

## BAB II TINJAUAN PUSTAKA

### 2.1 Tinjauan Literature

Tinjauan pustaka atau tinjauan literature dilakukan untuk mengetahui perbedaan penelitian yang dilakukan dengan penelitian sebelumnya yang terkait dengan penelitian ini. Ada beberapa penelitian yang berkaitan dengan penelitian ini ditunjukkan pada Tabel 2.1.

**Tabel 2.1** Tinjauan Literature

No	Penulis	Tahun	Judul Penelitian
1	Deni Utama, Asahar Johar, Funny Farady Coastera	2016	Aplikasi Pemesanan Makanan dan Minuman Restaurant Berbasis Client Server Dengan Platform Android dan PHP Menggunakan Metode <i>Centralized</i> DBMS <i>Architecture</i> (Studi Kasus:Café Cempakoe Kota Bengkulu)
2	Busran, Wina Anggraini	2016	Perancangan Aplikasi Pemesanan Makanan dan Minuman Berbasis Sistem Operasi Android (Studi Kasus : Pecel Lele Lela)

Penelitian yang dilakukan oleh Deni Utama, Asahar Johar, Funny Farady Coastera (2016) dari Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu. Judul penelitian “Aplikasi Pemesanan Makanan dan Minuman Restaurant Berbasis Client Server Dengan Platform Android dan PHP Menggunakan Metode *Centralized* DBMS *Architecture* (Studi Kasus: Café Cempakoe

Kota Bengkulu)”. Penelitian ini mengangkat permasalahan pada usaha restoran dalam praktek pemesanan makanan dan minuman masih menggunakan cara manual, dimana pelayan menghampiri pengunjung dan mencatat pesanan pada sebuah kertas yang nantinya. Diserahkan ke dapur dan kasir. hal ini kemungkinan untuk terjadi kesalahan penulisan pemesanan yang rangkap, selain itu dengan proses pemesanan manual seperti ini sering terjadi kesalahan dalam urutan pemesanan akibat bertumpuknya nota pemesanan, hal tersebut berdampak negatif pada kepuasan pelanggan. Hasil penelitian ini yaitu sebuah aplikasi pemesanan makanan dan minuman restaurant berbasis client server dengan platform android dan PHP menggunakan metode *Centralized DBMS Architecture*. Berdasarkan hasil uji coba yang telah dilakukan aplikasi dapat memotong waktu pemesanan 5 jam 49 menit 55 detik dari total semua waktu pemesanan menggunakan kertas.

Penelitian yang dilakukan oleh Busran, Wina Anggraini (2016) dari Institut Teknologi Padang. Judul penelitian “Perancangan Aplikasi Pemesanan Makanan dan Minuman Berbasis Sistem Operasi Android (Studi Kasus: Pecel Lele Lela)”. Penelitian didasarkan dalam bisnis rumah makan cepat saji pecel lele Lela pelayanan sangat diutamakan sekali demi kepuasan pelanggan. Salah satu proses pelayanan yang sangat mempengaruhi adalah proses pelayanan dalam melakukan pemesanan menu makanan dan minuman. Proses pelayanan yang baik dan nyaman bisa membuat daya tarik pada pelanggan. Hal ini mendorong pemilik perusahaan untuk meningkatkan proses pelayanan kearah yang lebih baik. Sehingga pada penelitian ini dirancang sebuah aplikasi pemesanan makanan dan minuman berbasis sistem operasi android. Hasil dalam penelitian ini yaitu sebuah aplikasi pemesanan makanan dan minuman berbasis android, aplikasi ini mempermudah pelayanan

pada restaurant pecel lele lela. Aplikasi ini memiliki *user interface* yang menarik sehingga akan membantu pengguna dalam menggunakan aplikasi.

Berdasarkan uraian ke dua tinjauan literature diatas, dapat disimpulkan bahwa layanan pemesanan makanan dapat dilakukan secara *online* dapat mempermudah pengguna untuk melakukan proses pemesanan lebih cepat dan lebih efektif dan mempermudah pelanggan dalam menentukan pilihan menu makanan yang di sediakan.

## **2.2. Android**

Menurut (Hansun, Kristanda, & Saputra, 2018) android menawarkan pendekatan yang menyeluruh dalam pengembangan aplikasi. Artinya, satu aplikasi android yang dibangun dapat berjalan di berbagai perangkat yang menggunakan sistem operasi android baik itu *smartphone, smartwatch, tablet*, dan perangkat lainnya. Perkembangan teknologi android yang begitu pesat juga tidak dapat dilepas dari peran AOSP (*Android Open Source Project*) yang bertanggung jawab dalam pengembangan sistem operasi android dan dipimpin langsung oleh *google*. Android telah mengalami perkembangan yang begitu pesat, ditandai dengan banyaknya versi sistem operasi android ini. Pada tabel 2.4 menunjukkan beberapa versi berserta tanggal rilisnya.

## **2.3. Unified Modelling Language (UML)**

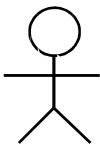
Menurut (A.S. and Shalahuddin, 2015) UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group (OMG)*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. UML terbaru adalah UML 2.3



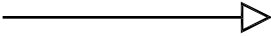

yang terdiri dari 4 macam spesifikasi, yaitu diagram *interchange specification*, *UML infrastructure*, *UML Superstructure*, dan objek *constraint language*.

### 1.3.1. Use Case Diagram

Menurut (A.S. and Shalahuddin, 2015) “*Use Case Diagram* merupakan pemodelan untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara besar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol-simbol *use case* ditunjukkan pada Tabel 2.2.

**Tabel 2.2** Simbol-Simbol *Use Case Diagram*

Simbol	Keterangan
<p><i>Use Case</i></p> <p>Nama <i>use case</i></p>	<p>Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antara unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal.</p>
<p>Aktor / <i>actor</i></p> 	<p>Orang, aktor, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

<p>Ekstensi / <i>extend</i></p> <p style="text-align: center;">&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 ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu, biasanya <i>Use Case</i> tambahan memiliki nama depan yang sama dengan <i>Use Case</i> yang ditambahkan, arah panah menuju pada <i>use case</i> yang dituju.</p>
<p><i>Include</i></p> <p style="text-align: center;">&lt;&lt;include&gt;&gt;</p> 	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> di mana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini.</p>
<p>Generalisasi</p> 	<p>Relasi antara kelas dengan makna generalisasi-generalisasi (umum khusus)</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan</p>

Sumber : (A.S. and Shalahuddin, 2015)

### 2.3.2. *Class Diagram*

Menurut (A.S. and Shalahuddin, 2015) *Class* diagram merupakan gambaran dari stuktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

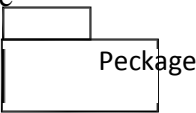
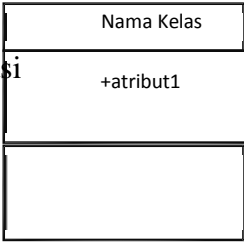



Kelas main, yaitu kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

- 1) Kelas yang menangani tampilan sistem, yaitu kelas yang mendefinisikan dan mengatur tampilan ke pemakai.




- 2) Kelas yang diambil dari pendefinisian *usecase*, yaitu kelas yang menangani fungsi yang harus ada diambil dari pendefinisian *usecase*.
- 3) Kelas yang diambil dari pendefinisian data, yaitu kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Simbol-simbol bentuk *class diagram* ditunjukkan pada Tabel 2.4.

**Tabel 2.3** Simbol-Simbol *Clas Diagram*

Simbol	Keterangan
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas.
Operasi 	Kelas pada struktur system
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>Association</i> 	Relasi antar kelas dengan makna umum, asosiasi juga disertai dengan <i>multiplicity</i>
Simbol	Keterangan
Kebergantungan / <i>depend</i> <i>ency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.

**Tabel 2.4** Simbol-Simbol *Clas* Diagram (lanjutan)

Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )
Asosiasi berarah / <i>Directed Assosiation</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antara kelas dengan makna generalisasi-generalisasi (umum khusus)

Sumber : (A.S. and Shalahuddin, 2015)

### 2.3.3. *Activity Diagram*



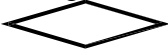



Menurut (A.S. and Shalahuddin, 2015) *Activity Diagram* adalah diagram aktivitas yang menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- 1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- 2) Urutan atau pengelempokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

Simbol-simbol model *activity diagram* ditunjukkan pada Tabel 2.5.

**Tabel 2.5** Simbol-Simbol *Activity*  
Diagram

<b>Simbol</b>	<b>Keterangan</b>
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan Diana lebih dari satu aktivitas digambarkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : (A.S. and Shalahuddin, 2015)

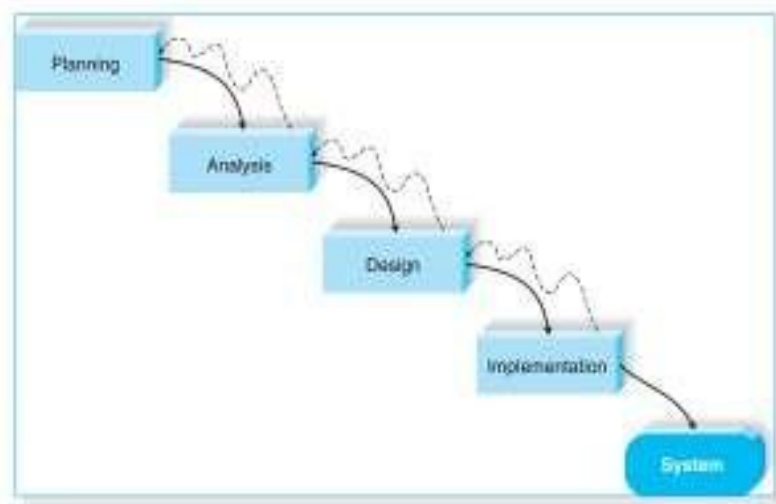
#### **2.4. Metode Pengembangan Sistem**

SDLC (*Software Development Life Cycle*) merupakan proses mengembangkan atau mengubah sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem (A.S. and Shalahuddin, 2015). Metode pengembangan SDLC air terjun (*waterfall*) menyediakan



pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung. Berikut adalah gambar metode *waterfall* dapat dilihat pada gambar dibawah ini :

Sumber : (Dennis, Wixom and Roth, 2012)



**Gambar 2.1** Sistem Model *Waterfall*

Uraian tahapan dalam model pengembangan sistem *waterfall* :

1. *Planning* (Perencanaan)

Dalam tahapan ini menjelaskan dan mengargumentasikan untuk melanjutkan proyek yang telah dipilih. Rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Pada tahap ini ditentukan secara detail rencana kerja yang harus dikerjakan, waktu yang diperlukan masing-masing tahap, sumber daya manusia, perangkat lunak, dokumentasi, perangkat keras. Kesalahan dalam tahap ini akan mengakibatkan keuntungan yang diperoleh tidak maksimal, bahkan bisa rugi.

2. *Analysis* (Analisis)

Tahap analisis, yaitu tahap berusaha mengenali permasalahan yang muncul pada pengguna dengan mendekomposisi dan

merealisasikan komponen-komponen sistem. Tujuan utama tahap analisis ini adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru. Menganalisa kebutuhan sistem sebagai bahan dalam membuat spesifikasi di tahap selanjutnya.

### 3. *Design* (Perancangan)

Tahap perancangan merupakan tahap mencari solusi permasalahan yang dibuat dari tahap analisis. Tahap mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang nyata untuk diimplementasikan.

### 4. *Implementation* (Implementasi)

Tahap implementasi, dimana kita mengimplementasikan perancangan sistem kesituasi nyata. Dimulai dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (pengkodean/*coding*).

### 5. *System* (Sistem)

Tahap ini dilakukan pengujian (*testing*) dan pemeliharaan, yang dapat dilakukan untuk menentukan apakah sistem/perangkat lunak yang dibuat sesuai dengan kebutuhan pengguna atau belum.

## 2.5. **Web Service**

Web Service dapat digambarkan sebagai sebuah sistem yang dirancang untuk dapat mendukung interaksi komunikasi antar mesin-mesin pada suatu jaringan. Teknologi web service memungkinkan pengguna untuk dapat menghubungkan berbagai jenis *software* yang dimiliki platform dan sistem operasi yang sama atau berbeda sekalipun. Jadi pengguna tetap mendapatkan sebuah

informasi dari suatu website tanpa harus mengunjungi website tersebut, cukup dengan mengetahui fungsi/method web service yang disediakan oleh web itu (Sibagariang, 2016).

## **2.6. MYSQL**

MySQL merupakan *software* RDBMS (atau server *database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*). Saat ini, MySQL banyak digunakan berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai keindustri, baik industry kecil, menengah, maupun besar. Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk open source dibawa GNU *General Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik (Raharjo, 2015).

## **2.7. PHP**

*Hypertext Preprocessor* (PHP) merupakan aplikasi perangkat lunak *opensource*, dimana kepanjangan dari PHP adalah *Hypertext Preprocessor* yang diatur dalam aturan *general Purpose licences* (GPL). Pemrograman PHP merupakan pemrograman yang sangat cocok di kembangkan dilingkungan web karena bisa diletakkan pada *script* HTML ataupun sebaliknya. PHP tergolong sebagai pemrograman dinamis karena mampu menghasilkan *website* yang bisa di ubah secara terus menerus hasilnya atau Kontennya tanpa harus masuk kedalam *coding*.

Hal tersebut bergantung pada permintaan terkini. Secara umum, pembuatan *database* sangat erat hubungannya untuk pembuatan web dinamis, sebagai tempat untuk sumber data yang akan di tampilkan. PHP juga merupakan bahasa pemrograman berbasis *server*. Ini berarti setiap pemrograman PHP harus diletakkan di *server* terlebih dahulu, kemudian diterjemahkan oleh web *server* dan hasilnya dikirim ke *browser client*. Kemampuan dari fitur PHP yang paling mendukung banyak basis data yaitu MSSQL, MySQL, Oracle, dan Postgre SQL. Bulan juni 2014, PHP 5.0 di liris, yang dalam versi ini sudah ditanamkan model pemrograman berorientasi objek, yang merupakan model yang banyak digunakan di semua bahasa pemrograman. Secara teknologi, bahasa pemrograman PHP sangat mirip dengan bahasa pemrograman yang berbasis web lain, contohnya bahasa ASP (*Active Server Page*), *Cold Fusion*, JSP (*Java Server Page*), ataupun perl (Buana, 2014).

## 2.8 **Metode First In First Out (FIFO)**

Menurut (Jusup, 2005) metode FIFO atau masuk pertama keluar pertama mengasumsikan bahwa barang yang dibeli awal dianggap akan lebih awal dijual atau digunakan, dan harga pokok perolehan barang yang dibeli lebih awal akan dibebankan lebih dahulu sebagai harga pokok penjualan. Pada pencatatan secara fisik, metode ini beranggapan bahwa barang yang ada paling awal dianggap dijual paling awal juga. Perbedaannya adalah dalam pencatatan secara perpetual dengan metode FIFO (First In First Out), perhitungan harga pokok yang dijual dilakukan pada saat terjadi penjualan. Keuntungannya adalah persediaan akhir mendekati harga pokok berjalan atau sekarang, karena barang yang masuk pertama merupakan barang yang pertama keluar, jumlah persediaan akhir terdiri dari pembelian yang paling baru. Ini

khususnya benar apabila perputaran persediaan sangat cepat. Sedangkan kelemahannya adalah harga pokok berjalan tidak sesuai dengan pendapatan yang berjalan pada laba rugi. Harga pokok yang paling lama dibebankan pada harga pokok yang lebih baru, yang menyebabkan menghasilkan harga pokok dan laba semu.

Dalam penerapan metode FIFO, untuk menentukan harga pokok digunakan asumsi atas arus biaya dimana urutan biaya terjadi adalah yang pertama masuk atau diperoleh, yang pertama keluar atau dijual. Menurut (Baridwan, 2004), harga pokok persediaan akan dibebankan sesuai dengan urutan terjadinya. Apabila ada penjualan atau pemakaian barang-barang maka harga pokok yang dibebankan adalah harga pokok yang paling terdahulu, disusul yang masuk berikutnya. Persediaan akhir dibebani harga pokok terakhir.