

LAMPIRAN

Lampiran 1 Panduan Monitoring Kandang Puyuh

A. Tujuan Penggunaan Sistem

Silakan ubah menjadi seperti berikut agar formatnya sama dengan contoh: Sistem SMARTQUAIL dirancang untuk memonitor dan mengendalikan kondisi lingkungan kandang puyuh petelur menggunakan teknologi IoT dan website. Sistem mampu membaca suhu, kelembapan, dan kadar amonia secara *real-time*, kemudian mengendalikan kipas, exhaust, pompa *spray*, dan lampu pemanas secara otomatis sesuai kondisi kandang. Selain itu, sistem juga menampilkan data monitoring pada dashboard website sehingga peternak dapat memantau kondisi kandang dari mana saja melalui koneksi internet.

B. Menyalakan Sistem

LANGKAH-LANGKAH

Sebelum menggunakan sistem, pastikan seluruh perangkat telah terhubung dengan baik, koneksi Wi-Fi tersedia, dan sumber daya listrik dalam kondisi normal.

1. Pastikan adaptor telah terhubung ke rangkaian dan seluruh komponen terpasang dengan benar.
2. Nyalakan sistem dengan menghubungkan catu daya ke NodeMCU ESP8266.
3. Tunggu beberapa detik hingga NodeMCU melakukan proses inisialisasi dan terhubung ke jaringan Wi-Fi.

4. Setelah sistem aktif, sensor DHT11 dan MQ-135 akan mulai membaca nilai suhu, kelembapan, dan kadar amonia di dalam kandang.
5. Buka website SMARTQUAIL melalui browser, kemudian lakukan login menggunakan akun yang telah terdaftar.
6. Dashboard akan menampilkan data suhu, kelembapan, kadar amonia, serta status kipas, exhaust, pompa spray, dan lampu pemanas secara real-time.
7. Sistem akan mengendalikan perangkat secara otomatis sesuai kondisi lingkungan kandang, sedangkan pengguna dapat terus memantau kondisi kandang melalui dashboard.

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing I

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Nurohim, S.ST, M.Kom

NIDN : 0625067701

NIPY : NIPY.09.017.342

Jabatan Struktural : -

Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir Mahasiswa

Berikut:

Nama : Azwar Dava Khalid

NIM : 23040077

Program Studi : D-3 TEKNIK KOMPUTER SEKOLAH VOKASI

Judul TA : RANCANG BANGUN SISTEM IOT UNTUK
PENGENDALIAN SUHU KELEMBAPAN DAN
AMONIA PADA KANDANG PUYUH PETELUR

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 9 Maret 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,



Arif Rakhman, SE, S.Pd, M.Kom
NIPY. 05.016.291

Dosen Pembimbing I,

Nurohim, S.ST, M.Kom
NIPY.09.017.342

Lampiran 3 Surat Kesiediaan Membimbing TA Pembimbing II

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Nurohim, S.ST, M.Kom

NIDN : 0625067701

NIPY : NIPY.09.017.342

Jabatan Struktural : -

Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir Mahasiswa

Berikut:

Nama : Azwar Dava Khalid

NIM : 23040077

Program Studi : D-3 TEKNIK KOMPUTER SEKOLAH VOKASI

Judul TA : RANCANG BANGUN SISTEM IOT UNTUK
PENGENDALIAN SUHU KELEMBAPAN DAN
AMONIA PADA KANDANG PUYUH PETELUR

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 9 Maret 2026

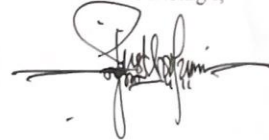
Mengetahui

Ka. Prodi DIII Teknik Komputer,



Arif Rakhman, SE, S.Pd, M.Kom
NIPY. 05.016.291

Dosen Pembimbing I,



Nurohim, S.ST, M.Kom
NIPY.09.017.342

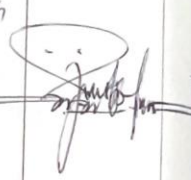
Lampiran 4 Surat Observasi

Lampiran 5 Bimbingan Laporan Pembimbing I TA

Lampiran 23
Bimbingan Laporan Pembimbing I TA

IK P2M PHB d.5.1.e.1

PEMBIMBING II: BIMBINGAN LAPORAN TA

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	25 NOV	Jarbar ketengah mazi ben um mengorok ke tengah	
2.	25/ 08 2025	Acc BAB 1, 2, 3 Lanjut Pembimbing 2	

Lampiran 6 Bimbingan Laporan Pembimbing II TA

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	25/ 05 2026	Bab 4. - Diagram Jruccur	
2.	25/ 05 2026	Bab 4 - Diagram Block - Diagram alir	
3	25/ 05 2026	Bab 5, 6 ACC Lup Ldy.	

Lampiran 7 Dokumentasi Observasi



Lampiran 8 Source Code

```
SISTEM IoT KANDANG PUYUH - ESP8266
VERSION: 5.0 FINAL PRODUCTION READY
Smart Quail Farming System
Real-time Monitoring + Instant Status Sync
Relay 4 Channel:
• IN1 = Kipas (Ventilasi)
• IN2 = Pompa Sprayer (Kelembaban)
• IN3 = Lampu Pemanas (Suhu)
Sensor:
• DHT11 (Suhu & Kelembaban)
• MQ135 (Ammonia Detection)
// || Server: smartquail.ruangprojek.cloud (HTTPS)
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
#include <DHT.h>
#include <ArduinoJson.h>

//=====
//
// SECTION 1: KONFIGURASI WIFI
//
//=====

const char* ssid      = "gratisoksu";
const char* password = "waplo123456";
```

```

//
=====

// SECTION 2: KONFIGURASI SERVER LARAVEL
//
=====

const char* serverHost = "smartquail.ruangprojek.cloud";
String urlSensorPost;
String urlSprayerStatus;
//
=====

// SECTION 3: KONFIGURASI PIN & SENSOR
//
=====

#define DHTPIN    D4
#define DHTTYPE   DHT11
DHT dht(DHTPIN, DHTTYPE);

#define MQ135_PIN    A0
#define RELAY_KIPAS  D1    // IN1 - Kipas Ventilasi
#define RELAY_POMPA  D2    // IN2 - Pompa Sprayer
#define RELAY_LAMPU  D5    // IN3 - Lampu Pemanas
//
=====

// SECTION 4: KALIBRASI KONDISI KANDANG PUYUH
(Optimized)

```

```

//
=====

float batasSuhuDingin          = 20.0;    // Lampu ON jika
suhu < 20°C

float batasSuhuPanas           = 35.0;    // Kipas ON jika
suhu > 35°C

float batasKelembapanRendah = 40.0;    // Sprayer ON jika
kelembapan < 40%

float batasKelembapanTinggi = 80.0;    // Kipas ON jika
kelembapan > 80%

float batasAmoniaPPM_Atas     = 4.0;    // Kipas/Sprayer
ON jika ammonia >= 4.0 ppm

float batasAmoniaPPM_Bawah    = 2.0;    // Stop jika
ammonia < 2.0 ppm

//
=====

// SECTION 5: DURASI TIMER AKTUATOR (millisecond)

//
=====

#define DURASI_POMPA           7000UL    // Sprayer ON
selama 7 detik

#define DURASI_JEDA_POMPA     15000UL    // Jeda 15 detik
sebelum spray ulang

#define DURASI_KIPAS          30000UL    // Kipas ON minimal
30 detik

//
=====

// SECTION 6: STATUS RELAY & TIMER

```

```

//
=====

bool  kipasAktif      = false;
bool  pompaAktif      = false;
bool  lampuAktif      = false;
unsigned long kipasStartTime = 0;
unsigned long pompaStartTime = 0;
// Status dari Web Dashboard (Manual Control)
bool  sprayerDariServer = false;
bool  kipasDariServer   = false;
bool  lampuDariServer   = false;
//
=====

// SECTION 7: INTERVAL KOMUNIKASI (millisecond)
//
=====

#define INTERVAL_KIRIM_SERVER  1000UL  // Kirim data ke
server setiap 1 detik

#define INTERVAL_CEK_TOMBOL    400UL   // Cek tombol web
setiap 0.4 detik

#define INTERVAL_BACA_MQ135     100UL   // Baca MQ135
setiap 100ms (balance)

unsigned long lastKirimTime = 0;
unsigned long lastCekTime   = 0;
unsigned long lastMQ135Time = 0;
//
=====
=====

```

```

// SECTION 8: GLOBAL VARIABLE SENSOR TERBARU
//
=====

float suhuTerbaru = 0;
float kelembapanTerbaru = 0;
float amoniaPPMTerbaru = 0;
int amoniaRAWTerbaru = 0;
//
=====

// SECTION 9: HTTP CLIENT (Keep-Alive untuk kecepatan)
//
=====

WiFiClientSecure clientGet;
HTTPClient httpGet;
bool isGetClientInit = false;
WiFiClientSecure clientPost;
HTTPClient httpPost;
bool isPostClientInit = false;
//
=====

// SECTION 10: UTILITY FUNCTIONS - Baca Sensor MQ135
//
=====

int bacaSensorStabil(int pin) {
long total = 0;

```

```

for (int i = 0; i < 5; i++) {
total += analogRead(pin);
delay(2);
}
return total / 5;
}

float konversiKePPM(int analogVal) {
float ppm = (float)analogVal / 1023.0 * 50.0;
return round(ppm * 10.0) / 10.0;
}

//
=====

// SECTION 11: UTILITY FUNCTIONS - Kirim Data Sensor ke
Server

//
=====

void kirimDataSensor(float suhu, float kelembapan, float
amoniaPPM) {
if (WiFi.status() != WL_CONNECTED) return;
if (!isPostClientInit) {
clientPost.setInsecure();
httpPost.setReuse(true);
httpPost.setTimeout(2000);
isPostClientInit = true;
}

httpPost.begin(clientPost, urlSensorPost);
httpPost.addHeader("Content-Type", "application/json");

```

```

httpPost.setUserAgent("ESP8266-IoT");
StaticJsonDocument<200> doc;
doc["suhu"] = suhu;
doc["kelembapan"] = kelembapan;
doc["amonia"] = amoniaPPM;
doc["kipas_active"] = kipasAktif;
doc["lampu_active"] = lampuAktif;
doc["sprayer_active"] = pompaAktif;
String jsonString;
serializeJson(doc, jsonString);
int httpCode = httpPost.POST(jsonString);
if (httpCode > 0) {
    if (httpCode != HTTP_CODE_OK && httpCode !=
HTTP_CODE_CREATED) {
        Serial.print("[ERROR] POST gagal! Code: ");
        Serial.println(httpCode);
    }
}

httpPost.end();
}

//
=====

// SECTION 12: UTILITY FUNCTIONS - Cek Tombol dari Web
//
=====
=====

```

```

void cekStatusAktuator() {
    if (WiFi.status() != WL_CONNECTED) return;
    if (!isGetClientInit) {
        clientGet.setInsecure();
        httpGet.setReuse(true);
        httpGet.setTimeout(3000);
        isGetClientInit = true;
    }
    httpGet.begin(clientGet, urlSprayerStatus);
    httpGet.setUserAgent("ESP8266-IoT");
    int httpCode = httpGet.GET();
    if (httpCode > 0 && (httpCode == HTTP_CODE_OK || httpCode
    == HTTP_CODE_CREATED)) {
        String response = httpGet.getString();
        StaticJsonDocument<200> doc;
        DeserializationError error = deserializeJson(doc,
        response);

        if (!error) {
            sprayerDariServer = doc["sprayer_active"].as<bool>();
            kipasDariServer = doc["kipas_active"].as<bool>();
            lampuDariServer = doc["lampu_active"].as<bool>();
        }
    }
    httpGet.end();
}

```

```

//
=====

// SECTION 13: SETUP - Inisialisasi Sistem
//
=====

void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.println("\n\n");
  Serial.println("=====");
  Serial.println("||");
  Serial.println("||");
  Serial.println("||      SISTEM IoT KANDANG PUYUH - v5.0");
  Serial.println("FINAL PRODUCTION ||");
  Serial.println("||");
  Serial.println("||      Smart Real-time Monitoring + Instant");
  Serial.println("Status Sync ||");
  Serial.println("||");
  Serial.println("=====");
  Serial.println("===== \n");

  dht.begin();

  pinMode(RELAY_KIPAS, OUTPUT);
  pinMode(RELAY_POMPA, OUTPUT);
  pinMode(RELAY_LAMPU, OUTPUT);
  digitalWrite(RELAY_KIPAS, HIGH);
  digitalWrite(RELAY_POMPA, HIGH);

```

```

digitalWrite(RELAY_LAMPU, HIGH);

urlSensorPost      = "https://" + String(serverHost) +
"/api/esp/sensor";

urlSprayerStatus   = "https://" + String(serverHost) +
"/api/esp/sprayer-status";

WiFi.begin(ssid, password);

Serial.print("[INIT] Menghubungkan WiFi");

int attempt = 0;

while (WiFi.status() != WL_CONNECTED && attempt < 20) {

delay(500);

Serial.print(".");

attempt++;

}

if (WiFi.status() == WL_CONNECTED) {

Serial.println("\n[SUCCESS] WiFi Terhubung!");

Serial.print("[INFO] IP Address: ");

Serial.println(WiFi.localIP());

} else {

Serial.println("\n[ERROR] WiFi Gagal!");

}

Serial.println("[INFO] Inisialisasi sensor...");

Serial.println("[INFO] MQ135 warming up (optimal di 2-3
menit)...\n");

Serial.println("=====
=====\\n");

}

//
=====
=====

```

```

// SECTION 14: MAIN LOOP

//
=====

void loop() {
  unsigned long sekarang = millis();
  //
  -----

  // TASK 1: Cek Tombol Web (400ms) - SUPER RESPONSIF
  //
  -----

  if (sekarang - lastCekTime >= INTERVAL_CEK_TOMBOL) {
    lastCekTime = sekarang;
    cekStatusAktuator();
  }
  //
  -----

  // TASK 2: Baca Sensor DHT11 Setiap Loop (INSTANT)
  //
  -----

  suhuTerbaru = dht.readTemperature();
  kelembapanTerbaru = dht.readHumidity();
  //
  -----

  // TASK 3: Baca Sensor MQ135 Setiap 100ms (BALANCE: cepat
+ stabil)

```

```

//


---




---


if (sekarang - lastMQ135Time >= INTERVAL_BACA_MQ135) {
lastMQ135Time = sekarang;
amoniaRAWTerbaru = bacaSensorStabil(MQ135_PIN);
amoniaPPMTerbaru = konversiKePPM(amoniaRAWTerbaru);
}
//


---




---


// TASK 4: Kirim Data ke Server (1 detik) - BATCHING
//


---




---


if (sekarang - lastKirimTime >= INTERVAL_KIRIM_SERVER)
{
lastKirimTime = sekarang;
if (!isnan(suhuTerbaru) && !isnan(kelembapanTerbaru)) {
kirimDataSensor(suhuTerbaru,          kelembapanTerbaru,
amoniaPPMTerbaru);
// Print formatted output
Serial.println("=====          DATA          SENSOR
=====");
Serial.print("|| Suhu          : ");
Serial.print(suhuTerbaru);
Serial.println(" °C          ||");
Serial.print("|| Kelembapan: ");
Serial.print(kelembapanTerbaru);
Serial.println(" %          ||");
}
}

```



```

//


---




---



// RELAY 1: KIPAS - Control Ventilasi
// Trigger: Suhu > 35°C OR Kelembapan > 80% OR Ammonia
// >= 4.0 ppm
//


---




---



if (kipasDariServer) {
  if (!kipasAktif) {
    kipasAktif = true;
    digitalWrite(RELAY_KIPAS, LOW);

    Serial.println("[✓ KIPAS] ✓ ON (Manual dari Web)");
  }

  kipasStartTime = sekarang - DURASI_KIPAS;
} else {
  bool kondisiKipasNyala = (suhuTerbaru > batasSuhuPanas)
  ||
  (kelembapanTerbaru > batasKelembapanTinggi) ||
  (amoniaPPMterbaru >= batasAmoniaPPM_Atas);

  bool kondisiKipasAman = (suhuTerbaru <= batasSuhuPanas)
  &&
  (kelembapanTerbaru <= batasKelembapanTinggi) &&
  (amoniaPPMterbaru < batasAmoniaPPM_Bawah);

  if (kondisiKipasNyala && !kipasAktif) {
    kipasAktif = true;
    kipasStartTime = sekarang;
    digitalWrite(RELAY_KIPAS, LOW);
  }
}

```

```

Serial.println("[⚡ KIPAS] ✓ ON (Otomatis - Kondisi tidak
normal)");

} else if (kipasAktif) {

if ((sekarang - kipasStartTime >= DURASI_KIPAS) &&
kondisiKipasAman) {

kipasAktif = false;

digitalWrite(RELAY_KIPAS, HIGH);

Serial.println("[⚡ KIPAS] ✗ OFF (Kondisi normal)");

}

}

}

//


---


// RELAY 2: SPRAYER - Control Kelembapan & Ammonia
// Trigger: Ammonia >= 4.0 ppm OR Kelembapan < 40%
//


---



if (sprayerDariServer) {
if (!pompaAktif) {
pompaAktif = true;
digitalWrite(RELAY_POMPA, LOW);

Serial.println("[SPRAYER] ✓ ON (Manual dari Web)");

}

pompaStartTime = sekarang - DURASI_JEDA_POMPA;

} else {

bool    kondisiPompaNyala    =    (amoniaPPMTerbaru    >=
batasAmoniaPPM_Atas)    ||    (kelembapanTerbaru    <

```

```

batasKelembapanRendah);bool      kondisiPompaAman      =
(amoniaPPMterbaru      <      batasAmoniaPPM_Bawah)&&
(kelembapanTerbaru >= batasKelembapanRendah);      if
(kondisiPompaNyala && !pompaAktif && (sekarang -
pompaStartTime >= DURASI_JEDA_POMPA)) {
pompaAktif = true;
pompaStartTime = sekarang;
digitalWrite(RELAY_POMPA, LOW);

Serial.println("[SPRAYER]      ✓      ON      (Otomatis      -
Ammonia/Kelembapan rendah)");
} else if (pompaAktif) {
if ((sekarang - pompaStartTime >= DURASI_POMPA) ||
kondisiPompaAman) {
pompaAktif = false;
pompaStartTime = sekarang;
digitalWrite(RELAY_POMPA, HIGH);

Serial.println("[SPRAYER]      ✗      OFF      (Ammonia/Kelembapan
normal)");
}
}
}

//

```

```

// RELAY 3: LAMPU - Control Suhu (Pemanas)
// Trigger: Suhu < 20°C
//

```

```

if (lampuDariServer) {

```

```

if (!lampuAktif) {
lampuAktif = true;
digitalWrite(RELAY_LAMPU, LOW);
Serial.println("[LAMPU] ✓ ON (Manual dari Web)");
}
} else {
if (suhuTerbaru > 0 && suhuTerbaru < batasSuhuDingin) {
if (!lampuAktif) {
lampuAktif = true;
digitalWrite(RELAY_LAMPU, LOW);
Serial.println("[LAMPU] ✓ ON (Otomatis - Suhu dingin)");
}
} else {
if (lampuAktif) {
lampuAktif = false;
digitalWrite(RELAY_LAMPU, HIGH);
Serial.println("[LAMPU] ✗ OFF (Suhu normal)");
}
}
}
delay(10);
}

```

