

LAPORAN PROYEK MIKROPROSESOR

Sistem Penyiraman Otomatis Berbasis Mikrokontroler





Disusun Oleh:

Muhammad Taufiq Alhidayah Syah	230210501040
Sultan budi luhur	230210501037
Muh Akhlatul Ihsan	230210502009
Joy Febrianto Hale	230210501038
M Aflah Ogi Daffah	230210502030

PRODI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
UNIVERSITAS NEGERI MAKASSAR
TAHUN 2024

LEMBAR KONTRIBUSI TIM

No	Nama Mahasiswa	NIM	Foto	Peran	Tanda Tangan	Kontribusi
1	MUHAMMAD TAUFIQ ALHIDAYAH SYAH	230210501040		Project Manager		Mengatur jalannya proyek
2	Sultan budi luhur	230210501037		Hardware		Merancang perangkat keras
3	M AFLAH OGI DAFFAH	230210502030		Dokumentasi		Menyusun laporan proyek
4	Joy Febrianto Hale	230210501038		Software		Membuat perangkat lunak

5	Muh Akhlatul Ihsan	230210502 009		Tester		Menguji dan memastika n sistem berjalan.
---	--------------------------	------------------	-----------------------------------------------------------------------------------	--------	-------------------------------------------------------------------------------------	------------------------------------------------------

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan yang berjudul "Sistem Penyiraman Otomatis Berbasis Mikrokontroler".

Dalam kesempatan ini, kami mengucapkan terima kasih kepada Bapak Dr. Satria Gunawan Zain, S.PD, MT., selaku dosen pembimbing, atas arahan dan bimbingannya selama proses pengerjaan proyek ini. Kami juga mengucapkan terima kasih kepada rekan-rekan tim yang telah bekerja sama dengan baik, serta semua pihak yang telah membantu secara langsung maupun tidak langsung dalam pelaksanaan proyek ini.

Laporan ini disusun dengan tujuan untuk menjelaskan konsep, proses, dan hasil dari pengembangan Sistem Penyiraman Otomatis Berbasis Mikrokontroler yang dirancang untuk mempermudah penyiraman tanaman secara efisien dan otomatis.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, sehingga kami sangat menghargai kritik dan saran yang membangun. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi inspirasi untuk pengembangan lebih lanjut di bidang teknologi.

Makassar, 11 Desember 2024

M AFLAH OGI DAFFAH

ABSTRAK

Proyek ini dilatarbelakangi oleh kebutuhan akan solusi otomatisasi yang efisien dalam berbagai sektor, khususnya otomasi industri. Tujuannya adalah merancang dan mengimplementasikan sistem berbasis mikroprosesor yang mampu menjalankan fungsi utama seperti pemrosesan data real-time dan pengendalian perangkat eksternal. Dengan menggunakan metode integrasi perangkat keras dan lunak, sistem ini dirancang untuk efisiensi daya dan stabilitas tinggi. Hasil pengujian menunjukkan bahwa sistem berfungsi sesuai skenario uji, memenuhi spesifikasi awal, serta menawarkan potensi aplikasi praktis di industri. Kesimpulannya, proyek ini telah berhasil mencapai tujuan yang ditetapkan, meskipun pengembangan lebih lanjut diperlukan pada aspek optimasi algoritma dan pengurangan biaya produksi untuk meningkatkan skalabilitas.

DAFTAR ISI

LAPORAN PROYEK MIKROPROSESOR.....	1
LEMBAR KONTRIBUSI TIM.....	2
KATA PENGANTAR.....	4
ABSTRAK.....	5
DAFTAR ISI.....	6
BAB 1 PENDAHULUAN	7
1.1 Latar Belakang.....	7
1.2 Rumusan Masalah.....	7
1.3 Tujuan Proyek	7
1.4 Manfaat Proyek.....	8
1.5 Batasan Masalah	8
BAB 2 TINJAUAN PUSTAKA	9
2.1 Landasan Teori	9
2.2 Penelitian Terdahulu	12
2.3 Kerangka Teoritis	13
BAB 3 METODOLOGI	14
3.1 Desain Sistem	14
3.2 Alat dan Bahan	16
3.3 Langkah Kerja	16
3.4 Pengujian Sistem.....	17
BAB 4 HASIL DAN PEMBAHASAN	18
4.1 Hasil Implementasi.....	18
BAB 5 KESIMPULAN DAN SARAN	21
5.1 Kesimpulan.....	21
5.2 Saran	21
Daftar Pustaka.....	22
LAMPIRAN	23

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Penyiraman tanaman merupakan aktivitas penting untuk menjaga pertumbuhan dan kesehatan tanaman. Namun, dalam praktiknya, penyiraman secara manual sering kali menjadi kurang efisien karena membutuhkan waktu, tenaga, dan perhatian yang konsisten, terutama bagi masyarakat perkotaan dengan aktivitas yang padat. Selain itu, penggunaan air yang tidak terkontrol sering kali menyebabkan pemborosan sumber daya.

Kemajuan teknologi di bidang otomasi menawarkan solusi inovatif untuk mengatasi masalah ini, salah satunya melalui pemanfaatan mikrokontroler untuk merancang sistem penyiraman otomatis. Dalam proyek ini, kami menggunakan Arduino sebagai mikrokontroler utama. Arduino merupakan perangkat pengendali open-source yang dirancang untuk mempermudah penggunaan elektronik dan pengendalian di berbagai bidang. Dengan fleksibilitas dan kemampuannya, Arduino sangat ideal untuk diterapkan dalam sistem penyiraman otomatis tanaman (Al-hafiz & Erlinda, 2020).

Selain Arduino, sistem ini juga dilengkapi dengan berbagai sensor, seperti sensor kelembapan tanah, suhu, dan cahaya, untuk mendeteksi kondisi lingkungan tanaman secara detail. Sistem ini dirancang agar dapat melakukan penyiraman tanaman secara terprogram berdasarkan parameter tertentu, seperti tingkat kelembapan tanah atau waktu yang telah ditentukan. Dengan adanya sistem ini, diharapkan proses penyiraman dapat berjalan lebih efisien, praktis, dan mendukung upaya konservasi air.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam proyek ini adalah sebagai berikut:

- Bagaimana merancang sistem penyiraman otomatis berbasis mikrokontroler yang dapat bekerja secara efisien dan praktis?
- Bagaimana sistem ini dapat mendeteksi kondisi lingkungan seperti kelembapan tanah untuk mengatur penyiraman secara otomatis?

1.3 Tujuan Proyek

Tujuan yang ingin dicapai melalui proyek ini adalah:

- Merancang dan mengimplementasikan sistem penyiraman otomatis berbasis mikrokontroler.
- Membuat sistem yang dapat mendeteksi kondisi kelembapan tanah dan menyesuaikan penyiraman secara otomatis.

1.4 Manfaat Proyek

Proyek ini diharapkan memberikan manfaat sebagai berikut:

- Membantu efisiensi penyiraman tanaman, baik dari segi waktu maupun penggunaan air.
- Memberikan solusi praktis untuk masyarakat yang memiliki keterbatasan waktu dalam merawat tanaman.
- Mendukung pengembangan teknologi berbasis mikrokontroler dalam bidang otomasi.
- Menjadi referensi untuk pengembangan sistem otomatisasi serupa di masa depan.

1.5 Batasan Masalah

Agar pembahasan proyek lebih terfokus, batasan masalah dalam proyek ini adalah:

- Sistem ini dirancang khusus untuk penyiraman tanaman skala kecil, seperti pot atau kebun rumah tangga.
- Mikrokontroler yang digunakan adalah jenis Arduino Uno.
- Parameter lingkungan yang dipantau meliputi kelembapan tanah, intensitas cahaya dan suhu.
- Sistem tidak mencakup fungsi pengisian ulang otomatis pada sumber air.

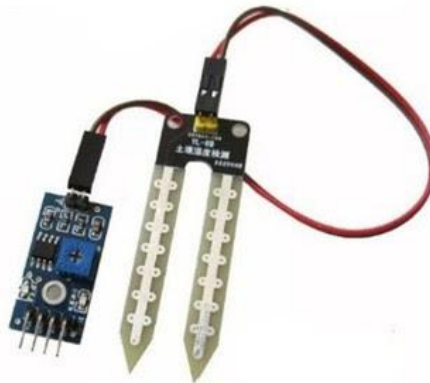
BAB 2 TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori mencakup konsep dan prinsip yang menjadi dasar dalam perancangan dan pengembangan sistem penyiraman otomatis:

a. Sensor Kelembaban Tanah

Sensor kelembaban tanah (soil moisture sensor) bekerja dengan prinsip perubahan resistansi listrik dalam tanah berdasarkan kadar air. Semakin banyak kandungan air dalam tanah, semakin kecil nilai resistansi, yang dapat diukur dalam bentuk tegangan. Nilai ini menjadi indikator untuk menentukan kebutuhan penyiraman.



Gambar 2.1 “Sensor Kelembaban Tanah”

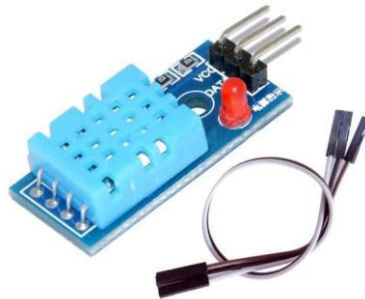
b. Sensor Suhu dan Cahaya

Sensor suhu digunakan untuk mengukur suhu lingkungan. Sistem dapat menyesuaikan frekuensi penyiraman berdasarkan suhu.



Gambar 2.2 “Sensor Cahaya”

Sensor cahaya mendeteksi intensitas cahaya untuk memantau kondisi lingkungan tanaman dan mengatur jadwal penyiraman tambahan di kondisi tertentu.



Gambar 2.3 “Sensor Suhu”

c. Mikrokontroler

Mikrokontroler adalah unit pemrosesan utama yang membaca data dari sensor, mengolah informasi, dan mengendalikan perangkat seperti pompa air berdasarkan logika yang diprogram. Mikrokontroler memungkinkan pengendalian otomatis serta integrasi dengan berbagai komponen sistem.



Gambar 2.4 “Arduino Uno”

d. Pompa Air dan Sistem Kontrol

Pompa air digunakan untuk mengalirkan air ke tanaman. Sistem kontrol (menggunakan relay) berfungsi menghidupkan atau mematikan pompa berdasarkan data dari sensor kelembaban tanah.



Gambar 2.5 “Pompa Air”

e. Timer RTC (Real-Time Clock)

Timer RTC digunakan untuk menjadwalkan penyiraman tanaman secara otomatis pada waktu tertentu. Modul ini memberikan akurasi waktu sehingga sistem penyiraman dapat bekerja sesuai jadwal.



Gambar 2.6 “RTC”

f. ESP32

ESP32 adalah mikrokontroler canggih buatan Espressif Systems yang mendukung Wi-Fi dan Bluetooth, ideal untuk aplikasi IoT. Dengan prosesor dual-core (240 MHz), banyak GPIO, ADC/DAC, dan konsumsi daya rendah, ESP32 cocok untuk proyek seperti smart home, sistem penyiraman otomatis, dan pemantauan lingkungan.



Gambar 2.7 “ESP32”

2.2 Penelitian Terdahulu

Penelitian mengenai penyiraman tanaman otomatis telah dilakukan oleh beberapa peneliti sebelumnya. Sistem irigasi tetes, misalnya, telah diterapkan pada budidaya cabai baik di lahan terbuka maupun rumah kaca, khususnya selama musim kemarau, untuk menjaga kelembapan tanah secara efektif. Sistem ini menggunakan pompa air otomatis guna mempermudah kinerja manusia dalam mengatur irigasi dan menjaga kadar air yang cukup. Beberapa studi juga mengimplementasikan logika fuzzy untuk mengendalikan sistem penyiraman otomatis. Logika fuzzy memungkinkan penentuan keputusan yang lebih presisi berdasarkan kelembapan tanah dan suhu sekitar. Sistem semacam ini terbukti mampu mengoptimalkan durasi kerja pompa air sesuai kondisi lingkungan.

Dalam penelitian terdahulu, metode berbasis SCADA (Supervisory Control and Data Acquisition) telah digunakan untuk memantau dan mengendalikan sistem secara terpusat. SCADA menyediakan dasbor interaktif yang menampilkan data kelembapan tanah dan suhu secara real-time, mempermudah pengguna untuk mengawasi dan mengontrol penyiraman secara efektif.

Penelitian ini melanjutkan dan mengembangkan studi sebelumnya dengan merancang sistem pemantauan kelembapan tanah dan suhu secara otomatis untuk beberapa tanaman dalam satu lokasi. Metode logika fuzzy Mamdani diterapkan untuk menghasilkan pola penyiraman yang optimal, sementara teknologi

SCADA Haiwell digunakan untuk memantau data dan mengelola kontrol otomatis melalui antarmuka terpusat.

2.3 Kerangka Teoritis

1. Kelembaban Tanah dan Penyiraman Otomatis

Kelembaban tanah adalah salah satu faktor utama dalam pertumbuhan tanaman. Kekurangan air dapat memperlambat proses fotosintesis dan mengganggu metabolisme tanaman. Sensor kelembaban tanah, seperti YL-69, bekerja dengan mendeteksi kadar air dalam tanah dan mengubahnya menjadi data digital yang dapat diproses untuk menentukan kebutuhan air

2. Sensor Lingkungan

Selain kelembaban tanah, suhu udara juga berpengaruh besar pada kebutuhan air tanaman. Sensor suhu, seperti DHT11 atau DHT22, digunakan untuk membaca temperatur lingkungan. Pada penelitian ini, data suhu digunakan untuk mengatur pola penyiraman, karena suhu tinggi biasanya meningkatkan kebutuhan air tanaman

3. Mikrokontroler dan Logika Fuzzy

Mikrokontroler, seperti Arduino Nano dan ESP32, bertindak sebagai pusat kendali dalam sistem. Data dari sensor diproses dengan menggunakan metode logika fuzzy Mamdani. Logika fuzzy memungkinkan sistem untuk menentukan output berdasarkan data input (kelembaban tanah dan suhu) menggunakan aturan-aturan yang telah ditetapkan, seperti durasi penyiraman berdasarkan tingkat kelembaban dan suhu

4. Sistem SCADA untuk Pemantauan

SCADA (Supervisory Control and Data Acquisition) adalah teknologi yang digunakan untuk memantau dan mengendalikan sistem secara terpusat. Dalam sistem ini, SCADA Haiwell digunakan untuk menampilkan data real-time dari sensor, seperti kadar kelembaban tanah dan suhu, serta untuk mengontrol pompa penyiraman. Teknologi ini memungkinkan pemantauan kondisi tanaman secara efisien di berbagai lokasi

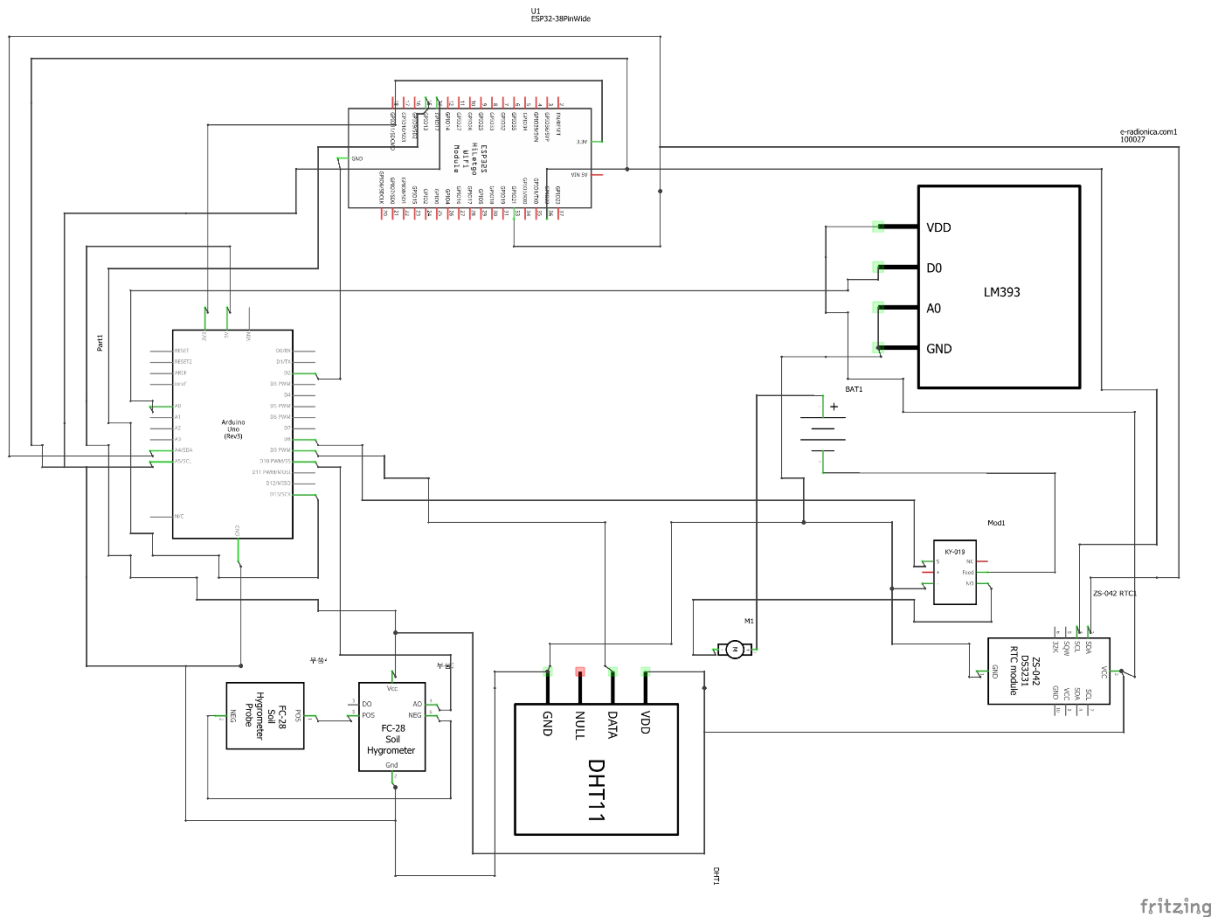
5. Automasi Penyiraman dan Penghematan Sumber Daya

Automasi penyiraman dirancang untuk meningkatkan efisiensi penggunaan air dan waktu. Dengan teknologi ini, penyiraman dilakukan hanya saat diperlukan, berdasarkan data real-time dari sensor. Pendekatan ini mengurangi pemborosan air dan meminimalkan intervensi manual

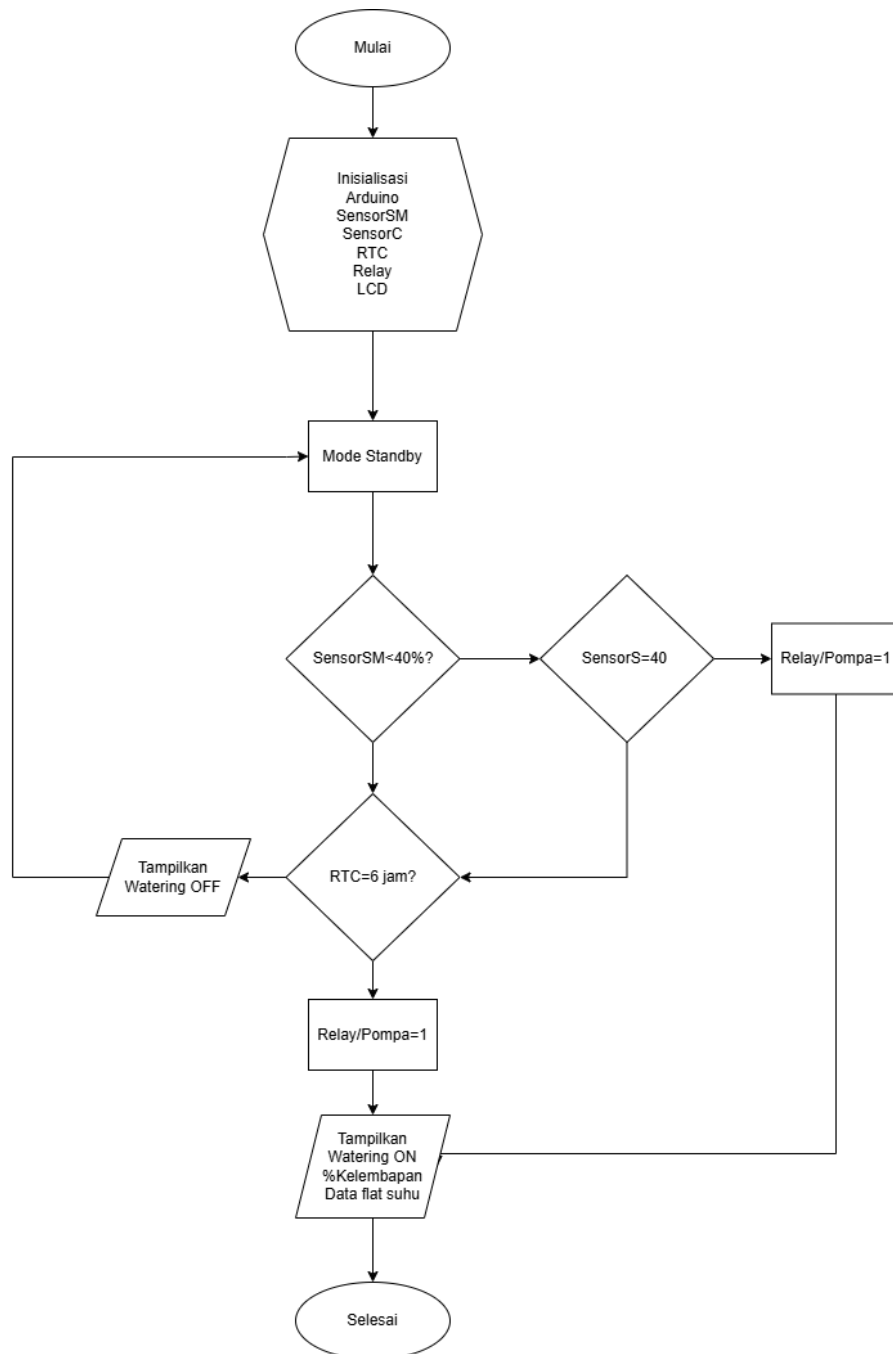
BAB 3 METODOLOGI

3.1 Desain Sistem

Sistem penyiraman otomatis berbasis mikrokontroler ini dirancang untuk mendeteksi kondisi lingkungan tanaman menggunakan berbagai sensor dan melakukan penyiraman secara otomatis berdasarkan parameter yang telah ditentukan.



Gambar 3.1 “menunjukkan skema proyek.”



Gambar 3.2 “flowchart cara kerja alat.”

Blok Diagram Sistem:

- Sensor Input:
- Sensor kelembapan tanah

- Sensor suhu
- Sensor cahaya

Mikrokontroler (Arduino dan ESP32):

Memproses data dari sensor dan menentukan tindakan yang akan diambil.

Aktuator Output:

Pompa air untuk melakukan penyiraman.

Catu Daya:

Sumber daya listrik untuk mengoperasikan sistem.

3.2 Alat dan Bahan

Daftar komponen hardware dan software yang digunakan dalam proyek ini adalah sebagai berikut:

Hardware:

- Arduino Uno: Mikrokontroler utama.
- ESP32: Mikrokontroler yang digunakan untuk terhubung ke web.
- Sensor kelembapan tanah: Mengukur tingkat kelembapan tanah.
- Sensor suhu: Mendeteksi suhu lingkungan.
- Sensor cahaya: Mengukur intensitas cahaya di sekitar tanaman.
- Pompa air mini: Untuk menyiram tanaman.
- Relay module: Mengontrol pompa air.
- Breadboard dan kabel jumper: Untuk merakit prototipe.
- Adaptor 12V: Sebagai sumber daya.

Software:

- Arduino IDE: Untuk memprogram mikrokontroler.
- Serial Monitor: Untuk memantau data dari sensor.

3.3 Langkah Kerja

Langkah kerja dalam pengembangan sistem ini meliputi:

Perancangan Sistem:

Membuat diagram alir sistem.

Menentukan komponen hardware dan software yang digunakan.

Implementasi:

Merakit perangkat keras sesuai desain.

Mengunggah program ke Arduino menggunakan Arduino IDE.

Pengujian Awal:

Memastikan semua sensor dan aktuator berfungsi dengan baik.

Menyesuaikan parameter pada program sesuai kebutuhan.

Integrasi Sistem:

Menghubungkan semua komponen dan menguji sistem secara keseluruhan.

Finalisasi:

Memperbaiki kesalahan yang ditemukan selama pengujian.

Mendokumentasikan hasil dan menyusun laporan.

3.4 Pengujian Sistem

Pengujian dilakukan untuk memastikan sistem bekerja sesuai dengan yang dirancang.

Metode Pengujian:

Pengujian Sensor: Memastikan semua sensor membaca data dengan akurat.

Pengujian Aktuator: Memastikan pompa air bekerja saat kondisi tertentu terpenuhi.

Pengujian Sistem Keseluruhan: Menjalankan sistem untuk menguji alur kerja dari awal hingga akhir.

Kriteria Keberhasilan:

Sensor kelembapan tanah dapat membaca data dengan akurat dan mengirimkan informasi ke mikrokontroler.

Pompa air aktif sesuai parameter yang telah ditentukan, seperti tingkat kelembapan di bawah ambang batas.

Sistem berjalan otomatis tanpa kesalahan logika atau teknis.

BAB 4 HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi

1. Penyajian data hasil pengujian

no	Parameter	Kondisi Awal	Input Sensor	Output Sistem	Hasil yang Diharapkan	Status
1	Kelembaban Tanah	>40% (Kering)	Sensor Soil Moisture = 45%	Pompa ON selama 10 detik	Berhasil mendeteksi kelembapan tanah	Berhasil
2	Kelembaban Tanah	<40% (Normal)	Sensor Soil Moisture = 30%	Pompa OFF	Berhasil mendeteksi kelembapan tanah	Berhasil
3	Suhu dan cahaya	suhu = < 40 dan cahaya >50	Sensor Suhu =30 dan cahaya 55	Pompa ON selama 15 detik	Berhasil mendeteksi suhu	Berhasil
4	Suhu dan udara	jika suhu berada dalam rentang 40 hingga 60 dan cahaya berada dalam rentang 40 hingga 50	Sensor Suhu = 45 dan sensor cahaya 45	Pompa ON	Berhasil mendeteksi	Berhasil

Percobaan pertama melakukan kelembapan tanah

Percobaan kedua mengetes pompa menyala atau tidak

Percobaan ketiga

Untuk Kendala nya pompa nya tidak bisa mengisap air di karena kan daya pompa kurang

2. Analisis Hasil

Interpretasi data yang digunakan

1. Kelembaban Tanah (>40%, Kering)

- Sensor berhasil mendeteksi kelembaban tanah pada level 45%.
- Pompa diaktifkan selama 10 detik, sesuai dengan ekspektasi untuk menyiram tanah yang kering.

2. Kelembaban Tanah (<40%, Normal)

- Sensor mendeteksi kelembaban tanah pada 30%, menandakan kondisi tanah sudah sangat lembab.
- Pompa tidak diaktifkan, sesuai harapan karena penyiraman tidak diperlukan.

3. Suhu dan Cahaya (Suhu <30°C, Cahaya >50)

- Sistem mendeteksi kondisi lingkungan dengan parameter suhu dan cahaya.
- Pompa diaktifkan selama 15 detik untuk penyiraman tambahan, sesuai harapan untuk mendukung kondisi tanaman di lingkungan terang.

4. Suhu dan Udara (Suhu 45°C, Cahaya >45)

- Sensor mendeteksi suhu dan cahaya tinggi.
- Pompa diaktifkan untuk penyiraman selama waktu yang telah diatur, memastikan tanaman tidak mengalami stres akibat suhu tinggi.

2. Kriteria Keberhasilan Pegujian

Keberhasilan

- Deteksi Sensor: Semua sensor (kelembaban, suhu, dan cahaya) mampu mendeteksi kondisi lingkungan sesuai parameter yang diinginkan.
- Pengaturan Output: Sistem berhasil mengontrol pompa sesuai dengan kondisi awal dan hasil yang diharapkan.

Kendala

Pompa Tidak Mengisap Air

- Disebutkan bahwa daya pompa kurang, menyebabkan masalah dalam penyiraman meskipun sistem sudah mengirim perintah ON.
- Hal ini mungkin disebabkan oleh daya pompa yang tidak memadai atau pengaturan mekanis yang perlu disesuaikan.

Efisiensi Energi

- Dalam mode penyiraman tambahan, durasi pompa perlu dikaji ulang agar tetap hemat energi tetapi mencukupi kebutuhan tanaman.

Rekomendasi Perbaikan

- Pemilihan Pompa: Gunakan pompa dengan daya hisap yang lebih baik untuk menghindari kegagalan pengisapan air.
- Kalibrasi Sensor: Pastikan sensor

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan, proyek ini berhasil mencapai sejumlah pencapaian utama. Sistem berbasis mikroprosesor telah dirancang dan diimplementasikan sesuai dengan tujuan awal, dengan komponen yang diuji menunjukkan kinerja sesuai spesifikasi. Seluruh fungsi utama sistem berjalan baik berdasarkan skenario uji yang dirancang, mencakup [sebutkan fitur utama]. Dari segi efisiensi dan stabilitas, sistem menunjukkan performa yang baik dalam hal [parameter relevan], serta tetap stabil selama pengujian tanpa kendala besar.

Selain itu, proyek ini memberikan peningkatan signifikan dalam pemahaman teknologi mikroprosesor, pemrograman embedded, serta teknik integrasi perangkat keras dan lunak, yang berkontribusi pada pengembangan keterampilan tim. Solusi yang dihasilkan juga memiliki relevansi industri dan potensi aplikasi pada [sektor relevan], memberikan nilai tambah dari sisi praktis.

Secara keseluruhan, proyek ini telah memenuhi tujuan yang ditetapkan, meskipun terdapat ruang untuk pengembangan lebih lanjut, seperti [area perbaikan], guna memastikan sistem dapat diimplementasikan secara lebih luas.

5.2 Saran

Untuk pengembangan lebih lanjut, sistem dapat ditingkatkan dengan mengintegrasikan teknologi Internet of Things (IoT) sehingga memungkinkan pemantauan dan pengendalian jarak jauh melalui aplikasi berbasis web atau smartphone. Penggunaan panel surya sebagai sumber energi akan membuat sistem lebih hemat daya dan ramah lingkungan. Selain itu, penambahan modul penyimpanan data untuk mencatat log kondisi lingkungan dapat membantu analisis jangka panjang. Sistem juga dapat diperluas skalanya untuk kebun atau lahan pertanian dengan jaringan sensor yang lebih kompleks. Penggunaan sensor yang lebih akurat, seperti sensor kelembapan kapasitif, dapat meningkatkan presisi data, sementara peningkatan pompa dengan spesifikasi daya hisap yang lebih baik akan mengatasi kendala teknis. Fitur tambahan seperti deteksi hujan dan notifikasi gangguan dapat menambah keandalan sistem. Terakhir, optimasi algoritma, misalnya dengan logika fuzzy, dapat membuat penyiraman lebih adaptif terhadap kondisi lingkungan dan kebutuhan tanaman.

Daftar Pustaka

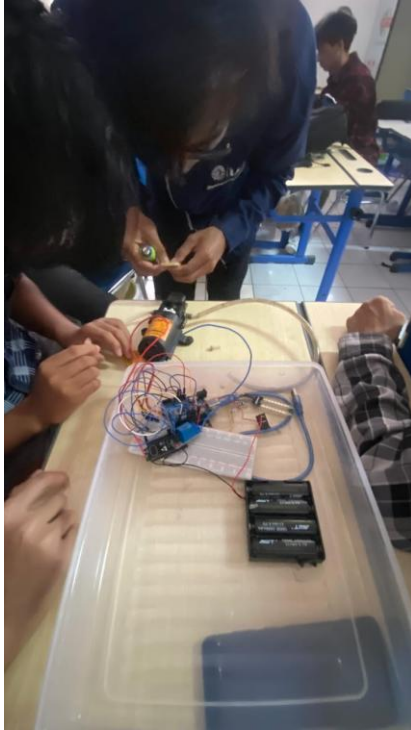
- Al-hafiz, N. W., & Erlinda. (2020). Perancangan sistem penyiraman tanaman otomatis menggunakan Arduino. *Jurnal Teknologi dan Open Source*, 3(2), 245–260.
- Umam, Z., & Irawan, D. (2024). Sistem pemantauan dan kendali penyiraman otomatis pada tanaman menggunakan logika fuzzy berbasis Outseal SCADA. *Jurnal Ampere*, 1–9.

LAMPIRAN

Foto







Github

<https://github.com/BuahPir/Sistem-Penyiraman/tree/main>

Kode Program

```
1 #include <DHT.h>
2 #include <Wire.h>
3 #include <RTClib.h>
4
5 // Definisi pin dan sensor
6 #define DHTPIN 9
7 #define DHTTYPE DHT11
8 DHT dht(DHTPIN, DHT11);
9
10 const int soilMoisturePin = 10;
11 const int relayPin = 8;
12 const int LedPin = 7;
13 int led = 0;
14 RTC_DS3231 rtc;
15
16 struct myStruct {
17   uint8_t pompa; // 1 byte instead of 4 bytes
18   uint8_t humidity; // 1 byte
19   uint8_t tanah; // 1 byte
20 };
21
22 myStruct myData;
23
24 void setup() {
25   Serial.begin(9600); // Memulai komunikasi serial
26   dht.begin();
27   Wire.begin(); // Memulai I2C
28   Wire.beginTransmission(0x21);
29   byte busStatus = Wire.endTransmission();
30   if (busStatus != 0) {
31     Serial.println("ESP32 is not found!");
32     while (1); // Wait forever
33   }
34   Serial.println("ESP32 is found.");
35
36   // Memeriksa koneksi RTC DS3231
37   if (rtc.begin()) {
38     Serial.println("Tidak dapat menemukan RTC DS3231");
39     while (1); // Hentikan program jika RTC tidak ditemukan
40   }
41
42   // Periksa apakah RTC kehilangan daya
43   if (rtc.lostPower()) {
44     Serial.println("RTC kehilangan daya. Mengatur waktu ke waktu kompilasi.");
45     rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Atur waktu RTC ke waktu kompilasi
46   }
47
48   pinMode(LedPin, OUTPUT);
49   pinMode(relayPin, OUTPUT);
50 }
51
52 String Detik(uint8_t x) {
53   String DetikStr;
54   if (x < 10) DetikStr = "0";
55   else DetikStr = "";
56   DetikStr = DetikStr + x;
57   return DetikStr;
58 }
59
60 String JamMenit(uint8_t h, uint8_t i) {
61   String JamMenit = "";
62   JamMenit = JamMenit + Detik(h) + ":" + Detik(i);
63   return JamMenit;
64 }
65
66 void loop() {
67   DateTime now = rtc.now(); // Membaca waktu saat ini dari RTC
68   int humidity = dht.readHumidity();
69   int sensorValue = analogRead(soilMoisturePin); // Membaca kelembapan tanah
70   int soilMoisturePercent = map(sensorValue, 1023, 0, 0, 100); // Konversi ke persentase
71   int SensorLDR = analogRead(ldr); // Membaca nilai sensor LDR
72   int lightPercent = map(SensorLDR, 0, 1023, 0, 100);
73
74   // Tampilkan data di serial monitor
75   Serial.print("Kelembapan Udara: ");
76   Serial.println(humidity);
77   Serial.print("Kelembapan Tanah: ");
78   Serial.println(soilMoisturePercent);
79   Serial.println("W");
80   Serial.print("Cahaya: ");
81   Serial.println(lightPercent);
82   Serial.println("W");
83   Serial.print("Jam: ");
84   Serial.println(JamMenit(now.hour(), now.minute()));
85
86   myData.humidity = dht.readHumidity();
87   myData.tanah = map(sensorValue, 1023, 0, 0, 100);
88
89   // Logika kontrol pompa dan LED
90   if (soilMoisturePercent < 30 || soilMoisturePercent < 40) {
91     myData.pompa = 1;
92     if (humidity < 40 || lightPercent > 50) {
93       digitalWrite(relayPin, LOW); // Aktifkan pompa
94       delay(1000);
95     } else if ((humidity > 40 && humidity < 60) || (lightPercent > 40 && lightPercent < 50)) {
96       digitalWrite(relayPin, LOW); // Aktifkan pompa
97       delay(1000);
98     } else {
99       digitalWrite(relayPin, LOW); // Aktifkan pompa
100      delay(1000);
101    }
102    digitalWrite(LedPin, HIGH);
103  } else {
104    digitalWrite(LedPin, LOW);
105    digitalWrite(relayPin, HIGH); // Nonaktifkan pompa
106    myData.pompa = 0;
107  }
108
109  // Kirim data ke ESP32
110  Serial.println(sizeof myData);
111  Wire.beginTransmission(0x21);
112  Wire.write((byte*)&myData, sizeof(myData)); // Mengirim seluruh struktur
113  Wire.endTransmission();
114
115  delay(1000); // Tunggu 2 detik
116 }
```

```

1  #include <Wire.h>
2  #include <HTTPClient.h>
3  #include <WiFi.h>
4
5  String url = "http://192.168.1.5/Penyiraman/index.php";
6  volatile bool flag = false;
7  byte n;
8
9  // Struktur data yang diterima melalui I2C
10 struct myStruct {
11     uint8_t pompa; // 1 byte instead of 4 bytes
12     uint8_t humidity; // 1 byte
13     uint8_t tanah; // 1 byte
14 };
15
16 myStruct myData; // Data yang diterima
17
18 const char* ssid = "Projek";
19 const char* password = "Tekom023";
20
21 IPAddress local_IP(192, 168, 1, 2);
22 IPAddress gateway(192, 168, 1, 1);
23 IPAddress subnet(255, 255, 255, 0);
24
25 String pompa;
26 void setup() {
27     Serial.begin(9600);
28     Wire.begin(0x23); // Initialize ESP32 as I2C slave with address 0x23
29     Wire.onReceive(receiveEvent); // Register receive event handler
30
31     // Konfigurasi WiFi Access Point
32     Serial.print("Setting up Access Point . . .");
33     Serial.println(WiFi.softAPConfig(local_IP, gateway, subnet) ? "Ready" : "Failed!");
34     Serial.print("Setting AP . . .");
35     Serial.println(WiFi.softAP(ssid, password) ? "Ready" : "Failed!");
36     Serial.print("IP Address = ");
37     Serial.println(WiFi.softAPIP());
38 }
39
40 void loop() {
41     if (flag) {
42         // Cetak data yang diterima
43         Serial.println("Data received:");
44         Serial.print("Message: ");
45         if (myData.pompa == 1) {
46             pompa = "Pompa_Aktif";
47             Serial.println("Pompa Aktif");
48         } else if (myData.pompa == 0) {
49             pompa = "Pompa_Tidak_Aktif";
50             Serial.println("Pompa_Tidak_Aktif");
51         }
52         Serial.print("Humidity: ");
53         Serial.println(myData.humidity);
54         Serial.print("Tanah: ");
55         Serial.println(myData.tanah);
56         // Kirim data ke server melalui HTTP
57         HTTPClient http;
58         String postData = url + "?status=" + String(pompa) + "&humidity=" + String(myData.humidity) + "&soil=" + String(myData.tanah);
59         http.begin(postData.c_str());
60         int httpCode = http.GET();
61         String payload = http.getString();
62         http.addHeader("Content-Type", "application/x-www-form-urlencoded");
63
64         // Cetak hasil HTTP
65         Serial.print("URL: "); Serial.println(url);
66         Serial.print("Data: "); Serial.println(postData);
67         Serial.print("HTTP Code: "); Serial.println(httpCode);
68         Serial.print("Payload: "); Serial.println(payload);
69         Serial.println("-----");
70         http.end();
71
72         // Reset flag
73         flag = false;
74     }
75 }
76
77 void receiveEvent(int howMany) {
78     n = howMany; // Jumlah byte yang diterima
79     Serial.print("Bytes received: ");
80     Serial.println(n);
81
82     // Periksa apakah ukuran data sesuai dengan ukuran struktur
83     if (howMany == sizeof(myData)) {
84         Wire.readBytes((byte*)&myData, sizeof(myData));
85         flag = true; // Tandai bahwa data baru telah diterima
86     } else {
87         Serial.print("Incorrect data size received. Expected: ");
88         Serial.print(sizeof(myData));
89         Serial.print(", Received: ");
90         Serial.println(howMany);
91     }
92 }

```

```

1  <?php
2  $conn = mysqli_connect("localhost", "root", "", "penyiraman");
3
4  if (mysqli_connect_errno()) {
5      die("Error connecting to database" . mysqli_connect_error());
6  }
7  $status= $_GET['status'];
8  $humidity = $_GET['humidity'];
9  $soil = $_GET['soil'];
10 $insert = "INSERT INTO penyiraman (status, humidity, soil) VALUES ('$status','$humidity','$soil')";
11
12 if (mysqli_query($conn, $insert)) {
13     echo "Success memasukkan record";
14 } else {
15     echo "Error: " . $sql . "<br>" . mysqli_error($conn);
16 }
17
18 ?>

```

```

1  <?php
2  $conn = mysqli_connect("localhost", "root", "", "penyiraman");
3
4  // Periksa koneksi database
5  if (mysqli_connect_errno()) {
6      die("Error connecting to database: " . mysqli_connect_error());
7  }
8
9  // Ambil data terakhir
10 $sql = "SELECT * FROM penyiraman ORDER BY id DESC LIMIT 1"; // Ambil data terakhir berdasarkan urutan ID
11 $qry = mysqli_query($conn, $sql);
12 $a = mysqli_fetch_assoc($qry);
13
14 // Tampilkan data dalam tabel
15 if ($a) {
16     echo "<table border='1' cellpadding='5' cellspacing='0'>";
17     echo "<tr>";
18         <th>Nomor</th>
19         <th>Status</th>
20         <th>Humidity (%)</th>
21         <th>Soil Moisture (%)</th>
22         <th>Waktu Sekarang</th>
23     </tr>";
24     echo "<tr>";
25     echo "<td>1.</td>";
26     echo "<td>" . htmlspecialchars($a['status']) . "</td>"; // Hindari XSS
27     echo "<td>" . htmlspecialchars($a['humidity']) . "</td>";
28     echo "<td>" . htmlspecialchars($a['soil']) . "</td>";
29     echo "<td>" . date("Y-m-d H:i:s") . "</td>"; // Ambil waktu saat ini
30     echo "</tr>";
31     echo "</table>";
32 } else {
33     echo "Tidak ada data.";
34 }
35
36 // Tutup koneksi database
37 mysqli_close($conn);
38 ?>

```