**Penjelasan dan Analisis Sistem Informasi Booking Lapangan Sepak Bola Berbasis Web**

**A. Deskripsi Umum Sistem**

**Penjelasan :**

Sistem yang dibangun adalah sebuah aplikasi web bernama "Booking Lapangan" yang berfungsi sebagai platform digital untuk manajemen dan pemesanan lapangan sepak bola. Tujuan utama dari sistem ini adalah untuk menggantikan proses pemesanan manual yang rentan terhadap kesalahan (seperti *double-booking*) dengan sistem yang terotomatisasi, efisien, dan transparan.

Aplikasi ini melayani dua jenis pengguna utama:

1. **Pengguna (User/Tamu):** Pengguna umum yang dapat melihat informasi lapangan, melakukan registrasi, login, memesan jadwal yang tersedia, melakukan konfirmasi pembayaran, serta melihat riwayat pemesanan mereka.
2. **Administrator (Admin):** Pengguna dengan hak akses tertinggi yang bertugas mengelola seluruh aspek sistem, termasuk mengelola konten situs (galeri), fasilitas (jadwal dan harga), memverifikasi pembayaran, mengonfirmasi pesanan, serta mengelola data pengguna, pesan kontak, dan feedback.

**Analisis :**

Pembangunan sistem ini didasari oleh kebutuhan untuk meningkatkan efisiensi operasional dan pengalaman pelanggan dalam proses penyewaan lapangan. Secara konvensional, proses ini seringkali melibatkan telepon atau pesan instan yang tidak efisien dan sulit dilacak.

Keunggulan sistem ini adalah:

* **Aksesibilitas:** Pengguna dapat mengecek ketersediaan jadwal dan melakukan booking kapan saja dan di mana saja selama terhubung dengan internet.
* **Transparansi:** Jadwal yang sudah dibooking akan langsung tidak tersedia bagi pengguna lain, menghilangkan risiko *double-booking*.
* **Efisiensi Manajemen:** Admin dapat mengelola semua data dari satu dasbor terpusat, mulai dari pesanan masuk, verifikasi pembayaran, hingga pengelolaan konten, yang secara signifikan mengurangi beban kerja administratif.
* **Terdokumentasi:** Semua transaksi dan interaksi pengguna (pesan, feedback) tercatat secara digital, memudahkan proses pelacakan dan audit di kemudian hari.

**B. Teknologi yang Digunakan**

**Penjelasan :**

Aplikasi ini dibangun menggunakan tumpukan teknologi (technology stack) yang modern, populer, dan andal untuk pengembangan aplikasi web berbasis PHP.

* **Bahasa Pemrograman:** PHP Versi 8.x, memanfaatkan fitur-fitur modern seperti *type-hinting* dan sintaks yang lebih bersih.
* **Framework Backend:** CodeIgniter 4, sebuah framework PHP yang ringan, cepat, dan modern. Framework ini digunakan sebagai fondasi utama aplikasi, menyediakan struktur, pustaka, dan alat bantu untuk mempercepat pengembangan.
* **Arsitektur:** Model-View-Controller (MVC), sebuah pola desain arsitektur yang dipaksakan oleh CodeIgniter 4 untuk memisahkan logika bisnis, data, dan tampilan.
* **Database:** MariaDB (kompatibel dengan MySQL), digunakan sebagai sistem manajemen database relasional untuk menyimpan semua data persisten seperti data pengguna, pesanan, jadwal, dan lain-lain.
* **Frontend:** HTML5, CSS3, dan JavaScript standar. Framework CSS **Bootstrap 4** digunakan untuk membangun antarmuka yang responsif dan konsisten dengan cepat.
* **Manajemen Dependensi:** Composer, digunakan sebagai manajer paket untuk PHP, terutama untuk instalasi dan pembaruan framework CodeIgniter itu sendiri.
* **Web Server Lokal:** Apache, yang dijalankan melalui paket XAMPP untuk lingkungan pengembangan.

**Analisis :**

Pemilihan teknologi ini didasarkan pada beberapa pertimbangan strategis:

* **CodeIgniter 4** dipilih karena performanya yang cepat, kurva belajar yang relatif landai, dan dokumentasi yang sangat lengkap. Strukturnya yang berbasis MVC mendorong praktik pengkodean yang baik, menghasilkan kode yang lebih terorganisir, mudah dirawat, dan skalabel.
* **PHP dan MariaDB/MySQL** adalah kombinasi klasik yang telah teruji dalam membangun jutaan aplikasi web. Dukungan komunitas yang luas dan integrasi yang mulus menjadikannya pilihan yang aman dan andal.
* Penggunaan **Bootstrap** secara signifikan mempercepat pengembangan antarmuka (UI). Tanpa harus menulis banyak CSS dari awal, kita bisa langsung fokus pada fungsionalitas sambil tetap mendapatkan tampilan yang modern dan responsif di berbagai perangkat (desktop, tablet, mobile).
* **Composer** adalah standar industri dalam ekosistem PHP modern. Penggunaannya memastikan bahwa semua pustaka dan framework yang kita gunakan memiliki versi yang tepat dan mudah dikelola.

**C. Fitur Utama dan Alur Kerja**

**1. Sisi Pengguna (User-Facing)**

**Penjelasan Fitur :**

* **Halaman Beranda:** Halaman utama yang menampilkan slogan dan ajakan bertindak (call-to-action) untuk melakukan booking atau melihat galeri.
* **Autentikasi:** Pengguna dapat melakukan **Registrasi** untuk membuat akun baru dan **Login** untuk masuk ke sistem. Sistem membedakan antara sesi tamu, user, dan admin.
* **Galeri:** Menampilkan kumpulan foto lapangan. Setiap foto dapat diklik untuk melihat detailnya di halaman baru yang berisi gambar ukuran penuh dan deskripsi panjang. Halaman ini juga dilengkapi peta lokasi.
* **Form Booking:** Halaman interaktif di mana pengguna memilih tanggal. Menggunakan **AJAX (Asynchronous JavaScript and XML)**, sistem akan mengambil daftar slot waktu yang tersedia untuk tanggal dan hari (Weekday/Weekend) tersebut secara dinamis tanpa me-refresh halaman. Slot yang sudah dibooking atau sudah lewat waktunya (untuk hari ini) tidak akan ditampilkan. Form kontak (nama, email, no. HP) akan terisi otomatis jika pengguna sudah login.
* **Proses Pembayaran:** Setelah booking, pengguna diarahkan ke halaman konfirmasi yang berisi rincian pesanan dan instruksi pembayaran. Pengguna dapat mengunggah bukti transfer, yang kemudian mengubah status pembayaran menjadi "paid".
* **Riwayat Pemesanan:** Pengguna yang login dapat melihat daftar semua pesanan yang pernah mereka buat, lengkap dengan status pembayaran dan status pemesanan (misalnya, pending, confirmed, completed). Dari sini, pengguna dapat memberikan feedback untuk pesanan yang sudah selesai atau menghapus riwayat.
* **Profil Pengguna:** Pengguna dapat mengubah data pribadi mereka, termasuk nama, email, password, dan foto profil.

**Analisis Alur Kerja Pengguna :**  
Seorang pengguna baru akan melalui alur: Registrasi -> Login -> Melihat Galeri/Beranda -> Masuk ke Halaman Booking -> Memilih Tanggal (AJAX memuat jadwal) -> Memilih Slot -> Mengisi Nomor HP -> Submit Pesanan -> Menerima detail di Halaman Konfirmasi -> Melakukan Transfer -> Mengunggah Bukti Pembayaran -> Kembali ke Halaman Riwayat untuk memantau status -> Memberikan Feedback setelah pesanan selesai -> Logout. Alur ini dirancang untuk menjadi intuitif dan memberikan feedback yang jelas di setiap langkahnya.

**2. Sisi Administrator (Admin Panel)**

**Penjelasan Fitur :**

* **Dasbor:** Halaman utama admin yang menampilkan data statistik ringkas seperti total pesanan, pesanan pending, total pendapatan, dan jumlah pengguna/galeri.
* **Manajemen Pembayaran (CRUD):** Admin dapat melihat seluruh daftar pesanan yang masuk, melakukan pencarian, melihat detail setiap pesanan (termasuk bukti transfer yang diunggah pengguna), **mengubah status** pesanan (misalnya dari pending menjadi confirmed), menghapus data pesanan, dan mencetak struk pembayaran.
* **Manajemen Slot Waktu (CRUD):** Admin memiliki kontrol penuh untuk menambah, mengedit, dan menghapus slot waktu yang tersedia, termasuk menentukan hari (Weekday/Weekend), jam mulai, jam selesai, dan harga untuk setiap slot.
* **Manajemen Galeri (CRUD):** Admin dapat mengunggah foto baru ke galeri, memberikan judul dan deskripsi panjang, mengedit data yang sudah ada, serta menghapus foto.
* **Manajemen Interaksi:** Terdapat halaman khusus untuk melihat **Pesan Kontak** dan **Feedback Pengguna** yang masuk, lengkap dengan fungsi pencarian dan hapus.
* **Manajemen Pengguna (CRUD):** Admin dapat melihat daftar semua pengguna, mencari pengguna, serta menambah, mengedit (termasuk mengubah role), dan menghapus data pengguna.

**Analisis Alur Kerja Admin :**  
Admin akan login -> Melihat ringkasan di Dasbor -> Mendapat notifikasi (implisit) adanya pesanan baru -> Membuka Manajemen Pembayaran -> Memeriksa detail pesanan yang statusnya "Paid" -> Melihat bukti transfer -> Mengubah status pesanan menjadi "Confirmed" -> Pengguna kini bisa mencetak struk. Admin juga secara berkala bisa mengupdate galeri, jadwal, atau melihat pesan yang masuk.

**D. Analisis Keamanan dan Performa**

**Analisis Keamanan :**  
Aplikasi ini telah menerapkan beberapa praktik keamanan web standar:

1. **SQL Injection:** Pencegahan dilakukan secara otomatis oleh Query Builder dan Model dari CodeIgniter 4, yang menggunakan *prepared statements* dan *parameter binding*, sehingga input dari pengguna tidak bisa merusak query SQL.
2. **Cross-Site Scripting (XSS):** Semua data yang ditampilkan di View dilewatkan melalui fungsi esc() dari CodeIgniter, yang berfungsi untuk membersihkan output dan mengubah karakter HTML berbahaya menjadi entitas yang aman.
3. **Cross-Site Request Forgery (CSRF):** Setiap form di dalam aplikasi dilindungi dengan memanggil fungsi csrf\_field(), yang akan menghasilkan token tersembunyi. Token ini akan divalidasi oleh CodeIgniter pada setiap permintaan POST, mencegah serangan dari situs lain.
4. **Akses Tidak Sah:** Sistem rute dilindungi oleh **Filter**. LoginFilter memastikan hanya pengguna yang sudah login yang bisa mengakses halaman seperti riwayat dan booking. AdminFilter memberikan lapisan keamanan tambahan, memastikan hanya pengguna dengan role = 'admin' yang bisa mengakses semua URL di bawah /admin.
5. **Keamanan Password:** Password pengguna tidak disimpan sebagai teks biasa. UserModel menggunakan *hook* beforeInsert dan beforeUpdate untuk secara otomatis meng-hash password menggunakan fungsi password\_hash() (standar BCRYPT) sebelum menyimpannya ke database.

**Analisis Performa :**

* **AJAX pada Form Booking:** Penggunaan AJAX untuk memuat jadwal secara dinamis adalah keputusan kunci untuk meningkatkan pengalaman pengguna. Ini mencegah seluruh halaman dimuat ulang hanya untuk menampilkan daftar slot, membuat aplikasi terasa lebih cepat dan responsif.
* **Query Database yang Efisien:** Pada beberapa halaman admin (seperti daftar pembayaran), digunakan JOIN untuk mengambil data dari beberapa tabel (misalnya pemesanan dan users) dalam satu kali permintaan ke database, yang lebih efisien daripada melakukan beberapa query terpisah.
* **Konfigurasi Debug Toolbar:** Selama pengembangan, Debug Toolbar dikonfigurasi untuk tidak memuat kolektor data yang berat (seperti Sesi), sehingga mempercepat waktu muat halaman tanpa menghilangkan fungsionalitas *debugging* yang penting.

**Penjelasan dan Analisis Perancangan Sistem**

Perancangan sistem merupakan tahap krusial sebelum implementasi kode untuk memvisualisasikan alur kerja, aliran data, dan interaksi antar komponen. Dalam proyek ini, digunakan beberapa alat perancangan standar yaitu Flowchart, Diagram Konteks, dan Data Flow Diagram (DFD) Level 1 & 2.

**A. Flowchart: Alur Proses Booking Pengguna**

**Penjelasan Flowchart :**

Flowchart ini dirancang untuk memetakan *user journey* atau langkah-langkah logis yang dialami oleh seorang pengguna dari awal hingga akhir proses pemesanan. Diagram ini digambarkan secara horizontal (kiri ke kanan) untuk merepresentasikan alur waktu yang linear. Alur ini dibagi menjadi beberapa fase utama:

1. **Mulai:** Pengguna membuka halaman beranda.
2. **Autentikasi:** Sistem memeriksa apakah pengguna sudah login. Jika tidak, ia akan diarahkan ke alur Login/Registrasi. Jika ya, ia bisa langsung melanjutkan ke proses booking.
3. **Proses Booking:** Pengguna memilih menu booking, memilih tanggal dan slot waktu yang tersedia. Untuk tamu, ada langkah tambahan untuk mengisi data diri. Proses ini diakhiri dengan aksi pengguna mengklik tombol "Lanjutkan Pemesanan".
4. **Backend & Konfirmasi:** Setelah tombol diklik, sistem menyimpan data pesanan ke database dan menampilkan halaman konfirmasi yang berisi detail pesanan dan instruksi pembayaran.
5. **Pembayaran:** Pengguna melakukan pembayaran di luar sistem (transfer manual), kemudian kembali ke aplikasi untuk mengunggah bukti transfer. Sistem kemudian menyimpan bukti tersebut dan mengubah status pembayaran.
6. **Selesai:** Pengguna diarahkan ke halaman riwayat untuk melihat status pesanannya yang sudah ter-update. Ini menandakan satu siklus pemesanan telah selesai.

**Analisis Flowchart :**

Penggunaan flowchart ini sangat efektif untuk mengidentifikasi semua kemungkinan langkah dan kondisi yang akan dihadapi pengguna.

* **Kejelasan Alur:** Diagram ini dengan jelas menunjukkan adanya percabangan (kondisi) pada tahap "Sudah Login?" dan "Sudah Transfer?", yang menjadi dasar dalam perancangan logika if-else di dalam kode.
* **Identifikasi Kebutuhan Halaman:** Dari flowchart ini, kita dapat langsung mengidentifikasi halaman (View) apa saja yang perlu dibuat: Halaman Beranda, Login, Register, Booking, Konfirmasi, dan Riwayat.
* **Fokus pada Pengguna:** Dengan memetakan alur dari sudut pandang pengguna, kita dapat merancang antarmuka yang lebih intuitif dan memastikan setiap langkah memiliki tujuan yang jelas, mulai dari pemilihan jadwal hingga konfirmasi akhir.

**B. Diagram Konteks (DFD Level 0)**

**Penjelasan Diagram Konteks :**

Diagram Konteks (DFD Level 0) berfungsi sebagai pandangan "mata elang" dari keseluruhan sistem. Tujuannya adalah untuk mendefinisikan batasan sistem dan mengidentifikasi entitas eksternal yang berinteraksi dengannya, serta aliran data utama yang masuk dan keluar.

Dalam diagram ini:

* **Proses Tunggal:** Seluruh "Sistem Booking Lapangan" direpresentasikan sebagai satu lingkaran atau proses tunggal, tanpa mempedulikan detail internalnya.
* **Entitas Eksternal:** Diidentifikasi ada dua aktor utama yang berinteraksi dengan sistem, yaitu **Pengguna** dan **Admin**.
* **Aliran Data (Data Flow):** Panah-panah menunjukkan data apa saja yang dipertukarkan. Contohnya, Pengguna mengirimkan "Data Booking" ke sistem, dan sistem mengirimkan kembali "Jadwal Tersedia" kepada Pengguna. Demikian pula, Admin mengirimkan "Update Status Pesanan" dan menerima "Laporan & Daftar Pesanan".

**Analisis Diagram Konteks :**

Diagram Konteks ini sangat penting pada tahap awal perancangan karena:

* **Mendefinisikan Ruang Lingkup:** Dengan jelas menggambarkan apa saja yang menjadi bagian dari sistem dan apa yang berada di luarnya. Ini membantu tim pengembang untuk tetap fokus pada fitur-fitur yang telah disepakati.
* **Identifikasi Input dan Output Utama:** Diagram ini memaksa kita untuk memikirkan semua data yang diperlukan oleh sistem (input) dan semua informasi yang harus dihasilkan oleh sistem (output). Ini menjadi dasar untuk perancangan database dan antarmuka.
* **Komunikasi Sederhana:** Bentuknya yang sangat sederhana membuatnya mudah dipahami oleh semua pemangku kepentingan, termasuk yang tidak memiliki latar belakang teknis, untuk menyepakati gambaran umum dari proyek yang akan dibangun.

**C. Data Flow Diagram (DFD) Level 1**

**Penjelasan DFD Level 1 :**

DFD Level 1 adalah dekomposisi atau "pembedahan" dari proses tunggal pada Diagram Konteks. Ia memecah sistem menjadi beberapa proses utama yang lebih spesifik dan menunjukkan bagaimana data mengalir di antara proses-proses tersebut serta ke penyimpanan data (Data Store).

Proses-proses utama yang diidentifikasi adalah:

1. **Manajemen Pengguna:** Menangani registrasi dan login.
2. **Proses Pemesanan:** Menangani semua alur terkait booking, dari pengecekan jadwal hingga konfirmasi pembayaran.
3. **Manajemen Konten & Fasilitas:** Proses yang digunakan oleh admin untuk mengelola data master seperti jadwal, harga, dan galeri.
4. **Manajemen Interaksi:** Mengelola data yang masuk dari pengguna seperti pesan kontak dan feedback.
5. **Laporan & Dasbor:** Proses yang mengumpulkan data dari berbagai penyimpanan untuk disajikan kepada admin.

Penyimpanan Data (Data Store) yang diidentifikasi sesuai dengan tabel-tabel utama di database: users, pemesanan, jadwal\_master, galeri, pesan\_kontak, dan feedback.

**Analisis DFD Level 1 :**

DFD Level 1 memberikan pemahaman yang lebih dalam tentang arsitektur internal sistem:

* **Struktur Fungsional:** Diagram ini mengelompokkan fungsionalitas ke dalam modul-modul atau proses-proses logis. Pengelompokan ini menjadi dasar yang sangat baik untuk membuat struktur Controller di dalam arsitektur MVC (misalnya, AuthController, BookingController, AdminController).
* **Ketergantungan Data:** Dengan jelas menunjukkan proses mana yang membaca atau menulis ke tabel database tertentu. Contohnya, **Proses Pemesanan (2.0)** membaca dari jadwal\_master dan menulis ke pemesanan. Ini membantu dalam merancang query database yang efisien.
* **Dasar Perancangan Modul:** Setiap lingkaran proses pada DFD Level 1 dapat dianggap sebagai sebuah modul fungsional yang bisa dikembangkan secara terpisah namun tetap terintegrasi, memudahkan pembagian kerja dalam tim.

**D. Data Flow Diagram (DFD) Level 2: Detail Proses Pemesanan**

**Penjelasan DFD Level 2 :**

DFD Level 2 ini adalah "zoom in" yang lebih detail pada salah satu proses dari DFD Level 1, yaitu **Proses 2.0: Proses Pemesanan**. Tujuannya adalah untuk memetakan sub-proses dan aliran data yang terjadi di dalam alur booking.

Sub-proses yang diidentifikasi adalah:

* **2.1 Cek Jadwal Tersedia:** Menerima input tanggal dari pengguna, membaca data dari tabel jadwal\_master dan pemesanan, lalu mengirimkan kembali daftar slot yang tersedia.
* **2.2 Simpan Pemesanan Baru:** Menerima pilihan slot dari pengguna, lalu menyimpan data baru ke tabel pemesanan dengan status pending.
* **2.3 Proses Konfirmasi Bayar:** Menerima unggahan bukti bayar dari pengguna, lalu meng-update status bayar di tabel pemesanan menjadi paid.
* **2.4 Verifikasi oleh Admin:** Menerima input persetujuan dari Admin, lalu meng-update status pesanan di tabel pemesanan menjadi confirmed.

**Analisis DFD Level 2 :**

DFD Level 2 sangat berguna bagi programmer saat akan menulis kode karena:

* **Spesifikasi Fungsi/Method:** Setiap sub-proses (misalnya, "2.1 Cek Jadwal Tersedia") secara langsung dapat diterjemahkan menjadi sebuah fungsi atau *method* di dalam Controller (misalnya, fungsi getJadwal() di BookingController).
* **Detail Aliran Data:** Ini menunjukkan dengan sangat spesifik data apa yang dibutuhkan oleh setiap fungsi. Misalnya, fungsi getJadwal() membutuhkan input "Tanggal" dan mengakses dua tabel (jadwal\_master dan pemesanan) untuk menghasilkan output "Daftar Slot Tersedia".
* **Validasi Logika:** Sebelum menulis kode, diagram ini membantu memvalidasi apakah alur logikanya sudah benar dan tidak ada langkah yang terlewat, seperti proses notifikasi ke admin atau pembaruan status di setiap tahap. Ini mengurangi risiko bug logika selama implementasi.