BAB II

TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Abdullah dan Iswandi (2018) mengenai perancangan sistem pendaftaran *online* pasien pada klinik dengan metode FIFO berbasis *web service*, Abdullah dan Iswandi menggunakan metode FIFO untuk sistem pendaftaran *online* pasien klinik, sedangkan penelitian penulis menggunakan metode FIFO untuk pendaftaran vaksin di rumah sakit.

Berdasarkan penelitian yang dilakukan oleh Armanza, dkk (2019) mengenai aplikasi pencatatan persediaan barang menggunakan metode *First In First Out* pada PT Amco Multi Tech, Armanza, dkk menggunakan metode FIFO untuk pencatatan persediaan barang, sedangkan penelitian penulis menggunakan metode FIFO untuk proses pendaftaran.

Berdasarkan penelitian yang dilakukan oleh Ervina (2021) mengenai aplikasi Rancang Bangun Sistem Informasi Pendaftaran Pasien Rawat Jalan Berbasis *Android* (Studi Kasus:Puskesmas Pangean), Ervina menggunakan metode FIFO untuk pendaftaran pasien rawat jalan, sedangkan penelitian penulis menggunakan metode FIFO untuk pendafataran vaksin.

Berdasarkan penelitian yang dilakukan oleh Ritno, dkk (2021) mengenai aplikasi pendaftaran pasien rawat jalan pada rumah sakit brimob berbasis *android*, Ritno, dkk menggunakan metode FIFO untuk pendaftaran pasien rawat jalan

padarumah sakit brimob, sedangkan penelitian penulis menggunakan metode FIFO untuk pasien yang akan melakukan vaksin.

Berdasarkan penelitian yang dilakukan oleh Sembiring dan Sitorus (2019) mengenai implementasi *socket programming* dalam pembuatan sistem antrian pembayaran di unika dengan metode FIFO, Sembiring dan Sitorus menggunakan metode FIFO untuk sistem antrian pembayaran di unika, sedangkan penelitian penulis menggunakan metode FIFO untuk menghindari antrian pendaftaran vaksin.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang di ambil dari beberapa jurnal terkait penelitian:

II.2.1. Perancangan

Perancangan merupakan sebuah alur atau proses aplikasi macam-macam teknik prinsip dengan tujuan pendefinisian suatu sistem, perangkat, atau proses detail yang mencukupi untuk memungkinkan pengaktualan fisiknya. (Saputri, 2020:42).

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari berbagai elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Pengertian perancangan menurut para ahli diantaranya adalah:

1. Menurut Varzello/John Reuter III perancangan adalah tahap setelah analisis dari siklus pengembang sistem:Pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi:“Mengembarkan bagaimana suatu sistem dibentuk”.

2. Menurut John Buch & Gary Grudnitski perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
3. Menurut George M. Scott perancangan adalah menentukan bagaimana sistem akan menyelesaikan apa yang mesti diselesaikan; tahap ini menyangkut mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. (Hidayatulloh, dkk, 2020:20).

II.2.2. Aplikasi

Aplikasi merupakan penggunaan dalam suatu komputer, instruksi atau pernyataan yang di susun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Aplikasi merupakan perangkat lunak yang dapat menjadi alat bantu untuk mempermudah aktivitas manusia dalam kehidupan setiap hari. Aplikasi memiliki arti suatu Teknik pemrosesan data aplikasi untuk suatu pemecahan masalah yang berdasarkan pada sebuah komputansi yang diharapkan ataupun pemrosesan data yang di harapkan. (Wowiling, dkk, 2020:508).

Aplikasi merupakan suatu kelompok *file* (*class*, *form*, *report*) yang bertujuan untuk melakukan kegiatan tertentu yang saling berhubungan, misalnya seperti *payroll*, aplikasi *fixed asset*, dan yang lainnya. Menurut Kamus Besar Bahasa Indonesia (KBBI), Aplikasi merupakan suatu sistem yang dirancang untuk mengelola data dengan aturan serta ketentuan tertentu dan menggunakan bahasa pemograman tertentu. Program aplikasi adalah program komputer yang ditulis

menggunakan Bahasa pemrograman dan dipergunakan untuk menyelesaikan suatu masalah dan melakukan pekerjaan kebutuhan pengguna. (Solli dan Sompie, 2021:536).

II.2.3. Pendaftaran

Pendaftaran memiliki dua arti, pendaftaran berasal dari kata dasar daftar. Pendaftaran adalah sebuah honomin karena arti-artinya memiliki ejaan dan pelafalan yang sama tetapi maknanya berbeda. Pendaftaran memiliki arti dalam kelas nomina atau kata benda sehingga Pendaftaran dapat menyatakan nama dari seseorang, tempat, atau semua benda dan segala yang dibendakan. (Septiani, dkk, 2021:46).

II.2.4. Vaksin

Vaksin merupakan agen biologis yang memiliki respons imun terhadap antigen spesifik yang berasal dari patogen penyebab penyakit menular. Edward Jenner mengembangkan vaksin pertama pada 1796 yaitu menggunakan cacar sapi untuk diinokulasi terhadap cacar. Hal tersebut pada akhirnya menjadi suatu agen pemberantas cacar secara global, yang secara resmi dinyatakan pada tahun 1980. Sejak itu, vaksin telah membantu menekan penyebaran beberapa penyakit menular termasuk polio. Vaksin merupakan sesuatu yang dianggap sebagai salah satu kemenangan terbesar dalam sejarah kedokteran. Hingga hari ini, seluruh manusia hidup dalam periode pengembangan vaksin yang paling sukses. Vaksin sudah

banyak digunakan untuk mencegah berbagai macam penyakit. (Sari dan Sriwidodo, 2020:206).

II.2.5. Metode *First In First Out* (FIFO)

Metode *First In First Out* (FIFO) merupakan metode dimana barang pertama yang masuk berarti barang tersebutlah yang pertama keluar. Menurut "dengan metode FIFO, biaya persediaan dihitung berdasarkan asumsi bahwa barang akan dijual atau dipaki sendiri dan sisa dalam persediaan menunjukkan pembelian atau produksi yang terakhir". FIFO (*First In First Out*) merupakan algoritma penjadwalan non *preemptive*, tidak berprioritas. Setiap proses diberi jadwal eksekusi berdasarkan urutan waktu kedatangannya. Begitu proses mendapatakan jatah eksekusi maka proses akan dijalankan sampai selesai. (Sembiring dan Sitorus, 2019:29).

Dalam membicarakan sistem antrian ada beberapa karakteristik yang harus ditentukan yaitu:

1. Tingkat kedatangan (λ)

   Yaitu jumlah kendaraan/orang yang datang pada tempat pelayanan untuk dilayani (orang/sat waktu) atau (kend/sat waktu). Tingkat kedatangan bias berpola konstan (*Deterministic*) atau pola kedatangan poisson/eksponensial (acak).

2. Tingkat pelayanan (μ)

   Merupakan jumlah orang /kend yang dapat dilayani pada tempat pelayanan per satuan waktu. Pola tingkat pelayanan sama dengan tingkat kedeatangan.

Rumus:

$$T_{\mu} = \frac{\lambda}{\mu} \quad \text{..... (1)}$$

Keterangan:

T = Waktu Antrian

λ = Tingkat Kedatangan

μ = Tingkat Pelayanan. (Abdullah dan Iswandi, 2019:108).

II.2.6. *Android*

Android Software Development Kit (SDK) adalah sebuah alat untuk para pengembang aplikasi berbasis *Android*. Di dalam *android* SDK terdapat level API yang di dalamnya adalah alat perbaikan kesalahan program, pustaka-pustaka, *emulator*, dokumentasi, kode sumber serta tutorial. Level-level API tersebut adalah inisialisasi dari setiap *platform android* yang berisi paket dalam pengembangan aplikasi *Android*. Dengan adanya *android* SDK ini memudahkan para pengembang untuk menerapkan contoh kode sumber dari contoh aplikasi yang ada pada setiap API. (Maurits, 2020:22).

Android merupakan sistem operasi berbasis linux yang menyediakan akses kepada pengguna untuk menggunakan suatu aplikasi. *Android* memiliki *tools* yang dapat membantu para pengembang pada saat membangun suatu aplikasi. *Android* menyediakan *platform* terbuka bagi pengembang dalam mengembangkan aplikasi android mereka. Berikut ini adalah kelebihan-kelebihan dari sistem operasi *android:*

1. *Android* bersifat *open source*, dengan begitu para pengembang memiliki peluang besar untuk dapat membuat dan mengembangkan aplikasi-aplikasi.
2. Terdapat *Google Play Store* yang menyediakan berbagai macam aplikasi dengan fungsinya masing-masing, sehingga pengguna dapat memilih aplikasi-aplikasi apa saja yang ingin digunakan.
3. Setiap aplikasi berbasis *android* juga dikembangkan secara *up to date*, sehingga banyak fitur-fitur baru yang akan muncul ketika memperbaharui suatu aplikasi tersebut. (Wowiling, dkk, 2021:508).

II.2.7. *Hyper Text Markup Language* (HTML)

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word. (Sugijanto, dkk, 2020:2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML. (Rahmasari, 2019:415).

II.2.8. *Hypertext Preprocessor* (PHP)

PHP merupakan singkatan dari "*Hypertext Preprocessor*". PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrogramn C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberpa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019 :415).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script*/program tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020:2).

II.2.9. MySQL

MySQL adalah *Relational Databases Manajemen System* (RDBMS) yang didistribusikan gratis dibawah lisensi GPL (*General Public Licence*). MySQL sendiri merupakan turunan salah satu konsep utama dalam *database* sejak lama yaitu SQL (*Structured Query Language*), SQL merupakan suatu Bahasa yang dipakai didalam pengambilan data pada *relational database* atau basis data yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan Bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. (Solli dan Somprie, 2021:537).

II.2.10. *Unified Modelling Language* (UML)

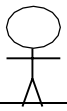
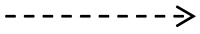
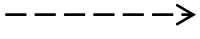
Unified Modelling Language (UML) yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019:39).

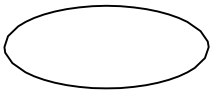
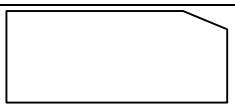
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol *Use Case*

Gambar	Nama	Keterangan
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Depedency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Include*	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikan paket yang menampilkan sistem secara terbatas.

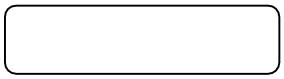
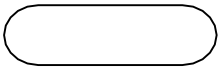
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber:Andikos, 2019:39)

2. Diagram Aktivitas (*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. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

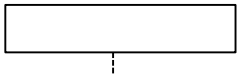
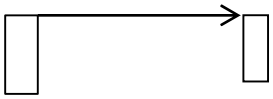
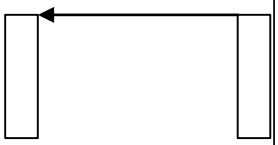
Gambar	Nama	Keterangan
	*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi.
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Activity Final*	Bagaimana objek dibentuk dan dihancurkan
	*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber:Andikos, 2019:39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

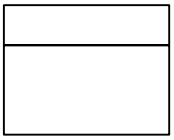
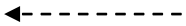
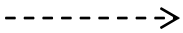
Gambar	Nama	Keterangan
	*Lifeline*	Objek entity, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yangmemuat informasi-informasi tentang aktifitas yang terjadi.
	*Message*	Spesifikasi dari komunikasi antar objek yangmemuat informasi-informasi tentang aktifitas yang terjadi.

(Sumber:Andikos, 2019:39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
	*Depedency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Assocation	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber:Andikos, 2019:39)