



# **Software Requirements Spesification**

**Untuk**

## **Sistem Informasi Adopsi Hewan ‘Adopt A Buddy’ Berbasis Web Menggunakan Framework React dan Laravel**

**Version 1.0**

**Oleh :**

**Sofia Ismar Hermansyah (F 521 23 052)**

**Annisa Diandra Wahani (F 521 23 053)**

**Dede Al Fandi (F 521 23 058)**

**Dian Wulandari (F 521 23 060)**

**Aura Rahmadani (F 521 23 061)**

**Sistem Informasi  
Jurusan Teknologi Informasi  
Fakultas Teknik  
Universita Tadulako  
2025**

# Daftar Isi

<b>1. Pendahuluan.....</b>	<b>3</b>
1.1 Tujuan .....	3
1.2 Deskripsi Umum Dokumen .....	3
1.3 Saran Pembacaan Dokumen .....	3
1.4 Ruang Lingkup.....	4
1.5 Definisi dan Istilah.....	4
1.6 Referensi .....	6
<b>2. Deskripsi Kebutuhan Fungsional .....</b>	<b>6</b>
2.1 Fitur.....	6
2.2 Karakteristik Pengguna.....	7
2.3 Kebutuhan Perangkat Lunak dan Perangkat Keras.....	7
2.4 Dokumentasi Untuk Pengguna .....	8
<b>3. Desain dan Perancangan Sistem.....</b>	<b>8</b>
3.1 Arsitektur Sistem .....	8
3.2 Basis Data (Database).....	9
3.3 UML (Unified Modelling Language) .....	11
<b>4. Kebutuhan Antar Muka .....</b>	<b>17</b>
4.1 Antar Muka Pengguna .....	17
4.2 Antar Muka Perangkat Lunak.....	17
4.3 Antar Muka Komunikasi.....	18
<b>5. Persyaratan Non Fungsional.....</b>	<b>18</b>
5.1 Persyaratan Performasi .....	18
5.2 Persyaratan Keamanan.....	18
5.3 Hasil Lainnya .....	19

# 1. Pendahuluan

## 1.1 Tujuan

Dokumen ini merupakan Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau Software Requirements Specification (SRS) untuk Sistem Informasi Adopsi Hewan. Tujuan dari penulisan dokumen ini adalah untuk memberikan penjelasan secara menyeluruh mengenai perangkat lunak yang akan dikembangkan, baik berupa gambaran umum sistem maupun detail fungsional yang akan diimplementasikan.

Pengguna dari dokumen ini adalah pengembang (developer), penguji (tester), administrator sistem, serta pihak pengelola tempat penampungan hewan (animal shelter) yang akan menggunakan sistem ini. Dokumen ini akan dijadikan acuan dalam proses perancangan, pengembangan, dan evaluasi sistem agar pengembangannya lebih terarah, terukur, serta meminimalkan ambiguitas dalam implementasi

## 1.2 Deskripsi Umum Dokumen

Dokumen ini secara garis besar terdiri dari lima bab dengan perincian sebagai berikut :

- Bab 1 Pendahuluan. Memuat tujuan penulisan dokumen, deskripsi umum dokumen, saran pembacaan dokumen, ruang lingkup, dan referensi.
- Bab 2 Deskripsi Kebutuhan Fungsional. Memuat fitur, karakteristik pengguna, kebutuhan perangkat lunak dan perangkat keras, dan dokumentasi untuk pengguna.
- Bab 3 Desain dan Perancangan Sistem. Memuat basis data (*database*) berupa ERD (*Entity Relationship Diagram*) dan UML (*Unified Modelling Language*) berupa *Use Case*, *Class Diagram*, dan *Sequence Diagram*.
- Bab 4 Kebutuhan Antar Muka Eksternal. Memuat antar muka pengguna dan antar muka komunikasi.
- Bab 5 Persyaratan Non Fungsional. Memuat persyaratan performansi, persyaratan keamanan, dan hal lainnya.

## 1.3 Saran Pembacaan Dokumen

Dokumen ini dirancang agar dapat dipahami oleh berbagai pihak yang terlibat dalam pengembangan sistem. Beberapa saran pembacaan:

- Bab 2 direkomendasikan untuk dibaca oleh pengembang dan penguji karena berisi detail kebutuhan fungsional sistem.

- Bab 3 berguna untuk perancang dan pengembang sistem dalam memahami struktur dan alur kerja sistem.
- Bab 4 dapat dibaca oleh tim desain UI/UX dan pihak pengguna untuk memahami rancangan antarmuka.
- Bab 5 bermanfaat bagi administrator dan manajer proyek untuk memastikan sistem memenuhi kriteria performansi dan keamanan.

Dokumen ini penting bagi:

- Developer: untuk memastikan implementasi sistem sesuai kebutuhan.
- Tester: sebagai acuan dalam pengujian fungsi dan validasi sistem.
- *User* dan *Administrator Shelter*: untuk memahami fitur sistem dan cara penggunaannya.
- Penulis Dokumentasi: sebagai panduan dalam penyusunan manual pengguna dan panduan sistem.

#### 1.4 Ruang Lingkup

Perangkat lunak yang akan dikembangkan adalah Sistem Informasi Adopsi Hewan, yang digunakan oleh organisasi atau lembaga penampungan hewan untuk mengelola proses adopsi secara digital. Fungsi utamanya meliputi:

- Pencatatan data hewan yang tersedia untuk diadopsi.
- Pendaftaran calon pengadopsi dan proses verifikasi.
- Manajemen proses adopsi (pengajuan, persetujuan, dan pelacakan).
- Fitur komunikasi antara pihak *shelter* dan calon pengadopsi.

#### 1.5 Definisi dan Istilah

Istilah / Akronim	Deskripsi / Definisi
Sistem	Mengacu pada <i>Sistem Informasi Adopsi Hewan</i> yang dikembangkan untuk mengelola proses adopsi hewan secara digital, mulai dari pendaftaran, seleksi, hingga persetujuan adopsi.
Pengguna ( <i>User</i> )	Seseorang atau entitas yang berinteraksi langsung dengan sistem. Pengguna dibagi menjadi dua peran utama: Calon Pengadopsi dan Administrator/Staf Penampungan.
Calon Pengadopsi	Individu yang menggunakan sistem untuk mencari, memilih, dan mengajukan permohonan adopsi terhadap hewan tertentu.

Administrator / Staf Penampungan	Pengguna internal dari organisasi penampungan yang bertanggung jawab dalam pengelolaan data hewan, verifikasi calon pengadopsi, dan pengaturan status adopsi.
Hewan / Hewan Peliharaan	Makhluk hidup yang terdaftar di dalam sistem, seperti kucing, anjing, kelinci, dan hewan peliharaan lainnya yang siap untuk diadopsi.
Penampungan ( <i>Shelter / Rescue Group</i> )	Organisasi atau lembaga yang bertugas merawat dan menampung hewan sebelum diadopsi, serta menjadi pihak utama yang mengoperasikan sistem.
Profil Hewan	Halaman atau entri data dalam sistem yang berisi informasi lengkap tentang hewan, seperti nama, ras, usia, jenis kelamin, riwayat kesehatan, foto, dan status adopsi.
Aplikasi Adopsi ( <i>Adoption Application</i> )	Formulir digital yang diisi oleh calon pengadopsi sebagai bentuk permohonan untuk mengadopsi hewan tertentu.
Status Adopsi	Menunjukkan kondisi terkini dari proses adopsi suatu hewan. Status dapat berupa <i>Tersedia</i> , <i>Tertunda</i> , <i>Diadopsi</i> , atau <i>Perlu Tindak Lanjut Medis</i> .
Verifikasi Rumah ( <i>Home Check</i> )	Proses pemeriksaan terhadap lingkungan tempat tinggal calon pengadopsi untuk memastikan kelayakan dan keamanan bagi hewan yang akan diadopsi.
<i>Match / Kecocokan</i>	Algoritma atau kriteria dalam sistem yang digunakan untuk menilai kesesuaian antara karakteristik hewan dan preferensi calon pengadopsi.
NDA ( <i>Non-Disclosure Agreement</i> )	Perjanjian kerahasiaan yang mungkin diterapkan antara pihak pengembang dan organisasi penampungan untuk menjaga kerahasiaan data dan informasi sistem.
API ( <i>Application Programming Interface</i> )	Antarmuka pemrograman yang memungkinkan sistem berkomunikasi dengan aplikasi lain atau layanan pihak ketiga (misalnya sistem pembayaran atau media sosial).
DBMS ( <i>Database Management System</i> )	Sistem manajemen basis data yang digunakan untuk menyimpan, mengelola, dan memproses data hewan, pengguna, serta proses adopsi.
SRS ( <i>Software Requirements Specification</i> )	Dokumen resmi yang menjelaskan kebutuhan, fungsi, dan spesifikasi teknis dari perangkat lunak yang akan dikembangkan.

## 1.6 Referensi

Dokumen-dokumen yang dijadikan acuan dalam penyusunan *Software Requirements Specification (SRS)* ini antara lain:

- IEEE Std 830-1993, *IEEE Recommended Practice for Software Requirements Specifications*.
- Pressman, Roger S. (2001). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.

## 2. Deskripsi Kebutuhan Fungsional

### 2.1 Fitur

Sistem Informasi Adopsi Hewan dikembangkan untuk memfasilitasi proses adopsi hewan dengan lebih efisien, transparan, dan aman. Aplikasi ini menghubungkan calon adopter dengan lembaga penyelamat atau penampungan hewan (*animal shelter*), serta menyediakan fitur pengelolaan data hewan dan proses administrasi adopsi. Fitur utama yang disediakan oleh sistem meliputi:

#### 1. Manajemen Data Hewan

Admin dapat menambahkan, mengedit, dan menghapus data hewan seperti nama, jenis, usia, status kesehatan, serta keterangan adopsi.

#### 2. Pendaftaran dan Login Pengguna

Sistem mendukung autentikasi dua jenis pengguna, yaitu:

- Admin (Shelter): pengelola data hewan dan permohonan adopsi.
- Pengguna (Adopter): calon pengadopsi yang dapat melihat dan mengajukan adopsi hewan.

#### 3. Pencarian dan Filter Hewan

Pengguna dapat mencari hewan berdasarkan jenis, umur, jenis kelamin, lokasi, atau status adopsi.

#### 4. Formulir Pengajuan Adopsi

Pengguna yang tertarik dapat mengisi formulir adopsi yang berisi data pribadi, pengalaman merawat hewan, serta alasan ingin mengadopsi.

#### 5. Persetujuan dan Verifikasi Adopsi

Admin dapat meninjau, menerima, atau menolak permohonan adopsi berdasarkan kriteria tertentu.

## 6. Riwayat Adopsi dan Notifikasi

Pengguna dapat melihat riwayat adopsi mereka dan menerima notifikasi terkait status permohonan adopsi.

## 7. Sistem Donasi dan Dukungan Shelter

Pengguna dapat memberikan donasi secara online untuk membantu perawatan hewan di shelter.

## 8. Laporan dan Statistik Adopsi

Sistem menyediakan laporan otomatis untuk pihak shelter mengenai jumlah hewan yang diadopsi, belum diadopsi, dan jumlah donasi yang diterima.

## 2.2 Karakteristik Pengguna

Sistem Informasi Adopsi Hewan dirancang untuk dua kelompok pengguna utama dengan hak akses berbeda sebagaimana dijelaskan pada tabel berikut:

Jenis Pengguna	Deskripsi dan Hak Akses
Admin (Shelter)	Bertanggung jawab mengelola data hewan, memverifikasi permohonan adopsi, mengelola laporan, serta memperbarui status hewan.
Pengguna (User/Adopter)	Dapat melihat daftar hewan, mengisi formulir pengajuan adopsi, melacak status permohonan, memberikan donasi, dan melihat riwayat adopsi.

## 2.3 Kebutuhan Perangkat Lunak dan Perangkat Keras

Untuk antar muka pemakai, Sistem Informasi Adopsi Hewan (Adopt A Buddy) menggunakan antarmuka berbasis Web Graphic User Interface (GUI) yang dapat diakses melalui browser modern pada berbagai perangkat. Pengguna dapat mengoperasikannya menggunakan keyboard, mouse, atau layar sentuh (touchscreen). Sistem ini dapat dijalankan pada berbagai sistem operasi seperti Windows, GNU/Linux, dan macOS, selama terdapat dukungan browser web modern dan koneksi internet stabil.

Aplikasi “Adopt A Buddy” dapat diakses baik secara individual (standalone) maupun secara terhubung melalui jaringan (client-server), sehingga memungkinkan interaksi antara admin shelter dan calon adopter secara daring.

Untuk kebutuhan perangkat keras, aplikasi ini berjalan di **atas** komputer pribadi (PC), laptop, atau perangkat mobile dengan spesifikasi minimal sebagai berikut:

- Prosesor: Dual-core 1.8 GHz atau lebih tinggi
- Memori (RAM): 2 GB atau lebih

- Penyimpanan (Hard Disk/SSD): Minimal 20 GB ruang kosong
- Perangkat Jaringan: Koneksi internet stabil dengan kecepatan minimal 10 Mbps
- Perangkat Input: Keyboard, mouse, atau layar sentuh

Untuk antar muka perangkat lunak, aplikasi ini dikembangkan menggunakan bahasa pemrograman PHP dan JavaScript, dengan dukungan framework modern seperti Laravel (untuk sisi server) dan ReactJS (untuk sisi pengguna).

Basis data yang digunakan adalah MySQL, dihubungkan melalui ORM (Object Relational Mapping) dari framework Laravel. Server aplikasi dijalankan menggunakan Apache atau Nginx, dan dikembangkan dalam lingkungan sistem operasi Windows 10 atau Ubuntu 22.04 LTS.

## **2.4 Dokumentasi Untuk Pengguna**

Dokumentasi untuk pengguna akan disertakan bersama perangkat lunak PawCare, agar pengguna dapat dengan mudah memahami proses instalasi dan penggunaan aplikasi. Dokumentasi ini disediakan dalam bentuk file digital (PDF) dan halaman bantuan (help page) di dalam aplikasi.

# **3. Desain dan Perancangan Sistem**

## **3.1 Arsitektur Sistem**

Sistem informasi adopsi hewan ini dibangun dengan arsitektur web client–server, di mana pengguna dapat mengakses sistem melalui browser atau perangkat berbasis web, sedangkan seluruh proses pengolahan data dan logika bisnis dijalankan di server. Arsitektur ini dipilih agar sistem mudah diakses dari berbagai perangkat, responsif, dan dapat diintegrasikan dengan berbagai layanan eksternal.

### **1. Frontend: React.js**

Bagian frontend sistem menggunakan React.js, sebuah library JavaScript yang memungkinkan pembuatan antarmuka pengguna yang interaktif dan responsif. React.js memudahkan pembuatan komponen-komponen UI yang dapat digunakan kembali, seperti daftar hewan, form pengajuan adopsi, dan dashboard pengguna. Dengan React.js, tampilan sistem dapat menampilkan data secara dinamis dari backend tanpa perlu memuat ulang halaman, sehingga pengalaman pengguna menjadi lebih cepat dan nyaman.



## 2. Backend: Laravel (PHP)

Backend sistem dibangun menggunakan Laravel, framework PHP yang menyediakan arsitektur MVC (Model-View-Controller) sehingga memisahkan logika bisnis, data, dan tampilan dengan jelas. Laravel memudahkan pengelolaan request dari frontend, validasi data, autentikasi pengguna, serta implementasi API yang aman dan terstruktur. Semua proses seperti pengajuan adopsi, update status hewan, dan manajemen data kesehatan dilakukan di backend agar data tetap konsisten dan aman.

## 3. Database: MySQL

Untuk penyimpanan data, sistem menggunakan MySQL sebagai database relasional. MySQL digunakan untuk menyimpan informasi pengguna, data hewan, permohonan adopsi, catatan kesehatan, dan laporan adopsi. Struktur database dirancang sederhana namun lengkap dengan lima tabel utama agar memudahkan integritas data dan mempermudah query antar tabel. Penggunaan MySQL juga mendukung backup dan recovery data dengan mudah sehingga data penting tidak hilang.

## 4. Authentication & Storage: Firebase

Untuk autentikasi pengguna dan penyimpanan file seperti foto hewan, sistem memanfaatkan Firebase. Firebase Authentication memungkinkan pengguna mendaftar dan login dengan aman, baik menggunakan email/password maupun login sosial. Sementara Firebase Storage digunakan untuk menyimpan gambar hewan, sehingga foto dapat diakses secara cepat dari frontend tanpa membebani server utama. Integrasi ini membuat sistem lebih scalable dan mengurangi beban server backend.

### 3.2 Basis Data (Database)

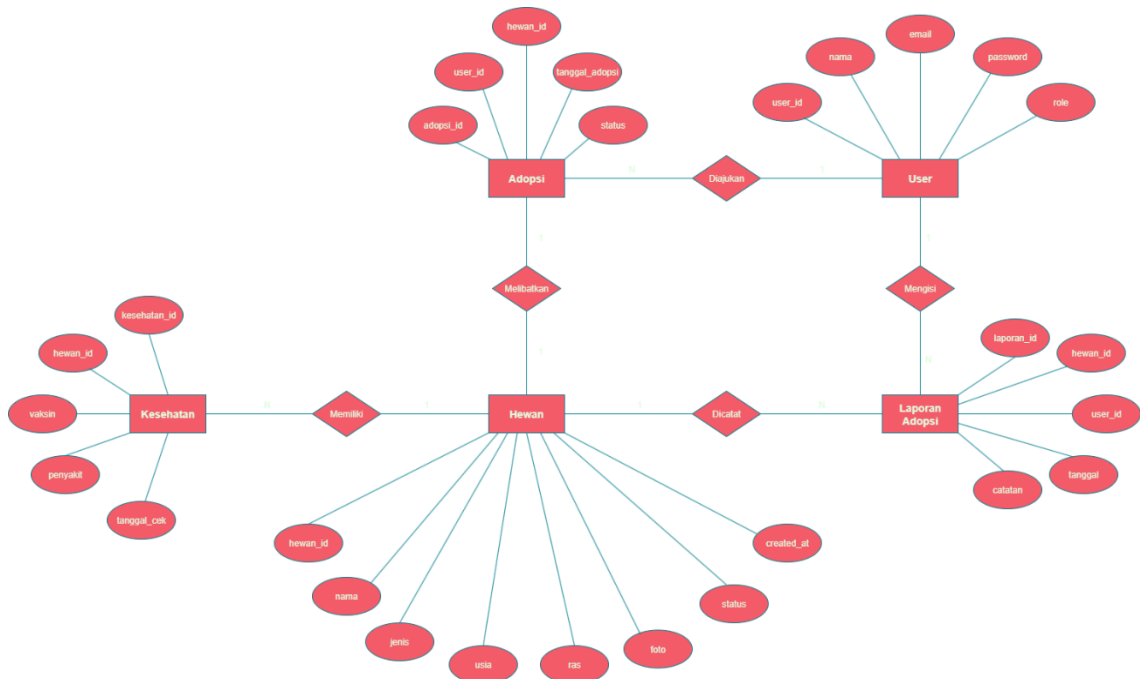
Data yang diolah pada aplikasi ini disimpan di Database PawCare, yang terdiri atas lima tabel utama: User, Hewan, Adopsi, Kesehatan, dan Laporan\_Adopsi.

- Pada tabel User terdapat field id, nama, email, password, dan role, yang menyimpan data pengguna termasuk admin dan calon adopter.
- Pada tabel Hewan terdapat field id, nama\_hewan, jenis, ras, usia, foto, status, created\_at, dan updated\_at, yang menyimpan informasi hewan yang tersedia untuk diadopsi, termasuk data kesehatan dasar dan status adopsi.
- Pada tabel Adopsi terdapat field id, user\_id, hewan\_id, tanggal\_adopsi, dan status, yang mencatat permohonan adopsi dari pengguna beserta status verifikasi dari admin.

- Pada tabel Kesehatan terdapat field id, hewan\_id, vaksin, penyakit, dan tanggal\_cek, yang menyimpan catatan kesehatan dan vaksinasi hewan.
- Pada tabel Laporan\_Adopsi terdapat field id, hewan\_id, adopter\_id, tanggal, dan catatan, yang menyimpan laporan adopsi serta catatan tambahan dari admin terkait proses adopsi yang telah selesai.

Database ini dirancang agar mudah diakses oleh backend Laravel dan mendukung integritas data antar tabel melalui relasi foreign key, sehingga setiap data adopsi, kesehatan, dan laporan selalu terkait dengan pengguna dan hewan yang relevan.

Untuk basis data ini dibuatkan ERD (Entity Relationship Diagram) sebagai berikut :



Keterangan ERD (Entity Relationship Diagram):

- Setiap pengguna (User) dapat mengajukan satu atau lebih permohonan adopsi (Adopsi), sedangkan setiap permohonan adopsi hanya diajukan oleh satu pengguna. Artinya, relasi antara User dan Adopsi bersifat satu ke banyak (1:N), dengan User sebagai pihak satu dan Adopsi sebagai pihak banyak.
- Setiap hewan (Hewan) hanya dapat diadopsi melalui satu permohonan adopsi pada suatu waktu, sedangkan setiap permohonan adopsi terkait dengan satu hewan. Relasi antara Hewan dan Adopsi bersifat satu ke satu (1:1), karena satu adopsi hanya berlaku untuk satu hewan.
- Setiap hewan (Hewan) dapat memiliki satu atau lebih catatan kesehatan (Kesehatan), sedangkan setiap catatan kesehatan hanya berhubungan dengan satu hewan. Dengan

demikian, relasi antara Hewan dan Kesehatan bersifat satu ke banyak (1:N), dengan Hewan sebagai pihak satu dan Kesehatan sebagai pihak banyak.

- Setiap hewan (Hewan) dapat muncul dalam satu atau lebih laporan adopsi (Laporan\_Adopsi), sedangkan setiap laporan adopsi hanya terkait dengan satu hewan. Relasi ini juga bersifat satu ke banyak (1:N), karena satu hewan dapat dicatat di beberapa laporan sesuai dengan aktivitas adopsi yang dilakukan.
- Setiap pengguna (User) dapat memiliki satu atau lebih laporan adopsi (Laporan\_Adopsi), sedangkan setiap laporan adopsi hanya dimiliki oleh satu pengguna. Sehingga, relasi User dan Laporan\_Adopsi juga bersifat satu ke banyak (1:N).

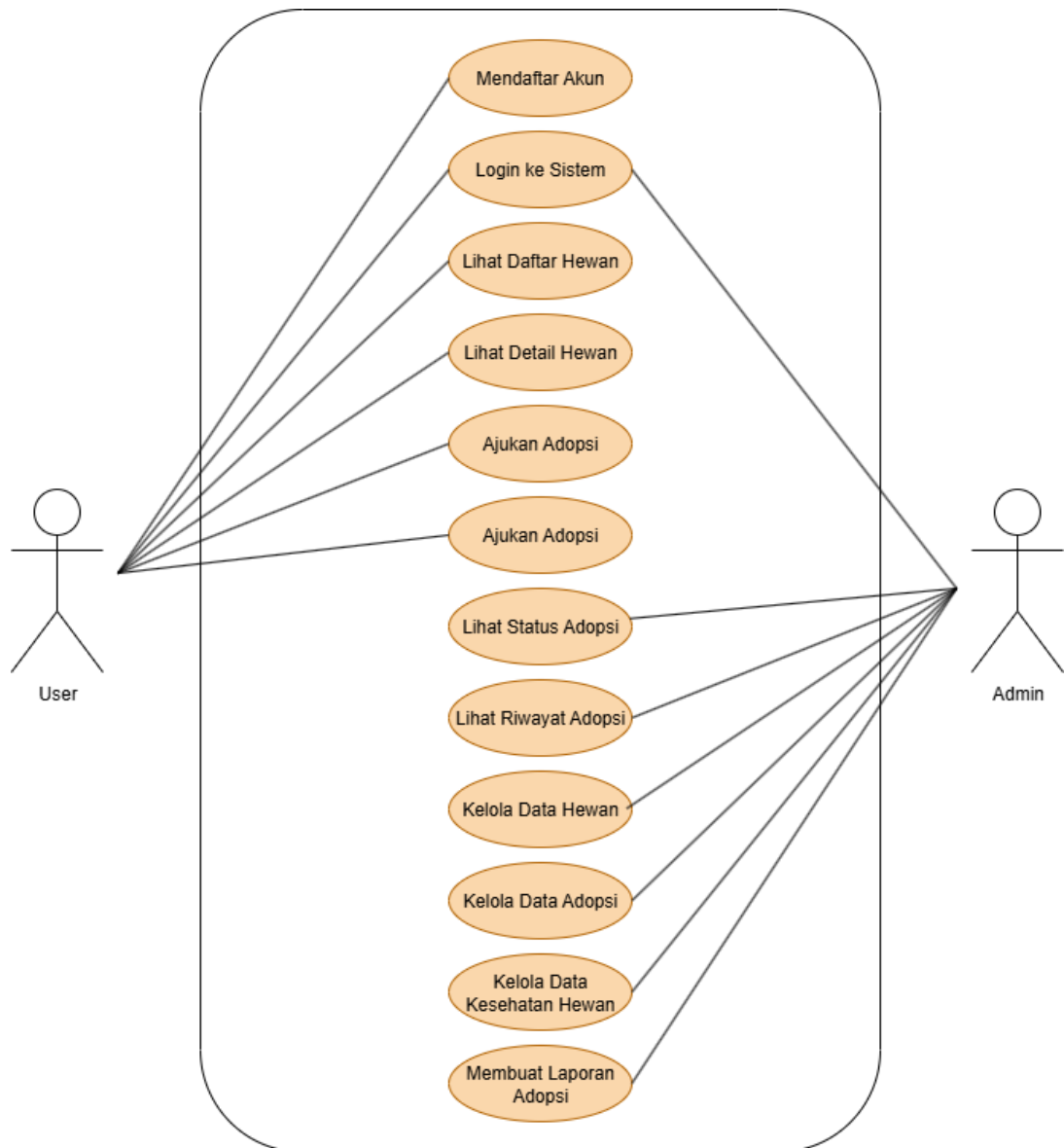
### **3.3 UML (Unified Modelling Language)**

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan untuk merancang dan memvisualisasikan sistem perangkat lunak secara terstruktur. UML membantu pengembang, analis, dan pemangku kepentingan lainnya memahami sistem sebelum proses implementasi dimulai, sehingga memudahkan komunikasi, perencanaan, dan dokumentasi. UML yang diberikan berupa Use Case Diagram dan Class Diagram beserta Flowchart Proses Sistem.

#### **1. Use Case Diagram**

Diagram ini menggambarkan interaksi antara aktor (pengguna atau admin) dengan sistem. Use Case Diagram membantu menjelaskan fungsional sistem dari perspektif pengguna, seperti mendaftar akun, melihat daftar hewan, mengajukan adopsi, memverifikasi adopsi, dan membuat laporan.

Berikut adalah gambar Use Case Diagram:



Keterangan Use Case Diagram:

- Use Case Mendaftar Akun: Memuat proses pendaftaran akun baru yang dilakukan oleh User untuk membuat akun di sistem. Data disimpan ke database pada tabel User dengan informasi nama, email, password, dan role.
- Use Case Login ke Sistem: Memuat proses autentikasi yang dilakukan oleh User dan Admin untuk masuk ke sistem. Sistem memverifikasi kredensial dengan data yang tersimpan di tabel User.
- Use Case Lihat Daftar Hewan: Memuat proses melihat daftar hewan yang tersedia untuk adopsi, dilakukan oleh User. Data ditampilkan dari database pada tabel Hewan yang menampilkan informasi semua hewan beserta statusnya.

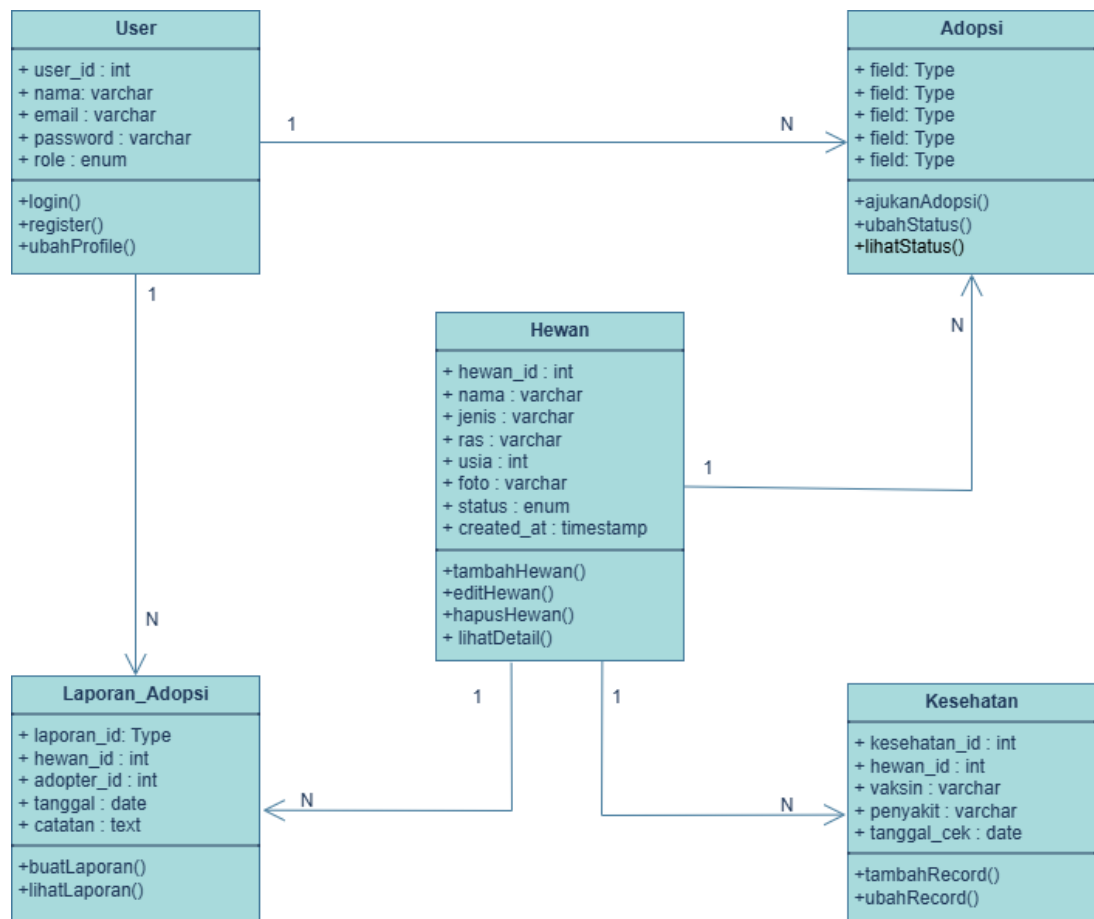
- Use Case Lihat Detail Hewan: Memuat proses melihat informasi lengkap hewan tertentu yang dilakukan oleh User. Data diambil dari database pada tabel Hewan dan tabel Kesehatan untuk menampilkan detail hewan termasuk riwayat kesehatannya.
- Use Case Ajukan Adopsi: Memuat proses pengajuan adopsi hewan yang dilakukan oleh User ke dalam sistem. Data disimpan ke database pada tabel Adopsi dengan status awal "Pending" menunggu verifikasi Admin.
- Use Case Lihat Status Adopsi: Memuat proses melihat perkembangan permohonan adopsi yang dilakukan oleh User dan Admin. Data diambil dari database pada tabel Adopsi untuk menampilkan status (Pending/Diterima/Ditolak) dari setiap pengajuan adopsi.
- Use Case Lihat Riwayat Adopsi: Memuat proses melihat histori adopsi yang telah selesai, dilakukan oleh User dan Admin. Data ditampilkan dari database pada tabel Laporan\_Adopsi yang berisi catatan adopsi yang sudah disetujui.
- Use Case Kelola Data Hewan: Memuat proses manajemen data hewan (tambah, edit, hapus) yang dilakukan oleh Admin ke dalam sistem. Data diproses pada database tabel Hewan untuk mengelola informasi hewan yang tersedia.
- Use Case Kelola Data Adopsi: Memuat proses verifikasi dan perubahan status adopsi yang dilakukan oleh Admin. Data diambil dan diubah pada database tabel Adopsi untuk menyetujui atau menolak permohonan adopsi dari User.
- Use Case Kelola Data Kesehatan Hewan: Memuat proses manajemen catatan kesehatan hewan (tambah, edit riwayat vaksin dan penyakit) yang dilakukan oleh Admin. Data disimpan ke database pada tabel Kesehatan untuk mencatat kondisi kesehatan setiap hewan.
- Use Case Membuat Laporan Adopsi: Memuat proses pembuatan laporan adopsi yang berhasil diselesaikan, dilakukan oleh Admin. Data disimpan ke database pada tabel Laporan\_Adopsi setelah permohonan adopsi disetujui dan proses adopsi selesai.

## 2. Class Diagram

Class Diagram digunakan untuk menggambarkan struktur statis sistem, termasuk kelas, atribut, metode, dan relasi antar kelas. Diagram ini memudahkan tim pengembang untuk memahami bagaimana data dan logika bisnis diorganisasikan,

misalnya hubungan antara pengguna, hewan, permohonan adopsi, catatan kesehatan, dan laporan adopsi.

Berikut adalah gambar Class Diagram:



Keterangan Class Diagram:

- Class User

Atribut: Menyimpan data pengguna meliputi user\_id sebagai identifier unik, nama, email, password untuk autentikasi, dan role untuk membedakan tipe pengguna (admin/user biasa). Method: Memiliki fungsi login() untuk autentikasi, register() untuk pendaftaran akun baru, dan ubahProfile() untuk mengubah data profil. Relasi: Berelasi one-to-many dengan Adopsi (1 user dapat mengajukan banyak adopsi) dan one-to-many dengan Laporan\_Adopsi (1 user dapat memiliki banyak laporan adopsi).

- Class Hewan

Atribut: Menyimpan informasi hewan yang tersedia untuk adopsi meliputi hewan\_id sebagai identifier, nama hewan, jenis (kucing/anjing/dll), ras, usia, foto untuk tampilan visual, status ketersediaan (tersedia/diadopsi), dan created\_at untuk mencatat waktu pendaftaran hewan. Method: Dilengkapi

fungsi tambahHewan() untuk menambah data hewan baru, editHewan() untuk mengubah informasi hewan, hapusHewan() untuk menghapus data, dan lihatDetail() untuk menampilkan informasi lengkap. Relasi: Berelasi one-to-many dengan Adopsi (1 hewan bisa diajukan adopsi berkali-kali), one-to-many dengan Laporan\_Adopsi (1 hewan bisa muncul di banyak laporan), dan one-to-many dengan Kesehatan (1 hewan memiliki banyak catatan kesehatan).

- Class Adopsi

Atribut: Berisi field-field untuk mencatat proses pengajuan adopsi (detail field bertipe Type menunjukkan ada beberapa atribut yang menyimpan data permohonan). Method: Memiliki ajukanAdopsi() untuk membuat pengajuan baru, ubahStatus() untuk admin mengubah status permohonan, dan lihatStatus() untuk user melihat perkembangan permohonan. Relasi: Berelasi many-to-one dengan User (banyak adopsi dari 1 user) dan many-to-one dengan Hewan (banyak pengajuan untuk 1 hewan yang sama).

- Class Laporan\_Adopsi

Atribut: Menyimpan laporan adopsi yang sudah selesai dengan laporan\_id, hewan\_id dan adopter\_id sebagai foreign key, tanggal adopsi, dan catatan tambahan proses adopsi. Method: Dilengkapi buatLaporan() untuk generate laporan baru setelah adopsi berhasil dan lihatLaporan() untuk menampilkan riwayat adopsi. Relasi: Berelasi many-to-one dengan User (banyak laporan dari 1 user) dan many-to-one dengan Hewan (banyak laporan untuk 1 hewan yang pernah diadopsi berkali-kali).

- Class Kesehatan

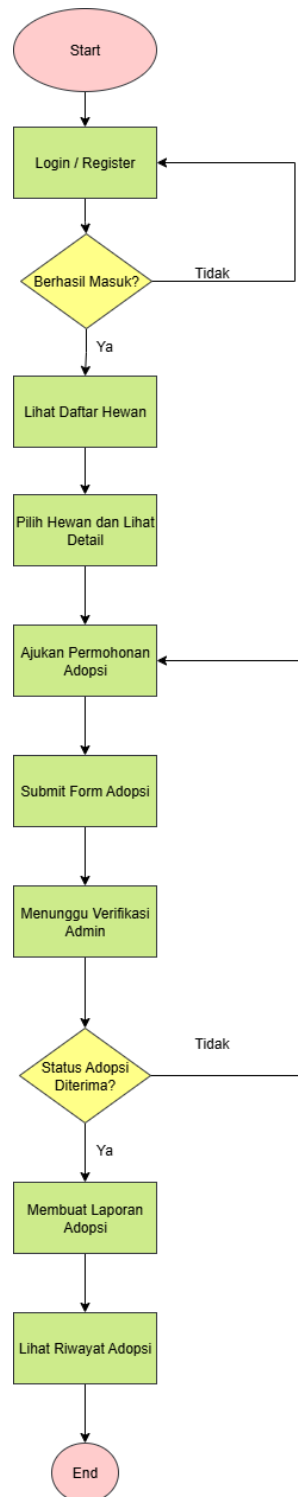
Atribut: Mencatat riwayat kesehatan hewan meliputi kesehatan\_id sebagai identifier, hewan\_id sebagai foreign key, vaksin yang sudah diberikan, penyakit yang pernah diderita, dan tanggal\_cek untuk tracking pemeriksaan. Method: Memiliki tambahRecord() untuk menambah catatan kesehatan baru dan ubahRecord() untuk update informasi kesehatan. Relasi: Berelasi many-to-one dengan Hewan (banyak catatan kesehatan untuk 1 hewan).

### 3. Flowchart Proses Sistem

Flowchart digunakan untuk menjelaskan alur proses sistem secara visual, mulai dari input pengguna hingga output sistem. Flowchart membantu memahami langkah-

langkah proses bisnis, misalnya alur pengajuan adopsi hewan, verifikasi oleh admin, update status, dan pembuatan laporan.

Beriku adalah gambar Flowchart:



Sistem dimulai ketika pengguna membuka browser dan mengakses aplikasi. Pengguna harus login atau register terlebih dahulu—jika gagal, muncul pesan error dan kembali ke halaman login. Setelah login berhasil, pengguna masuk ke dashboard dan dapat melihat daftar



hewan yang tersedia untuk adopsi. Pengguna memilih hewan yang diminati untuk melihat detail lengkapnya, kemudian mengisi form adopsi online dan mengirimkan permohonan. Data adopsi tersimpan di database dengan status "Pending" menunggu verifikasi admin. Admin memverifikasi permohonan dan mengubah statusnya menjadi "Diterima" atau "Ditolak". Jika diterima, pengguna mendapat notifikasi, proses adopsi selesai, dan data tercatat di laporan adopsi; jika ditolak, pengguna mendapat notifikasi penolakan dan bisa mengajukan adopsi hewan lain. Pengguna dapat melihat riwayat adopsi mereka kapan saja untuk mengecek hewan yang sudah berhasil diadopsi. Sistem selesai ketika pengguna keluar dari aplikasi.

## **4. Kebutuhan Antar Muka**

### **4.1 Antar Muka Pengguna**

Antar muka pengguna pada sistem informasi adopsi hewan ini dirancang berbasis web responsif, sehingga dapat diakses melalui perangkat komputer, laptop, tablet, maupun smartphone. Desain antarmuka menekankan kemudahan navigasi, konsistensi elemen visual, dan pengalaman pengguna yang intuitif (user-friendly). Pengguna dapat dengan mudah melihat daftar hewan yang tersedia, mengakses detail hewan, mengajukan adopsi, memantau status permohonan, serta melihat riwayat adopsi. Bagi admin, antar muka menampilkan dashboard yang memudahkan pengelolaan data hewan, verifikasi adopsi, pencatatan kesehatan, dan pembuatan laporan adopsi.

### **4.2 Antar Muka Perangkat Lunak**

Antar muka perangkat lunak menjelaskan integrasi antara sistem utama dengan modul atau layanan tambahan yang mendukung operasionalnya. Sistem ini mengintegrasikan backend Laravel dengan MySQL sebagai database utama, serta Firebase Authentication dan Storage untuk autentikasi dan penyimpanan file media. Integrasi ini memastikan data pengguna, hewan, adopsi, dan catatan kesehatan tersimpan dengan aman, serta modul-modul sistem dapat saling berinteraksi secara efisien. Desain antar muka perangkat lunak juga memungkinkan skalabilitas dan fleksibilitas, sehingga di masa depan sistem dapat dikembangkan dengan menambahkan fitur baru seperti notifikasi, chat adopter-admin, atau integrasi dengan layanan pihak ketiga lainnya.

### **4.3 Antar Muka Komunikasi**

Antar muka komunikasi menjelaskan bagaimana sistem berinteraksi dengan komponen eksternal. Sistem ini menggunakan arsitektur client-server, di mana frontend React.js berkomunikasi dengan backend Laravel melalui API RESTful. Seluruh data, termasuk informasi hewan, permohonan adopsi, dan statusnya, dikirim dari backend ke frontend dalam format JSON. Selain itu, sistem juga terintegrasi dengan Firebase untuk autentikasi pengguna dan penyimpanan file gambar hewan, sehingga proses login, registrasi, dan upload foto dapat berjalan dengan cepat dan aman.

## **5. Persyaratan Non Fungsional**

### **5.1 Persyaratan Performasi**

Agar dapat digunakan dengan baik, perangkat lunak Sistem Informasi Adopsi Hewan (Adopt A Buddy) harus dijalankan pada komputer dengan spesifikasi minimum sebagaimana dijelaskan pada sub bab 2.3 (Kebutuhan Perangkat Lunak dan Perangkat Keras). Selain itu, sistem harus memenuhi kriteria performansi berikut:

1. Waktu Respon Cepat

Setiap proses seperti login, pencarian data hewan, dan pengajuan adopsi harus memiliki waktu respon maksimal 3 detik.

2. Kapasitas Data

Sistem mampu menampung sedikitnya 500 data hewan, 1.000 data pengguna, dan 1.000 riwayat adopsi tanpa menurunkan performa.

3. Koneksi Stabil

Sistem dapat beroperasi secara optimal pada koneksi jaringan minimal 10 Mbps dan mendukung penggunaan bersamaan (*multi-user*) hingga 50 pengguna secara simultan.

4. Ketersediaan Sistem (Availability)

Server sistem harus memiliki tingkat ketersediaan minimal 99% uptime untuk memastikan layanan adopsi daring tetap dapat diakses pengguna kapan pun.

### **5.2 Persyaratan Keamanan**

Aplikasi “Adopt A Buddy” dirancang dengan dua jenis tingkatan pengguna, yaitu Admin Shelter dan Pengguna (Adopter), masing-masing dengan hak akses berbeda. Beberapa hal yang perlu diperhatikan dalam penerapan keamanan sistem adalah:

#### 1. Pengaturan Hak Akses (Role Management)

Hanya pengguna dengan peran Admin Shelter yang dapat mengubah data hewan, mengelola laporan, dan menyetujui pengajuan adopsi. Pengguna Adopter hanya dapat melakukan registrasi, melihat data hewan, serta mengajukan permohonan adopsi.

#### 2. Autentikasi dan Enkripsi Data

- Setiap pengguna wajib melakukan login menggunakan akun terverifikasi.
- Password disimpan dalam bentuk terenkripsi menggunakan algoritma aman seperti bcrypt.
- Komunikasi data antara client dan server diamankan dengan protokol HTTPS (SSL/TLS).

#### 3. Keamanan Basis Data

- Database MySQL dilindungi dengan sistem autentikasi pengguna dan hak akses terbatas.
- Proses backup database dilakukan secara berkala untuk menjaga keamanan dan keutuhan data.
- Proses backup dapat dilakukan melalui *interface* seperti PHPMyAdmin atau mekanisme otomatis melalui server.

#### 4. Keamanan Server dan Jaringan

- Server harus dilengkapi dengan firewall dan anti-malware protection.
- Akses administratif dibatasi hanya untuk pihak yang berwenang.

### 5.3 Hasil Lainnya

- Aplikasi Sistem Informasi Adopsi Hewan (Adopt A Buddy) secara umum tidak memerlukan perawatan khusus baik pada sisi *hardware* maupun *software*. Perawatan data dilakukan oleh Administrator Shelter selaku pengelola server basis data atau pegawai yang dipercayakan untuk bagian tersebut.
- Aplikasi “Adopt A Buddy” secara khusus dirancang untuk mengelola proses adopsi hewan peliharaan, mulai dari pendataan hewan, pengajuan adopsi, hingga konfirmasi adopsi oleh pihak shelter. Meski demikian, perangkat lunak ini dapat dikembangkan dan dimodifikasi untuk keperluan lain seperti pendataan hewan hilang, pelaporan kesejahteraan hewan, atau sistem donasi online.

- Aplikasi “Adopt A Buddy” dapat dijalankan di berbagai platform sistem operasi yang mendukung web browser modern serta dilengkapi dengan database MySQL, server Apache dan koneksi jaringan internet (optional untuk mode offline lokal).