

BAB II LANDASAN TEORI

Bab II Landasan Teori mengandung penjabaran teori-teori dasar pembahasan dalam penelitian dan tinjauan pustaka digunakan untuk memberikan informasi penelitian yang serupa sebagai perbandingan. Berikut dasar teori dan tinjauan pustaka dalam penelitian.

2.1 Dasar Teori

Dasar teori dalam penelitian Rancang Bangun Sistem Antrian *Online* dengan *Websocket* menggunakan Metode *Single Channel-Multi Phase* sebagai berikut:

2.1.1 Pengertian Antrian

Queuing theory atau Teori antrian merupakan bagian penting dalam operasi dan juga alat yang berharga dalam manajemen operasi. Seorang insinyur Denmark yang bernama A.K. Erlang yang memperkenalkan teori ini. Krajewski, dkk (2010) mengatakan bahwa, antrian ialah satu atau sekelompok “pelanggan” yang menunggu pelayanan. Teori antrian adalah teori yang menyangkut studi matematis dari antrian-antrian atau baris-baris penungguan.

2.1.2 Komponen Dasar Antrian

Berikut adalah komponen dasar antrian.

1. Kedatangan

Antrian selalui dimulai dengan kedatangan, misalnya pasien, pesanan, dan lain-lain. Hal ini sering disebut dengan proses *input*. Proses *input* termasuk sumber kedatangan (*calling population*), dan proses kedatangan yang pada umumnya adalah variabel acak.

2. Pelayanan

Mekanisme pelayanan terdiri dari satu atau lebih fasilitas pelayanan, atau satu atau lebih pelayan. Tiap-tiap fasilitas pelayanan kadang-kadang disebut sebagai saluran (*channel*).

3. Antrian

Antrian merupakan inti dari analisa antrian. Sifat kedatangan dan proses pelayanan mempengaruhi terjadinya antrian. Jika ada kelebihan fasilitas pelayanan atau terdapat pelayan yang menganggur berarti tidak ada terjadi antrian.

Disiplin antri adalah aturan yang menjelaskan tentang melayani pengantri. Menurut Siagian (2003), terdapat 4 bentuk disiplin pelayanan yang sering digunakan, yaitu:

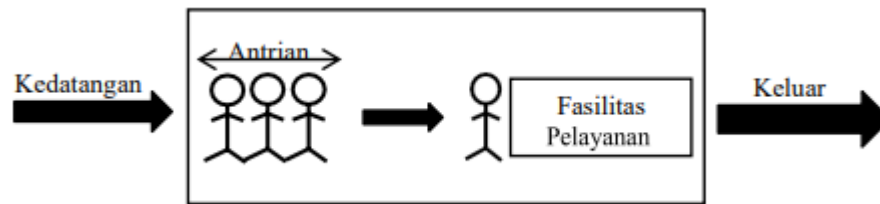
- a. *First Come First Served (FCFS)* atau *First In First Out (FIFO)* artinya, pengantri yang datang lebih dulu sampai atau masuk, lebih dulu menerima pelayanan. Misalnya, antrian loket pembelian tiket kereta.
- b. *Last Come First Served (LCFS)* atau *Last In First Out (LIFO)* artinya, pengantri yang datang atau masuk paling akhir, lebih dulu menerima pelayanan atau keluar. Misalnya, sistem antrian dalam elevator untuk lantai yang sama.
- c. *Service In Random Order (SIRO)* artinya, panggilan antrian yang berdasarkan peluang secara acak, tidak bergantung dengan kedatangan.
- d. *Priority Service (PS)* artinya, pelayanan diberikan kepada pelanggan yang mempunyai kepentingan atau prioritas yang lebih tinggi dibandingkan dengan yang memiliki prioritas yang lebih rendah, meskipun yang memiliki prioritas yang lebih tinggi data paling akhir, misalnya pasien yang memiliki keadaan yang lebih darurat dibanding dengan pasien lain dalam satu puskesmas.

2.1.3 Struktur Antrian

Menurut Pangestu Subagyo dkk (2002), struktur antrian dasar terdapat 4 model umum terjadi dalam sistem antrian. Berikut adalah penjabarannya:

1. *Single Channel-Single Phase*

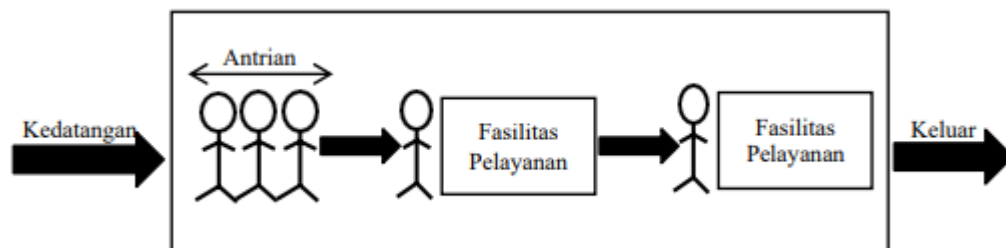
Single channel artinya yang memasuki sistem pelayanan hanya ada satu jalur atau ada satu fasilitas pelayanan. *Single phase* artinya hanya terdapat satu pelayanan. Berikut adalah gambar 2.1 mengenai *Single Channel-Single Phase*.



Gambar 2.1 *Single Channel-Single Phase* (Sumber: Kakiay, 2004)

2. *Single Channel-Multi Phase*

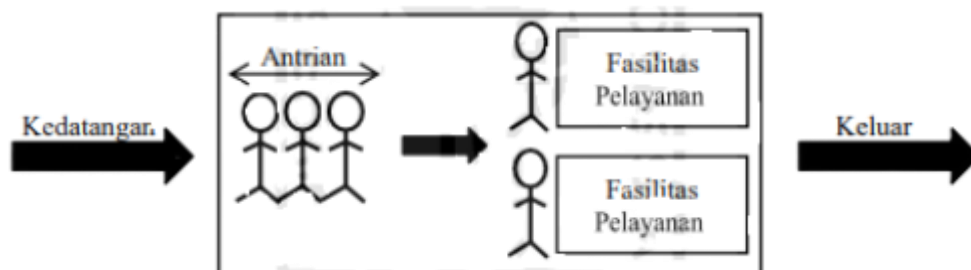
Multi phase artinya ada lebih dari satu pelayanan yang dilaksanakan secara berurutan (dalam *phase-phase*). Sebagai contoh: pencucian mobil. Berikut adalah gambar 2.2 mengenai *Single Channel-Multi Phase*.



Gambar 2.2 *Single Channel-Multi Phase* (Sumber: Kakiay, 2004)

3. *Multi Channel-Single Phase*

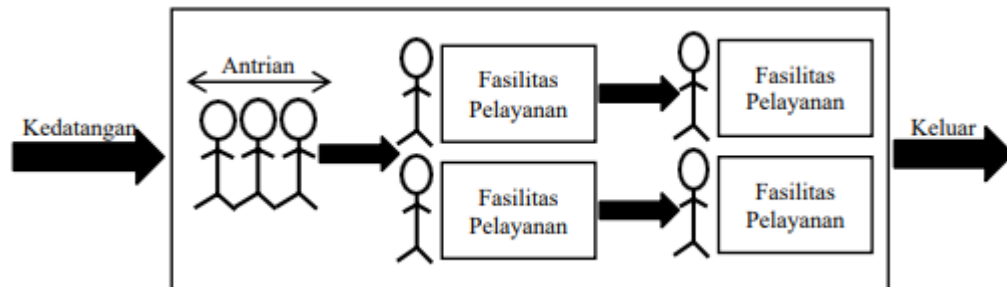
Sistem *multi channel-single phase* terjadi bila lebih dari satu fasilitas pelayanan yang dialiri oleh antrian tunggal. Sebagai contoh model ini adalah antrian pada *teller* sebuah bank. Berikut adalah gambar 2.3 mengenai *Multi Channel-Single Phase*.



Gambar 2.3 *Multi Channel-Single Phase* (Sumber: Kakiay, 2004)

4. *Multi Channel-Multi Phase*

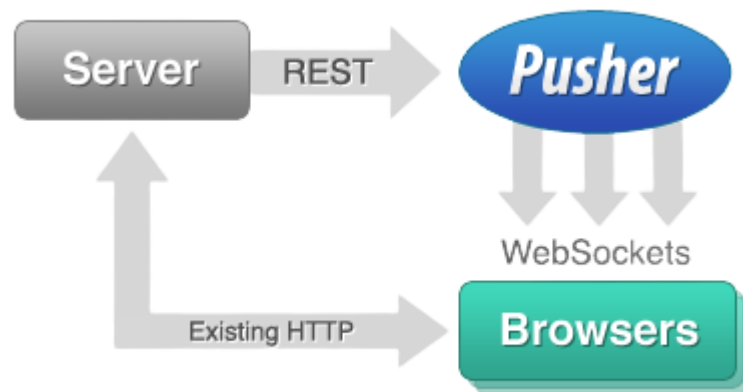
Sistem *multi channel-multi phase* yaitu mempunyai beberapa fasilitas pelayanan pada setiap tahapnya. Berikut adalah gambar 2.4 mengenai *Multi Channel-Multi Phase*.



Gambar 2.4 *Multi Channel-Multi Phase* (Sumber: Kakiay, 2004)

2.1.4 **WebSocket**

Menurut Wang, dkk (2013) websocket adalah sebuah protocol dimana merupakan fitur konektivitas baru dari HTML5. WebSocket dapat melakukan perpindahan data 2 arah dalam suatu waktu atau yang dikenali dengan transmisi data full-duplex. WebSocket merupakan standar baru Web dan aplikasi mobile dalam komunikasi secara *real time*. WebSocket dibangun untuk digunakan di browser web dan server web, tetapi dapat juga dapat digunakan oleh aplikasi client atau server. WebSocket merupakan protokol saluran komunikasi yang dapat melakukan komunikasi saling berbagi informasi secara interaktif dalam waktu yang sama antar dua belah pihak atau sering di sebut dengan *full-duplex* melalui koneksi TCP tunggal. Berikut merupakan arsitektur WebSocket dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur WebSocket (Sumber: medium.com/laravel-pusher-real-time-notification)

WebSocket merupakan bagian dari HTML5. WebSocket dapat mengurangi *latency* dan lalu-lintas jaringan yang tidak penting daripada dengan polling dan long-polling yang sering dimanfaatkan untuk menjalankan koneksi dua arah dengan cara menjaga dua koneksi tetap terhubung. WebSocket sering digunakan ketika suatu sistem bersifat *realtime* dan adanya interaksi data antar pengguna, contohnya seperti aplikasi *chatting*. Aplikasi *chatting* akan otomatis memunculkan data ketika ada *client* lain mengirim data.

2.1.5 *Pusher*

Pusher adalah API *host* untuk mempercepat, mempermudah dan aman dalam menambahkan fungsionalitas *real-time scalable* melalui *WebSockets* untuk *web* dan aplikasi *mobile*. *Pusher* merupakan salah satu penyedia layanan *web socket* yang populer. Istilah-istilah yang ada didalam *pusher*:

1. *Channel*

Channel adalah cara membedakan *stream* data didalam aplikasi. Aplikasi dapat memiliki beberapa *channel* dan satu *channel* dapat memiliki beberapa klien.

2. *Events*

Events sama dengan *server* mengirimkan data ke klien sehingga pengguna dapat melihat pesan di *chat room*. Event dipicu oleh penerbit Perpustakaan, dan pelanggan dapat berlangganan untuk event ini.

Ada tiga jenis channel:

1. *Public Channel*: *channel* yang siapa pun dapat berlangganan, selama mereka tahu nama *channel*.
2. *Private Channel*: *channel* yang hanya autentikasi pengguna dapat berlangganan.
3. *Presence Channel*: mirip dengan *private channel*, tetapi memungkinkan untuk memberitahu klien lain terhubung dengan informasi tentang klien yang terhubung.

2.1.6 Sistem Informasi

Menurut Putri dan Siptiana (2019), Sistem Informasi adalah cara yang diatur untuk mengumpulkan, memasukkan dan mengolah serta menyimpan data, dan cara yang diatur untuk melaporkan, mengendalikan, mengelola bahkan menyimpan informasi sehingga organisasi dapat mencapai tujuan.

Menurut Jogiyanto H. M. (2000), Sistem Informasi merupakan kegiatan dari langkah-langkah yang diorganisasikan, apabila dijalankan akan memberikan informasi dalam pengambilan keputusan dan pengendalian.

2.1.7 Website

Menurut Hidayat, (2010) website atau situs gabungan halaman-halaman bersifat statis maupun dinamis yang membentuk satu keterkaitan yang menampilkan informasi teks, gambar, animasi, suara, dan atau perpaduan semuanya, yang masing masing dihubungkan dengan jaringan-jaringan halaman”.

Menurut Simarmata, (2010) website dapat diartikan sebagai alat bantu untuk menciptakan sistem informasi global yang mudah berdasarkan hypertext.

2.1.8 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah bahasa pemodelan pada rekayasa perangkat lunak, untuk menyediakan cara standar untuk menggambarkan sistem yang akan dibangun. Menurut Booch (2005) UML adalah standar untuk pemodelan rancangan yang akan dibangun. UML sering dimanfaatkan untuk menggambarkan dan membangun, dokumen artefak dari *software intensive system*.


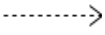

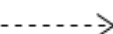

Menurut Nugroho (2009), UML (*Unified Modeling Language*) adalah pemodelan sistem atau *software* yang berorientasi objek. Pemodelan yang digunakan untuk menyederhanakan permasalahan yang sehingga lebih mudah untuk dipahami dan dipelajari.

Menurut Nugroho (2009), UML (*Unified Modeling Language*) adalah Metodologi gabungan antara metode-metode OMT (*Object Modeling Technique*), OOSE (*Object Oriented Software Engineering*), serta Booch dan metode lainnya, merupakan sebuah metodologi yang saat ini sering digunakan dalam analisis dan merancang sistem yang berorientasi objek karena seringnya penggunaan bahasa “pemrograman berorientasi objek” (OOP), berikut adalah jenis jenis UML (*Unified Modeling Language*).






1. Usecase Diagram

Use case diagram adalah proses atau perbuatan memisahkan interaksi antar sistem dan aktor. Use case diagram mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri dengan menceritakan bagaimana sistem digunakan. Berikut merupakan notasi yang digunakan dalam use case diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1 Notasi *Use Case Diagram*

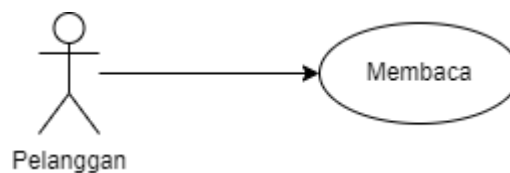
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen 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 <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

Tabel 2.1 Notasi *Use Case Diagram* (Lanjutan)

6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	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
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

Use case diagram berguna dalam tiga hal:







- Menjelaskan fasilitas yang ada (*irequirement*).
- Komunikasi dengan klien.
- Membuat test dari kasus-kasus secara umum.

Gambar 2.6 *Use Case Diagram* (Sumber: Nugroho, 2005)

2. Activity Diagram

Activity diagram digunakan untuk memodelkan proses-proses yang terjadi dalam sistem. Activity diagram menggambarkan aktivitas yang bisa dilakukan oleh sistem. Berikut merupakan notasi yang digunakan dalam activity diagram dapat dilihat pada Tabel 2.2.

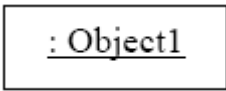


Tabel 2.2 Notasi *Activity Diagram*

NO	SIMBOL	KETERANGAN
1		Titik Awal
2		Titik Akhir
3		Activity
4		Pilihan untuk mengambil keputusan
5		Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
6		Rake; Menunjukkan adanya dekomposisi


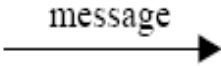
3. Sequence Diagram

Sequence diagram merupakan sebuah diagramnya yang menggambarkan alur atau kegiatan interaksi objek yang terjadi dalam sistem yang disusun berdasarkan urutan waktu. Berikut merupakan notasi yang digunakan dalam sequence diagram dapat dilihat pada Tabel 2.3.

Tabel 2.3 Notasi *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Object</i>	Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma
2		<i>Actor</i>	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan symbol pada Actor Use Case Diagram.
3		<i>Lifeline</i>	Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.

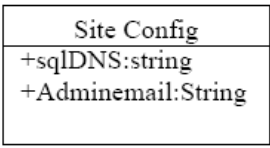
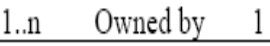
Tabel 2.3 Notasi *Sequence Diagram* (Lanjutan)

4		<i>Activation</i>	Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
5		<i>Message</i>	Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object.




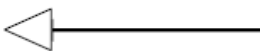
4. Class Diagram

Class diagram merupakan sebuah diagram yang memetakan sistem dengan, memodelkan kelas, atribut, serta menggambarkan hubungan antara kelas. Berikut merupakan notasi yang digunakan dalam class diagram dapat dilihat pada Tabel 2.4.

Tabel 2.4 *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Class</i>	<i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan method dari sebuah <i>class</i> .
2		<i>Assosiation</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 class, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> (Contoh: One-to-one, one-to-many, many-to-many)

Tabel 2.4 *Class Diagram* (Lanjutan)

3		<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.
4		<i>Dependency</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
5		<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.
6		<i>Generalization</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi obyek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “parent”-nya/induknya.

2.1.9 PHP

Menurut Madcoms (2013), PHP merupakan singkatan dari “Hypertext Preprocessor”. Awalnya PHP merupakan singkatan dari *Personal Home Page* (situs personal), PHP pertama kali dibuat pada tahun 1995 oleh Rasmus Lerdorf, dan

bernama FI (*Form Interpreter*), berupa kumpulan-kumpulan *script* yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode tersebut untuk umum. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML.

Menurut Arief (2011) PHP merupakan bahasa *server-side-scripting* yang terhubung dengan HTML dan digunakan untuk membuat halaman web yang dinamis. PHP merupakan *server-side-scripting* maka sintaks PHP dieksekusi pada *server* kemudian hasilnya dikirim ke *browser* dengan format HTML.

Menurut Nugroho (2016), PHP atau (*Personal Home Page*) merupakan bahasa skrip yang terdapat didalam HTML untuk dijalankan bersifat *server side*". PHP termasuk dalam *opensource product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas.

2.1.10 Mysql

Menurut Nugroho (2016) "MySQL (*My Structured Query Language*) sering disebut dengan DBMS (*Database Management System*) atau program pengelolaan database". Database memiliki fungsi sebagai tempat penyimpanan data yang dimasukkan melalui form website dan sebaliknya. MySQL merupakan penyimpanan data yang fleksibel dan cepat aksesnya sangat dibutuhkan dalam sebuah website yang interaktif dan dinamis.

Normalisasi pada basis data merupakan proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entity dan relasinya. Menurut Kusri (2007) bentuk-bentuk normalisasi adalah sebagai berikut:

1. Bentuk tidak normal (Unnormalized Form) pada bentuk ini tidak ada format baku yang harus diikuti, dalam menyimpan kumpulan data sehingga memungkinkan adanya data yang tidak lengkap atau terduplikasi.
2. Bentuk normal pertama (1NF atau First Normal Form) tabel yang termasuk dalam normal pertama (1NF) terjadi apabila nilai kolom dalam setiap baris bernilai tunggal. Artinya jika ada nama kolom yang berulang cukup diwakili oleh sebuah nama kolom.
3. Bentuk normal kedua (2NF atau second Normal Form). Tabel yang termasuk bentuk normal kedua (2NF) terjadi apabila berada dalam bentuk normal pertama, semua kolom bukan kunci primer. Suatu kolom disebut tergantung sepenuhnya

terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama.

4. Normal Ketiga (3NF atau Third Normal Form) suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.
5. Boyce-code Normal Form (BCNF) Normalisasi pada tahap ini mempunyai dari bentuk normal ketiga menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap attribute harus bergantung fungsi pada attribute super key.

2.2 Kajian Penelitian Terdahulu

Berikut ini adalah ringkasan kajian penelitian terdahulu yang digunakan dalam penelitian ini. Kajian penelitian terdahulu dapat dilihat pada Tabel 2.5.

Tabel 2.5 Kajian Penelitian Terdahulu

1	Judul	Perancangan Aplikasi Antrian Pasien Di Rumah Sakit Menggunakan Metode Fast (R. Kurniati, Jaroji)
	Tahun	2018
	Deskripsi	Dalam penelitian ini akan dirancang sebuah aplikasi antrian pasien secara online dengan menerapkan metode <i>Priority Queue</i> dan menggunakan metode FAST sebagai kerangka kerja penelitian
	Metode	<i>Framework for the Application of System Thinking (FAST)</i>
	Hasil	Dengan menggunakan sistem pasien mendapatkan informasi berapa jumlah antrian yang sudah diambil, antrian yang sudah dilayani, dan pasien juga bisa memperkirakan waktu yang diperlukan dalam hal antrian.
2	Judul	Sistem Pemanggilan Antrian Menggunakan Websocket (Nur Rachmat)
	Tahun	2016
	Deskripsi	Pada penelitian ini dikembangkan sistem pemanggilan antrian berbasis web memanfaatkan WebSocket sebagai protocol komunikasi untuk mengirim dan menerima pesan berbasis teks antara petugas loket dan layar informasi antrian.
	Metode	Websocket
	Hasil	Sistem pemanggilan antrian dibangun dengan memanfaatkan protocol WebSocket. Petugas loket dapat melakukan pemanggilan nomor antrian atau mengulang proses pemanggilan nomor antrian menggunakan sistem. Didalam sistem juga dapat memutar suara dengan Javascript Audio API.

Tabel 2.5 Kajian Penelitian Terdahulu (Lanjutan)

3	Judul	Perancangan Sistem Pendaftaran Online Pasien Pada Klinik Dengan Metode FIFO Berbasis Web Service (Dahlan Abdullah, Iswandi)
	Tahun	2015
	Deskripsi	Penelitian ini merancang Sistem Pendaftaran Online Pasien Pada Klinik menggunakan metode FIFO (First in first out) yaitu dengan single channel (satu pintu) ataupun multi channel (banyak pintu)
	Metode	FIFO (First In First Out)
	Hasil	Sistem Pendaftaran Online Pasien membantu Klinik Kimia Farma Bireuen dalam pendaftaran antrian untuk layanan medis dokter serta memberikan penjelasan kepada pasien mengenai jasa pengobatan atau layanan medis yang ditawarkan
4	Judul	Pengembangan Dan Implementasi Aplikasi Antrian Loker Pelayanan Masyarakat Menggunakan Metode Multi Channel Single Phase (Studi Kasus: Disdukcapil Kabupaten Kubu Raya) (Muhammad Sony Maulana)
	Tahun	2016
	Deskripsi	Aplikasi antrian yang dirancang untuk membantu proses pelayanan pada Dinas Kependudukan dan Pencatatan Sipil Kubu Raya, memiliki fasilitas pemanggilan antrian dengan suara melalui speaker berdasarkan nomor urut dan loket antrian serta tampilan pada TV display yang membantu masyarakat melihat nomor antrian dan loket pelayanannya.
	Metode	<i>Multi Channel-Single Phase</i>
	Hasil	Penerapan aplikasi dapat dilakukan secara offline intranet berbasis clientserver dengan memanfaatkan jaringan kabel ataupun wireless. Aplikasi yang dirancang memiliki tiga pengguna yaitu admin, operator dan masyarakat. Berdasarkan hasil pengujian, penerapan aplikasi dengan text dan gambar sebagai display informasi dan suara saat pemanggilan antrian berjalan dengan baik.
5	Judul	Perancangan Sistem Informasi Antrian Online Pasien RS. Seto Hasbadi menggunakan SMS Gateway Berbasis Android (Hendarman Lubis, Indah Dwijyanthi Nirmala, Satria Eka N.)
	Tahun	2019
	Deskripsi	Perancangan Sistem Informasi Antrian Online untuk Pasien pada Rumah Sakit Seto Hasbadi dalam bentuk aplikasi berbasis Android menggunakan SMS gateway yang bertujuan merancang sistem informasi antrian online untuk pasien, sehingga proses pendaftaran antrian dapat dilakukan secara online serta memudahkan pasien dalam mendapatkan seputar informasi jadwal praktek dokter dan memudahkan pasien dalam pengambilan nomor urut antrian.
	Metode	Prototype
	Hasil	Dari hasil pengujian, sistem dapat memudahkan pasien dalam mendaftar di salah satu poli, melihat dokter dan jadwal yang sudah ditentukan, serta melihat riwayat berobat.