

Rancang Bangun Sistem Pelaporan Fasilitas Berbasis *Multi Platform* dengan *Push Notification System*

Rizki Yulian Agusti¹, Anindito², Sembada Denrineksa Bimorogo³

^{1,2,3}Program Studi Informatika, Fakultas Sains dan Teknologi Pertahanan, Universitas Pertahanan RI, Bogor, Indonesia
surel: ¹yulianrizki10@gmail.com, ²anindito96@gmail.com, ³denri5693@gmail.com

Info Artikel

Sejarah artikel:

Diterima 05-06-2025

Revisi 02-07-2025

Diterima 17-07-2025

Kata kunci:

Fasilitas

Flutter

Laravel

Notifikasi

Pelaporan

ABSTRAK

Sistem pelaporan fasilitas merupakan solusi berbasis teknologi yang bertujuan untuk meningkatkan efektivitas dalam proses pelaporan kerusakan sarana dan prasarana. Dalam praktiknya, pelaporan secara konvensional melalui pesan singkat atau panggilan pribadi seringkali mengakibatkan informasi yang tidak terdokumentasi dengan baik serta keterlambatan dalam penanganan. Penelitian ini merancang dan mengembangkan sistem pelaporan fasilitas berbasis *multi-platform* dengan menggunakan Flutter sebagai pembuat aplikasi, Laravel untuk *backend* API, dan MySQL sebagai basis data. Sistem ini dilengkapi dengan fitur utama berupa formulir pelaporan kerusakan yang mendukung unggahan gambar, kategori, deskripsi, serta lokasi kejadian. Selain itu, sistem juga memiliki fitur *push notification* yang memberikan notifikasi status laporan secara *real-time* kepada pengguna. Pengelolaan laporan dilakukan secara terstruktur dan aman melalui *backend* dengan penerapan *middleware* dan kontrol akses berbasis peran (*role-based access control*). Hasil pengembangan menunjukkan bahwa sistem ini dapat meningkatkan transparansi, mempercepat tindak lanjut perbaikan, serta memberikan pengalaman pengguna yang baik melalui antarmuka yang responsif. Ke depan, sistem ini dapat dikembangkan lebih lanjut dengan integrasi sistem eksternal dan penambahan fitur evaluasi layanan untuk meningkatkan akuntabilitas dan kualitas pengelolaan fasilitas.

Penulis yang sesuai:

Rizki Yulian Agusti

Program Studi Informatika Fakultas Sains dan Teknologi Pertahanan Universitas Pertahanan RI

Email: yulianrizki10@gmail.com

1. PENDAHULUAN

Sistem pelaporan fasilitas merupakan salah satu bentuk inovasi teknologi informasi yang dirancang untuk mempermudah proses penyampaian dan penanganan laporan kerusakan atau permasalahan terkait sarana dan prasarana. Dalam lingkungan institusi seperti pendidikan, perkantoran, maupun fasilitas umum, keberadaan sistem ini sangat penting guna memastikan kelangsungan operasional yang nyaman dan efisien[1]. Sistem pelaporan yang baik harus mampu mendokumentasikan laporan secara sistematis, memberikan notifikasi status penanganan, serta mendukung komunikasi dua arah antara pelapor dan pihak pengelola[2].

Namun, dalam praktiknya, masih banyak institusi yang menerapkan metode pelaporan secara konvensional, seperti melalui pesan singkat atau panggilan pribadi. Metode ini kerap menimbulkan sejumlah kendala, antara lain kurangnya dokumentasi, keterlambatan tindak lanjut, hingga hilangnya laporan sebelum diproses[3]. Sistem seperti ini tidak hanya menyulitkan dalam aspek transparansi dan akuntabilitas, tetapi juga menghambat efisiensi kerja. Permasalahan tersebut sebagian besar disebabkan oleh belum tersedianya *platform* digital yang memungkinkan pelaporan dilakukan secara *real-time* dan terdokumentasi secara rapi[2].

Berbagai penelitian sebelumnya telah mencoba mengatasi tantangan ini melalui pengembangan sistem pelaporan berbasis teknologi. Prasetyo merancang sistem pelaporan fasilitas publik berbasis web untuk mendukung inisiatif Smart City dan keterbukaan data[4]. Suyanto dan Andri mengembangkan aplikasi pelaporan kerusakan jalan menggunakan pendekatan *Rapid Application Development* (RAD)[5]. Sementara penelitian di lingkungan pendidikan juga menunjukkan hasil positif dalam penggunaan aplikasi pelaporan berbasis mobile untuk meningkatkan pelayanan dan respon terhadap keluhan fasilitas kampus[6].

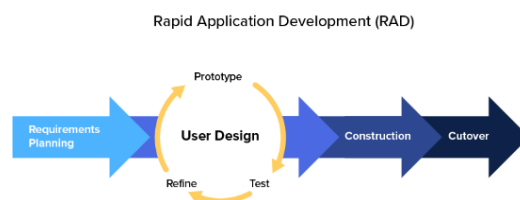
Meski begitu, sebagian besar sistem yang dikembangkan masih berfokus pada platform tunggal atau belum sepenuhnya mengintegrasikan fitur notifikasi *real-time*, padahal komunikasi aktif merupakan elemen penting dalam sistem pelaporan yang efektif. Selain itu, belum banyak sistem yang menerapkan pengelolaan akses berbasis peran secara ketat untuk menjamin keamanan data dan hak akses pengguna.

Penelitian ini bertujuan untuk merancang dan membangun sistem pelaporan fasilitas berbasis *multi-platform* yang mampu mengatasi permasalahan pelaporan konvensional yang tidak terdokumentasi dengan baik dan lambat dalam penanganan. Sistem dikembangkan menggunakan Flutter untuk aplikasi *mobile*, Laravel sebagai *backend* API, dan MySQL sebagai basis data[7]. Tujuan utama dari pengembangan ini adalah menyediakan platform digital yang memungkinkan pelaporan kerusakan dilakukan secara *real-time*, terdokumentasi dengan baik, serta mudah diakses melalui berbagai perangkat[8]. Selain itu, sistem ini dirancang untuk menyertakan fitur notifikasi instan (*push notification*) yang memberikan informasi status laporan secara langsung kepada pelapor, serta menerapkan kontrol akses berbasis peran guna menjamin keamanan dan efisiensi dalam pengelolaan data[9].

Kebaruhan dalam penelitian ini terletak pada integrasi teknologi *multi-platform* dengan sistem notifikasi *real-time* dalam konteks pelaporan fasilitas. Berbeda dari penelitian sebelumnya yang umumnya terbatas pada satu platform dan tidak menyertakan komunikasi dua arah yang aktif, sistem ini menggabungkan kemudahan akses melalui aplikasi *mobile* dan *web*, serta mendukung transparansi melalui notifikasi status laporan[10]. Selain itu, penelitian ini juga menekankan aspek keamanan dengan penerapan *middleware* dan pengaturan hak akses berbasis peran (*role-based access control*), yang masih jarang diimplementasikan dalam sistem pelaporan fasilitas serupa[11]. Inovasi ini diharapkan mampu meningkatkan kualitas layanan dan efisiensi pengelolaan fasilitas di berbagai jenis lingkungan pengguna.

2. METODE

Penelitian ini menggunakan metode *Rapid Application Development* (RAD) sebagai pendekatan utama dalam merancang dan mengembangkan sistem pelaporan fasilitas berbasis *multi-platform*. RAD dipilih karena mampu mempercepat proses pengembangan perangkat lunak melalui tahapan iteratif dan umpan balik pengguna yang kontinu[12]. Penelitian ini bersifat terapan dengan pendekatan kualitatif-deskriptif, yang bertujuan menghasilkan solusi berbasis sistem terhadap masalah nyata di lapangan.



Gambar 1. *Rapid Application Development*

Berikut adalah tahapan-tahapan dalam proses metode *Rapid Application Development* (RAD)[13]:

2.1 *Requirements Planning*

Perancangan merupakan tahap untuk membuat representasi visual dari sistem menggunakan diagram. Pada proses ini, perancangan sistem dilakukan berdasarkan identifikasi masalah dan perencanaan kebutuhan sebelumnya. Dalam penelitian ini, rancangan sistem digambarkan menggunakan diagram UML, diagram use case, dan diagram aktivitas, untuk membantu pengembang dalam mengimplementasikan kebutuhan sistem ke dalam program aplikasi.

Konstruksi adalah tahap pelaksanaan penulisan kode program yang dilakukan oleh *programmer*. Tahap ini merupakan proses pengkodean dari desain sistem yang telah dibuat[14]. Dalam penelitian ini, desain sistem diimplementasikan menggunakan bahasa pemrograman Dart dengan *framework* Flutter untuk pengembangan aplikasi *front-end* (mobile), serta Laravel sebagai *framework backend* yang dibangun menggunakan bahasa pemrograman PHP. Flutter dipilih karena kemampuannya dalam membangun antarmuka pengguna yang responsif dan menarik untuk berbagai platform dengan satu basis kode, sedangkan Laravel digunakan karena memiliki struktur arsitektur yang rapi, keamanan yang baik, serta dukungan fitur lengkap untuk mempercepat proses pengembangan *backend*. Untuk sistem manajemen basis data, digunakan MySQL karena kompatibel dengan Laravel dan mampu menangani data dalam skala menengah hingga besar secara efisien.

Cutover adalah tahap pengujian dari sistem yang telah dibangun. Seluruh sistem diuji untuk memastikan bahwa sistem tersebut memenuhi kebutuhan dan berjalan sebagaimana mestinya[15]. Dalam penelitian ini, proses pengujian aplikasi menggunakan metode *Blackbox testing* dengan menyiapkan beberapa skenario pengujian untuk menguji apakah sistem berjalan sesuai dengan rencana.

Penelitian ini memiliki tujuan untuk merancang aplikasi *multi-platform* yang berguna dalam proses pelaporan fasilitas pada sebuah instansi atau organisasi. Aplikasi ini tidak hanya menyediakan fitur pelaporan saja, namun memiliki fitur notifikasi secara real-time terhadap progres pelaporan yang dibuat. Proses perancangan aplikasi ini menggunakan metode RAD yang menghasilkan beberapa fitur yang responsif, mudah digunakan, serta mampu mempercepat alur komunikasi antara pelapor dan pihak pengelola fasilitas, seperti fitur pelaporan kerusakan, pelacakan status laporan, notifikasi real-time, hingga histori laporan yang terdokumentasi dengan baik.

```

    usecaseDiagram
        actor SK as Staf Kantor
        actor Admin
        actor SR as Staf Rumah

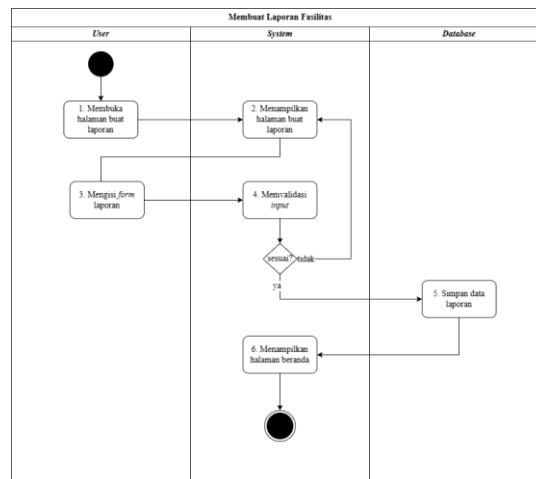
        usecase U1((Identifikasi Laporan Fasilitas))
        usecase U2((Mengetahui Data Alam))
        usecase U3((Mengetahui Data Alam))
        usecase U4((Menambahkan Alam))
        usecase U5((Login))
        usecase U6((Mengetahui Status Laporan (dan Menambahkan Fasilitas)))
        usecase U7((Mengetahui Laporan))
        usecase U8((Menambahkan Fasilitas Laporan))

        SK --> U1
        SK --> U2
        SK --> U3
        SK --> U4
        SK --> U5
        Admin --> U4
        Admin --> U3
        Admin --> U3
        Admin --> U5
        SR --> U6
        SR --> U7
        U1 --> U5
        U2 --> U5
        U3 --> U5
        U4 --> U5
        U5 --> U6
        U5 --> U7
        U5 --> U8
    
```

Sistem pelaporan fasilitas yang dikembangkan memiliki sejumlah fitur utama yang diklasifikasikan ke dalam beberapa kelompok *use case* sesuai dengan fungsionalitas dan peran pengguna. Use case diagram di atas menggambarkan interaksi antara tiga aktor utama, yaitu Staf Satker, Admin, dan Staf Rumga, dengan sistem pelaporan fasilitas berbasis multi-platform. Setiap aktor memiliki hak akses dan fungsionalitas yang berbeda sesuai dengan peran

mereka dalam sistem. Staf Satker dapat melakukan login ke dalam sistem, membuat laporan fasilitas, melihat dan mencari laporan, serta menerima notifikasi terkait laporan yang dibuat. Selain itu, Staf Satker juga memiliki akses untuk melihat data akun. Admin memiliki wewenang untuk menambahkan akun baru, mengedit, dan menghapus data akun, yang semuanya bergantung pada fitur melihat akun, serta diawali dengan proses login. Sementara itu, Staf Rumga memiliki peran dalam menangani laporan yang masuk, yaitu dengan login ke dalam sistem, melihat laporan, mengubah status laporan dan memberikan feedback, serta menghapus laporan jika diperlukan. Setiap fungsi utama sistem diawali dengan proses login dan beberapa fitur saling bergantung melalui relasi *include* dan *extend*, yang menunjukkan bahwa fungsi-fungsi tertentu merupakan bagian dari atau perluasan dari fungsi lain. Diagram ini secara keseluruhan menjelaskan alur kerja sistem pelaporan fasilitas yang dibangun, serta menjabarkan hak akses masing-masing pengguna untuk memastikan sistem berjalan secara efektif dan terstruktur.

3.2. Activity Diagram

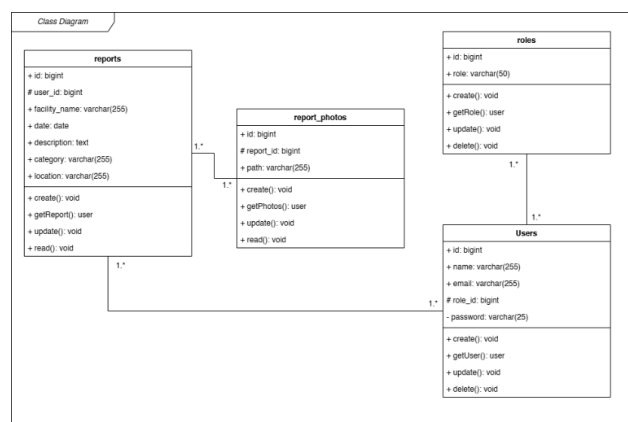


Gambar 3. Activity Diagram

Activity diagram di atas menggambarkan alur proses dalam membuat laporan fasilitas pada sistem pelaporan berbasis *multi-platform*. Proses dimulai dari pengguna (user) yang membuka halaman pembuatan laporan, kemudian sistem akan menampilkan halaman form laporan yang harus diisi. Setelah pengguna mengisi form, sistem akan melakukan validasi terhadap input yang diberikan. Jika data yang dimasukkan tidak sesuai, sistem akan mengarahkan kembali ke form laporan. Namun, jika input sudah valid, maka data laporan akan disimpan ke dalam *database*. Setelah proses penyimpanan berhasil, sistem akan menampilkan halaman beranda kepada pengguna sebagai tanda bahwa laporan telah berhasil dibuat.

Diagram ini menunjukkan bahwa sistem secara otomatis memverifikasi kevalidan data dan hanya akan menyimpan laporan yang memenuhi ketentuan. Proses ini menggambarkan alur yang sederhana, terstruktur, dan efisien, yang bertujuan untuk memberikan pengalaman pelaporan yang mudah dan cepat bagi pengguna.

3.3 Class Diagram

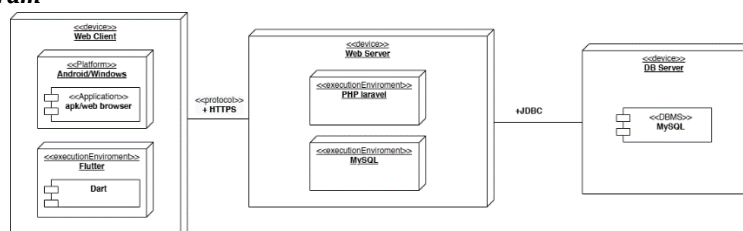


Gambar 4. Class Diagram

Struktur data sistem pelaporan fasilitas dirancang menggunakan empat kelas utama, yaitu Users, Reports, Report_Photos, dan Role. Kelas Users menyimpan informasi pengguna dengan atribut seperti nama, email, dan peran. Setiap pengguna memiliki relasi dengan kelas Role yang menentukan hak akses, seperti admin, staf teknis, dan pelapor.

Kelas Reports mencatat laporan kerusakan fasilitas dengan atribut seperti nama fasilitas, tanggal, deskripsi, kategori, dan lokasi. Setiap laporan dapat memiliki banyak foto pendukung yang disimpan dalam kelas Report_Photos. Relasi antar kelas menggunakan pendekatan one-to-many, seperti satu pengguna memiliki banyak laporan dan satu laporan memiliki banyak foto. Seluruh struktur ini diimplementasikan menggunakan Laravel dan Eloquent ORM untuk memastikan pengelolaan data yang efisien dan terstruktur.

3.4. Deployment Diagram



Gambar 5. Deployment Diagram

Arsitektur fisik sistem dirancang dengan pendekatan multi-platform yang terdiri atas tiga komponen utama, yaitu *Web Client*, *Web Server*, dan *Database Server*. Komponen *Web Client* mencakup aplikasi mobile berbasis Android yang dikembangkan menggunakan Flutter, serta antarmuka berbasis web untuk pengguna desktop. Flutter digunakan karena mendukung pengembangan lintas platform dengan satu basis kode, serta mampu menghasilkan antarmuka yang responsif dan efisien.

Web Server berfungsi sebagai pusat logika aplikasi. Komponen ini menjalankan Laravel sebagai backend framework untuk menangani proses autentikasi, pengelolaan laporan, serta komunikasi data. Laravel dipilih karena kemampuannya dalam mengelola API secara efisien, serta dukungan fitur keamanan seperti middleware dan role-based access control.

Database Server bertugas menyimpan dan mengelola seluruh data sistem dengan menggunakan MySQL sebagai sistem manajemen basis data. Web server akan berkomunikasi dengan database untuk melakukan operasi CRUD, dan mengirimkan hasilnya kembali ke client. Arsitektur ini memungkinkan pemrosesan data secara terpusat, aman, dan real-time.

3.5. Alat dan Bahan

Penelitian ini menggunakan dua jenis perangkat keras, yaitu laptop Lenovo ThinkPad i14 dengan prosesor Intel Core i7 generasi ke-11 sebagai alat utama untuk pengembangan backend menggunakan Laravel dan integrasi database MySQL, serta smartphone Redmi 14C sebagai perangkat uji untuk memastikan kompatibilitas dan performa aplikasi pada platform Android.

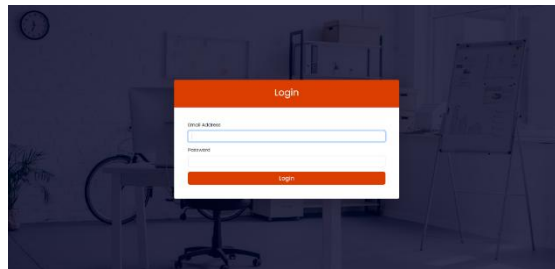
Dari sisi perangkat lunak, pengembangan aplikasi memanfaatkan Flutter sebagai *framework* utama untuk membangun aplikasi *mobile* lintas *platform*, dengan Dart sebagai bahasa pemrogramannya. Laravel digunakan sebagai *backend framework* karena kemampuannya dalam membangun RESTful API secara efisien dan aman. Untuk penyimpanan data, digunakan MySQL sebagai sistem manajemen basis data, dan PHP versi 8.1 untuk menjalankan logika *server-side*. Kombinasi alat dan bahan ini dipilih untuk menunjang proses pengembangan sistem pelaporan fasilitas yang efisien, aman, dan responsif di berbagai perangkat.

3.6. Implementasi

Proses implementasi sistem merupakan tahap pemrograman dari sistem yang telah dirancang menjadi produk akhir yang dapat digunakan. Dalam penelitian ini, fitur-fitur dalam sistem dibagi ke dalam beberapa modul, seperti beranda (dashboard), pelaporan fasilitas, status laporan, dan manajemen pengguna. Setiap modul terdiri dari beberapa submodul. Sebagai contoh, modul pelaporan fasilitas terdiri dari submodul buat laporan, lihat laporan, dan detail laporan. Antarmuka pengguna (user interface) berperan sebagai penghubung antara pengguna dengan sistem aplikasi.

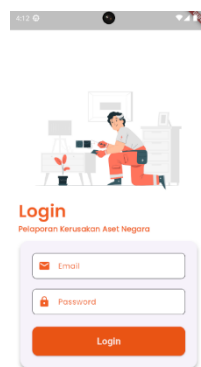
Melalui antarmuka ini, pengguna dapat berinteraksi dengan sistem secara mudah dan konsisten, baik dalam memberikan perintah, menerima keluaran (output), maupun menavigasi berbagai fitur dalam sistem.

3.6.1. Login



Gambar 6. *Login Webview*

Pada gambar 6, tampilan login pada *WebView* ini dirancang untuk memungkinkan pengguna mengakses sistem berdasarkan kredensial mereka. Formulir login terdiri dari dua input utama, yaitu *Email Address* dan *Password*, serta sebuah tombol *Login* untuk mengirimkan data autentikasi. Desainnya menggunakan warna merah dan putih, menciptakan kontras yang jelas untuk meningkatkan keterbacaan dan pengalaman pengguna. *WebView* pada gambar 6 digunakan dalam aplikasi berbasis *web* untuk staf dan admin, di mana setelah pengguna memasukkan *email* dan kata sandi, data akan dikirim ke *backend* Laravel melalui API yang dilindungi oleh *middleware* autentikasi seperti *auth:sanctum*.



Gambar 7. *Login Mobile*

Pada gambar 7, tampilan *login* pada aplikasi *mobile* ini dirancang dengan antarmuka yang sederhana. Desainnya menggunakan warna dominan jingga dan putih, dengan ilustrasi teknisi yang sedang memperbaiki perangkat sebagai elemen *visual* yang mendukung tema aplikasi.

Formulir *login* terdiri dari dua *input* utama, yaitu *Email* dan *Password*. Fitur *login* pada aplikasi pelaporan kerusakan aset negara digunakan oleh staf satuan kerja (satker) untuk mengakses sistem dan melaporkan kerusakan fasilitas. Melalui proses autentikasi ini, staf satker dapat masuk ke dalam sistem menggunakan *email* dan *password* yang telah terdaftar, sehingga sistem dapat memastikan bahwa laporan yang dibuat berasal dari pengguna yang terdaftar. Dengan adanya fitur login ini, sistem juga dapat mencatat identitas pelapor, mengelompokkan laporan berdasarkan pengguna, serta membatasi akses ke fitur tertentu hanya untuk pengguna yang memiliki hak sesuai perannya, seperti *admin* atau staf rumga. Setelah pengguna memasukkan kredensial yang benar, sistem akan mengautentikasi data melalui API yang terhubung ke *backend* Laravel, memastikan bahwa hanya pengguna terdaftar yang dapat mengakses fitur dalam aplikasi.

3.6.2. Membuat Laporan

Gambar 8. Membuat Laporan

Pada gambar 8, *form* tambah laporan pada aplikasi *mobile* ini dirancang untuk memudahkan pengguna dalam melaporkan kerusakan fasilitas. Formulir ini memiliki beberapa *input* utama, yaitu Judul Kerusakan, Tanggal Kejadian yang dapat dipilih melalui kalender, Kategori Kerusakan dalam bentuk *dropdown*, Lokasi yang memberikan contoh format *input*, serta Deskripsi untuk menjelaskan detail kerusakan.

Selain itu, terdapat fitur unggah gambar yang memungkinkan pengguna mengambil foto langsung dari kamera atau memilih gambar dari galeri sebagai bukti pendukung laporan. Semua data yang dimasukkan akan dikirim ke *backend* Laravel. Semua data diisi oleh pengguna dalam form tambah laporan akan dikirimkan ke server melalui sebuah permintaan (request) API. Di sisi server, *framework* Laravel akan menerima data tersebut melalui *endpoint* dalam bentuk *request* POST, dan kemudian melakukan proses seperti validasi data, penyimpanan ke *database*, serta pengelolaan file gambar. Setelah berhasil diproses, sistem *backend* juga akan memberikan *respons* balik kepada aplikasi *mobile* sebagai konfirmasi bahwa laporan telah berhasil dikirim atau jika ada kesalahan dalam *input* data.

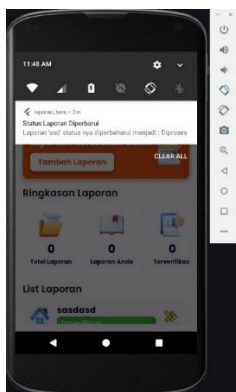
3.6.3. Feedback Laporan

Gambar 9. Feedback Laporan

Gambar 9 di atas menunjukkan antarmuka aplikasi yang digunakan untuk mengelola laporan kerusakan fasilitas. Dalam halaman ini, *admin* atau staf rumga dapat mengubah status laporan kerusakan, seperti memilih status "Terverifikasi", dan memberikan *feedback* terkait laporan tersebut. *Feedback* dapat dimasukkan melalui kolom yang tersedia, dan setelah itu, *admin* atau staf rumga dapat menyimpan perubahan dengan menekan tombol "Simpan". Selain itu, terdapat kolom untuk mendeskripsikan kerusakan dan menambahkan foto fasilitas yang rusak, seperti AC di ruang kelas yang disebutkan dalam laporan.

3.6.4. Push Notification





Gambar 10. Push Notification

Fitur notifikasi diimplementasikan menggunakan metode *push notification* dengan bantuan *library Pusher* pada sisi *backend* Laravel, serta integrasi *library flutter_pusher_client* di sisi aplikasi Flutter. Ketika status laporan diperbarui oleh pihak Rumga melalui sistem, Laravel akan memicu *event* yang dikirim melalui *Pusher Channels*, lalu aplikasi Flutter akan menerima dan menampilkan notifikasi kepada pengguna tanpa harus membuka aplikasi terlebih dahulu.

3.7. Blackbox Testing

Pengujian dilakukan menggunakan metode *Black Box Testing*, yaitu dengan menguji fungsi sistem berdasarkan *input* dan *output* tanpa melihat kode program.

Tabel 1. Testing Login

No.	Nama Pengujian	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Mengisi form login field email	Klik field email dan mengisi email tidak sesuai ketentuan	Terdapat kesalahan saat login pada aplikasi	Sesuai Harapan
2.	Mengisi form login field email	Klik field email dan mengisi email sesuai ketentuan	Field password ditampilkan	Sesuai Harapan
3.	Mengisi form login field email dan password	Klik field email dan mengisi email sesuai ketentuan, kemudian klik field password dan tidak mengisi sesuai ketentuan	Terdapat kesalahan saat login pada aplikasi	Sesuai Harapan
4.	Mengisi form login field email dan password	Klik field email dan mengisi email sesuai ketentuan, kemudian klik field password dan mengisi sesuai ketentuan	Berhasil Masuk dan halaman Dashboard ditampilkan	Sesuai Harapan

Tabel 2. Testing Laporan

No.	Nama Pengujian	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Pengujian Form Laporan	User mengisi form laporan dengan data valid dan mengirim laporan	Sistem menyimpan laporan dan menampilkan notifikasi sukses	Sesuai Harapan
2.	Validasi Form Laporan	User mengisi form dengan data kosong atau tidak valid	Sistem menampilkan pesan error sesuai input yang salah	Sesuai Harapan
3.	Pengubahan Status Laporan	Staf Rumga mengubah status laporan menjadi "Diproses" atau "Selesai"	Status laporan berubah sesuai yang dipilih	Sesuai Harapan
4.	Pemberian Feedback	Staf Rumga memberikan feedback setelah mengubah status laporan	Feedback disimpan dan ditampilkan di detail laporan	Sesuai Harapan
5.	Pencarian Laporan	User mencari laporan dengan kata kunci	Sistem menampilkan laporan sesuai kata kunci	Sesuai Harapan
6.	Pencarian Laporan tidak valid	User mencari laporan dengan kata kunci yang salah	Sistem tidak dapat menampilkan laporan karena kata kunci salah	Sesuai Harapan
7.	Hapus Laporan	Staf Rumga menghapus laporan yang ditolak	Sistem menghapus laporan di database	Sesuai Harapan

Tabel 3. Testing Laporan

No.	Nama Pengujian	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
-----	----------------	--------------------	-----------------------	-----------------

1.	Penambahan akun oleh <i>admin</i> yang valid	<i>Admin</i> menambahkan data akun sesuai dengan format dan semua <i>field</i> terisi	Sistem menyimpan data akun dan menampilkan halaman manajemen akun	Sesuai Harapan
2.	Penambahan akun oleh <i>admin</i> yang tidak valid	<i>Admin</i> menambahkan data akun tidak sesuai dengan format dan tidak semua <i>field</i> terisi	Sistem menampilkan pesan <i>error</i> sesuai <i>input</i> yang salah	Sesuai Harapan
3.	Pengubahan data akun oleh <i>admin</i> yang valid	<i>Admin</i> mengedit data akun sesuai dengan format dan semua <i>field</i> terisi	Sistem menyimpan data akun dan menampilkan halaman manajemen akun	Sesuai Harapan
4.	Pengubahan data akun oleh <i>admin</i> yang tidak valid	<i>Admin</i> mengedit data akun tidak sesuai dengan format dan tidak semua <i>field</i> terisi	Sistem menampilkan pesan <i>error</i> sesuai <i>input</i> yang salah	Sesuai Harapan
7.	Hapus Akun	<i>Admin</i> menghapus akun yang sudah tidak aktif	Sistem menghapus akun di <i>database</i>	Sesuai Harapan

Secara keseluruhan, hasil pengujian *blackbox* pada fitur *login*, laporan, dan manajemen akun menunjukkan bahwa aplikasi berfungsi sesuai dengan harapan. Semua skenario pengujian, baik yang melibatkan data valid maupun tidak valid, menghasilkan hasil yang sesuai dengan desain yang diinginkan. Pengujian pada setiap fitur, seperti *login*, pengelolaan laporan, dan manajemen akun, menunjukkan bahwa sistem dapat menangani *input* dengan baik, memberikan *feedback* yang sesuai, serta menyimpan atau menampilkan data dengan benar. Keseluruhan proses pengujian membuktikan bahwa fitur-fitur tersebut berfungsi dengan baik dan valid, memenuhi ekspektasi yang telah ditetapkan.

4. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa sistem pelaporan fasilitas berbasis *multi-platform* telah berhasil dikembangkan menggunakan pendekatan *Rapid Application Development* (RAD). Sistem ini menyediakan dua *platform* utama, yaitu aplikasi *mobile* dan *web*, yang saling terintegrasi untuk memfasilitasi proses pelaporan dan tindak lanjut kerusakan fasilitas secara efisien.

Fitur utama seperti pengelompokan laporan berdasarkan kategori, lokasi, dan deskripsi, serta dukungan unggah foto, membantu memastikan bahwa setiap laporan terdokumentasi dengan jelas dan sistematis. Selain itu, penerapan fitur push notification memungkinkan pengguna menerima pembaruan status laporan secara *real-time*, sehingga meningkatkan transparansi dan komunikasi antara pelapor dan pengelola laporan. Secara keseluruhan, sistem ini mampu meningkatkan efektivitas dan kecepatan dalam pengelolaan laporan fasilitas, serta memberikan pengalaman pengguna yang lebih baik melalui integrasi teknologi yang tepat dan *user-friendly*.

REFERENSI

- [1] N. Sudjud and Mohamad Ali Akbar, "Rancang Bangun Aplikasi Pelaporan Anti Kekerasan Seksual (AKAS) Berbasis Android," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 14, no. 2, pp. 162–169, 2022, doi: 10.37424/informasi.v14i2.182.
- [2] Britney Hanna Maria Siwu, Vania Yori Rampo, and Salaki Reynaldo Joshua, "413-Article Text-1652-1-10-20221125," <https://jurnal.ugp.ac.id/index.php/JURTIE/article/view/413>, vol. 4, no. 2, pp. 1–10, 2022, doi: 10.14716/ijtech.v0i0.0000.
- [3] J. Biostatistik, I. Kesehatan, N. Negari, and T. Eryando, "Analisis Penerimaan Sistem Informasi Pencatatan dan Pelaporan Kasus COVID-19 (Aplikasi Silacak Versi 1.2.5) Menggunakan Technology Acceptance Model (TAM) di UPT Puskesmas Cipadung Kota Bandung," *J. Biostat. Kependudukan, dan Inform. Kesehat.*, vol. 1, no. 3, 2021, doi: 10.7454/bikfokes.v1i3.1015.
- [4] A. D. Prasetyo, I. A. Kautsar, and N. L. Azizah, "Rancang Bangun Aplikasi Pelaporan Fasilitas Umum Berbasis Web Service Dalam Rangka Menuju Sidoarjo Smart City Dan Open Data," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 4, pp. 1271–1280, 2022, doi: 10.29100/jupi.v7i4.3259.
- [5] S. Suyanto and A. Andri, "Implementasi Rapid Application Development Dalam Pengembangan Aplikasi Pelaporan Kerusakan Jalan," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 5, no. 2, p. 89, 2020, doi: 10.29100/jupi.v5i2.1758.
- [6] Y. W. A. Rustam, "Perancangan Aplikasi Mobile Katalog Furniture Menggunakan Teknologi Augmented Reality," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 13, no. 2, pp. 97–122, 2021, doi: 10.37424/informasi.v13i2.121.
- [7] Z. D. Kussasih, H. Tolle, and F. Al Huda, "Pengembangan Aplikasi Bimbingan Akademik Mahasiswa Filkom Menggunakan Pendekatan Multi-Platform dengan Metode Agile System Development," *J. Pengemb. Teknol. ...*, vol. 3, no. 7, pp. 1–7, 2024, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13648>
- [8] D. S. Ramdan, "Aplikasi Desktop Multi Platform Untuk Redis Client Berbasis Teknologi Web Menggunakan Framework ElectronJS dan



- ReactJS,” *J. TEDC*, vol. 14, no. 3, pp. 226–231, 2020.
- [9] A. A. Zikra, M. Reza Kuntara, K. Sari, and A. Penelitian, “Perancangan Sistem Pelaporan Masyarakat untuk Fasilitas Pelayanan Publik di Kota Palu Berbasis Android dengan Metode Waterfall,” *J. Kolaboratif Sains*, vol. 7, no. 7, pp. 2523–2532, 2024, doi: 10.56338/jks.v7i7.5633.
- [10] Nelly Sofi and Riza Dharmawan, “Perancangan Aplikasi Bengkel Csm Berbasis Android Menggunakan Framework Flutter (Bahasa Dart),” *J. Tek. dan Sci.*, vol. 1, no. 2, pp. 53–64, 2022, doi: 10.56127/jts.v1i2.125.
- [11] S. N. Laila and M. F. Azima, “Sistem Pelaporan, Penanganan dan Monitoring Kerusakan Laboratorium Komputer pada Institut Informatika dan Bisnis Darmajaya,” *J. Tek.*, vol. 17, no. 1, pp. 21–34, 2023, [Online]. Available: <https://zenodo.org/record/7927622>
- [12] N. Hidayat and K. Hati, “Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE),” *J. Sist. Inf.*, vol. 10, no. 1, pp. 8–17, 2021.
- [13] Lukman Santoso and Juni Amanullah, “Pengembangan Sistem Informasi Akademik Berbasis Website Menggunakan Metode Rapid Application Development (Rad),” *Elkom J. Elektron. dan Komput.*, vol. 15, no. 2, pp. 250–259, 2022, doi: 10.51903/elkom.v15i2.943.
- [14] S. Sunardi, A. Fadlil, F. Al-anshori, and S. Saifullah, “Information System Development Based-on ERP and RAD Methods: Application For Activities Information Broadcasting,” *JUITA J. Inform.*, vol. 8, no. 2, p. 149, 2020, doi: 10.30595/juita.v8i2.7684.
- [15] M. Raihan, G. Qowindra, and J. Wiratama, “G-Tech : Jurnal Teknologi Terapan Application Development (RAD) Method for the Garment Industry in Indonesia,” vol. 7, no. 2, 2023.