LAMPIRAN

a. Kriteria Kelulusan

PEMBERITAHUAN SEBELUM UJIAN :
Jika pengumpulan dokumen Tugas Akhir di perpustakaan melewati batas akhir semester berjalan, maka mahasiswa harus menyelesaikan registrasi dan KRS semester berikutnya.

KRITERIA KELULUSAN UJIAN TUGAS AKHIR

1. Lulus dengan memperhatikan catatan ujian tugas akhir, dan atau melakukan perbaikan atau penyempurnaan naskah dan atau produk dalam waktu maksimum dua bulan dari tanggal ujian tugas akhir, yaitu
mulai Senin, 28 Juli 2025 sampai dengan Minggu, 28 September 2025
Jika dalam waktu yang ditentukan mahasiswa tersebut tidak dapat menyelesaikan, maka mahasiswa yang bersangkutan dianggap tidak lulus ujian.
2. Tidak lulus, disarankan oleh Ketua Tim Penguji untuk mempelajari ulang materi, merombak produk/naskah, atau mengganti judul.

Ketentuan bagi peserta yang tidak lulus ujian tugas akhir.
1) Mahasiswa wajib menempuh ujian tugas akhir ulang
2) Kesempatan ujian tugas akhir ulang hanya diberikan dalam rentang waktu maksimum 6 bulan, setelah ujian sidang/pendadaran
3) Jika sampai batas waktu maksimum 6 bulan tersebut belum dapat diajukan/diselesaikan, maka calon peserta ujian dinyatakan sebagai mahasiswa peserta Tugas Akhir baru, dengan segala ketentuan yang berlaku bagi peserta baru
4) Mahasiswa yang akan menempuh ujian tugas akhir ulang ini diwajibkan membayar biaya ujian sesuai tarif yang ditetapkan.

b. Catatan Pendadaran

Hari, tanggal	:	Senin, 28 Juli 2025	
Waktu	:	08.00	
Nama	:	SEPTUAJI MALIK SIDIK	
No. Mahasiswa / Prodi	:	225411074 / Informatika	

No	Hal yang harus diperbaiki	Pemberi Catatan
1.	Tambahkan daftar pustaka. Tulisan Internet of Things (IoT) cukup sekali saja. Yang lainnya ditulis IoT saja.	Pak Wagito
2.	Isi Tabel 1 spasi, indentasi paragraf diperhatikan.	Pak Pius

c. Hasil Ujian

KEPUTUSAN HASIL UJIAN PENDADARAN

Sesuai dengan hasil sidang pendadaran pada tanggal 28 Juli 2025 maka

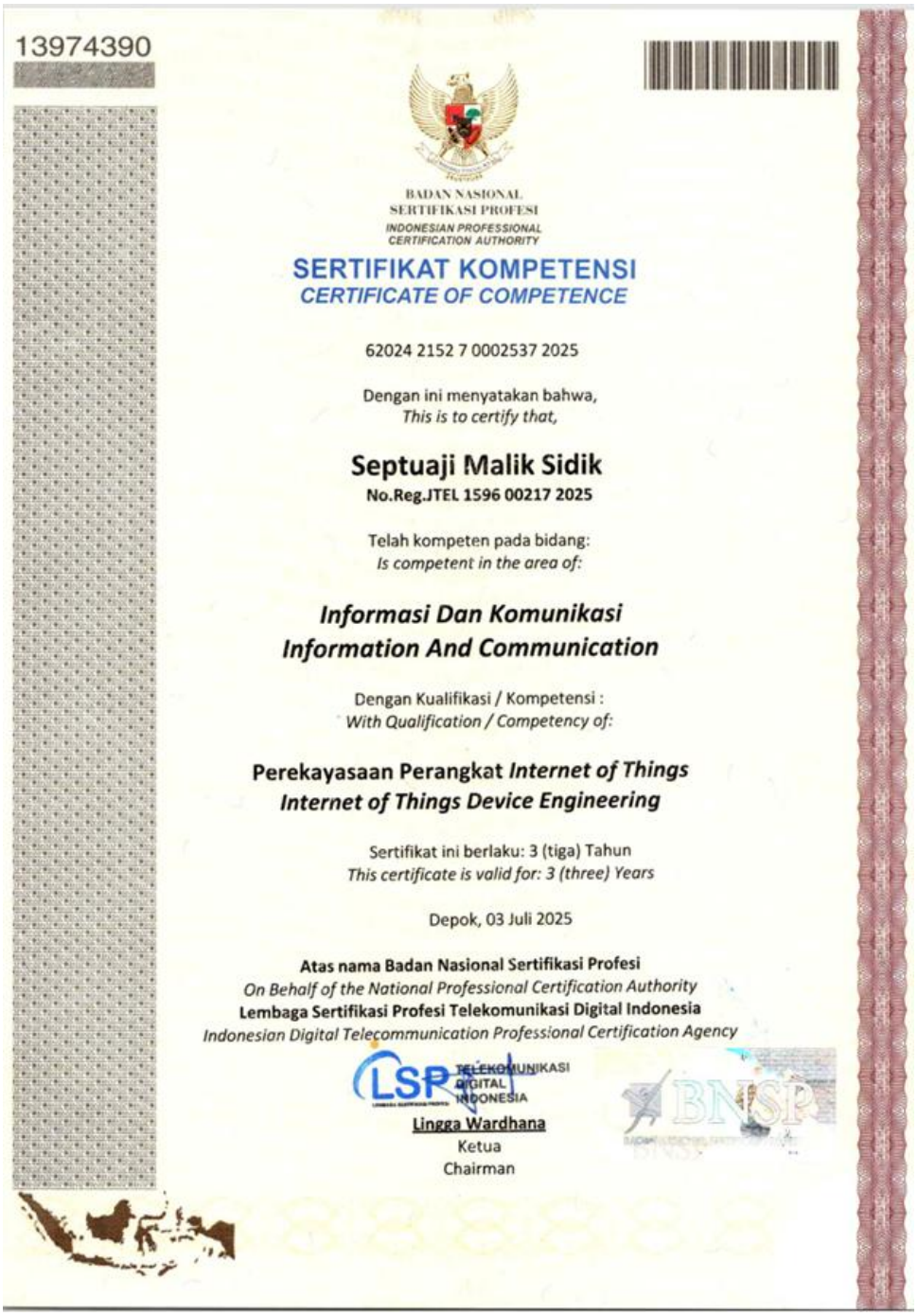
Nama Mahasiswa	SEPTUAJI MALIK SIDIK
NIM / Program Studi	225411074 / Informatika
Jenjang	S1
dinyatakan	LULUS

Ketua Penguji Wagito, S.T., M.T.

d. Sertifikat Profesional

- Sertifikat Pelatihan Berbasis Kompetensi 32 JP Pelatihan Perekayasaan Perangkat *Internet of Things*

- Sertifikat BNSP Skema Perkayasaan Perangkat *Internet of Things*



13974390

BADAN NASIONAL
SERTIFIKASI PROFESI
INDONESIAN PROFESSIONAL CERTIFICATION AUTHORITY

SERTIFIKAT KOMPETENSI
CERTIFICATE OF COMPETENCE

62024 2152 7 0002537 2025

Dengan ini menyatakan bahwa,
This is to certify that,

Septuaji Malik Sidik
No.Reg.JTEL 1596 00217 2025

Telah kompeten pada bidang:
Is competent in the area of:

Informasi Dan Komunikasi
Information And Communication

Dengan Kualifikasi / Kompetensi :
With Qualification / Competency of:

Perekayasaan Perangkat *Internet of Things*
Internet of Things Device Engineering

Sertifikat ini berlaku: 3 (tiga) Tahun
This certificate is valid for: 3 (three) Years

Depok, 03 Juli 2025

Atas nama Badan Nasional Sertifikasi Profesi
On Behalf of the National Professional Certification Authority
Lembaga Sertifikasi Profesi Telekomunikasi Digital Indonesia
Indonesian Digital Telecommunication Professional Certification Agency

LSP TELEKOMUNIKASI DIGITAL INDONESIA

Lingga Wardhana
Ketua
Chairman

BNSP

Daftar Unit Kompetensi

List of Unit(s) of Competency

NO	Kode Unit Kompetensi *Code of Competency Unit*	Judul Unit Kompetensi *Title of Competency Unit*
1.	J.61IOT01.004.1	Mengintegrasikan Perangkat Keras dan Perangkat Lunak (*Firmware*) untuk *Device* IoT *Integrating Hardware and Software (Firmware) for Iot Devices*
2.	J.61IOT01.005.1	Menguji Coba *Device* IoT *Testing the IoT Devices*
3.	J.61IOT01.008.1	Membuat Program Visual Antarmuka pada Perangkat Berbasis Web yang Terintegrasi dengan Mikrokontroler *Creating Visual Interface Programs on Web-Based Devices Integrated with Microcontroller*
4.	J.61IOT01.010.1	Menguji Coba Aplikasi IoT *Trial Testing on IoT Applications*
5.	J.61IOT01.018.1	Melakukan Instalasi Perangkat IoT (*Device*) Sesuai Desain *Installing IoT Devices (Devices) According to the Design*
6.	J.61IOT01.019.1	Melakukan Instalasi *Firmware* pada Perangkat (*Device*) Secara *Over The Air* (OTA) *Installing Firmware on the Device (Device) Over The Air (OTA)*
7.	J.61IOT01.026.1	Menerapkan Perimeter Keamanan pada Perangkat IoT *Implementing Security Perimeter on IoT Devices*
8.	J.61IOT01.041.1	Mengaplikasikan *Patch* Keamanan pada Perangkat IoT *Applying Security Patches on IoT Devices*

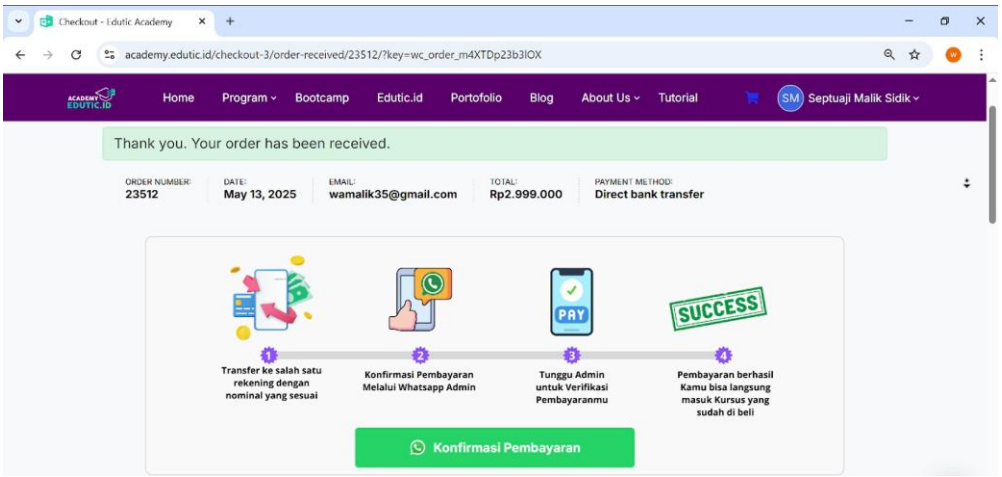
Depok, 03 Juli 2025

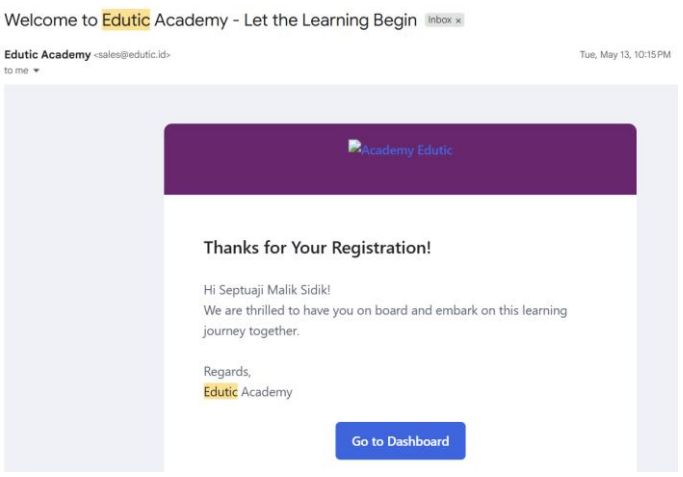
Lembaga Sertifikasi Profesi
Telekomunikasi Digital Indonesia
Indonesian Digital Telecommunication Professional Certification Agency

Septuaji Malik Sidik
Tanda tangan pemilik
Signature of holder

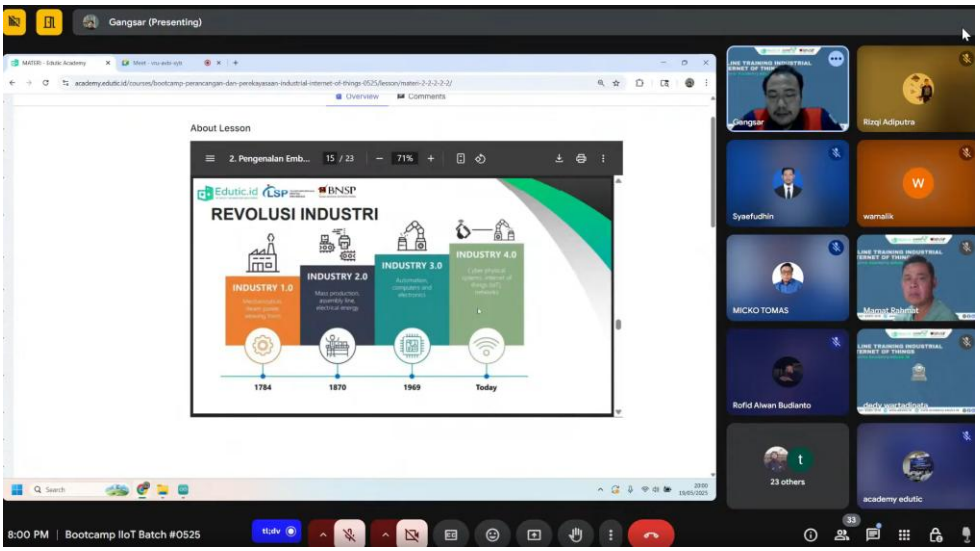
Muhammad Sabarudin
Kepala Bagian Sertifikasi
Head of Certification

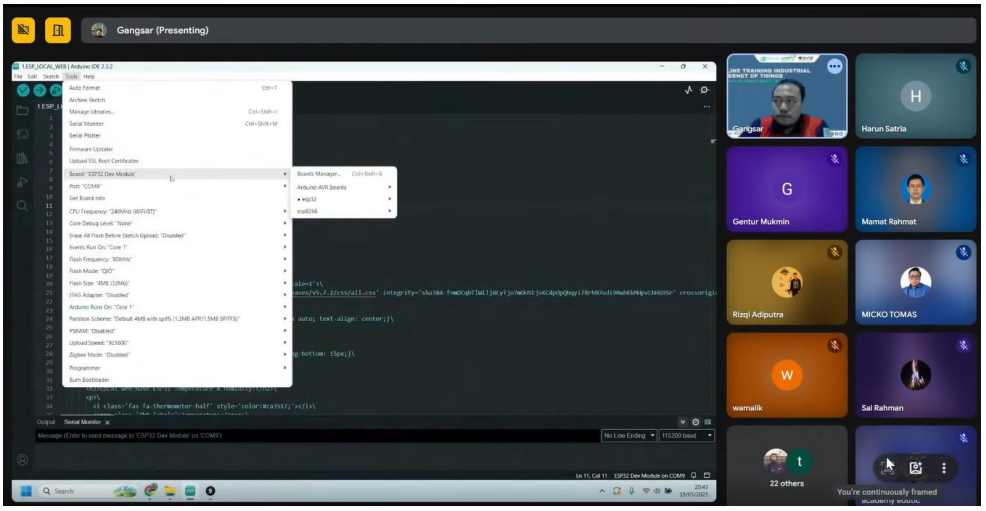
e. Bukti Pendaftaran

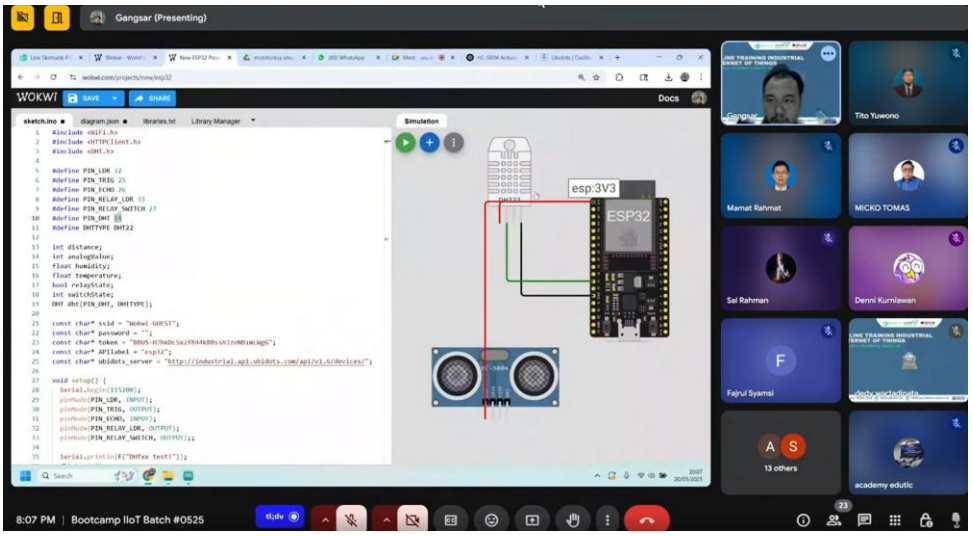


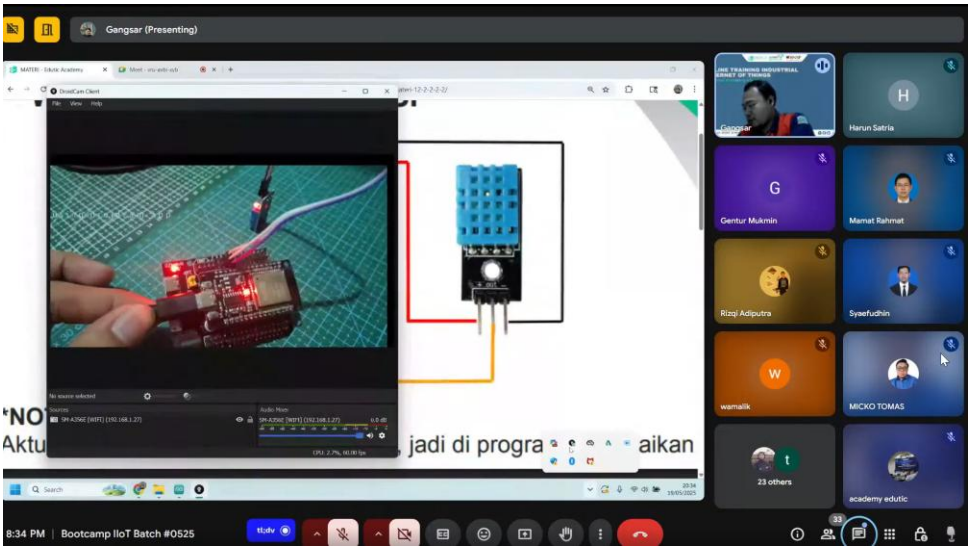


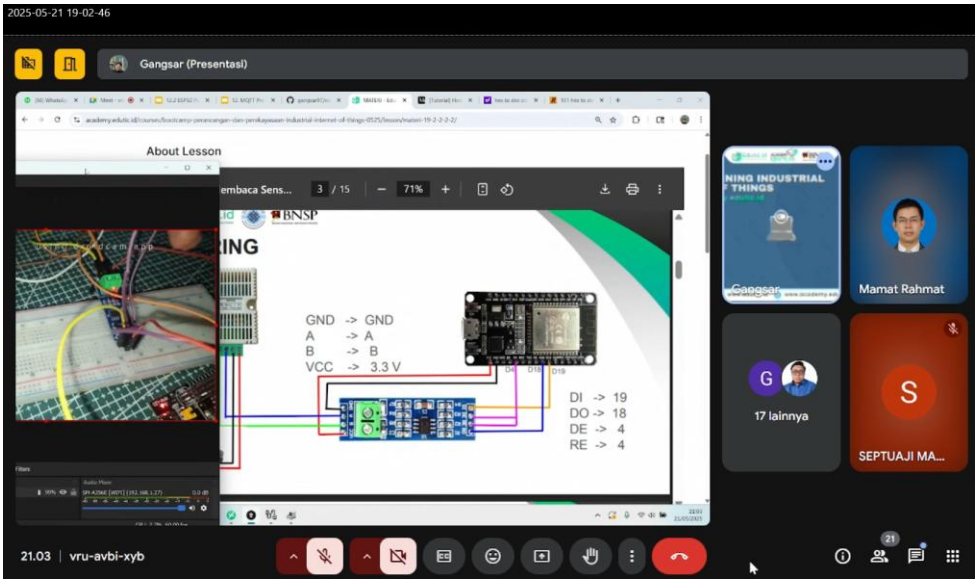
f. Foto-Foto Selama Proses Pelatihan Berbasis Kompetensi

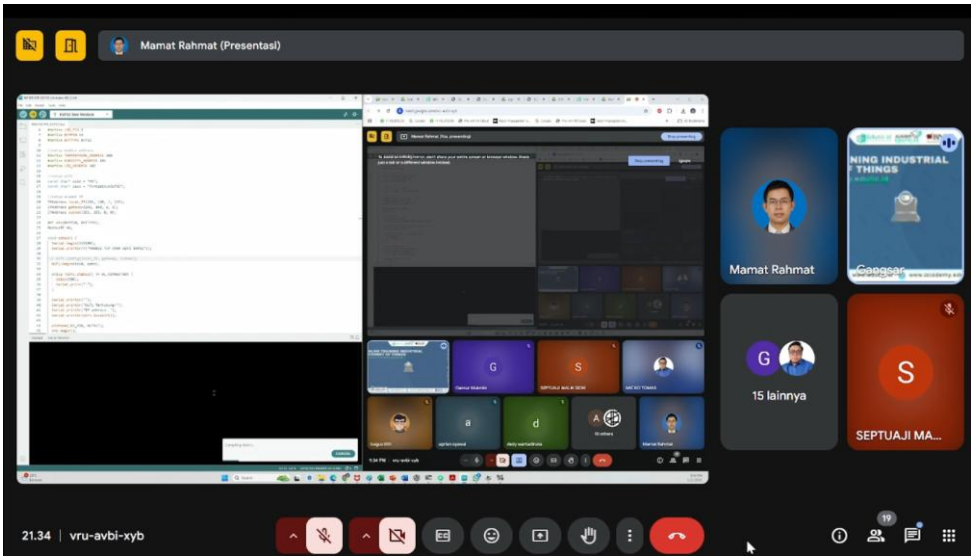



Gangsar (Presenting)
Gangsar
Harun Satria
Gentur Mukmin
Mamat Rahmat
Rizqi Adiputra
MICKO TOMAS
wamalik
Sai Rahman
22 others
You're continuously framed

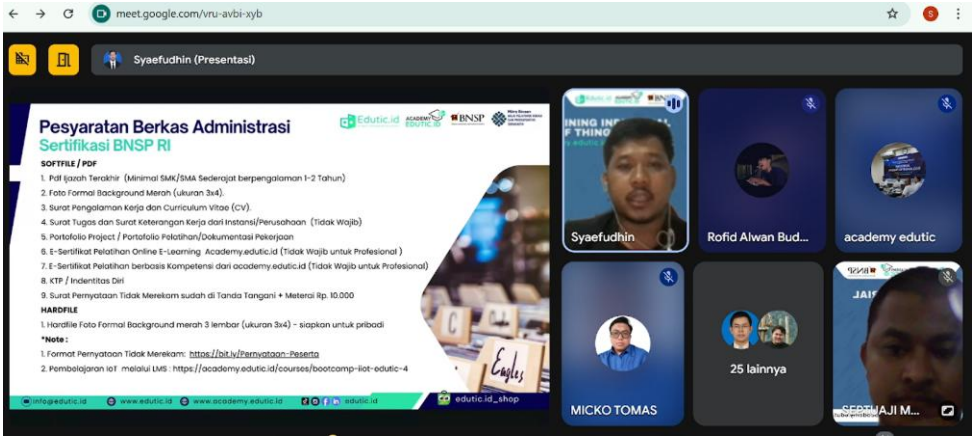

Gangsar (Presenting)
WOKWI
Docs
Simulation
esp:3V3
ESP32
Tito Yuwono
Mamat Rahmat
MICKO TOMAS
Sai Rahman
Denni Kurniawan
Fajrul Syamsi
13 others
academy edutic
8:07 PM | Bootcamp IIoT Batch #0525


Gangsar (Presenting)
Harun Satria
Gentur Mukmin
Mamat Rahmat
Rizqi Adiputra
Sysefudhin
wamalik
MICKO TOMAS
23 others
academy edutic
8:34 PM | Bootcamp IIoT Batch #0525

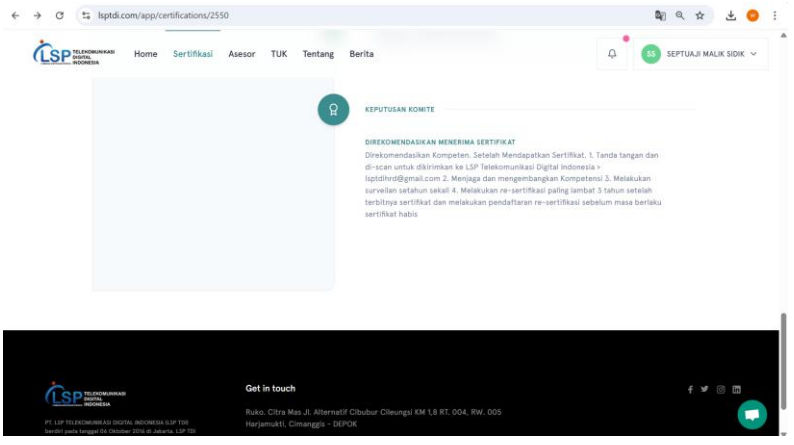




g. Foto - Foto Selama Berproses Memperoleh Sertifikasi




EDUTIC.ID

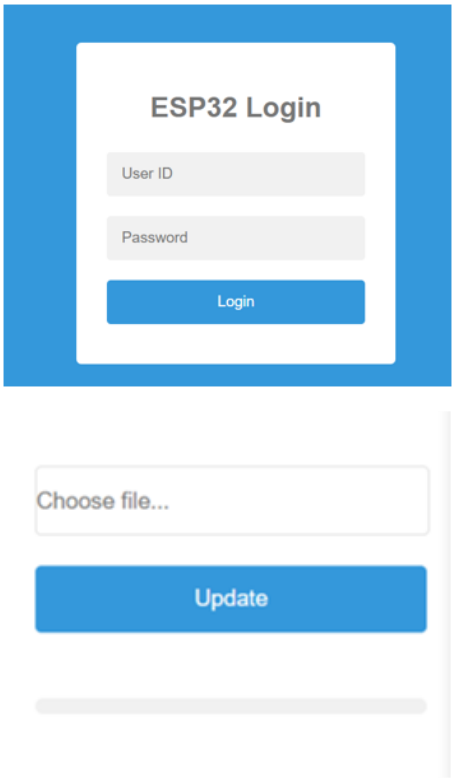

lsptdi.com/app/certifications/2550
Home
Sertifikasi
Asesor
TUK
Tentang
Berita
SEPTUAJI MALIK SIDIK
KEPUTUSAN KOMITE
DIREKOMENDASIKAN MENERIMA SERTIFIKAT
Direkomendasikan Kompeten. Setelah Mendapatkan Sertifikat, 1. Tanda tangan dan di-scan untuk dikirimkan ke LSP Telekomunikasi Digital Indonesia > lsptdihrd@gmail.com 2. Menjaga dan mengembangkan Kompetensi 3. Melakukan surveilan setahun sekali 4. Melakukan re-sertifikasi paling lambat 3 tahun setelah terbitnya sertifikat dan melakukan pendaftaran re-sertifikasi sebelum masa berlaku sertifikat habis
Get in touch
Ruko. Citra Mas Jl. Alternatif Cibubur Cileungsi KM 1,8 RT. 004, RW. 005
Harjamukti, Cimanggis - DEPOK

h. Rekam jejak proyek yang untuk sertifikasi

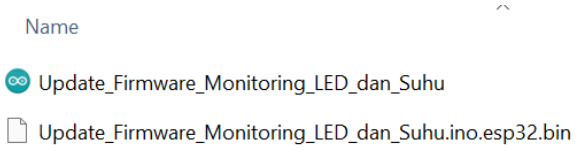
Seluruh persyaratan portofolio wajib dikumpulkan dalam google drive di link berikut :

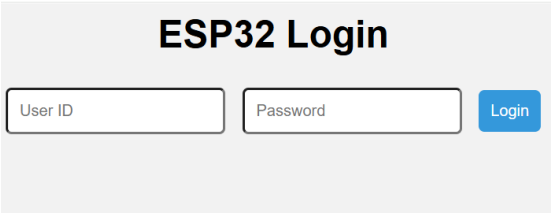
https://drive.google.com/drive/u/0/folders/1HJCiJJgtX71tEtx9dN3On3s8IOjsNkFZ

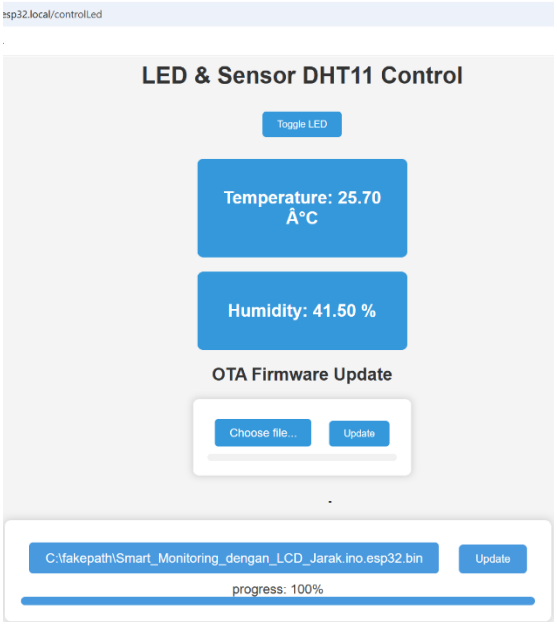
1. Firmware Over-the-Air (FOTA/OTA) Programming dan Smart Monitoring
   - Halaman Log in dan Konfigurasi Update firmware



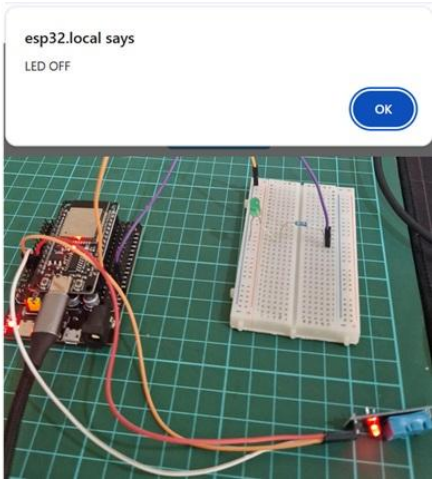
   - Update Firmware secara FOTA untuk monitoring suhu dan kelembapan dan kontrol lampu esp32



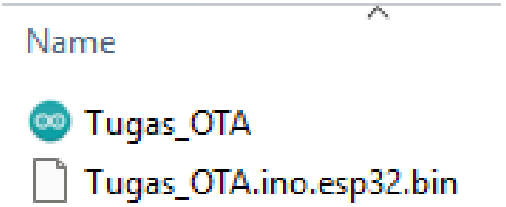

ESP32 Login
User ID
Password
Login

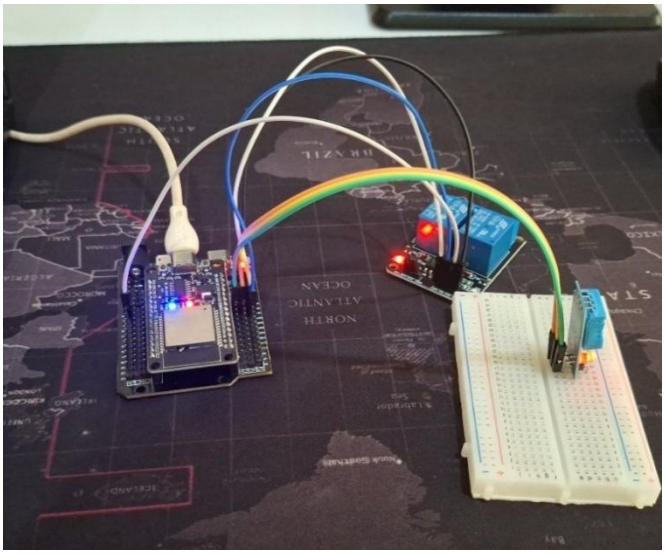

esp32.local/controlLed
LED & Sensor DHT11 Control
Toggle LED
Temperature: 25.70 Â°C
Humidity: 41.50 %
OTA Firmware Update
Choose file...
Update
C:\fakepath\Smart_Monitoring_dengan_LCD_Jarak.ino.esp32.bin
Update
progress: 100%


esp32.local says
LED ON
OK

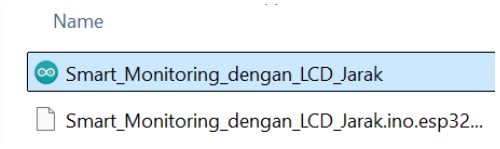

esp32.local says
LED OFF
OK

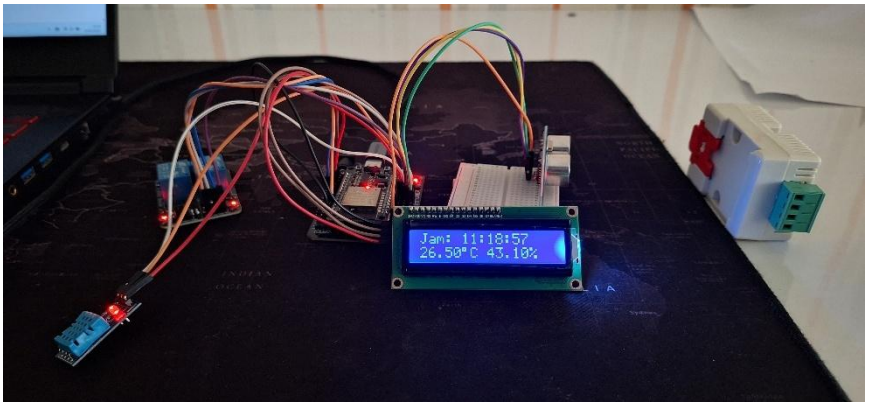
- Update Firmware Secara Fota Untuk Monitoring Suhu Dan Kelembapan, Kontrol Lampu Esp32 Dan Data Logger.





- Update Firmware Secara Fota Untuk Monitoring Suhu, Kelembapan, Jarak, Kontrol Lampu Esp32, Data Logger Dan Menampilkan Ke LCD.





	A	B	C	D
1	Timestamp,Temperature,Humidity, Distance			
2	11:37:28,26.70,41.00,10.24			
3	11:37:32,26.70,40.90,10.24			
4	11:37:38,26.70,40.80,10.24			
5	11:37:43,26.70,40.70,10.24			
6	11:37:48,26.70,40.60,10.24			
7	11:37:58,26.80,40.40,10.24			
8	11:37:58,26.80,40.40,10.24			
9	11:38:03,26.80,40.40,10.24			

- Kode Program Halaman Log In Dan Konfigurasi Update Firmware

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <Update.h>

const char* ssid = " Galaxy A54 5G EEB7";
const char* password = " malikmalik ";

WebServer server(80);
```

```
/* Login Page */
String login_page = R"rawliteral(
  <!DOCTYPE html>
  <html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
     scale=1.0">
    <title>ESP32 Login</title>
    <style>
      body {
        background: #3498db;
        font-family: sans-serif;
        font-size: 14px;
        color: #777;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
      }
      form {
        background: #fff;
        width: 100%;
        max-width: 300px;
        padding: 30px 20px; /* Added left/right padding to card */
        border-radius: 5px;
        box-shadow: 0px 0px 15px rgba(0,0,0,0.2);
        text-align: center;
      }
      h1 {
```

```
    margin-bottom: 20px;
   }
   input {
    width: 100%;
    height: 44px;
    border-radius: 4px;
    margin: 10px 0;
    font-size: 15px;
    background: #f1f1f1;
    border: none;
    box-sizing: border-box; /* Ensures padding stays inside width
  */
    padding: 0 10px; /* Optional: internal spacing inside input text
  */
   }
   .btn {
    background: #3498db;
    color: #fff;
    border: none;
    cursor: pointer;
   }
  </style>
</head>
<body>
  <form name="loginForm" onsubmit="return check(this)">
   <h1>ESP32 Login</h1>
   <input name="userid" placeholder="User ID">
   <input name="pwd" placeholder="Password"
  type="password">
   <input type="submit" class="btn" value="Login">
  </form>
  <script>
```

```
    function check(form) {
      if (form.userid.value === 'admin' && form.pwd.value ===
   'admin') {
        window.location.href = '/serverIndex';
        return false;
      } else {
        alert('Error: Invalid Username or Password');
        return false;
      }
    }
  </script>
 </body>
 </html>
)rawliteral";

/* Main OTA Update Page */
String main_page = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-
   scale=1.0">
 <title>ESP32 Update</title>
 <style>
  body {
    background: #3498db;
    font-family: sans-serif;
    font-size: 14px;
    color: #777;
    display: flex;
```

```
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}
form {
  background: #fff;
  max-width: 300px;
  width: 100%;
  padding: 30px;
  border-radius: 5px;
  text-align: center;
  box-shadow: 0px 0px 15px rgba(0,0,0,0.2);
}
#file-input, input {
  width: 100%;
  height: 44px;
  border-radius: 4px;
  margin: 10px auto;
  font-size: 15px;
}
input {
  background: #f1f1f1;
  border: 0;
  padding: 0 15px;
}
#file-input {
  padding: 0;
  border: 1px solid #ddd;
  line-height: 44px;
  text-align: left;
```

```
    display: block;
    cursor: pointer;
   }
   #bar, #prgbar {
    background-color: #f1f1f1;
    border-radius: 10px;
   }
   #bar {
    background-color: #3498db;
    width: 0%;
    height: 10px;
   }
   .btn {
    background: #3498db;
    color: #fff;
    cursor: pointer;
   }
  </style>
</head>
<body>
  <form method="POST" action="#" enctype="multipart/form-data"
    id="upload_form">
   <input type="file" name="update" id="file"
    onchange="sub(this)" style="display:none">
   <label id="file-input" for="file">   Choose file...</label>
   <input type="submit" class="btn" value="Update">
   <br><br>
   <div id="prg"></div>
   <br>
   <div id="prgbar"><div id="bar"></div></div>
  </form>
```

```
<script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.mi
  n.js"></script>
<script>
  function sub(obj){
    var fileName = obj.value.split('\\\\');
    document.getElementById('file-input').innerHTML = '   ' +
  fileName[fileName.length - 1];
  }

  $('form').submit(function(e){
    e.preventDefault();
    var form = $('#upload_form')[0];
    var data = new FormData(form);

    $.ajax({
      url: '/update',
      type: 'POST',
      data: data,
      contentType: false,
      processData: false,
      xhr: function() {
        var xhr = new window.XMLHttpRequest();
        xhr.upload.addEventListener('progress', function(evt) {
          if (evt.lengthComputable) {
            var per = evt.loaded / evt.total;
            $('#prg').html('progress: ' + Math.round(per * 100) + '%');
            $('#bar').css('width', Math.round(per * 100) + '%');
          }
        }, false);
        return xhr;
```

```
      },
      success: function(d, s) {
        console.log('success!');
      },
      error: function (a, b, c) {
        console.log('error');
      }
    });
  });
 </script>
</body>
</html>
)rawliteral";


/* Setup function */
void setup(void) {
 Serial.begin(115200);

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

 Serial.println("");
 Serial.print("Connected to ");
 Serial.println(ssid);
 Serial.print("IP address: ");
 Serial.println(WiFi.localIP());
```

```
// memulai DNS server , untuk memberi domain pada esp32
if (!MDNS.begin("esp32")) {   // alamat web ---> http://esp32.local
  Serial.println("Error setting up MDNS responder!");
  while (1) {
    delay(1000);
  }
}

Serial.println("mDNS responder started");

//handle login page
server.on("/", HTTP_GET, []() {
  server.sendHeader("Connection", "close");
  server.send(200, "text/html", login_page);
});

//handle main page
server.on("/serverIndex", HTTP_GET, []() {
  server.sendHeader("Connection", "close");
  server.send(200, "text/html", main_page);
});

//handle untuk upload program
server.on("/update", HTTP_POST, []() {
  server.sendHeader("Connection", "close");
  server.send(200, "text/plain", (Update.hasError()) ? "FAIL" : "OK");
  ESP.restart();
}, []() {
```

```
    HTTPUpload& upload = server.upload();
    if (upload.status == UPLOAD_FILE_START) {
      Serial.printf("Update: %s\n", upload.filename.c_str());
      if (!Update.begin(UPDATE_SIZE_UNKNOWN)) {
        Update.printError(Serial);
      }
    } else if (upload.status == UPLOAD_FILE_WRITE) {
      if (Update.write(upload.buf, upload.currentSize) !=
    upload.currentSize) {
        Update.printError(Serial);
      }
    } else if (upload.status == UPLOAD_FILE_END) {
      if (Update.end(true)) {
        Serial.printf("Update Success: %u\nRebooting...\n",
    upload.totalSize);
      } else {
        Update.printError(Serial);
      }
    }
  });

  //memulai server
  server.begin();
  Serial.println("HTTP server started");
 }
 void loop(void) {
  server.handleClient();
  delay(2);
}
```

- Kode Program Update Firmware secara FOTA untuk monitoring suhu, kelembapan, jarak, kontrol lampu esp32, data logger dan menampilkan ke LCD

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <Update.h>
#include <DHT.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>


// Konfigurasi WiFi
const char *ssid = "AndroidAP_7159";
const char *password = "ajiajiaji";


// Konfigurasi LED
const int ledPin = 2; // Pin LED ESP32
const int Relay1 = 27; // Pin LED ESP32


int ledState = LOW;   // Status LED
```

```
// Konfigurasi sensor DHT11
#define DHTPIN 14
#define DHTTYPE DHT11
#define I2C_ADDR 0x27
#define LCD_COLUMNS 16
#define LCD_LINES 2
#define TRIG_PIN 25
#define ECHO_PIN 26

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS,
    LCD_LINES);
float temperature = 0.0;
float humidity = 0.0;
float distance = 0.0;

// Konfigurasi server
WebServer server(80);

// Konfigurasi NTP Client
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "id.pool.ntp.org", 25200,
    60000); // UTC+7 ( Waktu Indonesia Barat )
```

```
// Data logger
String dataLog =
    "Timestamp,Temperature,Humidity,Distance\n";


unsigned long lastUpdate = 0;
const unsigned long updateInterval = 5000; // update setiap 3
    detik



void UpdateLCDStatus(float t, float h, bool lampuNyala) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Lampu ");
  lcd.print(lampuNyala ? "ON " : "OFF");


  delay(2000); // tampilkan status ON/OFF selama 0.5 detik
  String waktu = timeClient.getFormattedTime();
  updateSensorDisplay(t, h, waktu); // lanjut tampilkan suhu
    dan kelembapan
}


void updateSensorDisplay(float t, float h, String waktu) {
  lcd.clear(); // Bersihkan LCD
```

```
  lcd.setCursor(0, 0); // Baris 0
  lcd.print("Jam: ");
  lcd.print(waktu); // Format: HH:MM:SS

  lcd.setCursor(0, 1); // Baris 1
  lcd.print(t);
  lcd.print((char)223); // Derajat
  lcd.print("C ");
  lcd.print(h);
  lcd.print("%");
}

void readDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  float duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration / 58;
}
```

```
// Halaman login
String loginPage =
  R"rawliteral(
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>ESP32 Login</title>
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
    alpha3/dist/css/bootstrap.min.css">
  <style>body {background-color: #f8f9fa; text-align: center;
    font-family: Arial;}</style>
</head>
<body>
  <div class="container">
    <h1 class="text-primary mt-4">ESP32 Login</h1>
    <form onsubmit="checkLogin(event)">
      <input id="userid" class="form-control my-2"
    placeholder="Username" required>
      <input id="pwd" type="password" class="form-control my-
    2" placeholder="Password" required>
```

```
      <button class="btn btn-primary w-100">Login</button>
    </form>
  </div>
  <script>
    function checkLogin(event) {
      event.preventDefault();
      const user = document.getElementById('userid').value;
      const pass = document.getElementById('pwd').value;
      if (user === 'admin' && pass === 'admin') {
        window.location.href = '/control';
      } else {
        alert('Invalid username or password!');
      }
    }
  </script>
</body>
</html>
)rawliteral";

// Halaman kontrol LED dan sensor
String controlPage() {
  String html = R"rawliteral(
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>ESP32 Control</title>
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
    alpha3/dist/css/bootstrap.min.css">
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/6.0.0/css/all.min.css">
  <style>
    body {
      background: #f4f6f9;
      font-family: 'Arial', sans-serif;
    }
    .card {
      border-radius: 15px;
    }
    #alertBox {
      display: none;
      position: fixed;
      top: 20px;
```

```
      right: 20px;
      z-index: 9999;
    }
    #log {
      margin-top: 20px;
    }
    table {
      width: 100%;
      margin-top: 20px;
      border-collapse: collapse;
    }
    th, td {
      border: 1px solid #ddd;
      text-align: center;
      padding: 8px;
    }
    th {
      background-color: #f2f2f2;
    }
  </style>
</head>
<body>
  <div class="container py-5">
```

```
<h1 class="text-center text-primary mb-4"><i class="fas fa-laptop-house"></i> ESP32 Control</h1>


<div class="text-center mb-4">
  <button class="btn btn-success btn-lg" onclick="toggleLed()"><i class="fas fa-lightbulb"></i> Toggle LED</button>
</div>


<div class="row">
  <div class="col-md-6">
    <div class="card shadow-sm text-center p-4">
      <h2><i class="fas fa-temperature-high text-danger"></i> Temperature</h2>
      <p id="temperature" class="display-5 text-primary">--°C</p>
    </div>
  </div>
  <div class="col-md-6">
    <div class="card shadow-sm text-center p-4">
      <h2><i class="fas fa-tint text-info"></i> Humidity</h2>
      <p id="humidity" class="display-5 text-primary">--%</p>
    </div>
```

```
  </div>
</div>

<div class="text-center mt-4">
  <a href="/serverIndex" class="btn btn-warning btn-lg"><i class="fas fa-upload"></i> Firmware Update</a>
  <a href="/downloadLog" class="btn btn-secondary btn-lg"><i class="fas fa-file-download"></i> Download CSV</a>
</div>

<canvas id="chart" width="400" height="200" class="mt-4"></canvas>

<table>
  <thead>
    <tr>
      <th>Timestamp</th>
      <th>Temperature (°C)</th>
      <th>Humidity (%)</th>
      <th>Distance (cm)</th>
    </tr>
  </thead>
  <tbody id="dataLogTable"></tbody>
```

```
  </table>
</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
  function toggleLed() {
    fetch('/toggleLed').then(response=>
  response.text()).then(message => {
      alert(message);
    });
  }
  function updateSensors() {
    fetch('/sensorData').then(response=>
  response.json()).then(data => {
      document.getElementById('temperature').textContent =
  data.temperature + ' °C';
      document.getElementById('humidity').textContent =
  data.humidity + ' %';
      addToChart(data.temperature, data.humidity,
  data.distance);
      updateTable(data.timestamp, data.temperature,
  data.humidity, data.distance);
    });
  }
```

```
function addToChart(temp, hum, dist) {
  const now = new Date().toLocaleTimeString();
  chart.data.labels.push(now);
  chart.data.datasets[0].data.push(temp);
  chart.data.datasets[1].data.push(hum);
  chart.data.datasets[2].data.push(dist);
  chart.update();
}

function updateTable(timestamp, temp, hum, dist) {
  const table = document.getElementById('dataLogTable');
  const row =
`<tr><td>${timestamp}</td><td>${temp}</td><td>${hum}</
td><td>${dist}</td></tr>`;
  table.innerHTML += row;
}

const ctx =
document.getElementById('chart').getContext('2d');
const chart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: [],
```

```
        datasets: [
          {
            label: 'Temperature (°C)',
            borderColor: 'red',
            data: [],
          },
          {
            label: 'Humidity (%)',
            borderColor: 'blue',
            data: [],
          },
          {
            label: 'Distance (cm)',
            borderColor: 'green',
            data: [],
          }
        ]
      }
    });

    setInterval(updateSensors, 5000);
  </script>
</body>
```

```
</html>
)rawliteral";
  return html;
}

// Halaman firmware update
String serverIndex =
R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <title>OTA Update</title>
  <style>
    body {font-family: Arial; text-align: center; margin-top:
    50px;}
    #bar {background: #3498db; height: 10px; width: 0%;
    margin-top: 20px;}
  </style>
</head>
<body>
  <h1>Update Firmware</h1>
  <form method="POST" enctype="multipart/form-data"
    id="upload_form">
    <input type="file" name="update">
```

```
    <input type="submit" value="Update">
  </form>
  <div id="bar"></div>
  <script>
    document.getElementById('upload_form').onsubmit =
    function(e) {
      e.preventDefault();
      const form = new FormData(this);
      const xhr = new XMLHttpRequest();
      xhr.open("POST", "/update");
      xhr.upload.onprogress = function(event) {
        const percent = Math.round((event.loaded / event.total) *
    100);
        document.getElementById('bar').style.width = percent +
    '%';
      };
      xhr.onload = function() {
        alert(xhr.responseText);
        if (xhr.responseText === 'OK') location.reload();
      };
      xhr.send(form);
    };
  </script>
</body>
```

```
</html>
)rawliteral";


// Setup
void setup() {
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
  pinMode(Relay1, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Serial.println(F("DHTxx test!"));
  dht.begin();
  lcd.init();
  lcd.backlight();
  WiFi.begin(ssid, password);
  WiFi.mode(WIFI_STA);
  Serial.println("");


  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

if (MDNS.begin("esp32")) {
  Serial.println("MDNS responder started");
}

timeClient.begin();

server.on("/", HTTP_GET, []() { server.send(200, "text/html",
  loginPage); });

server.on("/control", HTTP_GET, []() { server.send(200,
  "text/html", controlPage()); });

server.on("/toggleLed", HTTP_GET, []() {
  ledState = !ledState;
  digitalWrite(ledPin, ledState);
  UpdateLCDStatus(temperature, humidity, ledState);
```

```
  server.send(200, "text/plain", ledState ? "LED ON" : "LED
  OFF");
});

server.on("/sensorData", HTTP_GET, []() {
  timeClient.update();
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  readDistance();
  String timeStamp = timeClient.getFormattedTime();
  dataLog += timeStamp + "," + String(temperature) + "," +
  String(humidity) + "," + String(distance) + "\n";
  updateSensorDisplay(temperature, humidity, timeStamp);
  String json = "{\"timestamp\":\"" + timeStamp +
  "\",\"temperature\":\"" + String(temperature) +
  "\",\"humidity\":\"" + String(humidity) + "\",\"distance\":\"" +
  String(distance) + "\"}";
  server.send(200, "application/json", json);
});

server.on("/downloadLog", HTTP_GET, []() {
  server.send(200, "text/csv", dataLog);
});
```

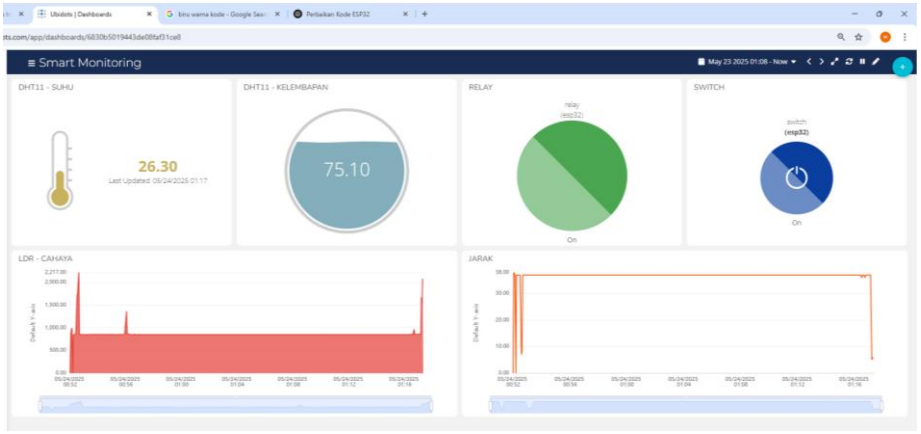
```
  server.on("/serverIndex", HTTP_GET, []() { server.send(200,
    "text/html", serverIndex); });


  server.on("/update", HTTP_POST, []() {
    server.send(200, "text/plain", Update.hasError() ? "FAIL" :
    "OK");
    ESP.restart();
  }, []() {
    HTTPUpload& upload = server.upload();
    if (upload.status == UPLOAD_FILE_START)
    Update.begin(UPDATE_SIZE_UNKNOWN);
    else if (upload.status == UPLOAD_FILE_WRITE)
    Update.write(upload.buf, upload.currentSize);
    else if (upload.status == UPLOAD_FILE_END)
    Update.end(true);
  });


  server.begin();
  Serial.println("HTTP server started");
}


// Loop
void loop() {
  server.handleClient();
```

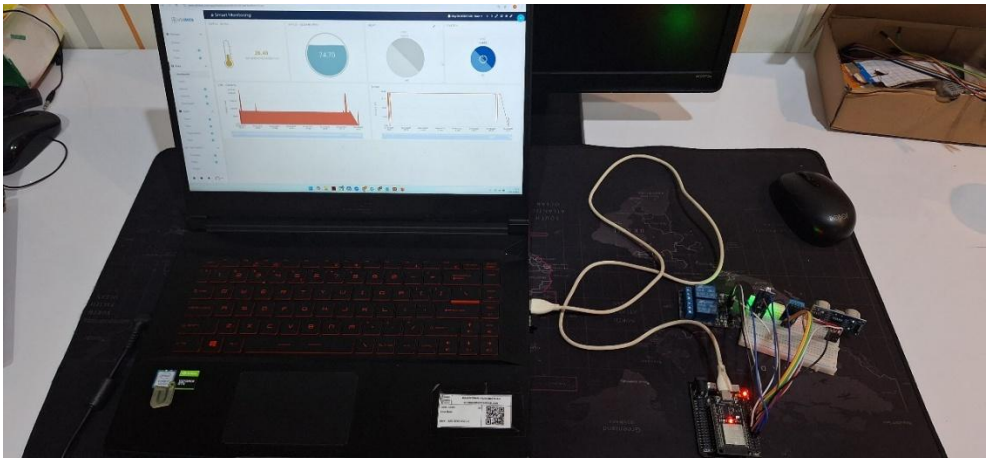
```
  // Update suhu & kelembapan setiap 3 detik (jika tidak
    sedang tombol ditekan)
  if (millis() - lastUpdate > updateInterval) {
    lastUpdate = millis();
    timeClient.update(); // Tambahkan ini agar waktu akurat
    String waktu = timeClient.getFormattedTime();
    float temp = dht.readTemperature();
    float humi = dht.readHumidity();
    temperature = temp;
    humidity = humi;
    updateSensorDisplay(temp, humi, waktu);  // tampilkan ke
    LCD
}}
```

2. Smart Monitoring Temperature, Humidity, Lumen, dan Control Relay dengan Platform IOT UBIDOTS

- Tampilan Smart Monitor pada Mobile Phone

- Tampilan Smart Monitor Pada Web





- Kode Program

```
#include <WiFi.h>

#include <HTTPClient.h>

#include <DHT.h>


#define PIN_LDR 32

#define PIN_TRIG 25

#define PIN_ECHO 26

#define PIN_RELAY_LDR 33

#define PIN_RELAY_SWITCH 27

#define PIN_DHT 14

#define DHTTYPE DHT11


unsigned long prevMillis;


int distance;
```

```
int analogValue;
float humidity;
float temperature;
bool relayState;
int switchState;
DHT dht(PIN_DHT, DHTTYPE);


const char* ssid = "Galaxy A54 5G EEB7";          // nama wifi
const char* password = "malikmalik";              // password wifi
const char* token = "BBUS-
    sm5gpWSv6lJ9UwcO7UqSu2eLQTOofW";    // token
    ubidots
const char* APIlabel = "esp32";                   // nama
    device
const char* ubidots_server =
    "http://industrial.api.ubidots.com/api/v1.6/devices/";


void setup() {
  Serial.begin(115200);
  pinMode(PIN_LDR, INPUT);
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
  pinMode(PIN_RELAY_LDR, OUTPUT);
  pinMode(PIN_RELAY_SWITCH, OUTPUT);
```

```
    Serial.println(F("DHTxx test!"));
  dht.begin();


  WiFi.begin(ssid, password);
  Serial.print("Menghubungkan ke WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("Terhubung ke WiFi");
}


void loop() {
  if (millis() - prevMillis >= 2000) {
    prevMillis = millis();


    readUltrasonic();
    readLDR();
    readDHT();
    Switch();
    sendDataToUbidots();  }}
void readUltrasonic() {
  digitalWrite(PIN_TRIG, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);

  int duration = pulseIn(PIN_ECHO, HIGH);
  distance = duration / 58.0;
  Serial.print("Jarak dalam Cm: ");
  Serial.println(distance);
  Serial.print("Jarak dalam Inch: ");
  Serial.println(duration / 148.0);
}

void readLDR() {
  analogValue = analogRead(PIN_LDR);
  relayState = false;

  Serial.print("LDR Value: ");
  Serial.print(analogValue);

  if (analogValue > 2000) {
    Serial.println(" => Dark");
    digitalWrite(PIN_RELAY_LDR, LOW);
    relayState = true;
  } else {
```

```
    Serial.println(" => Very bright");
    digitalWrite(PIN_RELAY_LDR, HIGH);
    relayState = false;
  }
  Serial.print("Relay: ");
  Serial.println(relayState ? "ON" : "OFF");
}


void readDHT() {
  humidity = dht.readHumidity();
  temperature = dht.readTemperature();
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  Serial.print(F("Humidity: "));
  Serial.print(humidity);
  Serial.print(F("%  Temperature: "));
  Serial.print(temperature);
  Serial.println("C");
}
void Switch() {
  if (WiFi.status() == WL_CONNECTED) {
```

```
HTTPClient http;


String url = String(ubidots_server) + APIlabel +
"/switch/lv?token=" + token;
http.begin(url);


int httpResponseCode = http.GET();
if (httpResponseCode > 0) {
  String payload = http.getString();
  switchState = payload.toInt();


  if (switchState == 1) {
    digitalWrite(PIN_RELAY_SWITCH, LOW);
  } else {
    digitalWrite(PIN_RELAY_SWITCH, HIGH);
  }


  Serial.print("Relay Switch dari Ubidots: ");
  Serial.println(switchState == 1 ? "ON" : "OFF");
} else {
  Serial.print("Gagal membaca data dari Ubidots: ");
  Serial.println(httpResponseCode);
}
```

```
    http.end();
  }
}

void sendDataToUbidots() {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;

    String url = String(ubidots_server) + APIlabel + "/?token="
    + token;
    http.begin(url);
    http.addHeader("Content-Type", "application/json");

    String jsonPayload = "{";
    jsonPayload += "\"jarak\":{\"value\":" + String(distance) +
    "},";
    jsonPayload += "\"suhu\":{\"value\":" + String(temperature,
    2) + "},";
    jsonPayload += "\"kelembaban\":{\"value\":" +
    String(humidity, 2) + "},";
    jsonPayload += "\"ldr\":{\"value\":" + String(analogValue) +
    "},";
```

```
    jsonPayload += "\"relay\":{\"value\":" + String(relayState ? 1
    : 0) + "},";
    jsonPayload += "\"switch\":{\"value\":" +
    String(digitalRead(PIN_RELAY_SWITCH) == LOW ? 1 : 0)
    + "}";
    jsonPayload += "}";

    int httpResponseCode = http.POST(jsonPayload);
    if (httpResponseCode > 0) {
      Serial.println("Data terkirim ke Ubidots.");
    } else {
      Serial.print("Error dalam pengiriman: ");
      Serial.println(httpResponseCode);
    }

    http.end();
  } else {
    Serial.println("WiFi tidak terhubung");
  }
}
```