



INNOVATIVE: Journal Of Social Science Research

Volume 5 Nomor 4 Tahun 2025 Page 12347-12365

E-ISSN 2807-4238 and P-ISSN 2807-4246

Website: <https://j-innovative.org/index.php/Innovative>

Penerapan Aplikasi Android Terintegrasi Firebase dalam Sistem IoT untuk Monitoring Kolam Lele Berbasis Realtime Database dan Firestore dengan Pengujian Blackbox

I Made Aditya Wirawan^{1✉}, I Komang Alit Astawa², Putu Ananta Sastra Wijaya³, I Gusti Agung Putu Raka Agung⁴, I Wayan Shandyasa⁵
Universitas Udayana
Email: madeaditya97@gmail.com^{1✉}

Abstrak

Budidaya ikan lele menghadapi tantangan dalam pemantauan kondisi kolam yang konsisten dan berkelanjutan. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi Android yang terhubung dengan layanan Firebase guna memantau kondisi kolam berbasis sistem *Internet of Things*. Aplikasi ini memungkinkan pengguna untuk melihat data sensor melalui antarmuka seluler dan menyimpan riwayat pengamatan menggunakan dua layanan *database*, yaitu *Realtime Database* dan *Firestore*. Pengembangan dilakukan menggunakan pendekatan pemodelan perangkat lunak dan pengujian sistem dilakukan melalui metode *blackbox*. Hasil pengujian menunjukkan bahwa aplikasi dapat menampilkan data, mencatat histori, dan merespons masukan pengguna sesuai harapan. Temuan ini menunjukkan bahwa sistem yang dikembangkan dapat menjadi solusi digital untuk mendukung pemantauan kolam lele secara terintegrasi melalui perangkat *mobile*.

Kata Kunci: *Aplikasi Mobile, Blackbox Testing, Firebase, Internet of Things.*

Abstract

Catfish farming faces challenges in consistently monitoring pond conditions. This study aims to design and develop an Android application connected to Firebase services to support pond condition monitoring based on an Internet of Things system. The application enables users to view sensor data via a mobile interface and store observation history using two database services: Realtime Database and Firestore. The development process applied software modeling, and the system was tested using the blackbox method. The results show that the application successfully displays data, records history, and responds to user input as expected. These findings indicate that the developed system can serve as a digital solution to support integrated pond monitoring through mobile devices.

Keyword: *Application Mobile, Blackbox Testing, Firebase, Internet of Things.*

PENDAHULUAN

Perkembangan teknologi digital saat ini telah membawa transformasi besar dalam berbagai sektor kehidupan, termasuk di bidang perikanan. Salah satu inovasi penting yang muncul adalah penerapan *Internet of Things* (IoT), yakni konsep yang menghubungkan berbagai perangkat untuk saling bertukar data melalui jaringan internet. IoT telah berkembang pesat dan mulai diimplementasikan dalam berbagai sistem pemantauan otomatis, termasuk pada budidaya ikan air tawar seperti ikan lele. Dengan memanfaatkan sensor dan konektivitas jaringan, sistem berbasis IoT mampu mengumpulkan data secara langsung saat kejadian berlangsung dari lingkungan kolam, seperti suhu air, pH, ketinggian air, dan parameter penting lainnya yang berpengaruh terhadap pertumbuhan serta kesehatan ikan.

Seiring dengan itu, pesatnya perkembangan aplikasi *mobile* berbasis Android telah meningkatkan efisiensi dalam pemantauan dan pengendalian sistem IoT. Android merupakan sistem operasi *mobile* paling banyak digunakan secara global, dengan lingkungan pengembangan yang fleksibel dan terbuka. Salah satu keunggulan utamanya adalah kemampuan untuk dikembangkan secara modular, sehingga fitur dalam aplikasi dapat disesuaikan ditambah atau dikurangi sesuai kebutuhan pengguna dan tingkat kompleksitas sistem. Fleksibilitas ini sangat penting dalam mendukung sistem pemantauan berbasis IoT yang sering kali memerlukan penyesuaian fitur seiring dengan perubahan kebutuhan atau skala usaha.

Integrasi antara sistem IoT dan aplikasi Android memerlukan platform yang mampu mengelola dan menyinkronkan data secara efisien. Firebase, sebagai layanan *Backend-as-a-Service* (BaaS) dari Google, menjadi solusi yang mendukung pengembangan aplikasi *mobile* yang terintegrasi dengan sistem IoT. Firebase menyediakan dua layanan basis data

utama: *Realtime Database*, untuk sinkronisasi data secara langsung, dan *Cloud Firestore*, untuk pencatatan data historis serta pemrosesan data yang lebih fleksibel. Kombinasi keduanya membuat sistem dapat berjalan tanpa infrastruktur server lokal, sehingga proses pengembangan dan pemeliharaan aplikasi menjadi lebih sederhana.

Melihat terbukanya peluang penerapan teknologi ini, diperlukan penelitian yang tidak hanya mengembangkan sistem *monitoring* kolam lele berbasis IoT, tetapi juga mengimplementasikannya dalam aplikasi Android yang terintegrasi dengan Firebase, serta mengevaluasi kinerjanya. Untuk memastikan seluruh fungsi berjalan sesuai kebutuhan pengguna, digunakan metode pengujian *blackbox* yang berfokus pada pengujian *input* dan *output*, tanpa melihat struktur kode internal.

Penelitian ini bertujuan untuk merancang, mengembangkan, dan menguji sebuah aplikasi Android terintegrasi dengan Firebase dalam sistem IoT, guna memantau kondisi kolam budidaya ikan lele secara langsung saat kejadian berlangsung. Fokus utama penelitian ini adalah merancang integrasi antara teknologi IoT dan aplikasi Android secara optimal untuk mendukung aktivitas budidaya, sekaligus memberikan kemudahan dalam pemantauan jarak jauh melalui perangkat *mobile*.

METODE PENELITIAN

Jenis Penelitian

Penelitian ini menggunakan metode *Research and Development (R&D)*, yaitu suatu proses sistematis yang terdiri dari beberapa tahapan untuk merancang, mengembangkan, atau menyempurnakan suatu produk yang telah ada maupun menciptakan produk baru (Mutmainah dkk., 2024).

Penelitian ini hanya membuat aplikasi berbasis Android untuk pemantauan dan mengubah perintah pada Firebase. Fokus utama dalam penelitian ini adalah pengembangan aplikasi Android sebagai antarmuka pengguna untuk memantau dan mengendalikan sistem melalui Firebase. Adapun konteks *Internet of Things (IoT)* yang digunakan adalah pada sistem pemantauan kolam lele, namun aspek perangkat keras tidak menjadi bagian dari kajian dalam tulisan ini karena telah dijelaskan secara terpisah pada publikasi lain.

Rancangan Penelitian

Pada tahap rancangan penelitian, penelitian ini menerapkan pendekatan model ADDIE yang terdiri dari lima langkah utama, yaitu analisis (*analysis*), perancangan (*design*),

pengembangan atau pembuatan (*development/production*), penerapan atau distribusi (*implementation/delivery*), dan evaluasi (*evaluation*). Seluruh tahapan tersebut dilakukan secara terstruktur dan sistematis.

a) Analisis

Pada tahap analisis, penulis mengidentifikasi kebutuhan dan permasalahan yang terjadi dalam proses pemantauan kolam ikan lele. Proses ini dilakukan melalui wawancara dan observasi langsung terhadap pengelola kolam untuk mengetahui alur kegiatan *monitoring* yang berjalan secara manual. Selain itu, data terkait parameter penting seperti suhu dan pH air dikumpulkan, disertai dengan telaah literatur yang relevan untuk memahami praktik dan teknologi yang umum digunakan dalam sistem *monitoring* berbasis *Internet of Things* (IoT) pada sektor akuakultur.

b) Perancangan

Pada tahap desain, penulis menyusun rancangan sistem *monitoring* kolam ikan lele berbasis aplikasi Android. Perancangan meliputi antarmuka pengguna (UI), alur navigasi aplikasi, serta struktur komunikasi antara aplikasi, *Firestore Realtime Database*, dan *Firestore*. Diagram UML seperti *use case* dan *activity diagram* digunakan untuk menggambarkan fungsionalitas sistem serta hubungan antar komponen. Tujuan dari tahap ini adalah memastikan rancangan aplikasi sesuai dengan kebutuhan pengguna yang telah diidentifikasi pada tahap analisis.

c) Pengembangan atau Produksi

Tahap pengembangan dilakukan dengan membangun aplikasi Android menggunakan Android Studio. Bahasa pemrograman yang digunakan adalah Java dan XML. Integrasi *Firestore* dilakukan dengan menghubungkan aplikasi ke *Realtime Database* untuk pengambilan data sensor dan ke *Firestore* untuk pencatatan histori data. Fitur-fitur utama seperti tampilan data suhu, pH, kontrol jadwal, dan histori pemantauan dikembangkan berdasarkan desain yang telah disusun sebelumnya.

d) Implementasi atau Pengiriman

Setelah aplikasi dikembangkan, sistem diimplementasikan dengan menghubungkan aplikasi Android ke perangkat IoT yang digunakan untuk membaca parameter kolam. Implementasi juga mencakup pengaturan koneksi *Firestore* dan pengujian awal pada lingkungan kolam ikan lele. Aplikasi kemudian diserahkan kepada pengguna untuk digunakan dalam kegiatan monitoring harian guna mengamati keandalan dan kemudahan pemakaian.

e) Evaluasi

Tahap evaluasi dilakukan untuk menilai fungsionalitas aplikasi menggunakan metode *blackbox testing*. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur dalam aplikasi berjalan sesuai dengan fungsinya berdasarkan *input* pengguna. Evaluasi juga dilakukan terhadap kemudahan penggunaan, respons aplikasi, serta ketepatan data yang ditampilkan.

Objek Penelitian

Objek penelitian dilakukan pada kolam lele yang dibuat menyerupai kolam asli Pondok Naren, yang berlokasi di Desa Mambang, Kecamatan Selemadeg Timur, Kabupaten Tabanan.

Metode Pengumpulan Data

a. Metode Wawancara

Mewawancarai pemilik kolam untuk mendapatkan wawasan langsung tentang penggunaan aplikasi LeSmart jika diterapkan pada kolam Pondok Naren. Pertanyaan mencakup fungsionalitas dari fitur-fitur yang tersedia pada aplikasi LeSmart.

b. Metode Observasi

Mengamati secara langsung penggunaan aplikasi LeSmart baik itu penerimaan data sensor dan pengiriman perintah aktuator pada kolam lele. Melalui observasi, kita dapat memperoleh pemahaman yang lebih mendalam mengenai cara penggunaan sistem pada kolam lele, sekaligus mengidentifikasi berbagai potensi permasalahan (*bug*) maupun peluang perbaikan pada aplikasi LeSmart.

c. Metode Studi Literatur

Metode studi pustaka dalam penelitian adalah pendekatan yang digunakan untuk mengkaji, merangkum, serta menyusun informasi yang bersumber dari literatur ilmiah dan referensi lain yang relevan dengan topik penelitian.

HASIL DAN PEMBAHASAN

Analisis

Berdasarkan hasil wawancara dan observasi yang dilakukan terhadap salah satu peternak ikan lele di Desa Mambang, Kecamatan Selemadeg Timur, Kabupaten Tabanan, atas nama Made Agus Sanjaya, S.Sn., diperoleh sejumlah kebutuhan spesifik dalam pengelolaan kolam budidaya. Untuk menjawab kebutuhan tersebut, dikembangkan sebuah aplikasi Android yang memiliki kemampuan komunikasi dua arah antara sistem IoT dan pengguna. Aplikasi ini dilengkapi dengan berbagai fitur utama, antara lain: autentikasi pengguna, pemantauan parameter lingkungan kolam, pengendalian aktuator, notifikasi

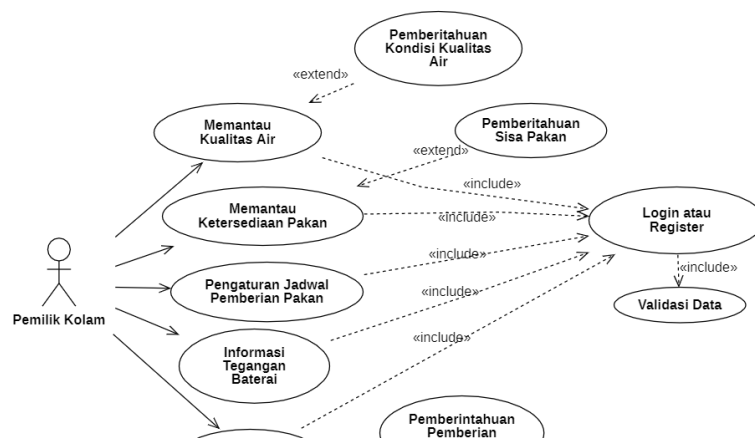
otomatis, penjadwalan pemberian pakan, Menu panduan penanganan masalah pada kolam, dan visualisasi data historis. Adapun data yang dimonitor melalui fitur pemantauan meliputi: suhu air, tegangan baterai, pH air, tinggi permukaan air, serta ketersediaan pakan. Aplikasi mengambil data pada Firebase yang sebelumnya dikirimkan oleh perangkat *hardware*.

Perancangan

Proses perancangan sistem menggunakan UML, *use case diagram* dan *activity diagram*. Perancangan *database* menggunakan diagram noSQL.

a. Use Case Diagram

Use case diagram merupakan gambaran visual dari interaksi antara pengguna dan sistem. Diagram ini digunakan untuk memahami berbagai fungsi yang tersedia dalam sistem serta mengetahui pihak-pihak yang memiliki akses terhadap fungsi tersebut (Taufan dkk., 2022). Gambar *Use Case Diagram* dari pembuatan aplikasi dapat dilihat pada Gambar 1.



Gambar 1. Use Case Diagram

Pemilik kolam sebagai aktor dapat memantau kualitas air (pH dan suhu), sisa pakan, serta status baterai secara nirkabel melalui aplikasi LeSmart. Pemilik juga dapat mengatur jadwal pemberian pakan otomatis, menerima notifikasi jika kualitas air tidak normal, dan memantau status pemberian pakan.

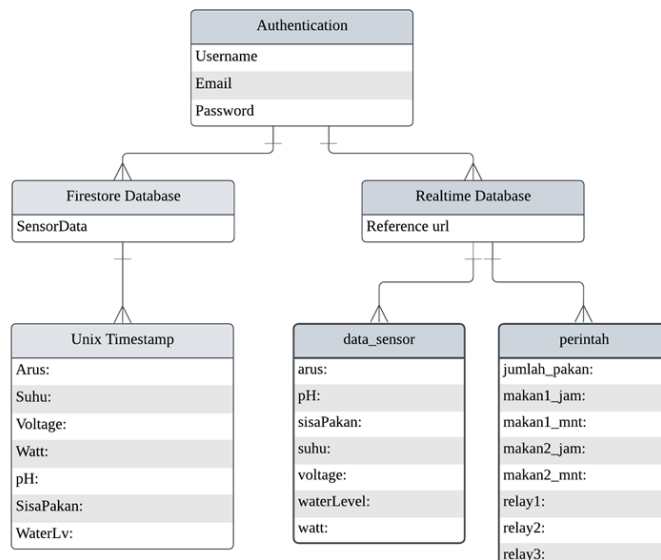
b. Activity Diagram

Activity diagram merupakan bentuk pemodelan yang digunakan untuk menggambarkan alur aktivitas dalam suatu sistem. Diagram ini berfungsi untuk menjelaskan proses kerja program tanpa perlu melihat kode program atau antarmuka pengguna (Puturuhu, 2022). Activity Diagram aplikasi LeSmart bisa dilihat pada Gambar 2.

mengontrol pengurasan dan pengisian air kolam dengan perintah yang mengaktifkan aktuatur seperti *solenoid valve* DC dan motor servo. Aplikasi juga memberikan notifikasi terkait status pemberian pakan, pengurasan, maupun pengisian air secara nirkabel.

c. Diagram Pemodelan Data NoSQL

Non-relational database, atau yang dikenal sebagai NoSQL (singkatan dari *Not Only SQL*), adalah jenis basis data yang tidak bergantung sepenuhnya pada perintah SQL dalam pengelolaan datanya. Basis data ini dirancang untuk kebutuhan khusus sesuai dengan model data tertentu dan menawarkan skema yang fleksibel, sehingga cocok digunakan dalam pengembangan aplikasi modern (Haryanto dkk., 2024). Empat kategori pemodelan data, yaitu berbasis nilai-kunci, dokumen, kolom, dan graf. Diagram NoSQL aplikasi LeSmart bisa dilihat pada Gambar 3.



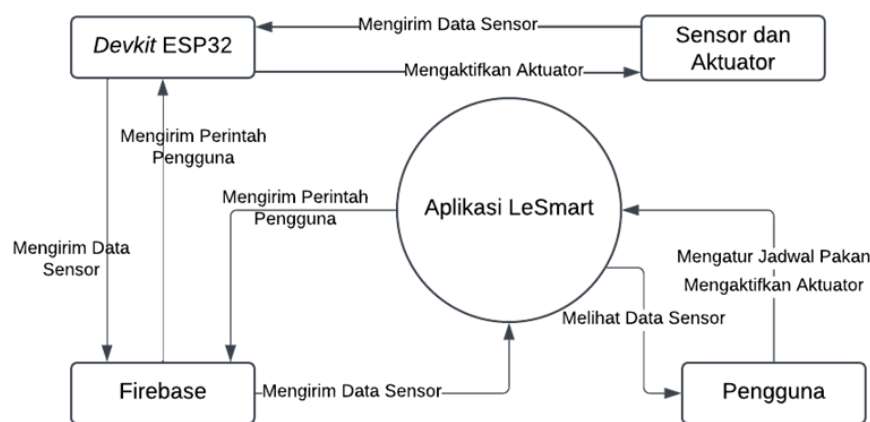
Gambar 3. Diagram NoSQL

Diagram pemodelan data NoSQL menggambarkan hubungan antar entitas utama yaitu *Authentication*, *Firestore Database*, dan *Realtime Database*. Entitas *Authentication* mencakup atribut *Username*, *Email*, dan *Password* yang digunakan untuk proses autentikasi pengguna agar dapat mengakses aplikasi dan mengontrol sistem. *Firestore Database* berfungsi sebagai penyimpanan data historis dari hasil pembacaan sensor dengan atribut seperti Arus, Suhu, Voltage, Watt, pH, SisaPakan, dan WaterLv yang disimpan menggunakan *timestamp* Unix untuk keperluan *monitoring* jangka panjang. Sementara itu, *Realtime Database* berperan dalam menyimpan dan memperbarui data yang terbagi menjadi dua bagian, yaitu *data_sensor* berisi data terkini dari sensor seperti arus, pH, suhu, sisa pakan, tegangan, watt, dan ketinggian air, serta perintah yang mencakup atribut seperti jumlah_pakan, jadwal pemberian pakan (*makan1_jam*, *makan1_mnt*, *makan2_jam*, *makan2_mnt*), dan *relay* (*relay1*, *relay2*, *relay3*) untuk mengendalikan aktuatur secara

langsung melalui aplikasi.

d. Diagram konteks

Diagram konteks merupakan diagram yang menggambarkan suatu lingkaran besar yang dapat mewakili seluruh proses yang terdapat dalam suatu sistem (Sari dkk., 2021). Diagram konteks menggambarkan hubungan antara entitas-entitas yang terdapat diluar sistem dan masukan serta keluaran pada sistem aplikasi LeSmart. Diagram konteks aplikasi LeSmart dapat dilihat pada Gambar 4.



Gambar 4. Diagram Konteks

Interaksi antar komponen dalam sistem Aplikasi LeSmart melibatkan ESP32 *devkit* yang berfungsi menghubungkan sensor dan aktuator dengan Firebase, mengirim data sensor ke Firestore, serta menerima perintah dari Firebase untuk mengontrol aktuator. Sensor digunakan untuk mengukur parameter seperti suhu, pH, ketinggian air kolam, arus, dan tegangan baterai, sementara aktuator seperti motor DC, motor servo, dan *solenoid valve* menerima perintah dari aplikasi untuk pemberian pakan dan pengelolaan air. Firebase berperan sebagai *database cloud* yang menyimpan data sensor dan menjadi penghubung komunikasi dua arah antara ESP32 *devkit* dan aplikasi. Aplikasi LeSmart digunakan oleh pengguna untuk memantau data sensor secara langsung saat kejadian berlangsung dan mengontrol aktuator, termasuk pengaturan jadwal pemberian pakan dan pengurusan air kolam. Pengguna atau pemilik kolam dapat mengakses semua fitur tersebut melalui aplikasi LeSmart untuk mengelola kolam.

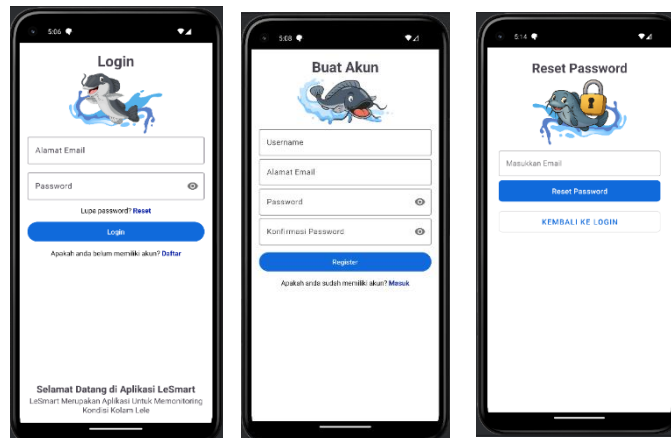
Pengembangan atau produksi

Pada tahap ini dilakukan pembuatan aplikasi menggunakan Android Studio. Android Studio merupakan IDE (Integrated Development Environment) resmi untuk membangun aplikasi Android dan tersedia secara gratis karena bersifat open source. Platform ini menyediakan emulator yang mendukung berbagai jenis perangkat dengan beragam ukuran

dan bentuk. Selain itu, Android Studio juga dilengkapi dengan template kode dan integrasi GitHub, sehingga memudahkan para pengembang dalam membuat aplikasi menggunakan contoh kode yang tersedia atau mengimpor proyek langsung dari GitHub (Medikano dkk., 2023).

a. Aumentikasi

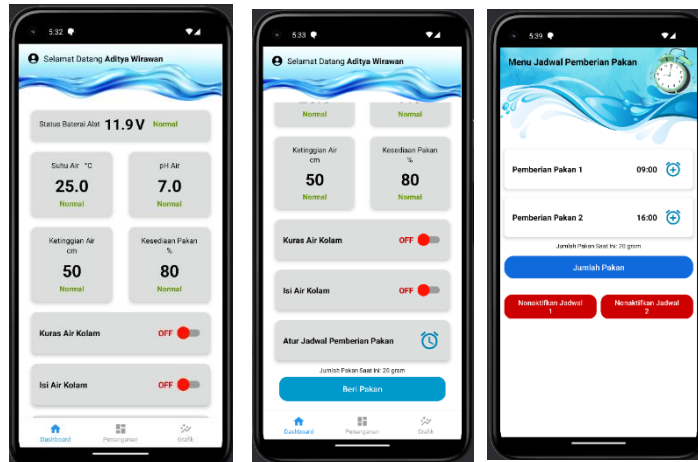
Pada aumentikasi terdapat fitur *login*, *register* dan *reset password* yang dapat dilihat pada Gambar 5.



Gambar 5. Autentikasi

Tampilan menu *login* pada aplikasi LeSmart diperuntukkan bagi pengguna terdaftar dengan memasukkan *email* dan *password* yang diverifikasi melalui *database Firebase* agar dapat mengakses *dashboard*. Bagi pengguna yang belum memiliki akun, tersedia menu *register* yang mengharuskan pengguna mengisi *username*, *email*, *password*, dan konfirmasi *password*, di mana data tersebut akan disimpan di *Firebase* untuk keperluan *login*. Selain itu, terdapat halaman *reset password* sebagai fitur pemulihan akun, di mana pengguna cukup memasukkan *email* terdaftar dan *Firebase* akan mengirimkan tautan *reset password* ke *email* tersebut, dengan tombol "Reset Password" untuk memulai proses pengiriman dan tombol "Kembali ke Login" untuk kembali ke halaman *login*.

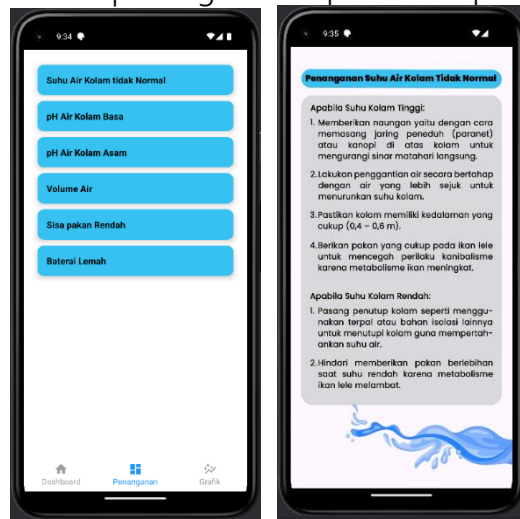
b. Menu *Dashboard* dan pengaturan jadwal pemberian pakan dapat dilihat pada Gambar 6.



Gambar 6. Menu *Dashboard*

Tampilan menu *Dashboard* pada aplikasi LeSmart menampilkan pembacaan parameter sensor seperti suhu, pH, ketinggian air kolam, ketersediaan pakan, dan tegangan baterai, lengkap dengan indikator status masing-masing parameter. Pada bagian atas *dashboard* terdapat informasi status baterai, sedangkan bagian lainnya tersedia fitur kontrol berupa tombol *switch* untuk mengatur proses pengurasan dan pengisian air kolam sesuai kebutuhan. Pengguna juga dapat mengatur jadwal pemberian pakan melalui fitur "Atur Jadwal Pemberian Pakan" serta memberikan pakan secara manual melalui tombol biru "Beri Pakan". Pada menu pengaturan jadwal pemberian pakan, tersedia opsi untuk mengubah jadwal dengan menekan ikon jam yang diarahkan ke fitur *time picker*, serta menghapus jadwal dengan menekan tombol merah. Jumlah pakan yang diberikan dapat diatur melalui tombol "Jumlah Pakan", di mana pengguna memasukkan dosis pakan dalam satuan gram sesuai kebutuhan.

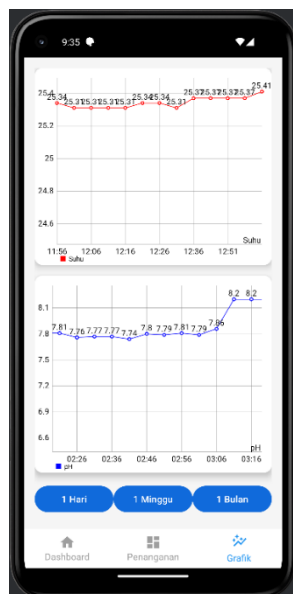
c. Menu Penanganan dan detail penanganan dapat dilihat pada Gambar 7.



Gambar 7. Menu Penanganan dan Detail Penanganan

Menu Penanganan pada aplikasi LeSmart berfungsi sebagai panduan bagi pengguna untuk mengatasi berbagai permasalahan yang dapat terjadi pada kolam lele, seperti suhu air kolam tidak normal, pH air terlalu basa atau asam, ketinggian air yang tidak sesuai, sisa pakan dalam penyimpanan rendah, dan baterai lemah. Halaman Detail Penanganan menyediakan saran atau solusi spesifik untuk setiap permasalahan tersebut, membantu pengguna mengambil tindakan tepat dalam menangani kondisi suhu, pH, ketinggian air, ketersediaan pakan, serta menjaga performa baterai tetap optimal.

d. Menu Grafik dilihat pada Gambar 8.



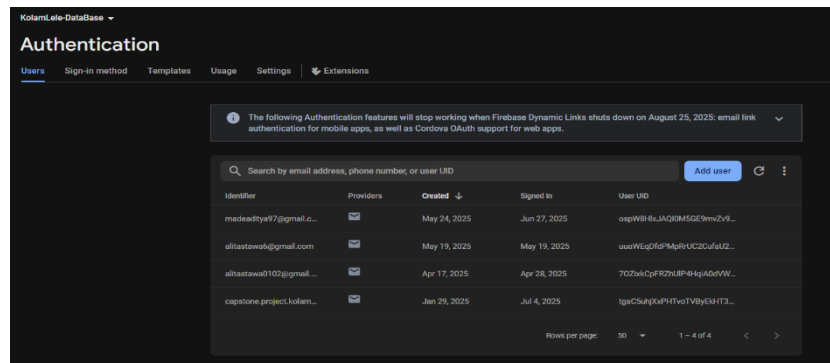
Gambar 8. Menu Grafik

Tampilan menu grafik pada aplikasi LeSmart yang mendukung pengguna untuk memantau perkembangan ikan lele berdasarkan data harian, mingguan, dan bulanan. Fitur

ini memvisualisasikan parameter penting seperti pertumbuhan ikan yaitu suhu dan pH air yang berperan dalam kualitas air kolam lele.

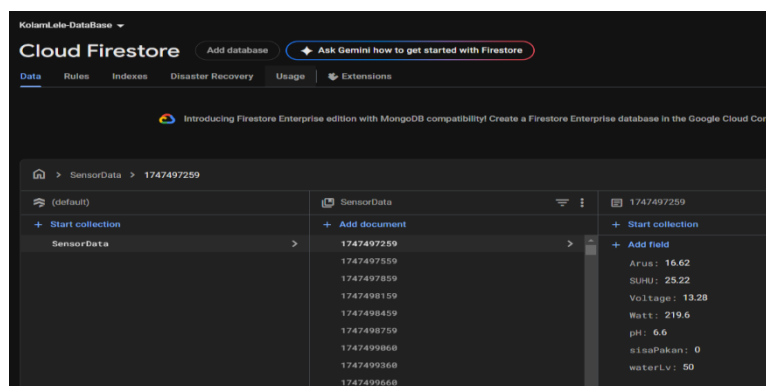
1. Implementasi atau Pengiriman

Pada tahap ini, dilakukan proses integrasi antara aplikasi dengan layanan Firebase untuk memastikan komunikasi data. Pengiriman dan penerimaan data dilakukan dengan memanfaatkan beberapa fitur yang disediakan oleh Firebase, seperti *Realtime Database* untuk sinkronisasi data secara langsung, *Firestore Database* sebagai penyimpanan data dengan berformat JSON, serta *Firestore Authentication* untuk mengelola proses otentikasi pengguna.



Gambar 9. Firebase Authentication

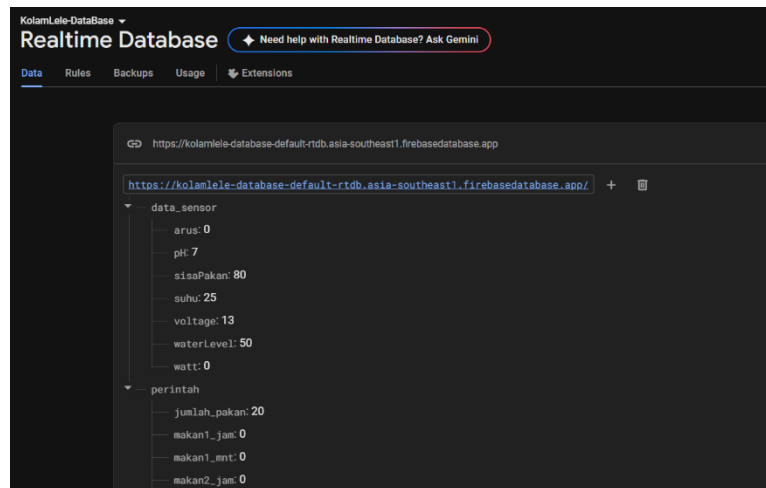
Fitur *Authentication* pada Gambar 9 dalam aplikasi LeSmart menggunakan *Firestore Authentication* untuk mengelola akses pengguna. Data seperti *email*, tanggal pendaftaran, dan riwayat *login* disimpan secara aman di platform tersebut. Melalui halaman ini, admin dapat melihat daftar pengguna yang terdaftar, termasuk *email (identifier)*, tanggal pembuatan akun (*created*), terakhir *login (signed in)*, serta *User UID* unik dari masing-masing pengguna.



Gambar 10. Firebase Cloud Firestore

Fitur *Cloud Firestore* pada Gambar 10 dalam aplikasi LeSmart berfungsi sebagai penyimpanan data historis hasil pembacaan sensor secara *cloud-based*. Setiap data sensor seperti arus, suhu (SUHU), tegangan (Voltage), watt, pH, sisa pakan (sisaPakan), dan

ketinggian air (waterLV) disimpan dengan format *timestamp*. Data-data ini digunakan untuk menampilkan riwayat pemantauan kondisi kolam lele, sehingga pengguna dapat melihat perubahan parameter dari waktu ke waktu melalui aplikasi LeSmart.



Gambar 11. *Firestore Realtime Database*

Fitur *Realtime Database* pada Gambar 11 dalam aplikasi LeSmart berfungsi sebagai media penyimpanan dan pembaruan data sensor serta perintah kontrol secara langsung saat kejadian berlangsung. Bagian *data_sensor* menyimpan data parameter terbaru seperti arus, pH, sisa pakan, suhu, tegangan, ketinggian air, dan watt yang dikirim ke aplikasi. Sedangkan bagian *perintah* berisi data pengaturan seperti jumlah pakan dan jadwal pemberian pakan otomatis.

2. Evaluasi

Pada tahap evaluasi menggunakan metode *blackbox testing* yang dapat dilihat pada Tabel 1.

Tabel 1. Metode *blackbox testing*

No	Fitur Yang Diuji	Skenario	Output Yang Diharapkan	Hasil
1	<i>Register</i> (valid).	Masukkan <i>Username, email</i> (yang belum terdaftar), <i>password</i> dan konfirmasi <i>password</i> .	Data pengguna baru terdaftar di <i>database</i> Firebase dan menerima <i>email</i> untuk verifikasi.	Sesuai
2	<i>Register</i> (<i>username, email, password</i> dan konfirmasi <i>password</i> kosong).	Menekan tombol <i>register</i> tanpa menuliskan <i>username, email, password</i> dan konfirmasi <i>password</i> .	Gagal melakukan <i>register</i> dan menampilkan pesan "Lengkapi semua <i>form</i> dengan benar".	Sesuai

No	Fitur Yang Diuji	Skenario	<i>Output</i> Yang Diharapkan	Hasil
3	<i>Register</i> (konfirmasi <i>password</i> tidak sama).	Memasukan konfirmasi <i>password</i> 654321 tidak sama dengan <i>password</i> 123456.	Gagal melakukan <i>register</i> dan menampilkan pesan "Konfirmasi <i>password</i> harus sama!".	Berhasil
4	<i>Register</i> (jumlah karakter <i>password</i> kurang).	Memasukan <i>password</i> 1234 dan konfirmasi <i>password</i> 1234 kurang dari enam karakter.	Gagal melakukan <i>register</i> dan menampilkan pesan "The given <i>password</i> is invalid, <i>password</i> should be at least 6 character!".	Berhasil
5	<i>Register</i> (format <i>email</i> salah).	Masukkan alamat <i>email</i> madeaditya97gmail.com tanpa menggunakan "@" dan <i>password</i> 12345678.	Gagal <i>register</i> dan menampilkan pesan "The email address is badly formatted".	Berhasil
6	<i>Register</i> (<i>email</i> sudah digunakan).	Menuliskan alamat <i>email</i> yang sudah pernah didaftarkan sebelumnya.	Gagal <i>register</i> dan menampilkan pesan "The email address is already in use by another account".	Berhasil
7	<i>Login</i> (valid).	Masukkan alamat <i>email</i> madeaditya97@gmail.com dan <i>password</i> 0987654321 yang telah terdaftar.	Berhasil <i>login</i> ke menu <i>dashboard</i> .	Berhasil
8	<i>Login</i> (<i>email</i> atau <i>password</i> salah).	Masukkan alamat <i>email</i> admin@gmail.com dan <i>password</i> 12345678 yang belum terdaftar.	Gagal <i>login</i> ke menu <i>dashboard</i> dan menampilkan pesan "Alamat <i>email</i> atau <i>password</i> yang anda masukan salah".	Berhasil
9	<i>Login</i> (<i>email</i> dan <i>password</i> kosong).	Menekan tombol <i>login</i> tanpa menuliskan alamat <i>email</i> dan <i>password</i> .	Gagal <i>login</i> ke menu <i>dashboard</i> dan menampilkan pesan "Silahkan isi <i>email</i> dan <i>password</i> terlebih dahulu".	Berhasil
10	<i>Login</i> (tanpa verifikasi <i>email</i>).	Masukkan alamat <i>email</i> dan <i>password</i> yang sudah didaftarkan tetapi tidak melakukan verifikasi <i>email</i> .	Gagal <i>login</i> ke menu <i>dashboard</i> dan menampilkan dialog box untuk verifikasi <i>email</i> terlebih dahulu atau kirim ulang verifikasi.	Berhasil
11	<i>Reset password</i> (valid).	Masukkan alamat <i>email</i> madeaditya97gmail.com yang sebelumnya sudah terdaftar.	Berhasil mengirimkan <i>email</i> verifikasi ke pengguna untuk melakukan <i>reset password</i> , setelah <i>email</i> berhasil terkirim pengguna	Berhasil

No	Fitur Yang Diuji	Skenario	Output Yang Diharapkan	Hasil
			diarahkan ke <i>website reset password</i> .	
12	<i>Reset password</i> (<i>email</i> kosong).	Menekan tombol <i>reset password</i> tanpa menuliskan <i>email</i> .	Gagal melakukan <i>reset password</i> dan menampilkan pesan "Silahkan isi <i>email</i> terlebih dahulu".	Berhasil
13	<i>Reset password</i> (format <i>email</i> salah).	Masukkan alamat <i>email</i> <i>madeaditya97gmail.com</i> tanpa menggunakan "@".	Gagal <i>reset password</i> dan menampilkan pesan " <i>The email address is badly formatted</i> ".	Berhasil
14	Pemantauan sensor.	Membaca data suhu air kolam, pH air kolam, tegangan baterai, ketinggian air kolam, dan sisa pakan.	Data tampil pada menu <i>dashboard</i> sesuai data_ <i>sensor realtime database</i> Firebase.	Berhasil
15	Pemantauan sensor (tanpa koneksi internet).	Membuka aplikasi tanpa terkoneksi internet.	Data yang tampil pada menu <i>dashboard</i> masih data yang lama dan menampilkan pesan "Tidak ada koneksi internet. Silahkan periksa jaringan Anda".	Berhasil
16	Pengurasan air kolam (ON).	Ketuk tombol <i>switch</i> kuras air kolam yang sebelumnya <i>off</i> .	Kuras air kolam menjadi ON berwarna hijau dan <i>Value relay2</i> (kuras air kolam) pada <i>Firestore Realtime Database</i> diperbarui menjadi <i>true</i> .	Berhasil
17	Pengurasan air kolam (OFF).	Ketuk tombol <i>switch</i> kuras air kolam yang sebelumnya <i>on</i> .	Kuras air kolam menjadi OFF berwarna merah dan <i>Value relay2</i> (kuras air kolam) pada <i>Firestore Realtime Database</i> diperbarui menjadi <i>false</i> .	Berhasil
18	Pengisian air kolam (ON).	Ketuk tombol <i>switch</i> isi air kolam yang sebelumnya <i>off</i> .	Isi air kolam menjadi ON dan <i>Value relay3</i> (isi air kolam) pada <i>Firestore Realtime Database</i> diperbarui menjadi <i>true</i> .	Berhasil
19	Pengisian air kolam (OFF).	Ketuk tombol <i>switch</i> isi air kolam yang sebelumnya <i>on</i> .	Isi air kolam menjadi OFF dan <i>Value relay3</i> (isi air kolam) pada <i>Firestore Realtime Database</i> diperbarui menjadi <i>false</i> .	Berhasil

No	Fitur Yang Diuji	Skenario	Output Yang Diharapkan	Hasil
20	Pemberian pakan otomatis.	Memilih waktu menggunakan <i>time picker</i> dengan ketuk <i>icon</i> jam, masukan jadwal 1 pukul 09:00 dan jadwal 2 pukul 16:00.	<i>TextView</i> jadwal 1 dan jadwal 2 berubah sesuai yang dipilih pada <i>time picker</i> , <i>value</i> makan1_jam, makan1_mnt, makan2_jam, dan makan2_mnt pada Firebase <i>Realtime Database</i> berubah sesuai <i>input</i> tombol nonaktifkan jadwal 1 dan jadwal 2 menjadi aktif atau <i>visible</i> .	Berhasil
21	Nonaktifkan pemberian pakan otomatis.	Ketuk tombol nonaktifkan jadwal 1 atau jadwal 2.	<i>TextView</i> jadwal 1 atau jadwal 2 menjadi <i>off</i> , <i>value</i> menjadi makan1_jam: 0, makan1_mnt: 0, makan2_jam: 0, dan makan2_mnt:0 pada Firebase <i>Realtime Database</i> dan tombol yang diketuk menjadi hilang atau <i>invisible</i> .	Berhasil
22	Pemberian jumlah pakan.	Ketuk tombol jumlah pakan, masukan jumlah pakan 110 gram.	Berhasil mengubah <i>value</i> perintah jumlah_pakan di Firebase menjadi 110.	Berhasil
23	Pemberian pakan manual.	Ketuk tombol beri pakan, konfirmasi bahwa yakin ingin memberi pakan.	<i>Value</i> relay1 (beri pakan) diperbarui menjadi <i>true</i> , tombol tidak bisa langsung diketuk kembali, menunggu <i>countdown</i> selesai.	Berhasil
24	Menu penanganan.	Ketuk salah satu <i>CardView</i> penanganan untuk melihat detail informasi penanganan permasalahan kolam.	Menampilkan halaman detail penanganan sesuai data <i>cardview</i> yang dipilih.	Berhasil
25	Menu Grafik.	Membaca <i>history</i> data sensor suhu dan pH yang disimpan pada <i>Firestore Database</i> , ketuk salah satu tombol 1 hari, 1 minggu atau 1 bulan.	Menampilkan grafik <i>history</i> data sensor suhu dan pH, mengelompokkan dan mencari rata-rata sesuai hari atau mingguan.	Berhasil

No	Fitur Yang Diuji	Skenario	Output Yang Diharapkan	Hasil
26	Notifikasi.	Pemberitahuan kepada pengguna jika data sensor tidak normal, sedang melakukan pengurasan air kolam, pengisian air kolam serta sedang melakukan pemberian pakan.	Menampilkan notifikasi (<i>push notification</i>) saat data sensor berada diluar batas normal, sedang melakukan pengurasan air kolam, pengisian air kolam serta sedang melakukan pemberian pakan.	Berhasil
27	Logout.	Ketuk <i>username</i> untuk menampilkan opsi <i>Logout</i> dan ketuk <i>yes</i> .	Berhasil <i>Logout</i> dari akun dan menampilkan menu <i>login</i> .	Berhasil

SIMPULAN

Penelitian ini menunjukkan bahwa aplikasi Android LeSmart yang terintegrasi Firebase mampu mendukung sistem IoT *monitoring* kolam lele. Fitur utama seperti pemantauan suhu, pH, ketinggian air, sisa pakan, kontrol aktuator (pengurasan, pengisian, pemberian pakan), penjadwalan otomatis, notifikasi, grafik historis, dan menu penanganan masalah pada kolam lele. Evaluasi dengan metode *blackbox testing* menunjukkan bahwa seluruh fitur aplikasi berjalan sesuai fungsinya dan dapat digunakan secara optimal oleh pengguna. Mengisi keterbatasan sistem berbasis platform pihak ketiga seperti Blynk versi gratis. Pengembangan lanjutan disarankan pada integrasi AI prediktif serta pengujian multi-kolam untuk menilai skalabilitas dan keandalan sistem di tingkat produksi massal.

DAFTAR PUSTAKA

- Haryanto, B., Ardiansyah, A., & Kurniasih, M. (2024). Pengenana Database NoSql dan Perbandingannya Dengan Database Relasional. *Jurnal IPSIKOM*, 12(1), 1–7.
- Medikano, A., Sumartono, R. P., Agustina, T., Aisyah, N., & Wirawan, R. (2023). Perancangan Aplikasi Android E-Learn Armeta Dengan Pendekatan Meode Waterfall. *Jurnal Sistem Informasi dan Aplikasi (JSIA)*, 1(1), 34–49. <https://doi.org/10.52958/jsia.v1i1.6450>
- Mutmainah, S., & Wahyuningsih, E. (2024). Implementasi Website E-Commerce Pada Penjualan Makanan Dan Minuman di Kedai “Mesen Kopi” Purworejo. *INNOVATIVE: Journal Of Social Science Research*, 4(4), 9170–9178.
- Puturuhi, V. (2022). Sistem Informasi Manajemen Penelitian dan Pengabdian PNPB Pada Politeknik Negeri Ambon. *Jurnal SIMETRIK*, 12(1), 553–560.

- Sari, L., & Siregar, G. Y. K. S. (2021). Perancangan Aplikasi Pendataan Data Kepegawaian Negeri Sipil Pada Dinas Komunikasi Dan Informatika Kota Metro. *JMIK Jurnal Mahasiswa Ilmu Komputer*, 1(1), 115–135. <https://doi.org/10.24127/.v2i1.1235>
- Taufan, M. A., Rusdianto, D. S., & Ananta, M. T. (2022). Pengembangan Sistem Otomatisasi Use Case Diagram berdasarkan Skenario Sistem menggunakan Metode POS Tagger Stanford NLP. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(8), 3733–3740. <http://j-ptiik.ub.ac.id>