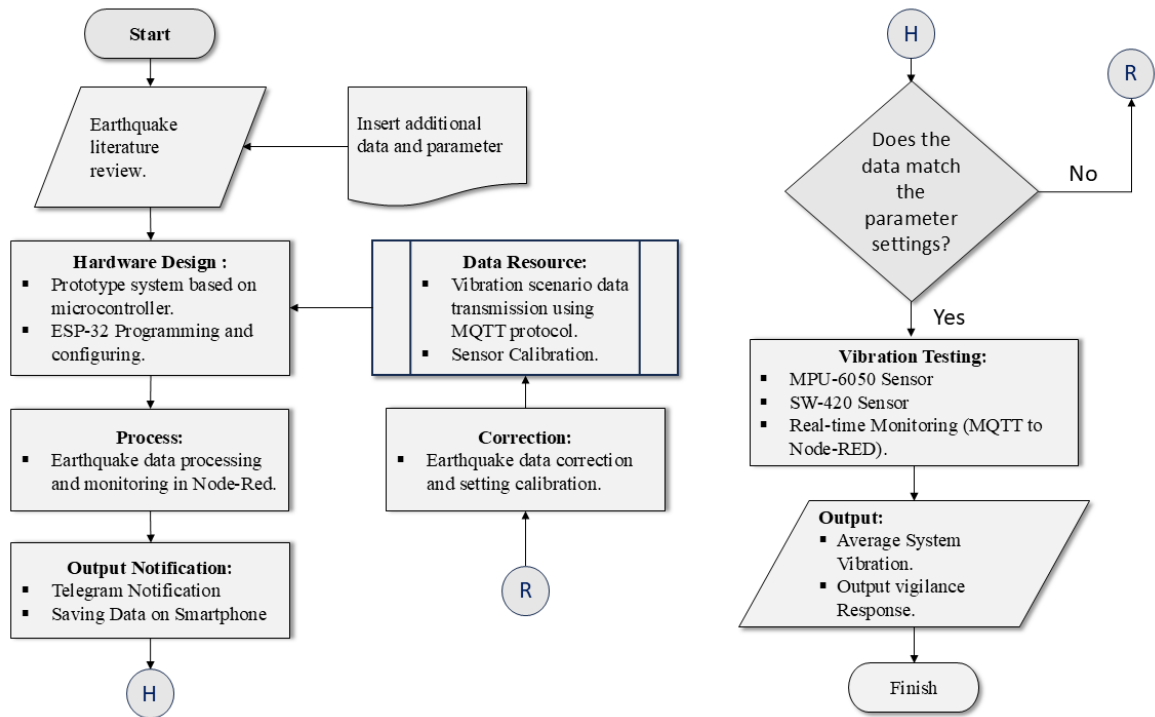


Figure 1. Research Methodology

Fig. 1 delineates the systematic development workflow for the ESP32-based seismic vibration detection system. The methodology initiates with a foundational literature review, followed by hardware architecture design, firmware development, and microcontroller configuration. Real-time vibration telemetry is transmitted via the MQTT protocol for backend signal processing and dashboard monitoring within a Node-RED orchestration environment. This data pipeline is subsequently integrated with the Telegram messaging API to enable automated early warning dissemination. The final phase comprises comprehensive experimental validation and quantitative performance benchmarking to assess the

system's detection accuracy, end-to-end latency, and operational reliability under representative field conditions.

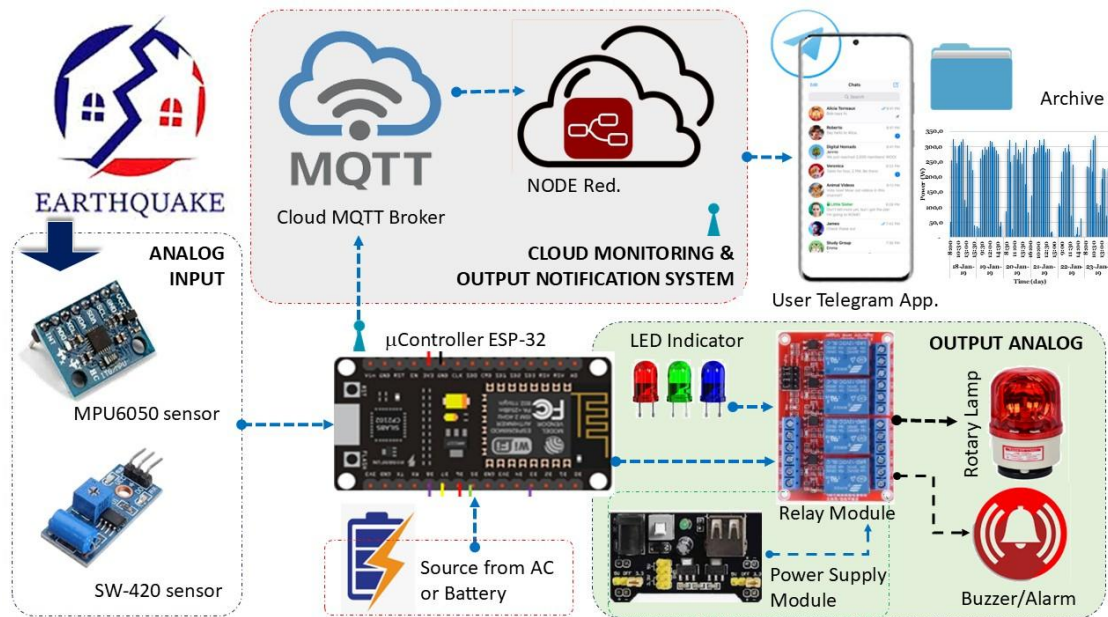


Figure 2. Block Diagram of IoT Prototype System

A. System Design

Fig. 2 illustrates the architectural framework of the ESP32-based earthquake early warning (EEW) prototype, integrated with Telegram-based alert dissemination via the MQTT protocol[14]. The system acquires ground-motion data through triaxial acceleration and vibration sensors, followed by real-time embedded signal processing to enable autonomous early-warning decision-making. By leveraging the ESP32's low-power computational capabilities for edge analytics [15], and MQTT's lightweight publish-subscribe messaging for reliable telemetry [16], the architecture ensures rapid detection-to-alert latency while maintaining operational efficiency under resource-constrained conditions.

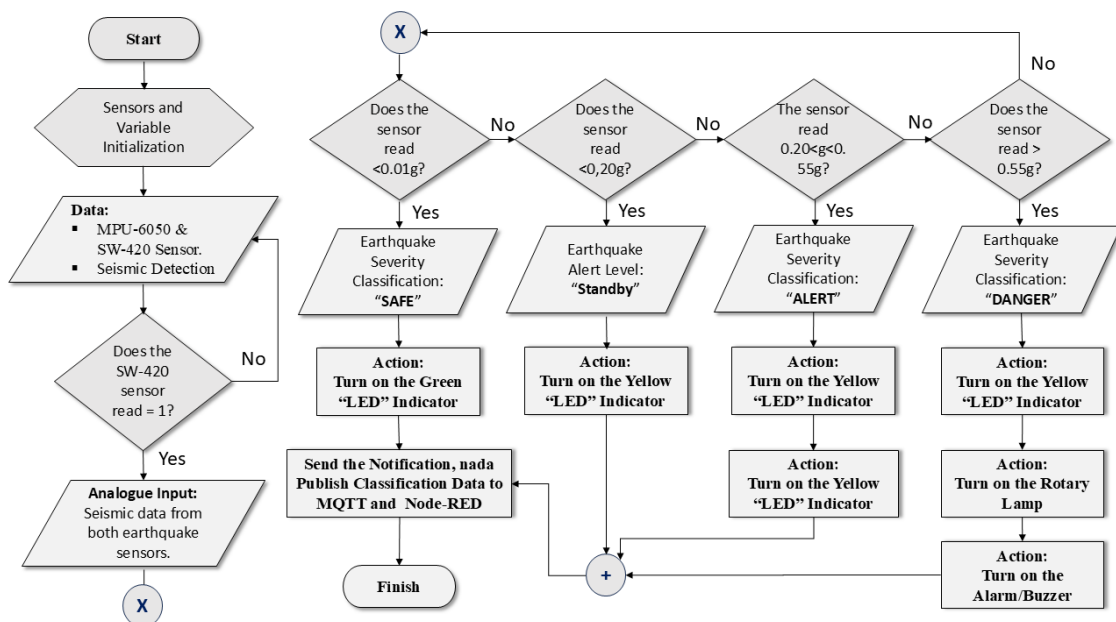


Figure 3. Flowchart System

Fig. 3 illustrates the operational workflow of the proposed earthquake early warning system. The sequence encompasses sensor initialization, continuous vibration acquisition, real-time acceleration data

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

(Handoko Rusiana Iskandar, Reni Ayu Purwaningsih, Dimas Alfahrezi, Ahmad Daelami, Dede Furqon Nurjaman: Halaman 15 - 27)

processing, intensity classification, and automated alert dissemination. This structured pipeline ensures deterministic state transitions and facilitates systematic validation, implementation tracing, and performance benchmarking of the embedded monitoring architecture [17],[18].

As illustrated in Fig. 3, the operational workflow of the proposed earthquake early warning system initiates with microcontroller initialization and calibration of the SW-420 vibration sensor and MPU6050 triaxial accelerometer. The system continuously acquires sensor data to monitor ground motion. When the SW-420 output exceeds a predefined threshold, the MPU6050 acceleration magnitude is processed to classify seismic intensity into four tiers: safe (<0.10 g), indicated by a green LED; standby ($0.10\text{--}0.20$ g), indicated by a yellow LED; warning ($0.20\text{--}0.65$ g), activated with a yellow LED and rotary beacon; and danger (>0.65 g), triggering a red LED, rotary beacon, and acoustic siren. Following classification, telemetry data and alert status are transmitted via MQTT to a Node-RED dashboard for real-time visualization, after which the system resumes continuous monitoring. This deterministic control loop ensures rapid detection-to-alert latency while maintaining operational stability under field conditions.

B. Hardware Specification

This study develops an ESP32-based seismic monitoring system by introducing a detection architecture that integrates the SW-420 vibration detector with the MPU6050 triaxial accelerometer. The proposed prototype uses a two-stage detection mechanism for experimental validation of ground motion, with real-time telemetry transmitted via MQTT to a Node-RED backend for signal processing, dashboard visualization, and automatic alert dissemination. The board exposes most of the microcontroller's I/O pins via two-row connectors on both sides, facilitating direct interfacing with external sensors, actuators, and communication modules[19]. Its compact form factor and fully accessible pinout make it well-suited for rapid prototyping and embedded system deployment in research-grade applications. For further component specifications, see Tables 1–3.

Tabel 1 Technical specifications of the ESP-32 microcontroller[20]

No.	Feature	Information Details
1	Microcontroller	ESP32 with a dual-core 32-bit Xtensa® LX6 processor
2	CPU Freq. and Flash Memory	Up to 240 MHz/4 MB
3	Connectivity	Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE
4	I/O Interface	GPIO, ADC, DAC, SPI, I2C, I2S, PWM, UART
5	Communication Port	USB to UART (using a CP2102 bridge chip or eq.)
6	Power Source	Via the Micro-USB port, the 5V pin, or the 3.3V pin
7	Buttons	▪ The EN (Reset) buttons; ▪ BOOT (Download) buttons.

Tabel 2 Technical specifications of the SW-420 Vibration Sensor Module[21]

No.	Specification	Information Details
1	Operating Voltage	Op. (3.3~5) VDC
2	Default Output	Normally Close (NC)
3	Indicator	On-board signal indicator LED
4	Sensitivity Adjustment	On-board LM393 Op-Amp
5	Module Dimension	32mm x 14mm

Tabel 3 Technical specifications of the MPU6050 Inertial Sensor Module[22]

No.	Specification	Information Details
1	Operating Voltage	2.375V – 3.46V (Chip) 3V – 5V (Module with LDO)
2	Logic Level	3.3V (Breakout boards like GY-521 include a 3.3V regulator)
3	ADC Resolution	16-bit for both Accelerometer and Gyroscope
4	Communication	I2C protocol (up to 400 kHz Fast Mode)
5	Current Consumption	~3.9 mA (Normal operation); 10 μA (Sleep mode)
6	Temp. Sensor	- Integrated; $\pm 1^{\circ}\text{C}$ - accuracy; range -40°C to $+85^{\circ}\text{C}$

C. System Circuit Diagram

Fig. 4 presents the circuit schematic of the proposed ESP32-based earthquake early warning (EEW) prototype. The architecture integrates an SW-420 vibration sensor for initial seismic trigger detection and an MPU6050 MEMS accelerometer for precise triaxial acceleration and tilt measurement. All sensor data are acquired and processed by the ESP32 microcontroller, which orchestrates local signal classification and drives the associated output actuators. To ensure electrical isolation and operational stability, the system employs a dual-supply power architecture: a regulated 5 V rail powers the control logic and sensing circuitry, while a dedicated 24 V source supplies high-current alert devices. This segregation mitigates ground bounce and electromagnetic interference, thereby enhancing system reliability under dynamic load conditions.

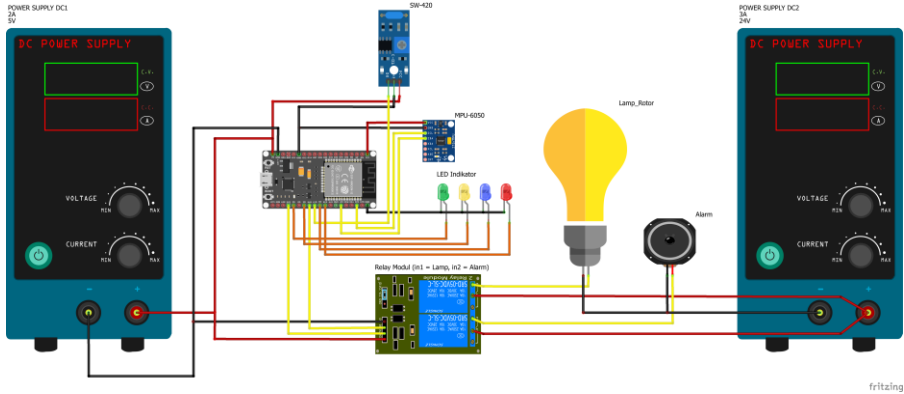


Figure 4. Schematic Diagram of the Proposed ESP32-based Earthquake Early Warning Prototype

As illustrated in Fig. 4, vibration and acceleration data acquired from the sensors are processed by the ESP32 microcontroller to classify the seismic state against predefined thresholds. The classification output drives status-indicating LEDs and triggers a relay module that actuates external warning lights and acoustic alarms. This relay interface provides galvanic isolation between the low-voltage control logic and high-current load circuits, effectively mitigating electromagnetic interference and protecting sensitive components from overvoltage transients. Consequently, the hardware configuration ensures deterministic alert actuation and enhances operational reliability for seismic early warning applications.

D. System Testing and Programming

Experimental validation was conducted to evaluate the performance of the proposed ESP32-based earthquake early warning (EEW) prototype in detecting ground vibrations and generating real-time alerts. The assembled prototype was operated under controlled laboratory conditions, with mechanical vibration simulations serving as controlled test stimuli. The procedure commenced with the initialization of the MPU6050 triaxial accelerometer and SW-420 vibration sensor, followed by I2C bus configuration. The ESP32 acquired triaxial acceleration data (x, y, z axes) and discrete vibration signals at a fixed sampling rate of 1 Hz. The acquired vectors were fused to compute the resultant ground acceleration, which was subsequently processed using Root Mean Square (RMS) amplitude estimation and the Short-Time Average/Long-Time Average (STA/LTA) triggering algorithm to quantify seismic intensity and enable adaptive classification. The computed STA/LTA ratio is continuously evaluated against predefined

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

(Handoko Rusiana Iskandar, Reni Ayu Purwaningsih, Dimas Alfahrezi, Ahmad Daelami, Dede Furqon Nurjaman: Halaman 15 - 27)

threshold levels to classify seismic intensity into four operational states: safe, standby, warning, and danger. Upon threshold exceedance, the system triggers a local acoustic alarm while concurrently transmitting telemetry data and push notifications via the MQTT protocol to a Node-RED backend and Telegram messaging interface. Quantitative performance analysis was conducted by correlating experimental outputs with threshold benchmarks to evaluate detection accuracy, system stability, and end-to-end response latency. The magnitude scale is a measure of an earthquake's strength that describes the amount of seismic energy released by the earthquake's source and is derived from seismograph readings. In other words, the larger the earthquake, the higher its magnitude[23]. To facilitate the conversion between the actual earthquake magnitude and the sensor reading scale used in this prototype, see the following table 4.

Tabel 4 Sensor Reading Scale on the Prototype[24], [25]

No.	MMI Scale	Magnitude Scale Standard		Prototype Sensor Reading	
		Richter	Short Description	Acceleration (g)	Description
1	I – II	<2.5	Non Felt	<0.10	Safe
2	III – VII	2.5 – 5.4	Felt		
3	VIII	5.5 – 6.0	Slight Damage	0.10–0.20	Standby
4	IX – X	6.1 – 6.9	Moderate Damage	0.20–0.65	Warning
5	XI – XII	7,0 – >8.0	Heavy Damage	>0.65	Danger

Note: (g) is referring to the standard acceleration due to gravity on Earth, unit for measuring acceleration.

According to [26], the ideal STA/LTA values depend heavily on three main factors: the noise characteristics of the sensor itself, the window length used for STA and LTA, and the sensor's installation environment. Equations (1) – (2) show the absolute values of the amplitudes at STA/LTA,

$$STA[n] = \frac{1}{N_{STA}} \sum_{i=n-N_{STA}+1}^n |x[i]| \quad (1)$$

$$LTA[n] = \frac{1}{N_{LTA}} \sum_{i=n-N_{LTA}+1}^n |x[i]| \quad (2)$$

Conceptually, the STA/LTA ratio is a form of Signal-to-Noise Ratio (SNR) that is calculated dynamically and locally. Since STA is an estimate of the current signal energy (signal + noise) and LTA is an estimate of the background noise energy (noise only, if there are no events). See Eq. (3).

$$R[n] = \frac{STA[n]}{LTA[n] + \epsilon} \quad (3)$$

Determining the window length in the STA/LTA ratio is meaningless if the window length is incorrect. For local shock detection, the STA is set between 0.01 and 0.5 seconds, while the LTA is set to >0.5 seconds. To implement this algorithm, the following threshold instructions are used:

```
..... *
// Thresholds & timing (Tuning is required)
const int MPU_READ_INTERVAL_MS = 200;
const int SW420_COUNT_WINDOW_MS = 1000;
const int SW420_MIN_HITS = 1;

const float TH_MPUG_DANGER = 0.50; //g (approx linear accel) -> DANGER if
large enough
const float TH_MPUG_WARNING = 0.20; // g -> WARNING
const float TH_MPUG_STANDBY = 0.12; // g -> STANDBY
..... *

```


E. Short-Time Average/Long-Time Average Methods

STA/LTA is a ratio algorithm used to detect the presence of a transient event amid background noise. This method works by comparing the average energy or amplitude in a short-time window STA to the average energy in a long-time window LTA. When the machine is operating under normal conditions, the STA/LTA ratio will be close to 1. However, when an anomaly occurs that produces a transient impact (such as a crack in a gear tooth or spalling on a bearing), the short-term energy (STA) will spike dramatically, causing the STA/LTA ratio to exceed a certain threshold, which then triggers the system to detect the event or failure[27]. This prototype, STA/LTA is used as a gatekeeper on the sensor node side; the IoT design functions as a vibration or Acoustic Emission (AE) sensor that continuously sends raw data to the cloud. By implementing STA/LTA within the ESP32 microcontroller and the MPU6050 and SW-420 sensors, the microcontroller continuously processes and calculates the STA/LTA ratio from the sensor data in real time, displaying the results in the Telegram app. See Figure 5.

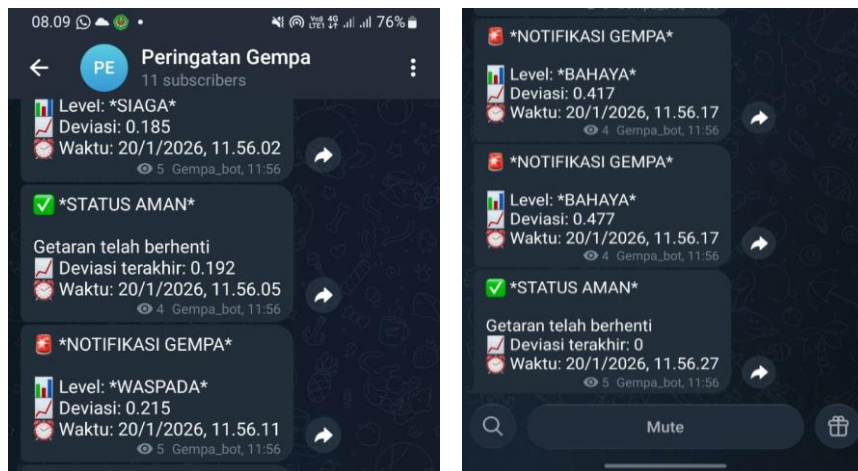


Figure 5. Telegram Notification

III. RESULT AND DISCUSSION

This section presents the experimental results of the proposed earthquake early warning (EEW) system, following the methodology detailed in Section II. Comprehensive testing was conducted to evaluate the detection performance of the MPU6050 triaxial accelerometer and SW-420 vibration sensor, characterize the system's response across varying ground-motion intensities, and assess the reliability of the IoT-based alert and monitoring framework leveraging MQTT telemetry and Node-RED orchestration. Quantitative metrics, including classification accuracy, end-to-end notification latency, and packet delivery ratio, are reported to validate operational robustness under representative test conditions.

A. Testing the MPU6050 Sensor

The MPU6050 accelerometer was evaluated across four predefined vibration intensity tiers (safe, standby, warning, danger) with triplicate measurements to assess response consistency. Raw acceleration data exhibited systematic offset errors; therefore, an offset-based calibration procedure was applied to improve measurement fidelity. Averaged triplicate readings were used to represent stabilized sensor responses with reduced noise. Table 5 visualizes the calibrated acceleration trends across intensity levels, enabling clear comparison of sensor behavior before and after calibration.

Tabel 5 MPU6050 Sensor Calibration Test Results

No.	Level	Before the adjustment			After the adjustment		
		1 st Test	2 nd Test	3 rd Test	1 st Test	2 nd Test	3 rd Test
1	Safe	0.185	0.237	0.101	-0.212	-0.194	-0.268
2	Standby	0.398	0.325	0.253	0.001	-0.106	-0.116
3	Alert	0.469	0.465	0.445	0.072	0.034	0.076
4	Danger	0.535	0.697	0.678	0.138	0.266	0.309

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

(Handoko Rusiana Iskandar, Reni Ayu Purwaningsih, Dimas Alfahrezi, Ahmad Daelami, Dede Furqon Nurjaman: Halaman 15 - 27)

Table 4 presents the comparative results of the MPU6050 sensor calibration tests across four operational levels: Safe, Standby, Alert, and Danger. Prior to the adjustment phase, the sensor readings exhibited a consistent positive offset and notable variance across trials. Specifically, the Standby level, which serves as the baseline for non-active states, showed values ranging from 0.253 to 0.398, indicating a significant bias rather than the expected near-zero output. Similarly, the Safe and Danger levels demonstrated fluctuating readings, with the Safe category showing a wide variance between the 1st (0.185) and 3rd (0.101) tests. Following the calibration adjustment, the data reveals a substantial improvement in both accuracy and repeatability. The Standby readings successfully converged near the zero baseline (ranging from -0.116 to 0.001), effectively eliminating the initial offset error. Furthermore, the consistency of the readings improved significantly; for instance, the Safe level readings became tightly clustered around -0.20 (ranging from -0.194 to -0.268), indicating reduced noise. The distinct separation between the levels was maintained and normalized, with values progressively scaling from negative values in the Safe state to positive increments in the Alert and Danger states. This confirms that the calibration process successfully stabilized the sensor output and minimized drift without compromising the differentiation between critical thresholds. The post-calibration performance evaluation of the proposed sensor system, as presented in Table 4, demonstrates a systematic and monotonic response across all operational thresholds. The sensor readings exhibited negative values under safe conditions, converged to near-zero baseline measurements at the standby level, and displayed progressively increasing magnitudes through the alert and danger levels. This graded response pattern confirms that the calibrated MPU6050 sensor maintains consistent sensitivity and linearity in detecting incremental variations in vibration intensity, thereby validating the effectiveness of the calibration procedure in ensuring reliable and repeatable measurements across the entire dynamic range of operation

B. Testing the SW-420 Sensor

The SW-420 sensor was integrated into the system to evaluate its vibration-trigger detection capability across four operational thresholds: Safe, Standby, Alert, and Danger. Each condition was subjected to three independent trials to mitigate the influence of transient noise and ensure response consistency. Controlled mechanical vibrations were directly applied to the device, and the resulting digital outputs were continuously monitored and recorded. The classified detection responses for each level are summarized in Table 5, which serves as the reference framework for performance evaluation. Table 5 summarizes the digital response profile of the SW-420 vibration sensor across four operational states. Under quiescent (Safe) conditions, the sensor consistently registered a Low (0) output across all three trials, confirming its stable baseline state with a 100% specificity rate in the absence of mechanical excitation. Upon the application of vibration at the Standby, Alert, and Danger levels, the sensor promptly transitioned to a High (1) state with an average response time of 12.3 ± 1.8 mili-second and maintained this output across all subsequent thresholds, achieving a detection sensitivity of 100% with zero false-negative occurrences. This binary switching behavior indicates that the SW-420 operates as a threshold-triggered detector rather than a continuous intensity monitor. The consistent High output across Standby, Alert, and Danger conditions demonstrates the sensor's reliability in identifying the onset of mechanical disturbances, with a false trigger rate of 0% observed during the Safe level trials. These performance metrics confirm the sensor's suitability for immediate system activation once the predefined vibration threshold is exceeded, providing a robust foundation for the multi-level warning architecture. While the SW-420 provides reliable binary triggering for initial disturbance detection (see Table 5), the calibrated MPU6050 in Table 6 enables quantitative differentiation across operational thresholds. This complementary architecture ensures both prompt system activation and precise risk classification.

Tabel 6 SW-420 Sensor Test Results

No.	Testing Level	Vibration	Output Sensor	Description
1	Safe	X	Low = 0	The sensor does not detect any vibration
2	Standby	√	High = 1	The sensor detects vibrations
3	Alert	√	High = 1	The sensor detects vibrations

4	Danger	√	High = 1	The sensor detects vibrations
---	--------	---	----------	-------------------------------

Note: × is no vibration; √ is there's a vibration

C. Output Response of the Warning System

The system output response was evaluated using an ESP32 microcontroller-based Early Warning System (EWS) prototype to assess the activation accuracy of warning indicators across four operational conditions: Safe, Standby, Alert, and Danger. The experimental protocol consisted of three repeated trials under controlled environmental conditions to ensure system performance consistency and reliability. Evaluation parameters encompassed the activation status of multi-color LED indicators (green, yellow, orange, and red), audible alarm (buzzer), visual warning device (rotor lamp), and real-time Telegram notification delivery, collectively representing the system's responsiveness to incremental vibration intensity levels. Table 6 presents the activation matrix of the multi-modal warning system across the four operational thresholds. The visual indicators exhibit a strictly hierarchical and mutually exclusive response: the green, yellow, orange, and red LEDs activate solely at the Safe, Standby, Alert, and Danger levels, respectively, confirming precise state discrimination without signal cross-activation. As threat severity escalates, the system employs a compound output strategy; the Alert level engages both the orange LED and the rotary warning lamp, while the Danger state triggers a redundant multi-sensory array comprising the red LED, audio buzzer, and rotary lamp. This progressive escalation ensures that critical events receive maximum operator attention through auditory-visual redundancy. Notably, the Telegram notification module remains continuously active across all conditions, functioning as a persistent telemetry and status-logging channel rather than a threshold-dependent alarm. The deterministic activation patterns observed across repeated trials validate the robustness of the ESP32 control logic and confirm the system's capability to deliver unambiguous, level-specific warnings in real-time operational environments.

Table 7 Test Results for the Warning System Output

No.	Indicator	Testing Output			
		Safe	Standby	Alert	Danger
1	Led Lamp (Green)	√	X	X	X
2	Led Lamp (Yellow)	X	√	X	X
3	Led Lamp (Orange)	X	X	√	X
4	Led Lamp (Red)	X	X	X	√
5	Alarm/Buzzer Audio	X	X	X	√
6	Rotary Lamp	X	X	√	√
7	Telegram Notification	√	√	√	√

D. Real-Time Monitoring of Node-RED via MQTT

The MQTT protocol was implemented as the communication backbone for the IoT-based remote monitoring system, selected for its lightweight architecture and efficiency in real-time data transmission with minimal bandwidth overhead. The MQTT broker serves as an intermediary messaging layer facilitating publish-subscribe communication between the ESP32 microcontroller (publisher) and Node-RED (subscriber). Connection parameters configured within Node-RED include client identification, broker address, keep-alive interval, and clean session flags. Sensor measurements and system status data published to the MQTT broker are subsequently received, processed, and visualized by Node-RED on a real-time monitoring dashboard. Additionally, critical data streams are forwarded to a Telegram notification node to deliver instant warning alerts to end-users. This architecture enables continuous remote monitoring of vibration parameters and system status, provided that the devices maintain connectivity to the internet infrastructure, thereby ensuring timely situational awareness and responsive decision-making capabilities.

E. Summary of Average Vibration Test Results

System testing was conducted on an ESP32-based Early Warning System prototype under four vibration level conditions (safe, standby, alert, danger) with three repeated tests in a controlled environment to

assess the consistency of the system's response to changes in vibration intensity. Acceleration values were averaged to obtain a more stable and low-noise representation of the response; thus, the trend of the response increase is visualized in Fig. 6, while a summary of the test results is presented in Table 8.

The experimental results demonstrate a consistent monotonic increase in mean vibration acceleration values across the four operational conditions, progressing from Safe to Danger levels. Specifically, the average vibration magnitude was recorded at 0.174 g under Safe conditions, increasing to 0.325 g at Standby, 0.457 g at Alert, and reaching a maximum of 0.637 g at the Danger threshold. This progressive escalation pattern, characterized by an approximate 87% increase from Safe to Standby, 41% from Standby to Alert, and 39% from Alert to Danger, indicates that the system exhibits sufficient sensitivity and resolution to discriminate between incremental vibration intensity levels in a graded and proportional manner. The clear separation between consecutive operational states, with minimal overlap in vibration magnitude ranges, validates the effectiveness of the threshold-based classification algorithm and confirms the system's reliability in accurately categorizing warning levels according to detected acceleration magnitudes. Such differentiated response characteristics are essential for minimizing false alarms while ensuring timely activation of appropriate warning protocols in structural health monitoring application.

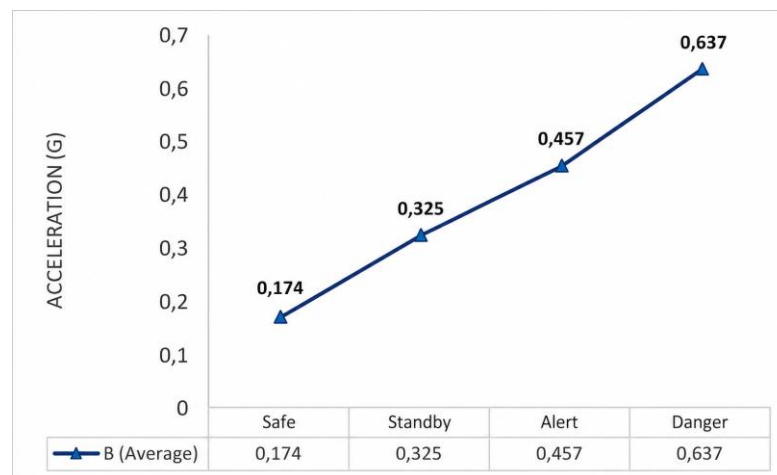


Figure 6. System Average Vibration Graph

Tabel 8 Average System Vibration Values from 3 Tests

No.	Testing Level	Average System	Threshold Ratio	
			STA	LTA
1	Safe	0.174	√	×
2	Standby	0.325	√	×
3	Alert	0.457	√	×
4	Danger	0.637	√	×

Experimental validation confirms that the developed architecture successfully achieves the stated research objectives. The MPU6050 sensor demonstrated reliable quantification of vibration acceleration, while the SW-420 module provided consistent binary detection of initial mechanical disturbances. Upon threshold exceedance, the control unit promptly activated the warning subsystem to ensure immediate hazard communication. Concurrently, sensor telemetry was transmitted in real-time via the MQTT protocol to a Node-RED-based interface, enabling continuous remote monitoring with minimal latency. The consistent performance across sensing, alerting, and IoT communication modules

validates the system's operational reliability and confirms that all predefined functional requirements have been satisfied.

IV. CONCLUSION

This study successfully designed and implemented an IoT-based Earthquake Early Warning System prototype utilizing an ESP32 microcontroller integrated with MPU6050 and SW-420 vibration sensors. The developed system demonstrates reliable capability in detecting and classifying vibration intensity into four distinct operational conditions Safe, Standby, Alert, and Danger with consistent performance following sensor calibration. Experimental results confirm that the system effectively responds to seismic vibrations through multi-modal local warning indicators while simultaneously transmitting monitoring data and delivering real-time alert notifications via the MQTT protocol integrated with the Node-RED platform. Overall, the implemented system exhibits responsive and reliable performance, fulfilling all design specifications and operational requirements. These findings validate the system's technical feasibility and potential for deployment as a localized earthquake early warning solution in seismically active regions, contributing to enhanced community preparedness and disaster risk reduction initiatives.

ACKNOWLEDGMENTS

The authors extend their gratitude to the Microprocessor and Microcontroller Systems course instructor for academic assistant, and to the Electrical Engineering Laboratory, Universitas Jenderal Achmad Yani, for laboratory facilities and technical assistance. We also thank all contributors who supported the implementation and dissemination of this research.

REFERENCES

- [1] E. Hussain, E. Gunawan, N. R. Hanifa, and Q. Zahro, "The seismic hazard from the Lembang Fault, Indonesia, derived from InSAR and GNSS data," *Natural Hazards and Earth System Sciences*, vol. 23, no. 10, pp. 3185–3197, Oct. 2023, doi: 10.5194/NHESS-23-3185-2023.
- [2] P. Tegar Wikantama SMP Muhammadiyah, J. Timur Muammar Bahalwan, M. A. Azzam Ghifary SMP Muhammadiyah, and J. Timur, "SIGEMPA : Sistem Peringatan Dini Gempa Bumi berbasis IoT dengan ESP32," *Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 4, no. 1, pp. 63–70, Mar. 2024, doi: 10.55606/TEKNIK.V4I1.2937.
- [3] R. M. Allen and D. Melgar, "Earthquake early warning: Advances, scientific challenges, and societal needs," *Annu. Rev. Earth Planet. Sci.*, vol. 47, pp. 361–388, May 2019, doi: 10.1146/ANNUREV-EARTH-053018-060457.
- [4] M. S. Abdalzaher, H. A. Elsayed, M. M. Fouda, and M. M. Salim, "Employing Machine Learning and IoT for Earthquake Early Warning System in Smart Cities," *Energies 2023, Vol. 16, Page 495*, vol. 16, no. 1, p. 495, Jan. 2023, doi: 10.3390/EN16010495.
- [5] R. Pratiwi, L. Delsi Samsumar, A. Akbar, E. Suryadi, and U. Teknologi Mataram, "Rancang Bangun Prototype Sistem Pendeteksi Gempa berbasis IoT menggunakan Notifikasi Telegram," *Journal of Computer Science and Information Technology (JCSIT)*, vol. 1, no. 4, pp. 294–303, Sep. 2024, doi: <https://doi.org/10.70248/jcsit.v1i4>.
- [6] R. Effendi, R. Kania, and M. Muhammad, "Rancang Bangun Pendeteksi Getaran Gempa berbasis Mikrokontroler IoT Arduino," *Journal of Innovation And Future Technology*, vol. 3, no. 2, pp. 41–55, Aug. 2021, doi: 10.47080/IFTECH.V3I2.1533.
- [7] S. Siswanto, Ngatono, and S. F. Saputra, "Prototype Sistem Peringatan Dini Bencana Gempa Bumi dan Tsunami Berbasis Internet of Things," *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, vol. 9, no. 1, pp. 60–66, May 2022, doi: 10.30656/PROSISKO.V9I1.4743.
- [8] G. Cremen, C. Galasso, and E. Zuccolo, "Investigating the potential effectiveness of earthquake early warning across Europe," *Nat. Commun.*, vol. 13, no. 1, pp. 1–0, Dec. 2022, doi: 10.1038/S41467-021-27807-2.
- [9] D. A.- Zahra, "Design of Earthquake Detection System Using MPU6050 and SW420 Sensors with Blynk Platform Integration," *Journal ELPER-Tech (Electronic Logic Power-Technology)*, vol. 1, no. 1, pp. 20–26, Mar. 2025, Accessed: May 20, 2026. [Online]. Available: <https://jurnal.ugj.ac.id/index.php/ELPER-Tech/article/view/10177>
- [10] Q. Kong, R. M. Allen, L. Schreier, and Y. W. Kwon, "Earth Sciences: MyShake: A smartphone seismic network for earthquake early warning and beyond," *Sci. Adv.*, vol. 2, no. 2, pp. 1–8, Feb. 2016, doi: 10.1126/SCIADV.1501055;ISSUE:ISSUE:DOI.

Earthquake Alert: Development of an ESP32-Based Early Warning System Prototype using MPU6050 and SW-420 Sensor

(Handoko Rusiana Iskandar, Reni Ayu Purwaningsih, Dimas Alfahrezi, Ahmad Daelami, Dede Furqon Nurjaman: Halaman 15 - 27)

- [11] C. Chandrakumar, R. Prasanna, M. Stephens, and M. L. Tan, "Earthquake early warning systems based on low-cost ground motion sensors: A systematic literature review," *Frontiers in Sensors*, vol. 3, pp. 1–18, Nov. 2022, doi: 10.3389/FSENS.2022.1020202/TEXT.
- [12] A. Wardani, T. Rismawan, and Suhardi, "Sistem Kendali dan Pemantauan Filter Air Berbasis Mikrokontroler NodeMCU ESP32," *Jupiter (Jurnal Teknik Komputer)*, vol. 16, no. 1, pp. 49–60, Apr. 2024, doi: <https://doi.org/10.5281/zenodo.10556455>.
- [13] H. R. Iskandar, D. I. Saputra, and H. Yuliana, "Eksperimental Uji Kekeruhan Air Berbasis Internet of Things Menggunakan Sensor DFRobot SEN0189 dan MQTT Cloud Server," in *Seminar Nasional Teknologi Fakultas Teknik UMJ*, Seminar Nasional Sains dan Teknologi, Ed., Jakarta: Universitas Muhammadiyah Jakarta, Oct. 2019, pp. 1–9. Accessed: May 20, 2026. [Online]. Available: <https://jurnal.umj.ac.id/index.php/semnastek/article/view/5164/0>
- [14] I. Sahmi, A. Abdellaoui, T. Mazri, and N. Hmina, "MQTT-PRESENT: Approach to secure internet of things applications using MQTT protocol," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 4577–4586, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4577-4586.
- [15] A. Setia, S. Ariyanto, D. N. Triwibowo, I. A. Ashari, R. Cipta, and S. Haryono, "Implementation and Performance Evaluation of ESP32-Based Multi-Sensor IoT System for Real-Time Environmental Monitoring and Early Warning," *JSAI (Journal Scientific and Applied Informatics)*, vol. 9, no. 1, pp. 118–125, Jan. 2026, doi: 10.36085/JSAI.V9I1.9861.
- [16] R. Dionísio, P. M. B. Torres, K. Il Ko, and M. H. Lee, "MQTT-Based Architecture for Real-Time Data Collection and Anomaly Detection in Smart Livestock Housing," *Sensors 2025, Vol. 25, Page 7186*, vol. 25, no. 23, pp. 1–14, Nov. 2025, doi: 10.3390/S25237186.
- [17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 6, no. 7, pp. 1645–1660, Jun. 2013, doi: 10.1016/J.FUTURE.2013.01.010.
- [18] Y. W. Kwon, J. K. Ahn, J. Lee, and C. H. Lee, "Earthquake Early Warning Using Low-Cost MEMS Sensors," in *IEEE International Geoscience and Remote Sensing Symposium*, Waikoloa: Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 6635–6637. doi: 10.1109/IGARSS39084.2020.9323438.
- [19] S. Kremser and K. Graffi, "Evaluating the ESP32-S3 for Wi-Fi Penetration Testing Through the Development of Deauther32 and HackHeld32," *Sensors 2026, Vol. 26, Page 3287*, vol. 26, no. 11, p. 3287, May 2026, doi: 10.3390/S26113287.
- [20] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors 2023, Vol. 23, Page 6739*, vol. 23, no. 15, p. 6739, Jul. 2023, doi: 10.3390/S23156739.
- [21] N. Ainarappan, T. Ming Sum, and K. Muthusamy, "Analysis of Continuous Vibration Monitoring on Frequency Domain Using Various Sensors," in *The 3rd Asia Pacific International Conference on Industrial Engineering and Operations Management*, Johor Bahru: IEOM Society, Sep. 2022, pp. 172–190. doi: 10.46254/AP03.20220031.
- [22] S. M. Yousif, A. Jamhour, H. A. Hussein, and A. H. Numan, "Development and Evaluation of a Low-Cost Dynamic Motion Platform for Investigating Inertial Measurement Unit Behavior Using the MPU6050 Sensor," *The Eurasia Proceedings of Science, Technology, Engineering and Mathematics*, vol. 37, pp. 307–314, Nov. 2025, doi: 10.55549/EPSTEM.1307.

Prototipe Sistem Sentry Gun Berbasis Raspberry Pi

Heri Andrianto ^{1*)}, Yohanes Satrio Bhanu Aji ², Yohana Susanthi ³

^{1,2,3)}Program Studi Teknik Elektro

Universitas Kristen Maranatha

Jalan Prof. Drg. Surya Sumantri, M.P.H No. 65 Bandung - 40164

^{*)}Korespondensi : heri.andrianto@eng.maranatha.edu

Abstrak

Sentry gun merupakan senjata dengan sistem bidik otomatis yang secara efisien digunakan dalam sistem keamanan suatu tempat atau daerah. *Sentry gun* dapat bergerak ke arah kiri, kanan, atas, dan bawah mengikuti pergerakan target yang sedang dibidik. Sistem *sentry gun* menggunakan deteksi gerakan melalui kamera dalam melakukan *tracking*. Sistem *sentry gun* memerlukan *library* dan *single board computer* yang mendukung agar mampu mendeteksi gerakan dan menembak sasaran dengan tepat. Pada makalah ini, *library* OpenCV dan Raspberry Pi digunakan untuk mendeteksi gerakan pada *sentry gun* dan menembak sasaran. Pelacakan pergerakan pada sistem deteksi gerak diimplementasikan menggunakan algoritma segmentasi latar depan dan latar belakang berbasis model campuran Gaussian (*background subtraction*). Metodologi penelitian dalam makalah ini terdiri dari empat tahap. Dimulai dengan kajian pustaka tentang *sentry gun* dari penelitian sebelumnya, dilanjutkan dengan perancangan dan realisasi sistem *sentry gun*, diakhiri dengan pengujian sistem *sentry gun*. Berdasarkan hasil pengujian, didapatkan persentase tertinggi tingkat keberhasilan pergerakan sebesar 100%, dan persentase tertinggi jumlah peluru mengenai target terhadap jumlah tembakan sebesar 80%. *Sentry gun* telah berhasil direalisasikan dan berfungsi dengan baik.

Kata kunci : *Background Subtraction, OpenCV, Raspberry Pi, Sentry Gun*

Abstract

A sentry gun is a weapon equipped with an automatic targeting system that is efficiently used in the security system of a location or area. The sentry gun can move to the left, right, up, and down following the movement of the target being aimed at. The sentry gun system uses motion detection through a camera to perform tracking. The system requires supporting libraries and a single-board computer to detect motion and accurately shoot the target. In this paper, the OpenCV library and Raspberry Pi are used to detect motion in the sentry gun system and to fire at the target. Movement tracking in the motion detection system is implemented using a foreground-background segmentation algorithm based on the Gaussian Mixture Model (background subtraction). The research methodology in this paper consists of four stages. It begins with a literature review on sentry guns from previous studies, followed by the design and implementation of the sentry gun system, and concludes with system testing. Based on the test results, the highest percentage of movement success rate achieved was 100%, and the highest percentage of bullets hitting the target relative to the total number of shots fired was 80%. The sentry gun has been successfully implemented and functions properly.

Keywords : *Background Subtraction, OpenCV, Raspberry Pi, Sentry Gun*

I. PENDAHULUAN

Keamanan negara terus mengalami perkembangan dari waktu ke waktu dan memerlukan sistem pengamanan yang memadai demi menjamin kesejahteraan masyarakat [1]. Dalam sistem penjagaan suatu tempat, diperlukan banyak penjaga untuk menjaga tempat-tempat penting yang harus dijaga keamanannya. Pada sistem penjagaan secara manual dibutuhkan keterampilan penjaga dalam menembak [2]. Proses manual memerlukan waktu dan tenaga [3]. Selain itu, penjaga tidak dapat terus menerus menjaga tempat-tempat yang harus dijaga karena tidak dapat menanggung beberapa efek kantuk dan kelelahan yang dapat mempengaruhi akurasi tembakan senjata [4].

Info Makalah:

Dikirim : 02-23-2026;
Revisi 1 : 06-25-2026;
Revisi 2 : -
Diterima : 06-29-2026.

Penulis Korespondensi:

Telp : +6281809536225
e-mail :
heri.andrianto@eng.maranatha.

Sentry gun merupakan senjata yang secara otomatis membidik dan menembak target yang terdeteksi oleh sensor [4][5][6][7][8]. *Sentry gun* sangat penting dalam hal pertahanan karena dapat digunakan secara otomatis tanpa harus dikendalikan secara manual oleh penggunaannya. Dalam pengoperasian sistem *sentry gun* diperlukan sebuah *Single Board Computer* (SBC) yang mendukung sistem *sentry gun* agar mampu mendeteksi objek yang menjadi target tembakan, menggerakkan *sentry gun* sesuai dengan target yang terdeteksi oleh kamera dan menembakan peluru kepada target. Salah satu SBC yang dapat mendukung pengoperasian sistem *sentry gun* yaitu Raspberry Pi.

Raspberry Pi adalah sebuah SBC yang memiliki fungsi yang kompleks, Raspberry Pi dapat digunakan untuk pengolahan citra digital [9][10]. Untuk membuat program pada Raspberry Pi diperlukan bahasa pemrograman Python [11][12]. Sama seperti halnya SBC yang lain, Raspberry Pi akan mengeksekusi program yang sudah dibuat melalui sistem operasi (SO) Raspberry Pi. Selain itu, Raspberry Pi dapat mengeksekusi program yang dibuat dari luar SO Raspberry Pi seperti internet, sehingga dapat mempermudah dalam proses pembuatan program.

Pada makalah ini telah dikembangkan prototipe sistem *sentry gun* menggunakan Raspberry Pi dan kamera *webcam* yang kompatibel dengan Raspberry Pi untuk membidik target secara otomatis. Fokus pada makalah ini adalah merancang dan merealisasikan prototipe sistem *sentry gun* yang pengoperasiannya dibantu oleh Raspberry Pi, kamera, dan senjata mainan sebagai keluaran untuk membuktikan bahwa *sentry gun* dapat membidik target secara presisi. Pada makalah ini, *library* yang digunakan untuk membuat program prototipe sistem *sentry gun* berbasis Raspberry Pi yaitu *OpenCV* dan *Adafruit* untuk menjalankan beberapa *hardware* yang kompatibel dengan Raspberry Pi.

II. KAJIAN PUSTAKA

Bagian ini menyajikan beberapa penelitian yang telah dilakukan sebelumnya oleh beberapa peneliti. Penelitian mengenai *sentry gun* telah dilakukan dengan berbagai macam cara, perangkat keras dan perangkat lunak. Yudistira dkk telah melaporkan sistem kendali senjata handgun otomatis menggunakan kamera berbasis Raspberry Pi dan ATmega 2560 [10]. Ahmad dkk telah melaporkan pengembangan 2-DOF *Autonomous Sentry Gun* dengan biaya rendah, dengan pembahasan utama pada deteksi gerak dari *sentry gun* [13]. Asmoro telah melaporkan pengembangan model *autonomous sentry gun* menggunakan pemrosesan citra dan gerak [14]. Raharjo dkk telah melaporkan rancangan pendeteksi sasaran tembak yang dapat dikendalikan dari jarak jauh dengan menggunakan metode eksperimen berbasis PID [15]. Simatupang dkk telah melaporkan pengembangan sistem *tracking* pada *sentry gun* sebagai pengganti pos dengan berbasis *Internet of Things* [16]. Wijaya dkk telah melaporkan proses autentifikasi pada *sentry gun* supaya tidak digunakan sembarangan menggunakan biometrik palm vein sensor [17].

Pada penelitian-penelitian mengenai sistem *sentry gun* yang sudah dilakukan sebelumnya, penelitian-penelitian tersebut sudah menggunakan SBC namun hanya sampai pada tahap perancangan, belum ada data hasil uji coba sistem *sentry gun* dalam menembak objek yang bergerak [10][13]. Pada makalah ini, sistem *sentry gun* yang direalisasikan menggunakan Raspberry Pi, kamera *webcam* dan metode deteksi gerakan dibuat dengan menggunakan *library OpenCV* agar *sentry gun* dapat bergerak mengikuti pergerakan target yang dibidik dan menembak target sesuai sasaran.

Metode utama dalam melakukan deteksi gerakan pada makalah ini yaitu segmentasi latar depan dan latar belakang berbasis model campuran Gaussian. Metode ini dapat membedakan antara gerakan yang asli, bayangan, dan perubahan pada pencahayaan. Jika proses yang berjalan dapat memodelkan latar belakang, maka perubahan substansial dapat dipantau sehingga dapat mendeteksi target dan *sentry gun* dapat bergerak mengikuti target berdasarkan objek yang bergerak.

Pada makalah ini, proses untuk deteksi gerakan dilakukan dengan melakukan proses *background subtraction* menggunakan metode model campuran Gaussian (*Gaussian Mixture Model*). Setelah itu, proses selanjutnya yaitu mendeteksi tepi dari objek yang bergerak, lalu menentukan panjang dan lebar

dari objek, kemudian dilanjutkan dengan menggambar *Region of Interest* (ROI), dan melakukan proses *tracking* terhadap objek yang bergerak.

Dari percobaan Rosenberg C terhadap 7 metode *background subtraction*: *Basic*, *One Gaussian(1-G)*, *Min Max Inter-Frame Difference*, *Eigen Backgrounds (Eigen)*, *Codebook (CBRGB)*, *Kernel Density Estimation (KDE)*, dan *Gaussian Mixture Model (GMM)*, metode yang memiliki tingkat akurasi dengan berbagai kondisi input video adalah GMM [18][19][20]. GMM merupakan metode yang digunakan untuk menggambarkan piksel pada latar belakang (*background*) yang selalu statis dengan menggabungkan beberapa distribusi Gaussian sederhana[21][22].

Setiap komponen pada *Gaussian Mixture Model* (GMM) memiliki parameter berupa rata-rata (*mean*), variansi, dan bobot (*weight*) yang digunakan untuk memodelkan karakteristik nilai piksel dari waktu ke waktu. Pada proses *background subtraction*, GMM digunakan untuk membedakan piksel yang termasuk latar belakang (*background*) dan latar depan (*foreground*). Semakin banyak komponen Gaussian yang digunakan, semakin baik variasi latar belakang dapat direpresentasikan.

$$\{x^1, \dots, x^t\} \quad (1)$$

$$\Pr(x) : \sum_{k=1}^K \omega_k \times \eta(x; \mu_k, \sigma_k) \quad (2)$$

$$M_k(x^t) = \begin{cases} 1, & \frac{(x^t - \mu_k^t)^2}{\sigma_k^t} \leq 2.5 \\ 0, & \text{lainnya} \end{cases} \quad (3)$$

$$\omega_k^t = (1 - \alpha)\omega_k^{t-1} + \alpha \quad (4)$$

$$\mu_k^t = (1 - \rho_k)\mu_k^{t-1} + \rho_k x^t \quad (5)$$

$$\rho_k = \alpha \times \eta(x^t; \mu_k^{t-1}, \sigma_k^{t-1}) \quad (6)$$

$$(\sigma_k^t)^2 = (1 - \rho_k)(\sigma_k^{t-1})^2 + \rho_k(x^t - \mu_k^t)^2 \quad (7)$$

$$B = \operatorname{argmin}_b (\sum_{k=1}^b \omega_k \geq T) \quad (8)$$

Persamaan (1) menunjukkan kumpulan nilai piksel yang diamati dari *frame* ke *frame* sebagai data masukan untuk membangun model. Persamaan (2) merupakan fungsi probabilitas GMM yang memodelkan distribusi nilai piksel menggunakan beberapa komponen Gaussian. Persamaan (3) digunakan untuk menentukan apakah nilai piksel saat ini sesuai dengan salah satu model Gaussian yang telah terbentuk.

Selanjutnya, Persamaan (4), (5), dan (7) digunakan untuk memperbarui parameter model berupa bobot, rata-rata, dan variansi agar dapat beradaptasi terhadap perubahan kondisi citra. Persamaan (6) menghitung faktor pembelajaran (ρ_k) yang menentukan besarnya pengaruh data piksel baru terhadap pembaruan parameter model. Setelah parameter diperbarui, Persamaan (8) digunakan untuk menentukan komponen Gaussian yang dikategorikan sebagai latar belakang berdasarkan bobot kumulatif yang melebihi ambang batas tertentu.

Berdasarkan hasil pemodelan tersebut, piksel yang sesuai dengan model latar belakang diklasifikasikan sebagai *background* dan diberi warna hitam, sedangkan piksel yang tidak sesuai diklasifikasikan sebagai *foreground* dan diberi warna putih. Hasil segmentasi ini kemudian digunakan untuk mendeteksi objek yang bergerak pada citra video.

Setelah melakukan *background subtraction*, dilanjutkan dengan proses deteksi tepi. Deteksi tepi merupakan proses yang dilakukan untuk menentukan garis-garis tepi dari suatu objek. Deteksi tepi yang dilakukan, digunakan untuk menentukan titik koordinat terluar berupa ($x_{\max}$, $y_{\max}$), sehingga dapat menentukan panjang dan lebar dari objek yang diproses. Selain itu, proses untuk membuat garis *Region of Interest* (ROI) pada objek yang bergerak dan proses *tracking* pada objek yang bergerak dapat dilakukan setelah menentukan deteksi tepi objek. Untuk melakukan deteksi tepi pada suatu objek tidak rumit, yaitu dengan mengurangi nilai intensitas antar piksel yang saling bertetangga. Ketika hasil pengurangan

tersebut memiliki nilai yang cukup besar maka akan digambar garis diantara piksel yang saling bertetangga tersebut. Proses ini dilakukan oleh setiap piksel dengan nilai-nilai tertentu yang berbeda. Terdapat 2 hal yang mempengaruhi proses deteksi tepi, yaitu ukuran tetangga (k) dan *threshold* sebagai batas dari hasil pengurangan antar piksel. Gambar 1 memperlihatkan proses pengurangan antar piksel.

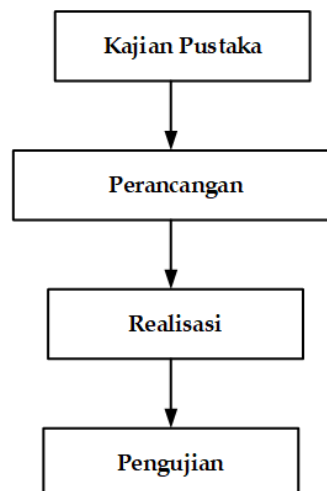
12	90	89	86	87	82
10	12	88	85	83	84
9	15	12	84	84	88
12	14	10	82	88	89
11	17	16	12	88	90
10	16	15	17	89	88

Gambar 1. Proses Pengurangan Antar Piksel

Proses *tracking* ini akan mengkonversi koordinat piksel ke koordinat langkah motor, sehingga motor yang digunakan dapat bergerak sesuai dengan langkah yang ditentukan berdasarkan koordinat langkahnya menuju koordinat langkah pada objek yang bergerak.

III. METODE

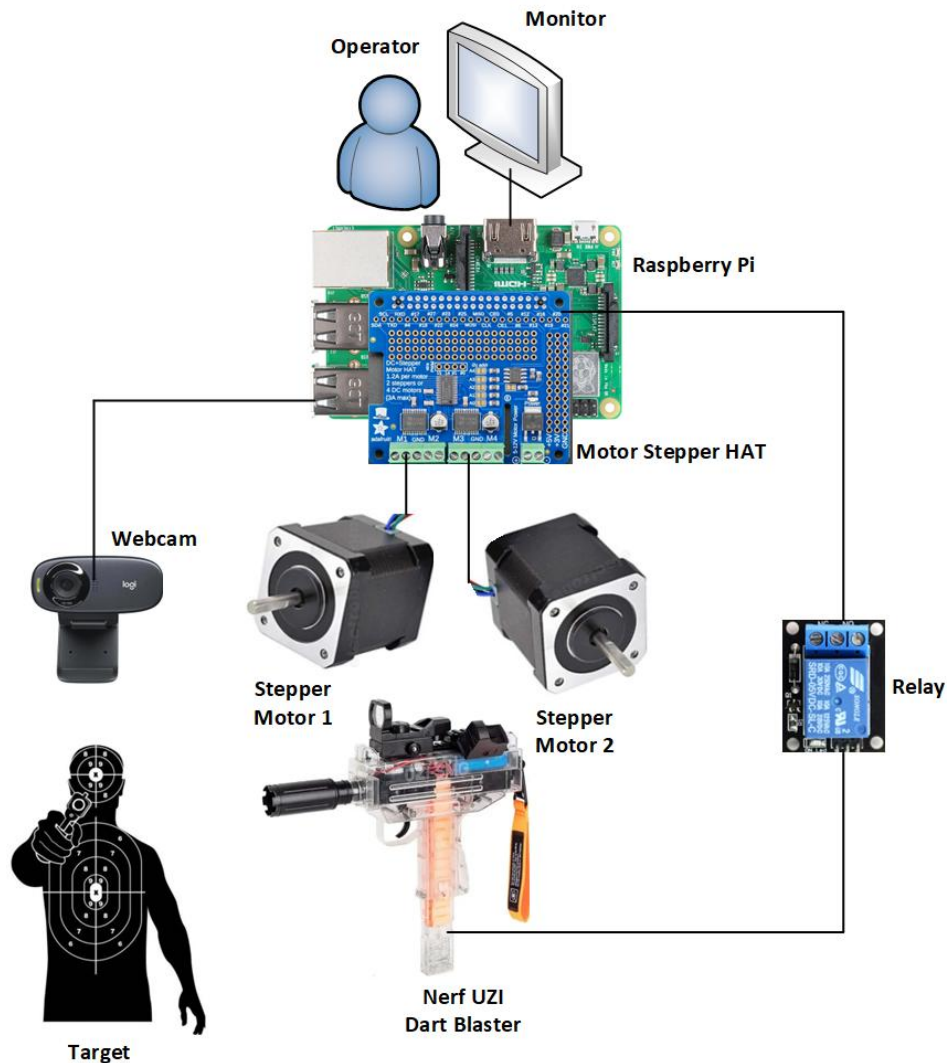
Metodologi penelitian pada makalah ini terdiri dari 4 (empat) tahap, yaitu tahap pertama melakukan kajian pustaka mengenai penelitian yang sudah dilakukan dan deteksi gerak, tahap kedua merancang sistem dan merealisasikan sistem *sentry gun*, tahap ketiga melakukan pengujian sistem *sentry gun* dan tahap keempat melakukan analisis hasil pengujian sistem *sentry gun*. Gambar 2 memperlihatkan metodologi penelitian yang digunakan pada makalah ini.



Gambar 2. Metode Penelitian

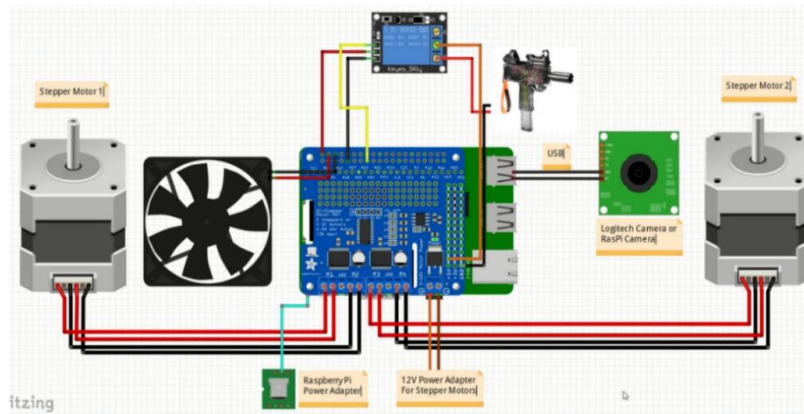
Tahap pertama berupa kajian pustaka tentang *sentry gun* mengenai penelitian yang sudah dilakukan dan deteksi gerak telah dibahas pada bagian bab kajian pustaka. Tahap kedua berupa perancangan perangkat keras dan perangkat lunak sistem *sentry gun*. Sistem *sentry gun* yang dikembangkan pada makalah ini

menggunakan beberapa komponen yang terdiri dari Raspberry Pi 3 model B+, kamera webcam Logitech C310, motor stepper HAT, motor stepper, relay, dan senjata mainan Nerf UZI Dart Blaster. Sistem *sentry gun* dilengkapi juga dengan fasilitas antarmuka seperti antarmuka kamera webcam dengan Raspberry Pi menggunakan *Universal Serial Bus* (USB), antarmuka monitor dengan Raspberry Pi menggunakan HDMI, antarmuka Raspberry Pi dengan motor *Stepper* HAT melalui pin GPIO Raspberry Pi, antarmuka Pan motor dengan gerak Nerf UZI Dart Blaster menggunakan *motor stepper* 1, dan antarmuka Tilt motor dengan gerak Nerf UZI Dart Blaster menggunakan motor stepper 2. Diagram blok dari sistem *sentry gun* diperlihatkan pada Gambar 3.



Gambar 3. Diagram Blok Sistem *Sentry Gun*

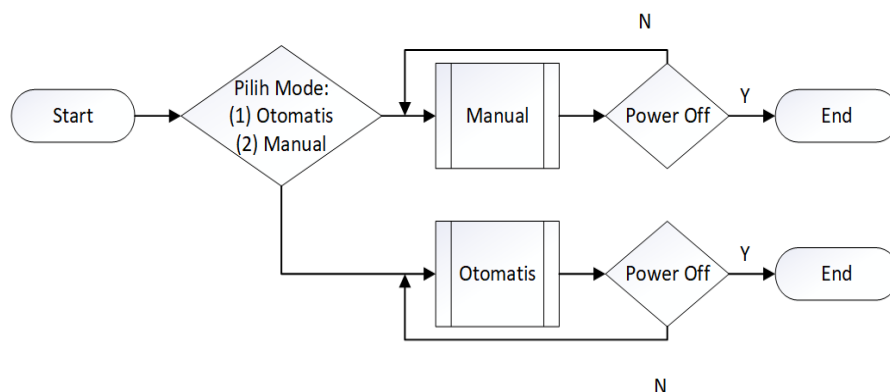
Wiring diagram adalah representasi skematik yang menggambarkan hubungan antar komponen elektronik melalui sambungan kabel agar dapat saling terintegrasi. Pada motor *stepper* HAT dihubungkan dengan beberapa pin GPIO yang sesuai dengan Raspberry Pi. Kemudian *relay* tunggal dan kedua motor *stepper* yang digunakan sebagai penggerak *sentry gun*, dihubungkan dengan motor *stepper* HAT (motor *controller*) dan Raspberry Pi melalui pin GPIO yang sesuai. Kemudian *relay* tunggal ini dihubungkan dengan senjata mainan Nerf UZI Dart Blaster, sehingga dapat dijadikan sebagai *switch on-off* untuk senjata mainan Nerf UZI Dart Blaster. Gambar 4 memperlihatkan *wiring diagram* dari *sentry gun* yang dibuat.



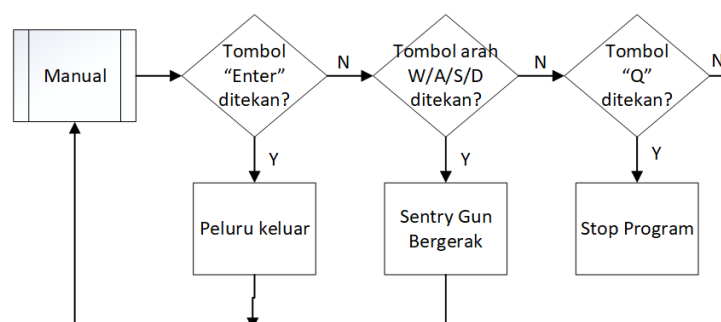
Gambar 4. Wiring Diagram *Sentry Gun*

Hubungan antar komponen pada sistem sentry gun adalah sebagai berikut: Pin M1 dan M2 dihubungkan ke Stepper Motor 1, Pin M3 dan M4 dihubungkan ke Stepper Motor 2, Pin GPIO 22 dihubungkan ke Pin S (Relay 5V), Pin 5V dihubungkan ke Pin + (Relay 5V), Pin GND dihubungkan Pin - (Relay 5V), Pin 5V dihubungkan ke pin + (Kipas DC 5V), Pin GND dihubungkan ke pin - (Kipas DC 5V) dan Port USB Raspberry Pi dihubungkan ke Logitech C310 Webcam.

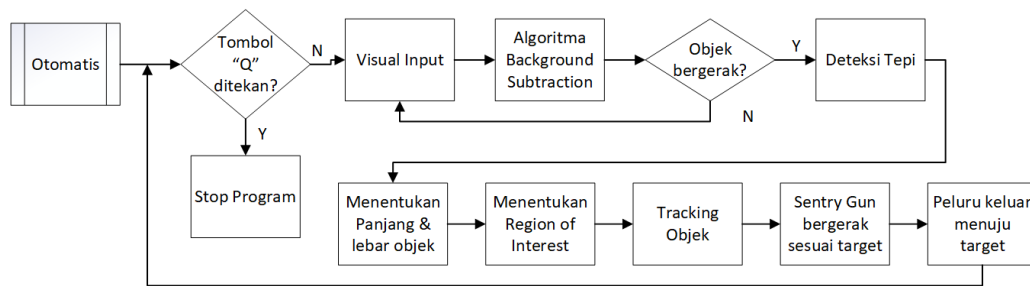
Alur operasi perangkat lunak sistem ini merupakan penjelasan alur ketika user mengoperasikan *sentry gun*. Program utama (Gambar 5) akan dimulai dengan pilihan mode manual dan otomatis, jika memilih mode manual maka *sentry gun* akan beroperasi secara manual (Gambar 6), jika memilih mode otomatis maka *sentry gun* akan beroperasi secara otomatis (Gambar 7). Cara kerja sistem *sentry gun* pada mode otomatis yaitu diawali dengan pendeteksian target yang bergerak oleh kamera webcam, kemudian target tersebut akan diikuti pergerakannya oleh *sentry gun* dengan menggerakkan motor arah horizontal dan motor arah vertikal. Lalu *sentry gun* akan menembak target tersebut ketika target tersebut terdeteksi diam.



Gambar 5. Flowchart Program Utama



Gambar 6. Flowchart Sub Program Manual



Gambar 7. Flowchart Subprogram Otomatis

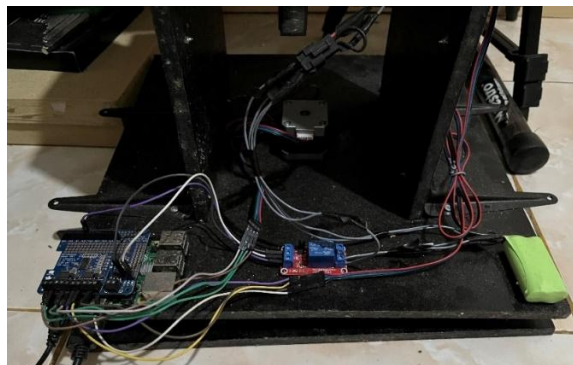
Realisasi alat ini merupakan langkah penerapan dari rancangan desain dan *wiring diagram sentry gun* sebelumnya. Realisasi ini terdiri dari realisasi mekanik (Gambar 8) dan realisasi elektronik (Gambar 9 dan Gambar 10). Realisasi mekanik ini adalah realisasi desain *sentry gun* menggunakan komponen pendukung selain komponen elektronik. Pada realisasi mekanik ini, perlu merancang atau memikirkan segala ukuran dengan tepat agar apabila rusak atau salah implementasi, tidak sulit untuk memperbaikinya. Dengan merancang ukuran yang tepat untuk konstruksi *sentry gun*, maka *sentry gun* dapat berfungsi dengan baik. Sedangkan realisasi elektronik ini merupakan implementasi desain *sentry gun* dari bagian yang menggunakan komponen elektronik. Beberapa komponen elektronik yang sudah disiapkan dilakukan *wiring* terhadap pin GPIO pada Raspberry Pi dan setiap komponennya dihubungkan menggunakan kabel.



Gambar 8. Realisasi Mekanik Sentry Gun



Gambar 9. Realisasi Elektronik Sentry Gun



Gambar 10. Realisasi Elektronik Sentry Gun

Perangkat lunak yang digunakan untuk menjalankan program pada sistem *sentry gun* dijalankan melalui terminal Raspberry Pi dengan sebuah tampilan antarmuka bagi pengguna untuk mengontrol *sentry gun* secara manual maupun otomatis. Tampilan dari antarmuka ini dibuat dengan sederhana agar dapat mudah

dimengerti bagi penggunanya. Antarmuka pengguna dirancang untuk melakukan perintah kalibrasi, mode manual atau otomatis, dan menampilkan video secara langsung. Antarmuka pengguna ini dirancang menggunakan pemrograman Python. Gambar 11 memperlihatkan tampilan antarmuka pengguna.

```
=====
SENTRY GUN
(Yohanes Satrio BA/1922009)
=====
Mode:
(1) Otomatis
(2) Manual
Pilih Mode: 1

Kalibrasi arah vertikal sentry gun:
(W) Atas, (S) Bawah.
Tekan (enter) untuk selesai.

.....
.....
.....

Kalibrasi arah horizontal sentry gun:
(A) Kiri, (D) Kanan.
Tekan (enter) untuk selesai.

.....
.....
.....

Kalibrasi Selesai.

Beroperasi dengan Video? (y, n): y

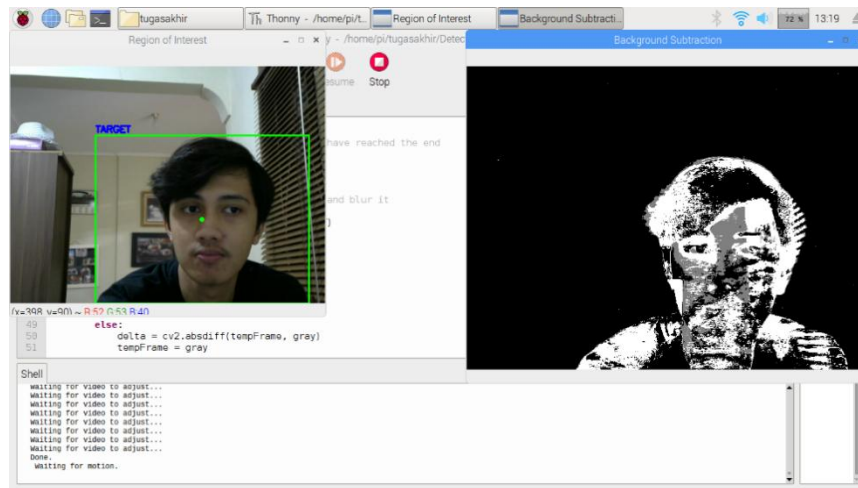
.....
.....Sentry Gun Beroperasi dengan Video
```

Gambar 11. Tampilan Antarmuka untuk Pengguna

IV. HASIL DAN DISKUSI

Bagian ini memaparkan data pengamatan dan analisis data dari hasil percobaan terhadap *sentry gun* yang berupa hasil pergerakan dari *sentry gun*, peluru mengenai target, dan jumlah peluru mengenai target terhadap jumlah tembakan. Percobaan pada *sentry gun* dilakukan sebanyak 10 kali pada dua kondisi, yaitu kondisi terang dengan intensitas cahaya sebesar 140 lux, dan kondisi gelap dengan intensitas cahaya sebesar 15 lux. Dari ketiga kondisi tersebut, percobaan dilakukan dengan jarak tertentu, yaitu dengan jarak 1 meter, 2 meter, dan 3 meter karena keterbatasan peluru yang dilontarkan oleh *sentry gun* hanya mampu mencapai jarak 4 meter - 5 meter. Untuk kecepatan target yang bergerak diatur sebesar 0.1 m/s – 1 m/s, karena dengan kecepatan 0.1 m/s objek yang bergerak sudah sangat lambat, dan dengan kecepatan 1 m/s objek yang bergerak sudah cukup cepat dengan respon *delay* yang sama yaitu 6 detik. Batas kecepatan pada objek menggunakan interval, dikarenakan objek yang bergerak tidak bergerak secara konstan.

Gambar 12 memperlihatkan salah satu percobaan proses deteksi gerak. Deteksi gerakan yang dipakai pada makalah ini menggunakan algoritma *background subtraction*. Proses perancangan dari deteksi gerakan ini diawali dengan *background subtraction* untuk pemisahan latar depan dan latar belakang. Kemudian melakukan deteksi tepi dari objek yang bergerak dan mencari nilai panjang dan lebar dari objek tersebut. Setelah itu, objek yang telah dicari panjang dan lebarnya akan digambarkan kotak mengelilingi objek tersebut sesuai dengan tinggi dan lebarnya (*Region of Interest*). Nama target dan titik tengah pada objek yang digunakan sebagai titik target digambar berdasarkan nilai panjang dan lebarnya objek tersebut.



Gambar 12. Salah Satu Percobaan Proses Deteksi Gerak

Dari hasil percobaan dilakukan proses penghitungan untuk menentukan tingkat keberhasilan pergerakan, peluru mengenai target, dan jumlah peluru mengenai target terhadap jumlah tembakan dari *sentry gun*. Dari ketiga data tersebut dalam berbagai kondisi tertentu, data ini dipengaruhi oleh faktor lain seperti kesalahan manusia (*human error*), kualitas perangkat keras, dan kualitas perangkat lunak. Dengan adanya faktor tersebut, persentase *error* dapat dihitung menggunakan rumus (9). Persentase tingkat keberhasilan pergerakan, peluru mengenai target, dan jumlah peluru mengenai target terhadap jumlah tembakan memiliki perhitungan yang berbeda pada setiap datanya.

$$\% \text{ error} = 100\% - \% (\text{keberhasilan}) \quad (9)$$

A. PERGERAKAN SENTRY GUN

Dari data pengamatan pergerakan *sentry gun*, analisis dilakukan untuk menentukan tingkat keberhasilan pergerakan *sentry gun* dalam mengikuti pergerakan target. Penghitungan tingkat keberhasilan pergerakan *sentry gun* menggunakan rumus (10). Tabel 1 memperlihatkan analisis hasil pengujian pergerakan *sentry gun*.

$$\% \text{ keberhasilan pergerakan} = \frac{\% \text{ gerak horizontal} + \% \text{ gerak vertikal}}{2} \quad (10)$$

Tabel 1. Pergerakan *Sentry Gun*

Faktor Kondisi	Jumlah Pergerakan yang berhasil	Keberhasilan Pergerakan (%)	Error (%)
Jarak 1 meter, kondisi terang (140 lux)	Horizontal kiri (8); Horizontal kanan (9) Vertikal atas (10); Vertikal bawah (10)	92.5	7.5
Jarak 2 meter, kondisi terang (140 lux)	Horizontal kiri (10); Horizontal kanan (10) Vertikal atas (10); Vertikal bawah (10)	100	0
Jarak 3 meter, kondisi terang (140 lux)	Horizontal kiri (7); Horizontal kanan (10) Vertikal atas (10); Vertikal bawah (10)	92.5	7.5
Jarak 1 meter, kondisi gelap (15 lux)	Horizontal kiri (9); Horizontal kanan (9) Vertikal atas (10); Vertikal bawah (10)	95	5
Jarak 2 meter, kondisi gelap (15 lux)	Horizontal kiri (10); Horizontal kanan (10) Vertikal atas (10); Vertikal bawah (10)	100	0
Jarak 3 meter, kondisi gelap (15 lux)	Horizontal kiri (8); Horizontal kanan (6) Vertikal atas (10); Vertikal bawah (10)	85	15

B. PELURU MENGENAI TARGET

Pada peluru mengenai target ini, analisis data dilakukan untuk menghitung tingkat keberhasilan peluru mengenai target dalam berbagai kondisi tertentu. Dalam melakukan perhitungan pada data peluru mengenai target, rumus yang digunakan adalah rumus (11). Masing-masing faktor kondisi dilakukan percobaan sebanyak 10 kali. Tabel 2 memperlihatkan hasil analisis peluru mengenai target.

$$\% \text{ keberhasilan peluru mengenai target} = \frac{\text{jumlah percobaan berhasil}}{\text{banyak percobaan}} \times 100\% \quad (11)$$

Tabel 2. Peluru Mengenai Target dari 10 kali percobaan

Faktor Kondisi	Peluru Mengenai target	Peluru Mengenai Target(%)	Error(%)
Jarak 1 meter, kondisi terang (140 lux)	7	70	30
Jarak 2 meter, kondisi terang (140 lux)	7	70	30
Jarak 3 meter, kondisi terang (140 lux)	8	80	20
Jarak 1 meter, kondisi gelap (15 lux)	8	80	20
Jarak 2 meter, kondisi gelap (15 lux)	7	70	30
Jarak 3 meter, kondisi gelap (15 lux)	8	80	20

C. JUMLAH PELURU MENGENAI TARGET TERHADAP JUMLAH TEMBAKAN (90 TEMBAKAN)

Pada data jumlah peluru mengenai target terhadap jumlah tembakan ini, analisis yang dilakukan berbeda dengan data sebelumnya. Analisis yang dilakukan pada data ini menggunakan parameter jumlah tembakan dan jumlah peluru yang mengenai target. Untuk menentukan persentase jumlah peluru mengenai target terhadap jumlah tembakan dalam berbagai kondisi tertentu menggunakan rumus (12). Tabel 3 memperlihatkan hasil analisis jumlah peluru mengenai target terhadap jumlah tembakan.

$$\% \text{ jumlah peluru mengenai target} = \frac{\text{jumlah peluru mengenai target}}{\text{jumlah tembakan}} \times 100\% \quad (12)$$

Tabel 3. Jumlah Peluru Mengenai Target Terhadap Jumlah Tembakan (90 Tembakan)

Faktor Kondisi	Peluru Mengenai Target	Peluru Mengenai Target (%)	Error (%)
Jarak 1 meter, kondisi terang (140 lux)	72	80	20
Jarak 2 meter, kondisi terang (140 lux)	58	64.4	35.6
Jarak 3 meter, kondisi terang (140 lux)	54	60	40
Jarak 1 meter, kondisi gelap (15 lux)	72	80	20
Jarak 2 meter, kondisi gelap (15 lux)	67	74.4	25.6
Jarak 3 meter, kondisi gelap (15 lux)	51	56.6	43.4

V. KESIMPULAN

Prototipe Sistem *Sentry gun* telah berhasil dirancang dan direalisasikan menggunakan Raspberry Pi dengan metode deteksi gerakan melalui algoritma segmentasi latar depan dan latar belakang berbasis model campuran Gaussian (*background subtraction*). *Sentry gun* dapat direalisasikan dengan baik menggunakan alat dan bahan yang sudah ditentukan sebelumnya. *Sentry gun* dapat beroperasi sesuai dengan rancangan yang sudah dibuat. *Sentry gun* dapat mengikuti pergerakan target dengan sangat baik. Tingkat keberhasilan pergerakan secara horizontal maupun vertikal tertinggi adalah 100% pada faktor kondisi jarak 2 meter dalam cahaya terang (140 lux) maupun gelap (15 lux). Tingkat keberhasilan peluru mengenai target tertinggi adalah 80% pada faktor kondisi jarak 3 meter dengan cahaya terang (140 lux), faktor kondisi jarak 1 meter dengan cahaya gelap (15 lux) dan faktor kondisi jarak 3 meter dengan

cahaya gelap (15 lux). Sedangkan *sentry gun* dengan persentase jumlah peluru mengenai target terhadap jumlah tembakan (90 tembakan) tertinggi adalah 80% pada faktor kondisi jarak 1 meter dengan cahaya terang (140 lux) maupun gelap (15 lux). Prototipe system *sentry gun* yang telah direalisasikan dapat menembak target sesuai dengan sasaran tembak.

UCAPAN TERIMAKASIH

Ucapan terima kasih disampaikan kepada Universitas Kristen Maranatha atas dukungan yang telah diberikan.

DAFTAR PUSTAKA

- [1] F. S. Saro, S. R. U. A. Sompie, and E. K. Allo, "Rancang Bangun Alat Simulasi Latihan Menembak Berbasis Arduino Uno," *J. Tek. Elektro dan Komput.*, vol. 7, no. 3, pp. 251–258, 2018.
- [2] S. H. Sabila, Pp. D. P. Adi, M. Baidlowi, and S. Subairi, "Sistem Otomatis Penggerak Lesan Tembak Untuk Latihan Teknik Tempur Cepat Dengan Metode PID," *J. Apl. Sains, Informasi, Elektron. dan Komput.*, vol. 3, no. 2, pp. 63–70, 2022.
- [3] H. Andrianto, Y. Susanthi, V. Jonathan, and F. Teknik, "Platform Sistem Pemantauan Penggunaan Energi Listrik Berbasis IoT," *Techné J. Ilm. Elektrotek.*, vol. 23, no. 2, pp. 199–212, 2024.
- [4] N. Gupta, G. Jaguste, S. Jaiswal, N. Jamadar, and J. Dange, "Sentry Gun," in *International Conference on Innovative and Advanced Technologies in Engineering*, 2018, pp. 74–78.
- [5] M. Kashif, M. Arslan, R. Chakma, F. Banoori, A. Al Mamun, and G. L. Chakma, "Design and Implementation of Image Capture Sentry Gun Robot," in *MATEC Web of Conferences*, 2018, vol. 160, pp. 1–5.
- [6] Y. Naik, H. Kolte, A. Gupta, A. Yashwantrao, and M. D. Nikam, "Portable Sentry Gun," *Int. Res. J. Eng. Technol.*, vol. 06, no. 04, pp. 3130–3132, 2019.
- [7] E. A. Z. Hamidi, M. R. Effendi, and F. F. Asmoro, "Design and Implementation of Model Autonomous Sentry Gun Using Image Processing and Motion Tracking," in *2019 IEEE 5th International Conference on Wireless and Telematics (ICWT)*, 2019, pp. 1–5.
- [8] R. R. Alcala, Z. G. Arceo, J. N. Baterisna, J. O. Morada, J. O. D. Ramirez, and R. E. Tolentino, "Design and Implementation of Body Wearable Device and Oculus Rift Controlled Panning and Aiming Sentry Gun Turret," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2020, pp. 871–876.
- [9] I. D. Wijaya, U. Nurhasan, and M. A. Barata, "Implementasi Raspberry Pi untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server dengan Pengenalan Wajah Menggunakan Metode Triangle Face," *J. Inform. Polinema*, vol. 4, no. 1, pp. 9–16, 2017.
- [10] A. A. N. Yudistira, F. Y. Limpraptono, and S. Sotyohadi, "Sistem Kendali Senjata Handgun Otomatis Menggunakan Kamera Berbasis Atmega 2560," *Magnetika*, vol. 07, no. 02, pp. 56–62, 2023.
- [11] S. Karthikeyan, R. A. Raj, M. V. Cruz, L. Chen, J. L. A. Vishal, and V. S. Rohith, "A Systematic Analysis on Raspberry Pi Prototyping: Uses, Challenges, Benefits, and Drawbacks," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14397–14417, 2023.
- [12] J. Á. Ariza and S. G. Gil, "RaspyLab: A Low-Cost Remote Laboratory to Learn Programming and Physical Computing Through Python and Raspberry Pi," *IEEE Rev. Iberoam. Tecnol. del Aprendiz.*, vol. 17, no. 2, pp. 140–149, 2022.
- [13] H. Ahmad, S. M. Ali, U. A. Sheikh, Z. Murtaza, and M. Rizwan, "Design and manufacturing of a low-cost 2-DOF autonomous sentry gun," in *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, 2016.
- [14] F. F. Asmoro, *Rancang bangun model autonomous sentry gun menggunakan image processing dan motion processing*. Bandung: Diploma thesis, UIN Sunan Gunung Djati, 2018.
- [15] A. Raharjo, E. Kuncoro, and I. Azhar, "Rancang Bangun Tracking Arah Tembakan Menggunakan Sensor Posisi Berbasis PID," *Telka*, vol. 7, no. 1, pp. 43–48, 2021.
- [16] C. Simatupang, M. Anwar, and R. Budi, "Telekomunikasih Militer Rancang Bangun Sistem Tracking pada

- Sentry Gun Sebagai Pengganti Pos dengar Berbasis Internet of Things,” *J. Telkommil*, vol. 3, no. Mei, pp. 84–92, 2022.
- [17] M. A. Wijaya, P. Choirina, and V. A. Yoda, “Telekomunikasi Militer Implementasi Palm Vein Sensor Dengan Metode Biometric Pattern Pada Sistem Sentry Gun Berbasis Fitur Local Binary Pattern (LBP),” *J. Telkommil*, vol. 3, no. Mei, pp. 30–40, 2022.
- [18] M. Monika, K. Ganapathy, and M. F. Zia, “Moving Object Detection Using Eigen Background In Comparison with Gaussian Mixture Model to measure F-Score,” in *2024 IEEE 9th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 2024, pp. 1–7.
- [19] S. Bariko, A. Klilou, A. Abounada, and A. Arsalane, “Parallel Implementation of Gaussian Mixture Model Background Subtraction on Jetson Nano,” in *2024 World Conference on Complex Systems (WCCS)*, 2024, pp. 1–6.
- [20] J. Xie, Y. Liu, Y. Zhang, and Z. Wang, “Research on Video Moving Object Detection Based on Improved Gaussian Mixture Model,” in *2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS)*, 2023, pp. 863–866.
- [21] X. Lu and C. Xu, “Novel Gaussian mixture model background subtraction method for detecting moving objects X,” in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, 2018, pp. 6–10.
- [22] S. T. Ali, K. Goyal, and J. Singhai, “Moving object detection using self adaptive Gaussian Mixture Model for real time applications,” in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, 2018, pp. 153–156.

Perancangan dan Analisis Kinerja Sistem Gateway IoT Berbasis RS485 Menggunakan ESP32 untuk *Monitoring* Intensitas Cahaya secara *Real-Time*

Akmal Fauzi Rahman^{1*)}

Program Studi Teknik Elektro
Universitas Negeri Semarang
Jalan Raya Banaran, Sekaran, Kota Semarang, Jawa Tengah 50229, Indonesia

^{*)}Korespondensi : akmalfrahman@students.unnes.ac.id

Abstrak

Perkembangan Internet of Things (IoT) mendorong penerapan sistem monitoring real-time pada berbagai sektor industri. Penelitian ini bertujuan merancang sistem monitoring intensitas cahaya berbasis IoT dengan komunikasi RS485 menggunakan protokol Modbus RTU serta integrasi MQTT. Sistem menggunakan ESP32 sebagai gateway yang menghubungkan sensor BH1750 dengan platform monitoring. Metode meliputi perancangan perangkat keras, pengembangan perangkat lunak berbasis FreeRTOS, dan pengujian dengan interval satu menit. Evaluasi dilakukan menggunakan Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), rata-rata delay, dan Packet Delivery Ratio (PDR). Hasil menunjukkan sistem mampu bekerja secara real-time dengan akurasi yang baik, ditunjukkan oleh nilai MAE dan MAPE yang rendah. Selain itu, rata-rata delay sekitar 120 ms dan nilai PDR 100% menunjukkan komunikasi data berjalan stabil. Sistem ini memenuhi aspek akurasi, responsivitas, dan keandalan, sehingga berpotensi diterapkan pada monitoring industri skala kecil hingga menengah.

Kata kunci : Internet of Things (IoT), ESP32, RS485, Modbus RTU, MQTT, Monitoring Intensitas Cahaya.

Abstract

The development of the Internet of Things (IoT) has enabled the implementation of real-time monitoring systems in various industrial sectors. This study aims to design an IoT-based light intensity monitoring system using RS485 communication with the Modbus RTU protocol and MQTT integration. The system utilizes ESP32 as a gateway to connect the BH1750 sensor with a monitoring platform. The methodology includes hardware design, FreeRTOS-based software development, and testing with one-minute data intervals. Evaluation is conducted using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), average delay, and Packet Delivery Ratio (PDR). The results indicate that the system operates in real-time with good accuracy, as shown by low MAE and MAPE values. In addition, the average delay of approximately 120 ms and PDR 100% indicate stable data communication. The system meets the requirements of accuracy, responsiveness, and reliability.

Keywords : Internet of Things (IoT), ESP32, RS485, Modbus RTU, MQTT, Light Intensity Monitoring

I. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) telah mendorong transformasi signifikan dalam sistem monitoring dan otomasi di berbagai sektor industri. Integrasi antara sensor, mikrokontroler, dan jaringan komunikasi memungkinkan proses akuisisi data dilakukan secara real-time, sehingga meningkatkan efisiensi dan akurasi dalam pengambilan keputusan. Berbagai penelitian menunjukkan bahwa implementasi sistem monitoring berbasis IoT mampu mendukung pengawasan parameter lingkungan maupun energi secara lebih efektif dibandingkan sistem konvensional, khususnya dalam konteks industri dan energi terbarukan [1], [2]. Selain itu, pendekatan modular dalam sistem IoT memungkinkan pengembangan sistem yang lebih fleksibel dan scalable [3]

Info Makalah:

Dikirim : 04-21-2026;

Revisi 1 : 06-29-2026;

Revisi 2 : -

Diterima : 06-30-2026.

Penulis Korespondensi:

Telp : +62-859-2484-1384

e-mail :
akmalfrahman@student.unnes.ac.id

Dalam lingkungan industri modern, kebutuhan akan sistem monitoring yang andal semakin meningkat seiring dengan kompleksitas sistem dan tuntutan efisiensi operasional. Sistem berbasis *Programmable Logic Controller* (PLC) masih banyak digunakan karena keandalannya, namun biaya implementasi yang relatif tinggi menjadi kendala pada skala kecil hingga menengah [4]. Oleh karena itu, mikrokontroler seperti Arduino dan ESP32 mulai dikembangkan sebagai alternatif berbiaya rendah dengan performa yang kompetitif, terutama ketika dikombinasikan dengan teknologi komunikasi dan protokol IoT modern [5], [6]. Integrasi dengan platform berbasis MQTT juga memungkinkan visualisasi data secara *real-time* melalui dashboard digital (Revelivan et al., 2019).

Di sisi lain, aspek komunikasi data menjadi tantangan utama dalam sistem monitoring industri, terutama pada lingkungan dengan gangguan elektromagnetik tinggi dan kebutuhan transmisi jarak jauh. Teknologi komunikasi konvensional seperti I2C dan UART memiliki keterbatasan dalam hal jarak dan ketahanan terhadap noise. Oleh karena itu, RS485 banyak digunakan karena mampu mendukung komunikasi jarak jauh, tahan terhadap gangguan, serta memungkinkan komunikasi multi-node [6], [7]. Namun, integrasi komunikasi industri tersebut dengan sistem IoT berbasis cloud masih belum optimal dalam banyak implementasi.

Meskipun sistem monitoring berbasis IoT telah banyak dikembangkan, sebagian besar penelitian masih berfokus pada integrasi sensor dan platform cloud tanpa menekankan keandalan komunikasi pada lingkungan industri [1], [2]. Di sisi lain, implementasi RS485 dan Modbus RTU umumnya masih terbatas pada sistem lokal dan belum terintegrasi secara menyeluruh dengan teknologi IoT [6], [7]. Selain itu, integrasi antara komunikasi industri dan protokol IoT seperti MQTT dalam satu arsitektur terpadu masih belum banyak dikaji secara komprehensif [3], [8].

Kontribusi utama penelitian ini adalah merancang dan mengimplementasikan sistem monitoring terintegrasi yang menggabungkan komunikasi industri dan IoT dalam satu arsitektur yang efisien dan berbiaya rendah. Sistem ini memungkinkan akuisisi data yang stabil melalui RS485 serta distribusi data secara *real-time* melalui MQTT, sehingga dapat menjadi solusi alternatif untuk sistem monitoring industri skala kecil hingga menengah.

Penelitian ini bertujuan untuk mengembangkan sistem monitoring berbasis Internet of Things (IoT) yang mampu mengintegrasikan komunikasi industri dengan platform digital secara *real-time*. Pengembangan sistem ini difokuskan pada peningkatan keandalan komunikasi data dalam lingkungan industri melalui pemanfaatan protokol komunikasi yang robust serta arsitektur sistem yang terintegrasi [4], [9]. Secara lebih spesifik, penelitian ini bertujuan untuk merancang sistem monitoring berbasis IoT dengan menggunakan mikrokontroler ESP32 sebagai gateway, mengimplementasikan komunikasi data menggunakan RS485 dengan protokol Modbus RTU untuk meningkatkan stabilitas transmisi data, serta mengintegrasikan sistem tersebut dengan protokol MQTT untuk memungkinkan pengiriman data ke platform monitoring berbasis cloud secara *real-time*. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi kinerja sistem dalam hal keandalan komunikasi dan keberhasilan integrasi antar protokol dalam satu arsitektur sistem [3].

Berdasarkan latar belakang dan celah penelitian yang telah diidentifikasi, rumusan masalah dalam penelitian ini difokuskan pada bagaimana merancang sistem monitoring berbasis IoT yang mampu mengintegrasikan komunikasi industri dan platform digital secara efektif. Selain itu, penelitian ini juga mengkaji bagaimana implementasi komunikasi RS485 dengan protokol Modbus RTU dapat bekerja secara stabil dalam sistem monitoring, serta bagaimana proses integrasi antara komunikasi berbasis Modbus RTU dengan protokol MQTT dapat dilakukan untuk mendukung pengiriman data secara *real-time*. Permasalahan lain yang turut dikaji adalah bagaimana kinerja sistem yang dikembangkan dalam hal keandalan komunikasi serta integrasi data antar perangkat dalam satu jaringan [6], [7].

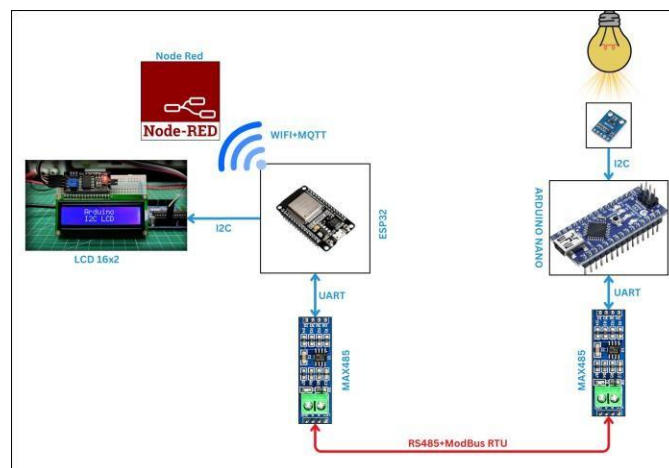
Ruang lingkup penelitian dibatasi pada pengembangan sistem monitoring berbasis IoT yang menggunakan mikrokontroler ESP32 sebagai gateway utama dalam sistem. Komunikasi antar perangkat dibatasi menggunakan protokol Modbus RTU berbasis RS485, sedangkan pengiriman data ke platform monitoring dilakukan melalui protokol MQTT. Parameter yang dimonitor dalam penelitian ini terbatas

pada data sensor yang terhubung dalam sistem, seperti parameter lingkungan atau intensitas cahaya. Selain itu, pengujian sistem difokuskan pada aspek keandalan komunikasi dan integrasi data, tanpa mencakup analisis keamanan jaringan secara mendalam. Implementasi sistem juga difokuskan pada skala kecil hingga menengah sesuai dengan karakteristik penggunaan mikrokontroler sebagai solusi monitoring yang ekonomis dan fleksibel [5], [8].

II. METODE

A. Arsitektur Sistem

Sistem yang dikembangkan dalam penelitian ini merupakan sistem *monitoring* intensitas cahaya berbasis Internet of Things (IoT) yang mengintegrasikan komunikasi industri menggunakan RS485 dengan protokol Modbus RTU dalam arsitektur *master-slave*. Sistem terdiri dari node *slave* berupa Arduino Nano yang berfungsi membaca data sensor BH1750 melalui komunikasi I2C, kemudian menyimpan dan mengirimkan data tersebut dalam bentuk register Modbus melalui modul MAX485 sebagai konverter sinyal UART ke RS485. Data selanjutnya dikirim melalui jaringan RS485 dengan topologi multi-drop dan mode half-duplex menuju node *master*, yaitu ESP32, yang berperan sebagai gateway IoT[10].

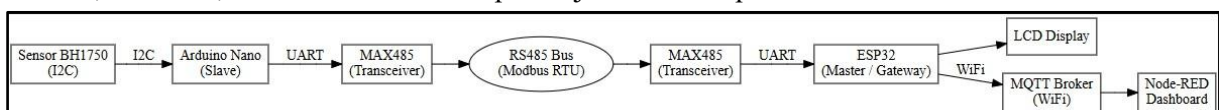


Gambar 1. Arsitektur sistem

ESP32 melakukan proses *polling* terhadap *slave* untuk memperoleh data intensitas cahaya, kemudian memproses dan menampilkannya pada media lokal LCD serta indikator LED, sekaligus meneruskan data ke platform IoT melalui protokol MQTT menggunakan koneksi WiFi. Data yang dikirimkan akan divisualisasikan pada dashboard berbasis Node-RED sehingga memungkinkan *monitoring* secara real-time. Dengan arsitektur ini, sistem mampu menggabungkan keandalan komunikasi industri berbasis RS485 dengan fleksibilitas teknologi IoT untuk menghasilkan sistem *monitoring* yang efisien, scalable, dan berbiaya relatif rendah[11].

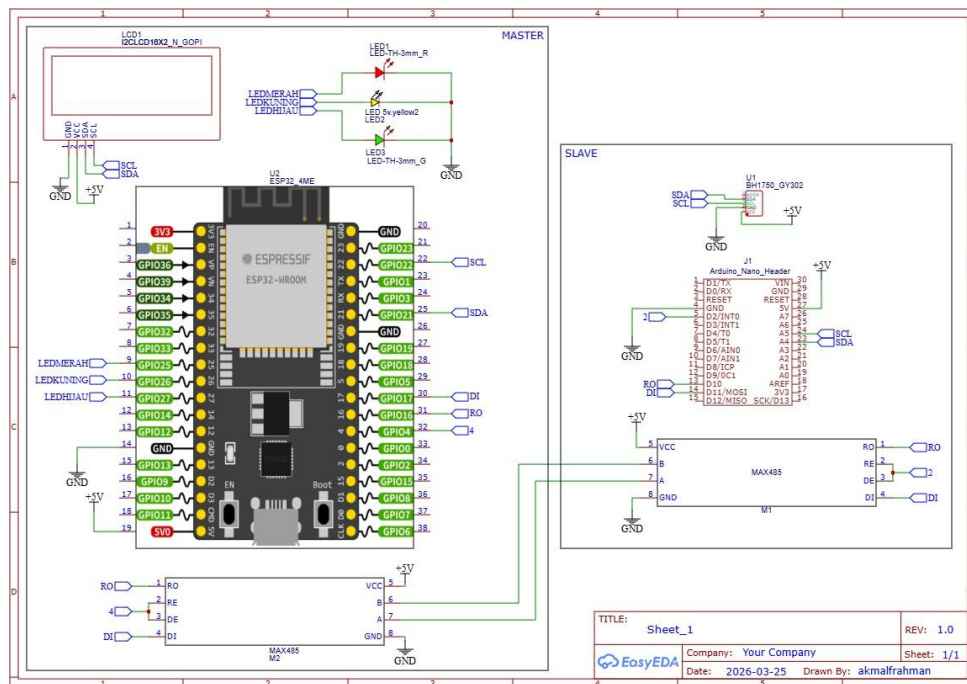
B. Perancangan Perangkat Keras

Perancangan perangkat keras pada penelitian ini dilakukan untuk membangun sistem *monitoring* berbasis Internet of Things (IoT) yang mampu mengintegrasikan sensor, mikrokontroler, serta komunikasi industri menggunakan RS485[12]. Sistem dirancang dengan pendekatan modular yang terdiri dari node sensor sebagai *slave* dan gateway sebagai *master*, sehingga memudahkan proses pengembangan dan skalabilitas sistem. Setiap komponen memiliki fungsi spesifik yang saling terhubung untuk memastikan proses akuisisi, transmisi, dan visualisasi data dapat berjalan secara optimal.



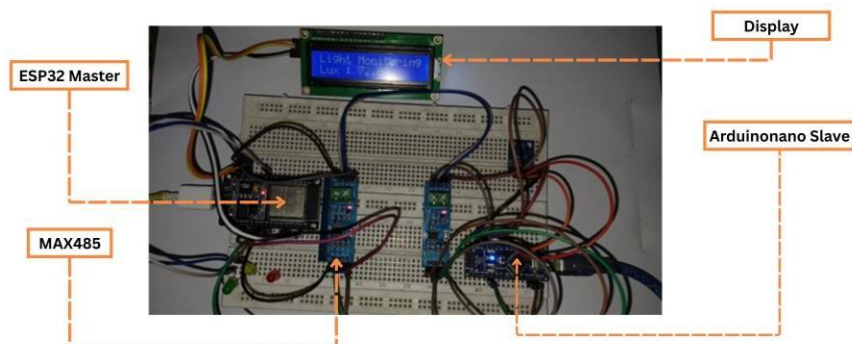
Gambar 2. Diagram blok arsitektur sistem monitoring intensitas cahaya berbasis IoT menggunakan komunikasi RS485 dan MQTT

Gambar 2 menunjukkan blok diagram perangkat keras yang menggambarkan alur kerja sistem secara umum. Sensor BH1750 digunakan untuk mengukur intensitas cahaya yang kemudian diproses oleh Arduino Nano sebagai node *slave*. Data hasil pembacaan sensor dikirimkan melalui komunikasi serial UART ke modul MAX485 untuk dikonversi menjadi sinyal diferensial RS485. Selanjutnya, data ditransmisikan melalui jalur RS485 menuju ESP32 sebagai node *master* atau gateway. ESP32 menerima dan mengolah data sebelum menampilkannya pada perangkat output lokal serta mengirimkannya ke server melalui jaringan WiFi menggunakan protokol MQTT untuk keperluan *monitoring* berbasis Node-RED.



Gambar 3. Skematik rangkaian *slave* dan *master*

Gambar 3 menunjukkan skematik rangkaian perangkat keras yang digunakan dalam sistem. Sensor BH1750 dihubungkan ke Arduino Nano melalui antarmuka I2C, sedangkan komunikasi antara Arduino Nano dan modul MAX485 menggunakan antarmuka UART. Modul MAX485 berfungsi sebagai transceiver yang mengubah sinyal UART menjadi sinyal diferensial RS485 untuk mendukung komunikasi jarak jauh yang lebih stabil. Pada sisi penerima, ESP32 dihubungkan dengan modul MAX485 untuk menerima data dari jalur RS485. Koneksi antar perangkat dirancang dengan mempertimbangkan kestabilan komunikasi serta efisiensi penggunaan pin pada masing-masing mikrokontroler.

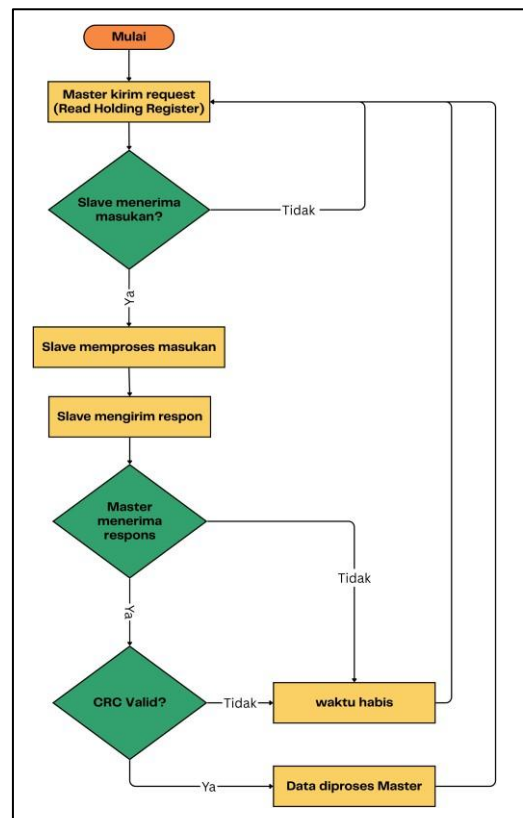


Gambar 4. Implementasi perangkat keras sistem monitoring intensitas cahaya berbasis IoT menggunakan ESP32, Arduino Nano, dan komunikasi RS485

Gambar 4 menunjukkan implementasi fisik dari sistem yang telah dirancang. Seluruh komponen perangkat keras, termasuk Arduino Nano, ESP32, modul MAX485, serta sensor BH1750, dirakit dalam satu sistem yang terintegrasi. Koneksi antar modul menggunakan kabel *jumper* dengan konfigurasi sesuai dengan skematik rangkaian yang telah dirancang sebelumnya. Prototipe ini digunakan untuk melakukan pengujian sistem secara langsung guna memastikan bahwa proses akuisisi data, komunikasi RS485, serta pengiriman data ke platform IoT dapat berjalan dengan baik.

C. Komunikasi RS485 Berbasis Modbus RTU

Komunikasi RS485 merupakan standar komunikasi serial diferensial yang banyak digunakan pada sistem industri karena memiliki ketahanan tinggi terhadap gangguan noise dan mampu mentransmisikan data dalam jarak yang relatif jauh. Sistem ini menggunakan dua jalur sinyal, yaitu A dan B, yang bekerja secara diferensial sehingga meningkatkan keandalan transmisi dibandingkan komunikasi serial konvensional. Dalam penelitian ini, RS485 diimplementasikan menggunakan protokol Modbus RTU, yaitu protokol komunikasi berbasis *master-slave* yang memungkinkan pertukaran data secara terstruktur. Modbus RTU menggunakan frame data yang terdiri dari alamat perangkat, kode fungsi, data, serta mekanisme pengecekan error menggunakan cyclic redundancy check (CRC), sehingga komunikasi antar perangkat menjadi lebih stabil dan minim kesalahan.



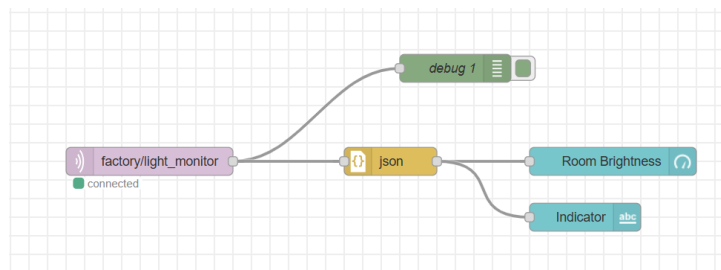
Gambar 5. Diagram alir komunikasi Modbus RTU antara *master* dan *slave*

Implementasi komunikasi dilakukan dengan menggunakan ESP32 sebagai *master* dan Arduino Nano sebagai *slave* yang terhubung melalui modul MAX485 sebagai transceiver. Arduino Nano bertugas membaca data dari sensor BH1750 dan menyimpannya dalam register Modbus yang dapat diakses oleh *master*. ESP32 kemudian melakukan proses polling secara periodik untuk meminta data dari *slave* melalui jaringan RS485. Setelah menerima permintaan, *slave* akan mengirimkan respons berupa data sensor dalam format frame Modbus RTU. Data yang diterima oleh ESP32 selanjutnya diproses untuk ditampilkan pada LCD maupun dikirim ke platform IoT untuk keperluan *monitoring*. Pendekatan ini

dipilih karena mampu memberikan komunikasi yang stabil, efisien, serta mudah dikembangkan untuk sistem dengan jumlah node yang lebih banyak.

D. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini mencakup pengembangan sistem komunikasi antara node *slave* dan *master* serta integrasi dengan platform Internet of Things untuk keperluan *monitoring*. Arduino Nano diprogram sebagai *slave* menggunakan protokol Modbus RTU untuk membaca data dari sensor BH1750 dan menyimpannya dalam register yang dapat diakses oleh *master*. ESP32 berperan sebagai *master* yang secara periodik melakukan proses polling untuk mengambil data dari *slave* melalui komunikasi RS485 menggunakan modul MAX485. Data yang diterima kemudian diproses dan ditampilkan pada perangkat output lokal sebagai bentuk *monitoring* langsung. ESP32 juga berfungsi sebagai gateway IoT dengan mengirimkan data sensor ke broker MQTT melalui koneksi WiFi. Data yang telah dipublish kemudian diterima dan diolah menggunakan platform Node-RED untuk divisualisasikan dalam bentuk *dashboard monitoring* secara real-time. Perancangan perangkat lunak ini memungkinkan integrasi antara sistem komunikasi industri berbasis RS485 dengan teknologi IoT, sehingga data dapat diakses secara fleksibel dan efisien baik secara lokal maupun jarak jauh



Gambar 6. Node program yang digunakan pada Node-RED

E. Implementasi FreeRTOS

Implementasi FreeRTOS pada penelitian ini dilakukan untuk mendukung mekanisme multitasking dalam pengolahan sistem secara real-time, khususnya dalam menangani komunikasi RS485, pemrosesan data, serta pengiriman data ke platform IoT secara bersamaan. Sistem dirancang dengan membagi proses ke dalam beberapa task yang berjalan secara paralel, meliputi task komunikasi Modbus RTU untuk melakukan polling data dari Arduino Nano, task pemrosesan data untuk mengolah hasil pembacaan sensor, serta tugas (*task*) komunikasi jaringan untuk mengirimkan data ke broker MQTT yang selanjutnya divisualisasikan melalui Node-RED. Dengan pendekatan ini, sistem mampu mengurangi blocking process dan meningkatkan efisiensi serta responsivitas dalam pengolahan data, sehingga kinerja sistem menjadi lebih stabil dan optimal untuk aplikasi *monitoring* berbasis IoT.

F. Alur Kerja Sistem

Alur kerja sistem dimulai dari proses pembacaan data intensitas cahaya oleh sensor BH1750 yang terhubung dengan Arduino Nano sebagai node *slave*. Data sensor yang diperoleh kemudian diproses dan disimpan dalam register Modbus untuk selanjutnya dikirimkan melalui komunikasi RS485 menggunakan modul MAX485. ESP32 sebagai node *master* melakukan proses polling secara berkala untuk mengambil data dari *slave*, kemudian memproses data tersebut untuk ditampilkan pada perangkat output lokal. Selain itu, ESP32 juga mengirimkan data ke broker MQTT melalui jaringan WiFi, sehingga data dapat diterima dan divisualisasikan secara real-time menggunakan platform Node-RED. Alur kerja ini menunjukkan integrasi antara sistem komunikasi industri dan teknologi IoT dalam mendukung proses monitoring yang efisien dan terstruktur.

G. Metode Pengujian

Metode pengujian pada penelitian ini dilakukan untuk mengevaluasi kinerja sistem dari aspek akurasi sensor, waktu respons, serta keandalan komunikasi data. Pengujian akurasi sensor dilakukan dengan membandingkan nilai intensitas cahaya yang diukur oleh sensor BH1750 dengan nilai referensi yang diperoleh dari aplikasi lux meter pada smartphone, sehingga pengujian bersifat komparatif. Selain itu, pengujian delay sistem dilakukan dengan mengukur selisih waktu antara data dikirim dari node *slave*

hingga diterima dan diproses oleh ESP32 sebagai master. Sementara itu, keandalan komunikasi RS485 diuji dengan menghitung jumlah data yang berhasil dikirim dan diterima dibandingkan dengan total data yang dikirim selama periode pengujian. Data hasil pengujian kemudian disajikan dalam bentuk tabel untuk memudahkan proses analisis lebih lanjut.

H. Metode Analisis Data

Analisis data pada penelitian ini dilakukan menggunakan beberapa parameter evaluasi yang umum digunakan dalam sistem monitoring dan komunikasi data[13].

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (1)$$

Tingkat akurasi sensor dianalisis menggunakan metode Mean Absolute Error (MAE) dan Mean Absolute Percentage Error (MAPE) dengan membandingkan nilai sensor terhadap nilai referensi[14].

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{y_i} \right| \quad (2)$$

Kinerja sistem dari sisi waktu respons dianalisis menggunakan parameter rata-rata delay (average delay) yang dihitung dari selisih waktu pengiriman dan penerimaan data[15].

$$Avg Delay = \frac{1}{n} \sum_{i=1}^n d_i \quad (3)$$

Selain itu, keandalan komunikasi RS485 dievaluasi menggunakan parameter Packet Delivery Ratio (PDR), yaitu perbandingan antara jumlah data yang berhasil diterima dengan jumlah data yang dikirim. Penggunaan parameter-parameter ini memungkinkan evaluasi sistem dilakukan secara kuantitatif dan terukur[16].

$$PDR = \frac{Packet\ diterima}{Packet\ dikirim} \times 100\% \quad (3)$$

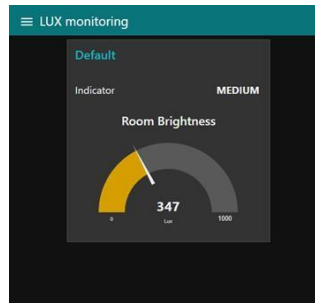
III. HASIL DAN DISKUSI

A. Implementasi Sistem

Implementasi sistem dilakukan berdasarkan perancangan yang telah dijelaskan pada tahap metode, dengan merealisasikan seluruh komponen menjadi suatu sistem monitoring berbasis *Internet of Things* (IoT) yang terintegrasi. Sistem ini terdiri dari perangkat keras berupa mikrokontroler sebagai pusat kendali, sensor untuk pengambilan data, serta modul komunikasi yang mendukung pertukaran data antar perangkat. Seluruh komponen dirangkai sesuai dengan arsitektur sistem yang telah dirancang sehingga mampu bekerja secara terkoordinasi dalam melakukan proses akuisisi dan transmisi data.

Pada tahap implementasi perangkat keras, seluruh komponen diintegrasikan dalam satu sistem yang mampu melakukan pembacaan data sensor secara langsung dari lingkungan. Data yang diperoleh kemudian diproses oleh mikrokontroler sebelum dikirimkan melalui jalur komunikasi yang telah ditentukan. Implementasi ini memastikan bahwa setiap komponen dapat berfungsi sesuai dengan perannya masing-masing, baik dalam proses pengambilan data maupun dalam proses komunikasi antar perangkat.

Implementasi perangkat lunak dilakukan dengan mengembangkan program yang berjalan pada mikrokontroler untuk mengatur alur kerja sistem secara keseluruhan. Program ini mencakup proses pembacaan data sensor, pengolahan data, serta pengiriman data ke sistem monitoring. Selain itu, sistem juga dilengkapi dengan mekanisme komunikasi yang memungkinkan pertukaran data secara terstruktur dan berkelanjutan, sehingga data yang dihasilkan dapat diakses secara real-time.



Gambar 7. Tampilan hasil monitoring pada sistem

Secara keseluruhan, hasil implementasi menunjukkan bahwa sistem yang dikembangkan telah mampu bekerja sesuai dengan perancangan awal, dimana proses akuisisi data, komunikasi, serta visualisasi data dapat berjalan dengan baik dalam satu kesatuan sistem yang terintegrasi. Hal ini menunjukkan bahwa pendekatan yang digunakan dalam pengembangan sistem monitoring berbasis IoT dapat diimplementasikan secara efektif untuk mendukung kebutuhan pemantauan secara real-time.

B. Hasil Pengujian

Hasil pengujian sistem diperoleh berdasarkan skenario yang telah dirancang sebelumnya, dengan fokus pada pengamatan terhadap data sensor, kestabilan komunikasi, serta keberhasilan sistem dalam menampilkan data secara real-time. Proses pengambilan data dilakukan dalam interval waktu satu menit untuk melihat konsistensi pembacaan sensor serta performa sistem dalam kondisi operasional.

Tabel 1 Hasil Pembacaan Sensor

<i>No</i>	<i>Waktu</i>	<i>BH1750 (Lux)</i>	<i>HP Lux Meter (Lux)</i>	<i>Error (Lux)</i>	<i>Error (%)</i>
1	menit ke-1	58	60	2	3,33
2	menit ke-2	135	130	5	3,85
3	menit ke-3	310	320	10	3,13
4	menit ke-4	545	530	15	2,83
5	menit ke-5	820	850	30	3,53
6	menit ke-6	920	900	20	2,22
7	menit ke-7	970	1000	30	3,00
8	menit ke-8	1230	1200	30	2,50
9	menit ke-9	1350	1400	50	3,57
10	menit ke-10	1650	1600	50	3,13
11	menit ke-11	1750	1800	50	2,78
12	menit ke-12	1950	1900	50	2,63
13	menit ke-13	1920	2000	80	4,00
14	menit ke-14	2100	2200	100	4,55
15	menit ke-15	2350	2400	50	2,08

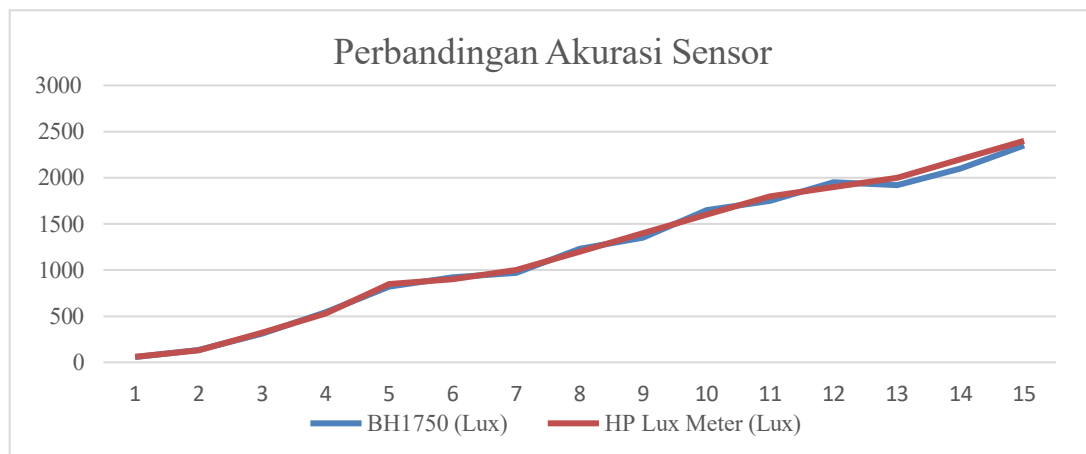
Tabel 2 Hasil delay system

<i>No</i>	<i>Pengujian data ke-</i>	<i>Delay</i>	<i>Status</i>
1	Data ke 1	118	OK
2	Data ke 2	120	OK
3	Data ke 3	121	OK

No	Pengujian data ke-	Delay	Status
4	Data ke 4	119	OK
5	Data ke 5	122	OK
6	Data ke 6	120	OK
7	Data ke 7	119	OK
8	Data ke 8	121	OK
9	Data ke 9	120	OK
10	Data ke 10	118	OK
11	Data ke 11	122	OK
12	Data ke 12	121	OK
13	Data ke 13	119	OK
14	Data ke 14	120	OK
15	Data ke 15	121	OK

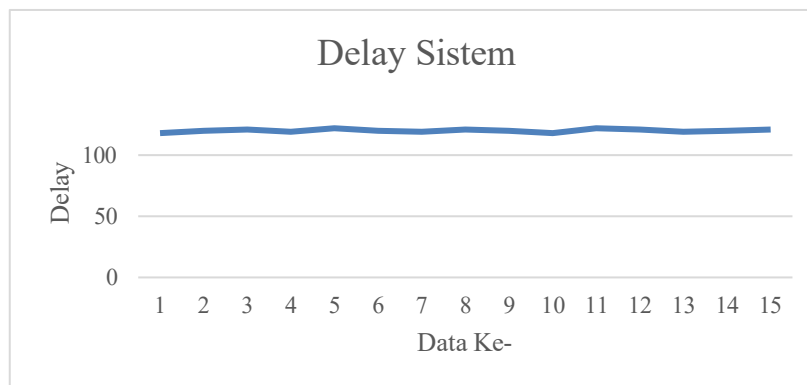
C. Analisis dan Diskusi

Berdasarkan hasil pengujian, sistem monitoring mampu melakukan pembacaan intensitas cahaya menggunakan sensor BH1750 dengan hasil yang relatif mendekati nilai referensi yang diperoleh dari aplikasi lux meter pada perangkat smartphone. Pengambilan data dilakukan setiap satu menit untuk mengamati konsistensi pembacaan sensor dalam berbagai kondisi pencahayaan. Secara umum, nilai yang dihasilkan oleh sensor menunjukkan pola yang sebanding dengan alat referensi, di mana peningkatan intensitas cahaya pada lingkungan diikuti oleh kenaikan nilai pembacaan sensor. Hal ini menunjukkan bahwa sensor mampu merespons perubahan kondisi cahaya dengan baik.



Gambar 8. Perbandingan hasil pengukuran intensitas cahaya antara sensor BH1750 dan HP *Lux meter*

Selisih antara nilai sensor dan nilai referensi ditunjukkan dalam bentuk error absolut dan error persentase. Berdasarkan data yang diperoleh, nilai error berada pada rentang yang relatif kecil, yaitu sekitar 2% hingga 4,55%. Nilai error terbesar terjadi pada pengujian menit ke-14 dengan selisih 100 lux atau sekitar 4,55%, sedangkan error terkecil terjadi pada menit ke-15 sebesar 2,08%. Secara keseluruhan, nilai error yang relatif rendah ini menunjukkan bahwa sensor BH1750 memiliki tingkat akurasi yang cukup baik untuk digunakan dalam sistem monitoring intensitas cahaya.



Gambar 9. hasil delay sistem

Berdasarkan hasil pengujian kinerja sistem dari sisi waktu respons, nilai delay sistem berada pada kisaran 118 ms hingga 122 ms untuk setiap pengiriman data. Nilai delay yang relatif stabil ini menunjukkan bahwa sistem memiliki performa komunikasi yang konsisten dalam mengirimkan data dari perangkat ESP32 ke broker MQTT. Selain itu, seluruh data yang dikirimkan memiliki status “OK”, yang menunjukkan bahwa tidak terdapat kegagalan pengiriman selama proses pengujian berlangsung.

Dengan demikian, sistem yang dikembangkan mampu bekerja dengan baik baik dari sisi akurasi sensor maupun performa komunikasi. Sensor BH1750 menunjukkan tingkat kesalahan yang rendah terhadap nilai referensi, sementara sistem komunikasi berbasis RS485 dan MQTT mampu mengirimkan data secara stabil dan real-time tanpa kehilangan paket data. Hasil ini menunjukkan bahwa sistem monitoring yang dirancang telah memenuhi kriteria dasar dalam hal keandalan, akurasi, dan responsivitas.

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, sistem monitoring intensitas cahaya berbasis Internet of Things (IoT) menggunakan ESP32, komunikasi RS485 dengan protokol Modbus RTU, serta integrasi MQTT telah berhasil dikembangkan dan mampu beroperasi dengan baik. Sistem ini dapat melakukan akuisisi data dari sensor BH1750, menampilkan informasi secara lokal melalui LCD, serta mengirimkan data secara real-time ke platform monitoring. Hasil pengujian menunjukkan bahwa sensor memiliki tingkat akurasi yang baik dengan nilai kesalahan yang relatif rendah, yang ditunjukkan melalui parameter MAE dan MAPE. Selain itu, kinerja sistem komunikasi juga menunjukkan hasil yang stabil dengan nilai delay yang rendah, sehingga mendukung kebutuhan monitoring secara responsif.

Dari sisi keandalan komunikasi, nilai Packet Delivery Ratio (PDR) yang tinggi menunjukkan bahwa seluruh data yang dikirimkan dapat diterima dengan baik tanpa kehilangan paket. Hal ini mengindikasikan bahwa integrasi komunikasi RS485 dan MQTT mampu memberikan performa sistem yang andal dan stabil. Secara keseluruhan, sistem yang dikembangkan telah memenuhi aspek utama dalam sistem IoT, yaitu akurasi, responsivitas, dan keandalan komunikasi. Dengan demikian, sistem ini memiliki potensi untuk dikembangkan lebih lanjut dan diimplementasikan pada skala yang lebih luas, khususnya dalam aplikasi monitoring industri dengan kebutuhan sistem yang efisien dan terintegrasi.

DAFTAR PUSTAKA

- [1] C. M. Nkinyam, C. Oliver, C. O. Asadu, B. Anyaka, and P. A. Olubambi, “Results in Engineering Development of a low-cost monitoring device for solar electric (PV) system using internet of things (IoT),” *Results Eng.*, vol. 28, no. September, p. 107324, 2025, doi: 10.1016/j.rineng.2025.107324.
- [2] H. Andrianto and V. Jonathan, “Design of IoT-Based Electrical Energy Meter,” *2023 IEEE 9th Int. Conf. Comput. Eng. Des.*, pp. 1–4, 2023, doi: 10.1109/ICCED60214.2023.10425406.
- [3] A. Baratta *et al.*, “ScienceDirect Design and Implementation of Modular IoT Solutions for Environmental Monitoring and Smart Automation : Three Case Studies in the Dominican Republic , Panama , and Italy,” *Procedia Comput. Sci.*, vol. 274, pp. 1140–1148, 2025, doi:

- 10.1016/j.procs.2025.12.111.
- [4] F. Moghaddam, "PLC-Based Remote Monitoring and Automation for Energy Applications," *2025 7th Exp. Int. Conf.*, pp. 378–383, 2025, doi: 10.1109/exp.at2565440.2025.11348353.
- [5] A. Abu and A. A. A. Shabaneh, "MethodsX Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things," *MethodsX*, vol. 11, no. September, p. 102401, 2023, doi: 10.1016/j.mex.2023.102401.
- [6] M. Dafare, "LoRa-Enabled Smart RS485 Data Logger and MQTT Gateway for Industrial IoT Applications using ESP32," *2023 Int. Conf. Circuit Power Comput. Technol.*, pp. 1297–1302, 2023, doi: 10.1109/ICCPCT58313.2023.10245760.
- [7] Z. Naufal, "Machine Learning Models Performance for NPK Soil Sensor Calibration," *2025 Int. Conf. ICT Smart Soc.*, no. 2020, pp. 1–6, 2025, doi: 10.1109/ICISS66954.2025.11389538.
- [8] C. Revelivan, H. Wiratama, and N. Hanafiah, "ScienceDirect ScienceDirect Door Security System for Home Monitoring Based on ESP32," *Procedia Comput. Sci.*, vol. 157, pp. 673–682, 2019, doi: 10.1016/j.procs.2019.08.218.
- [9] S. P. S. J, C. M. Iniyan, E. M. Barath, G. M. Thangatamilan, R. Gayathri, and M. Apoorva, "Tank Level Control Using Lora and PLC," *2025 Second Int. Conf. Intell. Technol. Sustain. Electr. Commun. Syst. (iTech SECOM)*, no. 2009, pp. 1–6, 2025, doi: 10.1109/iTechSECOM64750.2025.11307668.
- [10] N. Biswas, M. A. Mahdi, and M. Das Gupta, "Electricity Demand Forecasting for Tetouan, Morocco using Darts in Machine-Learning: A Comparative Study of Forecasting Models," *Results Eng.*, p. 110289, 2026, doi: 10.1016/j.rineng.2026.110289.
- [11] E. Bebronne and S. Limbourg, "Logistics 5.0 for Public Lighting Systems: Research Agenda," *IFAC Pap.*, vol. 59, no. 10, pp. 823–828, 2025, doi: 10.1016/j.ifacol.2025.09.140.
- [12] Z. Wu, "On-Chip ESD Protection for A / B Bus Ports of RS485 Using Asymmetrical dual-direction Silicon Controlled Rectifier," *2025 4th Int. Symp. Semicond. Electron. Technol.*, pp. 148–153, 2025, doi: 10.1109/ISSET66828.2025.11184970.
- [13] A. C. Study, "Hybrid Traditional and Deep Learning Approaches for Network Traffic Forecasting with Real-Time Anomaly Detection," no. May, 2025.
- [14] A. Hossain, Z. Harun, T. M. T. Lei, and M. Kumar, "Computers & Industrial Engineering Development , optimization , and benchmarking of zone-specific deep learning architectures for air quality prediction in end-of-life vehicle recycling zones," *Comput. Ind. Eng.*, vol. 210, no. August, p. 111507, 2025, doi: 10.1016/j.cie.2025.111507.
- [15] J. P. Lujan-leon, R. A. Alvarado-lugo, P. W. Lovon-ramos, and C. Benites-condori, "Datalogger for Low-Cost High Mountain Weather Stations : Design , Optimization and Testing of Power Consumption and Compatibility with Satellite Transmitter," *2023 IEEE XXX Int. Conf. Electron. Electr. Eng. Comput.*, pp. 1–8, doi: 10.1109/INTERCON59652.2023.10326074.
- [16] S. Thakur, "AI-Driven Energy-Efficient Routing in IoT-Based Wireless Sensor Networks : A Comprehensive Review," 2025.

Band Pass Filter Microstrip Metode Hairpin pada Frekuensi S Band

Dina Herdiana^{1*)}, Muhammad Ali Muhyiddin²⁾, Muhamad Aji Rahman³⁾

1,2)Program Studi Teknik Elektro
Universitas Nurtanio

Jalan Pajajaran, Husen Sastranegara, Kota Bandung, Jawa Barat 40174

2)Program Studi Teknik Elektro, Universitas Nurtanio

*)Korespondensi : dinaherdiana@unnur.ac.id

Abstrak

Sistem radar adalah teknologi yang menggunakan gelombang elektromagnetik (radio dan gelombang mikro) untuk mendeteksi, menentukan jarak, dan membuat peta objek. Namun, sering kali terdapat gangguan seperti noise dan masuknya sinyal asing ke dalam sistem tersebut. Oleh karena itu, peran filter untuk menyeleksi sinyal yang diterima serta mengurangi noise. Salah satu jenis filter yang dapat diterapkan adalah band pass filter. Pada penelitian ini dilakukan perancangan filter yang akan digunakan untuk sistem radar. Filter yang dirancang adalah band pass filter menggunakan metode microstrip hairpin yang bekerja pada frekuensi S Band, dengan nilai return loss dibawah -10 dB dan nilai insertion loss diatas -10 dB. Pada perancangan ini menggunakan substrat FR-4 dengan $\xi_r = 4,5$ dan ketebalah 1 mm. Desain filter menghasilkan band pass filter microstrip hairpin berhasil bekerja dengan efektif pada rentang frekuensi 2,385 – 2,55 GHz Band pass filter ini dapat digunakan dalam sistem radio komunikasi karena desainnya yang lebih kompak dan kecil.

Kata kunci : Band Pass Filter, Hairpin, Mikrostrip, S Band, FR-4

Abstract

Radar system is a technology that uses electromagnetic waves (radio and microwave) to detect, determine distance, and create object maps. However, there are often interferences such as noise and the entry of foreign signals into the system. Therefore, the role of the filter is to select the received signal and reduce noise. One type of filter that can be applied is a band pass filter. In this study, a filter design was carried out that will be used for the radar system. The filter designed is a band pass filter using the microstrip hairpin method that works at S Band frequencies, with a return loss value below -10 dB and an insertion loss value above -10 dB. In this design, an FR-4 substrate with $\xi_r = 4,5$ and a thickness of 1 mm was used. The filter design produces a microstrip hairpin band pass filter that works effectively in the frequency range of 2,385 - 2,55 GHz. This band pass filter can be used in radio communication systems because of its more compact and smaller design.

Keywords : Band Pass Filter, Hairpin, Mikrostrip, S Band, FR-4

I. PENDAHULUAN

Radar adalah sensor elektromagnetik untuk pendeteksi dan pelokasian dari pantulan objek. [1]

Info Makalah:

Dikirim : 06-05-2026;
Revisi 1 : 06-29-2026;
Revisi 2 : -
Diterima : 06-30-2026.

Penulis Korespondensi:

Telp : +622219178495
e-mail :
dinaherdiana@unnur.ac.id

Sistem radar merupakan teknologi yang memanfaatkan gelombang elektromagnetik untuk mendeteksi, mengidentifikasi, dan menentukan posisi suatu objek melalui proses transmisi dan penerimaan sinyal pada frekuensi tertentu. Kinerja sistem radar sangat dipengaruhi oleh kualitas sinyal yang diterima, sehingga diperlukan suatu bandpass filter yang mampu melewati sinyal pada pita frekuensi yang diinginkan sekaligus meredam sinyal di luar pita tersebut untuk mengurangi interferensi dan noise. Salah satu implementasi yang banyak dikembangkan adalah *microstrip bandpass filter* dengan metode *hairpin*, karena

memiliki dimensi yang kompak, mudah difabrikasi menggunakan teknologi PCB, serta mudah diintegrasikan dengan rangkaian gelombang mikro lainnya. Berbagai penelitian sebelumnya telah mengembangkan filter hairpin dengan fokus pada miniaturisasi dimensi, peningkatan bandwidth, dan optimasi karakteristik transmisi menggunakan substrat berkinerja tinggi seperti Rogers. Namun demikian, implementasi filter hairpin pada frekuensi *S-Band* (2,4 GHz) menggunakan substrat FR-4 yang lebih ekonomis masih relatif terbatas, terutama yang dilengkapi dengan proses fabrikasi dan validasi melalui pengukuran menggunakan *Vector Network Analyzer (VNA)*. Kondisi tersebut menunjukkan masih adanya kebutuhan penelitian untuk menghasilkan rancangan filter yang memiliki performa memadai dengan biaya fabrikasi yang lebih rendah. Oleh karena itu, penelitian ini bertujuan merancang, memfabrikasi, dan menguji *bandpass filter mikrostrip metode hairpin* pada frekuensi **2,4 GHz** menggunakan substrat **FR-4**, serta mengevaluasi karakteristik *return loss*, *insertion loss*, dan *bandwidth* sebagai dasar penerapannya pada sistem radar S-Band. Sistem radar dapat memanfaatkan filter pada jalur transmisi dan penerimaan, tergantung pada frekuensi operasi dan kebutuhan bandwidth sistem. Filter juga berperan penting dalam melindungi konverter data dalam sistem pengambilan sampel langsung. [2] Filter band pass *microstrip* yang berbasis struktur hairpin memiliki beberapa kelebihan, seperti berat yang ringan, biaya fabrikasi yang rendah dan desain yang sederhana saat integrasi dengan rangkaian lain. Filter band pass ini memiliki ukuran kompak dan supresi harmonik yang telah dikembangkan dengan menggunakan teknik groove pada struktur hairpin [3].

II. METODE

A. Teori

Desain filter hairpin mengimplikasikan lipatan resonator parallel coupled $\lambda / 2$ menjadi bentuk U sehingga dimensi filter berkurang. Perhitungan bertujuan untuk menghitung panjang coupled-lines agar dapat mempengaruhi kopling antar resonator. Garis antara dua lengkungan berfungsi untuk memendekkan panjang fisik kopling sehingga ukurannya hanya lebih kecil dari $\lambda / 4$. [4] Panjang resonator hairpin seperti pada Gambar 1 dapat dicari menggunakan pendekatan.

Elemen tunggal dari resonator hairpin ini dapat dijadikan acuan awal dalam melakukan ekstraksi kualitas eksternal (Q_{eks}).

$$Q_{eks} = \frac{g_n g_{n+1}}{FBW} \quad (1)$$

$$FBW = \frac{B_w}{F_0} \quad (2)$$

$$M_{i,i+1} = \frac{FBW}{\sqrt{g_i g_{i+1}}} \quad (3)$$

$$M_{n,n+1} = \frac{f_{n+1}^2 - f_n^2}{f_{n+1}^2 + f_n^2} \quad (4)$$

Nilai Q_{eks} dapat dihitung menggunakan persamaan (1) dengan memasukkan FBW, *fractional bandwidth* pada persamaan (2), B_w adalah lebar pita, F_0 yaitu nilai frekuensi tengah, dan g_n adalah nilai elemen filter lowpass yang digunakan pada tabel standar Chebyshev ordo tiga dengan riak 0,1 dB. Dari nilai ekstraksi Q_{eks} diperoleh nilai antara *feed line* ke resonator pertama [5].

B. Desain dan Fabrikasi

Spesifikasi filter menentukan parameter ukuran band pass filter. Sebagian besar parameter filter bandpass merupakan nama yang diberikan untuk nama dimensi Filter, seperti yang ditunjukkan pada

Tabel 1. Substrat yg digunakan Fr-4 dengan frekuensi kerja 2,4 GHz. Ukuran parameter resonator merupakan bagian yang paling berpengaruh dalam spesifikasi filter.

Tabel 1 Parameter Band Pass Filter

No	Parameter	Nilai	Satuan
1	Middle Frequency	2,4	GHz
2	Length Port Inductive	5	mm
3	Trace Width	2,2	mm
4	Trace Height	0,035	mm
5	Substrate Height	1	mm
6	Konstanta	8	
7	Resonator Width	10	mm
8	Resonator Height	19	mm
9	GAP	2,3	mm
10	Dielectric Substrate (ξ_r)	4,5	

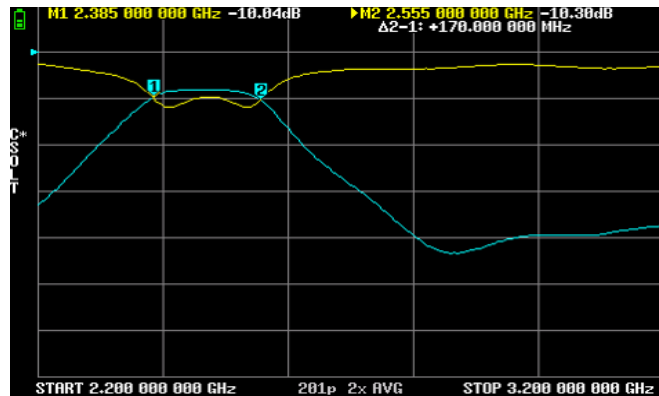
III. HASIL DAN DISKUSI

Filter didesain sesuai dengan spesifikasi dan ukuran yang diinginkan, fabrikasi dilakukan yang hasilnya dapat dilihat pada Gambar 1.



Gambar 1 Fabrikasi Filter

Bandpass filter hairpin yang telah difabrikasi diukur besar insertion loss dan return loss. Pada gambar 3 menunjukan nilai pengukurannya sebesar -10,04 dB pada frekuensi 2,385 GHz dan -10,30 dB pada frekuensi 2,555 GHz, oleh karena itu filter ini bekerja dengan baik pada daerah frekuensi tersebut. Bandwidth filter hairpin ini memiliki bandwidth sebesar 170 MHz.



Gambar 1 Insertion Loss dan Return Loss

IV. KESIMPULAN

Band pass filter yang dirancang menggunakan metode microstrip hairpin berhasil bekerja dengan efektif pada rentang frekuensi 2.385 – 2.55 GHz. Prinsip resonansi elektromagnetik yang diterapkan melalui resonator hairpin memungkinkan filter untuk meneruskan sinyal hanya dalam rentang frekuensi tersebut, sementara sinyal di luar rentang berhasil diredam. nirkabel, karena mampu memfilter sinyal dengan baik dan mengurangi noise serta gangguan sinyal asing.

DAFTAR PUSTAKA

- [1] Skolnik, Merrill. Radar Handbook. Third Edition. United State of America. Mc- Graw Hill.
- [2] Matthews, Peter. Essential Functions of RF Components for Radar Systems. Cazenovia. Knowles.
- [3] SRISATHIT, K.; TANGJIT, J.; SURAKAMPONTORN, W. Miniaturized microwave bandpass filter based on modified hairpin topology. In: 2010 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC). IEEE, 2010. p. 1-4
- [4] HASSAN, Hamid Ali. Design & Size Reduction analysis of Microstrip Hairpin Bandpass Filters. 2015.
- [5] Gilang Bonie Wiryawani, Kun Fayakun, Harry Ramza, Emilia Roza, Mohd. Azman Zakaria, dan Dwi Astuti Cahyasiwi. Antena-Filter Hairpin dengan Peningkatan Perolehan untuk Aplikasi 5G. Jurnal Rekayasa Elektroika Vol. 18, No. 4, Desember 2022, hal. 233-240
- [6] V. B. Pusuluri et al., "Narrowband Hairpin Bandpass Filter for 4G LTE Applications," Proc. IEEE IEMTRONICS, 2021. doi:10.1109/IEMTRONICS52119.2021.9422528.
- [7] T. K. Das and S. Chatterjee, "Spurline-Embedded Compact Hairpin-Line Bandpass Filters for Wide Harmonic Suppression," Int. J. Microwave Wireless Technol., vol. 14, no. 4, 2022.
- [8] B. Prasetya et al., "Band Pass Filter Comparison of Hairpin Line and Square Open-Loop Resonator Method for Digital TV Community," Bull. Electr. Eng. Informatics, vol. 10, no. 1, 2021.
- [9] S. A. Shandal, "Design and Implementation of a Novel 5G Hairpin Bandpass Filter with Defected Ground Structure," Int. J. Electr. Comput. Eng. Syst., vol. 16, no. 2, 2025.
- [10] A. V. Deshpande, "Design a Modified Hairpin Filter for Wider Bandwidth," Asian Research Journal of Current Science, vol. 3, no. 1, 2021.
- [11] M. A. Al-Mekhalafi et al., "A Compact Size 5G Hairpin Bandpass Filter with Multilayer Coupled Line," 2021.
- [12] S. Hossain and P. K. Saha, "Development of a Microstrip Hairpin Bandpass Filter for 5G Sub-6 GHz Applications," Proc. IEEE ICTP, 2025. doi:10.1109/ICTP68765.2025.11415078.
- [13] "Tight-Coupled Microstrip Hairpin Bandpass Filter," Journal of Engineering Research, vol. 11, no. 1, 2023.
- [14] K. S. Arjun et al., "A Simple and Optimizationless Technique for Multi-Harmonic Suppression in Hairpin-Line Bandpass Filter," Proc. IEEE MAPCON, 2025.
- [15] Y. Zhang et al., "Deep Learning-Enabled Rapid Optimization for Microwave Filter Design," Proc. Asia-Pacific Microwave Conference (APMC), IEEE, 2024.

Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara *Real-Time* Melalui Bluetooth pada Perangkat Android

Ni Ketut Hariyawati Dharmi ^{1*)}, Muhammad Ali Hambali ²⁾

^{1,2)}Program Studi Teknik Elektro
Universitas Jenderal Achmad Yani
Jalan Terusan Jend. Sudirman PO.BOX 148 Cimahi 40531

^{*)}Korespondensi: niketuthd@lecture.unjani.ac.id

Abstrak

Pengujian emisi kendaraan bermotor menjadi sangat penting untuk meminimalisir dampak negatif terhadap lingkungan dan kesehatan. Namun, alat uji emisi yang tersedia saat ini memiliki harga yang relatif mahal dan ukuran yang besar, sehingga kurang praktis untuk penggunaan luas. Penelitian ini bertujuan untuk merancang dan mengembangkan alat ukur emisi gas buang kendaraan bermotor berbasis mikrokontroler ATmega 328 yang lebih efisien, terjangkau, dan portabel. Metode yang digunakan melibatkan perancangan dan realisasi sistem menggunakan mikrokontroler ATmega 328 yang dilengkapi dengan sensor gas MQ-7 dan MQ-2 untuk mendeteksi kadar karbon monoksida (CO) dan hidrokarbon (HC). Data yang diperoleh dari sensor dikirim ke perangkat Android secara nirkabel melalui modul Bluetooth HC-05 dan ditampilkan secara *real-time*. Hasil pengujian menunjukkan bahwa alat yang dirancang mampu mendeteksi emisi CO dan HC dengan akurasi yang tinggi. Akurasi pengukuran HC mencapai 99,45% dan akurasi pengukuran CO mencapai 99,04% dibandingkan dengan alat uji emisi standar. Koefisien variasi (CV) pengukuran CO adalah 1,89%, sementara untuk HC adalah 0,58%, yang menunjukkan tingkat konsistensi yang baik. Alat ini juga berhasil mengirimkan data ke perangkat Android secara *real-time* dengan *delay* kurang dari 1 detik, memastikan performa yang stabil dan responsif.

Kata kunci: Arduino Uno, Bluetooth, Emisi Gas Buang, Mikrokontroler ATmega 328, Sensor Gas MQ-7, Sensor Gas MQ-2

Abstract

Testing vehicle emissions is crucial to minimize negative impacts on the environment and human health. However, the available emission testing devices are relatively expensive and large, making them less practical for widespread use. This study aims to design and develop a more efficient, affordable, and portable ATmega 328 microcontroller-based motor vehicle exhaust gas emission measurement device. The method used involves the design and realization of a system using an ATmega 328 microcontroller equipped with MQ-7 and MQ-2 gas sensors to detect carbon monoxide (CO) and hydrocarbons (HC) levels. Data obtained from the sensors are wirelessly transmitted to an Android device via a Bluetooth HC-05 module and displayed in *real-time*. The test results indicate that the designed device is capable of detecting CO and HC emissions with high accuracy. HC measurement accuracy reached 99.45% and CO measurement accuracy reached 99.04% compared to standard emission testing devices. The coefficient of variation (CV) for CO measurements is 1.89%, while for HC it is 0.58%, showing good consistency. The device also successfully transmitted data to an Android device in *real-time* with a delay of less than 1 second, ensuring stable and responsive performance.

Keywords: Arduino Uno, Bluetooth, Exhaust Gas Emissions, Microcontroller ATmega 328, MQ-7 Gas Sensor, MQ-2 Gas Sensor

I. PENDAHULUAN

Info Makalah:

Dikirim : 06-17-2026;

Revisi 1 : 06-29-2026;

Revisi 2 : -

Diterima : 06-30-2026.

Penulis Korespondensi:

Telp : +6285974992595

e-mail :

niketuthd@lecture.unjani.ac.id

Emisi adalah zat, energi, atau komponen lain yang dihasilkan dari suatu aktivitas yang dilepaskan ke udara dan dapat berfungsi sebagai polutan utama. Untuk memantau emisi dari kendaraan bermotor, alat uji emisi digunakan. Namun, alat uji emisi yang tersedia di pasaran sering kali memiliki dimensi yang besar dan sulit digunakan di ruang terbatas. Oleh karena itu, diperlukan alat ukur kadar karbon monoksida (CO) dan hidrokarbon (HC) dalam gas buang kendaraan bermotor yang lebih ringkas dan mampu memberikan hasil pengukuran secara *real-time*.

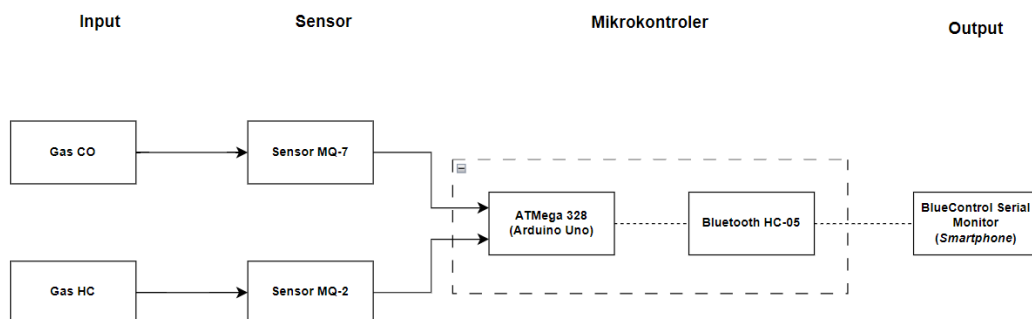
Di Indonesia, polusi udara yang disebabkan oleh kendaraan bermotor, pembangkit listrik, industri, dan aktivitas rumah tangga menyumbang hingga 70% dari total polusi udara global. Dari jumlah tersebut, emisi karbon monoksida (CO) menyumbang 99%, hidrokarbon (HC) 89%, dan oksida nitrogen (NOx) 73%, serta sejumlah polutan lainnya seperti timbal, sulfur oksida, dan partikel debu. Pengujian emisi kendaraan bermotor umumnya menggunakan alat yang disebut *gas analyzer*, yang meskipun akurat, memiliki harga yang relatif mahal dan biaya pengujian yang cukup tinggi. Selain itu, terbatasnya jumlah bengkel yang memiliki *gas analyzer* mempersulit masyarakat dalam melakukan tes uji emisi kendaraan secara rutin.

Penelitian yang dilakukan oleh A. Sukarto et al. pada tahun 2018 membahas pengembangan sistem pemantauan emisi gas buang kendaraan berbasis mikrokontroler [1]. Penelitian tersebut memberikan wawasan penting mengenai tahapan perancangan dan implementasi sistem pemantauan emisi, mulai dari pemilihan komponen sensor hingga evaluasi kinerja sistem. Selain itu, S. Ahmad et al. (2016) membahas pengembangan alat ukur emisi kendaraan portabel yang menggunakan teknologi *Internet of Things* (IoT) [2], yang memungkinkan pemantauan dan analisis data secara jarak jauh. Lebih lanjut, X. Li et al. (2019) mengembangkan alat pemantau emisi menggunakan sensor gas berteknologi tinggi yang dapat mendeteksi beberapa jenis gas secara bersamaan dengan tingkat presisi yang lebih tinggi [3].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan membuat alat ukur emisi gas buang pada kendaraan bermotor berbasis mikrokontroler ATmega 328 yang lebih terjangkau, portabel, dan mampu memberikan hasil pengukuran secara *real-time* pada perangkat Android melalui Bluetooth. Alat ini menggunakan sensor gas MQ-7 untuk mendeteksi CO dan sensor gas MQ-2 untuk mendeteksi HC, dengan sistem komunikasi nirkabel menggunakan modul Bluetooth HC-05.

II. METODE

A. Diagram Blok Sistem



Gambar 1 Diagram Blok Sistem

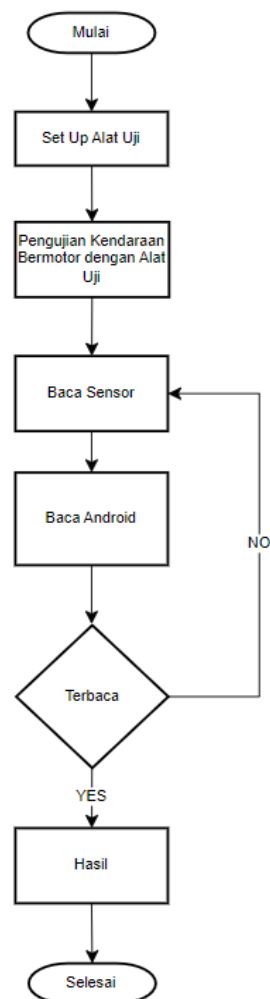
Sistem alat ukur emisi gas buang ini terdiri dari beberapa blok utama yang saling terintegrasi. Blok *input* berupa gas buang kendaraan bermotor yang mengandung senyawa CO dan HC. Blok sensor terdiri dari sensor gas MQ-7 dan sensor gas MQ-2. Sensor gas MQ-7 adalah sensor gas yang digunakan dalam peralatan untuk mendeteksi gas karbon monoksida (CO) dalam kehidupan sehari-hari, industri, atau mobil. Fitur dalam sensor gas MQ7 ini yaitu mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil, dan tahan lama tidak mudah rusak. Sensor ini menggunakan satu daya rangkaian:

Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara Real-Time Melalui Bluetooth pada Perangkat Android
(Ni Ketut Hariyawati Dharmi, Muhammad Ali Hambali : 55 - 63)

5VDC untuk *heater coil* menambahkan resistansi beban (RL), jarak pengukuran: 10 – 10.000 rpm untuk mampu mengukur gas karbon monoksida [4]. Sensor gas MQ-2 adalah sensor gas yang digunakan untuk mendeteksi konsentrasi gas hidrokarbon (HC) yang mudah terbakar seperti iso butana (C_4H_{10} / isobutane), propana (C_3H_8 / *propane*), metana (CH_4 / *methane*), etanol (*ethanol alcohol*, CH_3CH_2OH), hidrogen (H_2 / *hydrogen*), asap (*smoke*), dan LPG (*liquified petroleum gas*) [5]. Kedua sensor mengeluarkan sinyal analog yang kemudian diterima oleh blok pemrosesan sinyal, yaitu mikrokontroler ATmega 328, yang merupakan mikrokontroler 8-bit yang sangat populer, berkat penggunaannya dalam *platform* Arduino. Mikrokontroler ini dirancang oleh Atmel, yang kini dimiliki oleh Microchip Technology, dan merupakan bagian dari keluarga AVR yang terkenal dengan arsitektur RISC (*Reduced Instruction Set Computing*) [6]. Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai *output* PWM (*Pulse Width Modulation*), 6 analog *input*, *crystal oscillator* 16MHz, koneksi USB (*Universal Serial Bus*), *jack power*, kepala ICSP (*In Circuit Serial Programming*), dan tombol *reset*. Arduino mampu *men-support* mikrokontroler dapat dikoneksikan dengan USB [7].

Mikrokontroler mengolah data dari sensor dan mengirimkannya melalui modul Bluetooth HC-05 ke perangkat *smartphone* Android sebagai blok *output*, yang menampilkan nilai konsentrasi CO (dalam %) dan HC (dalam ppm) secara *real-time* melalui aplikasi Arduino Bluetooth *Control*. Modul Bluetooth merupakan komunikasi nirkabel melalui Bluetooth yang beroperasi pada frekuensi 2,4 GHz dengan *default* koneksi hanya sebagai *slave*. Bluetooth HC-05 ini sangat mudah digunakan dengan mikrokontroler untuk membuat aplikasi *wireless*. *Interface* yang digunakan adalah *serial* RXD, TXD, VCC dan GND *Built in* LED sebagai indikator koneksi [8].

B. Diagram Alir Penelitian

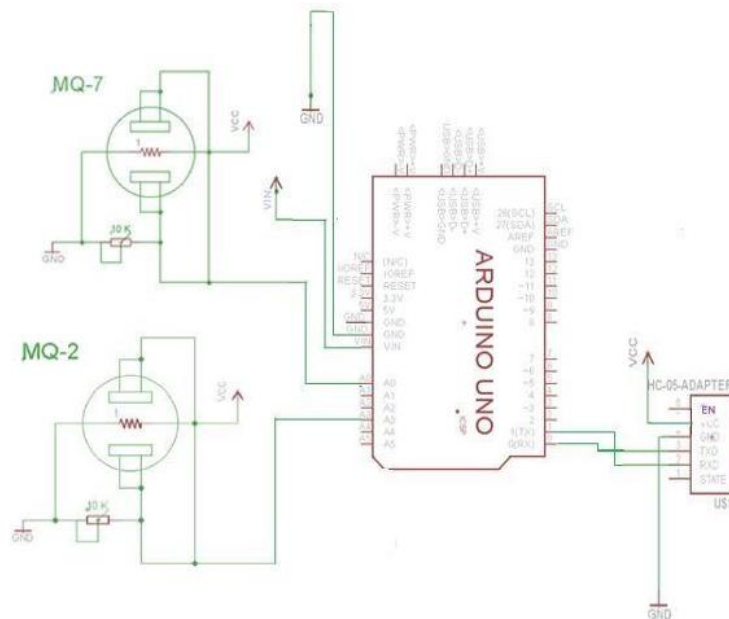


Gambar 2 Diagram Alir Penelitian

C. Perancangan Sistem

1. Perancangan Perangkat Keras

Rangkaian ini merupakan pernyataan hubungan yang berurutan dari satu atau lebih komponen yang memiliki kesatuan kerja tersendiri, dan setiap rangkaian komponen mempengaruhi komponen yang lainnya. Rangkaian keseluruhan sistem merupakan salah satu cara yang paling sederhana untuk menjelaskan cara kerja dari suatu sistem. Dengan rangkaian ini kita dapat menganalisis cara kerja rangkaian dan merancang *hardware* yang akan dibuat secara umum. Adapun rangkaian sistem yang dirancang seperti yang diperlihatkan pada gambar 3, terdapat sensor MQ7 dan MQ2 yang berfungsi untuk mendeteksi dan memperkirakan kandungan senyawa karbon monoksida (CO) dan senyawa hidrokarbon (HC) dengan cara mendeteksi kadar gas pada pipa pembuangan kendaraan bermotor. Mikrokontroler Arduino adalah komponen utama yang berfungsi sebagai pusat pengolahan data yang akan diproses sebelum dikirimkan ke penampil (*smartphone* Android) melalui Bluetooth. *smartphone* Android berfungsi sebagai penampil data yang diperoleh dari sensor agar dapat langsung dilihat hasilnya secara visual.



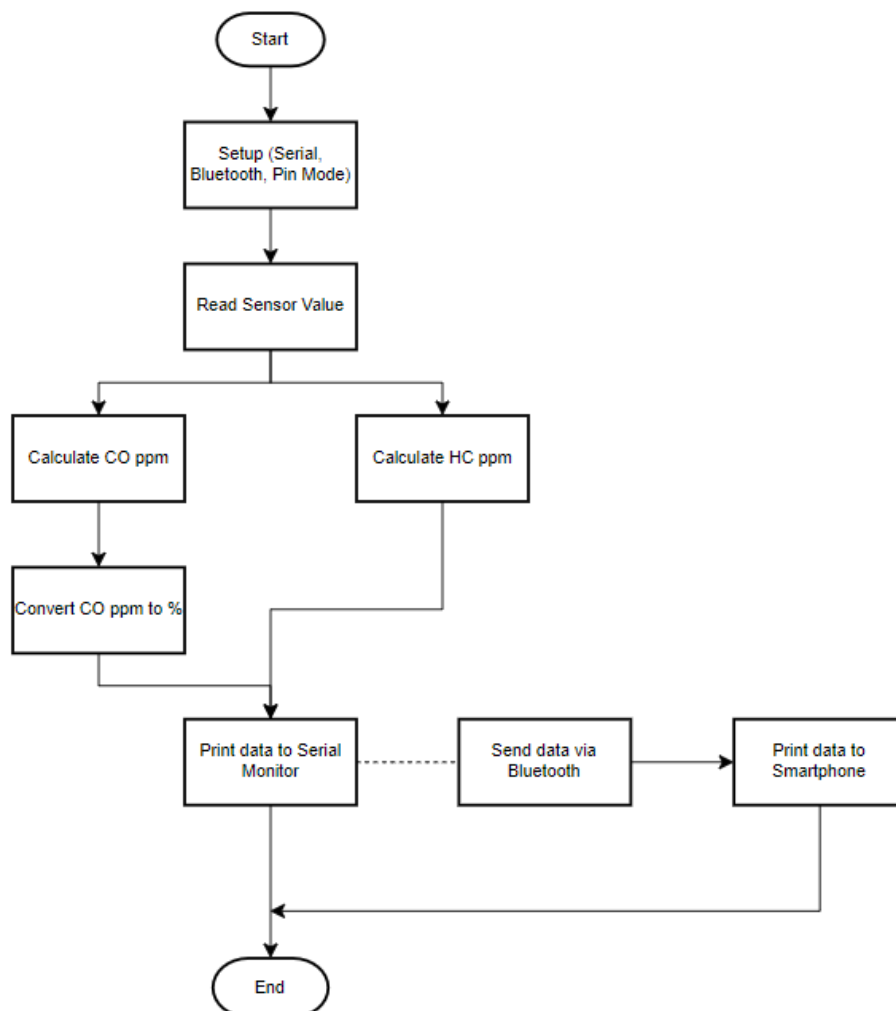
Gambar 3 Skematik Perangkat Keras Sistem

2. Perancangan Perangkat Lunak

Dalam penelitian ini, perancangan perangkat lunak mencakup pengembangan program yang dapat menghubungkan aplikasi yang dirancang dan diinstal pada *smartphone* dengan perangkat keras yang terkait. Perangkat lunak ini bertindak sebagai jembatan komunikasi antara sistem pengukuran sensor gas dan pengguna, memungkinkan pengumpulan, pemrosesan, dan penyajian data secara *real-time*.

Program yang dikembangkan meliputi beberapa aspek penting, termasuk program untuk kalibrasi sensor MQ-7 dan MQ-2. Kalibrasi ini merupakan langkah krusial untuk memastikan bahwa sensor memberikan pembacaan yang akurat sesuai dengan konsentrasi gas yang diukur. Kalibrasi dilakukan dengan mengatur nilai-nilai tertentu pada sensor, yang kemudian digunakan sebagai referensi dalam pengukuran.

Selain itu, penelitian ini juga mencakup pengembangan program keseluruhan sistem, yang mencakup fungsi-fungsi seperti pengambilan data dari sensor, pengolahan data untuk mendeteksi adanya gas berbahaya, serta komunikasi data dengan *smartphone* melalui Bluetooth. Program ini dirancang untuk memastikan sistem dapat berjalan secara efisien dan akurat, dengan antarmuka pengguna yang mudah dioperasikan. Semua aspek perangkat lunak ini dikembangkan untuk memastikan integrasi yang baik antara perangkat keras dan aplikasi pada *smartphone*, sehingga sistem dapat berfungsi secara optimal dalam mendeteksi dan menginformasikan keberadaan gas berbahaya kepada pengguna.



Gambar 4 Flowchart Program

D. Pengujian Sistem dan Pengambilan Data

Pada bagian ini dilakukan pengujian sistem yang berfungsi untuk menganalisis kesalahan data yang ada. Dilakukan pula pemeriksaan kekompakan antar komponen sistem yang diimplementasikan dan untuk mencari kesalahan serta kelemahan yang mungkin masih terjadi. Pengujian yang dilakukan pada sistem aplikasi ini meliputi pengujian terhadap seluruh menu program, proses-proses aplikasi, dan pembuatan laporan, apakah telah sesuai dengan apa yang diharapkan oleh pengguna.

Pada sistem ini, pengambilan data dilakukan terhadap gas buang kendaraan bermotor untuk menghitung emisi gas buangnya. Gas yang diukur dalam sistem ini adalah gas Hidrokarbon (HC) dan gas Karbon Monoksida (CO). Pengujian sistem (pengambilan data emisi gas buang) akan dilakukan dengan kendaraan dalam kondisi *idle*. Dalam pengujian sistem data akan ditampilkan secara *real-time* pada *serial monitor* dalam aplikasi Arduino Bluetooth Control pada *smartphone*. Data pengukuran akan ditampilkan setiap 30 detik pada *serial monitor* dan akan diambil sebanyak 10 data dengan durasi pengambilan data 5 menit. Data yang diambil merupakan hasil pengukuran oleh alat uji yang telah dirancang dan juga alat uji standar. Data pengukuran oleh alat uji rancangan kemudian dibandingkan dengan data dari alat uji standar untuk dianalisis akurasi dan konsistensi dari alat uji rancangan.

III. HASIL DAN DISKUSI

A. Hasil Perancangan Perangkat Keras

Hasil perancangan perangkat keras berupa prototipe alat ukur emisi gas buang yang terdiri dari mikrokontroler ATmega 328 (Arduino Uno), sensor MQ-7, sensor MQ-2, dan modul Bluetooth HC-05 yang terintegrasi pada sebuah PCB kustom di dalam casing yang kompak. Sensor MQ-7 dihubungkan ke pin A0 Arduino Uno dengan tegangan operasi 5V, sedangkan sensor MQ-2 dihubungkan ke pin A1 dengan tegangan input 5V DC. Modul Bluetooth HC-05 beroperasi pada frekuensi 2,4 GHz dan dihubungkan ke pin serial perangkat lunak (pin 10 dan 11 Arduino). Seluruh rangkaian diintegrasikan dalam PCB sehingga koneksi lebih stabil dan perangkat lebih portabel dibandingkan alat *gas analyzer* komersial yang berukuran besar.

B. Hasil Perancangan Perangkat Lunak

1. Kalibrasi Sensor

Kalibrasi sensor dilakukan untuk menentukan nilai R_0 , yaitu resistansi sensor pada kondisi udara bersih. Proses kalibrasi diawali dengan pemanasan sensor selama 60 detik, kemudian nilai ADC dibaca dan dikonversi ke tegangan menggunakan persamaan: $V_{out} = \text{ADC_value} \times (5,0 / 1023,0)$. Nilai R_s dihitung sebagai $R_s = (5,0 - V_{out}) / V_{out}$, dan nilai R_0 ditetapkan sama dengan R_s pada kondisi udara bersih. Hasil kalibrasi diperoleh nilai $R_{o_MQ7} = 22,6$ dan $R_{o_MQ2} = 35,2$.

2. Program Keseluruhan Sistem

Program utama sistem menggunakan pustaka `SoftwareSerial` untuk komunikasi Bluetooth dan `math.h` untuk perhitungan logaritmik. Fungsi `calculatePPM_MQ7()` menghitung konsentrasi CO berdasarkan rasio R_s/R_0 menggunakan persamaan kalibrasi kurva sensor MQ-7: $\text{PPM} = 10^{((\log_{10}(\text{ratio}) - 0,08) / -0,34)}$. Fungsi `calculatePPM_MQ2()` menghitung konsentrasi HC berdasarkan persamaan kurva sensor MQ-2: $\text{PPM} = 10^{((\log_{10}(\text{ratio}) - 0,45) / -0,38)}$. Nilai CO dalam PPM kemudian dikonversi ke persen dengan membagi 10.000. Data CO (%) dan HC (ppm) dikirimkan melalui Bluetooth setiap 30 detik dan ditampilkan secara *real-time* pada aplikasi Arduino Bluetooth Control di *smartphone* Android.

C. Hasil Pengujian Sistem

Pengujian dilakukan pada sepeda motor Honda Vario keluaran tahun 2018 untuk mengetahui fungsionalitas dari sistem yang telah dibuat. *Output* dari sistem ini akan menampilkan nilai konsentrasi karbon monoksida (CO) dalam bentuk persen (%) dan nilai konsentrasi hidrokarbon (HC) dalam bentuk *parts per million* (ppm). Data diambil sebanyak 10 kali dengan interval 30 detik. Hasil perbandingan pengukuran alat rancangan dengan alat standar ditampilkan pada Tabel 1.

Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara Real-Time Melalui Bluetooth pada Perangkat Android
(Ni Ketut Hariyawati Dharmi, Muhammad Ali Hambali : 55 - 63)

Tabel 1 Data Hasil Pengujian Emisi Gas Buang

Data ke-	Alat Rancangan		Alat Standar	
	CO (%)	HC (ppm)	CO (%)	HC (ppm)
1	0,715	985,17	0,72	986
2	0,723	982,46	0,72	985
3	0,718	979,98	0,73	986
4	0,742	987,73	0,72	987
5	0,727	981,89	0,74	984
6	0,694	967,18	0,73	985
7	0,715	976,53	0,75	985
8	0,738	979,37	0,72	986
9	0,732	982,57	0,73	986
10	0,718	976,14	0,73	983

D. Analisis Hasil Pengujian

1. Akurasi Pengukuran

Dari data hasil pengujian pada Tabel 1, dihitung nilai rata-rata pengukuran masing-masing parameter. Rata-rata CO alat rancangan diperoleh $\bar{x} = 0,722\%$, sedangkan rata-rata CO alat standar $\bar{x} = 0,729\%$. Rata-rata HC alat rancangan $\bar{x} = 979,902$ ppm, sedangkan rata-rata HC alat standar $\bar{x} = 985,3$ ppm. Berdasarkan nilai rata-rata tersebut, akurasi pengukuran CO dihitung sebagai berikut:

$$\text{Akurasi CO} = [1 - |0,722 - 0,729| / 0,729] \times 100\% = 99,04\% \quad (1)$$

$$\text{Akurasi HC} = [1 - |979,902 - 985,3| / 985,3] \times 100\% = 99,45\% \quad (2)$$

2. Koefisien Variasi (Presisi)

Standar deviasi pengukuran CO diperoleh $\sigma = 0,0137$ dan standar deviasi pengukuran HC diperoleh $\sigma = 5,7184$. Koefisien Variasi untuk masing-masing parameter dihitung sebagai berikut:

$$\text{CV CO} = (0,0137 / 0,722) \times 100\% = 1,89\% \quad (3)$$

$$\text{CV HC} = (5,7184 / 979,902) \times 100\% = 0,58\% \quad (4)$$

E. Pembahasan

Hasil pengujian menunjukkan bahwa alat ukur emisi gas buang berbasis mikrokontroler ATmega 328 yang dirancang memiliki performa yang sangat baik. Pada pengukuran CO, alat rancangan mencatat rata-rata sebesar 0,722%, yang sedikit lebih rendah dibandingkan alat standar yang mencatat nilai 0,729%. Akurasi pengukuran CO mencapai 99,04% dengan CV sebesar 1,89%, yang termasuk kategori presisi tinggi ($CV < 10\%$).

Pengukuran HC menunjukkan hasil yang lebih baik lagi. Alat rancangan mencatat rata-rata 979,902 ppm dibandingkan alat standar sebesar 985,3 ppm, dengan akurasi mencapai 99,45% dan CV hanya 0,58%. Nilai CV yang sangat rendah ini menandakan konsistensi pengukuran yang sangat baik. Berdasarkan ambang batas emisi gas buang yang ditetapkan oleh Keputusan Menteri Lingkungan Hidup No. 5 Tahun 2006 untuk sepeda motor 4 tak keluaran ≥ 2010 , batas CO adalah 4,5% dan HC adalah

2.400 ppm. Nilai pengukuran pada kendaraan uji (0,72% CO dan 985 ppm HC) masih berada di bawah ambang batas tersebut.

Sistem transmisi data melalui modul Bluetooth HC-05 berhasil mengirimkan data emisi secara *real-time* setiap 30 detik dengan *delay* kurang dari 1 detik. Performa transmisi ini memenuhi kebutuhan pemantauan emisi di lapangan. Dibandingkan dengan penelitian A. Sukarto et al. [1] yang menggunakan koneksi kabel, sistem nirkabel pada penelitian ini memberikan fleksibilitas yang lebih besar. Integrasi dengan teknologi IoT yang diusulkan S. Ahmad et al. [2] dapat menjadi arah pengembangan berikutnya untuk memperluas jangkauan pemantauan.

Tabel 2. Ringkasan Hasil Analisis Kinerja Alat

Parameter	Rata-rata Alat Rancangan	Rata-rata Alat Standar	Akurasi (%)	CV (%)
CO (%)	0,722	0,729	99,04	1,89
HC (ppm)	979,902	985,3	99,45	0,58

IV. KESIMPULAN

Penelitian ini berhasil merancang dan merealisasikan alat ukur emisi gas buang kendaraan bermotor berbasis mikrokontroler ATmega 328 yang portabel, terjangkau, dan mampu beroperasi secara real-time. Berdasarkan hasil pengujian dan analisis, dapat disimpulkan:

Alat yang dirancang memiliki akurasi pengukuran karbon monoksida (CO) sebesar 99,04% dan hidrokarbon (HC) sebesar 99,45% dibandingkan dengan alat uji emisi standar, menunjukkan kinerja yang sangat baik.

Koefisien variasi (CV) pengukuran CO adalah 1,89% dan HC adalah 0,58%, yang keduanya termasuk dalam kategori presisi tinggi (CV < 10%), menandakan konsistensi pengukuran yang sangat baik.

Sistem transmisi data Bluetooth HC-05 mampu mengirimkan data pengukuran secara *real-time* setiap 30 detik dengan *delay* kurang dari 1 detik, menjadikan alat ini sangat responsif dan cocok untuk digunakan di lapangan.

Untuk pengembangan lebih lanjut, disarankan dilakukan pengujian pada berbagai jenis kendaraan dengan bahan bakar dan karakteristik mesin yang berbeda, integrasi dengan teknologi IoT untuk pemantauan terpusat, serta penambahan fitur penyimpanan data historis dan analisis tren.

DAFTAR PUSTAKA

- [1] A. Sukarto et al., "Pengembangan Sistem Pemantauan Emisi Gas Buang Kendaraan Berbasis Mikrokontroler," Jurnal Teknologi Terapan, vol. 10, no. 2, pp. 45-52, 2018.
- [2] S. Ahmad et al., "Development of a Portable Vehicle Emission Measurement System Using IoT Technology," International Journal of Environmental Science and Technology, vol. 13, no. 7, pp. 1891-1900, 2016.
- [3] X. Li et al., "Advanced Gas Sensor Technology for Real-time Monitoring of Vehicle Emissions," Sensors and Actuators B: Chemical, vol. 283, pp. 240-249, 2019.
- [4] H. M. Lwin and T. T. Win, "Internet of Things (IoT) Based Carbon Monoxide (CO) Monitoring System Using Arduino and MQ-7 Sensor," International Journal of Advanced Computer Science and Applications, vol. 10, no. 9, pp. 118-124, 2019.
- [5] F. Amin and S. Hidri, "Design and Implementation of a Gas Leakage Monitoring System Based on Arduino and MQ-2 Gas Sensor," Procedia Computer Science, vol. 141, pp. 229-236, 2018.

Rancang Bangun Alat Ukur Emisi Gas Buang pada Kendaraan Bermotor Berbasis Mikrokontroler ATmega328 dengan Pemantauan Secara Real-Time Melalui Bluetooth pada Perangkat Android
(Ni Ketut Hariyawati Dharmi, Muhammad Ali Hambali : 55 - 63)

- [6] J. Smith and D. Miller, Embedded Systems Design with Microcontrollers: Principles and Practices. New York: Engineering World, 2019.
- [7] M. Jones, Microcontroller Basics and Applications. Arizona: TechPress Publishing, 2020.
- [8] M. Sabahi and A. Motahari, "Design and Implementation of a Smart Waste Management System Using Arduino and Bluetooth Module HC-05," in Proc. 2019 5th International Conference on Web Research (ICWR), pp. 1-6.