

BAB II

TINJAUAN PUSTAKA

2.1. Teori Terkait

Untuk membantu perancangan sistem ini terdapat penelitian sebelumnya yang relevan tentang Sistem informasi reservasi lapangan futsal. Salah satunya penelitian yang telah dilakukan oleh Muhammad fatih Fahmi Fadhlurrahman dan Dwi Ade (2020) yang berjudul “Aplikasi Penyewaan Lapangan Futsal Berbasis *Web*”. Penelitian ini dilakukan karena Ansha Futsal masih menggunakan metode manual dalam proses pengajuan sewa lapangan, serta pendataan laporan penyewaan yang masih menggunakan pencatatan buku. metode penyewaan yang masih manual ini memberikan dampak masalah kepada calon pelanggan, yaitu ketika sudah sampai di lokasi dan ternyata jam bermain yang diinginkan tidak ada, dan akan merasa kecewa dan hanya membuang-buang waktu saja. Itu menyebabkan akan adanya penurunan minat bermain pada ansha futsal yang akan berdampak juga kepada berkurangnya pendapatan pemilik lapangan. Oleh karena itu dibuat sebuah Aplikasi penyewaan lapangan futsal berbasis *web*, yang dibangun dengan *software* Balsamiq *Mockups* 3 untuk membuat tampilan *user interface* nya, *software* Sublime Text sebagai *script editor*. Sehingga menghasilkan sebuah aplikasi berbasis *website* untuk penyewaan lapangan futsal yang dapat diakses setiap saat sehingga menambah efisiensi waktu ketika ingin melakukan pemesanan [1]

Adapun penelitian yang juga relevan dilakukan oleh Achmad Nakhrowi, dkk (2017) Dalam jurnal berjudul “Implementasi *Framework laravel* Pada Sistem Informasi Pemesanan Penggunaan Lapangan Futsal Berbasis *Web* Di Zona6 Futsal Semarang” dengan menerapkan metode *Waterfall* dalam pengembangan sistem. Penelitian ini dilatar belakangi dengan proses bisnis pada tempat sewa lapangan futsal pada umumnya masih mengharuskan pelanggan untuk datang dalam melakukan penyewaan dan mengatur jadwal penyewaan yang diinginkan. Sehingga pelanggan tidak mengetahui jadwal yang masih kosong secara langsung. Setiap hari petugas melakukan pencatatan penyewaan dari pelanggan ke buku pesanan. Pada hari penyewaan, pelanggan melakukan pembayaran penyewaan. Hal ini juga dapat menimbulkan kesalahan dalam pencatatan. Sehingga dibuatnya sebuah sistem pemesanan lapangan futsal dengan menggunakan laravel sebagai *framework* pembuatan *webiste* [3].

Penelitian juga telah dilakukan oleh Dani Hidayatullah dkk (2022) dengan judul “Sistem Informasi Reservasi Pelayanan dan Penyewaan Fasilitas Lapangan Futsal Berbasis *Web* Dengan Metode *Waterfall*”. Penelitian ini bertujuan untuk mengembangkan sistem informasi berbasis *web* untuk penyewaan lapangan futsal pada Bandar Lampung *Sport Center* (BSC). Sistem ini diharapkan dapat mengatasi permasalahan yang ada dengan menggantikan metode manual yang masih digunakan (seperti pencatatan jadwal dan laporan keuangan menggunakan buku dan papan tulis), serta

dapat meningkatkan efisiensi, kualitas pelayanan, serta kemudahan dalam pengelolaan jadwal dan laporan keuangan.[2]

Penelitian yang relevan juga dilakukan oleh Nurul Khotimah (2022) dengan judul “Rancang Bangun Sistem Informasi Reservasi Lapangan Futsal Berbasis *Web Allium Futsal Caruban*”. Tujuan dibangunnya sistem informasi reservasi lapangan futsal ini adalah untuk mempermudah proses reservasi pada Allium Futsal Caruban. Membantu pengelola lapangan dalam pengolahan data dan informasi dan memudahkan pelanggan dalam proses reservasi. Pelanggan tidak perlu lagi datang ke tempat untuk melakukan reservasi dan menentukan jadwal yang diinginkan. Adapun Metode yang digunakan untuk mengembangkan sistem informasi reservasi lapangan futsal berbasis *web* ini yaitu dengan menggunakan metode RAD (*Rapid Application Development*) dan model UML untuk perancangannya. Sistem ini dibangun menggunakan bahasa pemrograman PHP, *database* MySQL, dan menggunakan *Visual Studio Code* sebagai *test editor*. Hasil dari penelitian ini adalah Sistem Informasi Reservasi Lapangan Futsal Berbasis *Web* Di Allium Futsal Caruban yang dapat diakses oleh 2 pengguna, yaitu admin dan *user*. Metode pengujian yang digunakan yaitu metode *black box*, hasil dari pengujian ini adalah semua menu berfungsi dengan normal dan berdasarkan dari proses tersebut fitur-fitur yang ada dapat dijalankan sesuai yang dibutuhkan [4].

Penelitian terdahulu juga dilakukan oleh Muhammad Rizki Augusta dkk (2021), dengan judul Pengembangan Sistem Layanan Pemesanan Lapangan Futsal dengan memanfaatkan Teknologi *Payment Gateway* (Studi

Kasus: Marcella Futsal Jombang). Penelitian ini menggunakan metode *waterfall* dalam pengembangan sistem layanan pemesanan lapangan futsal. Latar belakang dibuatnya sistem layanan pemesanan lapangan futsal untuk mengatasi beberapa permasalahan yang dihadapi dalam proses pemesanan, seperti terjadinya dua orang pemesan lapangan pada waktu yang sama dan penggunaan metode penyimpanan data yang masih manual. *Payment gateway* yang digunakan dalam sistem layanan pemesanan lapangan futsal ini yaitu Midtrans. Midtrans berfungsi untuk mempermudah pelanggan dalam melakukan transaksi pembayaran, baik untuk uang muka maupun biaya sewa lapangan. Sistem ini juga menyediakan fitur yang memungkinkan karyawan untuk mengarsipkan data pemesanan dengan lebih mudah, karena semua transaksi yang diterima oleh sistem dapat ditampilkan dalam sebuah *web* yang disediakan oleh Midtrans [5].

Adapun penelitian yang berjalan dan akan dilakukan adalah “Sistem Informasi Reservasi Lapangan Futsal Sipelem Berbasis Website Menggunakan Framework Laravel”.

2.2. Landasan Teori

2.2.1. Reservasi

Reservasi adalah pemesanan suatu fasilitas yang dilakukan sebelum calon pelanggan menggunakan fasilitas atau layanan jasa. Sebutan lain dari reservasi ialah *booking*, yang berarti memesan sebelum pelanggan datang ke tempat layanan jasa [2].

2.2.2. Sistem Informasi

Sistem informasi merupakan salah satu hasil pengembangan teknologi informasi yang banyak dimanfaatkan oleh organisasi untuk menjalankan kegiatan operasionalnya [4]. Sistem informasi lapangan futsal berbasis *web* ini sistem informasi tersebut tidak hanya berfungsi sebagai sarana untuk mengatur jadwal lapangan tetapi juga menyediakan *platform* untuk reservasi lapangan futsal.

2.2.3. Payment Gateway

Payment gateway adalah salah satu cara untuk memproses transaksi elektronik. fitur - fitur utama dari *payment gateway* adalah enkripsi pembayaran dan data pribadi, komunikasi antara lembaga keuangan yang terkait, *business* dan *customer*, serta otorisasi pembayaran [6] .

2.2.4. Framework Laravel



Gambar 2. 1 Logo Laravel

Framework adalah komponen pemrograman yang siap re-use (bisa digunakan ulang) kapan saja, sehingga programmer tidak harus membuat skrip yang sama untuk tugas yang sama. Sedangkan Laravel adalah sebuah *framework web* berbasis PHP yang *open-source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan

untuk pengembangan aplikasi *web* yang menggunakan pola MVC [7]

2.2.5. PHP (Hypertext Preprocessor)



Gambar 2.2. Logo PHP

Dalam menggunakan *framework laravel* PHP menjadi bahasa inti yang digunakan untuk membangun aplikasi *web* berbasis Laravel. PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada server tidak dijalankan pada *client* [8].

2.2.6. MySQL



Gambar 2.3. Logo MySQL

Sistem ini memanfaatkan MySQL sebagai *database*. MySQL adalah sebuah *database Open Source* populer di dunia. Penggunaannya sebagai *database* bahasa pemrograman populer seperti PHP dan

Java adalah hal umum. Untuk memudahkan penggunaan Mysql, terdapat *Software open source* berbasis GUI, yakni phpmyadmin yang dapat di *download* secara gratis [9].

2.2.7. Website

Sistem informasi yang akan dibuat adalah berbasis *Website*, definisi *Website* merupakan kumpulan informasi yang terdiri dari halaman *web* yang saling terhubung satu sama lain yang disediakan secara perorangan, kelompok, atau pun organisasi. Situs *web* yang baik menampilkan visual yang menarik dan berfungsi sesuai dengan kebutuhan pengguna [10].

2.2.8. Unified Modelling Language (UML)

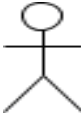
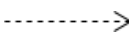
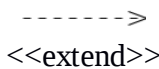
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek [11]. UML tersusun atas sejumlah elemen grafis membentuk 9 diagram-diagram. Dalam penelitian ini melakukan desain hanya 4 diagram yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. Berikut pengertian dan bentuk simbol-simbol dari diagram :

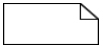
1. Usecase Diagram

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case*

merepresentasikan sebuah interaksi antara aktor dengan sistem. Misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [12]. Berikut simbol-simbol *use case*:

Tabel 2.1. Simbol Usecase

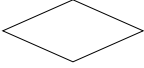
No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses atau sistem yang berinteraksi dengan sistem lain.
2		<i>Dependency</i>	Hubungan perubahan yang terjadi pada suatu elemen mandiri
3		<i>Generalization</i>	Hubungan generalisasi (umum-khusus) antar dua buah <i>use case</i>
4		<i>Include</i>	Relasi use case tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri.
5		<i>Extend</i>	Relasi tambahan yang Menunjukkan <i>use case</i> lain memerlukan <i>use case</i> . syarat untuk dijalankan atau agar fungsinya dapat berjalan dengan baik.
6		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i>

No.	Gambar	Nama	Keterangan
			yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut simbol-simbol dari *activity* diagram :

Tabel 2.2. Simbol Activity Diagram

No.	Simbol	Nama	Penjelasan
1		<i>Activity</i>	Memperlihatkan bagaimana kelas saling berinteraksi
2		<i>Decision</i>	Asosiasi percabangan jika ada pilihan yang lebih dari satu
3		<i>Initial Node</i>	Bagaimana objek diawali. Hanya ada satu initial dalam satudiagram.
4		<i>Activity Final Node</i>	Bagaimana objek diakhiri. Dalam satu diagram minimal ada satu final state
5		<i>Fork Node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran
6		<i>Control Flow</i>	Menunjukkan urutan aliran aktivitas.
7		<i>Partition</i>	Simbol yang membatasi aktivitas antar orang, organisasi, sistem

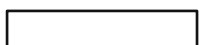
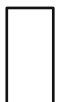
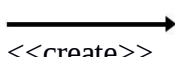
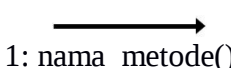
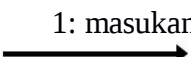
3. Sequence Diagram

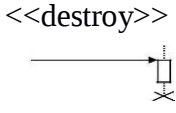
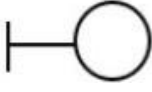
Sequence Diagram adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan. diagram ini terdiri dari dua dimensi, yaitu dimensi *vertikal* yang

menunjukkan waktu dan dimensi *horizontal* yang menunjukkan objek-objek. Tiap-tiap objek, termasuk *actor*, memiliki waktu aktif yang digambarkan dengan kolom *vertikal* yang disebut dengan *lifeline*. Sementara itu, pesan atau perintah digambarkan sebagai garis panah dari satu *lifeline* ke *lifeline* yang lain [14].

Berikut simbol-simbol dari *sequence diagram*:

Tabel 2.3. Simbol Sequence

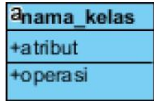
No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang , proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek
3		<i>Object</i>	Menyatakan objek yang berinteraksi pesan
4		<i>Timelife</i>	Menyatakan objek dalam berinteraksi pesan
5		Pesan tipe <i>Create</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi
6		Pesan tipe <i>Call</i>	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri
7		Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lain
8		Pesan tipe <i>Return</i>	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode

No.	Gambar	Nama	Keterangan
			menghasilkan suatu kembalian ke objek tertentu
9		Pesan tipe <i>Destroy</i>	Menyatakan suatu objek telah menjalankan suatu operasi
10		Boundary Class	Berupaan tepi dari sistem, seperti <i>user interface</i> atau alat yang berinteraksi dengan sistem lain
11		<i>Control Class</i>	Mengatur aliran dari informasi untuk sebuah skenario

4. Class Diagram

Class Diagram adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Desain model dari diagram kelas ini sendiri dibagi menjadi dua bagian. Bagian pertama merupakan penjabaran dari *database*. Bagian kedua merupakan bagian dari modul MVC, yang memiliki *class interface*, *class control*, dan *class entity*. [13]. Berikut simbol-simbol dari *class diagram* :

Tabel 2.4. Simbol Class

No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem
2	 Nama_interface	Antarmuka/ <i>Interface</i>	Sama dengan konsepn interface dalam

No.	Gambar	Nama	Keterangan
			pemrograman berorientasi objek
3		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum
4		Asosiasi berarah/ <i>Directed Association</i>	Relasi antar kelas dengan makna kelas
5		Generalisasi	Relasi antar kelas dengan makna generalisasi
6		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna whole-part