

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing Tugas Akhir Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

NAMA : Ida Afriliana, S.T., M.Kom.
NIDN : 0624047703
NIPY : 12.013.168
Jabatan Struktural : Wakil Dekan Sekolah Vokasi
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Abdiyasa Pradana Zakaria
NIM : 23040099
Program Studi : D-III Teknik Komputer
Judul TA : PENGEMBANGAN SISTEM MONITORING
DAN KONTROL PENYIRAMAN OTOMATIS
UNTUK PEMBIBITANTANAMAN DI GREEN
HOUSE KWT MELATI

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 2 Februari 2026

Menyetujui

Ka. Prodi DIII Teknik Komputer,

Pembimbing I,


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


Ida Afriliana, S.T., M.Kom.
NIPY. 12.013.168

Lampiran 2 Surat Kesiediaan Membimbing Tugas Akhir Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

NAMA : Ahmad Maulana, S.Kom, M.Tr.T
NIDN : 0621099003
NIPY : 11.011.097
Jabatan Struktural : Kepala Administrasi Akademik
Jabatan Fungsional : Asisten Ahli

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

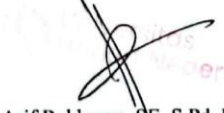
Nama : Abdiyasa Pradana Zakaria
NIM : 23040099
Program Studi : D-3 Teknik Komputer
Judul TA : PENGEMBANGAN SISTEM MONITORING
DAN KONTROL PENYIRAMAN OTOMATIS
UNTUK PEMBIBITAN TANAMAN DI GREEN
HOUSE KWT MELATI

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 2 Februari 2026

Mengetahui,

Ka. Prodi DIII Teknik Komputer,


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

Pembimbing II,


Ahmad Maulana, S.Kom, M.Tr.T
NIPY. 11.011.097

Lampiran 3 Surat Izin Observasi

 Universitas Harkat Negeri	Sekolah Vokasi Program Studi D-3 Teknik Komputer												
Nomor	: 629.03/D-3 T.Komp.HN/VI/2026												
Lampiran	: -												
Perihal	: Permohonan Izin Observasi Tugas Akhir (TA)												
Kepada Yth.													
Kepala Greenhouse KWT Melati													
Di Rumah Sastra Kita, Jl. Cereme No.4, Mangkukusuman, Kec. Tegal Tim., Kota Tegal, Jawa Tengah 52131													
Dengan Hormat,													
Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester V Program Studi Diploma III Teknik Komputer Universitas Harkat Negeri, Maka dengan ini kami mengajukan izin observasi pengambilan data di Greenhouse KWT Melati yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:													
<table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th>No</th><th>NIM</th><th>NAMA</th><th>NO HP</th></tr></thead><tbody><tr><td>1</td><td>23040099</td><td>ABDIYASA PRADANA ZAKARIA</td><td>883837313132</td></tr><tr><td>2</td><td>23040052</td><td>NAILA CAHYA WARDANI</td><td>081902851876</td></tr></tbody></table>		No	NIM	NAMA	NO HP	1	23040099	ABDIYASA PRADANA ZAKARIA	883837313132	2	23040052	NAILA CAHYA WARDANI	081902851876
No	NIM	NAMA	NO HP										
1	23040099	ABDIYASA PRADANA ZAKARIA	883837313132										
2	23040052	NAILA CAHYA WARDANI	081902851876										
Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.													
Tegal, 09 Juni 2026 Ka. Prodi DIII Teknik Komputer Universitas Harkat Negeri													
 Universitas Harkat Negeri Program Studi D-3 Teknik Komputer Arif Rakhman SE, S.Pd, M.Kom NIPY. 05.016.291													

Program Studi D-3 Teknik Komputer, Sekolah Vokasi
Gedung Utama Ruang B.09 Kampus Mataram - Jalan Mataram No.9 Kota Tegal
WhatsApp: +62-85291833673 | Surel: komputer@harkatnegeri.ac.id

Lampiran 4 Dokumentasi Observasi



Lampiran 5 Dokumentasi Pembuatan Alat



Lampiran 6 Dokumentasi Implementasi Alat



Lampiran 7 Sour Code

```
#include <WiFiManager.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <time.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
// =====
// PIN DEFINITIONS
// =====
#define DHTTYPE          DHT22
#define DHTPIN            15
#define PUMP_PENYIRAMAN  16
#define PUMP_VITAMIN_A   17
#define PUMP_VITAMIN_B   18
#define PUMP_AIR          19
#define KRAN_TANDON       26
#define KRAN_PENYIRAMAN  4
#define PUMP_LINGKUNGAN  14
#define TRIG_PIN          27
#define ECHO_PIN          23

unsigned char flowsensor      = 13;
const int    ph_Pin          = 35;
const int    tdsPin           = 32;
unsigned char flowsensortandon = 33;
unsigned long delayTime       = 1000;

// =====
// SENSOR VARIABLES
// =====
float voltage, ppmValue, ecValue;
float pHValue = 0;
float PH_step;
long duration;
float distance;
int waterLevel = 0;
int nilai_analog_PH;
double TeganganPh;

// Kalibrasi pH
float PH4 = 1.3670; // dikalibrasi dari ADC aktual
float PH7 = 0.8270; // dikalibrasi dari ADC aktual

// =====
// FLOW SENSOR
// =====
volatile int flowPulseCount = 0;
float vol = 0.0, l_minute;
unsigned long currentTime;
unsigned long cloopTime;
float volume_per_pulse = 0.004;

volatile int flowPulseCountTandon = 0;
float volTandon = 0.0;
```

```

// =====
// VITAMIN / AIR PUMP TIMING
// =====
unsigned long previousTimeVitaminAB = 0;
unsigned long previousTimeAir      = 0;
unsigned long delayVitaminAB       = 50000;
unsigned long delayAir             = 60000;
bool vitaminPumpActive             = false;
bool airPumpActive                 = false;

// =====
// TANDON AUTO-FILL FLAGS
// =====
bool autoTandonActive = false;
bool manualOverride   = false;

const int LEVEL_MULAI_ISI = 20; // % mulai isi otomatis
const int LEVEL_BERHENTI  = 70; // % berhenti isi otomatis

// =====
// KALIBRASI ULTRASONIK
// → Ukur fisik tandonmu lalu ganti dua nilai ini:
// JARAK_KOSONG = jarak sensor ke air saat tandon KOSONG (cm)
// JARAK_PENUH  = jarak sensor ke air saat tandon PENUH (cm)
// =====
const float JARAK_KOSONG = 70.0; // dikalibrasi: RAW 62.6cm + air 7cm
const float JARAK_PENUH  = 5.0; // estimasi saat tandon penuh

// =====
// LCD – coba 0x27 dulu, jika blank ganti ke 0x3F
// =====
LiquidCrystal_I2C lcd(0x27, 20, 4);
// Jika masih blank coba: LiquidCrystal_I2C lcd(0x3F, 20, 4);

DHT dht(DHTPIN, DHTTYPE);

// =====
// LCD HELPER – cetak baris dengan auto-padding spasi
// agar karakter lama tidak tersisa
// =====
void lcdPrint(uint8_t col, uint8_t row, String text) {
    lcd.setCursor(col, row);
    // pad spasi sampai 20 karakter agar baris bersih
    while (text.length() < (unsigned)(20 - col)) text += ' ';
    lcd.print(text.substring(0, 20 - col));
}

// Bersihkan satu baris saja
void lcdClearRow(uint8_t row) {
    lcd.setCursor(0, row);
    lcd.print("                "); // 20 spasi
}

```

```

// =====
//  FIREBASE PATHS
// =====
String relaytandon    = "/Control/Tandon";
String path           = "/Control/AutoVitamin";
String vitaminA       = "/Control/pomp1";
String vitaminB       = "/Control/pomp2";
String Air            = "/Control/pomp3";
String spari          = "/Control/pomp4";
String minppm         = "/SetPPM/MinPPM";
String maxppm         = "/SetPPM/MaxPPM";
String Vitamin        = "/Vitamin";
String Datapenyiraman = "/Control/penyiraman";
String modePath       = "/Control/Auto";
String waktu          = "/Setwaktu/waktu";

#define FIREBASE_HOST    "https://monitoring-60ab8-default-
rtadb.firebaseio.com/"
#define FIREBASE_AUTH    "1TlMbPqKdzIyIVUHEzPlJ5j0KxyGnXlbIKdbzpit"

FirebaseData  firebaseData;
FirebaseConfig firebaseConfig;
FirebaseAuth  firebaseAuth;

int currentHour, currentMinute;

// =====
//  INTERRUPT HANDLERS
// =====
void IRAM_ATTR flow() {
    flowPulseCount++;
}
void IRAM_ATTR flowTandon() {
    flowPulseCountTandon++;
}

// =====
//  NTP TIME
// =====
void configureTime() {
    configTime(7 * 3600, 0, "pool.ntp.org", "time.nist.gov");
    Serial.print("NTP sync");
    time_t now = time(nullptr);
    while (now < 8 * 3600 * 2) {
        delay(500);
        Serial.print(".");
        now = time(nullptr);
    }
    Serial.println(" OK");
    struct tm ti;
    localtime_r(&now, &ti);
    Serial.println(asctime(&ti));
}

// =====
//  SETUP

```

```
// =====
void setup() {
    WiFi.mode(WIFI_STA);
    Serial.begin(115200);

    // --- LCD init PERTAMA, sebelum WiFi ---
    // Ini penting agar LCD langsung menyala saat boot
    Wire.begin();           // pastikan I2C dimulai
    lcd.begin(20, 4);
    lcd.backlight();
    lcd.clear();
    lcdPrint(0, 0, "Sistem Monitoring");
    lcdPrint(0, 1, "Bibit Tanaman");
    lcdPrint(0, 2, "Menghubungkan WiFi..");
    delay(1000);

    // --- WiFi ---
    WiFiManager wm;
    bool res = wm.autoConnect("tselhome-C932", "A4DGFGTJGHA");
    if (!res) {
        lcd.clear();
        lcdPrint(0, 0, "WiFi GAGAL!");
        lcdPrint(0, 1, "Restart...");
        delay(2000);
        ESP.restart();
    }
    Serial.println("WiFi connected");
    lcd.clear();
    lcdPrint(0, 0, "WiFi Terhubung!");
    lcdPrint(0, 1, String(WiFi.localIP().toString()));
    delay(1500);

    configureTime();

    // --- Sensor & Relay ---
    dht.begin();
    pinMode(ph_Pin,          INPUT);
    pinMode(tdsPin,          INPUT);
    pinMode(PUMP_PENYIRAMAN, OUTPUT);
    pinMode(PUMP_VITAMIN_A,  OUTPUT);
    pinMode(PUMP_VITAMIN_B,  OUTPUT);
    pinMode(PUMP_LINGKUNGAN, OUTPUT);
    pinMode(PUMP_AIR,        OUTPUT);
    pinMode(KRAN_TANDON,     OUTPUT);
    pinMode(KRAN_PENYIRAMAN, OUTPUT);
    pinMode(TRIG_PIN,        OUTPUT);
    pinMode(ECHO_PIN,        INPUT);

    // Semua relay OFF (active LOW)
    digitalWrite(PUMP_PENYIRAMAN, HIGH);
    digitalWrite(PUMP_VITAMIN_A,  HIGH);
    digitalWrite(PUMP_VITAMIN_B,  HIGH);
    digitalWrite(PUMP_LINGKUNGAN, HIGH);
    digitalWrite(PUMP_AIR,        HIGH);
    digitalWrite(KRAN_TANDON,     HIGH);
    digitalWrite(KRAN_PENYIRAMAN, HIGH);
}
```

```

    pinMode(flowsensor, INPUT_PULLUP);
    pinMode(flowsensortandon, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(flowsensor), flow,
RISING);
    attachInterrupt(digitalPinToInterrupt(flowsensortandon),
flowTandon, RISING);

    currentTime = millis();

    // --- Firebase ---
    firebaseConfig.host = FIREBASE_HOST;
    firebaseConfig.signer.tokens.legacy_token = FIREBASE_AUTH;
    Firebase.begin(&firebaseConfig, &firebaseAuth);

    lcd.clear();
    lcdPrint(0, 0, "Sistem Siap!");
    delay(1500);
    lcd.clear();
}

// =====
// LOOP
// =====

// Timer LCD utama - update tiap 5 detik
unsigned long lcdTimer = 0;
uint8_t lcdPage = 0; // 0=waktu, 1=sensor1, 2=sensor2,
3=tandon
const unsigned long LCD_INTERVAL = 5000;

void loop() {
    float t = dht.readTemperature();
    float h = dht.readHumidity();

    // Baca semua sensor & kirim Firebase
    tds();
    ph();
    lingkungan();
    waterlevel(); // waterLevel diperbarui di sini

    // Kontrol aktuator
    seprai();
    isitandon(); // auto + manual tandon (pakai waterLevel terbaru)
    penyiraman();
    vitamin();

    // Simpan history
    history();

    // — LCD rotating display tiap 5 detik
    if (millis() - lcdTimer >= LCD_INTERVAL) {
        lcdTimer = millis();

        switch (lcdPage) {

```

```

case 0: {
    // Halaman 0: Tanggal & Waktu
    time_t now = time(nullptr);
    struct tm ti;
    localtime_r(&now, &ti);
    char dateBuf[11], timeBuf[9];
    strftime(dateBuf, sizeof(dateBuf), "%Y-%m-%d", &ti);
    strftime(timeBuf, sizeof(timeBuf), "%H:%M:%S", &ti);
    lcd.clear();
    lcdPrint(0, 0, "== Waktu Sekarang ==");
    lcdPrint(0, 1, String(dateBuf));
    lcdPrint(0, 2, String(timeBuf));
    lcdPrint(0, 3, "                ");
    break;
}

case 1: {
    // Halaman 1: Suhu, Kelembapan, pH
    t = isnan(t) ? 0 : t;
    h = isnan(h) ? 0 : h;
    lcd.clear();
    lcdPrint(0, 0, "== Lingkungan =====");
    lcdPrint(0, 1, "Suhu : " + String(t, 1) + " C");
    lcdPrint(0, 2, "Humid: " + String(h, 1) + " %");
    lcdPrint(0, 3, "pH   : " + String(pHValue, 2));
    break;
}

case 2: {
    // Halaman 2: PPM & EC
    lcd.clear();
    lcdPrint(0, 0, "== Nutrisi =====");
    lcdPrint(0, 1, "PPM  : " + String((int)round(ppmValue)));
    lcdPrint(0, 2, "EC   : " + String((int)round(ecValue)));
    lcdPrint(0, 3, "                ");
    break;
}

case 3: {
    // Halaman 3: Water Level Tandon + status
    String statusTandon;
    if (autoTandonActive) statusTandon = "Auto Mengisi...";
    else if (manualOverride) statusTandon = "Override Manual";
    else if (waterLevel <= LEVEL_MULAI_ISI) statusTandon =
"Level Rendah!";
    else if (waterLevel >= LEVEL_BERHENTI) statusTandon =
"Tandon Penuh";
    else
        statusTandon = "Normal";

    lcd.clear();
    lcdPrint(0, 0, "== Tandon Air =====");
    lcdPrint(0, 1, "Level: " + String(waterLevel) + " %");

    // Bar visual level air (18 karakter lebar)
    int barLen = map(waterLevel, 0, 100, 0, 18);
    String bar = "[";

```

```

        for (int i = 0; i < 18; i++) bar += (i < barLen) ? "#" : "-";
    };

    bar += "]";
    lcdPrint(0, 2, bar);
    lcdPrint(0, 3, statusTandon);
    break;
}
}

// Maju ke halaman berikutnya (0→1→2→3→0)
lcdPage = (lcdPage + 1) % 4;
}

delay(100);
}

// =====
// HISTORY
// =====
void history() {
    static int lastSavedMinute = -1;

    time_t now = time(nullptr);
    struct tm ti;
    localtime_r(&now, &ti);
    currentHour = ti.tm_hour;
    currentMinute = ti.tm_min;

    // Simpan tepat tiap jam 06-17, dan hanya sekali per menit
    if (currentHour >= 6 && currentHour <= 17 &&
        currentMinute == 0 && lastSavedMinute != currentHour) {
        lastSavedMinute = currentHour;

        float t = dht.readTemperature();
        float h = dht.readHumidity();
        t = isnan(t) ? 0 : t;
        h = isnan(h) ? 0 : h;

        char dateBuf[11], timeBuf[6];
        strftime(dateBuf, sizeof(dateBuf), "%Y-%m-%d", &ti);
        strftime(timeBuf, sizeof(timeBuf), "%H:%M", &ti);

        String histPath = "/History/" + String(dateBuf) + "/" +
String(timeBuf);
        Firebase.setFloat(firebaseData, histPath + "/ec",
(float)round(ecValue));
        Firebase.setFloat(firebaseData, histPath + "/hump", h);
        Firebase.setFloat(firebaseData, histPath + "/temp", t);
        Firebase.setFloat(firebaseData, histPath + "/ppm",
(float)round(ppmValue));
        Firebase.setFloat(firebaseData, histPath + "/ph", pHValue);
        Serial.println("History disimpan: " + histPath);
    }
}

// =====

```

```

// WATER LEVEL – ultrasonik + kalibrasi + diagnostik
//
// Jika terus "Timeout": cek kabel TRIG(27) dan ECHO(23)
// Untuk kalibrasi lihat nilai RAW di Serial Monitor:
//     tandon kosong -> JARAK_KOSONG
//     tandon penuh  -> JARAK_PENUH
// =====
void waterlevel() {
    static unsigned long lastRead    = 0;
    static int           timeoutCount = 0;

    if (millis() - lastRead < 500) return;
    lastRead = millis();

    // Pastikan TRIG LOW dulu
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(4);

    // Kirim pulsa TRIG 10us
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    // Baca ECHO timeout 25ms
    duration = pulseIn(ECHO_PIN, HIGH, 25000);

    if (duration == 0) {
        timeoutCount++;
        Serial.print("[Ultrasonic] Timeout #");
        Serial.println(timeoutCount);
        if (timeoutCount % 10 == 0) {
            Serial.println("[Ultrasonic] >> Cek kabel: TRIG=GPIO27,
ECHO=GPIO23");
            Serial.println("[Ultrasonic] >> Pastikan VCC sensor 5V dan
GND tersambung.");
        }
        return;
    }

    timeoutCount = 0;
    distance = duration * 0.034f / 2.0f;

    Serial.print("[Ultrasonic] RAW: ");
    Serial.print(distance, 1);
    Serial.print(" cm");

    // Validasi rentang fisik tandon
    if (distance < (JARAK_PENUH - 3.0f) || distance > (JARAK_KOSONG
+ 10.0f)) {
        Serial.println(" -> Out of range, skip.");
        return;
    }
    Serial.println(" -> OK");

    // Jarak jauh = kosong (0%), jarak dekat = penuh (100%)
    waterLevel = (int)map((long)(distance * 10),

```

```

        (long) (JARAK_KOSONG * 10),
        (long) (JARAK_PENUH * 10),
        0, 100);
waterLevel = constrain(waterLevel, 0, 100);

Serial.print("[Ultrasonic] WaterLevel: ");
Serial.print(waterLevel);
Serial.println(" %");

if (!Firebase.setInt(firebaseData, "/Data/WaterLevel",
waterLevel))
    Serial.println("[Ultrasonic] Firebase gagal: " +
firebaseData.errorReason());
}

// =====
// ISI TANDON – auto (level-based) + manual tanpa konflik
//
// AUTO : nyala jika level ≤ 20%, mati jika level ≥ 90%
// MANUAL ON : jalankan durasi dari Firebase (override reset)
// MANUAL OFF : paksa berhenti → set manualOverride = true
// auto diblokir sampai level ≥ 90% lagi
// =====
void isitandon() {

    // — Cek perintah manual Firebase

    if (Firebase.getString(firebaseData, relaytandon)) {
        String kondisi = firebaseData.stringData();

        if (kondisi == "1") {
            // User minta isi manual
            manualOverride = false;
            autoTandonActive = false;
            Serial.println("[Tandon] Manual ON");

            lcd.clear();
            lcdPrint(0, 0, "== TANDON =====");
            lcdPrint(0, 1, "Isi Manual...");
            lcdPrint(0, 2, "Level: " + String(waterLevel) + " %");
            lcdPrint(0, 3, " ");

            digitalWrite(KRAN_TANDON, LOW);

            if (Firebase.getString(firebaseData, waktu)) {
                int detik = firebaseData.stringData().toInt();
                if (detik > 0) delayTime = (unsigned long)detik * 1000UL;
            }
            delay(delayTime);
            digitalWrite(KRAN_TANDON, HIGH);
            Firebase.setString(firebaseData, relaytandon, "0");

            lcd.clear();
            lcdPrint(0, 0, "== TANDON =====");
            lcdPrint(0, 1, "Manual selesai.");
            lcdPrint(0, 2, "Level: " + String(waterLevel) + " %");

```

```

    delay(2000);
    lcd.clear();
    Serial.println("[Tandon] Manual selesai.");

} else if (kondisi == "0" && autoTandonActive) {
    // User paksa matikan saat auto jalan
    autoTandonActive = false;
    manualOverride = true;
    digitalWrite(KRAN_TANDON, HIGH);
    Firebase.setString(firebaseData, relaytandon, "0");

    lcd.clear();
    lcdPrint(0, 0, "== TANDON =====");
    lcdPrint(0, 1, "Dihentikan manual.");
    lcdPrint(0, 2, "Level: " + String(waterLevel) + " %");
    lcdPrint(0, 3, "Auto diblokir smt.");
    delay(2000);
    lcd.clear();
    Serial.println("[Tandon] Auto dihentikan paksa, override
aktif.");
}

} else {
    Serial.println("[Tandon]      Gagal      baca      Firebase:      "      +
firebaseData.errorReason());
}

// — Reset override jika level sudah cukup penuh


---


if (manualOverride && waterLevel >= LEVEL_BERHENTI) {
    manualOverride = false;
    Serial.println("[Tandon]      Override      direset,      level      "      +
String(waterLevel) + "%");
}

// — Logika AUTO


---


if (!manualOverride) {
    // Mulai isi
    if (!autoTandonActive && waterLevel <= LEVEL_MULAI_ISI) {
        autoTandonActive = true;
        digitalWrite(KRAN_TANDON, LOW);
        Firebase.setString(firebaseData, relaytandon, "auto_on");

        lcd.clear();
        lcdPrint(0, 0, "== TANDON =====");
        lcdPrint(0, 1, "Auto mengisi...");
        lcdPrint(0, 2, "Level: " + String(waterLevel) + " %");
        lcdPrint(0, 3, "Target: " + String(LEVEL_BERHENTI) + " %");
        Serial.println("[Tandon]      Auto      mulai      isi,      level:      "      +
String(waterLevel) + "%");
    }

    // Berhenti jika penuh
    if (autoTandonActive && waterLevel >= LEVEL_BERHENTI) {
        autoTandonActive = false;

```

```

        digitalWrite(KRAN_TANDON, HIGH);
        Firebase.setString(firebaseData, relaytandon, "0");

        lcd.clear();
        lcdPrint(0, 0, "== TANDON =====");
        lcdPrint(0, 1, "Tandon PENUH!");
        lcdPrint(0, 2, "Level: " + String(waterLevel) + " %");
        lcdPrint(0, 3, "Kran ditutup.");
        delay(3000);
        lcd.clear();
        Serial.println("[Tandon] Penuh, kran ditutup, level: " +
String(waterLevel) + "%");
    }
}

// =====
//  SEPRAI / PENDINGIN RUANGAN
// =====
void seprai() {
    static unsigned long prevMillis = 0;
    static bool aktif = false;
    const unsigned long DURATION = 30000;

    float t = dht.readTemperature();
    float h = dht.readHumidity();
    if (isnan(t) || isnan(h)) return;

    if (!Firebase.getString(firebaseData, spari)) return;
    String mode = firebaseData.stringData();

    if (mode == "1") {
        Serial.println("[Seprai] Manual ON");
        digitalWrite(PUMP_LINGKUNGAN, LOW);
        aktif = true;
        prevMillis = millis();
        Firebase.setString(firebaseData, spari, "0");
    } else if (mode == "0") {
        if (t >= 35 && !aktif) {
            Serial.println("[Seprai] Auto ON, suhu: " + String(t));
            digitalWrite(PUMP_LINGKUNGAN, LOW);
            aktif = true;
            prevMillis = millis();
        }
        if (aktif && millis() - prevMillis >= DURATION) {
            digitalWrite(PUMP_LINGKUNGAN, HIGH);
            aktif = false;
            Serial.println("[Seprai] Auto OFF");
        }
    }
}

// =====
//  PENYIRAMAN - jadwal otomatis & manual
// =====
void penyiraman() {

```

```

unsigned long delayTimeVitamin = 1000;
if (Firebase.getString(firebaseData, Vitamin)) {
    float m = firebaseData.stringData().toFloat();
    delayTimeVitamin = (unsigned long)(m * 60.0f * 1000.0f);
}

time_t now = time(nullptr);
struct tm ti;
localtime_r(&now, &ti);
currentHour = ti.tm_hour;
currentMinute = ti.tm_min;

if (!Firebase.getString(firebaseData, modePath)) return;
String mode = firebaseData.stringData();

// — OTOMATIS

if (mode == "on") {
    bool shouldWater = false;
    String times[] = {"pagi", "Pagifase2", "Siang", "Siangfase2", "sore", "Sorefase2"};
    for (int i = 0; i < 6; i++) {
        if (Firebase.getString(firebaseData, "/Jam/" + times[i])) {
            String s = firebaseData.stringData();
            if (currentHour == s.substring(0,2).toInt() &&
                currentMinute == s.substring(3,5).toInt()) {
                shouldWater = true; break;
            }
        }
    }
}

if (shouldWater) {
    // 1. Siram
    Serial.println("[Siram] Otomatis mulai");
    lcd.clear();
    lcdPrint(0, 0, "== PENYIRAMAN =====");
    lcdPrint(0, 1, "Menyiram tanaman...");
    lcdPrint(0, 2, "Harap tunggu...");
    lcdPrint(0, 3, " ");

    digitalWrite(PUMP_PENYIRAMAN, LOW);
    digitalWrite(KRAN_PENYIRAMAN, LOW);
    Firebase.setString(firebaseData, Datapenyiraman, "1");

    if (Firebase.getString(firebaseData, waktu)) {
        int d = firebaseData.stringData().toInt();
        if (d > 0) delayTime = (unsigned long)d * 1000UL;
    }
    delay(delayTime);
    digitalWrite(PUMP_PENYIRAMAN, HIGH);
    digitalWrite(KRAN_PENYIRAMAN, HIGH);
    Firebase.setString(firebaseData, Datapenyiraman, "0");

    // 2. Isi Vitamin A & B
    delay(30000);
    Serial.println("[Siram] Isi Vitamin AB");
}

```

```

        lcd.clear();
        lcdPrint(0, 0, "== VITAMIN =====");
        lcdPrint(0, 1, "Mengisi Vitamin AB");
        lcdPrint(0, 2, "Durasi: " + String(delayTimeVitamin/1000) +
" detik");

        digitalWrite(PUMP_VITAMIN_A, LOW);
        digitalWrite(PUMP_VITAMIN_B, LOW);
        Firebase.setString(firebaseData, vitaminA, "1");
        Firebase.setString(firebaseData, vitaminB, "1");
        delay(delayTimeVitamin);
        digitalWrite(PUMP_VITAMIN_A, HIGH);
        digitalWrite(PUMP_VITAMIN_B, HIGH);
        Firebase.setString(firebaseData, vitaminA, "0");
        Firebase.setString(firebaseData, vitaminB, "0");

        lcd.clear();
        lcdPrint(0, 0, "Penyiraman selesai");
        lcdPrint(0, 1, "Tandon diisi auto");
        lcdPrint(0, 2, "jika level <= " + String(LEVEL_MULAI_ISI) +
"%");
        delay(3000);
        lcd.clear();
        Serial.println("[Siram] Selesai. Tandon akan auto isi jika
perlu.");
    }
}

```

//

—

MANUAL

```

} else if (mode == "off") {
    if (!Firebase.getString(firebaseData, Datapenyiraman)) return;
    String cmd = firebaseData.stringData();

    if (cmd == "1") {
        Serial.println("[Siram] Manual ON");
        lcd.clear();
        lcdPrint(0, 0, "== PENYIRAMAN =====");
        lcdPrint(0, 1, "Siram manual...");
        lcdPrint(0, 2, "Harap tunggu...");

        digitalWrite(PUMP_PENYIRAMAN, LOW);
        digitalWrite(KRAN_PENYIRAMAN, LOW);

        if (Firebase.getString(firebaseData, waktu)) {
            int d = firebaseData.stringData().toInt();
            if (d > 0) delayTime = (unsigned long)d * 1000UL;
        }
        delay(delayTime);
        digitalWrite(PUMP_PENYIRAMAN, HIGH);
        digitalWrite(KRAN_PENYIRAMAN, HIGH);
        Firebase.setString(firebaseData, Datapenyiraman, "0");

        lcd.clear();
        lcdPrint(0, 0, "Siram manual selesai");
        delay(2000);
        lcd.clear();
    }
}

```

```

    } else if (cmd == "0") {
        digitalWrite(PUMP_PENYIRAMAN, HIGH);
        digitalWrite(KRAN_PENYIRAMAN, HIGH);
    }
}

// =====
// PH SENSOR
// Jika ADC < 100 → sensor belum terhubung / kabel lepas
// pH valid antara 0-14, di luar itu tidak dikirim ke Firebase
// =====
void ph() {
    // Rata-rata 5 pembacaan untuk mengurangi noise
    long sum = 0;
    for (int i = 0; i < 5; i++) {
        sum += analogRead(ph_Pin);
        delay(10);
    }
    nilai_analog_PH = sum / 5;

    Serial.print("[pH] ADC rata-rata: ");
    Serial.println(nilai_analog_PH);

    // Deteksi sensor tidak terhubung (ADC sangat rendah = noise)
    if (nilai_analog_PH < 100) {
        Serial.println("[pH] PERINGATAN: ADC terlalu rendah! Cek kabel sensor pH ke pin 35.");
        // Tampilkan peringatan di LCD halaman sensor
        // (tidak crash, hanya skip pengiriman)
        return;
    }

    TeganganPh = 3.3 / 4095.0 * nilai_analog_PH;

    // — Panduan kalibrasi (print tiap 30 detik) —————
    static unsigned long lastCalibPrint = 0;
    if (millis() - lastCalibPrint > 30000) {
        lastCalibPrint = millis();
        Serial.println("----- KALIBRASI pH -----");
        Serial.print(          "          ADC          sekarang          :          ");
        Serial.println(nilai_analog_PH);
        Serial.print( " Tegangan (V) : "); Serial.println(TeganganPh, 4);
        Serial.println(" Untuk kalibrasi:");
        Serial.println(" 1. Celup ke pH7 -> catat Tegangan -> masukkan ke PH7");
        Serial.println(" 2. Celup ke pH4 -> catat Tegangan -> masukkan ke PH4");
        Serial.print( " PH7 saat ini : "); Serial.println(PH7, 4);
        Serial.print( " PH4 saat ini : "); Serial.println(PH4, 4);
        Serial.println("-----");
    }

    PH_step = (PH4 - PH7) / 3.0;
    pHValue = 7.00 + ((PH7 - TeganganPh) / PH_step);
}

```

```

pHValue = round(pHValue * 100.0) / 100.0;

// Validasi rentang pH wajar (0-14)
if (pHValue < 0.0 || pHValue > 14.0) {
    Serial.print("[pH] Nilai di luar rentang (");
    Serial.print(pHValue, 2);
    Serial.println(") -> Perlu kalibrasi ulang PH4/PH7!");
    return;
}

Serial.print("[pH] Tegangan: "); Serial.print(TeganganPh, 3);
Serial.print(" V | pH: ");          Serial.println(pHValue, 2);

if (!Firebase.setFloat(firebaseData, "/Data/PH", pHValue))
    Serial.println("[pH]          Gagal          kirim:          "          +
firebaseData.errorReason());

    delay(200);
}

// =====
// LINGKUNGAN - suhu & kelembapan
// =====
void lingkungan() {
    float t = dht.readTemperature();
    float h = dht.readHumidity();
    if (isnan(t) || isnan(h)) { Serial.println("[DHT] Error");
return; }

    if (!Firebase.setFloat(firebaseData, "/Data/Hump", h))
        Serial.println("[DHT]          Hump          gagal:          "          +
firebaseData.errorReason());
    if (!Firebase.setFloat(firebaseData, "/Data/Temp", t))
        Serial.println("[DHT]          Temp          gagal:          "          +
firebaseData.errorReason());

    delay(200);
}

// =====
// TDS SENSOR
// =====
void tds() {
    int sensorValue = analogRead(tdsPin);
    voltage = sensorValue * (3.3f / 4095.0f);
    ppmValue = 2000.0f * voltage / 5.0f;
    ecValue = (4400.0f / 5.0f) * voltage;

    Serial.print("[TDS] PPM: "); Serial.print(ppmValue);
    Serial.print(" | EC: ");          Serial.println(ecValue);

    if          (!Firebase.setInt(firebaseData,          "/Data/PPM",
(int)round(ppmValue)))
        Serial.println("[TDS]          PPM          gagal:          "          +
firebaseData.errorReason());

```

```

        if (!Firebase.setInt(firebaseData, "/Data/EC",
(int)round(ecValue)))
            Serial.println("[TDS] EC gagal: " +
firebaseData.errorReason());
    }

// =====
// VITAMIN - auto (PPM-based) & manual
// =====
void vitamin() {
    unsigned long currentMillis = millis();

    unsigned long delayTimeVitamin = 1000;
    if (Firebase.getString(firebaseData, Vitamin)) {
        float m = firebaseData.stringData().toFloat();
        delayTimeVitamin = (unsigned long)(m * 60.0f * 1000.0f);
    }

    if (!Firebase.getString(firebaseData, path)) {
        Serial.println("[Vitamin] Gagal baca mode");
        return;
    }
    String mode = firebaseData.stringData();

    // ----- AUTO

    if (mode == "on") {
        float minPPM = 0, maxPPM = 0;
        if (!Firebase.getFloat(firebaseData, minppm)) return;
        minPPM = firebaseData.floatData();
        if (!Firebase.getFloat(firebaseData, maxppm)) return;
        maxPPM = firebaseData.floatData();

        // PPM rendah → tambah vitamin
        if (ppmValue < minPPM) {
            if (!vitaminPumpActive && currentMillis -
previousTimeVitaminAB >= delayVitaminAB) {
                previousTimeVitaminAB = currentMillis;
                Serial.println("[Vitamin] PPM rendah, pompa A&B ON");
                lcd.clear();
                lcdPrint(0, 0, "== VITAMIN ==");
                lcdPrint(0, 1, "PPM Rendah!");
                lcdPrint(0, 2, "Beri Vitamin A & B");
                lcdPrint(0, 3, "PPM: " + String((int)ppmValue));
                digitalWrite(PUMP_VITAMIN_A, LOW);
                digitalWrite(PUMP_VITAMIN_B, LOW);
                Firebase.setString(firebaseData, vitaminA, "1");
                Firebase.setString(firebaseData, vitaminB, "1");
                vitaminPumpActive = true;
            } else if (vitaminPumpActive && currentMillis -
previousTimeVitaminAB >= delayVitaminAB) {
                digitalWrite(PUMP_VITAMIN_A, HIGH);
                digitalWrite(PUMP_VITAMIN_B, HIGH);
                Firebase.setString(firebaseData, vitaminA, "0");
                Firebase.setString(firebaseData, vitaminB, "0");
                vitaminPumpActive = false;
            }
        }
    }
}

```

```

        lcd.clear();
    }
}

// PPM tinggi → tambah air
if (ppmValue > maxPPM) {
    if (!airPumpActive && currentMillis - previousTimeAir >=
delayAir) {
        previousTimeAir = currentMillis;
        Serial.println("[Vitamin] PPM tinggi, pompa air ON");
        lcd.clear();
        lcdPrint(0, 0, "== NUTRISI =====");
        lcdPrint(0, 1, "PPM Terlalu Tinggi!");
        lcdPrint(0, 2, "Tambah Air...");
        lcdPrint(0, 3, "PPM: " + String((int)ppmValue));
        digitalWrite(PUMP_AIR, LOW);
        Firebase.setString(firebaseData, Air, "1");
        airPumpActive = true;
    } else if (airPumpActive && currentMillis - previousTimeAir
>= delayAir) {
        digitalWrite(PUMP_AIR, HIGH);
        Firebase.setString(firebaseData, Air, "0");
        airPumpActive = false;
        lcd.clear();
    }
}

// _____ MANUAL

} else if (mode == "off") {

    // Vitamin A
    if (Firebase.getString(firebaseData, vitaminA)) {
        if (firebaseData.stringData() == "1") {
            Serial.println("[Vitamin] A manual ON");
            lcd.clear();
            lcdPrint(0, 0, "== VITAMIN A =====");
            lcdPrint(0, 1, "Pompa A Aktif...");
            lcdPrint(0, 2, "Durasi: " + String(delayTimeVitamin/1000)
+ "s");
            digitalWrite(PUMP_VITAMIN_A, LOW);
            delay(delayTimeVitamin);
            digitalWrite(PUMP_VITAMIN_A, HIGH);
            Firebase.setString(firebaseData, vitaminA, "0");
            lcd.clear();
        } else {
            digitalWrite(PUMP_VITAMIN_A, HIGH);
        }
    }

    // Vitamin B
    if (Firebase.getString(firebaseData, vitaminB)) {
        if (firebaseData.stringData() == "1") {
            Serial.println("[Vitamin] B manual ON");
            lcd.clear();
            lcdPrint(0, 0, "== VITAMIN B =====");

```

```

        lcdPrint(0, 1, "Pompa B Aktif...");
        lcdPrint(0, 2, "Durasi: " + String(delayTimeVitamin/1000)
+ "s");
        digitalWrite(PUMP_VITAMIN_B, LOW);
        delay(delayTimeVitamin);
        digitalWrite(PUMP_VITAMIN_B, HIGH);
        Firebase.setString(firebaseData, vitaminB, "0");
        lcd.clear();
    } else {
        digitalWrite(PUMP_VITAMIN_B, HIGH);
    }
}

// Air
if (Firebase.getString(firebaseData, Air)) {
    if (firebaseData.stringData() == "1") {
        Serial.println("[Air] Manual ON");
        lcd.clear();
        lcdPrint(0, 0, "== POMPA AIR =====");
        lcdPrint(0, 1, "Pompa Air Aktif...");
        digitalWrite(PUMP_AIR, LOW);
        delay(10000);
        digitalWrite(PUMP_AIR, HIGH);
        Firebase.setString(firebaseData, Air, "0");
        lcd.clear();
    } else {
        digitalWrite(PUMP_AIR, HIGH);
    }
}
}

delay(100);
}

```