

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini mengambil beberapa penelitian yang pernah dilakukan sebelumnya sebagai acuan dan pedoman dalam membangun aplikasi. Berikut adalah beberapa sumber acuan tersebut.

Tinjauan pustaka yang pertama diperoleh dari Muhammad Fauzi Masykur (2015) melakukan penelitian dengan judul “Perancangan Sistem Antrian Pelanggan Perbankan Berbasis *Mobile*”. Penelitian ini membangun sebuah sistem antrian berbasis *mobile* yang digunakan untuk menampilkan informasi kepadatan antrian pelanggan yang dapat dipantau dimana saja secara *online* menggunakan *smartphone*. Model antrian yang digunakan adalah *First In First Out* (FIFO).

Tinjauan pustaka yang kedua diperoleh dari Muhammad Junaedin (2019) melakukan penelitian dengan judul “Sistem Antrian *Online* Berbasis Web menggunakan *Framework* CodeIgneter (Studi Kasus DISNAKERTRANS DIY)”. Penelitian ini membangun sebuah sistem antrian berbasis web yang digunakan untuk mempermudah *user* dalam menunggu antrian dan berbagi informasi kepada para *user* tentang hasil wajib lapor ketenaga kerjaan yang sudah di resmikan dari pihak dinas ketenaga kerjaan. Model antrian yang digunakan adalah *First In First Out* (FIFO), sistem dibangun menggunakan *Framework* CodeIgneter (CI) dan juga menggunakan *SMTP Server* yang digunakan untuk mengirimkan *email* notifikasi ke perusahaan.

Tinjauan pustaka yang ketiga diperoleh dari Hendarman Lubis, Indah Dwijayanthi Nirmala, Satria Eka Nugroho (2019) melakukan penelitian dengan judul “Perancangan Sistem Informasi Antrian *Online* Pasien RS. Seto Hasbadi menggunakan SMS Gateway Berbasis Android”. Penelitian ini merancang sebuah sistem informasi antrian berbasis android yang digunakan untuk mempermudah proses pendaftaran antrian secara *online* baik dari pihak instansi rumah sakit maupun pasien, mempermudah pasien dalam mengetahui seputar informasi jadwal praktek dokter dan proses pengambilan nomor urut antrian. Metode yang digunakan dalam perancangan sistem ini menggunakan metode *prototyping*, juga memanfaatkan teknologi SMS Gateway yang digunakan untuk mempermudah pengelolaan data pasien sehingga diperoleh tampilan yang lebih menarik.

Tinjauan pustaka yang keempat diperoleh dari Oktavia Susanti (2020) melakukan penelitian dengan judul “Sistem Antrian Pembayaran SPP menggunakan *Quick Response Code* sebagai Identitas Antrian di STMIK AKAKOM Yogyakarta”. Penelitian ini membangun sebuah sistem antrian *online* berbasis web yang digunakan untuk membantu proses pelayanan pembayaran SPP menggunakan *Quick Response Code* sebagai id pembayaran sehingga pelayanan menjadi lebih cepat dan dapat terkoneksi dengan mahasiswa secara *real-time* sehingga memberikan informasi pelayanan antrian melalui *smartphone* tanpa perlu datang ke lokasi.

Tinjauan pustaka yang kelima diperoleh dari Wedi Aria Santana (2020) melakukan penelitian dengan judul “Aplikasi Sistem Antrian Berbasis Web”. Penelitian ini membangun sebuah sistem antrian *online* yang digunakan untuk

mengurangi kepadatan antrian dan juga menghemat waktu pelanggan dikarenakan pelanggan dapat melakukan *request* nomor antrian dari luar instansi perbankan, serta membantu instansi perbankan dalam memonitoring dan mengelola antrian. Sistem dibangun menggunakan *Framework* Laravel.

Penelitian “Sistem Antrian Pasien Rawat Jalan di Rumah Sakit Yowari “ ini membangun sebuah sistem antrian *online* berbasis *mobile* yang digunakan untuk membantu pasien dalam mendapatkan informasi secara efisien dan *real-time* tentang nomor antrian pemeriksaan yang sedang berjalan, jadwal praktek dokter dan dapat melakukan *booking* antrian pemeriksaan. Model antrian yang digunakan adalah *First In First Out* (FIFO). Metode yang digunakan dalam membangun sistem antrian ini adalah *Rapid Application Development* (RAD), menggunakan *Framework* NestJS dan Flutter. Perbandingan tinjauan pustaka bisa dilihat dalam tabel 2.1.

Tabel 2.1 Perbandingan Tinjauan Pustaka

No	Nama Peneliti	Judul	Metode (Teknologi)	Objek	Hasil Penelitian
1.	Muhammad Fauzi Masykur (2015)	Perancangan Sistem Antrian Pelanggan Berbasis <i>Mobile</i>	Metode <i>First In First Out</i> atau <i>Quick Response Code</i> atau <i>Mobile</i>	Antrian nasabah bank	Sistem antrian <i>online</i> nasabah bank
2.	Muhammad Junaedin (2019)	Sistem Antrian <i>Online</i> Berbasis Web menggunakan <i>Framework CodeIgneter</i> (Studi Kasus DISNAKERTRANS DIY)	Metode <i>First In First Out</i> atau <i>Framework CodeIgneter</i> , SMTP <i>Server</i> . Web	Antrian <i>customer</i> DISNAKERTRANS DIY	Sistem antrian <i>online customer</i> DISNAKERTRANS DIY
3.	Hendarman Lubis, Indah Dwijayanthi Nirmala, Satria Eka Nugroho (2019)	Perancangan Sistem Informasi Antrian <i>Online</i> Pasien RS. Seto Hasbadi menggunakan SMS <i>Gateway</i> Berbasis Android	Metode <i>Prototyping</i> atau SMS <i>Gateway</i> , Android	Antrian pasien Rumah Sakit Seto Hasbadi	Sistem antrian <i>online</i> pasien Rumah Sakit Seto Hasbadi
4.	Oktavia Susanti (2020)	Sistem Antrian Pembayaran SPP menggunakan <i>Quick Response Code</i> sebagai Identitas Antrian di STMIK AKAKOM Yogyakarta	<i>Quick Response Code</i> , Web	Antrian pembayaran SPP STMIK AKAKOM Yogyakarta	Sistem antrian <i>online</i> pembayaran SPP
5.	Wedi Aria Santana (2020)	Aplikasi Sistem Antrian Berbasis Web	<i>Framework</i> Laravel, Web	Antrian nasabah bank Lombok	Sistem antrian <i>online</i> nasabah bank
6.	Triwibowo Ridho Permana (2022)	Sistem Antrian Pasien Rawat Jalan di Rumah Sakit Yowari	Metode FIFO, <i>Rapid Application Development</i> atau <i>Framework</i> NestJS, Flutter, Mobile	Antrian pasien rawat jalan Rumah Sakit Yowari	Sistem antrian <i>online</i> pasien rawat jalan di Rumah Sakit Yowari

2.2 Dasar Teori

2.2.1 Antrian

Menurut Bronson (1996) antrian adalah proses yang berhubungan dengan kedatangan seseorang pelanggan pada saat fasilitas pelayanan, kemudian menunggu dalam suatu barisan (antrian) jika semua pelayanan sibuk dan akhirnya meninggalkan fasilitas tersebut.

Menurut Yamit (2004) antrian merupakan suatu kondisi dimana adanya keterlambatan pelayanan suatu objek akibat adanya antrian karena pelayanan mengalami kesibukan.

Menurut Siagian (2006) antrian adalah suatu garis tunggu dari nasabah yang memerlukan layanan dari satu atau lebih pelayanan. Kejadian garis tunggu timbul disebabkan oleh kebutuhan akan layanan melebihi kemampuan (kapasitas) pelayanan atau fasilitas layanan, sehingga nasabah yang tiba tidak bisa segera mendapat layanan disebabkan kesibukan pelayanan.

Menurut Salahudin (2010: 97) mengemukakan bahwa antrian atau *Queue* adalah salah satu struktur data yang memiliki sistem kerja pertama masuk dan pertama keluar seperti halnya antrian pada dunia nyata. *Queue* pada penambahan datanya hanya dapat dilakukan pada salah satu ujung dan penghapusan data hanya dapat dilakukan pada ujung yang lain. Elemen baru dimasukan pada bagian belakang dalam antrian dan penghapusan elemen hanya dilakukan pada bagian ujung depan elemen.

Berdasarkan definisi tersebut, maka dapat disimpulkan bahwa antrian adalah suatu proses yang berhubungan dengan kedatangan seseorang pada suatu fasilitas

pelayanan, kemudian menunggu dalam suatu antrian yang disebabkan oleh kebutuhan akan layanan melebihi kemampuan (kapasitas) pelayanan atau fasilitas layanan dan pada akhirnya meninggalkan fasilitas tersebut. Jadi, sistem antrian adalah himpunan pelanggan, pelayan dan suatu aturan yang mengatur kedatangan para pelanggan dan pemrosesan masalahnya.

2.2.2 *First In First Out (FIFO)*

Menurut Siagian (1987) *First In First Out (FIFO)* adalah salah satu bentuk disiplin antrian yang mana pelanggan yang lebih dulu datang (sampai), lebih dulu dilayani (keluar).

Menurut Kemal Nasir & Renggo Pribadi (2007: 40) *First In First Out (FIFO)* adalah sebuah algoritma untuk menerapkan disiplin antrian yang tidak berprioritas. Algoritma ini menggunakan struktur data. Algoritma FIFO sering di gunakan di berbagai macam pemecahan masalah kehidupan dan berbagai aplikasi serta teknologi yang ada. FIFO sendiri merupakan algoritma yang bersifat be urutan dan bergiliran namun tetap pada alur atau jalurnya sesuai dengan yang pertama kali masuk dan kemudian diproses sesuai dengan giliran.

2.2.3 **Android**

Android adalah sistem operasi seluler yang dikembangkan oleh Google. Ini digunakan oleh beberapa *smartphone* dan *tablet*. Sistem operasi (OS) android didasarkan pada kernel Linux. Android adalah sistem operasi *open-source*, artinya android menyediakan *platform* terbuka bagi para pengembang dalam menciptakan

aplikasi mereka sendiri untuk digunakan oleh berbagai macam perangkat *mobile*. (Anggit Kurniawati, 2020). Oleh karena itu, ponsel berbasis android seringkali memiliki antarmuka yang berbeda meskipun mereka menggunakan OS yang sama. Dalam pengembangan aplikasi android terdapat beberapa pilihan bahasa pemrograman yaitu Java, Kotlin, Python, C# dan Dart.

2.2.4 Flutter SDK

Flutter SDK merupakan perangkat *User Interface (UI) open-source* lintas *platform* yang dikembangkan oleh Google yang memungkinkan pengembang untuk mengembangkan aplikasi iOS, Android, Web (versi *beta*) dan Desktop (versi pengembangan) dengan basis kode yang sama. Sasaran pengembangannya adalah untuk memberikan aplikasi berkinerja tinggi di berbagai *platform* dan sangat cocok digunakan untuk mengembangkan aplikasi yang memelurkan tampilan yang indah. Fitur-fitur yang dimiliki oleh Flutter adalah :

1. *Hot reload* yang berjalan di *Dart Virtual Machine* untuk melihat perubahan tanpa melakukan *compile* ulang.
2. Setiap *project* flutter mempunyai sebuah *file* bernama *pubspec.yaml* atau sering juga disebut *pubspec*. *Pubspec file* menentukan *dependencies* yang dibutuhkan oleh sebuah *project* flutter seperti *package-package* tertentu (berserta versi-versinya), *font* dan juga gambar.
3. Kerangka kerja yang reaktif.
4. Kaya akan widget yang menerapkan material design.
5. Dart devtools untuk menguji dan men-debug aplikasi.

Flutter menggunakan bahasa pemrograman Dart yang bisa mudah dipahami jika sudah mengenal konsep pemrograman berorientasi objek (kelas, metode dan variabel) dan konsep pemrograman imperatif (perulangan dan kondisional). (flutter.dev, 2020).

2.2.5 *Framework NestJS*

NestJS adalah sebuah *Framework* untuk membangun aplikasi sisi *server* Node.js yang efisien dan skalabel. NestJS menggunakan JavaScript progresif, dibangun dengan dan sepenuhnya mendukung TypeScript (namun masih memungkinkan *developer* untuk membuat *code* dalam JavaScript murni) dan menggabungkan elemen *Object Oriented Programming* (OOP), *Functional Programming* (FP), dan *Functional Reactive Programming* (FRP).

NestJS menggunakan *Framework Server Hypertext Transfer Protocol* (HTTP) yang kuat seperti Express (*default*) dan secara opsional dapat dikonfigurasi untuk menggunakan Fastify juga.

NestJS menyediakan tingkat abstraksi di atas *Framework* NodeJS, tetapi juga memaparkan *Application Programming Interface* (API) mereka langsung ke *developer*. Ini memberi *developer* kebebasan untuk menggunakan banyak modul pihak ketiga yang tersedia untuk platform yang mendasarinya.

2.2.6 *MySQL*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan

sekitar 6 juta instalasi di seluruh dunia. MySQL adalah perangkat lunak yang merupakan sebuah implementasi dari sistem manajemen basis data relasional (RDBMS - *Relational Database Management System*) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan turunan yang bersifat komersial.

Pada berbagai kepentingan, MySQL memiliki fungsi dan kegunaan yang sangat penting dalam menangani data. Sebagai contoh, MySQL dapat menangani persediaan barang dan mencatat data penjualan dalam sistem toko *online* (*e-commerce*). Dengan adanya sistem MySQL maka manajemen toko *online* dapat mengetahui persediaan barang yang ada didalamnya apakah masih tersedia atau kosong. Selain itu MySQL dapat memberikan laporan berupa informasi penjualan baik secara harian, mingguan, bulanan bahkan sampai tahunan. (YM Kusuma Ardhana., S.T., M.Kom., 2014).

2.2.7 Metode *Rapid Application Development* (RAD)

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode RAD. Menurut Roger S. Pressman, RAD adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. RAD melingkupi fase-fase sebagai berikut:

1. Pemodelan Bisnis

Aliran informasi di antara fungsi-fungsi bisnis dimodelkan dengan suatu cara untuk menjawab pertanyaan-pertanyaan berikut: Informasi apa yang mengendalikan proses bisnis? Informasi apa yang dimunculkan? Siapa yang memunculkannya? Kemana informasi itu pergi? Siapa yang memprosesnya?

2. Pemodelan Data

Aliran informasi yang didefinisikan sebagai bagian dari fase pemodelan bisnis disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik (disebut atribut) masing-masing objek didefinisikan dan hubungan antara objek-objek tersebut didefinisikan.

3. Pemodelan Proses

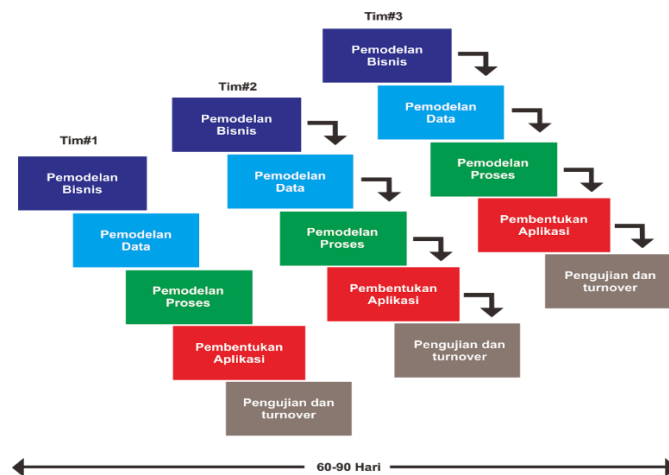
Aliran informasi yang didefinisikan di dalam fase pemodelan data ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.

4. Pembentukan Aplikasi

RAD mengasumsikan pemakaian teknik generasi ke-empat. Selain menciptakan perangkat lunak dengan menggunakan bahasa pemrograman generasi ketiga yang konvensional, RAD lebih banyak memproses kerja untuk memakai lagi komponen program yang ada (pada saat memungkinkan) atau menciptakan komponen yang bisa dipakai lagi (bila perlu). Pada semua kasus, alat-alat bantu otomatis dipakai untuk memfasilitasi konstruksi perangkat lunak.

5. Pengujian dan *Turnover*

Proses RAD menekankan pada pemakaian kembali, banyak komponen program yang telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus diuji dan semua interface harus dilatih secara penuh.



Gambar 2. 1 Bagan Metode *Rapid Application Development*

Gambar 2.1 adalah bagan metode *Rapid Application Development* yang merupakan metode pengembangan sistem yang digunakan pada penelitian ini.

2.2.8 *Black-Box* dan *White-Box Testing*

Pada *Black-box Testing* dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black-box Testing* ini lebih menguji ke tampilan luar (*interface*) dari suatu aplikasi agar mudah digunakan oleh *customer*. Pengujian ini tidak melihat dan menguji *source code* program. *Black-box Testing* bekerja dengan mengabaikan struktur *control* sehingga perhatiannya hanya terfokus pada informasi domain. (socs.binus.ac.id, 2020).

Keuntungan penggunaan metode *Black-box Testing* yaitu, pertama pengujian tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu. Kedua, pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan. Dan ketiga, programmer dan tester keduanya saling bergantung satu sama lain.

Sedangkan kekurangan dari metode *Black-box Testing* adalah kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer* dan beberapa bagian back-end tidak diuji sama sekali. (Tri Snadhika Jaya, 2018).

White-box Testing adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan melihat modul untuk memeriksa dan menganalisis kode *program* ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam *output* yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan, singkatnya *White-box Testing* ini menguji dengan cara melihat *source code* dari suatu aplikasi (*software*) yang diuji tanpa memperdulikan tampilan antar muka atau UI (*User Interface*) dari aplikasi tersebut. (socs.binus.ac.id, 2020).

White-box Testing merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Ketika program diterjemahkan kedalam kode sumber bahasa pemrograman, maka akan terjadi kesalahan pengetikan. Banyak yang terdeteksi dengan mekanisme pemeriksaan sintaks, tetapi banyak juga yang tidak terdeteksi sampai dengan dimulainya uji coba. Karena alasan tersebut, maka uji coba *White-box Testing* diperlukan selain *Black-box Testing*. (Muhammad Nuris, 2015)