

# **SRS — Sistem Informasi Rumah Sakit Hewan (Rekam Medis Digital)**

## **1. Ringkasan Eksekutif**

Sistem ini adalah aplikasi web untuk mengelola layanan klinik/rumah sakit hewan: pendaftaran pemilik, profil hewan, rekam medis lengkap, booking dokter, pembuatan tagihan, pembayaran online, jadwal vaksinasi, dan manajemen resep. Tujuan utama: digitalisasi rekam medis, meningkatkan efisiensi operasional klinik, serta memberikan akses mudah bagi pemilik hewan.

## **2. Tujuan dan Ruang Lingkup**

**Tujuan:** - Menyediakan rekam medis digital untuk tiap hewan peliharaan. - Menyederhanakan proses booking dan pembayaran. - Menyimpan histori perawatan, diagnosis, resep, dan vaksinasi.

**Ruang lingkup:** - Modul pemilik (Owners) - Modul hewan (Pets) - Modul dokter (Vets) - Rekam medis, diagnosis, resep - Jadwal vaksinasi & reminder - Booking & jadwal janji temu - Tagihan & pembayaran - Dashboard admin & laporan

## **3. Definisi, Singkatan, dan Akronim**

- User: Pemilik hewan.
- Pet: Hewan peliharaan.
- Vet: Dokter hewan.
- EMR: Electronic Medical Record.
- Admin: Pengelola sistem klinik.

#### 4. Pemangku Kepentingan

- Pemilik hewan: mendaftar, booking, melihat riwayat, bayar.
- Dokter hewan: melihat rekam, input diagnosis, resep. Dokter ada status (aktif atau tidak aktif)
- Receptionist/Admin klinik: mengelola jadwal, verifikasi pembayaran, laporan.
- Sistem Pembayaran: gateway pembayaran pihak ketiga.

#### 5. Gambaran Umum Sistem

Aplikasi berbasis web (responsive) + API untuk mobile app di masa depan. Terdiri dari beberapa modul: otentikasi, manajemen pengguna, manajemen hewan, rekam medis, jadwal/vaksin, billing & payment, notifikasi (email/SMS/push).

#### 6. Kebutuhan Fungsional (Functional Requirements)

Fitur	User (Pemilik Hewan)	Admin	Guest	Dokter
Login	✓	✓	✗	✓
Register / Buat Akun	✓	✗	✓	✓
Update Password	✓	✓	✗	✓
Melihat Profil	✓	✓	✗	✓

Update Profil	✓	✓	✗	✓
---------------	---	---	---	---

Fitur	User	Admin	Guest	Dokter
Menambah Hewan Peliharaan (Add Pet)	✓	✗	✗	✗
Mengedit Data Hewan	✓	✗	✗	✗
Menghapus Hewan	✓	✗	✗	✗
Melihat Daftar Hewan	✓	✓	✗	✓
Melihat Detail Hewan	✓	✓	✗	✓

Fitur	User	Admin	Guest	Dokter
Melihat Daftar Dokter	✓	✓	✓	✓
Menambah Dokter	✗	✓	✗	✗
Mengedit Dokter	✗	✓	✗	✗

Mengubah Status Dokter (Active/Inactive)	✗	✓	✗	
---	---	---	---	--

Fitur	User	Admin	Guest	Dokter
Membuat Appointment	✓	✗	✗	✗
Membatalkan Appointment	✓	✓	✗	✗
Melihat Appointment	✓	✓	✗	✓
Dokter melihat jadwal pasien	✗	✓	✗	✓

Fitur	User	Admin	Guest	Dokter
Melihat Riwayat Medis Hewan	✓	✓	✗	✓
Menambah Rekam Medis	✗	✓	✗	✓
Dokter Menambah Rekam Medis	✗	✓	✗	✓

Menambah Diagnosis	✗	✓	✗	✓
Menambah Obat / Medications	✗	✓	✗	✓

Fitur	User	Admin	Guest	Dokter
Melihat Jadwal Vaksin Hewan	✓	✓	✗	✓
Notifikasi Jadwal Vaksin	✓	✓	✗	✓
Menambah Jadwal Vaksin	✗	✓	✗	✓

Fitur	User	Admin	Guest	Dokter
Melihat Tagihan	✓	✓	✗	✗
Membayar Tagihan	✓	✗	✗	✗
Input Pembayaran Manual (misal bayar cash)	✗	✓	✗	✗
Melihat Riwayat Pembayaran	✓	✓	✗	✗

Fitur	User	Admin	Guest	Dokter
Notifikasi Appointment	✓	✓	✗	✓
Notifikasi Tagihan	✓	✓	✗	✓
Notifikasi Vaksin	✓	✓	✗	✓

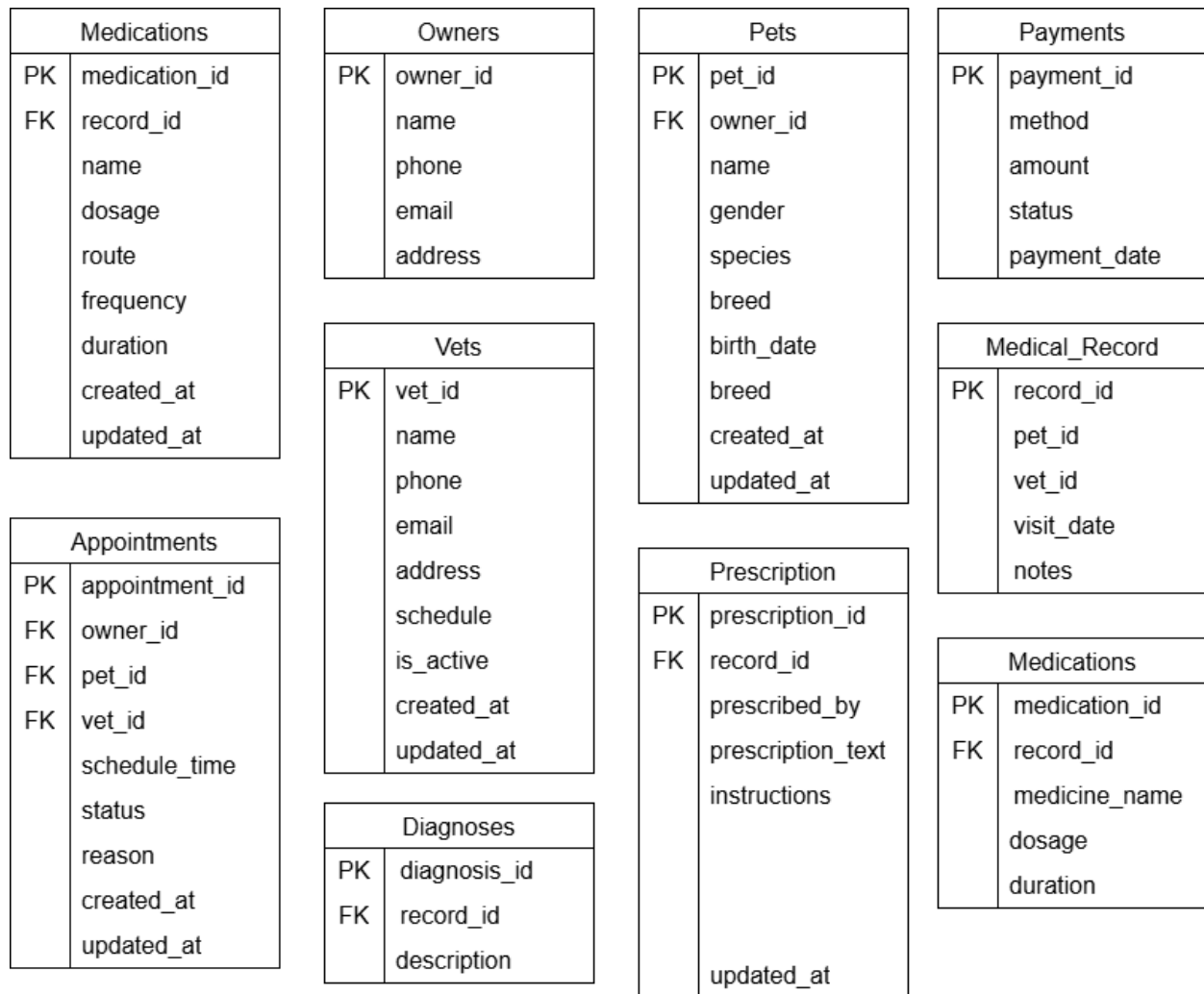
Fitur	User	Admin	Guest	Dokter
Melihat Semua Data Owner	✗	✓	✗	✗
Melihat Semua Rekam Medis	✗	✓	✗	✗
Melihat Semua Appointment	✗	✓	✗	✓
Export Data (PDF/Excel)	✓	✓	✗	✓

## 7. Alur Data (Data Flow Overview)

- User -> (web) -> API -> DB (Owners, Pets, Appointments).
- Dokter -> (web) -> API -> DB (Medical\_Records, Diagnoses, Medications).
- Payment Gateway <-> API -> DB (Bills, Payments).
- Scheduler (cron) -> mengecek Vaccination\_Schedule -> kirim notifikasi.

## 8. Desain Basis Data (ERD & Skema)

Berikut ERD dalam draw.io:



## 9. Desain API (Endpoint Contoh)

- POST /api/auth/register - register owner
- POST /api/auth/login - login
- GET /api/pets - daftar hewan owner
- POST /api/pets - tambah hewan
- GET /api/vets - daftar dokter
- POST /api/appointments - buat appointment
- GET /api/medical-records?pet\_id= - riwayat medis

- POST /api/medical-records - buat rekam medis (oleh vet)
- GET /api/bills?owner\_id= - tagihan
- POST /api/payments - input pembayaran

Autentikasi: JWT Bearer token.

## 12. Keamanan & Privasi

- Gunakan HTTPS (TLS) untuk semua komunikasi.
- Simpan password hashed (bcrypt/argon2) + salt.
- Enkripsi sensitive fields bila perlu (mis. nomor kartu jika disimpan — tapi jangan menyimpan CVV).
- Audit log untuk perubahan rekam medis.
- Hak akses berbasis role: Owner, Vet, Admin.

## 13. Non-Functional Ops

- Backup: snapshot DB harian + backup incremental.
- Logging: simpan audit log, akses log, error log.
- Monitoring: gunakan tool monitoring (Prometheus/Grafana) untuk health checks.

## 14. Acceptance Criteria (Kelayakan)

1. Owner dapat mendaftar, menambah 1 hewan, dan membuat appointment.
2. Vet dapat membuat medical record yang tersimpan dan muncul di riwayat Owner.
3. Sistem membuat bill otomatis untuk medical record sederhana.
4. Owner dapat melunasi bill melalui payment mock (sandbox) dan status terupdate.
5. Reminder vaksin dikirimkan minimal via email (sistem cron).



## 15. Roadmap Pengembangan (Iterasi)

**MVP (1–2 bulan):** Auth, Owners, Pets, Vets, Appointments, Medical\_Records, Bills, Payments (sandbox), Vaccination\_Schedule, basic UI.

**Phase 2:** Notifikasi SMS/Push, real payment gateway, image upload, search & filter, reporting.

**Phase 3:** Mobile app, AI support (diagnosis suggestion), teleconsultation (chat/video), integrasi perangkat IoT (microchip reader).

## 16. Lampiran: Use Case Detail (contoh UC4 — Pemeriksaan)

**Actor:** Vet **Precondition:** Vet login dan memiliki appointment yang dikonfirmasi. **Main Flow:**

1. Vet membuka appointment. 2. Vet mengisi chief\_complaint, pemeriksaan fisik, hasil lab (jika ada). 3. Vet memasukkan diagnosis dan resep. 4. Sistem menyimpan Medical\_Record dan men-generate Bill. 5. Owner menerima notifikasi tagihan.