

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Laboratorium**

komputer Dalam jurnal Frizal Luthfi Hadyan tahun 2014, Mengatakan bahwa Laboratorium komputer merupakan sarana yang sangat dibutuhkan pada lingkungan perkantoran maupun pendidikan. Setiap kegiatan yang dilakukan pada laboratorium komputer harus mendapat pengawasan dari penanggung jawab laboratorium komputer itu sendiri, karena sangat penting untuk pendataan penggunaan laboratorium serta kapan saja laboratorium digunakan, laboratorium komputer juga membutuhkan suatu sistem informasi yang memudahkan proses kegiatan yang dilaksanakan pada laboratorium komputer, baik itu dalam pendidikan maupun perkantoran.

#### **2.2. Definisi Aplikasi *Mobile***

*Mobile* aplikasi adalah program yang digunakan orang untuk melakukan sesuatu dari satu tempat ke tempat yang lain, misalnya *telepon mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. sistem Aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat *nirkabel* seperti *leptop*, telepon seluler.

#### **2.3. Definisi *Website***

*Website* adalah keseluruhan halaman - halaman web yang terdapat dari sebuah *domain* yang mengandung informasi. *Domain* adalah nama unik yang dimiliki oleh sebuah institusi sehingga bisa di akses melalui *internet*, misalnya: *ephi.id*,

*yahoo.com, google.com* dan lain - lain. Untuk mendapat sebuah domain harus disewa melalui register - register yang ditentukan, (Agustina Simangunsong 2018).

## **2.4. Perangkat Lunak Pengembangan Sistem**

Pengembangan sistem untuk membangun aplikasi pemesanan berbasis *mobile* diperlukan beberapa perangkat lunak yang digunakan dalam membangun aplikasi tersebut. Beberapa perangkat lunak yang digunakan adalah sebagai berikut :

### **2.4.1. Sublime text**

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi *Python API*. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. Aplikasi Sublime merupakan suatu aplikasi text editor yang sangat berguna untuk menulis sejumlah *code* serta mampu membuka berbagai macam jenis file. Selain itu, Sublime Text juga mendukung berbagai bahasa pemrograman seperti C++, C, C#, CSS, ASP, html, dan sebagainya.(idcloudhost.com)

### **2.4.2. MySQL**

Dalam (Solichin, A. 2016) menjelaskan bahwa *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. Selain itu *MySQL software* merupakan suatu aplikasi yang sifatnya *open source* serta *server* basis data *MySQL* memiliki kinerja sangat cepat, *reliable* dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*.

### **2.4.3. Adobe XD**

*Adobe XD* adalah perangkat lunak yang bisa digunakan oleh para desainer aplikasi *mobile*. *Adobe xd* bisa memudahkan desainer aplikasi *mobile* dalam

pengembangan *UX/UI*, *adobe xd* ini sudah menyediakan fitur *UI* Desain dan juga *UX* Desain sebagai *prototype* tanpa membutuhkan *third-party* atau aplikasi lain untuk membantu membuat sebuah *prototype*. ([garudapixel.com](http://garudapixel.com), [dewaweb.com](http://dewaweb.com)).

#### **2.4.4. PHP**

*Hypertext Preprocessor* (PHP) PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada *HTML*. PHP merupakan singkatan dari “*PHP Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen *HTML* sekaligus bekerja disisi *server* (*server-side HTML-embedded scripting*) (West, 2013).

#### **2.4.5. JavaScript**

*JavaScript* merupakan modifikasi dari bahasa *c++* dengan pola penulisan yang lebih sederhana (Sidik & Pohan, 2010).

#### **2.4.6. Bootstrap**

*Bootstrap* adalah kerangka ramping, intuitif, dan kuat, *framework front-end mobile* pertama untuk mempercepat dan memudahkan pengembangan *web*. *Bootstrap* menggunakan *HTML*, *CSS*, dan *Javascript*. *Bootstrap* dikembangkan oleh Mark Otto dan Jacob Thornt di Twitter ([Tutorialspoint.com](http://Tutorialspoint.com), 2014).

### **2.5. Multilevel Feedback Queue.**

Dalam (Verawati, I., & Sulistiyono, M. 2018) menjelaskan bahwa *Multilevel feedback queue* adalah salah satu algoritma yang berdasar pada model antrian *Multi Channel Single Server*. Kelebihan mendasar yang dimiliki oleh *multilevel feedback queue* adalah kemungkinan adanya suatu proses berpindah dari satu antrian ke antrian lainnya, misalnya dengan prioritas yang lebih rendah ataupun lebih tinggi. *Algoritma* ini didefinisikan melalui beberapa parameter, antara lain :

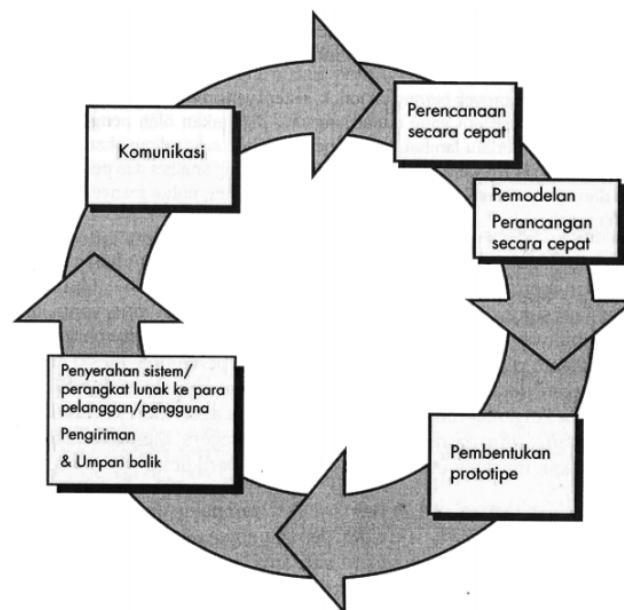
1. Jumlah antrian.
2. *Algoritma* penjadwalan tiap antrian.

3. Kapan menaikkan proses ke antrian yang lebih tinggi.
4. Kapan menurunkan proses ke antrian yang lebih rendah.
5. Antrian mana yang akan dimasuki proses yang membutuhkan.

*Algoritma* ini biasa digunakan dalam pengembangan *Operating System*. Terutama dalam penjadwalan *CPU*. Jika suatu proses menyita *CPU* terlalu lama, maka proses itu akan dipindahkan ke antrian yang lebih rendah. Hal ini menguntungkan proses interaksi karena proses ini hanya memakai waktu *CPU* yang sedikit. Demikian pula dengan proses yang menunggu terlalu lama. Proses ini akan dinaikkan tingkatannya. Ada tiga jenis kategori yang digunakan dalam *multilevel feedback queue* ini, yaitu *High Priority*, *Normal Priority* dan *Low Priority*

## 2.6. Metode Pengembangan Perangkat Lunak

Pressman (2012) menjelaskan meskipun pembuatan *prototype* dapat di gunakan sebagai *model* proses yang berdiri sendiri, pembuatan *prototype* lebih umum digunakan sebagai teknik yang dapat diimplementasikan di dalam konteks setiap model proses perangkat lunak, dan paradigma pembuatan *prototype* seringkali membantu tim pengembang perangkat lunak dan para *stakeholder* untuk memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas. Berikut ini adalah penjelasan dari gambar 2.2



**Gambar 2.1** Tahapan Pengembangan Aplikasi

1. Komunikasi tahap komunikasi ini adalah tahapan komunikasi antara *developer* dan pelanggan mengenai tujuan pembuatan *software*, mengidentifikasi apakah kebutuhan diketahui.
2. Perancangan secara cepat tahap perancangan secara cepat ini adalah tahapan perancangan cepat setelah terjalin komunikasi.
3. Pemodelan perancangan secara cepat tahap pemodelan perancangan secara cepat ini adalah tahapan segera membuat model, dan pemodelan cepat fokus pada gambaran dari segi *software* apakah *visible* menurut pelanggan.
4. Pembentukan *prototype* tahap pembentukan prototipe ini adalah tahapan pemodelan cepat menuntun pada pembuatan dari *prototype*.
5. Penyerahan sistem / perangkat lunak ke para pengguna / pengguna pengiriman dan umpan balik tahap penyerahan sistem / perangkat lunak ke para pengguna / pengguna pengiriman dan umpan balik tahapan ini adalah *prototype* yang dikirimkan kemudian dievaluasi oleh pengguna.

## **2.7. Pengujian Kotak Hitam**

Pressman (2012) menjelaskan bahwa pengujian kotak hitam, juga disebut pengujian perilaku, yang berfokus pada persyaratan fungsional perangkat lunak. Teknik Pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut :

1. fungsi yang salah atau hilang,
2. kesalahan antarmuka,
3. kesalahan dalam struktur data atau akses basis data eksternal,
4. kesalahan perilaku atau kinerja, dan
5. kesalahan inisialisasi dan penghentian.

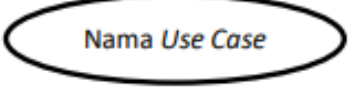




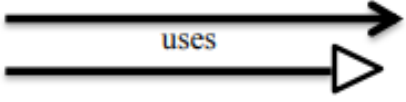
## **2.8. *Unifed Modelling language (UML)***

Dalam (Marini, M. 2019) menjelaskan bahwa *Unified Modeling Language (UML)* merupakan sebuah bahasa pemodelan objek standar sebagai ganti dari pendekatan atau metode berorientasi objek standar. UML adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. UML dapat digunakan untuk mempermudah pengembangan Aplikasi yang berkelanjutan.

### **2.8.1 *Use Case Diagram***

*Use case* atau *use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut *aktor* dan *use case*. *Aktor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi *aktor* belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *aktor* (Rosa dan Shalahuddin, 2013).

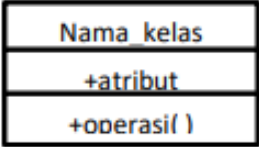




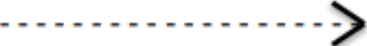

Tabel 2.1 Simbol Use Case Diagram

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit dan aktor.</p>
<p>Aktor / Actor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.</p>
<p>Asosiasi / Association</p> 	<p>Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi.</p>
<p>Ekstensi / extend</p> <p>&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan.</p>
<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.</p>
<p>Include / Use Case</p> <p>&lt;&lt;include&gt;&gt;</p> <p>uses</p> 	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya.</p>

### 2.8.2 Activity Diagram

Dalam (Marini, M. 2019) menjelaskan bahwa *Activity* diagram menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah *system* atau proses bisnis yang ada pada menu perangkat lunak. Tahap perancangan *activity* diagram menjabarkan masing – masing *activity* pada perancangan use case. Simbol – simbol *Activity* Diagram dapat dilihat pada tabel 2.2.

Tabel 2.3 Simbol *Activity* Diagram

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur system.
<p>Antarmuka / <i>interface</i></p>  <p>Nama_ <i>interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
<p>Asosisasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosia biasanya disertai dengan <i>multiplicity</i> .
<p>Asosisasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>generalisasi</p> 	Relasi antarkelas dengan makna Generalisasi – spesialisasi (umum khusus).
<p>Kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua – bagian

## 2.9 Penelitian Terdahulu

Berikut merupakan jurnal terkait dengan penelitian terdahulu :

NO	NAMA	JUDUL	TERBIT / TAHUN	KETERANGAN
1	Tri Wahyu Prasetyo, Wiharto & Afrizal Doewes	Pemodelan Penjadwalan <i>Multilevel Feedback Queue</i> Menggunakan <i>Dynamic Time Quantum</i> Pada Kasus Pemesanan	JURNAL ITSMART Desember 2015	Penelitian ini membahas mengenai sistem penjadwalan memasak yang efisien dengan mengunakan algoritma <i>Multilevel Feedback Queue</i> Menggunakan <i>Dynamic Time Quantum</i> . Algoritma ini



		Makanan di Restoran administrasi (Customer dan Admin)		menjadwalkan pesanan dengan cara mengelompokkan pesanan berdasarkan lama waktu memasaknya ke dalam queue sesuai dengan besar kecilnya time quantum queue. Kemudian dilakukan penjadwalan sesuai algoritma pada masing-masing queue secara berurutan mulai dari queue dengan prioritas tertinggi ke prioritas yang lebih rendah
2	Adrianus, Dian Andrian Ginting	Implementasi <i>Algoritma Multilevel Feedback Queue</i> Untuk Pembuatan Aplikasi Pemesanan Makanan Pada Restoran Dengan Platform <i>Android</i> Dan <i>IOS</i>	Universitas Lampung / 2015	Penelitian ini membahas tentang pembuatan aplikasi web pemesanan makanan pada <i>Android</i> dan <i>iOS</i> dengan menerapkan metode <i>Multilevel Feedback Queue</i> untuk membuat antrian pemesanan makanan pada koki
3	Isnawaty, Muthmainah Liyana & Subardin	Rancang Bangun Aplikasi penjadwalan Ruang Meeting hotel menggunakan <i>Algoritma Multiple Feedback Queue (Mfq)</i> Berbasis android Menggunakan Layanan Sms “Studi Kasus Hotel Plaza Inn Kendari”	semanTIK, 2, Jul-Des 2016	Penelitian ini membahas mengenai Manajemen rapat adalah merencanakan, mengorganisasikan, memimpin, dan mengontrol rapat yang menyebarkan jadwal meeting yang bertabrakan sehingga petugas hotel harus memeriksa semua jadwal ruangan apabila ada pemesanan ruangan meeting yang baru ataupun adanya perubahan jadwal meeting yang sudah terinput sebelumnya, hal ini menjadi tidak efisien dari segi waktu. Sistem penjadwalan ruang meeting hotel dengan menerapkan metode <i>Multiple Feedback Queue (MFQ)</i> memiliki sistem kerja menggunakan smartphone <i>Android</i> . Penerapan metode <i>MFQ</i> membuat aplikasi penjadwalan ruang meeting dapat terdistribusi dengan baik