

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian yang dilakukan oleh Muhammad Wildan Rizkia dan Asti Herliana dalam jurnal *Aplikasi Website Reservasi Meja Biliar di Shooters Pool & Cafe* menjelaskan pengembangan sistem reservasi meja billiard berbasis *website* yang bertujuan memudahkan pelanggan melakukan pemesanan secara *online*. Sistem yang dikembangkan menyediakan fitur registrasi, *login*, reservasi meja, serta pengelolaan data oleh admin sehingga proses pemesanan menjadi lebih efektif dibandingkan dengan sistem manual. Penelitian tersebut berfokus pada digitalisasi proses reservasi meja dan pengelolaan data pelanggan, namun belum membahas integrasi layanan pendukung seperti pemesanan makanan dan minuman, pembayaran uang muka (DP) secara *online*, maupun penyajian *dashboard* monitoring operasional secara *real-time*. Oleh karena itu, penelitian ini mengembangkan sistem yang lebih lengkap dengan menambahkan fitur pemesanan *f&b*, pembayaran melalui *payment gateway*, serta monitoring status meja secara *real-time* untuk meningkatkan kualitas pelayanan [5].

Pada penelitian lain yang dilakukan oleh Muhammad Saif Akbar, Ade Novia Maulana, dan Hery Afriadi dengan judul *Perancangan Sistem Informasi Booking Online Meja Billiard Berbasis Web di Spot Billiard* mengembangkan sistem *booking* menggunakan metode *Waterfall*, *framework Laravel*, *database MySQL*, dan pengujian *Black Box Testing*. Hasil penelitian

menunjukkan bahwa sistem mampu membantu pelanggan melakukan reservasi secara mandiri serta mempermudah pengelola dalam mengelola transaksi dan laporan penyewaan. Meskipun demikian, ruang lingkup penelitian tersebut masih berfokus pada proses *booking* dan pencatatan transaksi sehingga belum menyediakan fitur manajemen menu *f&b* yang terintegrasi, pembayaran DP secara *online*, serta laporan pendapatan berdasarkan periode tertentu. Berdasarkan kesenjangan tersebut, penelitian ini mengembangkan sistem informasi reservasi yang tidak hanya mengelola pemesanan meja, tetapi juga menyediakan fitur pemesanan menu, pembayaran *online*, laporan pendapatan berdasarkan tanggal, serta *dashboard* yang menampilkan status meja secara *real-time* [6].

Penelitian lain yang dilakukan oleh Jeremia Joshua Tamedo, I Gkg Puritan Wijaya Adh, dan I Gede Nika Wirawan dengan judul *Sistem Reservasi Billiard Online Berbasis Website Pada Koufu Billiard & Cafe* berfokus pada pengembangan sistem reservasi yang mampu meningkatkan efisiensi pelayanan melalui penyediaan informasi ketersediaan meja secara daring dan pengelolaan data reservasi yang lebih terstruktur. Penelitian tersebut menunjukkan bahwa penerapan sistem berbasis *website* dapat mengurangi kesalahan pencatatan, mempercepat proses pelayanan, serta memudahkan admin dalam melakukan pengelolaan data pelanggan dan jadwal reservasi. Namun, penelitian tersebut lebih menitikberatkan pada peningkatan efektivitas proses reservasi dan administrasi, sehingga belum membahas pemanfaatan data transaksi sebagai laporan analisis pendapatan

maupun penyediaan informasi operasional yang dapat digunakan sebagai bahan evaluasi bagi pihak manajemen. Oleh karena itu, penelitian ini tidak hanya mengembangkan sistem reservasi berbasis *website*, tetapi juga menyediakan fitur laporan pendapatan berdasarkan periode, monitoring status meja secara *real-time*, serta pengelolaan transaksi yang terintegrasi untuk mendukung sistem *website* [7].

2.2 Landasan Teori

Adapun landasan teori pada “Sistem informasi reservasi meja billiard jay’s berbasis *website* menggunakan *framework laravel*” berpacuan dengan berbagai penelitian terdahulu di antaranya:

2.2.1 Sistem Informasi

Sistem informasi adalah sekumpulan komponen yang saling berhubungan yang mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam organisasi atau tempat usaha, sistem informasi berperan penting dalam membantu pengelolaan data menjadi informasi yang bermanfaat. Informasi tersebut dapat digunakan untuk mendukung kegiatan operasional, mempermudah proses pengambilan keputusan, meningkatkan efisiensi kerja, serta membantu memberikan pelayanan yang lebih baik kepada pelanggan. Dengan adanya sistem informasi, proses yang sebelumnya dilakukan

secara manual dapat menjadi lebih cepat, terorganisir, dan mengurangi risiko terjadinya kesalahan dalam pengolahan data [8].

2.2.2 Reservasi

Reservasi adalah proses pemesanan tempat atau layanan yang dilakukan sebelum waktu penggunaan untuk memastikan ketersediaan. Dengan reservasi, pelanggan dapat memperoleh layanan tanpa harus menunggu lama atau kehabisan tempat, sementara penyedia layanan dapat mengatur jadwal dengan lebih teratur dan efisien, reservasi membantu dalam mengatur jadwal penggunaan, mengelola kapasitas, serta mempermudah proses operasional agar berjalan lebih efektif dan efisien. Selain itu, sistem reservasi juga dapat mengurangi terjadinya kesalahan pencatatan, menghindari bentrokan jadwal, serta meningkatkan kualitas pelayanan kepada pelanggan. Oleh karena itu, penerapan sistem reservasi, khususnya yang berbasis *website*, menjadi solusi yang memudahkan pelanggan dalam melakukan pemesanan kapan saja dan di mana saja, sekaligus membantu pengelola dalam mengelola data reservasi secara lebih cepat, akurat, dan terstruktur [9].

2.2.3 Website

Website, yang juga dikenal sebagai situs atau portal, merupakan sekumpulan halaman *web* yang saling terhubung satu sama lain. Halaman utamanya disebut *home page*, sedangkan halaman lain yang berisi informasi tambahan disebut sebagai *web page*, *website*

digunakan sebagai media untuk menyampaikan informasi, menyediakan layanan, maupun mendukung berbagai aktivitas secara *online*. Dalam perkembangannya, *website* tidak hanya berfungsi sebagai media penyampaian informasi, tetapi juga dapat digunakan sebagai sarana transaksi, komunikasi, dan pengelolaan data melalui sistem informasi berbasis *web*. Dengan adanya *website*, pengguna dapat mengakses informasi dan menggunakan berbagai layanan kapan saja dan di mana saja selama terhubung dengan internet. [10].

2.2.4 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. Fitur utama yang dimiliki oleh *payment gateway* meliputi enkripsi data pembayaran dan data pribadi pengguna untuk menjaga keamanan transaksi, komunikasi secara otomatis antara pelanggan, penyedia layanan, dan lembaga keuangan, serta proses otorisasi pembayaran untuk memastikan bahwa transaksi telah berhasil dilakukan. Selain itu, *payment gateway* juga menyediakan proses verifikasi transaksi secara *real-time* sehingga pembayaran dapat dikonfirmasi dengan cepat, akurat, dan aman. [11].

2.2.5 Laravel

Laravel adalah *framework PHP* yang menggunakan konsep MVC (*Model View Controller*) dan dirilis dengan lisensi *MIT*. *Framework* ini digunakan untuk pengembangan aplikasi web dan dirancang agar proses pengembangan menjadi lebih efisien. Dengan *Laravel*, developer bisa mengurangi biaya dan waktu pengembangan karena banyak fitur yang sudah disediakan, sehingga aplikasi yang dihasilkan tetap berkualitas dan mudah untuk dipelihara [12].



Gambar 2. 1 Logo Laragon

2.2.6 Visual Studio Code

Visual Studio Code (VS Code) adalah teks editor yang ringan dan handal, dibuat oleh *Microsoft*, dan bisa dijalankan di berbagai sistem operasi seperti *Windows*, *Linux*, dan *Mac*. Editor ini secara langsung mendukung bahasa pemrograman seperti *JavaScript*, *TypeScript*, dan *Node.js*. Selain itu, *VS Code* juga bisa dipakai untuk bahasa lain seperti *C++*, *C#*, *Python*, *Go*, atau *Java* dengan memasang *plugin* dari marketplace-nya [13].



Gambar 2. 2 Logo *Visual Studio Code*

2.2.7 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* [14].



Gambar 2. 3 Logo *Bootstrap*

2.2.8 HTML

HTML (Hypertext Markup Language) merupakan salah satu format yang digunakan dalam pembuatan dokumen maupun aplikasi yang dijalankan melalui halaman *web*. Dengan kata lain, *HTML*

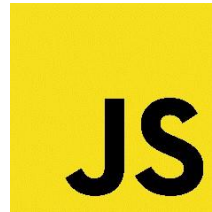
menjadi dasar dari setiap tampilan atau struktur yang muncul pada sebuah *website*. Oleh karena itu, sebelum seseorang dapat membuat sebuah program atau aplikasi yang berjalan di atas halaman *web*, penting untuk terlebih dahulu memahami dan menguasai *HTML*. Pemahaman *HTML* akan memudahkan dalam mengatur tampilan, menyusun konten, dan membangun struktur sebuah halaman *web* secara benar dan terorganisir [15].



Gambar 2. 4 Logo *HTML*

2.2.9 Java Script

Javascript adalah bahasa pemrograman yang digunakan untuk membuat aplikasi dan *website* interaktif di sisi klien (*client-side*). *JavaScript* memungkinkan developer *web* untuk membuat aplikasi *web* yang lebih interaktif, dinamis, dan responsif dengan menambahkan fungsionalitas seperti validasi form, animasi, interaksi pengguna, dan manipulasi DOM (*Document Object Model*) [16].



Gambar 2. 5 Logo Java Script

2.2.10 Cascading Style Sheet (CSS)

Cascading Style Sheets merupakan skrip yang digunakan untuk mengatur tampilan atau desain pada sebuah *website*. Meskipun *HTML* memiliki kemampuan dasar untuk mengatur tampilan, namun kemampuan tersebut masih terbatas. Oleh karena itu, *CSS* digunakan untuk memberikan pengaturan tampilan yang lebih lengkap sehingga struktur *website* yang dibangun dengan *HTML* dapat terlihat lebih rapi, teratur [17].



Gambar 2. 6 Logo CSS

2.2.11 Laragon

Laragon adalah *software* yang berfungsi untuk membuat *server* lokal di komputer, yang sangat membantu pengembang dalam mengelola proyek *web* mereka dengan lebih efisien. *Laragon* mendukung berbagai teknologi seperti *PHP*, *MySQL*, *Apache*, *Nginx* dan lainnya, serta menyediakan lingkungan pengembangan yang stabil, cepat dan mudah diakses [18].



Gambar 2. 7 Logo *Laragon*

2.2.12 MySQL

MySQL merupakan salah satu sistem basis data yang banyak digunakan dan mendukung penggunaan bahasa pemrograman *PHP*. *MySQL* memiliki *query* atau bahasa *SQL* (*Structured Query Language*) yang sederhana sehingga mudah dipahami. Selain itu, *MySQL* menggunakan karakter escape yang serupa dengan *PHP* dan dikenal sebagai salah satu basis data yang memiliki kinerja cukup cepat dalam pengolahan data [19].



Gambar 2. 8 Logo MySQL

2.2.13 PHP (*Hypertext Preprocessor*)

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* [20].



Gambar 2. 9 Logo PHP

2.2.14 Database

Database atau basis data bisa diartikan secara singkat sebagai tempat untuk menyimpan sekumpulan data secara sistematis yang

mana umumnya disimpan di dalam komputer. Suatu *database* memerlukan perangkat lunak untuk mengolah dan memanipulasi data seperti menambah, menghapus, megubah dan menampilkan data, perangkat lunak itu disebut *DBMS (Database Management System)*. suatu *database* juga dapat didefinisikan terdiri dari kumpulan tabel-tabel yang menyimpan data serta informasi [21]



Gambar 2. 10 Logo *Database*

2.2.15 UML (*Unifield Modeling Language*)

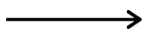
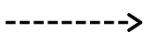
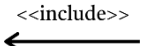
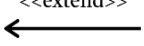
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek [22]. *UML* tersusun atas sejumlah elemen grafis membentuk 9 diagram-diagram. Dalam penelitian ini melakukan desain hanya 4 diagram yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*.

1. *Use Case Diagram*

Use Case Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan fungsi-fungsi utama (use case) dari suatu sistem dan interaksi antara pengguna (aktor) dengan sistem tersebut. Diagram ini menunjukkan apa saja yang bisa dilakukan oleh pengguna terhadap sistem, namun tidak menjelaskan bagaimana fungsi tersebut dijalankan secara teknis [23]. Berikut simbol-simbol *use case diagram* :

Tabel 2. 1 *Use Case Diagram*

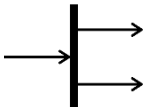
No	Gambar	Simbol	Keterangan
1.		Aktor	Mewakili peran seorang <i>system</i> yang lain, atau alat ketika berkomunikasi dengan <i>Use Case</i> .
2.		Use Case	Mewakiili Abstraksi dan interaksi antara sebuah <i>System</i> dan <i>Actor</i> .

No	Gambar	Simbol	Keterangan
3.		<i>Association</i>	Abstraksi penghubung antar <i>Actor</i> dengan <i>Use Case</i> .
4.		<i>Generalisasi</i>	Menunjukkan spesialisasi <i>Actor</i> yang terlibat <i>Use Case</i>
5.		<i>Include</i>	Menunjukan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya.
6.		<i>Extend</i>	Menunjukan bahwa suatu <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi.

2. Activity Diagram

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

Tabel 2. 2 Activity Diagram

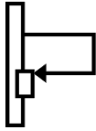
No	Gambar	Simbol	Keterangan
1.		<i>Initial node</i>	Merupakan awal aktivitas yang akan dibuat.
2.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi.
3.		<i>Decicion</i>	Menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
4.		<i>Fork Join node</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan
5.		<i>Fork node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran.
6.		<i>Activity Final node</i>	Merupakan akhir aktivitas yang telah dibuat.

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. Berikut simbol-simbol dari *sequence diagram* :

Tabel 2. 3 *Sequence Diagram*

No	Gambar	Simbol	Keterangan
1.		<i>Entity Class</i>	Menggambarkan awal dari sebuah sistem, menjadi dasar penyusunan basisdata.
2.		<i>Boundary Class</i>	Menggambarkan komunikasi antara sistem seperti <i>Form</i> .

No	Gambar	Simbol	Keterangan
3.		<i>Control Class</i>	Penghubung antara <i>boundary</i> dengan <i>table</i> .
4.		<i>Recursive</i>	Mengirimkan pesan yang akan dikirim ke dirinya sendiri.
5.		<i>Activation</i>	Menghubungkan proses durasi aktivasi sebuah proses.
6.		<i>Life Line</i>	Menghubungkan tiap-tiap objek yang mana terdapat <i>activation</i> .
7.		<i>Message</i>	Sebagai pengirim pesan ke tiap-tiap <i>Class</i> .

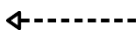
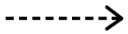
4. Class Diagram

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*. Berikut simbol-simbol dari *class diagram* :

Tabel 2. 4 Class Diagram

No	Gambar	Simbol	Keterangan
1.		<i>Class</i>	Menggambarkan kumpulan objek-objek dengan atribut dan operasi yang sama.
2.		<i>Nary Association</i>	Sebagai upaya menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Collaboration</i>	Urutan aksi-aksi yang ditampilkan oleh sistem yang menghasilkan suatu hasil yang tertukar bagi suatu <i>actor</i> .

No	Gambar	Simbol	Keterangan
4.		<i>Generalization</i>	Menggambarkan dimana hubungan objek <i>descendent</i> berperilaku dan struktur data objek yang di atasnya objek induk <i>ancestor</i> .
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang tidak mandiri.