

Rancang Bangun Aplikasi Layanan Adopsi dan Penyelamatan Kucing Berbasis REST-API

Septianto Bagus Hidayatullah¹, Mikhail Shams Afzal Karim², Bahiskara Ananda Arryanto³, Dian Maharani⁴, Salsa Pramudhita Agustiardani⁵, Fawwaz Ali Akbar^{6*}

^{1,2,3,4,5,6} Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

¹22081010178@student.upnjatim.ac.id

²22081010169@student.upnjatim.ac.id

³22081010181@student.upnjatim.ac.id

⁴22081010184@student.upnjatim.ac.id

⁵22081010187@student.upnjatim.ac.id

*Corresponding author email: fawwaz_ali.fik@upnjatim.ac.id

Abstrak— Pertumbuhan populasi kucing liar di perkotaan, khususnya di Surabaya, menimbulkan berbagai permasalahan terkait kesejahteraan hewan dan lingkungan. Komunitas Penyelamat Kucing Terlantar Surabaya (KPKTS) berupaya menanggulangi permasalahan tersebut, namun masih menghadapi kendala dalam pendataan dan proses adopsi yang belum terdigitalisasi. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi Meowcare berbasis web sebagai solusi sistem informasi untuk mempermudah proses adopsi kucing dan pelaporan penyelamatan. Metode pengembangan yang digunakan adalah model Waterfall, dengan tahapan meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Perancangan sistem database menggunakan Conceptual Data Model (CDM) dan Physical Data Model (PDM) sedangkan perancangan berorientasi objek menggunakan Use Case Diagram, Activity Diagram, dan Class Diagram. Aplikasi dikembangkan menggunakan React Native pada sisi antarmuka dan Lumen sebagai backend API. Pengujian dilakukan dengan metode Black Box Testing dan pengujian API menggunakan Postman. Hasil pengujian menunjukkan bahwa seluruh fitur utama, baik untuk pengguna umum maupun admin, berjalan sesuai harapan. Aplikasi ini diharapkan dapat meningkatkan efisiensi proses adopsi, memperluas jangkauan penyelamatan kucing, serta meningkatkan kesadaran masyarakat mengenai kesejahteraan hewan.

Kata Kunci— aplikasi adopsi kucing, REST-API, KPKTS, waterfall, komunitas hewan, sistem informasi.

I. PENDAHULUAN

Pertumbuhan populasi kucing yang pesat dan tidak terkendali telah menjadi masalah baru di Kota Surabaya, khususnya lingkungan dan kesejahteraan hewan. Kurangnya kepedulian terhadap hewan terlantar menyebabkan kesejahteraan hewan menjadi terabaikan [1]. Hal ini disebabkan oleh minimnya pemahaman masyarakat mengenai kesejahteraan hewan dan lemahnya implementasi hukum dalam mengatur pelanggaran tersebut [2]. Selain itu, populasi kucing yang tidak terkendali seringkali berakhir menjadi hama untuk lingkungan.

Populasi kucing yang berkembang tanpa pengawasan dan minimnya akses terhadap layanan adopsi membuat kondisi ini semakin memprihatinkan. Seringkali dijumpai kucing hidup dalam kondisi yang tidak layak seperti mengalami kelaparan, penyakit, hingga kekerasan dari manusia. Melihat keprihatinan tersebut, terbentuklah komunitas pecinta kucing yang berfokus

pada penyelamatan kucing jalanan. Salah satu komunitas yang aktif di wilayah Surabaya, yaitu KPKTS (Komunitas Penyelamat Kucing Terlantar Surabaya).

KPKTS dibentuk atas dasar rasa kepedulian terhadap kucing terlantar yang disebabkan oleh rusaknya ekosistem dan kurangnya edukasi masyarakat mengenai pentingnya keberadaan kucing dan hewan lainnya [3]. KPKTS berfokus pada penyelamatan, pemberian makan kucing jalanan, sterilisasi, hingga pencarian adopter yang layak. Selain berdampak pada kesejahteraan hewan, populasi kucing liar yang tidak terkendali dapat memicu masalah lingkungan dan kesehatan masyarakat. Kucing jalanan sering menjadi vektor penyebaran penyakit zoonosis, seperti toksoplasmosis dan rabies, yang dapat membahayakan manusia [4]. Dalam upaya mengurangi pertumbuhan populasi hewan terlantar, edukasi berperan penting untuk menumbuhkan kepedulian masyarakat dan mendorong keterlibatan mereka dalam proses adopsi [5]. Berbagai individu dan komunitas penyelamat hewan memiliki peran penting dalam mengurangi jumlah hewan terlantar. Mereka secara sukarela memberikan waktu, tenaga, dan sumber daya untuk menyediakan perawatan serta tempat tinggal yang layak bagi hewan, sehingga memberi peluang bagi hewan tersebut untuk memperoleh kehidupan yang lebih baik [6].

Namun, dalam praktiknya, proses dokumentasi, pemantauan, dan penyaluran adopsi masih banyak dilakukan secara manual atau terbatas melalui media sosial, yang rawan menyebabkan data tidak terkelola dengan baik, serta kurangnya sistem informasi yang mendukung proses tersebut secara efisien dan transparan. Pemanfaatan sistem digital dalam layanan adopsi dan penyelamatan hewan memungkinkan pengelolaan data yang lebih terstruktur, mempercepat proses administrasi, serta memperluas jangkauan informasi ke masyarakat luas [7].

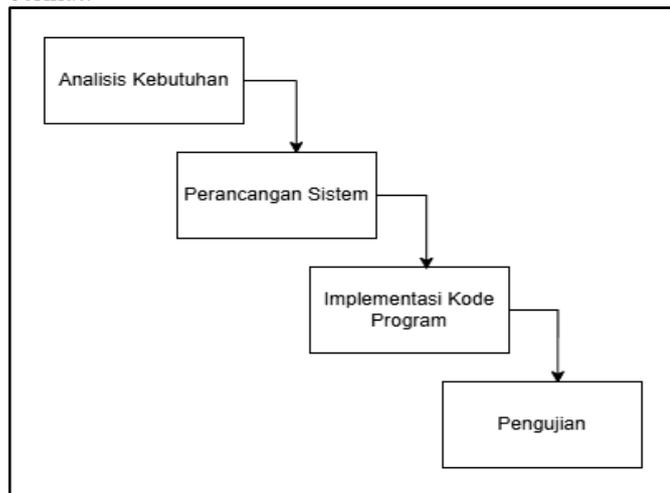
Berdasarkan kondisi tersebut, tantangan utama yang dihadapi yaitu membangun sistem yang menunjang proses adopsi dan pelaporan penyelamatan kucing secara efisien dan terstruktur. Selain itu, dibutuhkan sistem yang dirancang agar mudah diakses dan digunakan oleh pengguna, seperti dalam bentuk aplikasi mobile. Aplikasi mobile adalah sebuah sistem perangkat lunak yang dirancang untuk dapat berjalan dalam perangkat bergerak, seperti smartphone [8]. Salah satu pendekatan teknis yang umum digunakan untuk masalah tersebut yaitu pemanfaatan REST API. REST

(Representational State Transfer) merupakan gaya arsitektur pembuatan sistem untuk memudahkan komunikasi antar perangkat dengan menggunakan pola komunikasi client-server, yaitu client mengirimkan permintaan ke server dan server akan menerima permintaan tersebut dengan mengembalikan response [9]. REST-API sendiri dapat diartikan sebagai pertukaran data antara aplikasi dan server dalam format JSON (*JavaScript Object Notation*).

Dalam penelitian ini, penulis mengusulkan perancangan aplikasi mobile Meowcare yang dibangun dengan memperhatikan kebutuhan KPKTS sebagai mitra komunitas. Sistem ini dibangun tidak hanya untuk memfasilitasi adopsi kucing dari KPKTS, namun juga dapat diakses oleh masyarakat umum yang sedang mencari hewan peliharaan. Dengan adanya penelitian ini, diharapkan dapat membantu komunitas dalam menyelamatkan lebih banyak kucing dan mempermudah proses adopsi untuk pecinta kucing.

II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan model Waterfall sebagai metode pengembangan sistem. Metode waterfall merupakan metode perancangan dan pengembangan perangkat lunak yang dilakukan secara berurutan dan tidak saling tumpang tindih. Setiap tahap dalam proses pengembangan harus diselesaikan terlebih dahulu sebelum mengerjakan fase selanjutnya [10]. Waterfall menekankan dokumentasi terperinci pada setiap tahap pengembangan untuk memastikan pemahaman menyeluruh terhadap kebutuhan proyek, spesifikasi desain, dan pedoman implementasi. Hal ini bertujuan untuk memudahkan pemeliharaan dan pembaruan di masa depan [11]. Adapun tahapan model Waterfall yang diterapkan dalam pengembangan aplikasi ini ditunjukkan dalam Gbr.1, sebagai berikut:



Gbr. 1. Model *Waterfall*

A. Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan untuk mengidentifikasi kebutuhan sistem dan pengguna, baik kebutuhan fungsional maupun non-fungsional. Tujuan utama analisis kebutuhan yaitu untuk menentukan fitur utama dan alur kerja sistem sesuai dengan kebutuhan nyata di lapangan.

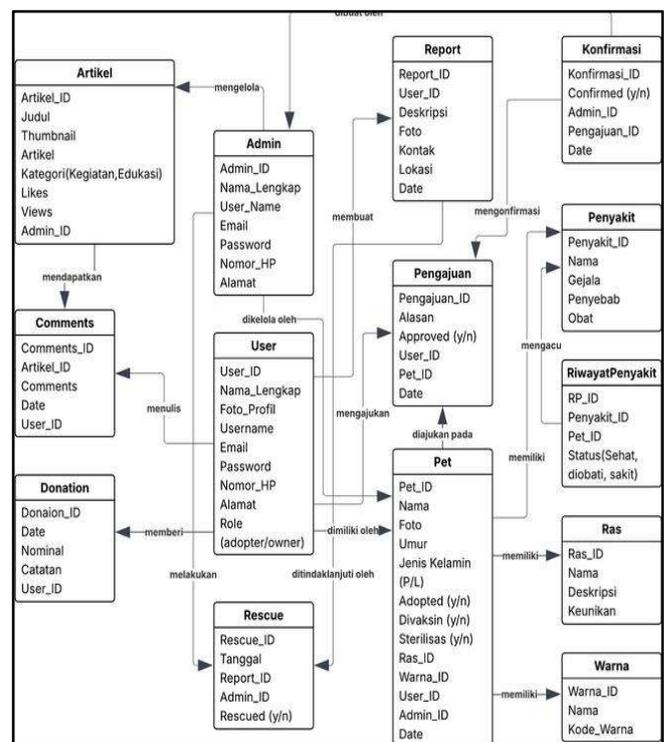
Kebutuhan fungsional merupakan persyaratan yang harus dilakukan oleh sistem agar dapat digunakan dengan baik. Dalam sistem ini, kebutuhan fungsional mencakup autentikasi pengguna, fitur adopsi kucing, pelaporan penyelamatan kucing, dan artikel edukasi. Sedangkan, kebutuhan non-fungsional merupakan persyaratan pendukung sistem untuk meningkatkan performa sistem dan pengalaman pengguna, diantaranya keamanan, aksesibilitas, skalabilitas, responsif, dan kemudahan penggunaan.

B. Perancangan Sistem

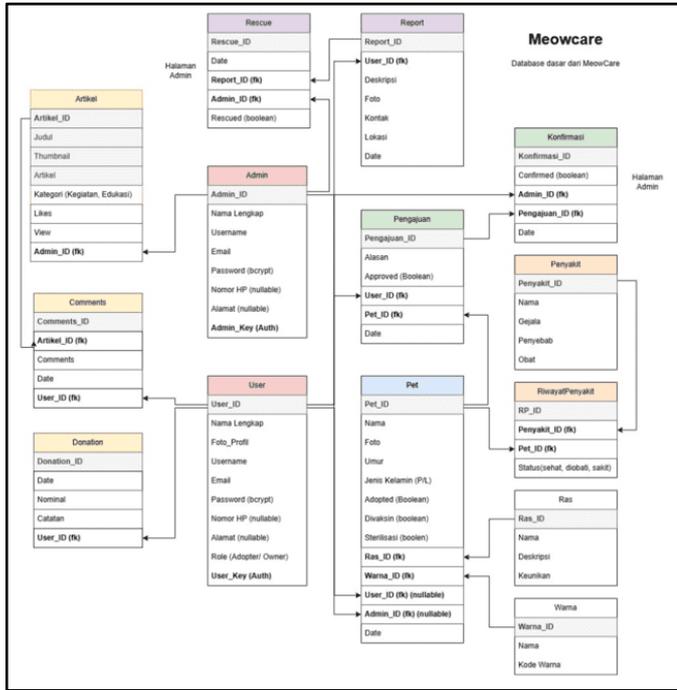
Perancangan sistem merupakan tahapan perencanaan untuk menyelesaikan masalah sebagai solusi perangkat lunak [8]. Proses ini mencakup desain algoritma arsitektur, antarmuka user, dan rancangan basis data. Dalam proses perencanaan ini terdapat beberapa perangkat perancangan sistem, seperti use case diagram, diagram relasi entitas, dan diagram activity.

1) Perancangan Basis Data

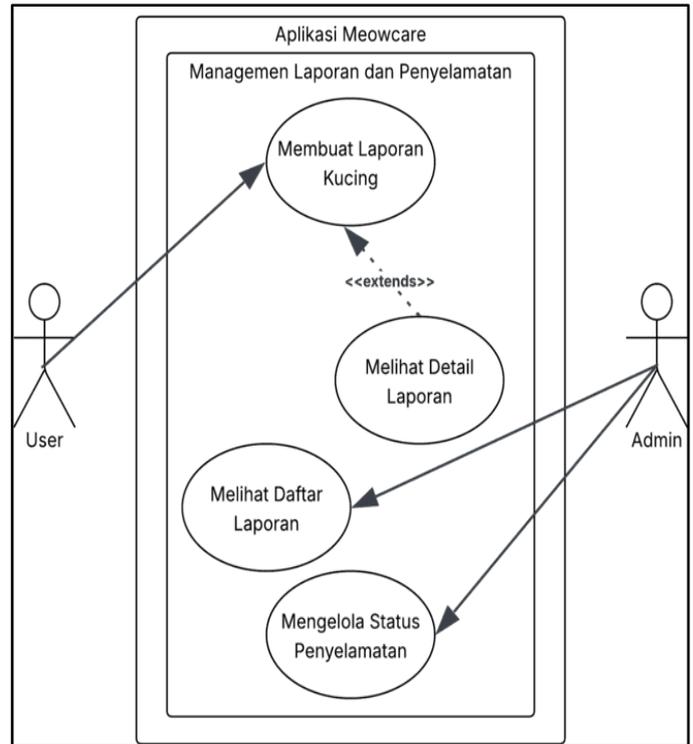
Basis data merupakan sekumpulan data yang menggambarkan aktivitas dari satu entitas atau lebih yang saling terhubung [12]. Hubungan antara entitas dalam basis data divisualisasikan melalui Entity Relationship Diagram (ERD). ERD merupakan tahap awal perancangan basis data yang menunjukkan secara detail hubungan atau relasi antar entitas (atribut) agar terbentuk secara terstruktur. Setelah perancangan ERD tersebut, pertama dikembangkan menjadi Conceptual Data Model untuk menggambarkan awal rangkaian dari relasi antar tabel, seperti yang ditunjukkan pada Gbr. 2, lalu dari CDM dikembangkan menjadi Physical Data Model yang menggambarkan struktur basis data secara teknis dan siap diimplementasikan. PDM mencakup nama tabel, kolom dan tipe data [13], seperti yang ditunjukkan pada Gbr. 3.



Gbr. 2. *Conceptual Data Model (CDM)* Aplikasi Meowcare



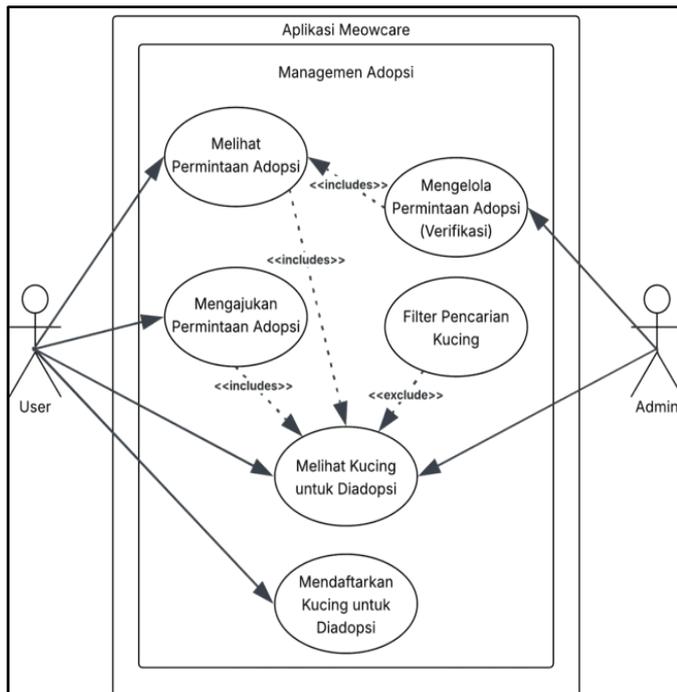
Gbr. 3. Physical Data Model (PDM) Aplikasi Meowcare



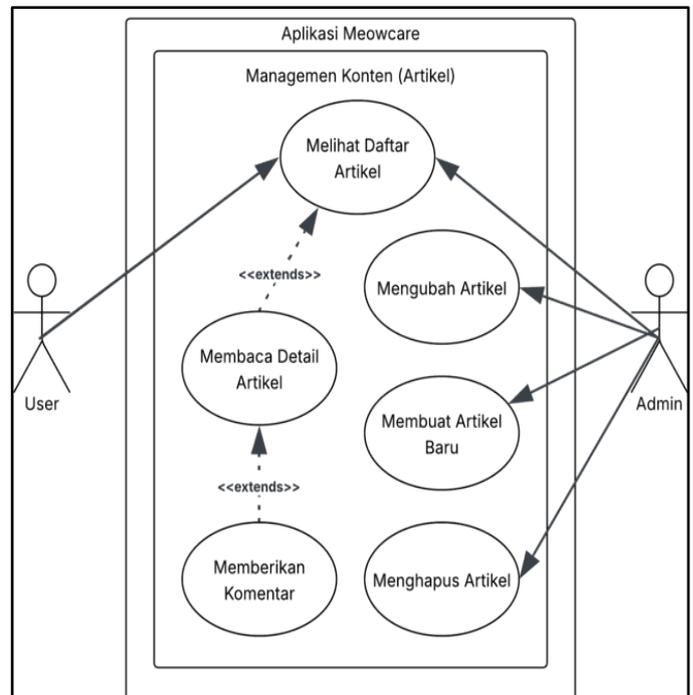
Gbr. 5. Use Case Diagram Managemen Laporan dan Penyelamatan

2) Use Case Diagram

Use Case Diagram adalah suatu jenis pemodelan yang menggambarkan sisi perilaku dari sistem yang dirancang [14]. Diagram ini mencakup berbagai fitur yang ada dalam sistem, memudahkan pengguna (aktor) dalam memahami cara mereka berinteraksi dengan sistem, seperti pada Gbr. 4 yang menunjukkan interaksi antara user dan admin dalam manajemen adopsi, Gbr. 5 pada manajemen laporan dan penyelamatan, dan Gbr. 6 pada manajemen artikel.



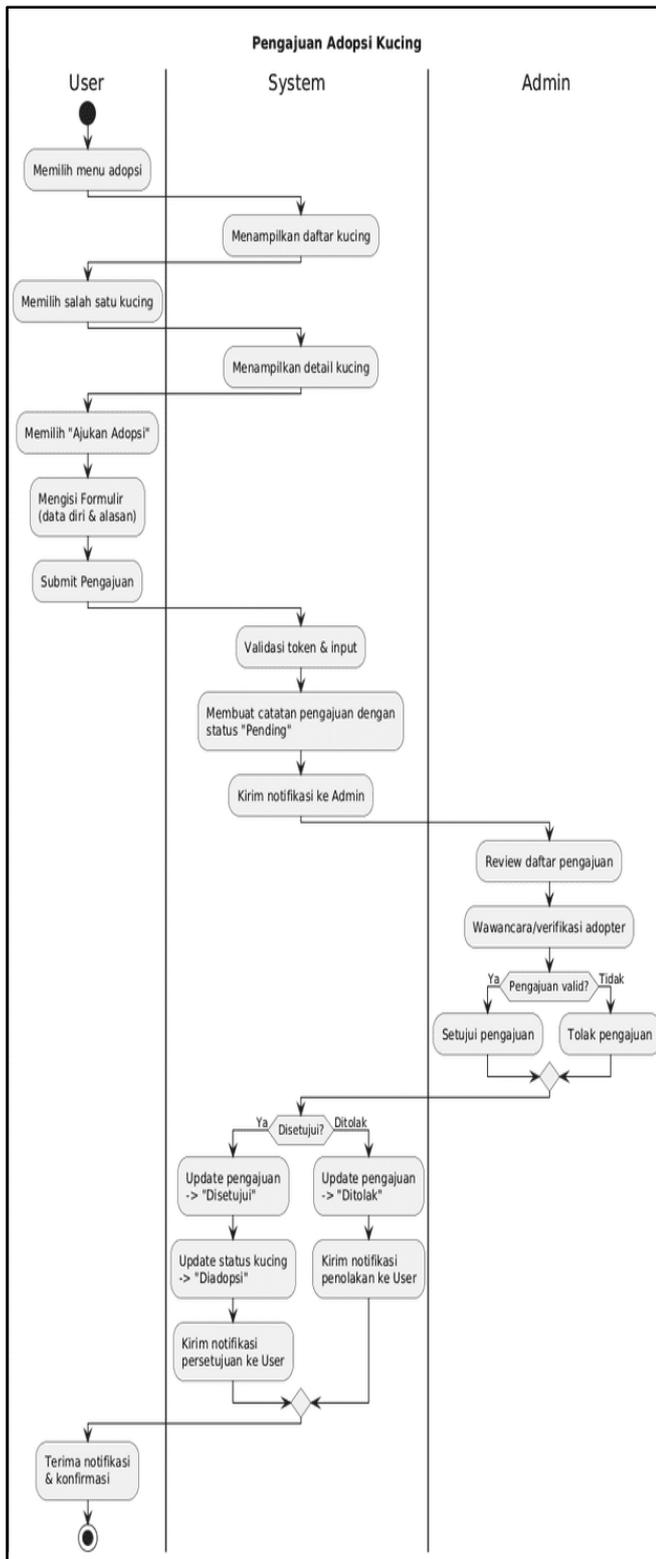
Gbr. 4. Use Case Diagram Managemen Adopsi



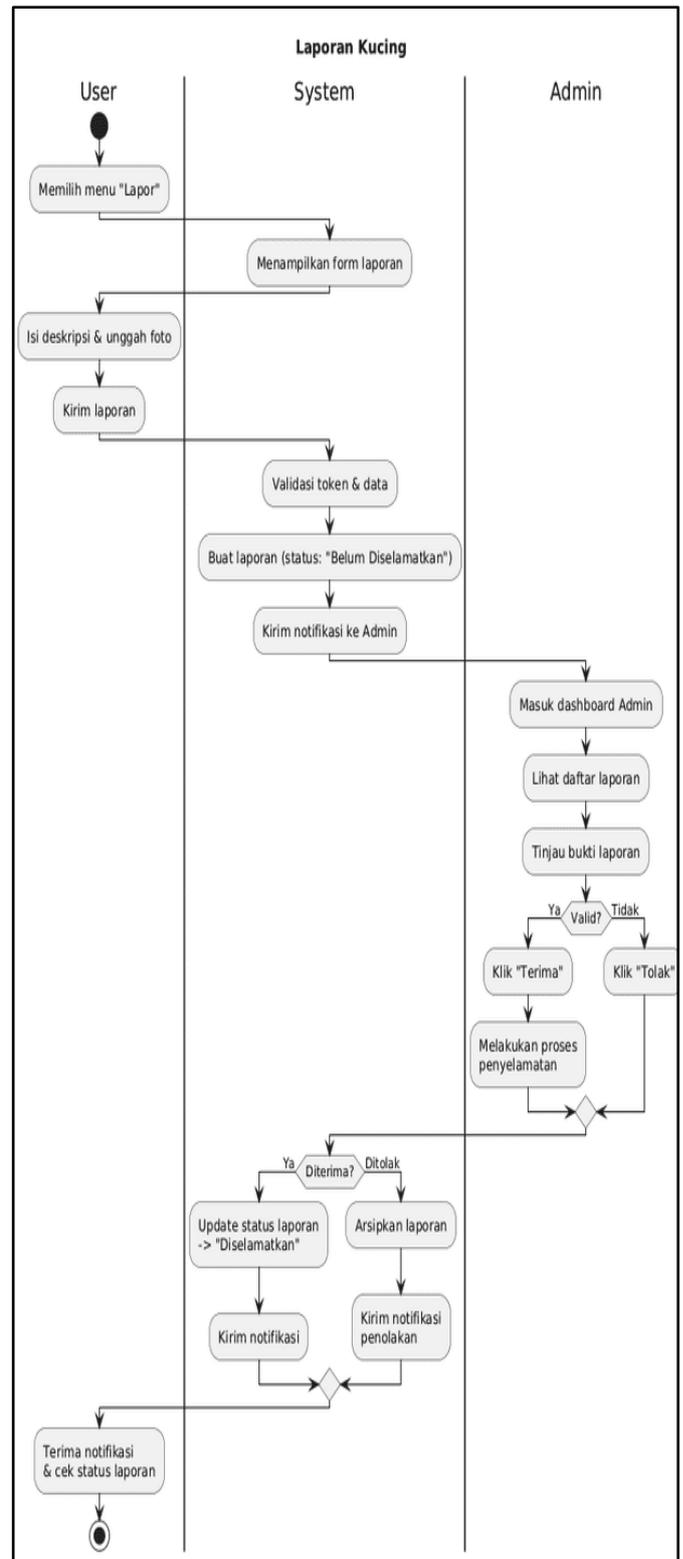
Gbr. 6. Use Case Diagram Managemen Artikel

3) Activity Diagram

Diagram Aktivitas adalah sebuah diagram yang mampu menggambarkan alur suatu sistem dalam bentuk prosedural [15]. Diagram ini mencakup rangkaian tindakan yang dilakukan baik oleh sistem maupun oleh pengguna. Dalam Gbr. 7, digambarkan rangkaian aktivitas pengguna dalam sistem adopsi kucing.



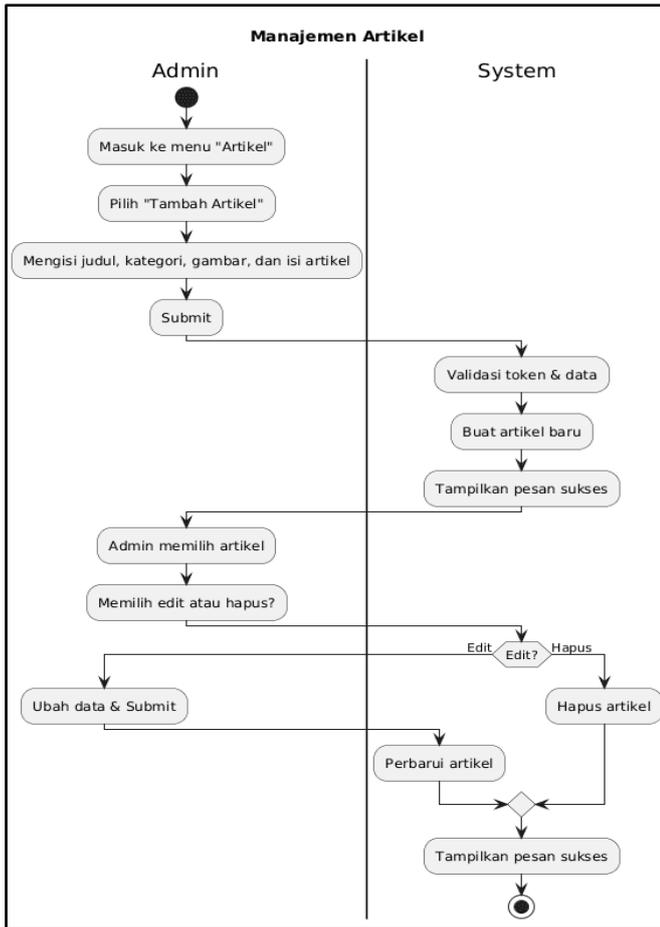
Gbr. 7. Activity Diagram Sistem Adopsi



Gbr. 8. Activity Diagram Laporan Kucing

Selain proses pengajuan adopsi, sistem juga mendukung beberapa fitur lain yang melibatkan interaksi pengguna dan admin. Salah satunya adalah fitur pelaporan kucing yang ditemukan oleh masyarakat. Aktivitas pengguna dan alur kerja sistem dalam menangani laporan tersebut digambarkan dalam Gbr. 6.

Di samping itu, sistem menyediakan panel manajemen artikel yang memungkinkan admin untuk mengelola artikel informatif terkait adopsi, perawatan hewan, dan aktivitas KPKTS. Proses ini melibatkan pembuatan, pengeditan, dan penghapusan artikel oleh admin, seperti ditunjukkan pada Gbr. 9.



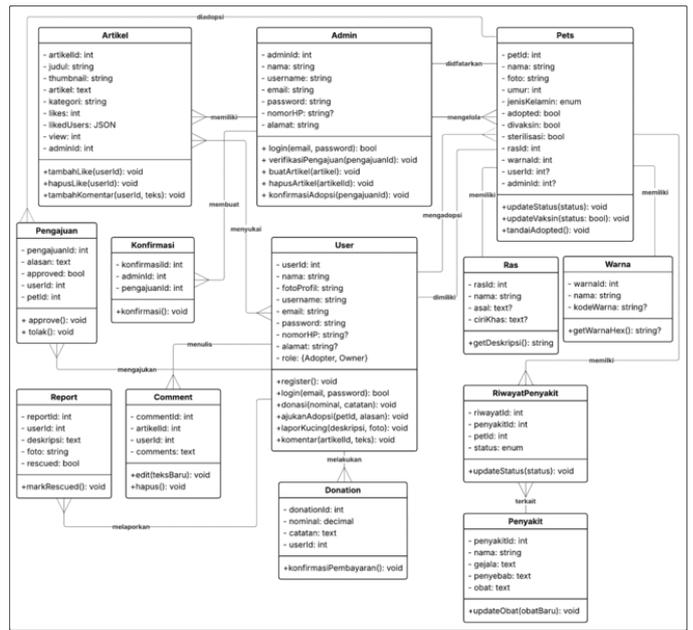
Gbr. 9. Activity Diagram Manajemen Artikel

4) Class Diagram

Class Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan struktur statis dari sistem dengan merepresentasikan kelas-kelas yang ada beserta atribut dan operasinya. Class Diagram dapat memberikan sebuah gambaran yang luas lagi tentang sebuah sistem dengan memberikan petunjuk kepada kelas serta dalam hubungannya [16]. Diagram ini juga menampilkan relasi antar kelas seperti asosiasi, agregasi, dan dependensi, yang membantu dalam memahami bagaimana setiap komponen dalam sistem saling terhubung secara logis. Pada sistem aplikasi MeowCare, class diagram menggambarkan entitas penting seperti User, Admin, Pet, Artikel, dan lain-lain, beserta interaksinya dalam proses adopsi, penulisan artikel, donasi, pelaporan, dan pengelolaan data kucing.

Seperti yang digambarkan dalam Gbr. 10, relasi dalam class diagram ini mencerminkan hubungan nyata dalam basis data, seperti satu Admin dapat membuat banyak Artikel, dan satu User dapat melakukan banyak Pengajuan adopsi atau Report. Selain atribut yang merepresentasikan data tiap entitas, class diagram juga dilengkapi dengan operasi (method) untuk menunjukkan fungsi atau perilaku dasar dari tiap kelas, seperti validasi data atau penghitungan statistik. Dengan begitu, diagram ini berperan penting dalam merancang arsitektur

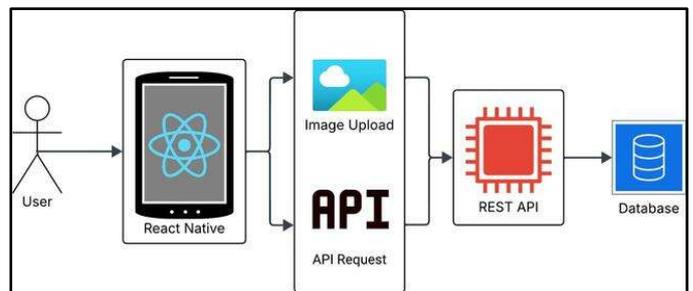
sistem yang terstruktur dan menjadi acuan utama dalam proses implementasi aplikasi MeowCare.



Gbr. 10. Class diagram

5) Arsitektur Sistem

Aplikasi Meowcare menggunakan arsitektur client-server berbasis API, dengan frontend mobile dibangun menggunakan React Native, dan backend dibangun menggunakan Laravel API dengan autentikasi Sanctum. Semua komunikasi antara frontend dan backend menggunakan HTTP request (fetch/Axios). Aplikasi ini menyimpan data dalam database MySQL[16] serta file sistem Laravel. Dalam Gbr. 11, digambarkan alur dari arsitektur aplikasi MeowCare.



Gbr. 11. Arsitektur sistem

C. Implementasi Kode Program

Tahap implementasi kode program merupakan proses mengubah desain sistem menjadi aplikasi yang fungsional. Pada penelitian ini, frontend atau antarmuka pengguna dikembangkan menggunakan React Native untuk mendukung platform mobile. Sementara itu, untuk teknologi backend yang digunakan adalah Lumen sebagai framework untuk membuat RESTful API yang menangani berbagai permintaan data seperti autentikasi, adopsi kucing, laporan, dan profil. Penggunaan API ini berperan sebagai penghubung antara frontend dengan basis data. Perancangan dan pengelolaan basis data pada pengembangan aplikasi ini menggunakan SQL guna

memastikan penyimpanan data dilakukan secara terstruktur dan efisien.

a. React Native (Antarmuka)

React Native merupakan framework pembangunan antarmuka aplikasi berbasis Javascript yang memungkinkan pengembang untuk membuat dan menggunakan komponen UI yang efisien [17] [18]. Fitur-fitur dalam aplikasi seperti daftar kucing, daftar artikel, kamera, dan navigasi dibangun menggunakan komponen React Native. Hal ini bertujuan untuk meningkatkan pengalaman pengguna agar tetap intuitif dan responsif.

b. Lumen (Backend)

Lumen merupakan sebuah -framework yang berbasiskan PHP yang ditujukan khusus untuk menciptakan layanan RESTful API yang efisien dan cepat [19]. Arsitektur Lumen menunjukkan kemiripan dengan Laravel, namun lebih ringkas dan ringan karena tidak memuat semua fitur yang dimiliki Laravel secara keseluruhan. Lumen mendukung banyak fitur krusial seperti pengaturan routing, middleware, validasi, serta autentikasi, dan juga dapat ditingkatkan menjadi aplikasi Laravel yang lengkap apabila diperlukan.

c. Rest-API (Integrasi)

Rest-API merupakan penghubung komunikasi antara frontend dan backend. Fitur-fitur di antarmuka pengguna mengirim dan menerima data melalui endpoint GET, POST, PUT, dan DELETE.

D. Pengujian

Setelah tahap implementasi, pengujian sistem dilakukan untuk memastikan bahwa semua fungsi bekerja sesuai dengan kebutuhan yang telah ditetapkan. Pengujian ini menggunakan metode Black Box Testing yang berfokus pada pengujian dari fungsionalitas elemen utama aplikasi, seperti formulir adopsi, unggahan data kucing, dan mekanisme masuk. Selain itu, pengujian API juga dilaksanakan untuk memastikan bahwa setiap endpoint yang dibuat dengan menggunakan kerangka kerja Lumen beroperasi dengan benar, memberikan respons yang tepat, serta mampu mengelola masukan dan kesalahan dengan baik. Pengujian API tersebut mencakup proses autentikasi, pengambilan serta penyimpanan informasi kucing, pengiriman formulir adopsi, dan fitur komunitas. Hasil dari pengujian ini dipergunakan untuk mendeteksi dan memperbaiki kesalahan, serta menyempurnakan sistem sebelum dirilis untuk digunakan oleh masyarakat.

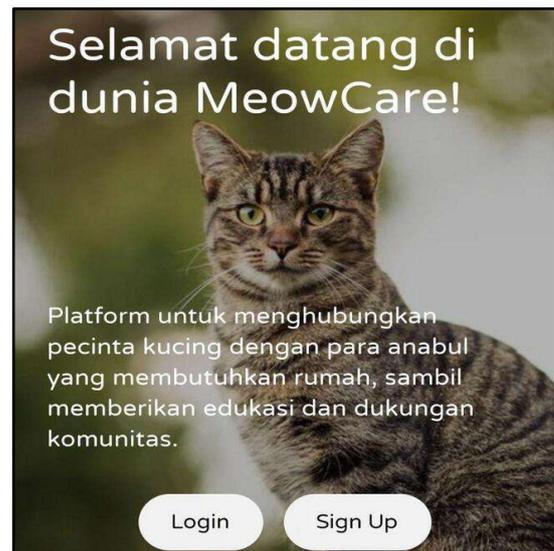
III. HASIL DAN PEMBAHASAN

Dalam bab ini, difokuskan pada hasil yang diperoleh dari masing-masing tahapan yang telah dibahas pada bab sebelumnya, mulai dari proses perancangan sistem, implementasi, hingga tahap pengujian. Setiap hasil dianalisis untuk mengetahui sejauh mana sistem yang dikembangkan telah memenuhi tujuan yang telah ditetapkan, serta untuk mengevaluasi efektivitas dan keandalan dari metode yang digunakan dalam penelitian ini.

A. Implementasi Sistem

Untuk menilai keberhasilan pengembangan aplikasi, dilakukan analisis terhadap tampilan visual aplikasi dan fitur-fitur di dalamnya serta hasil dari pengujian untuk memvalidasi apakah aplikasi berjalan sesuai dengan spesifikasi yang telah

ditentukan pada perancangan awal. Pada Gbr. 12 menunjukkan halaman landing dari aplikasi.



Gbr. 12. Tampilan Splash

1. Tampilan Register

Hasil implementasi halaman SignIn aplikasi Meowcare pada Gbr. 13 menunjukkan bahwa proses autentikasi berjalan sesuai dengan fungsinya. Pengguna diwajibkan memasukkan email dan password yang telah didaftarkan untuk dapat mengakses aplikasi. Jika data login yang dimasukkan valid, sistem akan mengarahkan pengguna ke halaman utama aplikasi dan pengguna dapat melakukan registrasi pada halaman SignUp.

Gbr. 13. Tampilan Register

2. Tampilan Login

Pada Gbr. 14 menunjukkan halaman login dimana useryang telah melakukan registrasi dapat memasukkan email dan password yang sudah dibuat, sehingga jika datanya sudah valid maka proses login akan berhasil dan akan masuk ke home aplikasi meowcare

Gbr. 14. Tampilan Login

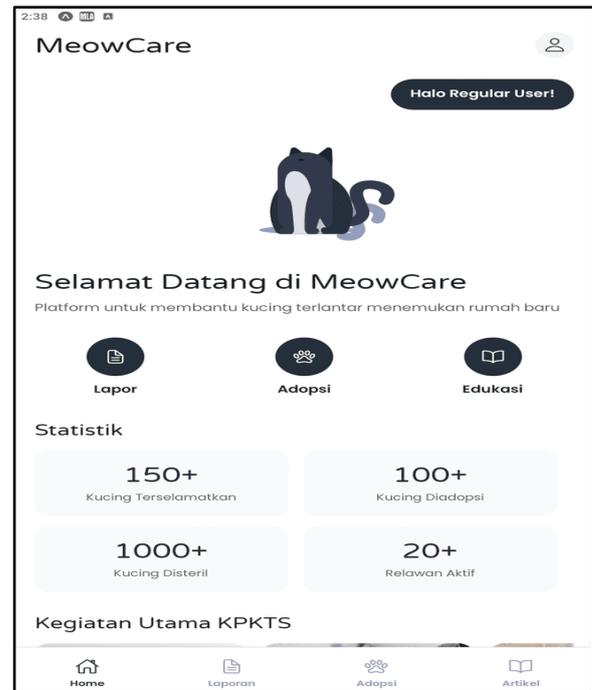
3. Tampilan Lupa Password

Pada Gbr. 15 menunjukkan halaman lupa password, dimana user dimintai untuk memasukkan email yang ingin diperbarui passwordnya, setelah itu user akan diarahkan untuk mengganti password dari email yang ia masukkan.

Gbr. 15. Tampilan Lupa Password

4. Tampilan Home User

Pada Gbr. 16 menunjukkan halaman home user untuk melihat dan mengakses beberapa fitur aplikasi meowcare, seperti laporan, adopsi, artikel, dan profil. adapun informasi mengenai aplikas meowcare dan mitra kpkts.



Gbr. 16. Tampilan Home User

5. Tampilan Laporan

Pada Gbr. 17 menunjukkan halaman laporan, dimana user dapat menambahkan data kucing yang ingin dilaporkan dengan memasukkan nama pelapor, email, nomor telepon, deskripsi mengenai kucing yang akan dilaporkan, kemudian mengupload gambar kucing untuk peninjauan data, agar dapat dilakukan proses adopsi.

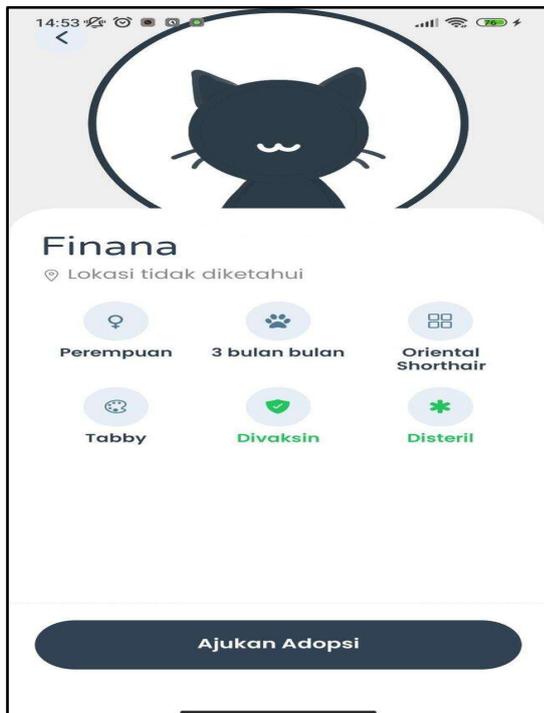
Gbr. 17. Tampilan Laporan

6. Tampilan Adopsi

Gbr. 18 menunjukkan halaman adopsi yang menampilkan daftar kucing yang siap untuk di adopt oleh user, dengan menampilkan data kucing berdasarkan nama kucing, jenis kelamin, ras, umur, dan warna kucing, serta terdapat alamat kucing yang dapat membantu user agar lebih mudah dalam proses adopsi. Untuk mengajukan adopsi user diharuskan untuk mengisi form untuk ditinjau lebih lanjut oleh owner untuk memastikan apakah user layak untuk melakukan adopsi.



Gbr. 18. Tampilan Adopsi



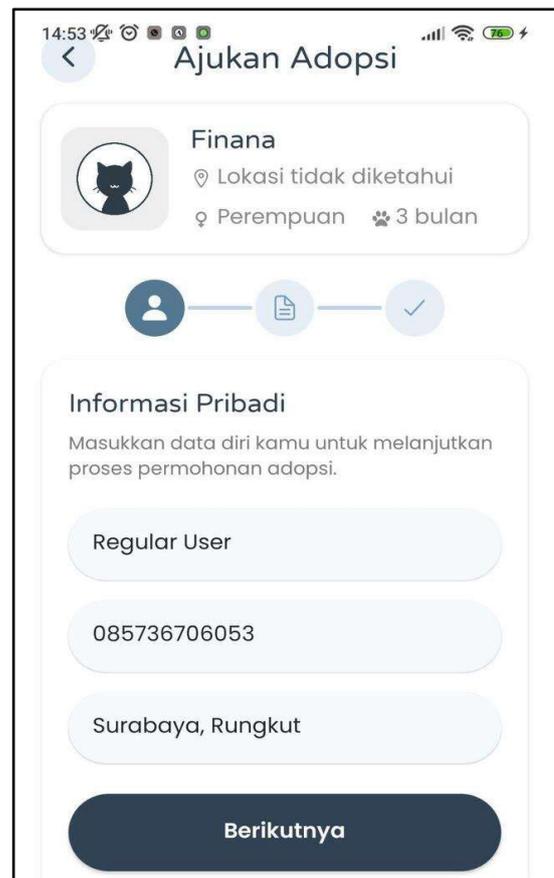
Gbr. 19. Tampilan Detil Kucing

7. Tampilan Detail Kucing

Pada Gbr. 19 menunjukkan halaman detail kucing akan berisi detail dari kucing yang dipilih oleh user, seperti namanya, usia, ras, warna, status vaksinasi, dan sebagainya. Selain itu juga akan terdapat tombol adopsi untuk user dapat mengadopsi kucing tersebut.

8. Tampilan Form Adopsi Kucing

Pada halaman ini, user akan diminta untuk melengkapi data pribadi miliknya dan mengisi alasan ia ingin mengadopsi kucing yang ia pilih. Setelah itu, permohonan untuk adopsi akan dikirimkan kepada admin untuk disetujui. Halaman seperti yang ditunjukkan pada Gbr. 20.



Gbr. 20. Tampilan Form Adopsi Kucing

9. Tampilan Tambah Kucing

Pada halaman tambah kucing, user dapat menambahkan kucing yang akan diadopsi dengan memasukan data nama kucing, umur kucing, jenis kelamin, ras, warna, dan dapat menambahkan deskripsi dari kucing yang akan ditambahkan. Halaman seperti yang ditunjukkan pada Gbr. 21.

10. Tampilan Artikel User

Halaman detail artikel berisi detail dari artikel yang dipilih oleh user dengan adanya fitur view, likes, dan juga komentar untuk meningkatkan interaksi antar user dan juga admin. Halaman seperti yang ditunjukkan pada Gbr. 22

Gbr. 21. Tampilan Tambah Kucing

Gbr. 22. Tampilan Detil Artikel User

11. Tampilan Profil

Pada halaman profil menampilkan data dari pengguna seperti nama, username, email, nomor telepon, alamat, dan foto profil. pengguna juga dapat mengupdate data sesuai dengan kebutuhan. kemudian, terdapat tampilan activity. Halaman seperti yang ditunjukkan pada Gbr. 23.

Gbr. 23. Tampilan Profil

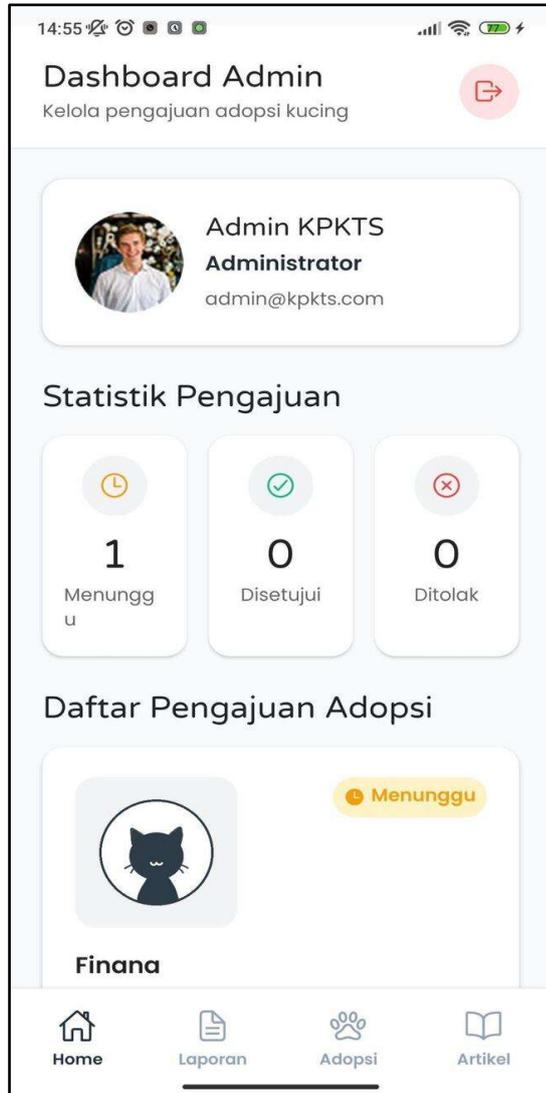
12. Tampilan Edit Profil

Halaman edit profil user ini berfungsi untuk melengkapi atau mengedit data dari user yang meliputi, Username, Nama Lengkap, Email, Nomor Telepon, Alamat, dan Foto Profil. Halaman seperti yang ditunjukkan pada Gbr. 24.

Gbr. 24. Tampilan Edit Profil

13. Dashboard Admin/ Daftar Pengajuan Adopsi

Dashboard admin ini berisi daftar-daftar dari adopsi yang diajukan oleh user, dengan status menunggu, disetujui, dan ditolak. Ketika pengajuan user diterima, status akan berubah menjadi disetujui dan admin tidak dapat menggantinya kembali. Halaman seperti yang ditunjukkan pada Gbr. 25.



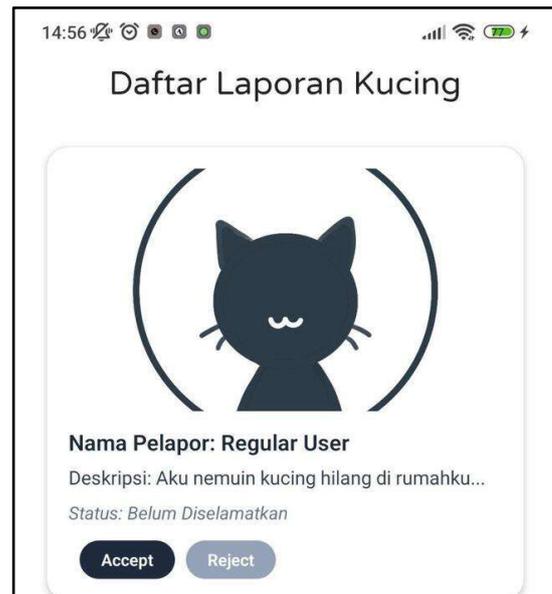
Gbr. 25. Dashboard Admin

14. Dashboard Laporan Admin

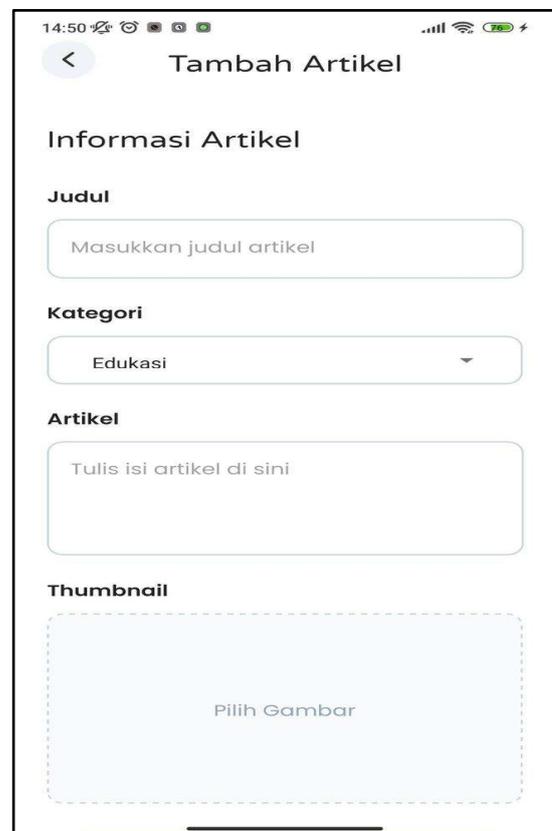
Sama seperti pada dashboard daftar pengajuan adopsi, pada dashboard laporan admin berisi daftar laporan kucing dari user. Pada halaman ini, admin dapat mengkonfirmasi laporan kucing tersebut untuk menindaklanjuti kembali laporan kucing tersebut. Halaman seperti yang ditunjukkan pada Gbr. 26.

15. Tampilan Tambah Artikel Admin

Di halaman ini, admin dapat menambahkan artikel baru, mengisi isi, kategori, thumbnail, dari artikel baru. Setelah artikel baru ditambahkan, artikel akan muncul di halaman artikel user maupun admin. Halaman seperti yang ditunjukkan pada Gbr. 27.



Gbr. 26. Dashboard Laporan Admin



Gbr. 27. Tampilan Tambah Artikel Admin

B. Implementasi Rest API

Rest API adalah arsitektur pertukaran data menggunakan protokol HTTP untuk mengakses, mengambil, dan menggunakan data dari server API secara langsung

menggunakan *method* GET, POST PUT, DELETE. Pada Tabel 1, menunjukkan daftar *endpoint* yang dikembangkan.

Tabel 1. Endpoint API

Endpoint	Method	Parameter	Keterangan
/users/{id}	GET	ID user	Memperoleh data user berdasarkan id
/users/register	POST	Nama lengkap, <i>username</i> , email, <i>password</i> , foto profil	Membuat data user
/users/upload_photo	POST	Nama lengkap, <i>username</i> , foto profil	Membuat data user dengan foto profil
/users/login	POST	email, <i>password</i>	Melakukan proses autentikasi
/admins/{id}	GET	ID admin	Memperoleh data admin berdasarkan id
/admins/register	POST	Nama lengkap, <i>username</i> , email, <i>password</i>	Membuat data admin
/admins/login	POST	Email, password	Melakukan proses autentikasi
/ras/{id}	DELETE	ID ras, nama, asal, penyebab	Menghapus data ras kucing berdasarkan id
/penyakit/{id}	DELETE	ID penyakit, nama, gejala, penyebab, obat	Menghapus data penyakit kucing berdasarkan id
/warna/{id}	DELETE	ID ras, nama, kode warna	Menghapus data warna kucing berdasarkan id
/pets	POST	Nama Umur Jenis_Kelamin Ras_ID Warna_ID Foto	Membuat data kucing
/comments	GET	-	Memperoleh semua data <i>comments</i>

	POST	Artikel_ID Comments	Membuat <i>comments</i>
/artikel	POST	Judul Artikel Kategori Thumbnail	Membuat data artikel
/artikel/likes/{artikelID}	PUT	-	Memperbarui artikel yang disukai berdasarkan id
/pengajuan	POST	Alasan Pet_ID	Memperbarui data pengajuan
/report/	POST	Deskripsi Foto	Membuat data report
/report/listreport	GET	-	Memperoleh semua data report

Untuk membangun aplikasi mobile, digunakan framework React Native dengan TypeScript yang berperan sebagai antarmuka pengguna (frontend). Sedangkan untuk menangani *request* dari aplikasi dan memberikan *response* dari server, digunakan Lumen sebagai backend yang ringan dan efisien. Lumen berfungsi untuk mengatur routing HTTP dan memproses berbagai operasi CRUD melalui serangkaian endpoint API. Di sisi server, Controller dibuat untuk menangani logika pengolahan data yang dikirimkan maupun diminta oleh pengguna. Gbr. 28 menunjukkan contoh implementasi endpoint */users* menggunakan metode GET pada controller Lumen.

```

0 references | 0 implementations
class UserController extends Controller
{
  0 references | 0 overrides
  public function getAll(): mixed
  {
    return response()->json(User::all());
  }

  0 references | 0 overrides
  public function getById($id): mixed
  {
    $user = User::find($id);
    return $user ? response()->json($user) : response()->json(['message' => 'User not found'], 404);
  }
}

```

Gbr. 28. Contoh Method Get pada Controller

Penggunaan *method* GET yang ditunjukkan pada Gbr 1, digunakan untuk menampilkan seluruh data pengguna yang dapat diakses melalui endpoint tertentu yang telah dirancang sebelumnya. Pada contoh ini, endpoint yang digunakan adalah `http://localhost:8000/users`. Bagian awal URL, yaitu `localhost`, menunjukkan bahwa aplikasi masih dijalankan secara lokal. Angka 8000 merupakan nomor port yang digunakan oleh server backend, dalam hal ini Lumen. Sementara itu, bagian `/users` merupakan path yang mengarah pada layanan API untuk mengambil data pengguna. Endpoint ini akan diproses oleh fungsi yang terdapat dalam controller. Gambar 3 memperlihatkan hasil dari pemanggilan endpoint tersebut, di mana data yang dikembalikan berada dalam format JSON. Data ini nantinya akan diolah dan ditampilkan oleh aplikasi mobile di sisi klien.

IV. PENGUJIAN SISTEM

Pengujian sistem dilakukan untuk mengetahui seberapa berhasil sebuah aplikasi yang sedang dikembangkan berjalan sesuai dengan kriteria dan fungsionalnya. Dalam pengujian dilakukan evaluasi untuk memastikan bahwa aplikasi sudah siap digunakan. Adapun hasil pengujian dari masing-masing halaman sistem dijelaskan pada penjabaran berikut.

1. Pengujian Halaman Akses

Pengujian pada halaman akses user dan admin dilakukan untuk memastikan bahwa proses autentikasi berjalan dengan baik dan sesuai dengan fungsi yang diharapkan. Tabel 2 menunjukkan hasil pengujian halaman akses:

Tabel 2. Pengujian Halaman Akses

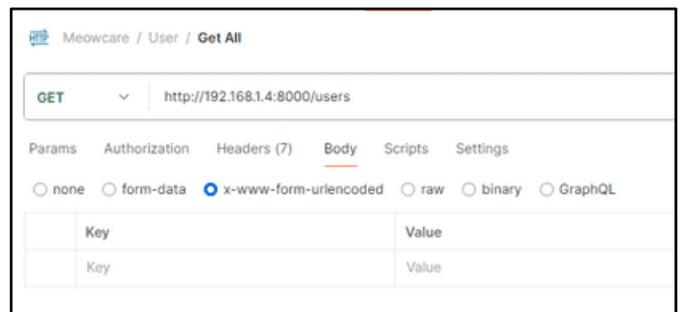
No	Kasus Uji	Hasil Yang Diharapkan	Hasil Pengujian	Status
1	Mengisi form <i>register</i> dengan mengisi nama lengkap, <i>username</i> , <i>email</i> dan <i>password</i> .	Data tersimpan ke database	Data tersimpan ke database	Valid
2	Melakukan <i>Login</i> ke sistem dengan email dan <i>password</i> yang benar	Tampil halaman <i>home</i> untuk <i>user</i> dan <i>dashboard</i> untuk admin	Tampil halaman <i>home</i> untuk <i>user</i> dan <i>dashboard</i> untuk admin	Valid
3	Mengisi form laporan dengan masukkan deskripsi dan foto	Data tersimpan ke database	Data tersimpan ke database	Valid
4	Mengisi form adopsi dengan memasukkan nama, umur, ras, jenis kelamin dan foto kucing	Data tersimpan ke database	Data tersimpan ke database	Valid
5	Menambahkan <i>like</i> dan komentar dalam artikel	Menampilkan komentar yang ditulis, jumlah <i>like</i> dan komentar bertambah	Menampilkan komentar yang ditulis, jumlah <i>like</i> dan komentar bertambah	Valid
6	Pilih halaman <i>Dashboard</i>	Tampil halaman dashboard	Tampil halaman dashboard	Valid
7	Pilih <i>List Report</i>	Tampil halaman <i>List Report</i> . Memilih opsi <i>accept</i> atau <i>reject</i> .	Tampil status report berhasil diperbarui sesuai opsi (<i>accept</i> atau	Valid

			<i>reject</i>) dan disimpan di database	
8	Pilih Artikel	Tampil halaman artikel. Membuat artikel dengan mengisi form judul, kategori, deskripsi, dan <i>thumbnail</i>	Berhasil menambahkan artikel dan disimpan di database	Valid
9	Pilih Tambah Kucing	Tampil halaman tambah kucing. Mengisi form nama kucing dan foto kucing	Berhasil menambahkan kucing dan disimpan di database	Valid

Berdasarkan tabel pengujian di atas, menunjukkan hasil bahwa keseluruhan proses yang diharapkan dalam halaman yang diakses oleh member dan admin berhasil dilakukan dengan sukses. Fitur-fitur utama untuk *user* seperti autentikasi, pengajuan adopsi, membaca artikel hingga membuat komentar berhasil diimplementasikan. Selain itu, fitur untuk admin seperti melihat laporan, membuat artikel, dan menambahkan kucing juga dapat berjalan dengan baik.

2. Pengujian API

Setelah tahapan implementasi, kita masuk ke tahapan testing. Disini kita akan melakukan pengujian untuk masing – masing endpoint API . Apakah API sudah berjalan sesuai yang diinginkan atau tidak. Perangkat lunak atau tools yang akan digunakan untuk pengujiannya yaitu Postman. Adapun cara menguji API dengan Postman adalah dengan mengatur *method* sesuai yang akan diuji dan memasukkan endpoint seperti pada Gbr 29.



Gbr. 29. Pengujian Endpoint API

Pengujian pada Gbr 33, dilakukan untuk mengambil seluruh data pengguna melalui endpoint */users*. Setelah *request* dikirimkan, server REST memberikan respon yang menunjukkan bahwa permintaan berhasil diproses, ditandai

dengan kode status 200. Hasil respon ditampilkan pada Gbr 30 dalam format JSON.

```

86  "Username": "joy",
87  "Email": "joy@gmail.com",
88  "Nomor_HP": null,
89  "Alamat": null,
90  "Role": "Adopter"
91  },
92  {
93    "User_ID": 12,
94    "Nama_Lengkap": "ningning",
95    "Foto_Profil": "images/profile/default.jpg",
96    "Username": "ningning",
97    "Email": "ningning@gmail.com",
98    "Nomor_HP": null,
99    "Alamat": null,
100   "Role": "Adopter"

```

Gbr. 30. Hasil Pengujian Endpoint API

Pengujian yang telah dilakukan menunjukkan bahwa API berfungsi dengan baik dan sesuai dengan kebutuhan sistem. Seluruh hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Daftar Pengujian API

Endpoint	Deskripsi Uji	Hasil
/users/id	Menampilkan data user berdasarkan id	Sukses
/users/register	Membuat data baru user dengan mengisikan nama lengkap, <i>username</i> , email, password, dan foto profil	Sukses
/users/login	Mengambil data user berdasarkan email dan <i>password</i>	Sukses
/users/logout	Mengakhiri sesi user	Sukses
/admins	Menampilkan semua data admin seperti admin id, nama lengkap, foto profil, <i>username</i> , email, nomor HP, dan alamat	Sukses
/admins/id	Menampilkan data user berdasarkan id	Sukses
/admins/register	Membuat data baru admin dengan mengisikan nama lengkap, <i>username</i> , email, dan password	Sukses
/admins/login	Mengambil data admin berdasarkan email dan <i>password</i>	Sukses
/admins/logout	Mengakhiri sesi admin	Sukses
/ras/id	Menampilkan data ras kucing	Sukses

	berdasarkan id	
/penyakit/id	Menampilkan data penyakit kucing berdasarkan id	Sukses
/warna/id	Menampilkan data warna kucing berdasarkan id	Sukses
/pets	Menampilkan semua data kucing	Sukses
/comments	Menampilkan semua komentar	Sukses
/artikel/id	Menampilkan artikel berdasarkan id	Sukses
/artikel/likes/id	Menampilkan artikel yang disukai berdasarkan id	Sukses
/pengajuan	Menampilkan semua data pengajuan	Sukses
/report	Menampilkan semua data laporan kucing	Sukses
/report/listreport	Menampilkan semua data laporan baik laporan kucing maupun adopsi	Sukses

Endpoint yang dibuat meliputi endpoint User dan Admin sebagai pengguna aplikasi dengan request dasar untuk login, mendaftar, dan sebagainya. Selanjutnya terdapat endpoint Pets dan Pengajuan, serta Konfirmasi yang digunakan untuk memanipulasi daftar peliharaan kucing yang tersedia, seperti menambahkan peliharaan baru, memulai pengajuan adopsi, hingga konfirmasi dari pihak admin. Sama seperti endpoint Pets sebelumnya, terdapat endpoint serupa yaitu Report, yang digunakan untuk memanipulasi data kucing laporan, untuk bisa ditindaklanjuti oleh Admin. Melengkapi endpoint Pets dan Report, terdapat endpoint tambahan yaitu Ras, Warna, dan Penyakit yang berfungsi untuk menyimpan dan memanipulasi data dari ras, warna, dan penyakit milik kucing untuk digunakan pada endpoint Pets dan Report. Selain itu, terdapat pula endpoint Artikel, untuk memanipulasi daftar Artikel yang bisa dilakukan oleh admin, serta beberapa fungsi seperti likes dan comments yang bisa dilakukan oleh user, dan endpoint Donation untuk membuat data donasi yang diterima.

V. KESIMPULAN

Berdasarkan hasil pengembangan dan pengujian, dapat disimpulkan bahwa API yang telah dikembangkan untuk sistem backend aplikasi Meowcare berhasil memenuhi spesifikasi dengan masing-masing endpoint menunjukkan hasil yang sukses. Aplikasi Meowcare sendiri juga sudah memenuhi kebutuhan dari komunitas KPKTS dalam mengelola proses penyelamatan dan adopsi kucing secara digital. Aplikasi ini menyediakan berbagai fitur seperti autentikasi pengguna,

pelaporan kucing terlantar, pengajuan adopsi, manajemen artikel edukatif, serta panel admin untuk memantau seluruh aktivitas. Pengujian menunjukkan seluruh fungsi sistem berjalan dengan baik dan memberikan pengalaman pengguna yang positif. Aplikasi Meowcare tidak hanya membantu proses internal komunitas, tetapi juga berpotensi meningkatkan partisipasi masyarakat luas dalam adopsi kucing dan kampanye kesejahteraan hewan. Pengembangan ke depan dapat difokuskan pada integrasi fitur pelacakan adopsi dan perluasan kerja sama antar komunitas penyelamat hewan.

REFERENSI

- [1] C. Evelina and N. Carina, "FASILITAS PENANGANAN HEWAN TERLANTAR," *Jurnal Sains, Teknologi, Urban, Perancangan, Arsitektur*, vol. 3, no. 2, pp. 2223–2223, Feb. 2022, doi: <https://doi.org/10.24912/stupa.v3i2.12378>.
- [2] A. D. Risnanda, "PERLINDUNGAN HUKUM DALAM KLASIFIKASI BENTUK KEKERASAN TERHADAP HEWAN DI INDONESIA," *Res Nullius Law Journal*, vol. 5, no. 2, pp. 123–134, Aug. 2023, doi: <https://doi.org/10.34010/rmlj.v5i2.9789>.
- [3] D. Ditinjau *et al.*, "MOTIF KOMUNITAS PENYELAMAT KUCING TERLANTAR SURABAYA (KPKTS) DALAM PENYELAMATAN KUCING JALANAN." Accessed: Jun. 17, 2025. [Online]. Available: http://digilib.uinsa.ac.id/58366/2/Septi%20Nurika%20Oktafiana_I03218018.pdf
- [4] A. Hanif, T. Dharmawan, and A. S. Pangestu, "Catstrate : Solusi Menekan Ledakan Populasi Kucing Lokal," *Anim. Welfare Sustain. Community*, no. October, pp. 0–3, 2017
- [5] D. Darine and A. Kadarisman, "Perancangan Media Informasi Dan Edukasi Hewan Terlantar Di Jakarta Design : Jakarta ' S Abandoned Animals Adoption Information and Education Media," *e-Proceeding Art Des.*, vol. 6, no. 2, pp. 1142–1147, 2020.
- [6] Jose Conary, Untung Subroto, and Meike Kurniawati, "Altruisme Pada Pembuat Shelter: Peran Dalam Penyelamatan Hewan Terlantar," *Didakt. J. Ilm. PGSD STKIP Subang*, vol. 10, no. 04 SE-Articles, pp. 252–271, 2024, [Online]. Available: <https://journal.stkipsubang.ac.id/index.php/didaktik/article/view/4686>
- [7] Y. Loui Pattinama, FERDIANSYAH, I. Susanti, and Painem, "Implementasi Rest API Web Service Dengan Otentifikasi JSON Web Token Untuk Aplikasi Properti," *Inform. J. Ilmu Komput.*, vol. 19, no. 1, pp. 81–89, 2023, doi: 10.52958/iftk.v19i1.5724.
- [8] "PENGANTAR APLIKASI MOBILE," *Google Books*, 2023. https://books.google.co.id/books?hl=id&lr=&id=2tLcEAAAQBAJ&oi=fnd&pg=PA35&dq=aplikasi+mobile&ots=6LnJi9gIKJ&sig=JvvuAwZXnJd-i4oX8ERmSghFJ40&redir_esc=y#v=onepage&q=aplikasi%20mobile&f=false (accessed Jun. 19, 2025).
- [9] W. G. Wardhana, I. Arwani, dan B. Rahayudi, "Implementasi Teknologi *Restful Web Service* Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis *Website* (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 2, pp. 680–689, Feb. 2020.
- [10] U. Senarath, "Waterfall Methodology, Prototyping and Agile Development," *Waterfall Methodology, Prototyping and Agile Development*, Jun. 2021, doi: <https://doi.org/10.13140/RG.2.2.17918.72001>.
- [11] S. Pargaonkar, "A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering," *International journal of scientific and research publications*, vol. 13, no. 8, pp. 120–124, Aug. 2023, doi: <https://doi.org/10.29322/ijsrp.13.08.2023.p14015>.
- [12] R. Ramakrishnan and J. Gehrke, *Database Management Systems*. 2017.
- [13] I. R. Mukhlis, I. F. Satibi, N. Sembilu, R. Rahmawati, V. R. Aulia, T. P. Rinjeni, T. L. I. Sugata, and P. K. F. Ananto, "Rancangan Basis Data Absensi Pegawai Menggunakan MySQL dengan Conceptual Data Model (CDM), Physical Data Model (PDM), dan Entity Relationship Diagram (ERD)," *Jurnal Ilmiah Computing Insight*, vol. 6, no. 2, 2024. doi: https://doi.org/10.30651/comp_insight.v6i2.24337.
- [14] Y. Sherley, Q. J. Ardian, and W. Kurnia, "Rancang Bangun Sistem Informasi Media Pembelajaran Berbasis Website (Studi Kasus: Bimbingan Belajar De Potlood)," *Jurnal Teknologi dan Sistem Informasi*, vol. 2, no. 3, pp. 136–147, Oct. 2021, doi: <https://doi.org/10.33365/jtsi.v2i3.879>.
- [15] V. Ayu, "Pemodelan Proses Pemilihan Rute pada Protokol Babel dengan Activity Diagram dan Transition System," vol. 12, no. 1, Jan. 2017, doi: <https://doi.org/10.24071/mt.v12i1.947>.
- [16] S. Şener and A. Türk, "Developing an Android-based Mobile Application for Temporary Animal Shelter Activities," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 9, pp. 381–389, Sep. 2019. <https://www.researchgate.net/publication/335867265>.
- [17] A. College, *Hybrid Mobile App Developemnt (React Native) Level 2*. Advanced Micro Systems Sdn Bhd.
- [18] M. A. Karim and A. R. Adriansyah, "Analisis dan Perancangan Aplikasi Mobile untuk Donasi menggunakan Metode Hybrid berbasis React Native," *Jurnal Informatika Terpadu*, vol. 8, no. 1, pp. 26–34, Mar. 2022, doi: <https://doi.org/10.54914/jit.v8i1.394>.
- [19] Herlambang, "Pengembangan Aplikasi Mobile I'm UII dengan Framework Flutter dan Lumen pada Fitur Reminder," pdf, Universitas Islam Indonesia, 2023. Accessed: Jun. 20, 2025. [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/45865>