

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Apridiansyah dan Fitriani (2021) mengenai Aplikasi Antrian Pembayaran Uang Kuliah Berbasis Android Menggunakan Algoritma FIFO Di Universitas Muhammadiyah Bengkulu, Apridiansyah dan Fitriani menggunakan Metode *First In First Out* (FIFO) dalam pembayaran kuliah, sedangkan penelitian penulis menggunakan metode *First In First Out* (FIFO) untuk pendaftaran siswa.

Berdasarkan penelitian yang dilakukan oleh Fatkhudin dan Alifiani (2017) mengenai Sistem Informasi Pendaftaran Pasien Pada Klinik Dr. Veri Kajian Kabupaten Pekalongan Berbasis Android, Fatkhudin dan Alifiani menggunakan Metode *First In First Out* (FIFO) dalam sistem pendaftaran pasien, sedangkan penelitian penulis menggunakan metode *First In First Out* (FIFO) untuk pendaftaran siswa.

Berdasarkan penelitian yang dilakukan oleh Febriansyah dan Aminah (2021) mengenai Aplikasi Antrian Pada Sekolah Tinggi Teknologi Pagar Alam Berbasis Web Service, Febriansyah dan Aminah menggunakan Metode *First In First Out* (FIFO) dalam sistem antrian pada Sekolah Tinggi Teknologi Pagar Alam, sedangkan

penelitian penulis menggunakan metode *First In First Out* (FIFO) untuk pendaftaran siswa.

Berdasarkan penelitian yang dilakukan oleh Fauzi dan Rahmi (2021) mengenai Penerapan Metode *First In First Out* (FIFO) Dalam Sistem Antrian Pelayanan Administrasi Mahasiswa (Studi Kasus: Daak Universitas Amikom Yogyakarta), Fauzi dan Rahmi menggunakan Metode *First In First Out* (FIFO) Dalam Sistem Antrian Pelayanan Administrasi Mahasiswa, sedangkan penelitian penulis menggunakan metode *First In First Out* (FIFO) untuk pendaftaran siswa.

Berdasarkan penelitian yang dilakukan oleh Novianto (2015) mengenai Model Sistem Antrian FIFO (First-In First-Out) Pada Pelayanan Front Office Kepada Mahasiswa (Studi Kasus : STIKOM Dinamika Bangsa), Novianto menggunakan Metode *First In First Out* (FIFO) untuk sistem antrian pelayanan, sedangkan penelitian penulis menggunakan metode *First In First Out* (FIFO) untuk pendaftaran siswa.

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 *First In First Out* (FIFO) untuk sistem antrian pembayaran, sedangkan penelitian penulis menggunakan metode *First In First Out* (FIFO) untuk pendaftaran siswa.

II.2. Landasan Teori

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

II.2.1. Implementasi

Proses implementasi ada proses penterjemahan dari pemodelan ke dalam pengkodean atau pembentukan antarmuka. Implementasi adalah tindakan–tindakan yang dilakukan oleh individu atau pejabat–pejabat, kelompok–kelompok pemerintah atau swasta yang diarahkan pada terciptanya tujuan–tujuan yang telah digariskan dalam keputusan kebijakan. (Oktaviani dan Sauda, 2019 : 179).

II.2.2. Metode *First In First Out* (FIFO)

Metode *First In First Out* (FIFO) merupakan 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 berurutan dan bergiliran namun tetap pada alur atau jalurnya sesuai dengan yang pertama kali masuk dan kemudian diproses sesuai dengan giliran. (Apridiansyah dan Fitriyani, 2021 : 92).

Dalam membicarakan sistem Pelayanan 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 bisa 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 kedatangan.

Rumus :

$$T = \frac{\lambda}{\mu} \dots\dots\dots(1)$$

Keterangan :

T = Waktu Pelayanan

λ = Tingkat Kedatangan

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

II.2.3. Aplikasi

Aplikasi merupakan suatu program komputer yang dapat digunakan pada berbagai perangkat elektronik, terutama pada komputer/laptop atau telepon seluler yang berbasis android. (Efendi, dkk, 2020 : 341).

Komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. (Nugraha, dkk, 2020 : 138).

II.2.4. Sistem Informasi

Sistem yang berarti suatu kesatuan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu sedangkan informasi merupakan data yang disampaikan, sehingga sistem informasi memiliki arti yaitu suatu cara tertentu untuk menyediakan yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses.

Sistem informasi terdiri dari empat komponen yaitu: aktivitas manajemen, kualitas informasi, struktur keputusan, dan aplikasi sistem informasi. Dari keempat komponen itu terdapat definisi masing-masing yaitu:

1. Aktivitas manajemen sistem informasi terkadang diklasifikasikan berdasarkan aktifitas yang didukungnya. Berdasarkan hal ini, terdapat pengelompokan pada level manajemen. Menurut Ebert dan Griffin dalam Kadir, pengelompokannya mencakup sistem informasi pengetahuan, sistem informasi operasional, sistem informasi taktis, sistem informasi strategis.
2. Kualitas informasi menurut O'Briens, di dalam winda septianita mengatakan "tingkat dimana informasi memiliki karakteristik isi, bentuk, dan waktu, yang memberikannya nilai buat para pemakai akhir tertentu".

3. Aplikasi sistem informasi suatu rangkaian sourcecode, kumpulan tabel di dalam database yang saling berinteraksi digerakkan oleh sebuah intruksi yang berorientasi objek dan memiliki algoritma dalam pengambilan keputusannya, dinyatakan berhasil dan cukup. (Hidayat, dkk, 2020 : 2).

II.2.5. Pendaftaran

Pendaftaran merupakan langkah awal dalam masuk atau bergabung dalam suatu instansi. Selain itu pendaftaran merupakan suatu proses administrasi yang terjadi setiap tahun untuk seleksi peserta didik baru berdasarkan nilai akademik agar dapat melanjutkan pendidikan pada jenjang yang lebih tinggi. (Budiarti dan Risyanto, 2020 : 2).

II.2.6. Siswa

Pengertian siswa merupakan pelajar yang duduk dimeja belajar setrata sekolah dasar maupun menengah pertama (SMP), sekolah menengah keatas (SMA). Siswa-siswa tersebut belajar untuk mendapatkan ilmu pengetahuan dan untuk mencapai pemahaman ilmu yang telah didapat dunia pendidikan. Siswa atau pesetra didik adalah mereka yang secara khusus diserahkan oleh kedua orang tuanya untuk mengikuti pembelajaran yang diselenggarakan di sekolah, dengan tujuan untuk menjadi manusia yang berilmu pengetahuan, berketrampilan, berpengalaman, berkepribadian, berakhlak mulia, dan mandiri. (Merpati, dkk, 2018 : 57).

II.2.7. Android

Android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam peranti bergerak. Pemrograman Android membutuhkan Android SDK (Software Development Kit) dan juga Java Development Kit. Setelah itu dibutuhkan Java IDE yang digunakan untuk menuliskan coding Android. Salah satu tools yang dapat digunakan dalam pemrograman Android adalah Eclipse Helios karena Eclipse sangat mendukung proses penulisan code Android dan menjadikan proses development Android menjadi lebih mudah. (Efendi, dkk, 2020 : 334).

Dalam perjalanannya Android telah merilis banyak versi tentunya dengan berbagai macam fitur dan perbaikan di tiap versi yang baru. Berikut versi-versi Android yang telah dirilis hingga saat ini :

1. Android 1.1
2. Android 1.5 (Cupcake)
3. Android 1.6 (Donut)
4. Android 2.0-2.1 (Eclair)
5. Android 2.2.3 (Froyo)
6. Android 2.3-2.3.7 (Gingerbread)
7. Android 3.0-3.2.6 (Honeycomb)
8. Android 4.0-4.0.4 (Ice Cream Sandwich)
9. Android 4.1-4.3.1 (Jelly Bean)

10. Android 4.4 (KitKat)
11. Android 5.0 (Lollipop)
12. Android 6.0 (Marshmallow)
13. Android 7.0 (Nougat)
14. Android 8.0 (Oreo)
15. Android 9.0 (Pie). (Nugraha, 2020 : 138).

II.2.8. PHP

Hypertext Preprocessor atau yang sering disebut dengan PHP merupakan bahasa pemrograman yang bersifat *Open Source*. Bahasa pemrograman ini bersifat *server side*, maksudnya adalah tanpa adanya server yang berjalan beriringan dengan program atau aplikasi *script* PHP tidak dapat berjalan. PHP merupakan *script* yang disisipkan dalam *code* HTML untuk membuat halaman pada *website* bekerja secara otomatis dan dapat berfungsi sebagai pengolah data pada *server* dimana *script* tersebut dijalankan. PHP memiliki kemampuan untuk memisahkan diri dari kode HTML saat sebuah website dilakukan “*View Page Source*” oleh *user*. (Septyasari, dkk, 2018 : 215).

PHP adalah singkatan dari *PHP Hypertext Preprocessor*. PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML (*Hypertext Markup Language*) untuk membuat halaman *web* yang dinamis. Maksud dari *server-side scripting* adalah sinaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di *server*

tetapi disertakan pada dokumen HTML sebagai pembangun halaman *web*. (Kosasih, dkk, 2017 : 115).

II.2.9. Basis Data

Basis data merupakan kumpulan beberapa data yang saling berelasi antara satu dengan yang lain sehingga data tersebut dapat dimanipulasi, ditampilkandan dicari dengan cepat. Data merupakan fakta mengenai objek, orang dan lainnya yang dinyatakan dengan nilai angka, karakter maupun simbol. Selain berisi sebuah data, basis data juga berisi metadata. Model basis data relasional merupakan suatu cara untuk merepresentasikan model data dalam perancangan basis data dimana model dari basis data relasional didasarkan pada *record*. Proses perancangan basis data terdiri tiga tahapan yaitu perancangan konseptual, perancangan logikal dan perancangan fisik. Perancangan konseptual adalah proses membangun model data yang digunakan dalam suatu perusahaan. Perancangan logikal adalah proses merancang model data yang digunakan dalam suatu perusahaan berdasarkan pada model data yang spesifik. Perancangan fisik adalah proses menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder, menggambarkan hubungan dasardan organisasi file yang digunakan untuk mencapai akses yang efisien pada data. (Wijaya, dkk, 2021 : 72).

Basis Data (*database*) terbentuk dari beberapa komponen. Berikut adalah komponen-komponen pembentuk *database* :

1. Tabel adalah sekumpulan data dengan struktur yang sedemikian rupa, terbentuk dari *record* dan *field*. Istilah tabel disini berbeda dengan istilah tabel pada HTML, walaupun secara visual hampir sama.
2. *Record* adalah sekumpulan *field* yang membentuk suatu objek tertentu.
3. *Field* adalah atribut dari objek yang memiliki tipe data tertentu. (Vambudi, 2021 : 26).

II.2.9.1. Normalisasi

Normalisasi merupakan teknik yang digunakan untuk memvalidasi model data. Serangkaian aturan diberlakukan pada data model logis untuk meningkatkan pengaturannya.

1. Bentuk tidak normal, bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.
2. Bentuk normal tahap pertama (1NF), sebuah tabel disebut 1NF jika tidak ada baris yang duplikat dalam tabel tersebut dan setiap cell bernilai tunggal.
3. Bentuk normal tahap kedua (2NF), bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.
4. Bentuk normal tahap ketiga (3NF)
5. Bentuk normal tahap keempat dan kelima. (Vambudi, 2021 : 26).

II.2.11. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instansi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Kosasih, dkk, 2017 : 115).

MySQL merupakan turunan dari *Structured Query Language* (SQL), SQL merupakan operasional basisdata yang dapat dikerjakan dengan mudah dan otomatis. MySQL bertugas mengatur dan mengelola data-data pada database. MySQL merupakan perangkat lunak relasi yang dapat digunakan secara gratis. MySQL adalah perangkat lunak sistem manajemen basisdata pada *Structured Query Language* (SQL) yang *multiuser*. MySQL merupakan manajemen basis data yang paling stabil digunakan sebagai media penyimpanan data. (Oktasari dan Kurniadi, 2019 : 150).

II.2.12. Unified Modeling Language (UML)

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 system yang besar dan kompleks. UML tidak hanya digunakan

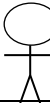
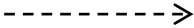

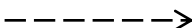

dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).


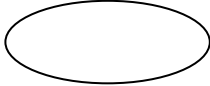


Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</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>).
	<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>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara




		objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	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
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

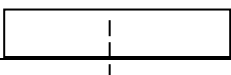
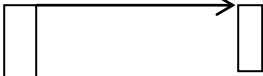
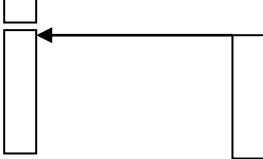
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(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*



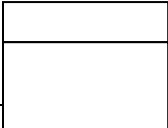

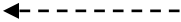
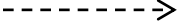

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang 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
	<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>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Andikos, 2019 : 39)