

BAB IV

HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dari penelitian yang telah dilakukan serta analisis terhadap data yang diperoleh. Dalam bab ini, hasil pengujian sistem *smart farming* berbasis IoT dipaparkan, mencakup performa sensor, efektivitas algoritma pengolahan data, serta respon dan pengendalian perangkat IoT.

4.1 Pengujian Implementasi Sistem

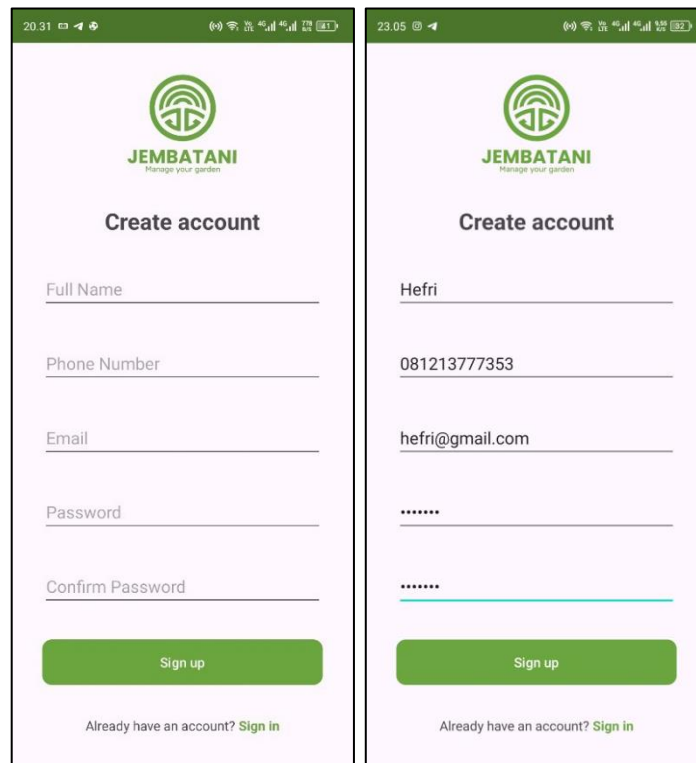
Berdasarkan perancangan yang sudah dilakukan sebelumnya, pada bagian ini menjelaskan secara rinci tentang bagaimana perangkat *smart farming* dapat merespon aplikasi yang dikembangkan dapat diimplementasikan dan diuji dalam situasi nyata. Diuraikannya hasil dari penerapan sistem, termasuk bagaimana aplikasi berfungsi, fitur-fitur utama berjalan, serta bagaimana data dikumpulkan, disimpan, dan diproses. Ini juga mencakup hasil pengujian yang menunjukkan kinerja sistem, responsivitas terhadap perintah, keakuratan data sensor yang dikumpulkan, serta sejauh mana sistem tersebut berhasil mencapai tujuan yang telah ditetapkan dalam desain awal.

4.1.1 Implementasi Sistem Akses Petani

Implementasi dari sistem yang sudah dirancang sebelumnya dengan hak akses petani disimpan pada *firebase realtime database* adalah sebagai berikut:

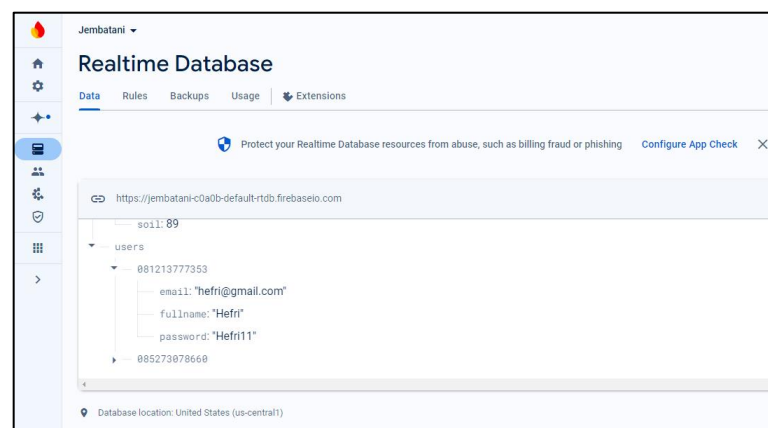
4.1.1.1 Implementasi Halaman Daftar Akun

Implementasi halaman daftar akun ini digunakan oleh petani sebagai pemilik lahan pertanian terdapat pada Gambar 4.1 . Petani harus melakukan pendaftaran untuk membuat akun, dengan mengisi biodata diri yang diminta pada halaman tersebut.



Gambar 4. 1 Halaman Daftar Akun

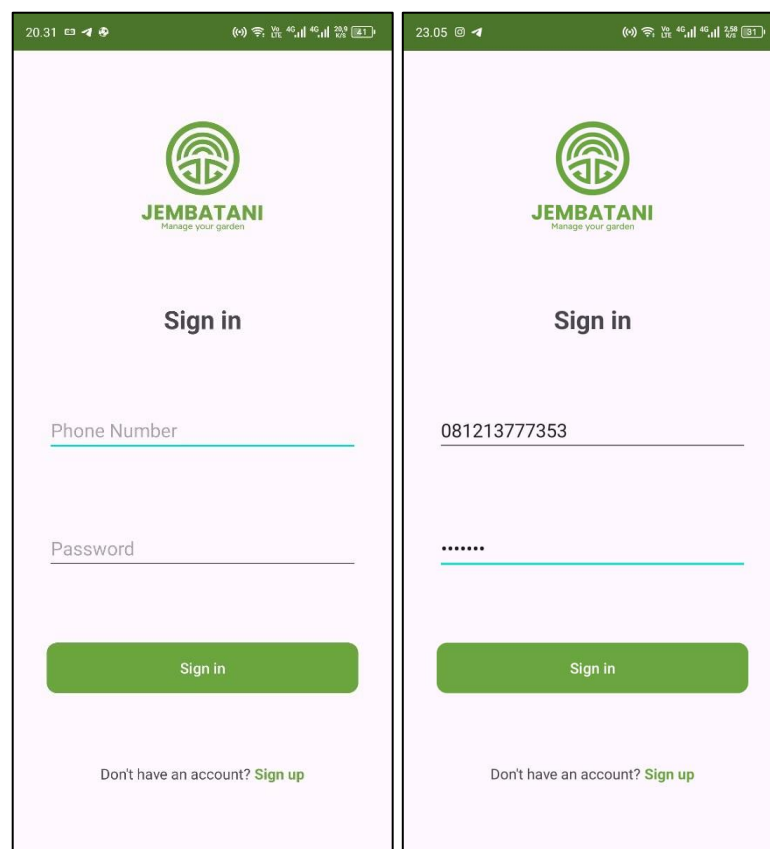
Setelah petani selesai mengisi data dirinya pada halaman daftar akun, dengan mengisi *fullname*, *phone number*, *email*, *password* dan *confirm password*. Data yang sudah terisi akan dikirim kepada *firebase realtime database* dan secara otomatis menyimpannya di dalam folder *users*, ketika petani menekan *button sign up*. Berikut Gambar 4.2 merupakan tampilan data yang sudah disimpan berdasarkan data yang sudah diisi oleh petani.



Gambar 4. 2 Database Daftar Akun

4.1.1.2 Implementasi Halaman Sign In

Setelah mendaftar akun, petani dapat melakukan *Sign In* pada halaman *Sign In* dengan masukan *phone number* dan *password* yang sudah didaftarkan. Selanjutnya aplikasi akan mengirimkan pesan *Successfully Logged In* sebagai indikator bahwa data yang digunakan terdaftar dan sesuai. Pada halaman ini terdapat tombol *sign up*, fungsi yang sebelumnya dilakukan untuk membuat akun. Implementasi halaman *sign in* akses akun petani terdapat pada Gambar 4.2.

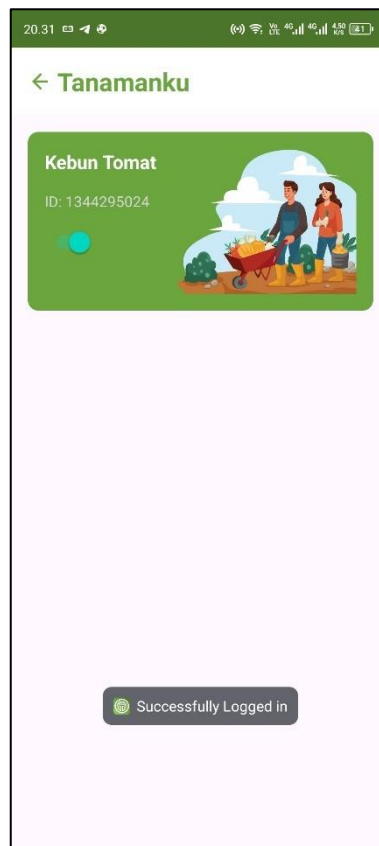


Gambar 4. 3 Halaman Sign In

4.1.1.3 Implementasi Halaman Menu

Ketika petani sudah berhasil melakukan langkah *sign in*, sistem akan menampilkan halaman menu. Halaman menu merupakan tampilan daftar kebun petani yang sudah aktif terintegrasi dengan perangkat IoT. Pada halaman ini terdapat ikon tombol “+” untuk menambahkan kebun yang akan diintegrasikan dengan perangkat IoT. Pada tampilan ini petani juga dapat masuk ke dalam halaman *dashboard* dengan

menekan kebun yang aktif pada halaman menu. Tampilan implementasi pada halaman ini dapat dilihat pada Gambar 4.3.



Gambar 4. 4 Halaman Menu

4.1.1.4 Implementasi Halaman Dashboard

Tampilan halaman *dashboard* berisi monitor lahan pertanian yang terdapat informasi detail tanaman yang sedang dilakukan manajemen pertumbuhan serta pemantauan lingkungan, pemantauan tanah dan pemantauan air, serta tombol untuk kontrol penyiraman.



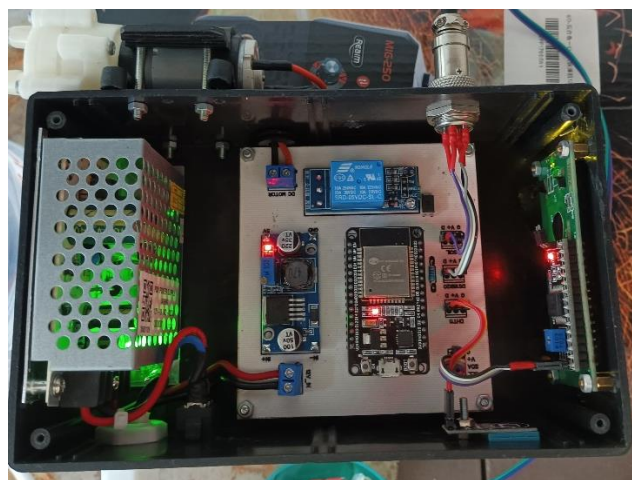
Gambar 4.5 Halaman Dashboard

4.1.2 Implementasi Monitoring Dan Penyiraman

Hasil implementasi sistem monitoring dan penyiraman pada pengembangan aplikasi Jembatani melibatkan penggunaan berbagai sensor IoT untuk mengumpulkan data serta mengendalikan penyiraman secara real-time.

4.1.2.1 Implementasi Sistem Monitoring

Pengujian implementasi perangkat keras monitoring dilakukan untuk memastikan bahwa perakitan komponen sensor dapat bekerja sesuai perancangan yang dilakukan dapat memberikan data yang selanjutnya ditampilkan pada aplikasi. Berikut hasil perancangan perangkat keras yang digunakan untuk mendapatkan data suhu, kelembaban, kelembaban tanah dan suhu tanah dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Alat Monitoring



Gambar 4. 7 Display Hasil Pengukuran Sensor

Berdasarkan Gambar 4.6 bentuk fisik alat *smart farming* dapat digunakan untuk mengirimkan data sensor secara realtime. Fitur ini memungkinkan petani dapat memonitor kondisi lahan dari jarak jauh, sistem ini mencakup sensor yang

mengukur berbagai parameter seperti suhu tanah, kelembaban tanah, suhu udara, dan kelembaban lingkungan. Dengan tampilan monitor pada aplikasi Jembatani bisa dilihat pada Gambar 4.8.

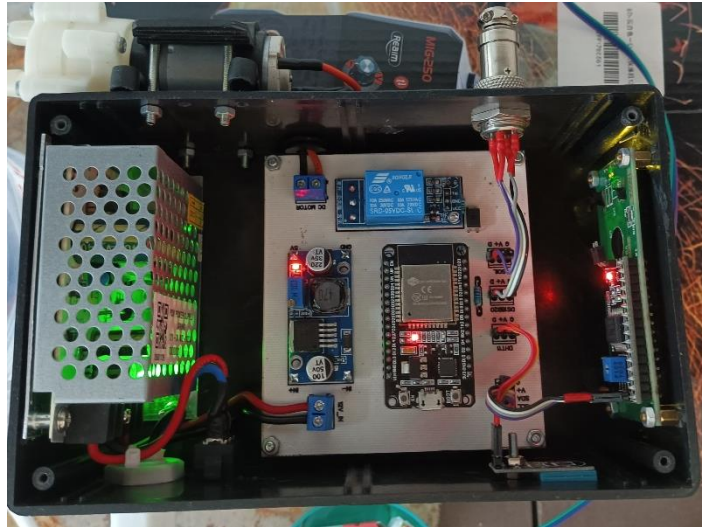


Gambar 4. 8 Tampilan Sistem Monitoring Aplikasi

Berdasarkan gambar 4.8 hasil uji coba tampilan pada aplikasi, dapat dinyatakan bahwa aplikasi Jembatani dapat berjalan dengan baik untuk menampilkan hasil sensor yang sebelum dikirim oleh alat melalui *firebase*. Diketahui aplikasi dapat menampilkan hasil pembacaan sensor DHT11 (Kelembaban 62% dan Temperatur 32%), sensor DS18B20 (Temperatur Tanah 29%) dan *Soil Moisture* (Kelembaban Tanah 60%).

4.1.2.2 Implementasi Sistem Penyiraman

Menyajikan pengujian hasil dari implementasi sistem penyiraman lahan pertanian, dengan menekan tombol penyiraman air pada aplikasi. Ini juga mencakup bagaimana aplikasi dapat menghidupkan pompa air secara real-time. Berikut hasil perancangan perangkat keras penyiraman air dapat dilihat pada Gambar 4.9.



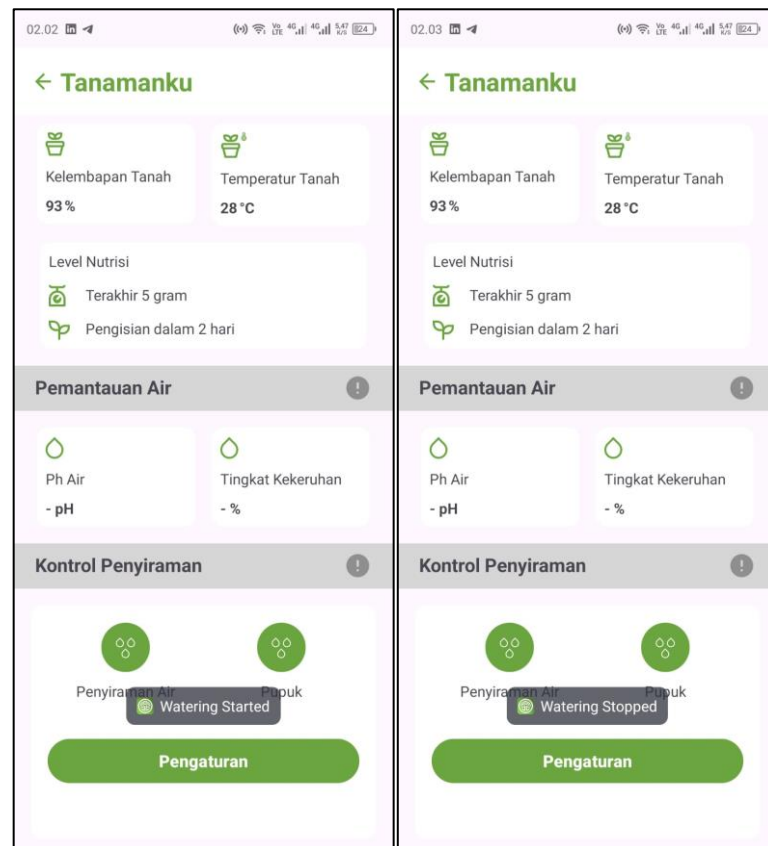
Gambar 4. 9 Alat Penyiraman Air



Gambar 4. 10 Display Status Relay

Berdasarkan Gambar 4.9 bentuk fisik alat penyiraman air dapat digunakan, dengan menghidupkan relay melalui tombol pada aplikasi. Fitur tombol penyiraman air pada aplikasi Jembatanani ini memungkinkan petani dapat melakukan penyiraman

lahan dari jarak jauh. Dengan tampilan penyiraman air pada aplikasi Jembatani bisa dilihat pada Gambar 4.11.



Gambar 4. 11 Fitur penyiraman air

Berdasarkan gambar 4.1 hasil uji coba penyiraman air pada aplikasi, dinyatakan bahwa aplikasi Jembatani dapat berjalan dengan baik untuk menghidupkan relay melalui *firebase*. Diketahui ketika menekan tombol dengan pesan *watering started*, maka aplikasi mengirimkan nilai 1 untuk menghidupkan *relay*. Dan sebaliknya ketika menekan tombol dengan pesan *watering stopped*, maka aplikasi mengirimkan nilai 0 untuk mematikan *relay*.

4.2 Pengujian Kecepatan dan Responsif

Pengujian kecepatan dan responsif pada aplikasi Jembatani bertujuan untuk memastikan bahwa sistem dapat menangani dan menampilkan data dari sensor IoT secara efisien dan cepat. Pengujian ini melibatkan pengukuran waktu yang dibutuhkan mulai dari pengiriman data oleh sensor, penerimaan dan penyimpanan

data pada *firebase realtime database*, hingga data tersebut muncul di antarmuka pengguna aplikasi. Berikut tabel pengujian kecepatan dan responsif sensor pada penelitian pengembangan aplikasi Jembatanani.

Tabel 4. 1 Hasil Pengujian Kecepatan dan Responsivitas Sensor

No	Parameter Uji	Metode Pengujian	Hasil Pengujian	Keterangan
1	Waktu Pengiriman Data	Mengukur waktu dari sensor mengirim data hingga diterima di Firebase	0 detik	Menggunakan stopwatch pada sensor dan Firebase
2	Waktu Pemrosesan di Firebase	Mengukur waktu Firebase memproses data hingga siap ditampilkan	1 detik	Waktu antara data diterima dan siap diproses
3	Waktu Tampilan di Aplikasi	Mengukur waktu dari data tersedia di Firebase hingga muncul di aplikasi	1 detik	Menggunakan log aplikasi dan Firebase
4	Latensi Jaringan	Mengukur total waktu dari pengiriman data oleh sensor hingga data tampil di aplikasi	2 detik (0+1+1)	Total waktu pengiriman, pemrosesan, dan tampilan
5	Responsivitas Aplikasi	Evaluasi seberapa cepat aplikasi merespon perubahan data	100% (selalu real-time) dengan koneksi internet	Respons aplikasi terhadap data real-time

Ketika melakukan pengujian kecepatan dan responsif penyiraman air pada aplikasi Jembatanani, terjadi penambahan delay hasil pengujian. Delay pada penyiraman air terjadi karena dalam perancangan ditambahkan *time interval*, agar perangkat dapat mengubah tampilan pada LCD display. Berikut adalah tabel penjelasan mengenai hasil pengujian penyiraman air.

Tabel 4. 2 Hasil Pengujian Kecepatan dan Responsivitas Akuator

No	Parameter Uji	Metode Pengujian	Hasil Pengujian	Keterangan
1	Waktu Pengiriman Perintah Akuator On	Mengukur waktu dari perintah dikirim hingga aktuator menyala	7 detik (2 detik + 5 detik delay)	Delay ditambahkan untuk simulasi kondisi lapangan
2	Waktu Pengiriman Perintah Akuator Off	Mengukur waktu dari perintah dikirim hingga aktuator mati	12 detik (2 detik + 10 detik delay)	Delay untuk memastikan siklus kerja selesai
3	Latensi Jaringan (Aktuator On/Off)	Mengukur waktu total dari pengiriman perintah hingga respon aktuator	7 detik untuk On, 12 detik untuk Off	Delay ditambahkan ke waktu jaringan
4	Responsivitas Akuator di Aplikasi	Evaluasi seberapa cepat aplikasi merespon perubahan status aktuator	Sesuai dengan hasil di atas	Hasil diperpanjang dengan delay simulasi

4.2.1 Perhitungan Simpangan Baku

Untuk mengetahui tingkat penyebaran data dari hasil pengujian, dilakukan perhitungan simpangan baku (“standard deviation”) sebagai berikut:

1. Data Hasil Pengujian

Data numerik diperoleh berdasarkan tabel 4.1 dengan nilai $x = \{0,1,1,2\}$.

2. Rumus Simpangan Baku

- x_i adalah nilai individu
- \bar{x} adalah rata-rata data
- n adalah jumlah data

3. Perhitungan:

- Hitung rata-rata (\bar{x}):

$$\bar{x} = \frac{\sum x_i}{n} = \frac{0 + 1 + 1 + 2}{4} = 1$$

- Hitung $(x_i - \bar{x})^2$ untuk setiap nilai:

$$(0 - 1)^2 = 1, (1 - 1)^2 = 0, (1 - 1)^2 = 0, (2 - 1)^2 = 1$$

- Jumlahkan hasil kuadrat:

$$\sum (x_i - \bar{x})^2 = 1 + 0 + 0 + 1 = 2$$

- Hitung simpangan baku:

$$S = \sqrt{\frac{2}{4 - 1}} = \sqrt{\frac{2}{3}} \approx 0,816$$

4. Hasil Simpangan Baku

Berdasarkan perhitungan simpangan baku berdasarkan data yang dimiliki pada penelitian adalah **0,816 detik**.

4.2.2 Analisis dan Penjelasan

Dari hasil perhitungan, simpangan baku sebesar 0,816 detik menunjukkan bahwa variasi waktu pengujian relatif kecil terhadap rata-rata, yang berarti sistem memiliki konsistensi yang cukup baik dalam pengiriman data, pemrosesan di Firebase, hingga tampilan di aplikasi. Namun, adanya perbedaan waktu pada parameter "Latensi Jaringan" (2 detik) memberikan kontribusi utama terhadap simpangan baku yang lebih besar dibandingkan jika hanya mempertimbangkan parameter lain. Adapun alasan terjadinya latensi jaringan 2 detik adalah:

1. Kualitas Jaringan Internet

Jaringan internet yang digunakan selama pengujian memengaruhi kecepatan pengiriman data ke Firebase dan pengembalian data ke aplikasi. Fluktuasi pada

jaringan, seperti kecepatan yang tidak stabil, dapat menyebabkan peningkatan waktu latensi.

2. Proses Komunikasi Antar Komponen

Latensi juga dapat disebabkan oleh proses pengolahan data dari sensor menuju Firebase dan kembali ke aplikasi melalui protokol komunikasi yang digunakan (REST API). Proses ini membutuhkan waktu lebih lama dibandingkan proses pengolahan lokal.

3. Server Firebase

Firestore sebagai platform cloud memiliki waktu pemrosesan tertentu yang dapat bervariasi tergantung pada beban server saat pengujian dilakukan.

4. Optimalisasi Sistem

Waktu latensi 2 detik dapat dioptimalkan lebih lanjut melalui peningkatan kecepatan komunikasi jaringan atau pengurangan beban data yang dikirimkan.

Meskipun hasil pengujian menunjukkan adanya latensi jaringan, simpangan baku yang relatif kecil mengindikasikan bahwa sistem secara keseluruhan mampu bekerja dengan konsistensi yang baik. Pengujian ini membuktikan bahwa integrasi antara sensor, Firestore, dan aplikasi telah berjalan dengan cukup efisien dalam kondisi pengujian yang ada.