

SRS – Sistem booking Tiket Kebun Binatang

projek:sistem booking tiket kebun binatang

1. Pendahuluan

1.1 Tujuan

Dokumen ini menjelaskan spesifikasi kebutuhan sistem pemesanan tiket kebun binatang untuk keperluan ujian/proyek sekolah. Sistem memfasilitasi pendaftaran user, login, pencarian hewan, pemesanan tiket, dan administrasi oleh admin.

1.2 Ruang lingkup

Aplikasi web sederhana dengan 2 peran utama:

- **User:** register, login, mencari hewan, booking tiket, melihat status booking.
- **Admin:** login, CRUD data hewan & tiket, melihat daftar booking, ACC/tolak booking.

1.3 Terminologi

- CRUD = Create, Read, Update, Delete
- ACC = Approve (setujui)
- Pending = Status awal booking sebelum disetujui

2. Kebutuhan Fungsional (Functional Requirements)

Modul User

- **UF-01 Register:** User daftar dengan **nama**, **username**, **password**. Validasi username unik.
- **UF-02 Login:** Otentikasi username & password.

- **UF-03 Cari Informasi Hewan:** Pencarian/penjelajahan daftar hewan (gambar + deskripsi).
- **UF-04 Booking Tiket:** Pilih jenis tiket, tanggal, jumlah → simpan booking dengan status **pending**.
- **UF-05 Lihat Status Booking:** User melihat daftar booking dan statusnya (pending/acc/ditolak).

Modul Admin

- **AF-01 Login Admin:** Otentikasi admin.
- **AF-02 CRUD Data Hewan:** Tambah/edit/hapus/list hewan.
- **AF-03 CRUD Data Tiket:** mengelola tiket
- **AF-04 Lihat Booking:** Menampilkan semua booking (filter pending/done).
- **AF-05 ACC/Tolak Booking:** Ubah status booking menjadi **acc** atau **ditolak**.

3. Kebutuhan Non-Fungsional

- **Keamanan:** Password harus disimpan hashed (minimal hashing). Input harus divalidasi untuk mencegah SQL Injection.
- **Kinerja:** Responsif untuk penggunaan lokal (XAMPP) dan dapat menangani puluhan request simultan pada lingkungan pengujian.
- **Usability:** UI sederhana, mudah dipahami (untuk tujuan ujian).
- **Platform:** Aplikasi berjalan pada browser modern (Chrome/Firefox) dan server lokal XAMPP.

4. Desain Basis Data (berdasarkan **kebun_binatang.sql** — ringkasan)

users

- **id_user** (INT PK)

- nama (VARCHAR)
- username (VARCHAR UNIQUE)
- password (VARCHAR) — simpan hashed
- role (ENUM: 'user','admin')

hewan

- id_hewan (INT PK)
- nama_hewan (VARCHAR)
- deskripsi (TEXT)
- gambar (VARCHAR / path)

tiket

- id_tiket (INT PK)
- jenis_tiket (VARCHAR)
- harga (DECIMAL)
- kuota (INT) (*opsional, jika proyek mendukung kuota*)

booking

- id_booking (INT PK)
 - id_user (FK → users)
 - id_tiket (FK → tiket)
 - tanggal_kunjungan (DATE)
 - jumlah (INT)
 - status (ENUM: 'pending','acc','ditolak')
 - created_at (DATETIME) (*opsional*)
-

5. Alur Sistem (singkat)

1. User registrasi → login
 2. User mencari hewan / melihat daftar
 3. User memilih tiket → mengisi form booking → status **pending** disimpan
 4. Admin login → melihat daftar booking → Accepted / tolak booking
 5. User mengecek status booking (manual notifikasi belum otomatis)
-

6. Kriteria Penerimaan

- User dapat register & login dengan valid.
 - Booking tersimpan di tabel **booking** dengan status **pending**.
 - Admin dapat melihat booking dan mengubah status menjadi **acc** atau **ditolak**.
 - Admin dapat melakukan CRUD data hewan & tiket.
 - Data sensitif (password) tidak ditampilkan mentah.
-

7. Kekurangan & Rekomendasi Pengembangan Lanjutan

- Belum ada pembayaran online (fitur offline/pending).
- Belum ada notifikasi email/WhatsApp otomatis.
- Belum ada manajemen kuota tiket otomatis (race condition potensial).
- Validasi & proteksi terhadap SQL Injection/perlindungan brute-force perlu ditingkatkan.
- UI dapat ditingkatkan agar responsif di mobile.