

**SOFTWARE REQUIREMENTS SPECIFICATION**  
**SISTEM BOOKING LAPANGAN OLAHRAGA**

**DISUSUN OLEH :**



**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS DAN INFORMATIKA**  
**UNIVERSITAS JENDERAL ACHMAD YANI**  
**TAHUN 2025**

# DAFTAR ISI

DAFTAR ISI .....	1
BAB I. INTRODUCTION .....	3
I.1 Overview .....	3
I.2 Goals and Objectives .....	3
I.3 Scope .....	3
I.4 Definitions .....	4
BAB II. GENERAL DESIGN CONSTRAINTS .....	5
II.1 Product Environment .....	5
BAB III. FUNCTIONAL REQUIREMENTS .....	<b>Error! Bookmark not defined.</b>
III.1 Modul Autentikasi (semua user) .....	<b>Error! Bookmark not defined.</b>
III.1.A. Use Case : Login & Redirect .....	<b>Error! Bookmark not defined.</b>
III.1.B. Use Case : Logout .....	<b>Error! Bookmark not defined.</b>
III.2 Modul Member .....	<b>Error! Bookmark not defined.</b>
III.2.A. Registrasi member .....	<b>Error! Bookmark not defined.</b>
III.2.B. Use Case : Cari & Booking lapangan .....	<b>Error! Bookmark not defined.</b>
III.2.C. Use Case : Upload Bukti Bayar .....	<b>Error! Bookmark not defined.</b>
III.3 Modul Admin .....	<b>Error! Bookmark not defined.</b>
III.3.A. Use Case : Validasi Pembayaran .....	<b>Error! Bookmark not defined.</b>
III.3.B. Use Case : Kelola Data Master (CRUD) .....	<b>Error! Bookmark not defined.</b>
III.3.C. Use Case : Laporan Keuangan .....	<b>Error! Bookmark not defined.</b>
BAB IV. NON-FUNCTIONAL REQUIREMENTS .....	<b>Error! Bookmark not defined.</b>
IV.1 Usability .....	<b>Error! Bookmark not defined.</b>
IV.2 Security .....	<b>Error! Bookmark not defined.</b>
IV.3 Performance .....	<b>Error! Bookmark not defined.</b>
IV.4 Reliabilty .....	<b>Error! Bookmark not defined.</b>

BAB V. EXTERNAL INTERFACE .....	<b>Error! Bookmark not defined.</b>
V.1 User Interface .....	<b>Error! Bookmark not defined.</b>
V.2 Software Interface .....	<b>Error! Bookmark not defined.</b>

# **BAB I. INTRODUCTION**

## **I.1 Overview**

Dokumen ini mendefinisikan spesifikasi kebutuhan perangkat lunak untuk Sistem Booking Lapangan Olahraga. Sistem ini merupakan aplikasi desktop yang dikembangkan menggunakan bahasa pemrograman Java. Aplikasi ini dirancang untuk mengelola seluruh ekosistem penyewaan lapangan, mulai dari pencarian jadwal oleh pelanggan hingga pelaporan pendapatan untuk pemilik (Owner).

Sistem menggunakan arsitektur Client-Server dengan database terpusat, memungkinkan integrasi data secara real-time antara berbagai peran pengguna (Pengguna, Karyawan, Admin, Owner) melalui aplikasi desktop yang terinstal di perangkat masing-masing.

## **I.2 Goals and Objectives**

1. Efisiensi Operasional: Menggantikan proses manual dengan sistem digital untuk mengurangi kesalahan pencatatan (double booking).
2. Kemudahan Akses: Memudahkan pengguna mencari dan memesan lapangan serta melakukan pembayaran secara transparan.
3. Kontrol Manajemen: Memberikan alat bagi Karyawan dan Admin untuk mengelola jadwal, pembayaran, dan pemeliharaan lapangan (maintenance).
4. Monitoring Bisnis: Menyediakan laporan akurat bagi Owner untuk memantau performa bisnis.

## **I.3 Scope**

Lingkup sistem mencakup fungsionalitas untuk empat aktor utama sesuai SDD:

1. Pengguna (Guest/Member): Registrasi, Login, Browse Lapangan, Booking, Pembayaran, Melihat Riwayat, dan Memberikan Rating.
2. Karyawan: Mengelola data lapangan (termasuk maintenance) dan jadwal operasional.
3. Admin: Mengelola data pengguna (User), memvalidasi pembayaran, dan mengelola booking.
4. Owner: Mengakses fitur khusus untuk men-generate dan melihat laporan keuangan/operasional.

## **I.4 Definitions**

1. SDD: Software Design Document.
2. Desktop App: Aplikasi native yang berjalan di atas OS Windows/Linux/macOS.
3. Booking: Proses reservasi slot waktu penggunaan lapangan.
4. Invoice: Dokumen tagihan digital yang dihasilkan otomatis setelah booking.

## **I.5 Document Conventions**

Prioritas fitur didefinisikan sebagai High (Kritis untuk bisnis), Medium (Penting tapi bisa ditunda), dan Low (Fitur tambahan/kosmetik).

## **I.6 Assumptions**

- Seluruh komputer pengguna terhubung ke internet untuk sinkronisasi data.
- Perangkat pengguna memiliki spesifikasi minimal RAM 4GB dan monitor resolusi 1366x768.
- Server database MySQL tersedia dan dapat diakses secara remote.

## **BAB II. GENERAL DESIGN CONSTRAINTS**

### **II.1 Product Environment**

- Platform: Aplikasi Desktop (Java SE).
- Database: MySQL (Relational Database Management System).
- Komunikasi: TCP/IP via JDBC Driver.
- Arsitektur: 3-Tier (Presentation Layer, Business Logic Layer, Data Access Layer).

### **II.2 User Characteristics**

Sesuai dengan desain use case, terdapat 4 karakteristik pengguna:

1. Pengguna (Member/Guest): Masyarakat umum. Membutuhkan antarmuka visual yang mudah (user-friendly) untuk booking cepat.
2. Karyawan: Staff operasional. Fokus pada pengelolaan teknis lapangan dan jadwal harian.
3. Admin: Pengelola sistem. Memiliki akses luas untuk manajemen user dan validasi keuangan.
4. Owner: Pemilik bisnis. Hanya membutuhkan akses ringkas untuk melihat dashboard laporan dan statistik.

### **II.3 Mandated Constrains**

- Bahasa Pemrograman: Java (Wajib).
- GUI Library: Swing atau JavaFX.
- Distribusi: File Installer (.exe / .jar).
- Keamanan Data: Password pengguna wajib dienkripsi (Hashing).

### **II.4 Potential System Evolution**

Sistem didesain modular agar di masa depan dapat mendukung integrasi dengan Payment Gateway otomatis (Virtual Account/E-Wallet) tanpa mengubah struktur inti database.

## **BAB III. NONFUNCTIONAL REQUIREMENTS**

### **III.1 Usability Requirements**

- Antarmuka harus membedakan akses menu berdasarkan Role saat login.
- Respon visual (feedback) harus muncul < 1 detik setelah tombol ditekan.
- Desain UI harus konsisten (warna, font, layout) di seluruh modul.

### **III.2 Operational Requirements**

- Sistem harus mampu menangani minimal 50 transaksi booking secara bersamaan (concurrency).
- Mendukung fitur Backup & Restore database untuk mencegah kehilangan data.

### **III.3 Performance Requirements**

- Login process < 2 detik.
- Generate Laporan Bulanan < 5 detik.
- Sinkronisasi status lapangan (Available/Booked) harus real-time.

### **III.4 Security Requirements**

- Authentication: Username & Password terverifikasi database.
- Authorization: Penerapan RBAC (Role-Based Access Control) yang ketat. Karyawan tidak boleh bisa menghapus User, Pengguna tidak boleh mengakses menu Admin.

### **III.5 External Interface**

- User Interface: Form input, Tabel data, Kalender visual, Notifikasi Pop-up.
- Hardware Interface: Mendukung input Keyboard/Mouse dan pencetakan Invoice ke Printer standar.

## **BAB IV. SYSTEM FEATURES**

### **IV.1 Feature : Autentikasi & Manajemen Akun**

#### **IV.1.A. Description & Priority**

Mengelola akses masuk dan pendaftaran pengguna baru ke dalam sistem.

Priority: High

#### **IV.1.B. Use Case : Login**

- Actor: Semua Aktor (Pengguna, Karyawan, Admin, Owner)
- Flow:
  1. User memasukkan email dan password.
  2. Sistem memvalidasi kredensial.
  3. Sistem mengarahkan user ke dashboard sesuai peran (misal: Owner ke Halaman Laporan, Member ke Halaman Pencarian).

#### **IV.1.C. Use Case : Register**

- Actor: Guest (Calon Pengguna)
- Flow:
  1. Guest mengisi form pendaftaran (Nama, Email, Password, No Telp).
  2. Sistem menyimpan data sebagai role 'Member'.
  3. Sistem mengirim notifikasi sukses.

#### **IV.1.D. Use Case : Kelola User**

- Actor: Admin
- Description: Admin dapat melihat daftar pengguna, mengedit profil, atau menonaktifkan akun yang melanggar aturan.



## **IV.2 Feature : Manajemen Booking & Transaksi**

### **IV.2.A. Description & Priority**

Inti dari aplikasi yang memungkinkan penyewaan lapangan dan pembayaran.

Priority: High

### **IV.2.B. Use Case : Browse & Booking Lapangan**

- Actor: Pengguna (Member)
- Flow:
  1. Member mencari lapangan berdasarkan lokasi/tipe.
  2. Member memilih slot waktu pada jadwal.
  3. Sistem membuat booking sementara (status: Pending Payment).
  4. Member melanjutkan ke pembayaran.

### **IV.2.C. Use Case : Melakukan Pembayaran**

- Actor: Pengguna (Member)
- Description: Member mengunggah bukti pembayaran atau input data pembayaran untuk booking yang telah dibuat.

### **IV.2.D. Use Case : Kelola Pembayaran (Validasi)**

- Actor: Admin
- Flow:
  1. Admin menerima daftar pembayaran masuk.
  2. Admin memverifikasi bukti bayar.
  3. Jika valid, status booking diubah menjadi Confirmed dan invoice diterbitkan.
  4. Jika invalid, pembayaran ditolak.

### **IV.2.E. Use Case : Melihat Riwayat Booking**

- Actor: Pengguna (Member)
- Description: Member dapat melihat daftar transaksi masa lalu dan status booking aktif.

### **IV.3 Feature : Manajemen Operasional Lapangan**

#### **IV.3.A. Description**

Pengelolaan aset fisik dan ketersediaan jadwal oleh staff internal.

Priority: High

#### **IV.3.B. Use Case : Kelola Lapangan**

- Actor: Karyawan
- Flow (Sesuai SDD Scenario):
  1. Karyawan memilih opsi Tambah/Edit/Hapus lapangan.
  2. Untuk Hapus: Sistem mengecek apakah ada booking aktif di lapangan tersebut.
  3. Jika tidak ada booking aktif, data lapangan dihapus.
  4. Jika ada, sistem menolak penghapusan.

#### **IV.3.C. Use Case : Kelola Jadwal**

- Actor: Owner, Admin
- Description: Mengatur jam operasional venue, hari libur, atau menutup lapangan untuk keperluan Maintenance.

### **IV.4 Feature : Pelaporan & Evaluasi**

#### **IV.4.A. Description**

Fitur analitik dan umpan balik untuk pemilik bisnis.

Priority: Medium

#### **IV.4.B. Use Case : Generate Laporan**

- Actor: Owner
- Flow:
  1. Owner memilih periode laporan (Harian/Bulanan).
  2. Sistem mengalkulasi total pendapatan dan jumlah booking.
  3. Sistem menyajikan laporan dalam bentuk tabel/grafik yang bisa diekspor (PDF/Excel).

#### **IV.4.C. Use Case : Memberikan rating**

- Actor: Pengguna (Member)
- Description: Setelah booking selesai (Completed), member dapat memberikan bintang (1-5) dan ulasan teks terhadap lapangan yang digunakan.

