

Sistem Irigasi *Vertical Garden* Berbasis IoT dengan Penjadwalan Adaptif

Faizah¹, Agus Prihanto²

^{1,2} Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹faizah.22165@mhs.unesa.ac.id

²agusprihanto@unesa.ac.id

Abstrak— Tanaman pada lingkungan hunian perkotaan sering mengalami penyiraman yang tidak konsisten akibat aktivitas pengguna yang padat serta perubahan kondisi lingkungan. Penelitian ini mengembangkan sistem irigasi otomatis berbasis Internet of Things (IoT) untuk *vertical garden* skala kecil menggunakan mikrokontroler ESP32 sebagai perangkat *edge*. Sistem bekerja berdasarkan jadwal penyiraman yang diperbarui secara adaptif menggunakan data historis kelembaban media tanam tanpa bergantung pada layanan *cloud*. Data kelembaban tanah direkam setiap 15 menit dan disimpan secara lokal untuk dianalisis menggunakan algoritma berbasis *if-then*. Algoritma menghitung perubahan kelembaban tanah sebelum dan sesudah penyiraman (*delta*), kemudian mengklasifikasikan kondisi media tanam menjadi KERING, CUKUP, atau KEBANYAKAN sebagai dasar penyesuaian durasi dan jadwal penyiraman pada siklus berikutnya. Verifikasi dilakukan dengan membandingkan hasil perhitungan otomatis pada ESP32 dan perhitungan manual menggunakan dataset yang sama, termasuk pengujian tambahan menggunakan data sintesis. Hasil pengujian menunjukkan bahwa sistem mampu menghasilkan keputusan klasifikasi, perubahan durasi, dan pembaruan jadwal yang sesuai dengan hasil perhitungan manual, sehingga algoritma adaptif yang diterapkan dapat mendukung mekanisme penjadwalan penyiraman berbasis data historis pada sistem irigasi *vertical garden*.

Kata Kunci— *Internet of Things (IoT)*, *vertical garden*, penjadwalan adaptif, *soil moisture*, data historis, irigasi otomatis.

I. PENDAHULUAN

Tanaman pada lingkungan hunian perkotaan semakin populer karena mampu menghadirkan nuansa alami pada ruang terbatas. Namun, gaya hidup dengan mobilitas tinggi sering menyebabkan perawatan tanaman, khususnya proses penyiraman, dilakukan secara tidak konsisten. Ketidakpastian penyiraman manual dapat memicu kondisi *under-irrigation* maupun *over-irrigation* yang berdampak pada kesehatan tanaman dan kestabilan kelembaban media tanam. Selain itu, kondisi fisik hunian perkotaan yang sempit dan minim cahaya alami juga membuat penempatan tanaman menjadi kurang ideal sehingga memengaruhi kualitas sirkulasi udara dan kenyamanan ruang [1]. Variabilitas kondisi lingkungan tropis turut memengaruhi proses perawatan tanaman. Musim kemarau dapat meningkatkan risiko *overheating* pada tanaman [2], sedangkan musim hujan berkepanjangan dapat meningkatkan kelembaban udara dan memicu pertumbuhan jamur akibat rendahnya intensitas cahaya [3]. Beberapa tindakan perawatan manual seperti penyemprotan rutin atau pengaturan jarak tanaman memang dapat membantu proses adaptasi [4], tetapi tindakan tersebut tetap menuntut konsistensi yang tidak selalu dapat dipenuhi pengguna.

Perkembangan *Internet of Things* (IoT) membuka peluang penerapan sistem irigasi otomatis untuk membantu proses perawatan tanaman pada lingkungan hunian. IoT memungkinkan perangkat fisik saling terhubung untuk mengumpulkan data sensor dan mengendalikan aktuator secara otomatis [5]. Teknologi ini juga mulai diterapkan pada sistem *vertical garden* sebagai solusi penghijauan untuk ruang terbatas [6]. *Vertical garden* merupakan metode penanaman menggunakan dinding atau struktur vertikal sehingga mampu mengoptimalkan area tanam pada lingkungan hunian perkotaan [7].

Berbagai penelitian telah mengembangkan sistem irigasi otomatis berbasis IoT menggunakan sensor kelembaban tanah dan mikrokontroler untuk mengatur proses penyiraman. Namun, sebagian besar sistem masih menggunakan pendekatan *static scheduling* atau *threshold-based triggering* sehingga kurang responsif terhadap perubahan kondisi media tanam dan lingkungan [8], [9]. Pendekatan tersebut berpotensi menyebabkan *over-irrigation* maupun *under-irrigation* ketika kondisi kelembaban tanah berubah secara bertahap. Selain itu, beberapa sistem masih bergantung pada layanan *cloud* untuk proses pemantauan dan pengambilan keputusan, sehingga berisiko mengalami keterlambatan maupun gangguan ketika koneksi internet tidak stabil. Pendekatan tersebut juga dapat menambah latensi dan konsumsi energi sehingga kurang ideal untuk sistem irigasi skala kecil pada lingkungan hunian [10]. Di sisi lain, pemrosesan lokal terbukti lebih stabil dan efisien untuk sistem skala kecil karena tidak bergantung pada server eksternal [11].

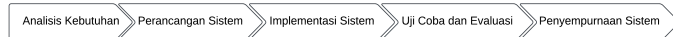
Beberapa penelitian lain telah berupaya meningkatkan akurasi pengambilan keputusan melalui pendekatan yang lebih adaptif. Kunt (2025) mengembangkan sistem berbasis LSTM untuk memprediksi kelembaban tanah dari data historis, namun memerlukan infrastruktur *cloud* dan komputasi intensif yang tidak efisien untuk skala kecil [12]. Pereira et al. (2023) menerapkan *rule-based threshold* menggunakan ESP32 yang sederhana dan mudah diterapkan, tetapi masih menggunakan nilai ambang statis yang tidak menyesuaikan perubahan kondisi [13]. Vaishnavi et al. (2025) menambahkan fitur *override* manual melalui aplikasi, namun sistemnya tetap bergantung pada *cloud* [14]. Abdelmoneim et al. (2025) membuktikan bahwa pendekatan *soil-based scheduling* 28,8% lebih efisien dibandingkan *weather-based scheduling*, namun jadwal yang digunakan masih bersifat statis [15]. Dengan demikian, penelitian mengenai *adaptive scheduling* berbasis data historis kelembaban tanah menggunakan pendekatan *rule-based* masih relatif terbatas, khususnya pada sistem *vertical garden*, dan sebagian besar penelitian masih berfokus pada pemicu penyiraman berbasis kondisi sensor *real-time*

bukan evaluasi dan pembaruan jadwal berdasarkan histori respons media tanam.

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan sistem irigasi otomatis berbasis IoT untuk *vertical garden* menggunakan algoritma *adaptive scheduling* berbasis aturan *if-then*. Sistem memanfaatkan data historis kelembaban tanah untuk mengevaluasi efektivitas penyiraman dan memperbarui jadwal serta durasi penyiraman pada siklus berikutnya. Proses adaptasi dilakukan berdasarkan perubahan kelembaban tanah sebelum dan sesudah penyiraman (Δ) yang kemudian diklasifikasikan sebagai KERING, CUKUP, atau KEBANYAKAN untuk menentukan keputusan penyesuaian jadwal [16]. Verifikasi dilakukan dengan membandingkan hasil perhitungan otomatis pada sistem dan perhitungan manual menggunakan dataset yang sama untuk memverifikasi kesesuaian keputusan adaptasi yang dihasilkan sistem.

II. METODOLOGI PENELITIAN

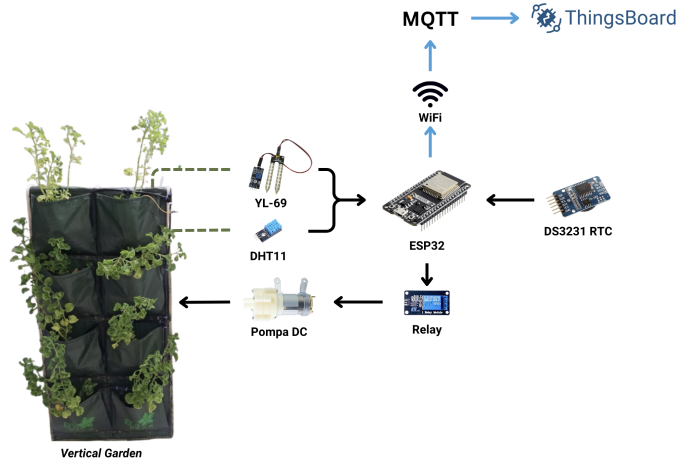
Penelitian ini menggunakan metode *Research and Development* (R&D) untuk mengembangkan sistem irigasi otomatis berbasis IoT dengan mekanisme *adaptive scheduling*. Tahapan penelitian meliputi analisis kebutuhan, perancangan arsitektur sistem, implementasi algoritma adaptif, kalibrasi sensor, pengumpulan data historis, serta pengujian dan verifikasi.



Gbr. 1 Metode *Research and Development*

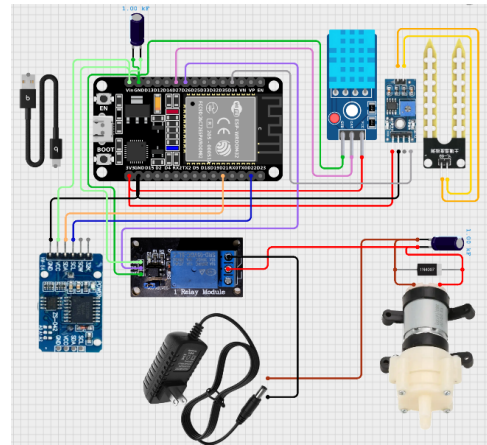
A. Arsitektur Sistem

Sistem dirancang menggunakan ESP32 DevKit V1 sebagai pengendali utama. ESP32 dipilih karena arsitektur *dual-core* dan integrasi WiFi bawaan yang memungkinkan operasi *edge-first* tanpa *gateway* tambahan, berbeda dengan Arduino Uno yang tidak memiliki WiFi *built-in* maupun ESP8266 yang hanya berjalan secara *single-core* [17]. Komponen yang diintegrasikan meliputi sensor kelembaban tanah YL-69, sensor suhu dan kelembaban udara DHT11, modul RTC DS3231, relay satu kanal, dan pompa DC 12V. Modul RTC DS3231 dipilih karena memiliki akurasi tinggi dan dilengkapi baterai cadangan sehingga waktu tetap tersimpan saat perangkat mengalami *restart* [18]. Arsitektur sistem keseluruhan ditunjukkan pada Gbr. 2.



Gbr. 2 Arsitektur Keseluruhan Sistem

Untuk menjaga kestabilan dan keamanan rangkaian, digunakan dioda *flyback* 1N4007 sebagai proteksi terhadap lonjakan tegangan balik (*back-EMF*) saat pompa dikendalikan melalui relay [19], serta kapasitor 1000 μ F untuk meredam fluktuasi tegangan akibat perubahan beban secara mendadak [20]. Sistem menerapkan arsitektur *edge-first*, di mana seluruh proses *sensing*, pengambilan keputusan, dan eksekusi dilakukan secara lokal pada ESP32 [21]. Pendekatan ini dipilih karena *edge computing* memungkinkan pemrosesan data lebih dekat ke sumbernya sehingga mengurangi latensi dan meningkatkan responsivitas [22][23]. Layanan *cloud* melalui *dashboard* ThingsBoard hanya digunakan untuk keperluan visualisasi dan analisis data skala besar [24], bukan untuk pengambilan keputusan. Perancangan sirkuit sistem ditunjukkan pada Gbr. 3.



Gbr. 3 Perancangan Sirkuit

B. Rancangan Algoritma Adaptif

Dalam penelitian ini, istilah "adaptif" merujuk pada kemampuan sistem memperbarui jadwal dan durasi penyiraman secara otomatis berdasarkan evaluasi data historis tanpa intervensi manusia. Aturan pengambilan keputusan bersifat tetap, sedangkan yang berubah adalah hasil keputusannya berdasarkan kondisi kelembaban media tanam yang terdeteksi [8]. Pendekatan berbasis aturan *if-then* diterapkan menggunakan konsep IFTTT sebagai kerangka logika kondisional [16].

Pemilihan perubahan kelembaban tanah (*soil before* dan *soil after*) sebagai variabel inti adaptasi didasarkan pada temuan Abdelmoneim et al. (2025) yang membuktikan bahwa pendekatan berbasis kelembaban tanah 28–52% lebih akurat dan lebih ringan untuk perangkat *edge* dibandingkan parameter cuaca [15]. Periode evaluasi tiga hari dipilih untuk menyeimbangkan stabilitas keputusan dan kecepatan adaptasi — periode terlalu pendek menghasilkan keputusan tidak stabil, sedangkan periode terlalu panjang memperlambat pembaruan jadwal.

Algoritma bekerja melalui tiga fase utama yang berjalan secara siklik: *monitoring*, eksekusi penyiraman, dan analisis adaptif. Seluruh data yang dikumpulkan disimpan dalam tiga jenis log seperti ditunjukkan pada Tabel I.

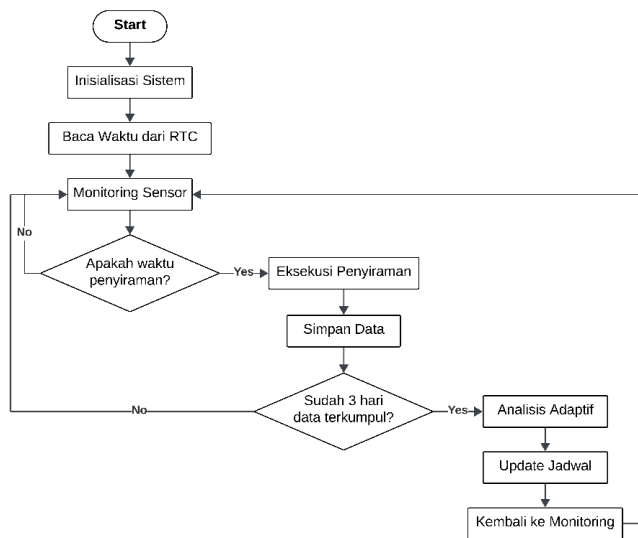
TABEL I
STRUKTUR DATA HISTORIS

Jenis Log	Interval	Informasi Utama yang Disimpan
Log <i>Monitoring</i>	Setiap 15 Menit	<i>Timestamp</i> , kelembapan tanah, suhu udara, kelembapan udara, status relay

Jenis Log	Interval	Informasi Utama yang Disimpan
Log Peristiwa	Setiap Penyiraman	Nilai <i>soil_before</i> , <i>soil_after</i> , selisih (Δ), durasi pompa, klasifikasi respons
Log Learning	Setiap 3 Hari	Nomor siklus, klasifikasi dominan, perubahan durasi penyiraman

Pada fase *monitoring*, ESP32 membaca sensor YL-69 dan DHT11 setiap 15 menit dan menyimpan hasilnya ke SPIFFS dalam format CSV sebagai log historis. Jadwal dan parameter adaptasi disimpan pada *Non-Volatile Storage* (NVS) agar tetap persisten saat perangkat mengalami *reset*.

Pada fase eksekusi penyiraman, sistem mengaktifkan pompa berdasarkan jadwal yang tersimpan, bukan reaktif terhadap sensor secara langsung. Setelah penyiraman selesai, sistem menunda 10 detik sebelum membaca ulang kelembaban tanah untuk mengukur dampak penyiraman. Alur kerja sistem ditunjukkan pada Gbr. 4.



Gbr. 4 Flowchart Sistem

Pada fase analisis adaptif, setiap tiga hari sistem menghitung delta kelembaban menggunakan persamaan:

$$\Delta = \text{soil_before} - \text{soil_after} \quad (1)$$

Nilai Δ diklasifikasikan berdasarkan *threshold* yang ditetapkan melalui proses kalibrasi (lihat Bagian II-C):

- KERING: $\Delta < \text{DELTA_MIN}$
- CUKUP: $\text{DELTA_MIN} \leq \Delta \leq \text{DELTA_MAX}$
- KEBANYAKAN: $\Delta > \text{DELTA_MAX}$

Logika algoritma secara keseluruhan dapat direpresentasikan dalam *pseudocode* berikut:

```
// Klasifikasi delta penyiraman
FUNGSI klasifikasiDelta(delta):
    JIKA delta < DELTA_MIN MAKA kembalikan "KERING"
    JIKA delta ≤ DELTA_MAX MAKA kembalikan "CUKUP"
    SELAINNYA kembalikan "KEBANYAKAN"
```

```
// Agregasi modus selama 3 hari
FUNGSI tentukanModus(kering, cukup, kebanyakan):
    JIKA kering paling dominan MAKA kembalikan
    "KERING"
    JIKA kebanyakan paling dominan MAKA kembalikan
    "KEBANYAKAN"
    SELAINNYA kembalikan "CUKUP"
```

```
// Penyesuaian durasi
JIKA modus == "KERING" MAKA durasi += STEP_DURASI
```

```
JIKA modus == "KEBANYAKAN" MAKA durasi -= STEP_DURASI
SELAINNYA durasi tetap
```

```
// Penyesuaian jadwal - Global Shift
JIKA M_norm_global < THRESHOLD_KERING MAKA
    jadwal_pagi -= GLOBAL_SHIFT
    jadwal_siang -= GLOBAL_SHIFT
JIKA M_norm_global > THRESHOLD_LEMBAB MAKA
    jadwal_pagi += GLOBAL_SHIFT
    jadwal_siang += GLOBAL_SHIFT
```

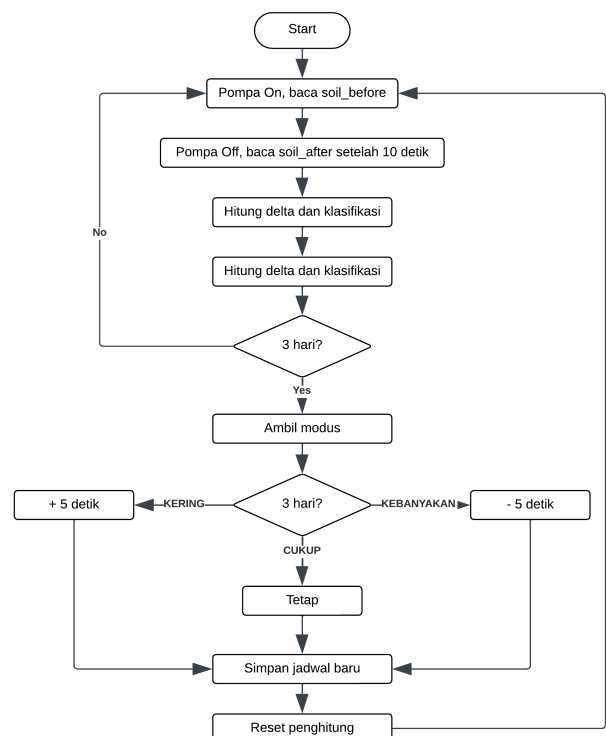
```
// Penyesuaian jadwal - Fine Tuning
JIKA M_norm_pagi < THRESHOLD_KERING MAKA jadwal_pagi
    -= FINE_TUNE
JIKA M_norm_pagi > THRESHOLD_LEMBAB MAKA jadwal_pagi
    += FINE_TUNE
JIKA M_norm_siang < THRESHOLD_KERING MAKA
    jadwal_siang -= FINE_TUNE
JIKA M_norm_siang > THRESHOLD_LEMBAB MAKA
    jadwal_siang += FINE_TUNE
```

```
SIMPAN jadwal dan durasi baru
RESET penghitung siklus
```

Keputusan adaptasi durasi ditentukan berdasarkan modus klasifikasi selama tiga hari seperti ditunjukkan pada Tabel II. Alur adaptasi durasi ditunjukkan pada Gbr. 5.

TABEL III
ATURAN ADAPTASI DURASI PENYIRAMAN

Kasus	Interval Hasil Evaluasi	Keputusan Sistem
1	Modus klasifikasi pagi dominan KERING	Durasi penyiraman pagi ditambah 5 detik
2	Modus klasifikasi pagi dominan KEBANYAKAN	Durasi penyiraman pagi dikurangi 5 detik
3	Modus klasifikasi pagi dominan CUKUP	Durasi penyiraman pagi dipertahankan
4	Modus klasifikasi siang dominan KERING	Durasi penyiraman siang ditambah 5 detik
5	Modus klasifikasi siang dominan KEBANYAKAN	Durasi penyiraman siang dikurangi 5 detik
6	Modus klasifikasi siang dominan CUKUP	Durasi penyiraman siang dipertahankan

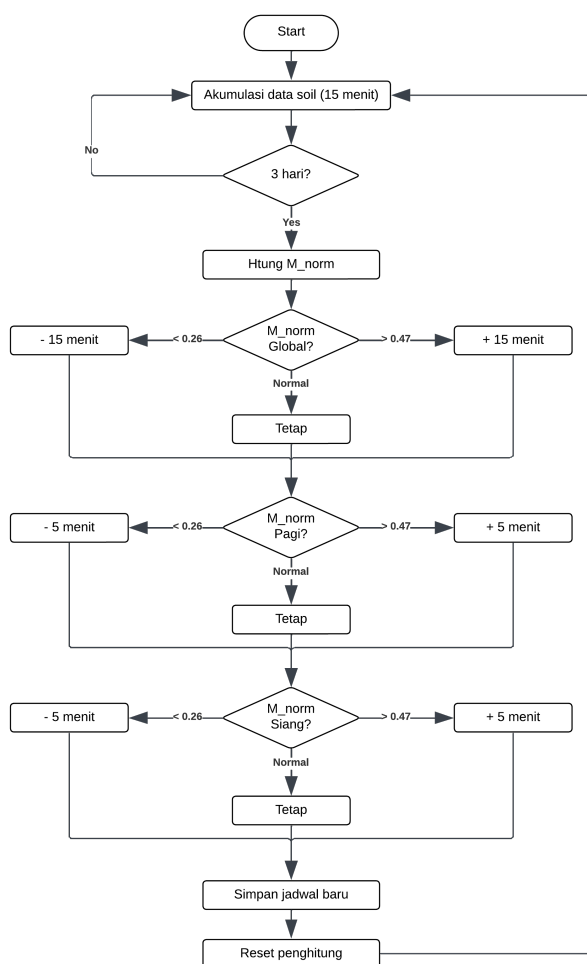


Gbr. 5 Flowchart Adaptasi Durasi Penyiraman

Selain adaptasi durasi, sistem memperbarui jadwal penyiraman berdasarkan rata-rata M_{norm} melalui mekanisme dua tahap: *Global Shift* ± 15 menit untuk kondisi menyeluruh dan *Fine Tuning* ± 5 menit untuk koreksi spesifik per sesi. Aturan adaptasi jadwal ditunjukkan pada Tabel III dan alurnya pada Gbr. 6.

TABEL IIIII
ATURAN ADAPTASI JADWAL PENYIRAMAN

Kasus	Interval Hasil Evaluasi	Keputusan Sistem
1	Nilai M_{norm} global berada di bawah <i>threshold</i> kering	Jadwal penyiraman dimajukan 15 menit (<i>Global Shift</i>)
2	Nilai M_{norm} global berada di atas <i>threshold</i> lembab	Jadwal penyiraman diundur 15 menit (<i>Global Shift</i>)
3	Nilai M_{norm} pagi berada di bawah <i>threshold</i> kering	Jadwal pagi dimajukan 5 menit (<i>Fine Tuning</i>)
4	Nilai M_{norm} pagi berada di atas <i>threshold</i> lembab	Jadwal pagi diundur 5 menit (<i>Fine Tuning</i>)
5	Nilai M_{norm} siang berada di bawah <i>threshold</i> kering	Jadwal siang dimajukan 5 menit (<i>Fine Tuning</i>)
6	Nilai M_{norm} siang berada di atas <i>threshold</i> lembab	Jadwal siang diundur 5 menit (<i>Fine Tuning</i>)
7	Nilai M_{norm} berada pada rentang normal	Jadwal dipertahankan



Gbr. 6 Flowchart Adaptasi Waktu Penyiraman

Jadwal hasil pembaruan disimpan ke NVS sebagai jadwal aktif untuk siklus berikutnya. Sistem tidak mendeteksi hujan secara langsung, melainkan dampaknya melalui perubahan nilai M_{norm} yang kemudian memicu penyesuaian jadwal secara otomatis.

C. Kalibrasi Sensor dan Penentuan Threshold

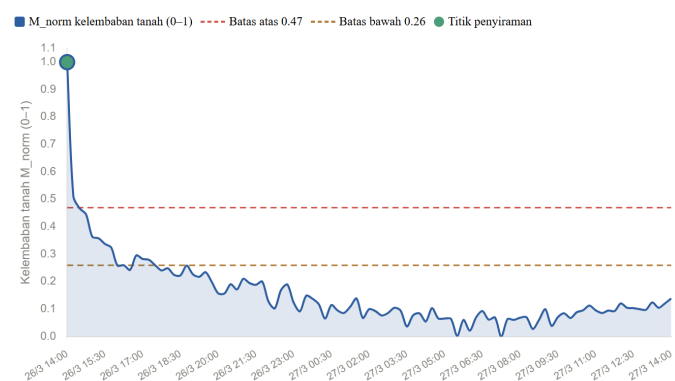
Nilai *threshold* delta (DELTA_MIN dan DELTA_MAX) tidak ditetapkan secara arbitrer, melainkan diperoleh melalui proses kalibrasi sensor dan pengujian eksperimental. Kalibrasi sensor YL-69 dilakukan dengan menetapkan dua titik ekstrem pada media tanam *cocopeat*. Nilai $S_{wet} = 1333$ diperoleh saat media dalam kondissi jenuh air, dan $S_{dry} = 3184$ saat media dibiarkan mengering selama 24 jam. Sensor kapasitif menawarkan ketahanan korosi yang lebih baik dibandingkan sensor resistif seperti YL-69 [15], namun YL-69 dipilih karena biaya rendah dan ketersediaan yang memadai untuk prototipe skala kecil. Nilai analog sensor dikonversi ke skala kelembaban ternormalisasi M_{norm} menggunakan persamaan:

$$M_{norm} = (S_{dry} - S) / (S_{dry} - S_{wet}) \quad (2)$$

Analisis kurva pengeringan *cocopeat* selama 24 jam mengonfirmasi fase transisi kelembaban berada pada rentang 0,47–0,26 M_{norm} , di mana nilai 0,47 merupakan titik perlambatan laju awal dan 0,26 adalah ambang batas menuju kestabilan laju perubahan. Rentang ini selanjutnya digunakan sebagai *threshold* pembaruan jadwal. Kurva pengeringan ditunjukkan pada Gbr. 7 dan hasil normalisasi nilai analog sensor ke skala M_{norm} ditunjukkan pada Gbr. 8.



Gbr. 7 Kurva Kelembaban Tanah



Gbr. 8 Hasil Normalisasi M_{norm}

Penentuan *threshold* delta dilakukan berdasarkan pengujian enam sesi penyiraman dengan durasi konstan 15 detik pada media *cocopeat*. Data hasil pengujian ditunjukkan pada Tabel IV.

TABEL IVV
DATA PENYIRAMAN UNTUK PENENTUAN *THRESHOLD* DELTA

No.	Tanggal	Waktu	Sebelum	Sesudah	Δ
1	31/03/2026 06:00	Pagi	2609	2421	188
2	31/03/2026 12:00	Siang	2411	1839	572
3	01/04/2026 06:00	Pagi	2691	2539	152
4	01/04/2026 12:00	Siang	2605	2467	138
5	02/04/2026 06:00	Pagi	2877	2851	26
6	02/04/2026 12:00	Siang	2802	2677	125

Nilai ekstrem $\Delta = 26$ (sesi 5) dan $\Delta = 572$ (sesi 2) diabaikan karena tidak representatif akibat kondisi sensor tidak stabil. Dari empat sesi yang valid diperoleh rentang delta stabil 125–188. DELTA_MIN ditetapkan sebesar 120, yaitu 5 satuan di bawah nilai stabil terendah (125), sedangkan DELTA_MAX ditetapkan sebesar 200, yaitu 12 satuan di atas nilai stabil tertinggi (188). Toleransi ini diberikan untuk mengakomodasi variasi minor pembacaan sensor. Nilai ini bersifat *site-specific* terhadap kombinasi sensor YL-69 dan media tanam *cocopeat* yang digunakan dalam penelitian ini.

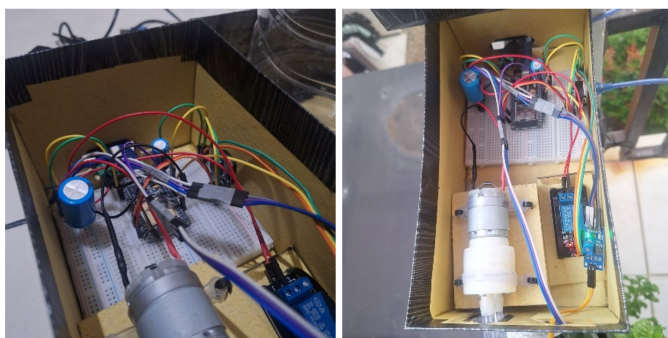
D. Skenario Verifikasi

Verifikasi dilakukan untuk memverifikasi bahwa algoritma yang berjalan di ESP32 menghasilkan keputusan yang identik dengan perhitungan manual menggunakan dataset yang sama. Pengujian mencakup tiga aspek: (1) klasifikasi delta per sesi penyiraman, (2) perubahan durasi berdasarkan modus klasifikasi, dan (3) pembaruan jadwal berdasarkan rata-rata M_{norm} . Selain data riil dari tiga hari pengumpulan (*cycle* 4), pengujian tambahan dilakukan menggunakan data sintesis untuk memverifikasi ketahanan algoritma terhadap variasi kondisi di luar rentang data aktual.

III. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Sistem irigasi berhasil diimplementasikan menggunakan ESP32 sebagai perangkat *edge* yang melakukan pembacaan sensor, pengendalian penyiraman, dan pencatatan data historis secara lokal. Data hasil pembacaan sensor ditampilkan melalui *dashboard* ThingsBoard untuk *monitoring* dan visualisasi secara *real-time*, sedangkan seluruh pengambilan keputusan tetap dilakukan secara lokal pada ESP32. Implementasi perangkat keras ditunjukkan pada Gbr. 9 hingga Gbr. 13.



Gbr. 9 Hardware Sistem Irigasi



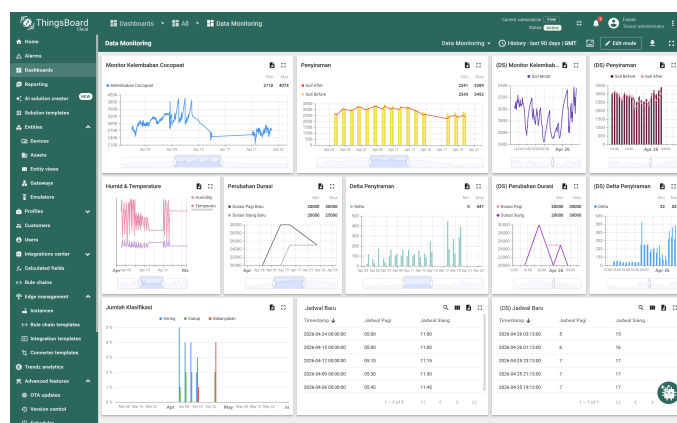
Gbr. 10 Penempatan Vertical Garden



Gbr. 11 Sistem Penyiraman



Gbr. 12 Vertical Garden



Gbr. 13 Dashboard Monitoring

B. Pengujian dan Kalibrasi Sensor

Kalibrasi sensor YL-69 menghasilkan nilai $S_{wet} = 1333$ pada kondisi jenuh air dan $S_{dry} = 3184$ pada kondisi kering setelah 24 jam, dengan rentang kerja sensor sebesar 1851. Analisis tren kelembaban dari 99 data menunjukkan rata-rata M_{norm} sebesar 0,15, yang mengindikasikan bahwa *cocopeat*

yang dibiarkan mengering dengan sekali penyiraman per hari dominan berada dalam kondisi kering. Analisis kurva pengeringan mengonfirmasi fase transisi pada rentang 0,47–0,26 M_{norm} , yang selanjutnya digunakan sebagai *threshold* pembaruan jadwal penyiraman.

C. Perbandingan Kalkulasi Sistem dan Manual

Pengujian verifikasi dilakukan menggunakan data *cycle* 4 dengan membandingkan hasil komputasi ESP32 terhadap perhitungan manual. Hasil agregasi modus klasifikasi selama tiga hari menunjukkan kondisi pagi: CUKUP dan siang: KERING, sehingga durasi penyiraman siang ditambah +5 detik. Nilai rata-rata kelembaban tanah menghasilkan M_{norm} global = 0,117, M_{norm} pagi = 0,094, dan M_{norm} siang = 0,122 — seluruhnya di bawah THRESHOLD_KERING (0,26) sehingga jadwal dimajukan. Hasil perubahan jadwal ditunjukkan pada Tabel V.

TABEL V
HASIL PERUBAHAN JADWAL PENYIRAMAN (CYCLE 4)

Parameter	Sebelum	Sesudah
Jadwal Pagi	05:10	05:00
Jadwal Siang	11:15	11:00
Durasi Pagi	30.000 ms	30.000 ms
Durasi Siang	20.000 ms	25.000 ms

Sistem menerapkan *safety limit* (pagi: 05:00–09:00; siang: 11:00–15:00) sebagai *guardrail* untuk mencegah pergeseran jadwal yang ekstrem. Hasil verifikasi menunjukkan bahwa seluruh keputusan klasifikasi, perubahan durasi, dan pembaruan jadwal yang dihasilkan ESP32 identik dengan hasil perhitungan manual. Progres adaptasi sistem selama lima *cycle* pengujian data riil ditunjukkan pada Tabel VI.

TABEL VI
PROGRES ADAPTASI SISTEM DARI DATA RIIL (5 CYCLE)

Cycle	Jadwal Pagi	Jadwal Siang	Durasi Pagi (ms)	Durasi Siang (ms)
Awal	06:00	12:00	15.000	15.000
1	05:45	11:45	20.000	20.000
2	05:30	11:30	25.000	20.000
3	05:10	11:15	30.000	20.000
4	05:00	11:00	30.000	25.000
5	05:00	11:00	25.000	20.000

D. Uji Adaptasi dengan Data Sintesis

Pengujian tambahan menggunakan data sintesis dilakukan untuk mengevaluasi respons algoritma terhadap berbagai kondisi yang dikontrol, mencakup kekeringan, saturasi air, gangguan sensor, dan konflik data. Siklus sintesis ditetapkan selama 120 menit dengan dua penyiraman setiap 40 menit dan pembacaan sensor setiap 5 menit. Mengingat durasi siklus uji yang singkat, perubahan waktu jadwal dibatasi pada *offset* maksimal 1 menit sebagai pengganti *offset* 15 menit yang digunakan pada data aktual. Hasil verifikasi *cycle* terakhir ditunjukkan pada Tabel VII, sedangkan progres adaptasi selama tujuh *cycle* ditunjukkan pada Tabel VIII.

TABEL VII
HASIL JADWAL BARU DATA SINTESIS (CYCLE 5)

Parameter	Sebelum	Sesudah
Jadwal Pagi	7 menit	7 menit
Jadwal Siang	17 menit	17 menit
Durasi Pagi	25.000 ms	25.000 ms
Durasi Siang	25.000 ms	20.000 ms

TABEL VIII
PROGRES ADAPTASI SISTEM DARI DATA SINTESIS (7 CYCLE)

Cycle	Jadwal Pagi (min)	Jadwal Siang (min)	Durasi Pagi (ms)	Durasi Siang (ms)
Awal	10	20	15.000	15.000
1	9	19	20.000	20.000
2	8	18	25.000	25.000
3	7	17	30.000	30.000
4	7	17	25.000	25.000
5	7	17	25.000	20.000
6	6	16	25.000	25.000
7	5	15	20.000	20.000

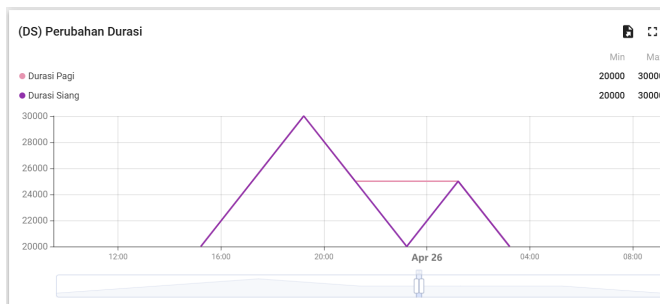
Nilai M_{norm} data sintesis (global = 0,31; pagi = 0,31; siang = 0,35) berada dalam rentang normal (0,26–0,47) sehingga jadwal tidak bergeser. Durasi siang berkurang karena modus klasifikasi menunjukkan KEBANYAKAN. Hasil verifikasi manual sepenuhnya selaras dengan *output* sistem, mengonfirmasi bahwa algoritma berperilaku konsisten pada kondisi di luar rentang data aktual.

E. Pembahasan

Sistem irigasi adaptif yang dikembangkan berhasil memenuhi ketiga tujuan penelitian. Pertama, sistem terbukti mampu beroperasi sepenuhnya sebagai perangkat *edge* tanpa ketergantungan pada layanan *cloud*. Kedua, algoritma *if-then* berhasil mengadaptasi durasi dan jadwal penyiraman secara bertahap berdasarkan data historis kelembaban tanah. Perubahan durasi pagi mencapai 30.000 ms sebelum menyesuaikan kembali seperti ditunjukkan pada Gbr. 14, sedangkan jadwal pagi dan siang secara bertahap dimajukan dari 05:45 dan 11:45 hingga batas minimum 05:00 dan 11:00. Pada data sintesis, algoritma menunjukkan respons yang lebih dinamis mengikuti variasi kondisi yang dikontrol, dengan durasi berfluktuasi antara 20.000–30.000 ms sesuai klasifikasi dominan per *cycle*, seperti ditunjukkan pada Gbr. 15. Ketiga, verifikasi pada data riil (*cycle* 4) dan data sintesis menunjukkan kesesuaian penuh antara hasil perhitungan ESP32 dan simulasi manual.



Gbr. 14 Grafik Perubahan Durasi



Gbr. 15 Grafik Perubahan Durasi Data Sintesis

Keterbatasan penelitian ini mencakup penggunaan satu titik sensor YL-69 yang belum sepenuhnya merepresentasikan distribusi kelembaban seluruh media tanam, serta belum adanya evaluasi dampak terhadap pertumbuhan tanaman secara langsung. Selain itu, sensor resistif YL-69 memiliki kelemahan terhadap korosi jangka panjang dibandingkan sensor kapasitif [15]. Nilai *threshold* yang diperoleh bersifat *site-specific* sehingga memerlukan kalibrasi ulang apabila diterapkan pada media tanam atau sensor yang berbeda.

IV. KESIMPULAN

Sistem irigasi otomatis berbasis IoT untuk *vertical garden* menggunakan ESP32 sebagai perangkat *edge* dengan mekanisme *adaptive scheduling* berbasis aturan *if-then* telah dikembangkan. Pembacaan sensor, pengendalian penyiraman, dan pencatatan data historis terbukti dapat dilakukan secara lokal tanpa ketergantungan pada layanan *cloud*. Durasi dan jadwal penyiraman disesuaikan secara bertahap berdasarkan klasifikasi delta kelembaban tanah dan nilai rata-rata M_{norm} dari data historis selama tiga hari. Hasil verifikasi pada data riil (*cycle* 4) dan data sintesis menunjukkan kesesuaian penuh antara keputusan klasifikasi, perubahan durasi, dan pembaruan jadwal yang dihasilkan ESP32 dengan hasil perhitungan manual, sehingga algoritma *adaptive scheduling* yang diusulkan mampu mendukung mekanisme penjadwalan penyiraman berbasis data historis pada sistem irigasi *vertical garden*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Agus Prihanto, S.T., M.Kom. selaku dosen pembimbing atas bimbingan dan arahan selama pelaksanaan penelitian ini, serta kepada Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya atas dukungan fasilitas dan akademis yang diberikan.

REFERENSI

- [1] S. Meriläinen and A. Tervo, "The changing typology of urban apartment buildings in Aurinkolahti," *Build. Cities*, vol. 6, no. 1, pp. 50–69, 2025, doi: 10.5334/bc.484.
- [2] M. S. Jahan, J. kabir Mim, S. Niittyviita, S. Moberg, M. Ahmad, and N. Hossain, "IoT Solution for Winter Survival of Indoor Plants," 2021, [Online]. Available: <http://arxiv.org/abs/2106.05130>
- [3] I. Ullah *et al.*, "Understanding Botrytis cinerea infection and gray mold management: a review paper on deciphering the rose's thorn," *Phytopathol. Res.*, vol. 6, no. 1, 2024, doi: 10.1186/s42483-024-00262-9.
- [4] A. S. M. Morsy and H. M. Mehanna, "Beneficial effects of antitranspirants on water stress tolerance in maize under different plant densities in newly reclaimed land," *Bull. Natl. Res. Cent.*, vol. 46, no. 1, 2022, doi: 10.1186/s42269-022-00934-6.
- [5] S. F. Ahmed, S. Shuravi, S. Afrin, S. J. Rafa, M. Hoque, and A. H. Gandomi, "The Power of Internet of Things (IoT): Connecting the Dots with Cloud, Edge, and Fog Computing," 2023, [Online]. Available: <http://arxiv.org/abs/2309.03420>
- [6] M. S. Farooq, R. Javid, S. Riaz, and Z. Atal, "IoT Based Smart Greenhouse Framework and Control Strategies for Sustainable Agriculture," *IEEE Access*, vol. 10, no. September, pp. 99394–99420, 2022, doi: 10.1109/ACCESS.2022.3204066.
- [7] L. Han, "Application of Lianas in Vertical Greening Landscape of Xi'an City," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 768, no. 1, 2021, doi: 10.1088/1755-1315/768/1/012155.
- [8] M. Benzaouia, B. Hajji, A. Mellit, and A. Rabhi, "Fuzzy-IoT smart irrigation system for precision scheduling and monitoring," *Comput. Electron. Agric.*, vol. 215, no. September, p. 108407, 2023, doi: 10.1016/j.compag.2023.108407.
- [9] N. Jaiswal, T. V. Kumar, and C. Shukla, "Smart drip irrigation systems using IoT: a review of architectures, machine learning models, and emerging trends," *Discov. Agric.*, vol. 3, no. 1, pp. 1–24, 2025, doi: 10.1007/s44279-025-00430-1.
- [10] P. Yu, F. Teng, W. Zhu, C. Shen, Z. Chen, and J. Song, "Cloud-edge-device collaborative computing in smart agriculture: architectures, applications, and future perspectives," *Front. Plant Sci.*, vol. 16, no. October, pp. 1–24, 2025, doi: 10.3389/fpls.2025.1668545.
- [11] S. Premkumar and A. N. Sigappi, "IoT-enabled edge computing model for smart irrigation system," *J. Intell. Syst.*, vol. 31, no. 1, pp. 632–650, 2022, doi: 10.1515/jisys-2022-0046.
- [12] Y. E. Kunt, "Development of a Smart Autonomous Irrigation System Using Iot and AI," pp. 1–16, 2025, [Online]. Available: <http://arxiv.org/abs/2506.11835>
- [13] G. P. Pereira, M. Z. Chaari, and F. Daroge, "IoT-Enabled Smart Drip Irrigation System Using ESP32," *Internet of Things*, vol. 4, no. 3, pp. 221–243, 2023, doi: 10.3390/iot4030012.
- [14] B. Vaishnavi, D. Nikhitha, K. L. Samprithi, and M. V. Mood, "ESP32-Based Real Time Soil Weather Monitoring and Smart Irrigation System in Urban Farming," *Jcsjournal.Com*, vol. 13, no. 02, pp. 243–247, 2025, [Online]. Available: <https://jcsjournal.com/admin/uploads/ESP32-Based Real Time Soil Weather Monitoring and Smart Irrigation System in Urban Farming.pdf>
- [15] A. A. Abdelmoneim, C. M. Al Kalaany, G. Dragonetti, B. Derardja, and R. Khadra, "Comparative Analysis of Soil Moisture- and Weather-Based Irrigation Scheduling for Drip-Irrigated Lettuce Using Low-Cost Internet of Things Capacitive Sensors," *Sensors*, vol. 25, no. 5, 2025, doi: 10.3390/s25051568.
- [16] R. Palliser-Sans, "RRULES: An improvement of the RULES rule-based classifier," 2021, [Online]. Available: <http://arxiv.org/abs/2106.07296>
- [17] M. Yantidewi, B. Jatmiko, I. Sucahyo, A. Kholiq, N. A. Lestari, and U. A. Deta, "Getting to know the last five years trend on microcontroller research: A bibliometric analysis," *J. Phys. Conf. Ser.*, vol. 2110, no. 1, 2021, doi: 10.1088/1742-6596/2110/1/012006.
- [18] R. Rahman and F. B. Gultom, "Desain Dan Perancangan Instrumen Monitoring Kekeruhan Air Dengan Sistem Real Time Clock (Rtc)," *J. Kumparan Fis.*, vol. 5, no. 1, pp. 23–30, 2022, doi: 10.33369/jkf.5.1.23-30.
- [19] J. Bayer, M. Zadeh, M. Schröder, and A. Dengel, "Functional Component Descriptions for Electrical Circuits based on Semantic Technology Reasoning," pp. 528–532, 2022, doi: 10.5220/0011322000003269.
- [20] R. Rehpade, G. Kharate, and J. Chopade, "Analysis of Decoupling Capacitor Performance in Improving Power Integrity in Two Layer/Three Layer Printed Circuit Boards," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 6, pp. 34–37, 2023, doi: 10.24018/ejece.2023.7.6.575.
- [21] E. Raj, M. Westerlund, and L. Espinosa-Leal, "Reliable Fleet Analytics for Edge IoT Solutions," 2021, [Online]. Available: <http://arxiv.org/abs/2101.04414>
- [22] G. Rong, Y. Xu, X. Tong, and H. Fan, "An edge-cloud collaborative computing platform for building AIoT applications efficiently," *J. Cloud Comput.*, vol. 10, no. 1, 2021, doi: 10.1186/s13677-021-00250-w.
- [23] R. Giffreda and *et al.*, "Edge and cloud computing for industrial internet of things: A review," *ACM Trans. Internet Things*, vol. 1, no. 1, pp. 1–25, 2024.
- [24] S. Dubey, "Cloud-Edge Architecture for IoT in Smart Building Automation Citation," *Comput. Eng. Technol. Innov.*, vol. 1, no. 3, pp. 178–186, 2024, [Online]. Available: www.ceti.reapress.com