

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian Aplikasi penyewaan jadwal lapangan futsal ini digunakan untuk memudahkan penyewa mengetahui jadwal lapangan futsal yang kosong, untuk memberikan kemudahan dalam melakukan proses penyewaan jadwal lapangan futsal tanpa harus datang langsung ke tempat futsal. Pembuatan aplikasi ini dilakukan dengan metode *Waterfall*. Aplikasi ini di implementasikan dengan bahasa pemograman PHP, database MySQL sebagai media penyimpanan data dan metode pengujian menggunakan *black box testing*. Tujuan dari penelitian ini adalah menghasilkan sistem informasi penyewaan melalui *website*. Aplikasi ini dapat digunakan oleh 3 pengguna, yaitu admin, petugas dan penyewa. Admin dapat melakukan olah data lapangan, harga sewa, booking, konfirmasi pembayaran dan halaman statis. Penyewa dapat melakukan booking via web, melihat informasi jadwal lapangan dan konfirmasi pembayaran.

2.2 Landasan Teori

2.2.1. Waterfall

Metode penelitian *waterfall* adalah konsep pengembangan yang menekankan pada langkah sistematis. Sehingga, proses penciptaan sebuah sistem harus dilakukan secara berurutan, mulai dari tahapan

identifikasi kebutuhan sampai ke proses perawatan.[3]

2.2.2. Reservasi

Reservasi adalah aktivitas yang dilakukan oleh konsumen sebelum melakukan pembelian. Untuk mencapai kepuasan konsumen, perusahaan perlu memiliki sistem pemesanan yang baik. Menurut Kamus Besar Bahasa Indonesia (KBBI), reservasi diartikan sebagai proses, tindakan, atau cara memesan sesuatu (tempat, barang, dan sebagainya) kepada orang lain.[4].

2.2.3. Website

Website , yang juga dikenal sebagai situs, situs web, atau portal, adalah kumpulan halaman web yang saling terhubung. Halaman utama dari sebuah *website* disebut *Home Page*, sedangkan halaman – halaman lainnya disebut *web page*. [5].

2.2.4. Laravel



Gambar 2. 1 Logo Laravel

Laravel adalah framework PHP berlisensi MIT yang mengadopsi konsep *Model-View-Controller* (MVC). *Framework* ini dirancang untuk pengembangan aplikasi berbasis MVP menggunakan PHP, dengan tujuan meningkatkan kualitas perangkat lunak. Laravel

membantu mengurangi biaya awal pengembangan dan pemeliharaan, sekaligus memberikan pengalaman kerja yang lebih efisien melalui sintaks yang jelas, ekspresif, dan hemat waktu.[6]

2.2.5. Visual Studio Code



Gambar 2. 2 Logo Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah *editor teks* yang kuat dan fleksibel, yang dikembangkan oleh Microsoft. Selain dapat digunakan pada sistem operasi Windows, VS Code juga mendukung Linux dan MacOS.[7]

2.2.6. Bootstrap



Gambar 2. 3 Logo Bootstrap

HTML, CSS, dan JavaScript yang dirancang untuk mempercepat proses pengembangan *web responsive* dan *mobile-first* (memprioritaskan perangkat seluler).

Selain bisa digunakan untuk mengembangkan *website* dengan

lebih cepat, Bootstrap juga merupakan *framework* gratis yang bersifat *open-source*. *Skrip* dan *syntax* yang disediakan Bootstrap bisa diterapkan untuk berbagai komponen dalam desain web.[8]

2.2.7. HTML



Gambar 2. 4 Logo HTML

HTML, singkatan dari *Hypertext Markup Language*, adalah bahasa *markup* yang digunakan untuk membuat struktur halaman sebuah website HTML berfungsi untuk menyusun elemen-elemen di halaman web agar saling terhubung dan dapat diakses melalui internet, menciptakan tampilan yang menarik dan fungsional.[9]

2.2.8. JavaScript



Gambar 2. 5 Logo Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat *website* menjadi lebih interaktif dan dinamis.[10]

2.2.9. Cascading Style Sheet (CSS)



Gambar 2. 6 Logo CSS

CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu website, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk menformat halaman web yang ditulis dengan HTML atau XHTML.

Ada dua cara yang bisa diterapkan untuk menggunakan CSS pada web. Cara yang pertama adalah dengan membuat CSS langsung didalam satu file HTML kita (*internal / inline style sheet*). Cara yang kedua adalah dengan cara memanggil CSS tersebut dari file CSS tersendiri (*external style sheet*).[11]

2.2.10. XAMPP



Gambar 2. 7 Logo XAMPP

XAMPP adalah software gratis dan *open source* yang dapat

diinstal di berbagai platform, seperti Windows, Linux dan OS X. Pasalnya, di dalam aplikasi XAMPP terdapat puluhan paket modul, bahasa pemrograman dan komponen lainnya. [12]

2.2.11. MySQL



Gambar 2. 8 Logo MySQL

MySQL adalah salah satu jenis *database server* yang sangat populer. Kepopulerannya disebabkan oleh penggunaan SQL sebagai bahasa utama untuk mengakses *database*. MySQL bersifat *open-source* dan dilengkapi dengan *source code* (kode yang digunakan untuk mengembangkan MySQL)[13].

2.2.12. Database

Database adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap database mempunyai *API* tertentu untuk membuat, mengakses, mengatur, mencari, menyalin data yang ada di dalamnya. *Database* yaitu kumpulan *file - file* yang berhubungan satu dengan yang lainnya, diatur sedemikian rupa sehingga dapat digunakan oleh beberapa program aplikasi database[14].

2.2.13. UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah metode dalam

rekayasa perangkat lunak yang digunakan untuk menggambarkan alur, cara kerja, fungsi, tujuan, dan mekanisme kontrol suatu sistem.

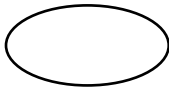
Dalam perancangan sistem informasi, terdapat empat model UML yang sangat efektif untuk menggambarkan desain sistem, yaitu: *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. [15]

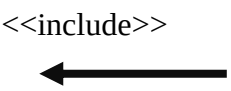
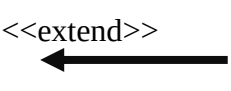
Dalam perancangan berbasis objek menggunakan UML, digunakan beberapa alat bantu sebagai berikut :

1 Use Case Diagram

Use case diagram adalah sebuah pemodelan yang menggambarkan perilaku (*behavior*) dari sistem informasi yang akan dikembangkan. *Use case* ini digunakan untuk mengidentifikasi fungsi - fungsi dalam sistem serta pihak-pihak yang memiliki hak akses terhadap fungsi - fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada table berikut:

Tabel 2. 1 Simbol Use Case Diagram

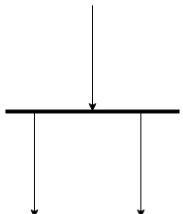
No	Simbol	Keterangan
1		<i>Use Case</i> : Representasi abstrak dari interaksi antara sistem dan aktor.
2		<i>Actor</i> : Merupakan peran yang dimainkan oleh individu, sistem lain, atau perangkat dalam berkomunikasi dengan <i>use case</i> .
3		<i>Association</i> : Representasi abstrak dari hubungan antara aktor dan <i>use case</i> .

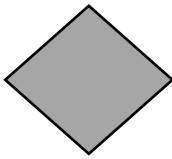
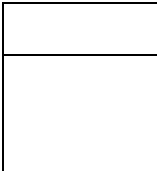
No	Simbol	Keterangan
4		Generalisasi: Menunjukkan spesialisasi aktor yang dapat berpartisipasi dalam <i>use case</i> .
5		Menunjukkan bahwa sebuah <i>use case</i> sepenuhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6		Menunjukkan bahwa sebuah <i>use case</i> merupakan penambahan fungsional dari <i>use case</i> lain jika kondisi tertentu terpenuhi.

2 Diagram Aktivitas (Activity Diagram).

Activity Diagram menggambarkan alur kerja atau aktivitas dalam suatu sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel berikut :

Tabel 2. 2 Simbol *Diagram Aktivitas (Activity Diagram)*

No	Simbol	Keterangan
1		<i>Start Point</i> , ditempatkan di pojok kiri atas dan menandakan awal dari aktivitas.
2		<i>End Point</i> , menandakan akhir dari aktivitas.
3		<i>Activities</i> , menggambarkan suatu proses atau kegiatan bisnis.
4		<i>Fork</i> / percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

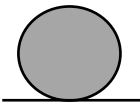
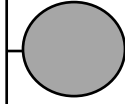
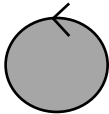
No	Simbol	Keterangan
5		<i>Decision Points</i> , menggambarkan titik keputusan dengan pilihan <i>true</i> atau <i>false</i> .
6		<i>Swimlane</i> , membagi <i>activity diagram</i> untuk menunjukkan siapa yang melakukan masing-masing aktivitas.

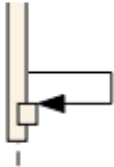
3 Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan perilaku objek dalam suatu use case dengan mendiskripsikan durasi hidup objek serta pesan yang dikirim dan diterima antar objek.

Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada tabel berikut:

Tabel 2. 3 Simbol Diagram Urutan (*Sequence Diagram*)

No	Simbol	Keterangan
1		<i>Entity Class</i> , merupakan bagian dari sistem yang berisi sekumpulan kelas entitas yang membentuk gambaran awal sistem dan menjadi dasar dalam menyusun basis data.
2		<i>Boundary Class</i> , berisi kumpulan kelas yang berfungsi sebagai antarmuka atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form input</i> dan <i>form cetak</i> .
3		<i>Control Class</i> , objek yang berisi logika aplikasi yang tidak bertanggung jawab terhadap entitas, contohnya adalah perhitungan dan aturan bisnis yang melibatkan berbagai objek.

No	Simbol	Keterangan
4		<i>Message</i> , simbol yang menunjukkan pengiriman pesan antar kelas.
5		<i>Recursive</i> , menggambarkan pengiriman pesan yang dilakukan objek kepada dirinya sendiri.
6		<i>Activation</i> , menggambarkan eksekusi operasi dari objek, dengan panjang kotak yang menunjukkan durasi aktivasi operasi tersebut.
7		<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, di sepanjang <i>lifeline</i> terdapat <i>activation</i> .

4 Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas berserta penjelasan mendetail mengenai setiap kelas dalam model desain sistem, yang juga menggambarkan aturan serta tanggung jawab entitas yang mempengaruhi perilaku sistem.

Class Diagram menunjukkan atribut dan operasi dari setiap kelas, serta constraint yang berhubungan dengan objek-objek yang saling terhubung.

Tabel 2. 4 Simbol *Diagram Kelas (Class Diagram)*

<i>Multiplicity</i>	<i>Keterangan</i>
1	Hanya satu dan tidak lebih.
0..*	Bisa tidak ada, atau satu, atau lebih.
1..*	Harus satu atau lebih.
0..1	Bisa tidak ada, dengan maksimal satu.
n..n	Batasan antara, misalnya 2..4 berarti minimal 2 dan maksimal 4.