

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada Penelitian (Maimunah, Hariyansyah, 2017) **Rancangan Aplikasi Penyewaan Lapangan Futsal Berbasis Web** yang dibangun dengan menggunakan bahasa pemrograman PHP (*Personal Home Page*) dan untuk databasenya menggunakan Mysql dapat memudahkan *customer* dalam memesan lapangan futsal tanpa batasan jarak dan waktu, *customer* hanya datang ketempat penyewaan pada saat jam dan waktu yang sudah ditentukan pada saat pemesanan. Permasalahannya ialah penyewa masih harus datang ke tempat futsalnya untuk *membooking* lapangan futsal serta penginputan data – data penyewa yang masih menggunakan catatan ke buku. Sehingga tidak ada informasi untuk pelanggan secara *real-time* yang akan menjadi kendala untuk si penyewa. Hal – hal yang sering terjadi akibat permasalahan tersebut yaitu penyewa terkadang tidak jadi *membooking* lapangan futsal dikarenakan tidak adanya informasi yang tepat saat lapangan sudah penuh terpakai dan masih sulitnya pendataan keuangan karena masih dalam bentuk pencatatan manual. Pengertian. Pada penelitian (Mamarimbing, 2016) **Penerapan Metode Waterfall Dalam Aplikasi Penyewaan Lapangan Futsal Akasia Berbasis Web** Menerapkan desain database serta menggunakan *software* PHP dan pemrograman, desain antar muka kedalam Bahasa dengan Xampp dalam pembuatan *website*. Permasalahannya ialah banyaknya peminat dalam berolahraga futsal menjadikan tempat penyewaan lapangan futsal sangat ramai, bahkan ketika saat ingin bermain dan memesan dengan datang langsung, terjadi antrean dalam penyewaan lapangan futsal. Proses manual yang terjadi membutuhkan banyak waktu. Dengan adanya *website* penyewaan lapangan futsal menjadikan penyewa dengan mudah melihat jadwal yang kosong dan dapat memesan lapangan tersebut susai dengan jadwal yang didapat, lebih efisiensi waktu dan tenaga. Pada penelitian (Merdekawati, 2019) **Sistem Informasi Penyewaan Lapangan Futsal Berbasis Web Pada Futsal**

Station Bekasi menggunakan bahasa pemrograman PHP dan *database* yang digunakan adalah MYSQL dalam perancangan *web* sistem informasi penyewaan lapangan futsal. Dan untuk perangkat lunak yang dibutuhkan yaitu, XAMPP, *Adobe Dreamweaver*. Permasalahannya ialah Futsal Station ini dalam proses pelayanan penyewaan lapangan futsal harus datang ketempat futsalnya untuk *membooking* lapangan futsal, pada penginputan data-data penyewa masih menggunakan kertas catatan. Bukti pembayaran saat ini masih menggunakan banyak kertas untuk membuat laporan pembayaran, sehingga jika ingin merekap data, pengelola kesulitan karena ada banyak kertas yang harus dikumpulkan.

Tabel 2.1 Tabel Penelitian Terdahulu

No	Penulis	Tahun	Judul	Permasalahan
1.	Maimunah, Hariyansyah	2017	Rancangan Aplikasi Penyewaan Lapangan Futsal Berbasis <i>Web</i>	penyewa masih harus datang ke tempat futsalnya untuk <i>membooking</i> lapangan futsal serta penginputan data – data penyewa yang masih menggunakan catatan ke buku. Sehingga tidak ada informasi untuk pelanggan secara <i>real-time</i> yang akan menjadi kendala untuk si penyewa.
2.	Mamarimbing	2016	Penerapan Metode <i>Waterfall</i> Dalam Aplikasi Penyewaan Lapangan Futsal Akasia Berbasis <i>Web</i>	banyaknya peminat dalam berolahraga futsal menjadikan tempat penyewaan lapangan futsal sangat ramai, bahkan ketika saat ingin bermain dan memesan dengan datang langsung, terjadi antrean dalam penyewaan lapangan futsal. Proses manual yang terjadi membutuhkan banyak waktu.
3.	Merdekawati	2019	Sistem Informasi Penyewaan Lapangan Futsal Berbasis <i>Web</i> Pada Futsal Station Bekasi	Futsal Station ini dalam proses pelayanan penyewaan lapangan futsal harus datang ketempat futsalnya untuk

				<p>membooking lapangan futsal, pada penginputan data-data penyewa masih menggunakan kertas catatan. Bukti pembayaran saat ini masih menggunakan banyak kertas untuk membuat laporan pembayaran, sehingga jika ingin merekap data, pengelola kesulitan karena ada banyak kertas yang harus dikumpulkan.</p>
--	--	--	--	--

Penelitian yang sudah dilakukan sebelumnya masih banyak permasalahan sehingga pada penelitian ini memperbaiki permasalahan penelitian sebelumnya yaitu dengan membantu pihak pengelola lapangan dalam mengolah data penyewaan dan penyewa dapat memilih lapangan sesuai yang dibutuhkan. Di penelitian ini sudah ada fitur sistem otomatis dimana jika penyewa telat membayar maka pemesanan lapangan secara otomatis akan batal yang pastinya sudah terdapat fitur notifikasi sehingga pengelola lapangan tidak perlu *realtime* untuk mengecek *web* setiap saat. Dipenelitian ini menggunakan metode *Framework CodeIgniter*.

2.2 Konsep Dasar Aplikasi

Pada bagian ini akan dijelaskan definisi dasar dari istilah yang terkait dengan pembuatan aplikasi berbasis *website* penyewaan lapangan olahraga dengan *framework codeigniter*.

2.2.1 Pengertian Olahraga

Menurut (Putra, 2019) Suatu olahraga adalah bentuk fisik yang terencana dan terstruktur yang melibatkan gerakan tubuh berulang-ulang dan ditunjukan untuk meningkatkan kebugaran jasmani.

Selain itu, olahraga juga dapat dilakukan untuk sekedar hobi, mencari keringat, mengisi waktu luang bahkan olahraga juga dapat membantu menurunkan berat badan, dan ada juga yang melakukannya untuk karir.

Selain untuk tujuan tertentu, olahraga memiliki banyak manfaat, dalam hal Kesehatan olahraga secara rutin dapat membantu menjauhkan dari beragam penyakit. Hal ini terjadi dikarenakan daya tahan tubuh meningkat. Olahraga secara teratur dan dapat menjadi aktivitas rutin dapat membantu mengurangi stres.

2.2.2 Pengertian Sewa

Menurut (Suhaimi Alfaris, 2020) Sewa-menyewa adalah suatu perjanjian dengan mana pihak yang satu mengikatkan dirinya untuk memberikan kepada pihak yang lainnya kenikmatan dari sesuatu barang, selama suatu waktu tertentu dan dengan pembayaran suatu harga yang oleh pihak yang tersebut terakhir itu disanggupi pembayarannya. Demikianlah definisi yang diberikan oleh Pasal 1548 KUHPdt, mengenai perjanjian sewa-menyewa. Sewa-menyewa, seperti halnya dengan jual-beli dan perjanjian-perjanjian lain pada umumnya, adalah suatu perjanjian konsensual. Artinya, sudah sah dan mengikat pada detik tercapainya sepakat mengenai unsur 2 pokoknya, yaitu barang dan harga.

2.2.3 Aplikasi WEB

Menurut (Arfianto & Nugrahanti, 2018) *Web* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan *protocol* HTTP (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser.

2.2.4 Penjelasan Protokol Kesehatan

Menurut (Mardiyah, 2019) Protokol kesehatan dibentuk tujuannya agar masyarakat tetap dapat beraktivitas secara aman dan tidak membahayakan keamanan atau Kesehatan orang lain. Jika masyarakat dapat mengikuti dengan baik aturan protokol Kesehatan, maka penularan COVID-19 dapat diminimalisir. Jika sedang berpergian masyarakat dihimbau agar memakai masker, dan jika kalau tidak terlalu penting masyarakat tidak dianjurkan berpergian. Karena angka kasus COVID-19 hingga saat ini terus mengalami peningkatan yang cukup tinggi di Indonesia.

Menurut (Putsanra, 2019) *new normal* adalah skenario untuk mempercepat penanganan COVID-19 dalam aspek kesehatan dan sosial ekonomi. Pemerintah Indonesia mengumumkan rencana untuk mengimplementasikan *scenario new normal* dengan mempertimbangkan studi epidemiologis dan kesiapan regional.

2.3 Tools Pemodelan yang Digunakan

Pada proyek akhir ini *tools* yang digunakan adalah UML, *Use Case Diagram*, *Entity-Relationship Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Activity Diagram*.

2.3.1 UML

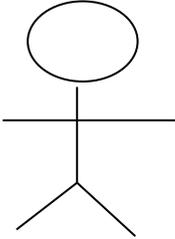
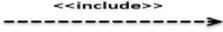
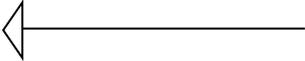
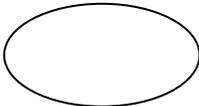
Menurut (Syarif & Nugraha, 2020) UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”. Beberapa pemodelan yang termasuk kedalam pemodelan UML seperti *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*.

2.3.1.1 Use Case Diagram

Menurut (Heriyanto, 2018) “Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.” Tabel 2.3 merupakan tabel simbol-simbol *use case diagram*.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.2 Simbol *Use Case Diagram*

No.	Nama Simbol	Simbol	Keterangan
1.	<i>Actor</i>		Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.	<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.	<i>Generalization</i>		Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4.	<i>Include</i>		Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.	<i>Extend</i>		Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.	<i>Association</i>		Apa yang menghubungkan antara objek satu
7.	Sistem		Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.	<i>Use Case</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

No.	Nama Simbol	Simbol	Keterangan
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

2.3.1.2 Class Diagram

Menurut (Hendini, 2016) *Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, *attribute (Attributes)*, operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel 2.3 Multiplicity Class Diagram

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

2.3.1.3 Sequence Diagram

Menurut (Setiawan, 2018) Diagram Aktifitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari suatu sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

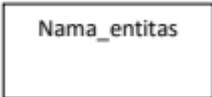
2.3.1.4 Activity Diagram

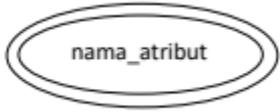
Menurut (Setiawan, 2018) Diagram Sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek.

2.3.2 ERD

Menurut (Minda Septiani, Nurul Afni, 2019) Pemodelan awal basis data yang paling banyak digunakan adalah ERD yang dikembangkan berdasarkan teori himpunan dalam bidang matematika.

Tabel 2.4 Simbol ERD

Simbol	Deskripsi
Entitas 	Entitas merupakan data inti yang akan disimpan; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
Atribut 	Kolom data yang butuh disimpan dalam suatu entitas
Atribut Kunci Primer 	Kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan.
Atribut multivalai/multivalue	Kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

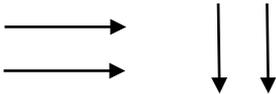
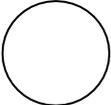
Simbol	Deskripsi
	
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi/Association 	Penghubung antara relasi dan entitas di mana dikedua ujungnya memiliki <i>multiplicity</i>

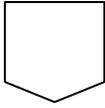
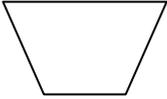
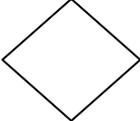
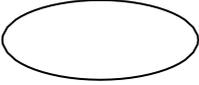
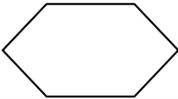
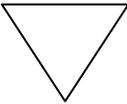
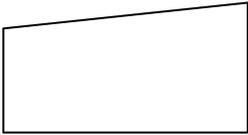
2.3.3 Flowchart

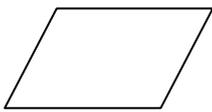
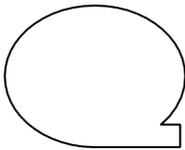
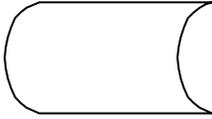
Menurut (Hidayat, 2020) *Flowchart* atau diagram alir merupakan representasi grafik dari langkah- langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri dari sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. *Flowchart* diawali dengan menerima *input*, pemrosesan *input*, dan diakhiri dengan menampilkan *output*.

Dalam penulisan *Flowchart* dikenal dua model, yaitu *system flowchart* dan *program flowchart*. *System Flowchart* adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa *file* di dalam media tertentu, sedangkan *program flowchart* adalah bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program.

Tabel 2.5 Simbol-Simbol Flowchart

NO	SIMBOL	KETERANGAN
1		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses.
2		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.

3		Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
4		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
5		Simbol <i>manual</i> , berfungsi untuk menyatakan suatu tindakan(proses) yang tidak dilakukan oleh komputer.
6		Simbol <i>decision</i> , berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak.
7		Simbol <i>terminal</i> , berfungsi untuk menyatakan permulaan atau akhir suatu program.
8		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberiharga awal.
9		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesinyang mempunyai <i>keyboard</i> .
10		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
11		Simbol <i>manual input</i> , berfungsi untuk Memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

12		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
13		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis.
14		Simbol <i>disk storage</i> , berfungsi untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
15		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui printer).
16		Simbol <i>punched card</i> , berfungsi untuk menyatakan input berasal dari kartu atau output ditulis ke kartu.

2.4 Tools Pembangunan Aplikasi

Berikut merupakan definisi dan penjelasan *tools* yang dibutuhkan dalam pembuatan dan pengembangan aplikasi.

2.4.1 PHP

Menurut (Sitohang, 2019) PHP adalah PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *web* berbasis *server* (*server side*) yang mampu memarsing kode PHP dari kode dengan ekstensi PHP sehingga menghasilkan tampilan *website* yang dinamis disisi *client*.

2.4.1.1 Codeigniter

Menurut (Subagia, 2018) *CodeIgniter* merupakan sebuah *web framework* yang dikembangkan oleh Rick Ellis dari Ellis Lab. *CodeIgniter* dirancang untuk

menjadi sebuah *web framework* yang ringan dan mudah digunakan. Bahkan dari Rasmus Lerdorf, sang pencipta bahasa pemrograman PHP mengatakan bahwa *Codeigniter* merupakan *web framework* yang mudah, cepat, dan handal. Dirilis pada dokumentasi *Codeignite*, *Codeigniter* merupakan *toolkit* bagi orang yang ingin membangun aplikasi *web* menggunakan PHP.

Berikut adalah konsep *Model-View-Controller* yang diterapkan pada *Codeigniter*:

1. *Model*, menggambarkan struktur data. Biasanya kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah dan memperbaharui informasi yang ada di *database*.
2. *View*, informasi yang diperlihatkan kepada user. *View* adalah halaman *web* yang terdiri dari HTML, CSS dan *Javascript*, akan tetapi pada *CodeIgniter*, *view* dapat juga sebagai potongan halaman seperti *header* atau *footer*. Bahkan dapat juga halaman RSS atau tipe halaman lainnya.
3. *Controller*, perantara *Model*, *View*, dan *resource* lainnya yang dibutuhkan untuk menangani HTTP *request* dan menghasilkan halaman *web*.

2.4.1.2 HTML

Menurut (Enterprise, 2018) HTML, digunakan untuk membuat struktur halaman *website*. Bisa dibayangkan secara umum bahwa HTML digunakan untuk mendesain *website*, meskipun dalam praktiknya HTML tidak berdiri sendiri sebab akan digabungkan dengan CSS atau *Script* lain, seperti *Javascript*. Pada kesempatan ini, kita akan menganggap HTML sebagai program untuk membuat desain sebuah *website*, dimana kita bisa menulis teks, memasukkan gambar, membuat *form*, dan sebagainya.

2.4.2 MySQL

Menurut (Enterprise, 2018) MySQL merupakan *server* yang melayani *database*. Untuk membuat dan mengelola *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL. *Database* sendiri dibutuhkan jika kita ingin *input* data dari *user* menggunakan *form* HTML untuk kemudian diolah PHP agar bisa disimpan kedalam *database* MySQL.

2.4.3 XAMPP

Menurut (Enterprise, 2018) XAMPP merupakan *server* yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi *programmer* pemula. Selain gratis, fiturnya tergolong lengkap dan gampang digunakan oleh *programmer* PHP tingkat awal, yang perlu dilakukan hanyalah menjalankan *module Apache* yang ada di dalam XAMPP tersebut.

2.4.4 Visual Studio Code

Menurut (Salamah, 2021) *Visual Studio Code* adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *TypeScript*, dan *Node.js*, dan serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via *marketplace*, *Visual Studio Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java*, dst).

2.5 Pengujian Aplikasi

Berikut merupakan beberapa jenis *tools* yang digunakan untuk pengujian aplikasi.

2.5.1 Black Box Testing

Menurut (Cholifah et al., 2018) Metode *Black box Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, Estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid.

