LAPORAN KERJA PRAKTEK
DEVELOPMENT OF ARDUINO PROGRAM AND COMPONENT CIRCUIT DIAGRAM DESIGN

International Course Program (ICP)

Disusun Oleh :

M. Rizky Aulia Hrp
(228160024)

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
2025

LEMBAR PENGESAHAN

LAPORAN KERJA PRAKTEK

DEVELOPMENT OF ARDUINO PROGRAM AND COMPONENT CIRCUIT DIAGRAM DESIGN

Diajukan Untuk Memenuhi Salah Satu Syarat Mata Kuliah Kerja Praktik Jenjang Studi S – 1 Program Studi Teknik Informatika

Oleh :

M. Rizky Aulia Hrp (228160024)

Medan, 14 September 2025

Mahasiswa

M. Rizky Aulia Hrp
NPM:228160024

Menyetujui
Dosen Pembimbing

Andre Hasudungan Lubis, S.Ti, M.Sc
NIDN.0126029101

Mengetahui,

Ketua Program Studi Teknik Informatika

Rizki Muliono, S.Kom, M.Kom
NIDN 0109038902

UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122
Website: www.teknik.uma.ac.id **E-mail**: univ_medanarea@uma.ac.id

BERITA ACARA DAN NILAI SEMINAR KERJA PRAKTEK

Pada hari ini 26 Juli 2025 telah diselenggarakan Seminar Kerja Praktek Program Studi Teknik Informatika untuk Tahun Akademik 2025/2026 atas :

Nama	: **M. Rizky Aulia Hrp**
NIM	: 228160024
Program Studi	: Teknik Informatika
Jenjang Pendidikan	: S1 (Sarjana)
Judul Kerja Praktek	: *Development of Arduino Program and Component Circuit Diagram Design*
Tempat Seminar	: Ruang Seminar Fakultas Teknik

Tanda Tangan Pembawa Seminar : ____________

Nilai Pembawa Seminar : 90(A)

Seminar Kerja Praktek bersangkutan disetujui/tidak disetujui dengan catatan perubahan seperti yang tercantum pada tabel berikut :

Saran:	***Andre Hasudungan Lubis S.Ti., M.Sc*** *Pembimbing Kerja Praktek*
Persetujuan Seminar:	
Saran:	***Rizki Muliono S.Kom, M.Kom*** *Ka. Prodi*
Persetujuan Seminar:	

PANITIA SEMINAR KERJA PRAKTEK:

No.	Jabatan	Nama Dosen	Tanda Tangan
1	Pembimbing Kerja Praktek	Andre Hasudungan Lubis S.Ti., M.Sc	1
2	Ka. Prodi	Rizki Muliono S.Kom, M.Kom	2

Medan, 26 Juli 2025
Ketua Prodi.

Rizki Muliono S.Kom, M.Kom

PREFACE

All praise and gratitude to Allah Almighty for His blessings and guidance, which have enabled the author to complete this project report for the International Course Program (ICP) entitled: "**DEVELOPMENT OF ARDUINO PROGRAM AND COMPONENT CIRCUIT DIAGRAM DESIGN**".

This report was prepared as part of the final assignment for the International Course Program (ICP) at Universitas Medan Area. The project integrates renewable energy concepts by utilizing sunlight, employing reflective mirrors for natural daylight illumination, and implementing an automatic solar-powered lighting system to support rural lighting solutions.

The author would like to express sincere gratitude to all parties who have contributed to the completion of this report:

1. My beloved parents and family for their endless prayers and support.
2. My academic supervisor and all the lecturers of the International Course Program (ICP) at Universitas Medan Area for their invaluable guidance, knowledge, and advice throughout the project.
3. My friends and teammates who have cooperated and provided support during the completion of this project.

The author realizes that this report is not without limitations and shortcomings. Therefore, constructive feedback and suggestions for improvement are highly appreciated. Finally, it is hoped that this report can provide both academic benefits and inspiration for future developments in renewable energy technology.

Medan, 31st July 2025

M. Rizky Aulia Hrp

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 Background

Electricity consumption continues to increase in line with economic development and technological innovation. As a result, the extraction of fossil energy sources, which are still the primary fuel for power generation, has also risen. Considering that fossil energy resources are limited, alternative renewable energy sources are now being sought, such as solar energy, wave energy, wind energy, and geothermal energy (L. Putriyana, 2021).

Solar energy, particularly in Indonesia with its tropical climate, is abundant. One limitation of solar energy is that it cannot be directly used to generate electricity. Solar energy must first be converted into electricity using devices such as solar panels, solar cell controllers, batteries, and inverters (Birhane, Modeling and Analysis of Photo-Voltaic Solar Panel under Constant Electric Load, 2019).

This solar energy can be utilized through a device called a solar panel (solar cell), which works by converting the absorbed sunlight intensity into electrical energy (Anglistia, 2019).

Solar cells have the advantage of being a practical energy source because they do not require transmission lines and can be installed modularly in every location that needs them. Unlike hydropower plants, which can only be installed in specific water flow areas, solar cells can be installed wherever sunlight is available, making them an environmentally friendly and sustainable alternative energy solution. However, current solar-powered lighting systems are not yet fully efficient, especially in optimizing illumination during both daytime and nighttime. Therefore, research is needed to design and develop an innovative lighting system by integrating reflective mirrors to maximize natural daylighting and automatic solar-powered lamps that can operate independently at night.

1.2 Objectives of the Project

a. To improve access to lighting in rural areas that are not yet covered by conventional electricity networks through the use of renewable energy solutions.
b. To design an energy-efficient solar-based lighting system with the optimization of reflective mirrors to enhance solar panel efficiency.
c. To develop an automatic solar-powered lamp that can operate independently at night, making it both energy-efficient and practical for residents.

d. To promote energy self-sufficiency in rural communities by utilizing abundant local solar energy resources while reducing dependence on fossil fuels or the national electricity grid (PLN).

1.3 Benefits of the Project

a. It can reduce operational costs after the initial installation.
b. It demonstrates the utilization of solar energy to reduce carbon emissions and dependence on fossil fuels, while also supporting environmental sustainability efforts.
c. It increases energy efficiency through the use of reflective mirrors, which optimize the sunlight absorption of solar panels, thereby increasing electrical output without enlarging the panel size.

CHAPTER II

LITERATURE REVIEW

The term solar panel is commonly used to refer to photovoltaic (PV) modules. A PV module is an assembly of photovoltaic cells mounted in a framework for installation. Photovoltaic cells utilize sunlight as an energy source and generate direct current (DC) electricity. A collection of PV modules is called a PV panel, and a system of multiple panels is referred to as an array. PV array systems supply solar-generated electricity to electrical equipment (Byregowda K. C, 2020).

Photovoltaic modules use the light energy (photons) from the sun to generate electricity through the photovoltaic effect. Most modules use wafer-based crystalline silicon cells or thin-film cells. The structural (load-bearing) part of the module can either be the top layer or the back layer. The cells must be protected from mechanical damage and moisture. Most modules are rigid; however, semi-flexible modules based on thin-film cells are also available.

The cells are electrically connected in series to achieve the desired voltage and then in parallel to increase the current. The wattage of a module is the mathematical product of its voltage and current. The manufacturer's specifications for solar panels are obtained under standard conditions, which do not necessarily reflect the actual operating conditions that the panels experience at the installation site.

Electrical connections of the modules are made in series to achieve the desired output voltage, or in parallel to provide the desired current capacity (amperage) for the solar panel or PV system (Pratik Pawar, 2018).

Electricity consumption continues to increase in line with economic development and technological innovation. As a result, the extraction of fossil energy sources, which are still the primary fuel for power generation, has also risen. Considering that fossil energy resources are limited, alternative renewable energy sources are now being sought, such as solar energy, wave energy, wind energy, and geothermal energy (L. Putriyana, 2021).

Solar energy, particularly in Indonesia with its tropical climate, is abundant. One limitation of solar energy is that it cannot be directly used to generate electricity. Solar energy must first be converted into electricity using devices such as solar panels, solar cell controllers, batteries, and inverters (Birhane, 2019).

CHAPTER III

METHODOLOGY

To realize this innovative concept, the research was conducted using a prototype design and development approach. This method was chosen because it allows the team to first build and test the hybrid lighting system concept on a controlled scale. The core of this methodology is the creation of a house model or miniature, which serves as a small-scale laboratory to test the system's effectiveness before real-world implementation.

3.1. Time and Place of Implementation

All activities, from concept design to final testing, were carried out within the campus environment of Universitas Medan Area. The project was conducted over a period of six months, from January to June 2025. The selection of this location ensured the availability of tools and a supportive environment for both the development and experimentation processes.

3.2. Tools and Materials

To realize this prototype, a set of tools and materials was required, which were categorized into two main groups.

3.2.1. Tools

These tools were used for assembling the house model and the electronic circuits, including:

a. Scissors and Cutter
b. Hot Glue Gun and Super Glue
c. Ruler and Marker
d. Soldering Iron, Screwdriver, and Pliers
e. Paint Brush

3.2.2. Materials

The materials used consist of items for constructing the physical model and the electronic components that serve as the brain of the system.

a. House Model Construction Materials:
   - Plywood, Cardboard, and Ice Cream Sticks to form the building structure.
   - Paint for finishing touches to make the model look presentable and realistic.

b. Hybrid Lighting System Components:
   - Solar Panel (Solar Cell): The main component that captures solar energy and converts it into electricity.
   - Reflective Mirrors: A simple solution to reflect sunlight into the interior as natural daytime lighting.
   - Battery and Battery Holder: Serve as the energy storage unit for nighttime use.
   - Charging Module (Solar Charge Controller): A smart device that protects the battery from overcharging and ensures power efficiency.
   - LED Lamp: Selected for its low power consumption while providing bright illumination.
   - Light Sensor (LDR): Acts as the system's "eye," automatically triggering the lamp to turn on when the surroundings become dark.
   - Switch and Wires: Connecting components that distribute electricity throughout the system

3.3 Procedure

This project was carried out through four systematic stages to ensure optimal result.

3.3.1. Literature Study and Design Stage

The first step was to build a foundation of knowledge by reviewing various sources related to solar energy and rural technology. Afterward, the ideas were translated into a concrete design, both for the house model and the hybrid lighting system. The model design aimed to create a replica of rural housing conditions, which would serve as the "test subject" for the implementation of this technology.

3.3.2. Model Implementation Stage

At this stage, the design was realized into a physical prototype. The solar-powered automatic lighting circuit was carefully assembled. In parallel, reflective mirrors were installed on the model at calculated angles to maximize light reflection. Both systems were then integrated into the model, forming a complete and unified lighting system.

Additionally, the solar panel was equipped with a solar tracker system that utilized LDR sensors to detect the direction of incoming sunlight. Signals from the LDRs controlled the servo motors, allowing the panel to adjust its orientation to follow the sun, thereby maximizing energy absorption. This system was also integrated with the automatic lighting circuit on the model.

3.3.3. Testing Stage

This was the crucial proof-of-concept phase. The prototype was tested in two scenarios:

a. Daylight Simulation: The model was placed under sunlight to measure how effectively the reflective mirrors illuminated the interior of the house.

b. Nighttime Simulation: The automatic lighting system was tested to observe whether the solar panel successfully charged the battery and whether the light sensors could autonomously activate the lamps in dark conditions.

3.4. Analysis Technique

The data obtained from the tests were analyzed descriptively. This means that the evaluation did not solely rely on numerical values but also included the interpretation of observations. For example, the analysis described the improvement of visual brightness inside the model due to the mirrors and the reliability of the automatic lighting system.

This analysis served as the basis for concluding whether the system is feasible and has the potential to be further developed to assist communities in remote rural areas.

CHAPTER IV

RESULTS AND DISCUSSION

4.1. Arduino System Design and Circuit Schematic

The Arduino-based system in this house model is designed to automatically rotate the solar panel to follow the direction of incoming sunlight. This system utilizes Light Dependent Resistor (LDR) sensors to detect light intensity and servo motors to adjust the solar panel's position according to the detected light source.

4.1.1. Components Used

The main components in this system consist of several integrated electronic devices, each with a specific function to support the performance of the Arduino-based solar tracker system. The components used are described as follows:

a. Arduino UNO

   The Arduino UNO serves as the main microcontroller of the system. It reads analog signals from the LDR sensors and processes the data into instructions to control the servo motors. Arduino UNO was selected due to its ease of programming, sufficient analog and digital pins, and extensive support from the community and libraries, which facilitates project development.

b. Four LDR (Light Dependent Resistor) Sensors

   The LDR sensors function as detectors of sunlight intensity. In this design, four LDRs are placed on four sides of the solar panel to compare light intensity from different directions. The brighter the received light, the lower the LDR's resistance, resulting in an output voltage that can be read by the Arduino's analog pins. The comparison of the four LDR values determines the movement direction of the solar panel.

c. Two Servo Motors

   The servo motors act as actuators to rotate the solar panel so that it faces the direction of incoming sunlight.

   a) The first servo controls the horizontal axis (X-axis), allowing the panel to move left or right.

   b) The second servo controls the vertical axis (Y-axis), allowing the panel to move upward or downward.

Servo motors were chosen because they can rotate at precise angles according to PWM (Pulse Width Modulation) signals from the Arduino and have sufficient torque to move the panel at the model scale.

d. Breadboard and Jumper Wires

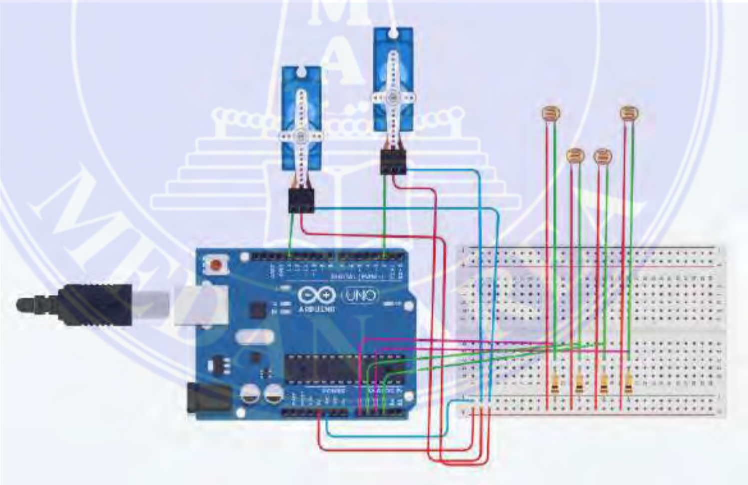
The breadboard serves as a solderless assembly medium, allowing components to be easily installed, rearranged, or replaced if needed. Jumper wires are used to connect components to the Arduino and to create a neat, modular circuit layout.

e. Resistors

Resistors are used as current limiters and to form voltage divider circuits for the LDR sensors. This ensures that the voltage levels read by the Arduino remain within the safe operating range (0–5V) and can be processed accurately.

4.1.2. Circuit Schematic

Figure below illustrates the circuit schematic of the Arduino system designed for this project. This schematic represents the connection between the LDR sensors, the servo motors, and the Arduino UNO, which serves as the core of the solar tracker system on the house model.



In this circuit, each component is connected to perform a specific function that supports the overall operation of the system. Four LDR (Light Dependent Resistor) sensors are positioned on four sides of the solar panel to detect the intensity of sunlight from different directions. Each LDR is connected to the analog pins A0, A1, A2, and A3 on the Arduino UNO through a series resistor, forming a voltage divider circuit. The resistor ensures that the voltage output from the LDR varies proportionally with the light intensity, allowing the Arduino to accurately read the differences in light received by each sensor.

Two servo motors are used as the actuators to adjust the position of the solar panel. The first servo motor is connected to the PWM digital pin 9 and is responsible for controlling the horizontal (X-axis) movement, while the second servo motor is connected to the PWM digital pin 10 to control the vertical (Y-axis) movement. Both servo motors receive control signals from the Arduino based on the comparison of the light intensity values detected by the LDR sensors.

The entire system is powered by the 5V output from the Arduino UNO, and all grounds are connected through the breadboard to form a common ground. This configuration makes the circuit tidy and allows for easier testing and maintenance.

With this setup, the system operates automatically. When any LDR sensor receives higher light intensity than the others, the Arduino processes the signal and sends a command to the servo motors to rotate the solar panel toward the brighter light source. This ensures that the solar panel consistently faces the optimal direction to maximize sunlight exposure.

4.1.3. System Working Principle

The working principle of the Arduino-based solar tracker system in this house model is to optimize solar energy absorption by continuously rotating the solar panel toward the direction of the strongest light source. The system operates in several integrated stages.

The process begins with **light intensity detection**. Four Light Dependent Resistor (LDR) sensors are positioned on four sides of the solar panel to detect sunlight from multiple directions. When sunlight hits the LDR, its resistance decreases, resulting in a voltage change that varies according to the intensity of the received light.

Next, the **Arduino reads the voltage signals** from the four LDR sensors through its analog input pins (A0, A1, A2, and A3). These analog voltage values are then converted into digital data using the internal Analog-to-Digital Converter (ADC) of the Arduino.

The system then performs **light intensity comparison** to determine which side receives the highest level of illumination. If the LDR on the left side detects stronger light, the solar panel is rotated to the left. Conversely, if the right LDR receives more light, the panel rotates to the right. The combination of readings from the upper and lower LDRs determines the vertical movement of the panel.

Based on the comparison results, the Arduino sends **Pulse Width Modulation (PWM) signals** to the two servo motors. The horizontal servo motor moves the panel along the X-axis, while the vertical servo motor adjusts the panel along the Y-axis. Through this mechanism, the solar panel automatically moves toward the direction of the brightest light source. This ensures that the panel consistently receives maximum sunlight exposure throughout the day, resulting in higher energy absorption efficiency compared to a static solar panel. By continuously following the sun's movement from morning until evening, the system effectively increases the overall performance of the solar energy capture on the model.

4.2. Arduino Programming Code

The Arduino program serves as the core logic of the solar tracking system in this house model. It is responsible for reading the light intensity detected by four LDR (Light Dependent Resistor) sensors placed on four sides of the solar panel. Based on these readings, the Arduino compares the values and controls two servo motors to rotate the panel in the direction of the highest light intensity. This allows the panel to continuously adjust its position for maximum sunlight exposure.

The following code was developed and uploaded to the Arduino UNO to achieve this automatic tracking function:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Servo.h>
#include <FS.h>
#include <LittleFS.h>

// ===== KONFIGURASI ACCESS POINT =====
const char* ap_ssid = "Solar_Tracker";
const char* ap_password = "12345678";

// ===== SERVO MOTOR =====
Servo horizontal;
Servo vertical;

// ===== POSISI SERVO DAN TRACKING =====
int servohori = 90;
int servovert = 45;
int prevHori = 90;
int prevVert = 45;

// ===== DATA GERAKAN =====
String lastMovement = "Tidak ada gerakan";
unsigned long lastMoveTime = 0;
int totalMovements = 0;
String movementHistory[10]; // Simpan 10 gerakan terakhir
int historyIndex = 0;

// ===== SIMULASI TRACKING (untuk demo) =====
bool autoTracking = true;
```

```
unsigned long lastTrackingTime = 0;
int trackingDirection = 1;
float simulationSpeed = 0.5; // Kecepatan simulasi

// ===== VARIABEL MONITORING =====
unsigned long lastLogTime = 0;
unsigned long lastUpdateTime = 0;

ESP8266WebServer server(80);

// ===== FUNGSI UNTUK MENDETEKSI GERAKAN =====
void detectMovement() {
  if (servohori != prevHori || servovert != prevVert) {
    String movement = "";

    // Deteksi gerakan horizontal
    if (servohori != prevHori) {
      int horiDiff = servohori - prevHori;
      if (horiDiff > 0) {
        movement += "Kanan " + String(horiDiff) + "°";
      } else {
        movement += "Kiri " + String(abs(horiDiff)) + "°";
      }
    }

    // Deteksi gerakan vertikal
    if (servovert != prevVert) {
      int vertDiff = servovert - prevVert;
      if (movement.length() > 0) movement += ", ";

      if (vertDiff > 0) {
        movement += "Naik " + String(vertDiff) + "°";
      } else {
        movement += "Turun " + String(abs(vertDiff)) + "°";
      }
    }

    // Simpan gerakan
    lastMovement = movement;
    lastMoveTime = millis();
    totalMovements++;

    // Tambah ke history
    movementHistory[historyIndex] = String(millis()/1000) + "s: " + movement;
    historyIndex = (historyIndex + 1) % 10;

    // Update posisi sebelumnya
    prevHori = servohori;
    prevVert = servovert;

    Serial.println("GERAKAN: " + movement);
  }
}

// ===== SIMULASI TRACKING MATAHARI =====
void simulateTracking() {
  if (!autoTracking) return;

  if (millis() - lastTrackingTime > 3000) { // Update setiap 3 detik
    lastTrackingTime = millis();

    // Simulasi gerakan mengikuti matahari
    int oldHori = servohori;
    int oldVert = servovert;

    // Gerakan horizontal (simulasi rotasi bumi)
    servohori += trackingDirection * random(1, 4);

    // Batasi gerakan
    if (servohori >= 170) {
```

```
      servohori = 170;
      trackingDirection = -1;
    } else if (servohori <= 10) {
      servohori = 10;
      trackingDirection = 1;
    }

    // Gerakan vertikal (simulasi elevasi matahari)
    if (servohori < 90) {
      // Pagi: naik
      servovert += random(0, 2);
    } else if (servohori > 90) {
      // Sore: turun
      servovert -= random(0, 2);
    } else {
      // Siang: sedikit variasi
      servovert += random(-1, 2);
    }

    // Batasi vertikal
    servovert = constrain(servovert, 1, 100);

    // Gerakkan servo fisik
    horizontal.write(servohori);
    vertical.write(servovert);

    // Deteksi gerakan
    detectMovement();
  }
}

// ===== FUNGSI ARAH =====
String getPanelDirection() {
  if (servohori < 45) return "Kiri Jauh";
  if (servohori < 75) return "Kiri";
  if (servohori < 105) return "Tengah";
  if (servohori < 135) return "Kanan";
  return "Kanan Jauh";
}

String getPanelHeight() {
  if (servovert < 25) return "Sangat Rendah";
  if (servovert < 45) return "Rendah";
  if (servovert < 65) return "Sedang";
  if (servovert < 85) return "Tinggi";
  return "Sangat Tinggi";
}

String getSunStatus() {
  if (servohori < 60) return "Matahari Pagi";
  if (servohori > 120) return "Matahari Sore";
  return "Matahari Siang";
}

// ===== SIMPAN DATA CSV =====
void saveToCSV() {
  String timestamp = String(millis() / 1000);
  String logEntry = timestamp + "," + String(servohori) + "," + String(servovert) + "\n";

  File file = LittleFS.open("/data.csv", "a");
  if (file) {
    file.print(logEntry);
    file.close();
  }
}

void initCSV() {
  if (!LittleFS.exists("/data.csv")) {
    File file = LittleFS.open("/data.csv", "w");
    if (file) {
```

```
      file.println("Waktu,Horizontal,Vertikal");
      file.close();
    }
  }
}

void setup() {
  Serial.begin(115200);

  if (!LittleFS.begin()) {
    Serial.println("File system error");
    return;
  }

  initCSV();

  // ===== SERVO =====
  horizontal.attach(D1);
  vertical.attach(D2);
  horizontal.write(servohori);
  vertical.write(servovert);

  // ===== WIFI =====
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ap_ssid, ap_password);

  // ===== WEB SERVER =====
  server.on("/", handleRoot);
  server.on("/data", handleData);
  server.on("/download", handleDownload);
  server.on("/clear", handleClear);
  server.on("/toggle", handleToggle);
  server.begin();

  Serial.println("Solar Tracker Started!");
  Serial.println("Connect to: " + String(ap_ssid));
  Serial.println("Open: http://" + WiFi.softAPIP().toString());

  // Inisialisasi history
  for (int i = 0; i < 10; i++) {
    movementHistory[i] = "";
  }

  saveToCSV();
}

void loop() {
  server.handleClient();

  // Simulasi tracking
  simulateTracking();

  // Auto save setiap 5 detik
  if (millis() - lastLogTime > 5000) {
    saveToCSV();
    lastLogTime = millis();
  }
}

void handleRoot() {
  String html = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Panel Surya Real-Time</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
        * {
            margin: 0;
```

```
        padding: 0;
        box-sizing: border-box;
    }

    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        margin: 0;
        padding: 10px;
        background: #f8f9fa;
        color: #333;
        min-height: 100vh;
    }

    .container {
        max-width: 900px;
        margin: 0 auto;
        padding: 0 10px;
    }

    /* Card Styles */
    .card {
        background: white;
        border-radius: 15px;
        padding: 25px;
        margin: 20px 0;
        box-shadow: 0 4px 20px rgba(0, 0, 0, 0.08);
        border: 1px solid #e9ecef;
        transition: all 0.3s ease;
        position: relative;
    }

    .card::before {
        content: '';
        position: absolute;
        top: 0;
        left: 0;
        right: 0;
        height: 4px;
        background: #007bff;
        border-radius: 15px 15px 0 0;
    }

    .card:hover {
        transform: translateY(-3px);
        box-shadow: 0 8px 25px rgba(0, 123, 255, 0.15);
    }

    /* Header */
    h1 {
        text-align: center;
        color: #007bff;
        font-size: 2.5em;
        font-weight: 700;
        margin-bottom: 30px;
    }

    /* Status Card */
    .status-card {
        background: linear-gradient(135deg, #007bff, #0056b3);
        color: white;
        text-align: center;
    }

    .status-card::before {
        background: #007bff;
    }

    .status-card .big-text {
        font-size: 2.2em;
        font-weight: 700;
```

```
    margin: 15px 0;
}

.status-card .medium-text {
    font-size: 1.4em;
    margin: 8px 0;
    opacity: 0.9;
}

/* Position Card */
.position-card {
    background: white;
}

.status-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 20px;
    margin: 25px 0;
}

.status-item {
    background: #f8f9fa;
    padding: 20px;
    border-radius: 12px;
    text-align: center;
    transition: all 0.3s ease;
    border: 2px solid #e9ecef;
}

.status-item:hover {
    background: #e3f2fd;
    border-color: #007bff;
    transform: scale(1.02);
}

.status-label {
    font-size: 1em;
    color: #6c757d;
    margin-bottom: 10px;
    font-weight: 500;
}

.status-value {
    font-size: 2em;
    font-weight: 700;
    color: #007bff;
}

/* Progress Bars */
.progress {
    margin: 20px 0;
}

.progress-label {
    font-size: 1.1em;
    font-weight: 600;
    margin-bottom: 10px;
    color: #495057;
}

.progress-bar {
    width: 100%;
    height: 30px;
    background: #e9ecef;
    border-radius: 15px;
    overflow: hidden;
    position: relative;
    border: 1px solid #dee2e6;
}
```

```
.progress-fill {
    height: 100%;
    background: linear-gradient(90deg, #007bff, #0056b3);
    transition: width 0.5s ease;
    border-radius: 15px;
}

.progress-text {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-weight: 700;
    color: white;
    font-size: 1em;
}

/* Movement Card */
.movement-card {
    background: #e3f2fd;
    border-left: 4px solid #007bff;
}

.movement-card::before {
    background: #007bff;
}

.movement-text {
    font-size: 1.3em;
    font-weight: 600;
    margin: 15px 0;
    color: #0056b3;
}

.movement-time {
    font-size: 1em;
    color: #6c757d;
}

.stats-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(130px, 1fr));
    gap: 15px;
    margin: 20px 0;
}

.stat-item {
    background: white;
    padding: 15px;
    border-radius: 12px;
    text-align: center;
    transition: all 0.3s ease;
    border: 2px solid #e9ecef;
}

.stat-item:hover {
    background: #f8f9fa;
    border-color: #007bff;
    transform: translateY(-2px);
}

.stat-value {
    font-size: 1.4em;
    font-weight: 700;
    color: #007bff;
}

.stat-label {
    font-size: 0.9em;
```

```
            color: #6c757d;
            margin-top: 5px;
        }

        /* History Card */
        .history-item {
            background: #f8f9fa;
            padding: 12px 15px;
            border-radius: 10px;
            margin: 8px 0;
            font-size: 0.95em;
            color: #495057;
            border-left: 4px solid #007bff;
            transition: all 0.3s ease;
        }

        .history-item:hover {
            background: #e3f2fd;
            transform: translateX(5px);
        }

        /* Buttons */
        .button {
            background: #007bff;
            color: white;
            border: none;
            padding: 12px 20px;
            border-radius: 8px;
            font-size: 1em;
            font-weight: 600;
            cursor: pointer;
            margin: 8px;
            transition: all 0.3s ease;
            box-shadow: 0 3px 10px rgba(0, 123, 255, 0.3);
        }

        .button:hover {
            background: #0056b3;
            transform: translateY(-2px);
            box-shadow: 0 5px 15px rgba(0, 123, 255, 0.4);
        }

        .button.danger {
            background: #dc3545;
            box-shadow: 0 3px 10px rgba(220, 53, 69, 0.3);
        }

        .button.danger:hover {
            background: #c82333;
            box-shadow: 0 5px 15px rgba(220, 53, 69, 0.4);
        }

        .button.success {
            background: #28a745;
            box-shadow: 0 3px 10px rgba(40, 167, 69, 0.3);
        }

        .button.success:hover {
            background: #1e7e34;
            box-shadow: 0 5px 15px rgba(40, 167, 69, 0.4);
        }

        .center {
            text-align: center;
        }

        /* Live Indicator */
        .live-indicator {
            position: fixed;
            top: 20px;
```

```
            right: 20px;
            background: #007bff;
            color: white;
            padding: 10px 15px;
            border-radius: 20px;
            font-size: 0.9em;
            font-weight: 600;
            z-index: 1000;
            box-shadow: 0 4px 15px rgba(0, 123, 255, 0.3);
        }

        .live-indicator::before {
            content: "●";
            margin-right: 8px;
            animation: blink 1s infinite;
        }

        @keyframes blink {
            0%, 50% { opacity: 1; }
            51%, 100% { opacity: 0.3; }
        }

        /* Responsive Design */
        @media (max-width: 768px) {
            .container {
                padding: 0 5px;
            }

            .card {
                padding: 20px;
                margin: 15px 0;
                border-radius: 15px;
            }

            h1 {
                font-size: 2em;
                margin-bottom: 20px;
            }

            .status-grid {
                grid-template-columns: 1fr;
                gap: 15px;
            }

            .stats-grid {
                grid-template-columns: repeat(2, 1fr);
                gap: 10px;
            }

            .status-card .big-text {
                font-size: 1.8em;
            }

            .status-card .medium-text {
                font-size: 1.2em;
            }

            .status-value {
                font-size: 1.6em;
            }

            .button {
                padding: 10px 15px;
                font-size: 0.9em;
                margin: 5px;
            }

            .live-indicator {
                top: 10px;
                right: 10px;
```

```
                padding: 8px 12px;
                font-size: 0.8em;
            }
        }

        @media (max-width: 480px) {
            body {
                padding: 5px;
            }

            .card {
                padding: 15px;
                margin: 10px 0;
            }

            h1 {
                font-size: 1.6em;
            }

            .status-card .big-text {
                font-size: 1.5em;
            }

            .status-card .medium-text {
                font-size: 1.1em;
            }

            .status-value {
                font-size: 1.4em;
            }

            .stats-grid {
                grid-template-columns: 1fr;
            }

            .button {
                width: 100%;
                margin: 5px 0;
            }
        }

        @media (min-width: 1200px) {
            .container {
                max-width: 1100px;
            }

            .stats-grid {
                grid-template-columns: repeat(3, 1fr);
            }
        }
    </style>
</head>
<body>
    <div class="live-indicator">LIVE</div>

    <div class="container">
        <h1>🌞 Panel Surya Real-Time</h1>

        <div class="card status-card">
            <div class="big-text" id="sunStatus">Matahari Siang</div>
            <div class="medium-text" id="panelDirection">Tengah</div>
            <div class="medium-text" id="panelHeight">Sedang</div>
        </div>

        <div class="card position-card">
            <div class="status-grid">
                <div class="status-item">
                    <div class="status-label">Posisi Kiri-Kanan</div>
                    <div class="status-value" id="horiValue">90°</div>
```

```
                </div>
                <div class="status-item">
                    <div class="status-label">Posisi Atas-Bawah</div>
                    <div class="status-value" id="vertValue">45°</div>
                </div>
            </div>

            <div class="progress">
                <div class="progress-label">Gerakan Kiri ↔ Kanan</div>
                <div class="progress-bar">
                    <div class="progress-fill" id="horiBar"></div>
                    <div class="progress-text" id="horiText">50%</div>
                </div>
            </div>

            <div class="progress">
                <div class="progress-label">Gerakan Atas ↕ Bawah</div>
                <div class="progress-bar">
                    <div class="progress-fill" id="vertBar"></div>
                    <div class="progress-text" id="vertText">45%</div>
                </div>
            </div>
        </div>

        <div class="card movement-card">
            <h3>🎯 Gerakan Terakhir</h3>
            <div class="movement-text" id="lastMovement">Tidak ada gerakan</div>
            <div class="movement-time" id="lastMoveTime">-</div>

            <div class="stats-grid">
                <div class="stat-item">
                    <div class="stat-value" id="totalMovements">0</div>
                    <div class="stat-label">Total Gerakan</div>
                </div>
                <div class="stat-item">
                    <div class="stat-value" id="trackingStatus">ON</div>
                    <div class="stat-label">Auto Tracking</div>
                </div>
                <div class="stat-item">
                    <div class="stat-value" id="updateTime">0s</div>
                    <div class="stat-label">Update Terakhir</div>
                </div>
            </div>
        </div>

        <div class="card">
            <h3>📜 Riwayat Gerakan</h3>
            <div id="movementHistory">
                <div class="history-item">Memuat riwayat...</div>
            </div>
        </div>

        <div class="card center">
            <h3>🎮 Kontrol & Data</h3>
            <button class="button success" onclick="toggleTracking()">🔄 Toggle Auto</button>
            <button class="button" onclick="downloadData()">📥 Download Data</button>
            <button class="button danger" onclick="clearData()">🗑 Hapus Data</button>
        </div>
    </div>

    <script>
        function updateData() {
            fetch('/data')
                .then(response => response.json())
                .then(data => {
                    // Update positions
                    document.getElementById('horiValue').textContent = data.horizontal + '°';
                    document.getElementById('vertValue').textContent = data.vertical + '°';
```

```
                    // Update progress bars
                    const horiPercent = ((data.horizontal - 5) / 170) * 100;
                    const vertPercent = ((data.vertical - 1) / 99) * 100;

                    document.getElementById('horiBar').style.width = Math.max(0, Math.min(100,
horiPercent)) + '%';
                    document.getElementById('vertBar').style.width = Math.max(0, Math.min(100,
vertPercent)) + '%';

                    document.getElementById('horiText').textContent = Math.round(horiPercent) +
'%';
                    document.getElementById('vertText').textContent = Math.round(vertPercent) +
'%';

                    // Update status
                    document.getElementById('sunStatus').textContent = data.sunStatus;
                    document.getElementById('panelDirection').textContent = data.direction;
                    document.getElementById('panelHeight').textContent = data.height;

                    // Update movement info
                    document.getElementById('lastMovement').textContent = data.lastMovement;
                    document.getElementById('lastMoveTime').textContent = data.lastMoveTime;
                    document.getElementById('totalMovements').textContent = data.totalMovements;
                    document.getElementById('trackingStatus').textContent = data.autoTracking ?
'ON' : 'OFF';
                    document.getElementById('updateTime').textContent = Math.floor(Date.now() /
1000) + 's';

                    // Update history
                    const historyDiv = document.getElementById('movementHistory');
                    historyDiv.innerHTML = '';
                    data.history.forEach(item => {
                        if (item && item.trim() !== '') {
                            const div = document.createElement('div');
                            div.className = 'history-item';
                            div.textContent = item;
                            historyDiv.appendChild(div);
                        }
                    });

                    if (historyDiv.children.length === 0) {
                        const div = document.createElement('div');
                        div.className = 'history-item';
                        div.textContent = 'Belum ada gerakan';
                        historyDiv.appendChild(div);
                    }
                })
                .catch(error => {
                    console.error('Error:', error);
                });
        }

        function toggleTracking() {
            fetch('/toggle')
                .then(response => response.text())
                .then(data => {
                    console.log(data);
                });
        }

        function downloadData() {
            window.location.href = '/download';
        }

        function clearData() {
            if (confirm('Hapus semua data?')) {
                fetch('/clear')
                    .then(response => response.text())
                    .then(data => {
                        console.log(data);
```

```
                    });
                }
            }

            // Auto update every 1 second
            updateData();
            setInterval(updateData, 1000);
        </script>
</body>
</html>
)rawliteral";

  server.send(200, "text/html", html);
}

void handleData() {
  // Buat array history untuk JSON
  String historyJson = "[";
  for (int i = 0; i < 10; i++) {
    if (movementHistory[i].length() > 0) {
      if (historyJson.length() > 1) historyJson += ",";
      historyJson += "\"" + movementHistory[i] + "\"";
    }
  }
  historyJson += "]";

  String timeAgo = "";
  if (lastMoveTime > 0) {
    int secondsAgo = (millis() - lastMoveTime) / 1000;
    timeAgo = String(secondsAgo) + " detik yang lalu";
  } else {
    timeAgo = "Belum ada gerakan";
  }

  String json = "{";
  json += "\"horizontal\":" + String(servohori) + ",";
  json += "\"vertical\":" + String(servovert) + ",";
  json += "\"direction\":\"" + getPanelDirection() + "\",";
  json += "\"height\":\"" + getPanelHeight() + "\",";
  json += "\"sunStatus\":\"" + getSunStatus() + "\",";
  json += "\"lastMovement\":\"" + lastMovement + "\",";
  json += "\"lastMoveTime\":\"" + timeAgo + "\",";
  json += "\"totalMovements\":" + String(totalMovements) + ",";
  json += "\"autoTracking\":" + String(autoTracking ? "true" : "false") + ",";
  json += "\"history\":" + historyJson;
  json += "}";

  server.send(200, "application/json", json);
}

void handleToggle() {
  autoTracking = !autoTracking;
  String response = autoTracking ? "Auto tracking ON" : "Auto tracking OFF";
  server.send(200, "text/plain", response);
}

void handleDownload() {
  File file = LittleFS.open("/data.csv", "r");
  if (!file) {
    server.send(404, "text/plain", "File tidak ada");
    return;
  }

  server.setContentLength(file.size());
  server.send(200, "text/csv", "");

  WiFiClient client = server.client();
  while (file.available()) {
    client.write(file.read());
  }
```

```
  file.close();
}

void handleClear() {
  LittleFS.remove("/data.csv");
  initCSV();

  // Reset movement data
  totalMovements = 0;
  lastMovement = "Tidak ada gerakan";
  lastMoveTime = 0;
  for (int i = 0; i < 10; i++) {
    movementHistory[i] = "";
  }
  historyIndex = 0;

  server.send(200, "text/plain", "Data dan riwayat gerakan dihapus");
}
```

This program reads analog inputs from the LDR sensors, calculates average light values from different sides, and adjusts the panel's orientation using the two servo motors. The use of tolerances prevents unnecessary movement due to small changes in light intensity, improving stability and system performance.

By integrating this Arduino program with the physical circuit, the solar tracker is able to dynamically follow the sun's movement across the sky, thereby enhancing the efficiency of solar energy capture compared to a static panel setup.

CHAPTER V

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Based on the design, development, and testing of the project Hybrid: Design and Development of a Sunlight-Based Lighting System Using Reflective Mirrors, an Automatic Solar-Powered Lamp for Rural Lighting, it can be concluded that this system presents an innovative and sustainable solution to address the issue of limited access to electricity in rural areas. The integration of reflective mirrors for daytime natural lighting and an automatic solar-powered lamp for nighttime illumination successfully provides a hybrid lighting system that is energy-efficient, environmentally friendly, and cost-effective.

The Arduino-based solar tracker system also proved to be highly effective in optimizing solar energy absorption. By utilizing four LDR sensors to detect sunlight intensity from multiple directions and two servo motors to adjust the solar panel along both horizontal and vertical axes, the panel consistently followed the direction of the brightest light throughout the day. This dual-axis tracking mechanism significantly improved the panel's exposure to sunlight compared to a static panel, resulting in a higher energy capture for powering the automatic lighting system at night.

Testing on the house model demonstrated that the system could operate autonomously. During daylight, reflective mirrors increased indoor brightness without consuming additional energy, while at night, the stored solar energy automatically powered the LED lamps as the LDR sensor detected low ambient light. These results confirm that the hybrid solar lighting system is both feasible and reliable for rural applications, providing a sustainable and independent energy solution.

5.2 Recommendation

To maximize the long-term performance and applicability of this hybrid lighting system, several improvements are recommended. Future development should focus on enhancing the efficiency of the solar tracker and increasing the capacity of energy storage, ensuring that the system can operate for extended periods, even under suboptimal sunlight conditions. The integration of smart monitoring or automation using IoT technology could provide real-time information about battery status, panel orientation, and lighting performance, further increasing reliability and user convenience.

Additionally, the use of weather-resistant materials and optimized reflective mirrors would improve durability and ensure consistent performance in real environmental conditions. For practical rural deployment, the system should also be adapted to local solar patterns and community needs. Providing basic training for residents on system maintenance and troubleshooting will encourage sustainable and independent operation.

Through these enhancements, the hybrid solar lighting system has the potential to become a practical, scalable, and eco-friendly solution for rural communities, helping to improve energy access while reducing reliance on fossil fuels and supporting environmental sustainability.

REFERENCES

Anglistia, S. (2019). Prototype Sistem Pelacakan Sinar Matahari Pada Sistem Pembangkit Listrik Tenaga Surya Berbasis Arduino. Universitas Muhammadiyah Surakarta.

Birhane, E. M. (2019). Modeling and Analysis of Photo-Voltaic Solar Panel under Constant Electric Load. J. Renew. Energy.

Birhane, E. M. (2019). Modeling and Analysis of Photo-Voltaic Solar Panel under Constant Electric Load. J. Renew. Energy, 1-9.

Byregowda K. C, R. V. (2020). Study and Analysis of Arduino Based Solar Tracking Panel. International Research Journal of Engineering and Technology, 7.

Fauzi, K. W. (2018). Fauzi, Kodrat Wirawan, TePerancangan dan Realisasi Solar Tracking System Untuk Peningkatan Efisiensi Panel Surya Menggunakan Arduino Uno. TELKA-Telekomunikasi, Elektronika, Komputasi dan Kontrol, 63-74.

Hermita, R. (2024). Perancangan Pembuatan Maket Ruang Guru Pada Sekolah Budi Agung Medan. Teknologias, 2-6.

L. Putriyana, A. F. (2021). Simulasi Numerik Lapangan Panas Bumi Sirung Pantar, Kabupaten Alor, Nusa Tenggara Timur. Ketenagalistrikan dan Energi Terbarukan, 1, 1-12.

Pratik Pawar, A. Y. (2018). Solar Tracking System Using Arduino. International Journal of Scientific & Engineering Research, 8(9).