

RANCANG BANGUN APLIKASI *ANDROID* UNTUK *MONITORING* DAN KONTROL IRIGASI OTOMATIS TANAMAN BROKOLI BERBASIS ESP32

Anggi Amalia^{1*}, Faisal Candrasyah Hasibuan², Rifqi Muhammad Fikri³

^{1, 2, 3}Universitas Telkom

aangggew@student.telkomuniversity.ac.id¹, faicanhasfcb@telkomuniversity.ac.id², rifmff@telkomuniversity.ac.id³

Abstrak

Penelitian ini membahas pengembangan sistem *monitoring* dan kontrol irigasi otomatis berbasis *Internet of Things* (IoT) untuk budidaya tanaman brokoli. Sistem ini menggunakan mikrokontroler ESP32 yang terhubung dengan dua sensor kelembapan tanah, sensor pH tanah, dan sensor pH air, yang mengirimkan data secara *real-time* ke *Firestore Realtime Database*. Data divisualisasikan melalui aplikasi *Android* yang dilengkapi fitur *dashboard*, grafik monitoring harian dan mingguan, kontrol manual dan otomatis katup solenoid, pencatatan riwayat penyiraman, serta notifikasi otomatis. Pengujian lapangan dilakukan pada dua petak tanaman brokoli seluas $\pm 20 \text{ m}^2$ dari total lahan petani $\pm 2.500 \text{ m}^2$. Hasil pengujian pada lahan uji menunjukkan bahwa sistem mampu menjaga kelembapan tanah pada kisaran 60–70%, menurunkan potensi tanaman tidak tumbuh optimal dibanding metode penyiraman manual, serta membantu efisiensi penggunaan air pada area yang diuji. Pengujian *white-box* dan *black-box* memastikan seluruh fungsi berjalan sesuai rancangan dan dapat diakses dengan baik oleh pengguna. Sistem ini berpotensi menjadi solusi pendukung bagi petani dalam memantau kondisi lahan brokoli dan mengoptimalkan proses penyiraman pada skala kecil yang dapat dikembangkan lebih lanjut untuk skala lahan yang lebih luas.

Kata kunci: Irigasi, IoT, ESP32, Kelembapan, pH, *Android*

I. PENDAHULUAN

Pertanian brokoli yang masih menggunakan sistem irigasi manual menghadapi berbagai kendala, terutama dalam menjaga konsistensi kelembapan tanah dan efisiensi penggunaan air[1]. Ketidakteraturan dalam penyiraman, keterbatasan pemantauan secara *real-time*, serta ketergantungan pada aktivitas manual menyebabkan proses irigasi menjadi kurang efektif dan berdampak pada pertumbuhan tanaman serta hasil panen.

Dalam menjawab tantangan tersebut, teknologi *Internet of Things* (IoT) menawarkan solusi melalui integrasi antara sensor lingkungan, mikrokontroler, serta aplikasi pemantauan yang memungkinkan proses irigasi berjalan secara otomatis dan dapat dikendalikan dari jarak jauh. Salah satu pendekatan yang diterapkan dalam penelitian ini adalah dengan mengembangkan sistem irigasi otomatis berbasis mikrokontroler ESP32, yang terhubung dengan sensor kelembapan dan pH tanah maupun air, serta terintegrasi dengan *Firestore Realtime Database* sebagai media komunikasi data antara perangkat dan aplikasi *Android*.

Permasalahan utama yang diangkat dalam penelitian ini adalah ketiadaan sistem otomatisasi irigasi yang mampu menyesuaikan penyiraman berdasarkan kondisi lingkungan aktual secara *real-time*, serta memberikan antarmuka yang mudah diakses oleh pengguna. Sistem ini diharapkan dapat meningkatkan efisiensi kerja petani dan mengurangi risiko kesalahan penyiraman secara manual, dengan dukungan antarmuka pengguna (UI) yang sederhana dan pengalaman

pengguna (UX) yang intuitif. Untuk mewujudkan hal tersebut, penelitian ini dilakukan melalui tahapan analisis kebutuhan, perancangan sistem, implementasi perangkat keras dan lunak, integrasi dengan *platform cloud Firestore*, serta pengujian sistem menggunakan metode *White Box*, *Black Box*, *Integration Test*, dan *Stress Test*. Seluruh proses dirancang untuk menghasilkan sistem yang tidak hanya berfungsi secara teknis, tetapi juga stabil, akurat, dan mudah digunakan di lapangan.

II. KAJIAN TEORI

A. *Internet of Things* (IoT)

Internet of Things (IoT) adalah konsep yang memungkinkan perangkat fisik saling terhubung melalui internet untuk saling bertukar data[2]. Dalam sistem ini, sensor dan aktuator dapat dikendalikan dan dimonitor secara *real-time*. IoT berperan penting dalam sistem irigasi otomatis dengan memungkinkan integrasi data sensor ke dalam aplikasi berbasis *cloud*, sehingga pengguna dapat memantau dan mengontrol perangkat dari jarak jauh.

B. Mikrokontroler ESP32

ESP32 merupakan mikrokontroler 32-bit dengan konektivitas Wi-Fi dan *Bluetooth* yang sering digunakan dalam pengembangan sistem IoT[3]. Dalam penelitian ini, ESP32 berfungsi sebagai pusat kendali yang mengumpulkan

data dari sensor dan mengirimkannya ke *Firestore Realtime Database*, serta menerima perintah dari aplikasi untuk mengontrol pompa dan katup solenoid.

C. *Firestore Realtime Database*

Firestore Realtime Database adalah layanan penyimpanan *cloud* dari *Google* yang memungkinkan sinkronisasi data secara *real-time* antara perangkat dan aplikasi[4]. Platform ini digunakan dalam penelitian sebagai media penyimpanan dan komunikasi antar ESP32 dan aplikasi *Android*, untuk memastikan data sensor dapat ditampilkan dan diperbarui secara langsung oleh pengguna.

D. UI/UX (*User Interface* dan *User Experience*)

Desain antarmuka pengguna (UI) dan pengalaman pengguna (UX) merupakan aspek penting dalam pengembangan aplikasi, terutama dalam sistem monitoring dan kontrol[5]. UI yang responsif dan UX yang intuitif dapat mempermudah petani dalam mengakses data sensor, mengontrol penyiraman secara manual, serta menerima notifikasi terkait kondisi lingkungan lahan.

E. Metode Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan perangkat lunak dan perangkat keras bekerja sesuai fungsi. Beberapa metode pengujian yang digunakan adalah *White Box* untuk menguji logika program, *Black Box* untuk menguji *input-output* sistem, *Integration Test* untuk menguji integrasi komponen, dan *Stress Test* untuk menguji kestabilan sistem dalam jangka waktu panjang[6].

III. METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan model pengembangan berbasis kebutuhan (*requirement-driven development*). Tujuan utamanya adalah merancang dan mengimplementasikan aplikasi monitoring dan kontrol irigasi otomatis berbasis sensor kelembapan tanah dan pH yang dapat menampilkan data secara *real-time* serta mendukung pengendalian katup secara manual maupun otomatis. Adapun tahapan metode yang dilakukan meliputi:

A. Analisis Kebutuhan Sistem

Proses analisis dilakukan untuk mengidentifikasi kebutuhan pengguna dan permasalahan di lapangan, khususnya pada penyiraman tanaman brokoli yang tidak

merata. Data diperoleh melalui wawancara petani, observasi langsung di lahan, dan studi literatur terkait sistem monitoring berbasis IoT. Hasil analisis digunakan untuk menyusun daftar kebutuhan fungsional dan non-fungsional aplikasi.

B. Perancangan Sistem

Perancangan dilakukan dengan membuat *flowchart*, *entity relationship diagram* (ERD), dan desain antarmuka pengguna berdasarkan rancangan awal di Figma. Sistem dibagi menjadi dua bagian utama: modul monitoring (menampilkan data kelembapan tanah 1 & 2, pH tanah, dan pH air, serta grafik harian & mingguan) dan modul kontrol (mengaktifkan/menonaktifkan katup solenoid secara manual atau otomatis).

C. Implementasi Aplikasi

Aplikasi dikembangkan menggunakan *Android Studio* dengan bahasa pemrograman Java/Kotlin untuk *frontend*. Data sensor dikirimkan oleh mikrokontroler ESP32 ke *Firestore Realtime Database*, yang kemudian diambil dan ditampilkan pada aplikasi. Sistem mendukung fitur riwayat penyiraman (*log watering*), notifikasi otomatis, serta pengaturan ambang batas penyiraman.

D. Integrasi Perangkat dan Aplikasi

ESP32 diprogram untuk membaca sensor kelembapan tanah dan pH, mengirimkan data ke *Firestore* secara berkala dengan *timestamp* NTP, dan mengontrol katup solenoid berdasarkan ambang batas yang diatur di aplikasi. Aplikasi dan perangkat saling terhubung melalui *Firestore* untuk pertukaran data *real-time*.

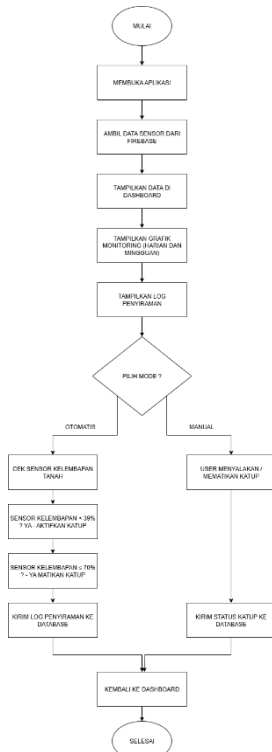
E. Pengujian Sistem

Pengujian dilakukan dengan dua metode. *White-box testing* digunakan untuk memverifikasi logika program pada mikrokontroler dan aplikasi, termasuk proses pengambilan data dari *Firestore* dan pengendalian katup. *Black-box testing* digunakan untuk memastikan semua fungsi aplikasi berjalan sesuai kebutuhan pengguna, seperti tampilan data sensor, kontrol katup, dan notifikasi.

F. Visualisasi Hasil Monitoring

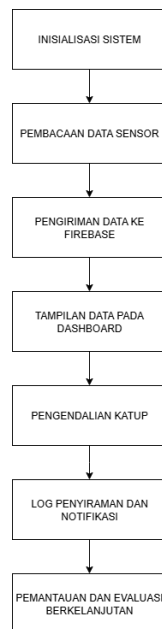
Hasil pembacaan sensor divisualisasikan dalam bentuk angka dan grafik harian maupun mingguan di aplikasi. Selain itu, pengguna dapat melihat riwayat penyiraman dan menerima notifikasi jika kelembapan tanah berada di bawah batas normal.

G. Flowchart dan alur kerja aplikasi



GAMBAR 1.
(FLOWCHART APLIKASI)

Flowchart dibuat sebagai representasi visual yang menggambarkan seluruh proses kerja aplikasi monitoring dan kontrol irigasi otomatis. Diagram ini mencakup mulai dari pembacaan data sensor kelembapan tanah dan pH, pengiriman data ke *Firestore*, pemrosesan data di aplikasi, hingga pengendalian katup secara manual maupun otomatis.



GAMBAR 2.
(ALUR KERJA APLIKASI)

Alur kerja aplikasi :

- 1) **Inisialisasi Sistem**
Saat aplikasi dijalankan, sistem melakukan inisialisasi koneksi ke *Firestore Realtime Database* untuk memastikan sinkronisasi data *real-time* antara perangkat mikrokontroler ESP32 dan aplikasi.
- 2) **Pembacaan Data Sensor**
ESP32 secara berkala membaca nilai sensor kelembapan tanah (dua titik sensor) dan sensor pH (tanah dan air). Data yang terbaca disimpan dengan *timestamp* menggunakan NTP (*Network Time Protocol*) untuk keakuratan waktu.
- 3) **Pengiriman Data ke Firestore**
Data sensor yang telah dibaca dikirim ke *Firestore Realtime Database* secara berkala. Data ini kemudian dapat diakses oleh aplikasi untuk ditampilkan *secara real-time* kepada pengguna.
- 4) **Tampilan Data pada Dashboard**
Aplikasi menampilkan data sensor terbaru dalam bentuk angka serta grafik monitoring harian dan mingguan yang memudahkan petani memantau kondisi tanah dan air.
- 5) **Pengendalian Katup Solenoid**
Petani dapat mengaktifkan atau menonaktifkan katup secara manual melalui tombol kontrol pada aplikasi. Selain itu, sistem juga secara otomatis mengendalikan katup berdasarkan ambang batas kelembapan tanah yang sudah diatur.
- 6) **Log Penyiraman dan Notifikasi**
Setiap aktivitas penyiraman, baik manual maupun otomatis, dicatat dalam log riwayat penyiraman (*watering logs*) yang dapat dilihat oleh pengguna. Sistem juga mengirimkan notifikasi otomatis jika kelembapan tanah berada di bawah batas yang telah ditentukan untuk mengingatkan pengguna melakukan penyiraman.
- 7) **Pemantauan dan Evaluasi Berkelanjutan**
Pengguna dapat terus memantau data dan riwayat penyiraman melalui aplikasi, sehingga dapat melakukan evaluasi terhadap efektivitas irigasi.

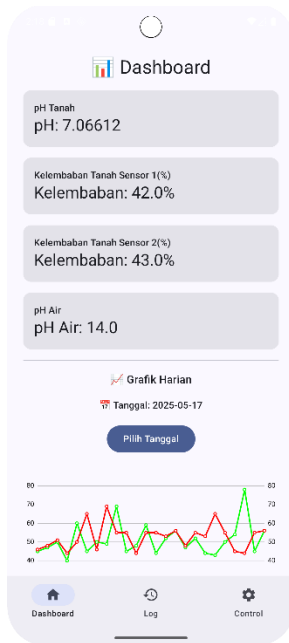
IV. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah aplikasi monitoring dan kontrol irigasi otomatis berbasis *Android* yang terintegrasi dengan mikrokontroler ESP32 dan sensor kelembapan tanah serta pH. Sistem ini dirancang untuk membantu petani dalam menjaga kondisi lahan brokoli agar tetap optimal dengan memantau data secara *real-time* dan mengontrol katup penyiraman baik secara otomatis maupun manual melalui aplikasi.

A. Implementasi Sistem Monitoring dan Kontrol

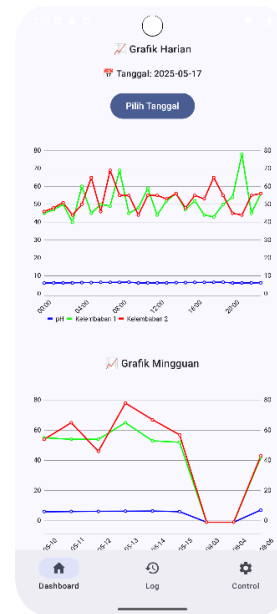
Aplikasi dikembangkan menggunakan *Android Studio* dengan bahasa pemrograman Kotlin, terhubung ke *Firestore Realtime Database* untuk menerima dan mengirim data secara *real-time*. Mikrokontroler ESP32 membaca data dari dua sensor kelembapan tanah, sensor pH tanah, dan sensor pH air, kemudian mengirimkannya ke *Firestore* menggunakan protokol HTTP.

Aplikasi menampilkan data tersebut dalam bentuk angka dan grafik (harian dan mingguan), serta menyediakan fitur kontrol dua katup solenoid yang dapat dioperasikan secara manual atau otomatis berdasarkan ambang batas kelembapan yang telah ditentukan pengguna.



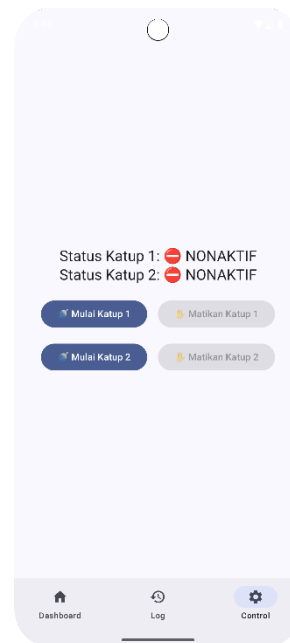
GAMBAR 3.
(TAMPILAN DASHBOARD UTAMA APLIKASI MONITORING)

Fitur *dashboard* menampilkan informasi kelembapan tanah 1, kelembapan tanah 2, pH tanah, dan pH air secara *real-time*. Pengguna juga dapat melihat status katup dan notifikasi sistem di halaman ini.



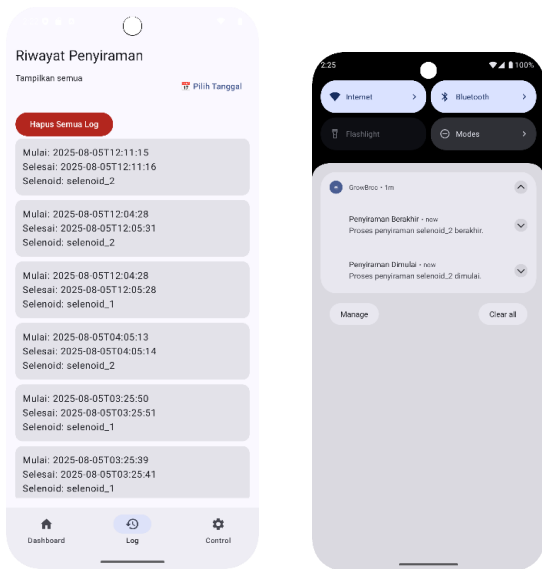
GAMBAR 4.
(TAMPILAN GRAFIK MONITORING HARIAN DAN MINGGUAN)

Grafik ini digunakan untuk melihat riwayat penyiraman selama satu hari dan satu minggu, sehingga petani dapat mengetahui kapan saja penyiraman dilakukan dan memastikan tanaman mendapat air yang cukup.



GAMBAR 5.
(TAMPILAN FITUR KONTROL MANUAL)

Kontrol manual memungkinkan petani melakukan penyiraman secara langsung melalui ponsel, sedangkan pada mode otomatis, sistem akan aktif ketika sensor kelembapan mendeteksi nilai antara 39–41 dan akan berhenti secara otomatis saat nilai kelembapan mencapai 70.



GAMBAR 6.
(TAMPILAN RIWAYAT PENYIRAMAN (*WATERING LOGS*) DAN
NOTIFIKASI OTOMATIS)

Riwayat penyiraman merekam data tanggal dan waktu pelaksanaan, termasuk informasi waktu mulai dan berhenti penyiraman, baik yang diaktifkan otomatis oleh sistem maupun secara manual oleh pengguna. Notifikasi dikirim saat proses penyiraman pada mode kontrol diaktifkan dan saat proses tersebut dihentikan.

B. Data Hasil Monitoring dan Kontrol

Pengujian lapangan dilakukan di dua petak tanaman brokoli seluas $\pm 20 \text{ m}^2$ dari total lahan $\pm 2.500 \text{ m}^2$. Sistem diuji selama periode pertumbuhan tanaman untuk memantau kelembapan tanah dan pH, sekaligus mengontrol penyiraman otomatis dan manual. Data pada Tabel 1 merupakan ringkasan hasil monitoring yang diperoleh selama pengujian.

TABEL 1.
(PERBANDINGAN KONDISI PENYIRAMAN DAN HASIL PANEN)

Parameter	Sebelum Sistem (Manual)	Sesudah Sistem (Otomatis + Monitoring)
Kelembapan Tanah (%)	42–78 (tidak merata)	60–69 (stabil)
Risiko Kekeringan	$\pm 35\%$ area lahan uji	$\pm 8\%$ area lahan uji
Risiko Kelebihan Air	$\pm 18\%$ area lahan uji	$\pm 4\%$ area lahan uji
Persentase Tanaman Kurang Optimal	$\pm 55\%$	$\pm 22\%$
Akses Data Monitoring	Tidak tersedia	<i>Real-time</i> via aplikasi <i>Android</i>

C. Pengujian Sistem

Pengujian *white-box* menunjukkan bahwa logika pembacaan data sensor, pengiriman data ke *Firestore*, dan kontrol katup sesuai dengan rancangan. *Black-box testing* memastikan semua fitur aplikasi dapat digunakan dengan baik, mulai dari menampilkan data sensor, mengatur ambang batas kelembapan, mengontrol katup, melihat riwayat penyiraman, hingga menerima notifikasi otomatis.

D. Dampak Terhadap Hasil Panen

Penggunaan sistem ini mampu menjaga kelembapan tanah di kisaran optimal 60–70%, sehingga mengurangi risiko gagal panen dari 50–70% menjadi sekitar 20–25%. Petani juga melaporkan penghematan waktu dan tenaga karena penyiraman dapat dipantau dan dikendalikan langsung melalui ponsel.

Hasil pengujian menunjukkan bahwa sistem bekerja sesuai dengan rancangan: data dari sensor kelembapan tanah dan pH dapat terbaca dan tersimpan secara otomatis di *Firestore*, nilai yang ditampilkan pada aplikasi konsisten dengan pembacaan langsung di lapangan, dan kontrol katup dapat dioperasikan baik secara otomatis maupun manual tanpa kendala. Visualisasi data monitoring dan riwayat penyiraman dapat diakses dengan baik oleh pengguna, sehingga memudahkan evaluasi kondisi lahan.

Petani memberikan tanggapan positif terhadap penggunaan aplikasi karena dapat memantau kondisi lahan dan mengendalikan penyiraman tanpa harus selalu berada di lokasi. Fitur notifikasi otomatis, grafik monitoring harian dan mingguan, serta pencatatan riwayat penyiraman dinilai membantu dalam menjaga kondisi kelembapan tanah tetap stabil dan mengurangi risiko gagal panen.

V. KESIMPULAN

Penelitian ini menghasilkan sebuah sistem monitoring dan kontrol irigasi otomatis berbasis *Android* yang terintegrasi dengan mikrokontroler ESP32, sensor kelembapan tanah, dan sensor pH. Sistem yang dikembangkan terbukti membantu petani dalam menjaga kondisi kelembapan tanah di lahan uji $\pm 20 \text{ m}^2$ pada kisaran optimal 60–69%, sehingga mengurangi potensi tanaman kurang optimal dibanding metode penyiraman manual.

Fitur yang diimplementasikan meliputi *dashboard* untuk pemantauan *real-time*, grafik monitoring harian dan mingguan, kontrol katup manual dan otomatis, pencatatan riwayat penyiraman lengkap dengan waktu mulai dan berhenti, serta notifikasi otomatis saat penyiraman diaktifkan atau dihentikan. Hasil pengujian *white-box* dan *black-box* menunjukkan bahwa seluruh fungsi sistem berjalan sesuai dengan rancangan, data sensor terbaca dan tersimpan secara konsisten, dan semua fitur dapat diakses dengan baik oleh pengguna.

Dengan demikian, sistem ini dapat menjadi solusi pendukung yang efektif untuk meningkatkan efisiensi penyiraman, menghemat waktu dan tenaga kerja, serta

meminimalkan risiko kehilangan hasil pada skala lahan kecil. Ke depan, sistem ini berpotensi dikembangkan untuk integrasi pengendalian berbasis prediksi cuaca, optimisasi konsumsi air, dan pengelolaan data berbasis *cloud analytics*, sehingga dapat diaplikasikan pada skala lahan yang lebih luas.

[4]

REFERENSI

- [1] 1] Acquaye, “*Operational research for sustainability: a synthesis of methods, applications and challenges*,” 2025, Taylor and Francis Ltd. doi: 10.1080/01605682.2025.2523362.
- [2] 2] J. J. Correa-Quiroz, M. A. Toribio-Barrueto, and C. Castro-Vargas, “*IoT System with ESP32 for Smart Drip Irrigation and Climate Monitoring in Greenhouses*,”

Emerging Science Journal, vol. 9, no. 3, pp. 1133–1157, Jun. 2025, doi: 10.28991/ESJ-2025-09-03-01.

3] D. Westari and S. Ilman, “Sistem Penyiraman Tanaman Otomatis Berbasis IoT Menggunakan ESP32, Moisture Sensor, DHT22 Sensor dan Blynk,” vol. 3, no. 4, pp. 314–321, 2024, doi: 10.55606/jtmei.v3i4.4941.

4] Y. E. Kunt, “*Development of a Smart Autonomous Irrigation System Using Iot and AI*,” Jun. 2025, [Online]. Available: <http://arxiv.org/abs/2506.11835>

5] Hanafi et al., “Perancangan Desain User Interface/User Experience: Systematic Literature Review,” 2025.

U. Aisyah Pringsewu et al., “*Aisyah Journal of Informatics and Electrical Engineering Internet of Things (IOT)*.” [Online]. Available: <http://jti.aisyahuniversity.ac.id/index.php/AJIEE>