

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini beberapa referensi terkait sistem pendaftaran pasien yang dapat digunakan sebagai acuan sebagai berikut :

Penelitian dilakukan oleh Rika Melyanti, Dedy Irfan, Ambiyar (2020) dalam “Rancang Bangun Sistem Antrian *Online* Kunjungan Pasien Rawat Jalan Pada Rumah Sakit Syafira Berbasis *Web*”. Hasil penelitian ini adalah membuat Sistem Antrian *Online* yang dapat mengatasi masalah seperti pengambilan nomor antrian, informasi jadwal dokter, dan laporan harian kunjungan pasien. Sistem antrian ini dirancang menggunakan bahasa pemrograman *PHP* dan basis data *MySQL*.

Penelitian dilakukan oleh Sri Lestari, Aldino Nur Ihsan, Andrian Nur Ihsan, Dionta, dan Julinar Sari Hutagalung (2023) dalam “Sistem Pendaftaran Pasien Secara *Online* di UPTD Puskesmas Seputih Banyak Berbasis *Web*”. Sistem ini dibuat menggunakan bahasa pemrograman *PHP* dan *Framework CodeIgniter* dengan basis data *MySQL*. Metode pengembangan sistem yang digunakan adalah *Waterfall*. Dengan adanya sistem ini, diharapkan proses pendaftaran dan pengambilan tiket antrian menjadi lebih mudah dan cepat, sehingga meningkatkan kenyamanan bagi pasien dalam mengakses layanan kesehatan di puskesmas tersebut .

Penelitian dilakukan oleh Muhammad Bagus Heramwan, Halimahtus Mukminna, Achmad Arif Alfin, dan Yudo Bismo Utomo (2023) dalam “Rancang

Bangun Aplikasi Pendaftaran Poli Berbasis *Web* Menggunakan *Framework Laravel* (Studi Kasus RSI Madinah Ngunut)”. Hasil aplikasi ini menggunakan *Framework Laravel* dan bertujuan untuk mempermudah proses pendaftaran pasien di poli. Fitur utama dari aplikasi ini meliputi pendaftaran pasien dan penomoran antrian secara otomatis. Implementasi aplikasi ini diharapkan dapat meningkatkan efisiensi dan akurasi dalam proses pendaftaran pasien serta mengurangi waktu tunggu pasien di rumah sakit. Sistem dibuat dengan bahasa pemrograman *PHP*.

Penelitian dilakukan oleh Chaeroen Niesa, Zakial Vikki, Agus Maulida (2023) dalam “Perancangan Sistem Informasi Pendaftaran Pasien pada Apotik Cahaya Medika Berbasis *Web*”. Hasil dari penelitian ini membuat sebuah sistem antrian berbasis *web* yang dibangun dengan bahasa pemrograman *PHP* dan *database MySQL*. Diharapkan proses antrian dapat berjalan dengan baik.

Penelitian oleh Andi Suprianto dan Asri Amaliza Fathia Matsea (2018) dalam “Rancang Bangun Aplikasi Pendaftaran Pasien *Online* Dan Pemeriksaan Dokter Di Klinik Pengobatan Berbasis *Web*”. Dalam Penelitian ini membahas perancangan berupa aplikasi pendaftaran pasien *online* dan pemeriksaan dokter yang diharapkan dapat membantu Admin (*staff* administrasi) dan *user* (pasien) dengan memanfaatkan koneksi internet. Aplikasi yang dibangun menggunakan program *PHP* dan *database MySQL*.

Tabel 2.1 berikut merupakan hasil perbandingan penelitian beberapa referensi yang dapat digunakan sebagai acuan sebagai berikut :

Tabel 2.1 Perbandingan Penelitian

PENELITI, TAHUN	OBJEK	METODE/BAHASA PEMROGRAMAN	KERANGKA KERJA	HASIL PENELITIAN
Rika Melyanti, Dedy Irfan,Ambiyar (2020)	Rumah Sakit Syafira	<i>Waterfall/PHP</i>	-	Hasil dari penelitian ini berupa sebuah Sistem Antrian <i>Online</i> yang dapat mengatasi masalah seperti pengambilan nomor antrian, info jadwal dokter dan laporan harian kunjungan pasien. Perancangan sistem antrian dibangun menggunakan Bahasa pemrograman <i>PHP</i> dan basis data <i>MySQL</i> .
Sri Lestari, Aldino Nur Ihsan, Andrian Nur Ihsan, Dionta, dan Julinar Sari Hutagalung (2023)	Puskesmas Seputih Banyak	<i>Waterfall/PHP</i>	<i>Framework CodeIgniter</i>	Hasil penelitian ini membuat sistem pengambilan tiket antrian menjadi lebih mudah dan cepat, sehingga meningkatkan kenyamanan bagi pasien dalam mengakses layanan kesehatan di puskesmas tersebut. Sistem ini dibuat menggunakan bahasa pemrograman <i>PHP</i> dan <i>Framework CodeIgniter</i> dengan basis data <i>MySQL</i>
Chaeron Niesa,Zakial Vikki, dan Agus Maulida (2021)	Apotik Cahaya Medika	<i>Waterfall/PHP</i>	-	Hasil penelitian ini membuat sebuah sistem antrian di Apotik Cahaya Medika berbasis <i>web</i> yang dibangun dengan bahasa pemrograman <i>PHP</i> dan terhubung <i>database MySQL</i> .
Muhammad Bagus Heramwan, Halimahtus Mukminna, Achmad Arif Alfin, dan Yudo Bismo Utomo (2023)	RSI Madinah Ngunut	<i>Waterfall/PHP</i>	<i>Framework Laravel</i>	Hasil penelitian ini merancang sebuah aplikasi pendaftaran poli berbasis <i>web</i> untuk RSI Madinah Ngunut. Aplikasi ini menggunakan <i>framework Laravel</i> dan bertujuan untuk mempermudah proses pendaftaran pasien di poli.

PENELITI, TAHUN	OBJEK	METODE/BAHASA PEMROGRAMAN	KERANGKA KERJA	HASIL PENELITIAN
Andi Suprianto dan Asri Amalia Fathia Matsea (2018)	Klinik Medisina	<i>Spiral/PHP</i>	<i>Bootstrap</i>	Hasil penelitian ini membuat perancangan berupa aplikasi pendaftaran pasien <i>online</i> dan pemeriksaan dokter di Klinik Pengobatan. Aplikasi ini dibangun menggunakan program aplikasi dengan bahasa pemrograman <i>PHP</i> dan menggunakan <i>Bootstrap</i> .
Arfan Marsetyo Pangestu (Usulan)	Puskesmas Prembun	<i>Waterfall/PHP</i>	<i>Framework Laravel</i>	Hasil penelitian ini menghasilkan sebuah <i>website</i> sistem pendaftaran antrian <i>online</i> untuk pasien umum rawat jalan di Puskesmas Prembun menggunakan <i>Framework Laravel</i> , <i>database MySQL</i> , dan bahasa pemrograman <i>PHP</i> .

2.2 Dasar Teori

2.2.1 Antrian

Menurut pernyataan Nengsih & Yustanti (2019:17), Antrian adalah suatu garis tunggu dari nasabah (satuan) yang memerlukan layanan dari satu atau lebih pelayanan (fasilitas layanan). Sedangkan menurut (Gunawan at el., 2022:129), Antrian adalah saat pelanggan dilayani, menunggu layanan saat penyedia layanan (*server*) sedang sibuk, menerima layanan dan meninggalkan sistem demi layanan.

Adapun model antrian yaitu:

1. **FIFO (First In, First Out):** Antrian di mana entitas (misalnya, pelanggan atau data) yang pertama kali datang akan dilayani atau diproses terlebih dahulu, sesuai dengan urutan kedatangan mereka.

2. **LIFO (Last In, First Out):** Antrian di mana entitas terakhir yang masuk akan dilayani atau diproses terlebih dahulu, seperti tumpukan atau stack data di mana item yang dimasukkan terakhir akan diambil terlebih dahulu.
3. **Priority Queue:** Antrian di mana setiap entitas memiliki prioritas tertentu, dan entitas dengan prioritas lebih tinggi akan dilayani atau diproses lebih dahulu daripada yang lain, terlepas dari urutan kedatangan.

2.2.2 Pasien

Pasien adalah orang yang memiliki kelemahan fisik atau mental yang menyerahkan pengawasan dan perawatannya, menerima dan mengikuti pengobatan yang ditetapkan oleh tenaga kesehatan atau para medis di rumah sakit (Anggraini & Oliver, 2019).

Menurut Undang-Undang No. 29 tentang Praktik Kedokteran tahun 2004, pasien memiliki kewajiban sebagai berikut dalam menerima pelayanan:

1. Pasien harus memberikan informasi yang lengkap dan jujur tentang kondisi kesehatannya.
2. Pasien diharapkan untuk mengikuti nasihat dan petunjuk dari dokter atau dokter gigi.
3. Pasien wajib mematuhi peraturan yang berlaku di fasilitas pelayanan kesehatan.
4. Pasien diharapkan memberikan imbalan atas pelayanan yang telah diterimanya sesuai dengan ketentuan yang berlaku.

2.2.3 Framework

Framework atau kerangka kerja adalah komponen siap pakai yang digunakan *developer* untuk menangani berbagai permasalahan dalam pemrograman, seperti pemanggilan variabel, *file*, koneksi ke *database* dan sebagainya (Jh, A. R., & Prastowo, A. T. 2021). Sedangkan pernyataan (Sallaby & Kanedi, 2020) *Framework* adalah kumpulan intruksi-intruksi yang yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang berulang dan menghemat waktu.

2.2.4 Laravel

Laravel merupakan sebuah *framework* yang memudahkan pengembang web dalam mengoptimalkan penggunaan *PHP* untuk pengembangan situs web. *PHP* sendiri merupakan bahasa pemograman yang cukup dinamis (Ibnu, 2022). *Framework* ini telah terbukti mampu mempermudah dalam proses pengembangan *website* dengan beberapa fitur unggulannya, yang beberapa di antaranya meliputi:

1. *Eloquent ORM* : *Framework Laravel* didasarkan pada *ORM Eloquent* yang menyediakan dukungan untuk hampir semua mesin di *database*. Bekerja dengan baik *MySQL* dan *SQLite*.
2. *Blade Template Engine* : *Framework* ini menggunakan *Blade* yang mampu mendesain *layout* yang unik. *Layout* yang didesain dapat digunakan di tampilan lain sehingga dapat menghadirkan konsistensi dalam desain dan struktur selama proses pengembangan.

3. *Routing Laravel* : dapat digunakan untuk membuat aplikasi yang tenang dengan mudah. Dimana dalam *framework* ini semua permintaan *request* dipetakan dengan bantuan *route*.
4. Terdapat *composer* : merupakan kumpulan modul dan *library* yang terkait. Fitur tersebut akan membantu dalam menyempurnakan dan meningkatkan fungsionalitas pada *website* yang akan dibangun, serta mempermudah proses pembaruannya.
5. *Testing and Debugging* : *Framework* ini dibangun dengan fitur proses pengecekan yang cukup lengkap. Ini bisa dinyatakan sebagai: "Ini mendukung proses pengecekan dengan *PHP Unit* dan *file phpunit.xml* yang dapat disesuaikan dengan aplikasi *web* yang sedang dibangun."
6. *Query Builder and ORM* : *Laravel Database Query Builder* menyediakan antarmuka yang lancar untuk membuat dan menjalankan *database query*. *Fitur* ini digunakan dalam menjalankan berbagai operasi di *database* dalam sebuah *website* dan mendukung berbagai sistem *database*.

Laravel memiliki kelebihan yang tidak dimiliki dari *PHP Framework* yang lainnya. Berdasarkan (Zanin & Wernke, 2019) dalam artikelnya yang berjudul "A Comparative study of PHP frameworks performance", *laravel* mempunyai keunggulan dibandingkan dengan *Symfony* dan *CodeIgniter*. *Laravel* memiliki permintaan per detik (*request per second*) tertinggi dibandingkan dengan *Symfony* dan *CodeIgniter*. Selain hal itu, *laravel* juga unggul pada waktu respon (*response time*). Namun, *Laravel* memiliki kekurangan dibandingkan kedua *framework* lainnya, yaitu dalam hal jumlah file (*number of files*).

2.2.5 MVC (*Model, View, Controller*)

Menurut pernyataan Peter Spath (2021:1), *MVC* merupakan pola desain perangkat lunak. *MVC* menggambarkan pemisahan terhadap perangkat lunak menjadi tiga elemen utama sebagai berikut :

1) *Model*: Mengelola data aplikasi. Ini harus dipahami dalam arti sempit. Bagian penting dari aplikasi berurusan dengan data, dan model dalam *MVC* menangani item data yang dapat dilihat dan diubah oleh pengguna..

2) *View*: Menjelaskan penyajian data dan elemen kontrol (*input*, tombol, kotak centang, menu, dan sebagainya) kepada pengguna. Tampilan juga dapat menggunakan perbedaan teknologi, seperti komponen *GUI (Graphical User Interface)* yang diinstal pada *PC* pengguna, dan aplikasi di ponsel, atau halaman *web* untuk dilihat di *browser*.

3) *Controller*: Menangani input pengguna dan menyiapkan data yang diperlukan untuk tampilan. Tampilan hanya menampilkan item model tanpa perlu mengetahui cara data disimpan atau diambil dari *database*, karena itu adalah tanggung jawab *controller*. Dengan demikian, *input* pengguna menentukan tindakan selanjutnya, dan *controller* berisi logika aplikasi.

2.2.6 PHP Hypertext Preprocessor (PHP)

Berdasarkan pernyataan Riri Fitri Sari dan Ardiati (2021:126): “*PHP* atau *Hypertext Preprocessor*” merupakan bahasa scripting yang dirancang untuk membuat halaman *web* dinamis. *PHP* dapat melakukan pengkodean seperti halnya *Java* dan *C#*, serta dilengkapi infrastruktur *OOP (Object-Oriented Programming)* yang digunakan untuk membuat aplikasi *web* kompleks, *modular*, dan *reusable*

menjadi lebih mudah.” Sedangkan menurut (Muhammad Romzi & Kurniawan, 2020) *PHP* atau *Hypertext Preprocessor* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis, dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*.

2.2.7 MySQL

Berdasarkan pernyataan Nugroho (2019), “*MySQL* adalah sebuah *database* yang paling populer di kalangan *Programmer Web*, dengan alasan bahwa program ini merupakan *database* yang sangat kuat dan stabil.” Sedangkan pernyataan dari (Utomo dkk.,2020) *MySQL* merupakan sistem manajemen basis data yang saling terhubung dan didistribusikan secara gratis namun dengan batasan *software* tidak boleh dijadikan produk dagang. *MySQL* adalah turunan dari konsep utama dalam basis data, yaitu *SQL (Structured Query Language)*.

2.2.8 WhatsApp API

Whatsapp adalah aplikasi *smartphone* yang menyediakan fitur untuk mengirim dan menerima pesan, gambar, dan video serta mampu melakukan panggilan. Aplikasi ini memanfaatkan koneksi internet untuk melakukan komunikasi data. *Api* adalah singkatan dari *Application Programming Interface*. *Api* dapat memungkinkan 2 aplikasi atau lebih untuk melakukan integrasi data secara bersamaan. Menurut Dewantya (2018), *Whatsapp Api* adalah layanan antarmuka yang berisi semua *functionalities* dari layanan *whatsapp* yang memungkinkan dapat terjadinya proses menerima dan mengirim pesan dan *file* dari masing-masing *user*.

2.2.9 API (*Application Programming Interface*)

API, singkatan dari (*Application Programming Interface*), memungkinkan pengembang untuk mengintegrasikan berbagai fungsi dan layanan antara dua aplikasi atau dengan sistem lain secara efisien. *API* terdiri dari elemen-elemen seperti fungsi, protokol, dan alat lain yang memudahkan pengembang dalam membangun aplikasi dengan cara yang efektif dan cepat. Salah satu tujuan utama penggunaan *API* adalah untuk mengurangi waktu dan usaha yang diperlukan dalam pengembangan *software* dengan menyediakan akses terstruktur ke fitur-fitur yang telah ada. (<https://www.codepolitan.com/>)

2.2.10 FIFO (*First In First Out*)

Menurut Pernyataan Riswan et al (2019), Model *First In First Out (FIFO)* adalah sebuah perancangan sistem yang didasarkan pada pemikiran bahwa persediaan barang paling awal, akan dijual terlebih dahulu. Karena metode ini, harga barang yang dibeli untuk pertama kali dibeli akan tercatat sebagai harga pokok penjualan. Dengan metode ini, aliran biaya yang sebenarnya dapat dipertahankan, dan stok yang lebih lama dapat dilikuidasi terlebih dahulu. Karena pengeluaran produk dari gudang biasanya lebih bergantung pada pengaturan barang, pendekatan (*FIFO*) seringkali lebih jelas dalam perhitungan harga pokok dibandingkan dengan aliran fisik barang.