

DAFTAR PUSTAKA

- Arafat, A., & Ibrahim, I. (2020). SISTEM ALAT MONITORING UNTUK PENGENDALI SUHU DAN KELEMBABAN GREENHOUSE BERBASIS INTERNET OF THINGS. *INFO-TEKNIK*. <https://doi.org/10.31602/TJI.V11I4.3642>
- Ardiansah, I., & Bafdal, N. (2022). *Implementasi Otomasi Berbasis Internet of Things (IoT) Menggunakan Mikrokomputer untuk Pemantauan Iklim Mikro Rumah Kaca Implementation of Internet of Things (IoT)-Based Automation Using Microcomputers for Greenhouse Microclimate Monitorin*. <https://consensus.app/papers/implementasi-otomasi-berbasis-internet-of-things-iot-ardiansah-bafdal/98faba349a7e58a0a54c8922cd20d9dd/>
- Arifin, S., Nugraha, A. W., Mukti, F. S., & Jatmika, S. (2024). MQTT Broker Optimization: Comparative Analysis of Round Robin and Least Response Time. *Jurnal Nasional Teknik Elektro*, 127–136. <https://doi.org/10.25077/jnte.v13n3.1260.2024>
- Aung, T., Min Htun, Z. M., & Latt, K. (2019). *PC Based DC Motor Speed Control using PID for Laboratory—Consensus*. https://consensus.app/papers/pc-based-dc-motor-speed-control-using-pid-for-laboratory-aung-htun/7cd71210f269587c8059cf8d1b8555c1/?utm_source=chatgpt
- Azhari, A., Nasution, T. I., & Azis, P. F. A. (2023). *MPU-6050 Wheeled Robot Controlled Hand Gesture Using L298N Driver Based on Arduino—Consensus*. https://consensus.app/papers/mpu6050-wheeled-robot-controlled-hand-gesture-using-l298n-azhari-nasution/33b86056ed4953a88cb21d355629e910/?utm_source=chatgpt
- Battikh, A., Zaid, R., Tayeh, A., Kittana, A., Jallad, J., Alsadi, S., Foqha, T., Alahdal, D., & Kanan, M. (2023). Greenhouse Automation using ESP32: A Comprehensive Study on Monitoring and Controlling Environmental Parameters for Optimal Plant Growth. *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, 1–6. <https://doi.org/10.1109/EICEEAI60672.2023.10590110>
- Boufrida, A., & Boufaida, Z. (2022). Rule extraction from scientific texts: Evaluation in the specialty of gynecology. *Journal of King Saud University - Computer and Information Sciences*, 34(4), 1150–1160. <https://doi.org/10.1016/j.jksuci.2020.05.008>
- C, P. V. (2024). Monitoring and Controlling of Greenhouse. *International Journal for Research in Applied Science and Engineering Technology*. <https://doi.org/10.22214/ijraset.2024.60709>
- Chhorn, S., Tep, S., Hel, C., & Pec, R. (2022). *Development of ESP32-Based Smart Greenhouse Controller—Consensus*. <https://consensus.app/papers/development-of-esp32based-smart->

greenhouse-controller-chhorn-
tep/bc6f75fd376b5016aff165997a690b53/?utm_source=chatgpt

- Dharani, M., Bharathi, M., Praveena, K., & Venkata, K. M. T. (2022). Intelligent IoT-based greenhouse monitoring and control system. *i-manager's Journal on Electronics Engineering*. <https://doi.org/10.26634/jele.12.4.19061>
- Fahmi, A. (2019). *Rancang Bangun Sistem Kontrol Dan Monitoring TanamanCabai Rawit Berbasis Web Dengan Menggunakan InternetOf Things*. <https://consensus.app/papers/rancang-bangun-sistem-kontrol-dan-monitoring-fahmi/6caee66da7cf505484e01b85b2e729db/>
- Farooq, M., Riaz, S., Helou, M. A., Khan, F. S., Abid, A., & Alvi, A. (2022). *A Survey on IoT in Agriculture for the Implementation of Greenhouse Farming—Consensus*. https://consensus.app/papers/a-survey-on-iot-in-agriculture-for-the-implementation-of-khan-riaz/e92bcbce738b5b4da483e8803e171d0e/?utm_source=chatgpt
- Fauziyah, M., Safitri, H., Murtono, A., Dewatama, D., & Aulianta, E. (2021). *Pengaturan cahaya pada pertumbuhan bunga krisan potong di dalam prototype greenhouse*. 19, 64–71. <https://doi.org/10.33795/ELTEK.V19I1.269>
- Firmansyah, N. M. A., Anwar, F. K., Alamsyah, A., & Puspitasari, A. J. (2025). Precision farming for coffee: Real-Time monitoring and control systems. *Journal of Physics: Conference Series*. <https://doi.org/10.1088/1742-6596/2945/1/012035>
- Fitriasani, N. (2019). *IMPLEMENTASI MONITORING GREENHOUSE MENGGUNAKAN SENSOR DHT22, BH1750 SERTAPENGENDALI PENYIRAMAN DAN PENCAHAYAAN OTOMATIS BERBASIS ANDROID*. <https://consensus.app/papers/implementasi-monitoring-greenhouse-menggunakan-sensor-fitriasani/46e79e40177551f698a3e2e7748e1411/>
- Gumilang, M. P. A., Djunaedy, E., & Iskandar, R. F. (2018). *Studi Pengaruh Penggunaan Filter Kalman Pada Pengukuran Intensitas Cahaya Dalam Sistem Smart Home*. 5. <https://consensus.app/papers/studi-pengaruh-penggunaan-filter-kalman-pada-pengukuran-iskandar-gumilang/b3f91c731b9a5b31ad349549b4e3fe5e/>
- Hatkar, A., Mate, S., Ranmale, R., N.s, R. D. P., & Patil. (2024). IOT Based DC Motor Speed and Direction Control. *International Journal of Scientific Research in Science and Technology*. <https://doi.org/10.32628/ijsrst24112110>
- Hidayat, M. A. J., & Amrullah, A. (2022). SISTEM KONTROL DAN MONITORING TANAMAN HIDROPONIK BERBASIS INTERNET OF THINGS (IoT) MENGGUNAKAN NODEMCU ESP32. *Jurnal SAINTEKOM*. <https://doi.org/10.33020/saintekom.v12i1.223>
- Khair, U., & Jaya, P. (2024). Kontrol Otomatis dan Notifikasi pada Alat Angkat Keramba Ikan Berbasis Internet of Things (IoT). *Voteteknika (Vocational*

- Kodali, N. (2024). Tailwind CSS Integration in Angular: A Technical Overview. *International Journal of Innovative Research in Science, Engineering and Technology*. <https://doi.org/10.15680/ijirset.2024.1309092>
- Kurniasari, A. A., Puspitasari, P. S. D., Perdanasari, L., Yuana, D. B. M., & Jumiatun. (2025). Enhancing Hydroponic Systems with ESP32: An IoT Approach to Real-Time Monitoring and Automation. *IOP Conference Series: Earth and Environmental Science*. <https://doi.org/10.1088/1755-1315/1446/1/012010>
- Li, deqing, mei, honghui, shen, yi, su, shuang, zhang, wenli, wang, junting, & zu, ming. (2018). *ECharts: A declarative framework for rapid construction of web-based visualization*—ScienceDirect. <https://www.sciencedirect.com/science/article/pii/S2468502X18300068?via%3Dihub>
- Ma'arif, A., Maharani, N., Aditya, P., Rosyady, Raditya, A., Baswara, C., & Nuryono, A. A. (2020). Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation. *2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE)*, 74–78. <https://doi.org/10.1109/ICIEE49813.2020.9277258>
- Matei, D.-I., Negroiu, A., & Bucur, G. (2024). DC MOTOR ANGULAR POSITION CONTROL SYSTEM USING ARDUINO PLATFORM. *Romanian Journal of Petroleum & Gas Technology*. <https://doi.org/10.51865/jpgt.2024.02.07>
- Mattos, J. C., & Oliveira, F. L. (2022). *JSGuide: A Tool to Improve JavaScript Algorithms Focusing on IoT Devices*—Consensus. <https://consensus.app/papers/jsguide-a-tool-to-improve-javascript-algorithms-focusing-mattos-oliveira/5229814f43225bf7ac6342bf950b23f8/>
- Mukhlis, M., & Azhariyah, S. (2024). *Framework CSS: Tailwind CSS Untuk Front-End Website Store PT. XYZ - Consensus*. <https://consensus.app/papers/framework-css-tailwind-css-untuk-frontend-website-store-pt-mukhlis-azhariyah/2551b9eb57e759019bc229f30044ed37/>
- P, A., N, C., Viswanathan, P., S, S. M., & S, R. J. (2024). IoT Technology for Monitoring and Control of Smart Greenhouses. *IRO Journal on Sustainable Wireless Systems*. <https://doi.org/10.36548/jsws.2024.1.002>
- Prasetyo, M. A., & Wardana, H. K. (2021). Rancang Bangun Monitoring Solar Tracking System Menggunakan Arduino dan Nodemcu Esp 8266 Berbasis IoT. *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*. <https://doi.org/10.24853/resistor.4.2.163-168>

- Rahul, S. G., Rajkumar, R., Abburi, R. S. K., Reddy, B. T., Raja, C., & Teja, K. P. (2024). Condition Monitoring of Greenhouse Farming for Controlled Environment. *2024 3rd International Conference for Advancement in Technology (ICONAT)*, 1–6. <https://doi.org/10.1109/ICONAT61936.2024.10775257>
- Sarigiannidis, P. G., Bibi, S., & Terzi, A. (2021). *Reuse Opportunities in JavaScript applications—Consensus*. <https://consensus.app/papers/reuse-opportunities-in-javascript-applications-sarigiannidis-bibi/ac116a892c1c50e198f72ac3cd8ddaea/>
- Shahrosi, M. Y. N., Harijanto, A., & Nuraini, L. (2023). *Rancang Bangun Prototype Sistem Monitoring Suhu, Kelembaban dan Intensitas Cahaya pada Tanaman Cabai Berbasis IoT - Consensus*. https://consensus.app/papers/rancang-bangun-prototype-sistem-monitoring-suhu-harijanto-shahrosi/1de6abbefb9559d0b01354873ab35da7/?utm_source=chatgpt
- Sharma, P., Pandkar, M., & Nambiar, S. (2024). Wireless Greenhouse Monitoring using ESP32 and Sensor Array for Sustainable Crop Production. *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*, 1–6. <https://doi.org/10.1109/AIIoT58432.2024.10574742>
- Simanjuntak, M. N. (2022). *PERANCANGAN SISTEM MONITORING DAN KENDALI DERAJAT KEASAMAN (PH) PADA TANAMAN HIDROPONIK SISTEM NUTRIENT FILM TECHNIQUE (NFT) MENGGUNAKAN METODE KONTROL PID*. <https://consensus.app/papers/perancangan-sistem-monitoring-dan-kendali-derajat-simanjuntak/af239280d552544ca4e8255edd111bff/>
- Veena, c. (2024). Monitoring and Controlling of Greenhouse. *International Journal for Research in Applied Science and Engineering Technology*, 12(4), 3727–3731. <https://doi.org/10.22214/ijraset.2024.60709>
- Wang, J., Tang, L., Mu, L., Rao, S., & Zhang, W. (2023). *Bluetooth Mesh Network-based Intelligent Light Replenishment System for Agricultural Greenhouses—Consensus*. https://consensus.app/papers/bluetooth-mesh-networkbased-intelligent-light-wang-tang/e89ea1b3c5575fddb1e314f1ff735758/?utm_source=chatgpt
- Wisaksono, A., & Arifani, N. B. (2024). *IOT-BASED VANAME SHRIMP WATER QUALITY MONITORING PROTOTYPE USING GOOGLE SPREADSHEET - Consensus*. <https://consensus.app/papers/iotbased-vaname-shrimp-water-quality-monitoring-wisaksono-arifani/1b702db797d751d0a075c5ada5e686e3/>
- Yang, Z., Jun, Z., Xie, J., Xi, Z., & Tang, Y. (2023). *Design and Implementation of Epidemic Big Data Visualization Platform Based on Vue + ECharts—Consensus*. <https://consensus.app/papers/design-and-implementation-of-epidemic-big-data-yang-jun/c840c1c129045603bb78e782ba2ccbbc/>

Yuwono, W. (2017). *SIMULATOR CRANE OTOMATIS PEMINDAH PETI KEMAS PADA PROSES BONGKAR MUAT DI PELABUHAN*.
<https://consensus.app/papers/simulator-crane-otomatis-pemindah-peti-kemas-pada-proses-yuwono/5189253befd052e09bdf836a15946c7a/>

LAMPIRAN

1. Code pengembangan Web Dashboard :

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta content="width=device-width, initial-scale=1.0" name="viewport" />
    <title>Greenhouse Light IoT Dashboard</title>
    <!-- Tailwind CSS -->
    <script src="https://cdn.tailwindcss.com"></script>
    <!-- ECharts -->
    <script
src="https://cdn.jsdelivr.net/npm/echarts@5.4.3/dist/echarts.min.js"></script>
    <!-- MQTT.js -->
    <script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
    <!-- Font Awesome -->
    <link
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
      rel="stylesheet"
    />
    <script>
      tailwind.config = {
```

```

theme: {
  extend: {
    colors: {
      primary: "#8BC34A",
      secondary: "#689F38",
      accent: "#388E3C",
      dark: "#2E7D32",
      light: "#DCEDC8",
      background: "#EFF3F0",
    },
  },
};
</script>
<style>
body {
  background: linear-gradient(135deg, #f5f7fa 0%, #eff3f0 100%);
  min-height: 100vh;
  font-family: "Inter", sans-serif;
  overflow-x: hidden;
}

.neumorphic-card {
  background: linear-gradient(145deg, #f0f0f0, #ffffff);
  border-radius: 20px;
  box-shadow: 12px 12px 24px #d9d9d9, -12px -12px 24px #ffffff;

```

```
    transition: all 0.3s ease;
}
```

```
.neumorphic-card:hover {
    box-shadow: 6px 6px 12px #d9d9d9, -6px -6px 12px #ffffff;
}
```

```
.neumorphic-button {
    border-radius: 50px;
    background: linear-gradient(145deg, #f0f0f0, #ffffff);
    box-shadow: 6px 6px 12px #d9d9d9, -6px -6px 12px #ffffff;
    transition: all 0.2s ease;
    border: none;
    outline: none;
}
```

```
.neumorphic-button:active {
    box-shadow: inset 4px 4px 8px #d1d1d1, inset -4px -4px 8px #ffffff;
}
```

```
.neumorphic-input {
    background: linear-gradient(145deg, #f0f0f0, #ffffff);
    border-radius: 50px;
    box-shadow: inset 5px 5px 10px #d9d9d9, inset -5px -5px 10px #ffffff;
    border: none;
    outline: none;
}
```

```

}

.status-on {
  box-shadow: 0 0 10px rgba(139, 195, 74, 0.8);
}

.echarts-container {
  height: 100%;
  width: 100%;
}

.light-effect {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  height: 100px;
  background: linear-gradient(
    180deg,
    rgba(255, 255, 255, 0.7) 0%,
    rgba(255, 255, 255, 0) 100%
  );
  z-index: 0;
  pointer-events: none;
}
</style>

```

```

</head>

<body class="bg-background p-4 md:p-8">

  <div class="light-effect"></div>

  <!-- Dashboard Header -->

  <header class="mb-8 relative">

    <div

      class="flex flex-col md:flex-row justify-between items-start md:items-
center"

    >

      <div>

        <h1

          class="text-3xl md:text-4xl font-bold text-dark flex items-center"

        >

          <i class="fas fa-seedling mr-3 text-primary"></i>

          Greenhouse Light IoT Dashboard

        </h1>

        <p class="text-gray-600 mt-2">

          Monitor and control your greenhouse lighting system in real-time

        </p>

      </div>

      <div class="mt-4 md:mt-0 flex flex-col md:flex-row items-start">

        <div

          class="flex items-center bg-white p-3 rounded-xl shadow-sm mr-0
md:mr-4 mb-4 md:mb-0"

        >

          <div

```

```

        class="w-3 h-3 rounded-full bg-green-500 mr-2"
        id="connection-status-icon"
    ></div>

    <span class="text-gray-600 text-sm" id="connection-status"
    >Connecting to MQTT...</span>
    >
</div>

<div class="neumorphic-card p-4 flex items-center">
    <div class="mr-3">
        <div class="w-2 h-2 rounded-full bg-green-500"></div>
    </div>
    <div>
        <p class="text-xs text-gray-500">Last update</p>
        <p class="text-sm font-medium" id="last-updated">Just now</p>
    </div>
</div>
</div>
</div>
</header>

<main class="grid grid-cols-1 lg:grid-cols-3 gap-6">
    <!-- Left Panel -->
    <div class="lg:col-span-2">
        <!-- Environmental Data Cards -->
        <div class="grid grid-cols-1 md:grid-cols-3 gap-6 mb-6">
            <!-- Light Intensity Card -->
            <div class="neumorphic-card p-6 relative overflow-hidden">

```

```

<div
  class="absolute -right-4 -top-4 bg-primary/10 w-16 h-16 rounded-
full"
></div>

<div class="flex justify-between items-start">
  <div>
    <p class="text-gray-600 text-sm">Light Intensity</p>
    <h2 class="text-3xl font-bold text-dark mt-1">
      <span id="light-intensity-value">450</span>
      <span class="text-xl">Lux</span>
    </h2>
  </div>
  <div class="bg-primary/10 text-primary p-2 rounded-full">
    <i class="fas fa-sun text-xl"></i>
  </div>
</div>

<div class="mt-4">
  <div class="w-full bg-gray-200 rounded-full h-2.5 mb-2">
    <div
      class="bg-gradient-to-r from-green-400 to-primary h-2.5 rounded-
full"
      style="width: 65%"
    ></div>
  </div>
  <p class="text-xs text-gray-500 flex justify-between">
    <span>Optimal: 40.000 Lux</span>

```

```

        <span class="font-medium text-primary" id="light-status"
        >Optimal</span>
    >
</p>
</div>
</div>
<!-- Humidity Card -->
</div>
<!-- Light Intensity Meter -->
<div class="neumorphic-card p-6 mb-6 h-96">
    <div class="flex justify-between items-center mb-6">
        <h3 class="text-xl font-semibold text-gray-800">
            Light Intensity Meter
        </h3>
        <div class="flex text-sm">
            <span class="px-2 py-1 bg-primary/10 text-primary rounded-full"
            >Real-time</span>
        >
    </div>
</div>
<div class="echarts-container" id="light-meter"></div>
</div>
<!-- Historical Light Data -->
<div class="neumorphic-card p-6 h-96">
    <div class="flex justify-between items-center mb-6">
        <h3 class="text-xl font-semibold text-gray-800">

```

Light Intensity History

</h3>

<div class="flex space-x-2">

<button

class="text-xs px-3 py-1 neumorphic-button text-gray-600"

data-interval="1"

>

1H

</button>

<button

class="text-xs px-3 py-1 neumorphic-button text-gray-600"

data-interval="2"

>

2H

</button>

<button

class="text-xs px-3 py-1 neumorphic-button text-gray-600"

data-interval="3"

>

3H

</button>

</div>

</div>

<div class="echarts-container" id="light-history"></div>

</div>

</div>

```

<!-- Right Panel -->

<div>

  <!-- Motor Control Panel -->

  <div class="neumorphic-card p-6 mb-6">

    <div class="flex justify-between items-center mb-6">

      <h3 class="text-xl font-semibold text-gray-800">Motor Control</h3>

      <div class="relative">

        <span class="flex items-center" id="motor-status-indicator">

          <span class="w-2 h-2 rounded-full bg-gray-400 mr-2"></span>

          <span class="text-sm text-gray-500" id="motor-status">OFF</span>

        </span>

      </div>

    </div>

    <div class="flex flex-col items-center">

      <div

        class="w-40 h-40 rounded-full mb-6 neumorphic-card status-off flex
items-center justify-center cursor-pointer relative"

        id="motor-control"

      >

        <div

          class="absolute inset-0 rounded-full bg-gray-200 opacity-70"

        ></div>

        <div

          class="bg-white w-36 h-36 rounded-full neumorphic-button flex
items-center justify-center"

        >

```

```

        <i class="fas fa-power-off text-5xl text-gray-400"></i>
    </div>
</div>
<p class="text-gray-600 text-center mb-6">
    Click the power button to control the greenhouse motor
</p>
<div class="grid grid-cols-2 gap-4 w-full">
    <div class="neumorphic-card p-4 text-center">
        <p class="text-sm text-gray-500 mb-1">Auto Mode</p>
        <div class="relative inline-block w-12 h-6">
            <label
                class="relative inline-block w-12 h-6 cursor-pointer"
                for="auto-mode"
            >
                <input class="sr-only" id="auto-mode" type="checkbox" />
                <div
                    class="block bg-gray-300 w-12 h-6 rounded-full"
                    id="auto-mode-track"
                ></div>
                <div
                    class="dot absolute left-1 top-1 bg-white w-4 h-4 rounded-full
transition"
                ></div>
            </label>
        </div>
    </div>
</div>

```

```

<div class="neumorphic-card p-4 text-center">
  <p class="text-sm text-gray-500 mb-1">Intensity Threshold</p>
  <p class="font-medium text-primary text-lg">70.000 lux</p>
</div>
</div>
</div>
</div>
<!-- Plant Recommendations -->
<div class="neumorphic-card p-6">
  <h3 class="text-xl font-semibold text-gray-800 mb-6">
    Plant Recommendations
  </h3>
  <div class="space-y-4">
    <div class="flex items-start">
      <div class="mr-4 bg-primary/10 p-2 rounded-xl">
        <i class="fas fa-seedling text-primary text-xl"></i>
      </div>
      <div>
        <h4 class="font-medium text-gray-800">
          Lettuce & Leafy Greens
        </h4>
        <p class="text-sm text-gray-600">
          Optimal light: 10.000-20.000 lux
        </p>
      </div>
    </div>
  </div>
</div>

```

```

<div class="flex items-start">
  <div class="mr-4 bg-orange-500/10 p-2 rounded-xl">
    <i class="fas fa-apple-alt text-orange-500 text-xl"></i>
  </div>
  <div>
    <h4 class="font-medium text-gray-800">Tomatoes</h4>
    <p class="text-sm text-gray-600">
      Optimal light: 30.000-50.000 lux
    </p>
  </div>
</div>
<div class="flex items-start">
  <div class="mr-4 bg-purple-500/10 p-2 rounded-xl">
    <i class="fas fa-spa text-purple-500 text-xl"></i>
  </div>
  <div>
    <h4 class="font-medium text-gray-800">Herbs</h4>
    <p class="text-sm text-gray-600">
      Optimal light: 20.000-30.000 lux
    </p>
  </div>
</div>
<div class="mt-8 pt-6 border-t border-gray-100">
  <h4 class="text-md font-medium text-gray-800 mb-4">
    Tips for Healthy Plants
  </h4>
</div>

```

```

</h4>
<ul class="space-y-3">
  <li class="flex items-start">
    <i class="fas fa-check-circle text-primary mr-2 mt-1"></i>
    <p class="text-sm text-gray-600">
      Keep light intensity consistent throughout the day
    </p>
  </li>
  <li class="flex items-start">
    <i class="fas fa-check-circle text-primary mr-2 mt-1"></i>
    <p class="text-sm text-gray-600">
      Water in the morning for optimal absorption
    </p>
  </li>
</ul>
</div>
</div>
</main>
<!-- Footer -->
<footer class="mt-10 text-center text-gray-500 text-sm">
  <p>Greenhouse IoT Dashboard • • Real-time Monitoring</p>
</footer>
<script>
  // Connect to MQTT
  const clientId =

```

```

    "greenhouse-dash-" + Math.random().toString(16).substring(2, 10);

const client = mqtt.connect("wss://broker.emqx.io:8084/mqtt", {
  clientId: clientId,
  clean: true,
  connectTimeout: 4000,
  reconnectPeriod: 1000,
});

// DOM elements
const lightValue = document.getElementById("light-intensity-value");
const lightStatus = document.getElementById("light-status");
const motorStatus = document.getElementById("motor-status");
const motorStatusIndicator = document.getElementById(
  "motor-status-indicator"
);
const motorControl = document.getElementById("motor-control");
const connectionStatus = document.getElementById("connection-
status");
const connectionStatusIcon = document.getElementById(
  "connection-status-icon"
);
const lastUpdated = document.getElementById("last-updated");

// Data arrays
let lightData = [];
const maxDataPoints = 20;

```

```

// Initialize ECharts

const lightMeterChart = echarts.init(
  document.getElementById("light-meter")
);

const lightHistoryChart = echarts.init(
  document.getElementById("light-history")
);


// MQTT connection handlers

client.on("connect", () => {
  console.log("Connected to MQTT broker");
  connectionStatus.textContent = "Connected";
  connectionStatusIcon.className =
    "w-3 h-3 rounded-full bg-green-500 mr-2";
  client.subscribe("greenhouse/sensor");
  client.subscribe("greenhouse/motor");
});

client.on("reconnect", () => {
  connectionStatus.textContent = "Reconnecting...";
  connectionStatusIcon.className =
    "w-3 h-3 rounded-full bg-yellow-500 mr-2";
});

client.on("offline", () => {

```

```

        connectionStatus.textContent = "Offline";

        connectionStatusIcon.className = "w-3 h-3 rounded-full bg-red-500 mr-
2";

    });

    client.on("error", (error) => {
        console.error("MQTT Error:", error);
    });

    client.on("message", (topic, message) => {
        console.log(`Received message on ${topic}: ${message.toString()}`);

        if (topic === "greenhouse/sensor") {
            updateLightData(message.toString());
        } else if (topic === "greenhouse/motor") {
            updateMotorStatus(message.toString());
        }
    });

    function updateLightData(data) {
        try {
            lastDataTime = Date.now(); // ➡ Tambahkan baris ini
            // Simpan waktu terakhir update ke localStorage
            localStorage.setItem("lastUpdateTime", lastDataTime);
            const sensorData = JSON.parse(data);
            if (sensorData.lux !== undefined) {

```

```

const luxValue = sensorData.lux;

lightValue.textContent = luxValue;

// Determine light status
if (luxValue < 25000) {
    lightStatus.textContent = "Low Light";
    lightStatus.className = "font-medium text-yellow-500";
} else if (luxValue > 65000) {
    lightStatus.textContent = "High Light";
    lightStatus.className = "font-medium text-red-500";
} else {
    lightStatus.textContent = "Optimal";
    lightStatus.className = "font-medium text-primary";
}

// Add to historical data
lightData.push({
    time: new Date().toLocaleTimeString([], {
        hour: "2-digit",
        minute: "2-digit",
    }),
    value: luxValue,
});

// Keep only last 20 data points
if (lightData.length > maxDataPoints) {

```

```

        lightData.shift();
    }

    // Update charts
    updateLightMeter(luxValue);
    updateLightHistoryChart();

    // Update timestamp
    lastUpdated.textContent = new Date().toLocaleTimeString([], {
        hour: "2-digit",
        minute: "2-digit",
    });
}
} catch (e) {
    console.error("Error parsing sensor data:", e);
}
}

function updateMotorStatus(status) {
    motorStatus.textContent = status;

    if (status === "ON") {
        motorStatusIndicator.innerHTML = `
            <span class="w-2 h-2 rounded-full bg-green-500 mr-2"></span>
            <span class="text-sm font-medium text-green-500">ON</span>
        `;
    }
}

```

```

    motorControl.classList.remove("status-off");

    motorControl.classList.add("status-on");

    motorControl.querySelector(".fa-power-off").className =
        "fas fa-power-off text-5xl text-primary";
} else {
    motorStatusIndicator.innerHTML = `
        <span class="w-2 h-2 rounded-full bg-gray-400 mr-2"></span>
        <span class="text-sm text-gray-500">OFF</span>
    `;

    motorControl.classList.remove("status-on");

    motorControl.classList.add("status-off");

    motorControl.querySelector(".fa-power-off").className =
        "fas fa-power-off text-5xl text-gray-400";
}
}

function toggleMotor() {
    const currentState = motorStatus.textContent;
    const newState = currentState === "ON" ? "OFF" : "ON";

    // Publish new state
    client.publish("greenhouse/motor", newState, { qos: 0, retain: false });
}

function updateLightMeter(value) {
    const options = {

```

```

series: [
  {
    type: "gauge",
    min: 0,
    max: 100000,
    splitNumber: 10,
    radius: "90%",
    center: ["50%", "50%"],
    axisLine: {
     LineStyle: {
        width: 15,
        color: [
          [0.3, "#90EE90"],
          [0.7, "#FDDD60"],
          [1, "#FF6E76"],
        ],
      },
    },
    axisLabel: {
      distance: -25,
      color: "#999",
      fontSize: 12,
    },
    anchor: {
      show: true,
      showAbove: true,

```

```

        size: 18,

        itemStyle: {
            color: "#FAC858",
        },
    },
    detail: {
        valueAnimation: true,
        fontSize: 20,
        offsetCenter: [0, "65%"],
        formatter: "{value} lux",
        color: "auto",
    },
    data: [
        {
            value: value,
        },
    ],
},
],
};

lightMeterChart.setOption(options);
}

function updateLightHistoryChart() {
    const times = lightData.map((d) => d.time);

```

```
const values = lightData.map((d) => d.value);
```

```
const options = {  
  grid: {  
    top: "10%",  
    left: "1%",  
    right: "1%",  
    bottom: "10%",  
    containLabel: true,  
  },  
  xAxis: {  
    type: "category",  
    boundaryGap: false,  
    data: times,  
  },  
  yAxis: {  
    type: "value",  
    min: 0,  
    max: 100000,  
    splitLine: {  
      show: true,  
     LineStyle: {  
        color: "#E4E4E4",  
      },  
    },  
  },  
  axisLabel: {
```

```

        formatter: "{value} lux",
    },
},
series: [
    {
        data: values,
        type: "line",
        smooth: true,
        lineStyle: {
            width: 4,
            color: "#8BC34A",
        },
        itemStyle: {
            color: "#8BC34A",
        },
        areaStyle: {
            color: {
                type: "linear",
                x: 0,
                y: 0,
                x2: 0,
                y2: 1,
                colorStops: [
                    {
                        offset: 0,
                        color: "rgba(139, 195, 74, 0.5)",

```

```

    },
    {
      offset: 1,
      color: "rgba(139, 195, 74, 0.1)",
    },
  ],
},
},
},
},
],
tooltip: {
  trigger: "axis",
  backgroundColor: "rgba(255,255,255,0.95)",
  borderColor: "#E4E4E4",
  textStyle: {
    color: "#333",
  },
  axisPointer: {
    type: "line",
    lineStyle: {
      color: "#DCEDC8",
    },
  },
},
},
};

```

```

    lightHistoryChart.setOption(options);
}

// Initial charts
updateLightMeter(450);
updateLightHistoryChart();

// Motor control click event
motorControl.addEventListener("click", toggleMotor);

// Auto mode toggle
const autoModeToggle = document.getElementById("auto-mode");
const autoModeTrack = document.getElementById("auto-mode-track");
autoModeToggle.addEventListener("change", function () {
    const isChecked = this.checked;

    if (isChecked) {
        this.nextElementSibling.nextElementSibling.style.transform =
            "translateX(24px)";
        autoModeTrack.style.backgroundColor = "#8BC34A";
        client.publish("greenhouse/auto", "ON");
    } else {
        this.nextElementSibling.nextElementSibling.style.transform =
            "translateX(0)";
        autoModeTrack.style.backgroundColor = "#D1D5DB";
        client.publish("greenhouse/auto", "OFF");
    }
});

```

```

    }

    console.log("Auto Mode toggled:", isChecked ? "ON" : "OFF");
  });

  // Responsive chart resizing
  window.addEventListener("resize", function () {
    lightMeterChart.resize();
    lightHistoryChart.resize();
  });

  // Generate initial data for demo
  const initialTime = new Date();
  for (let i = maxDataPoints; i > 0; i--) {
    const time = new Date(initialTime.getTime() - i * 30000);
    lightData.push({
      time: time.toLocaleTimeString([], {
        hour: "2-digit",
        minute: "2-digit",
      }),
      value: Math.floor(Math.random() * 300) + 350,
    });
  }
  updateLightHistoryChart();
</script>
<script>

```

```

let lastDateTime = Date.now();

// Coba ambil waktu update terakhir dari localStorage saat halaman
dibuka

const savedUpdateTime = localStorage.getItem("lastUpdateTime");
if (savedUpdateTime) {
    lastDateTime = parseInt(savedUpdateTime);
}

setInterval(() => {
    const now = Date.now();
    const elapsed = now - lastDateTime;

    if (elapsed > 5000) {
        connectionStatus.textContent = "Not Connected";
        connectionStatusIcon.className =
            "w-3 h-3 rounded-full bg-red-500 mr-2";

        // Ambil waktu terakhir update dari localStorage
        const storedTime = localStorage.getItem("lastUpdateTime");
        if (storedTime) {
            const lastDate = new Date(parseInt(storedTime));
            lastUpdated.textContent = lastDate.toLocaleTimeString([], {
                hour: "2-digit",
                minute: "2-digit",
            });
        } else {
            lastUpdated.textContent = "Unknown";
        }
    }
}, 5000);

```

```

    }
  } else {
    connectionStatus.textContent = "Connected";
    connectionStatusIcon.className =
      "w-3 h-3 rounded-full bg-green-500 mr-2";
    lastUpdated.textContent = "Just now";
  }
}, 1000);
</script>
<script>
  let activeInterval = null;

  const timeButtons = document.querySelectorAll("[data-interval]");
  timeButtons.forEach((button) => {
    button.addEventListener("click", () => {
      const hours = parseInt(button.getAttribute("data-interval"));

      if (activeInterval === hours) {
        activeInterval = null;
        updateLightHistoryChart();

        timeButtons.forEach((btn) => {
          btn.classList.remove(
            "text-green-600",
            "font-bold",
            "bg-primary/10"

```

```

    );
    btn.classList.add("text-gray-600");
  });
} else {
  activeInterval = hours;

  const now = new Date();
  const cutoff = now.getTime() - hours * 24 * 60 * 60 * 1000;

  const filtered = lightData.filter((d) => {
    const time = new Date();
    const [h, m] = d.time.split(":").map(Number);
    time.setHours(h, m, 0);
    return time.getTime() >= cutoff;
  });

  const times = filtered.map((d) => d.time);
  const values = filtered.map((d) => d.value);

  lightHistoryChart.setOption({
    xAxis: { data: times },
    series: [{ data: values }],
  });

  timeButtons.forEach((btn) => {
    btn.classList.remove(

```

```

        "text-green-600",
        "font-bold",
        "bg-primary/10"
    );
    btn.classList.add("text-gray-600");
});

button.classList.remove("text-gray-600");
button.classList.add(
    "text-green-600",
    "font-bold",
    "bg-primary/10"
);
}
});
});
</script>
</body>
</html>

```

2. Code Arduino IDE :

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <BH1750.h>
#include <HTTPClient.h>

// WiFi
const char* ssid = "Teknik21";

```

```

const char* password = "Teknik21";

// MQTT
const char* mqtt_server = "broker.emqx.io";
const int mqtt_port = 1883;
const char* mqtt_client_id = "esp32-bh1750-client";
const char* sensor_topic = "greenhouse/sensor";
const char* motor_topic = "greenhouse/motor";
const char* auto_topic = "greenhouse/auto";

// Motor driver pins
const int IN1 = 26;
const int IN2 = 27;
const int ENA = 25;

// Sensor
BH1750 lightSensor;

// MQTT
WiFiClient espClient;
PubSubClient client(espClient);

// Kontrol Mode
bool autoMode = false;
unsigned long motorStartTime = 0;
bool motorRunning = false;
float lastLux = -1;
bool actionTriggered = false;

void motorRight() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 255);
}

void motorLeft() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 255);
}

void motorOff() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 0);
}

```

```

}

void callback(char* topic, byte* payload, unsigned int length) {
    String message;
    for (int i = 0; i < length; i++) {
        message += (char)payload[i];
    }

    if (String(topic) == auto_topic) {
        autoMode = (message == "ON");
        Serial.print("Auto Mode: ");
        Serial.println(autoMode ? "ON" : "OFF");
    }

    if (String(topic) == motor_topic && !autoMode) {
        if (message == "ON") {
            motorRight();
            Serial.println("Motor ON (Manual)");
        } else if (message == "OFF") {
            motorOff();
            Serial.println("Motor OFF (Manual)");
        }
    }
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Connecting to MQTT...");
        if (client.connect(mqtt_client_id)) {
            Serial.println("Connected!");
            client.subscribe(motor_topic);
            client.subscribe(auto_topic);
        } else {
            Serial.print("Failed, rc=");
            Serial.print(client.state());
            Serial.println(" retrying in 5s");
            delay(5000);
        }
    }
}

void setup() {
    Serial.begin(115200);
    Wire.begin(18, 19);
    lightSensor.begin();
}

```

```

pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(ENA, OUTPUT);
motorOff();

WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println(" Connected!");

client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    float lux = lightSensor.readLightLevel();
    Serial.print("Lux: ");
    Serial.println(lux);

    // Kirim ke MQTT
    String payload = "{\"lux\": " + String(lux, 2) + "}";
    client.publish(sensor_topic, payload.c_str());

    // Kirim ke Google Sheets
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin("https://script.google.com/macros/s/AKfycbx_1Yefnt
jyJ2_VtYrF-hNuX4dnwbAB_H7x-
siG1SWfZC8gs8ckjNw9lQgnSBkvbGF5/exec");
        http.addHeader("Content-Type", "application/json");

        String json = "{\"lux\": " + String(lux, 2) + "}";
        int response = http.POST(json);
        Serial.print("Google Sheets Response: ");
        Serial.println(response);
        http.end();
    }
}

```

```

}

// Mode Auto: aksi hanya jika terjadi perubahan signifikan
if (autoMode && !motorRunning) {
  if (lastLux >= 0) { // pastikan bukan nilai awal
    if (lastLux <= 5000 && lux > 5000) {
      motorRight();
      motorStartTime = millis();
      motorRunning = true;
      actionTriggered = true;
      Serial.println("Motor RIGHT (Lux naik > 5000)");
    }
    else if (lastLux >= 5000 && lux < 5000) {
      motorLeft();
      motorStartTime = millis();
      motorRunning = true;
      actionTriggered = true;
      Serial.println("Motor LEFT (Lux turun < 5000)");
    }
  }
  lastLux = lux;
}

// Matikan motor setelah 10 detik
if (motorRunning && millis() - motorStartTime >= 10000) {
  motorOff();
  motorRunning = false;
  Serial.println("Motor OFF (Timer)");
}

delay(5000);
}

```

3. Data Tabulasi Kuisioner

X.I. Persepsi Kegunaan (PU)						X1 TOTAL	X.II. Persepsi Kemudahan Penggunaan (PEOU)					X2 TOTAL
X1.1	X1.2.	X1.3.	X1.4.	X1.5.			X2.1	X2.2	X2.3	X2.4	X2.5	
4	4	4	5	4		21	4	4	4	4	4	20
4	4	4	4	5		21	4	3	4	3	4	18
4	4	4	4	4		20	4	3	4	4	4	19
4	4	4	4	4		20	4	4	4	4	4	20
5	5	5	5	5		25	5	4	4	4	4	21
4	4	4	2	4		18	2	2	2	4	2	12
4	3	2	4	4		17	2	2	5	5	4	18
4	4	4	2	4		18	2	2	2	4	2	12
4	4	4	4	4		20	4	3	5	5	5	22
4	4	4	4	4		20	5	3	5	5	5	23
4	3	5	5	4		21	3	5	5	5	5	23
4	3	5	5	3		20	3	5	5	5	5	23
5	4	4	5	3		21	5	4	5	5	3	22
5	5	5	5	4		24	4	3	5	4	4	20
5	4	5	5	4		23	4	4	5	5	4	22
4	2	4	2	3		15	3	2	4	4	3	16
4	4	4	4	4		20	4	4	4	4	4	20
4	4	4	4	3		19	2	2	4	4	2	14
4	4	4	3	2		17	2	2	4	4	2	14
4	4	5	4	4		21	2	3	4	4	2	15
4	4	4	3	3		18	2	3	4	4	2	15

3	3	5	5	5	21	4	3	4	5	4	20
3	3	4	4	3	17	4	4	4	4	4	20
5	4	4	4	4	21	5	4	1	4	4	18
4	4	5	5	4	22	5	4	4	4	4	21
4	3	4	3	4	18	3	2	3	4	2	14
4	3	4	4	4	19	4	4	4	5	4	21
4	3	3	3	3	16	4	2	3	4	3	16
3	3	3	3	4	16	3	3	3	4	3	16
4	4	4	3	4	19	4	4	4	5	4	21
3	3	4	3	4	17	4	3	4	5	4	20
4	4	4	4	4	20	4	4	4	4	4	20
4	4	4	4	4	20	4	4	4	4	4	20
4	4	4	4	4	20	4	4	4	4	4	20
4	4	4	4	4	20	4	3	3	4	4	18
4	3	4	3	4	18	4	4	4	4	4	20
4	3	4	3	4	18	4	3	3	4	3	17
4	3	4	3	4	18	4	3	4	5	4	20
4	3	4	3	4	20	4	4	4	4	4	20
3	3	4	3	4	17	4	3	4	5	3	19
4	4	4	4	4	20	4	4	4	5	5	22

Keterangan : X1 = Item Pertanyaan tentang Presepsi Kenggunaan (PU)

X2 = Item Pertanyaan tentang Persepsi Kemudahan Penggunaan (PEOU)

X3 = Item Pertanyaan tentang Attitude Toward Using (ATU)

X4 = Item Pertanyaan tentang Behavioral Intention To Use (BITU)

4. Data Pengukuran Intensitas Cahaya di Greenhouse Instiper

Tabel Hasil Pengukuran Intensitas Cahaya Pagi Hari Pukul 09.08 Hingga 11.02

WIB

Waktu	Lux	Waktu	Lux	Waktu	Lux	Waktu	Lux
9:08:02	35276	9:36:48	45694	10:05:34	53453	10:34:19	54613
9:09:05	35303	9:37:52	45861	10:06:38	52067	10:35:23	54613
9:10:09	36315	9:38:56	47327	10:07:42	54322	10:36:27	54613
9:11:13	35420	9:39:59	47324	10:08:45	54094	10:37:31	54613
9:12:17	27733	9:41:05	45528	10:09:50	53255	10:38:35	54613
9:13:24	37596	9:42:09	46292	10:10:53	54613	10:39:39	54613
9:14:28	37204	9:43:12	47723	10:11:57	54613	10:40:42	54613
9:15:31	37463	9:44:16	48068	10:13:01	54233	10:41:47	54613
9:16:35	37935	9:45:20	48322	10:14:05	48493	10:42:50	54613
9:17:39	38248	9:46:24	48482	10:15:08	47409	10:43:54	54613
9:18:43	38318	9:47:28	49336	10:16:12	44695	10:44:57	54613
9:19:46	39058	9:48:32	50060	10:17:17	49811	10:46:01	54613
9:20:50	39092	9:49:36	49145	10:18:21	50486	10:47:04	54613
9:21:54	39870	9:50:40	49833	10:19:24	52163	10:48:08	54613
9:22:58	39810	9:51:43	50343	10:20:30	51745	10:49:12	54613
9:24:02	40410	9:52:48	51107	10:21:33	54022	10:50:17	54613
9:25:06	41163	9:53:52	51363	10:22:37	54613	10:51:20	54613
9:26:09	41573	9:54:56	51821	10:23:40	54613	10:52:25	54613
9:27:14	42037	9:55:59	51553	10:24:44	54613	10:53:32	54613

9:28:17	42688	9:57:03	52506	10:25:48	54613	10:54:35	54613
9:29:22	43424	9:58:07	52288	10:26:52	54613	10:55:40	54613
9:30:26	41725	9:59:11	52200	10:27:55	54520	10:56:44	54613
9:31:30	43192	10:00:14	53245	10:28:59	54613	10:57:47	54613
9:32:34	43199	10:01:18	52263	10:30:03	54613	10:58:51	54613
9:33:38	44654	10:02:21	52743	10:31:08	54613	10:59:54	54613
9:34:41	45290	10:03:26	52662	10:32:11	54613	11:00:59	54613
9:35:45	45335	10:04:30	52864	10:33:16	54613	11:02:02	54613

Tabel Hasil Pengukuran Intensitas Cahaya Siang Hari Pukul 11.03 Hingga 15.00
WIB

Waktu	Lux	Waktu	Lux	Waktu	Lux	Waktu	Lux
11:03:06	54613	12:02:46	54613	13:02:29	54613	14:02:09	41009
11:04:09	54613	12:03:49	54613	13:03:33	54613	14:03:13	40393
11:05:14	54613	12:04:54	54613	13:04:39	54613	14:04:17	9743
11:06:17	54613	12:05:57	54613	13:05:42	54613	14:05:20	7008
11:07:21	54613	12:07:01	54613	13:06:46	54613	14:06:24	6904
11:08:24	54613	12:08:05	54613	13:07:49	54613	14:07:27	8960
11:09:29	54613	12:09:09	54613	13:08:54	54613	14:08:31	15535
11:10:32	54613	12:10:12	54613	13:09:57	54613	14:09:35	8500
11:11:36	54613	12:11:16	54613	13:11:01	54613	14:10:39	5958
11:12:39	54613	12:12:20	54613	13:12:04	54613	14:11:43	5434
11:13:43	54613	12:13:32	54613	13:13:08	54613	14:12:47	5547

11:14:47	54613	12:14:35	54613	13:14:12	54613	14:13:50	6083
11:15:51	54613	12:15:39	54613	13:15:16	54392	14:14:54	8788
11:16:54	54613	12:16:43	54613	13:16:20	52559	14:15:57	31502
11:17:59	54613	12:17:46	54613	13:17:24	50161	14:17:01	38014
11:19:02	54613	12:18:50	54613	13:18:27	48853	14:18:05	37415
11:20:08	54613	12:19:54	54613	13:19:32	50562	14:19:09	37118
11:21:11	54613	12:20:57	54613	13:20:35	49404	14:20:12	37774
11:22:16	54613	12:22:01	54613	13:21:39	44374	14:21:17	36575
11:23:20	54613	12:23:05	54613	13:22:42	44588	14:22:20	36466
11:24:25	54613	12:24:09	54613	13:23:48	43193	14:23:24	35723
11:25:28	54613	12:25:12	54613	13:24:52	45068	14:24:27	35299
11:26:33	54613	12:26:16	54613	13:25:55	45723	14:25:32	34683
11:27:37	54613	12:27:19	54613	13:26:59	46373	14:26:35	35014
11:28:41	54613	12:28:24	54613	13:28:03	47735	14:27:39	34908
11:29:44	52748	12:29:27	54613	13:29:06	50787	14:28:42	34121
11:30:48	49091	12:30:31	54613	13:30:10	53060	14:29:47	34016
11:31:51	54196	12:31:35	54613	13:31:14	53814	14:30:50	33693
11:32:55	53891	12:32:40	54613	13:32:18	53562	14:31:54	32061
11:33:59	49264	12:33:43	54613	13:33:21	52723	14:32:57	30962
11:35:05	51812	12:34:47	54613	13:34:26	41293	14:34:01	7840
11:36:08	49351	12:35:50	54613	13:35:31	52115	14:35:05	6988
11:37:12	54613	12:36:54	54613	13:36:35	51692	14:36:09	11003

11:38:16	54613	12:37:58	54613	13:37:42	51515	14:37:13	29294
11:39:20	54613	12:39:02	54613	13:38:46	50772	14:38:17	10783
11:40:23	54613	12:40:05	54613	13:39:50	49417	14:39:20	29031
11:41:27	54613	12:41:09	54613	13:40:54	48360	14:40:25	15453
11:42:31	54613	12:42:13	54613	13:41:57	45993	14:41:28	16594
11:43:35	54613	12:43:17	54613	13:43:01	43319	14:42:32	22803
11:44:39	54613	12:44:21	54613	13:44:04	46076	14:43:35	5772
11:45:44	54613	12:45:24	54613	13:45:09	46451	14:44:40	5345
11:46:47	54613	12:46:28	54613	13:46:12	45430	14:45:43	4613
11:47:51	54402	12:47:33	54613	13:47:16	47515	14:46:47	4241
11:48:55	54613	12:48:37	54613	13:48:20	47193	14:47:50	4353
11:49:59	54613	12:49:42	54613	13:49:23	43549	14:48:56	3989
11:51:03	54613	12:50:45	54613	13:50:27	44848	14:49:59	3533
11:52:07	32382	12:51:50	54613	13:51:31	47583	14:51:04	3373
11:53:10	22563	12:52:53	54613	13:52:34	45760	14:52:07	3318
11:54:14	54613	12:53:57	54613	13:53:38	46335	14:53:11	3500
11:55:18	54613	12:55:03	54613	13:54:41	45335	14:54:15	3935
11:56:22	54613	12:56:07	54613	13:55:46	43160	14:55:19	4453
11:57:25	54613	12:57:10	54613	13:56:50	41815	14:56:22	7601
11:58:30	54613	12:58:14	54613	13:57:54	39664	14:57:26	6426
11:59:34	54613	12:59:18	54613	13:58:58	37388	14:58:30	8324
12:00:38	54613	13:00:22	54613	14:00:01	40366	14:59:34	11579

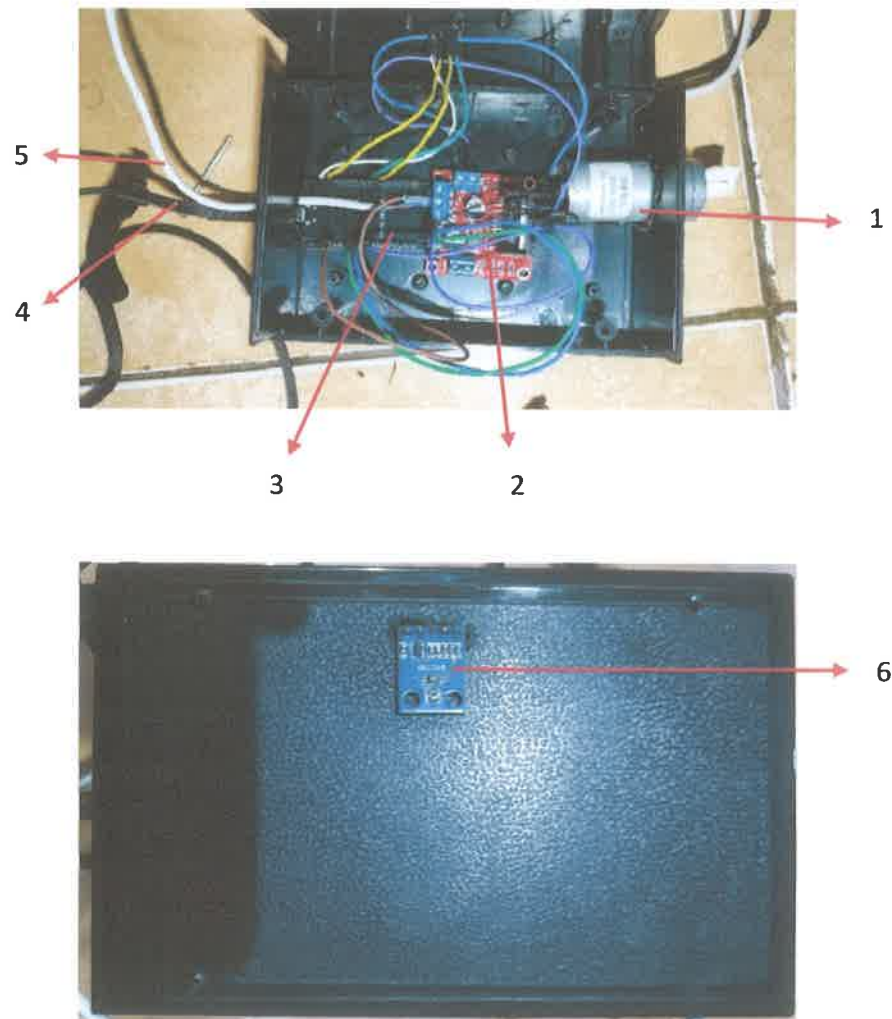
12:01:41	54613	13:01:25	54613	14:01:05	40992	15:00:37	22069
----------	-------	----------	-------	----------	-------	----------	-------

Tabel Hasil Pengukuran Intensitas Cahaya Sore Hari Pukul 15.01 Hingga 17.30
WIB

Waktu	Lux	Waktu	Lux	Waktu	Lux	Waktu	Lux
15:01:41	22526	15:40:05	3072	16:17:20	1914	16:54:37	327
15:02:44	21962	15:41:08	2987	16:18:24	1833	16:55:40	320
15:03:49	21225	15:42:13	2933	16:19:28	1853	16:56:45	290
15:06:17	7721	15:43:16	2958	16:20:32	1835	16:57:48	261
15:07:06	13330	15:44:21	3087	16:21:35	1750	16:58:52	261
15:08:10	20453	15:45:24	3287	16:22:39	1702	16:59:56	288
15:09:14	18025	15:46:29	3480	16:23:43	1729	17:01:00	288
15:10:18	10977	15:47:33	3638	16:24:47	1728	17:02:03	273
15:11:22	4370	15:48:37	3710	16:25:50	1593	17:03:07	245
15:12:27	3933	15:49:40	3671	16:26:54	1414	17:04:11	213
15:13:29	3748	15:50:44	3630	16:27:57	1205	17:05:15	178
15:14:34	3623	15:51:48	3606	16:29:01	1019	17:06:18	158
15:15:37	3683	15:52:52	3569	16:30:04	882	17:07:22	151
15:16:41	3828	15:53:55	3520	16:31:10	727	17:08:26	147
15:17:44	3988	15:55:01	3448	16:32:13	589	17:09:31	136
15:18:49	4082	15:56:04	3391	16:33:18	538	17:10:34	124
15:19:52	4065	15:57:08	3328	16:34:21	519	17:11:38	111
15:20:56	4010	15:58:12	3276	16:35:26	518	17:12:41	100

15:21:59	3897	15:59:16	3205	16:36:29	509	17:13:45	89
15:23:04	3642	16:00:19	3079	16:37:33	500	17:14:48	80
15:24:07	3415	16:01:23	2883	16:38:37	538	17:15:53	71
15:25:11	3223	16:02:27	2761	16:39:41	627	17:16:57	62
15:26:15	3230	16:03:31	2664	16:40:45	712	17:18:01	53
15:27:19	3357	16:04:35	2381	16:41:49	751	17:19:04	44
15:28:23	3573	16:05:39	2132	16:42:52	818	17:20:08	34
15:29:27	3711	16:06:42	1989	16:43:56	868	17:21:11	23
15:30:31	3740	16:07:46	1863	16:45:00	871	17:22:16	18
15:31:35	3455	16:08:49	1849	16:46:05	815	17:23:19	18
15:32:38	3094	16:09:54	2120	16:47:09	687	17:24:23	22
15:33:42	2955	16:10:57	2414	16:48:14	502	17:25:26	23
15:34:46	2934	16:12:01	2642	16:49:17	373	17:26:30	21
15:35:51	2955	16:13:04	2703	16:50:22	320	17:27:33	18
15:36:54	3007	16:14:08	2619	16:51:25	313	17:28:38	16
15:37:58	3078	16:15:11	2398	16:52:29	323	17:29:42	13
15:39:01	3143	16:16:16	2133	16:53:33	329	17:30:46	11

5. Spesifikasi Alat



Keterangan

1. Motor DC : 280 RPM

Komponen	Spesifikasi
Power Supply	6V DC (disarankan dari adaptor atau baterai)
Input	Arus DC dari L298N untuk memutar motor ke kanan atau kiri
Proses	Konversi energi listrik menjadi energi mekanik (putaran)
Output	Gerakan rotasi (untuk membuka/menutup shading net)

Komponen	Spesifikasi
Arus	DC

2. Driver Motor L298N

Komponen	Spesifikasi
Power Supply	5V – 12V DC untuk motor, 5V untuk logic (via jumper VCC atau eksternal)
Input	Sinyal logika digital dari ESP32 (IN1, IN2, ENA)
Proses	Mengatur arah dan kecepatan motor berdasarkan input logika dari ESP32
Output	Arus listrik ke motor DC (OUT1, OUT2) sesuai perintah arah dan kecepatan
Arus	DC

3. Mikrokontroler ESP32

Komponen	Spesifikasi
Power Supply	3.3V – 5V DC (melalui port USB atau pin VIN)
Input	Sinyal digital dari sensor BH1750 melalui protokol I2C
Proses	Pemrosesan data dari sensor, logika kontrol motor, pengiriman data ke Web Dashboard via WiFi
Output	Sinyal digital ke driver motor L298N dan ke Web Dashboard (via MQTT dan HTTP)
Arus	DC

4. Kabel Input Daya ESP32 3,3V

5. Kabel Input Daya Motor DC 6V

6. Sensor BH1750

Komponen	Spesifikasi
Power Supply	3.3V – 5V DC
Input	Intensitas cahaya dari lingkungan (lux)
Proses	Mengubah intensitas cahaya menjadi data digital menggunakan photodiode dan ADC internal
Output	Data digital intensitas cahaya dalam satuan lux melalui protokol I2C ke ESP32
Arus	DC