

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

Dalam jurnal penelitian terdahulu dengan judul Perancangan Aplikasi M-Commerce Berbasis Android Sebagai Media Informasi dan Penjualan Kambing. menghasilkan rancangan aplikasi yang digunakan untuk mempermudah mendapatkan informasi dan penjualan kambing yang dapat memepertemukan antara pembeli dan peternak kambing. (Retno Waluyo, dkk., 2018).

Dalam penelitian yang dilakukan oleh Muhammad Faisal Widad (2018) dengan judul “system informasi penjualan berbasis android pada toko busana faisalcolection (sipiton)” Penelitian ini bertujuan untuk membuat sebuah sistem informasi penjualan berbasis android yang handal dan efisien dengan menggunakan bahasa pemrograman java dan bahasa pemograman lainnya yang menunjang dalam perancangan sistem ini dan dapat terintegrasi dengan database sehingga lebih mudah untuk diterapkan

Dalam penelitian yang dilakukan oleh Amin Mahmud, dkk (2018), dengan judul “Aplikasi pemesanan menu berbasis Mobile (Studi kasus: rumah makan lulari brebes) “menghasilkan rancangan aplikasi berbasis Mobile yang dapat meminimalisir antrian pembeli.

Dalam penelitian yang dilakukan oleh Abdi Pandu Kusuma, dkk (2017) dengan judul “perancangan dan implementasi E-commerce untuk penjualan baju online berbasis Android “penelitian ini dilakukan untuk perancangan dan pembangunan aplikasi berbasis mobile serta pengujian dengan menggunakan metode white box. Metode ini diimplementasikan e-commerce ke dalam sebuah aplikasi penjualan baju berbasis mobile.

Dalam penelitian terdahulu yang dilakukan oleh Sonita, dkk (2018) “Aplikasi *e-order* menggunakan *fire base* dan algoritma *knuth morris pratt* berbasis Android “Penelitian ini bertujuan untuk memudahkan suatu pengguna dalam melakukan pemesanan makanan dan meningkatkan penjualan bagi pengusaha tersebut dengan menggunakan teknologi firebase.

Dalam penelitian terdahulu yang dilakukan oleh Rahmat Fadli Isnanto, dkk (2019) “Rancang Bangun Aplikasi M-Commerce Berbasis Android Sebagai Media Pemesanan Pada Distro Online “Penelitian ini bertujuan untuk Mobile Commerce ini dapat mempermudah pelanggan untuk melakukan pemesanan secara online dan mempermudah mendapatkan informasi mengenai distro online tersebut, baik letak maupun no telp yang dapat di hubungi.

Dalam penelitian terdahulu yang dilakukan oleh Adam Surya Wijaya,dkk (2021) “Rancang Bangun Aplikasi M-Commerce Berbasis Android Sebagai Media Pemesanan Pada Distro Online “ Penelitian ini bertujuan untuk Mobile Commerce ini dapat mempermudah pelanggan untuk melakukan pemesanan secara online dan mempermudah mendapatkan informasi mengenai distro online tersebut, baik letak maupun no telp yang dapat di hubungi aplikasi e-commerce pada UD Hoky Celluler Shop dapat mengatasi masalah dan juga memenuhi kebutuhan yang dibutuhkan oleh UD Hoky Celluler Shop.Aplikasi e-commerce memperluas bisnis UD Hoky Celluler Shop dan memungkinkan untuk mendapatkan pelanggan baru karena dengan menggunakan aplikasi e-commerce, UD Hoky Celluler Shop dapat menjual produknya secara online. Aplikasi e-commerce juga memudahkan UD Hoky Celluler Shop melihat stok produk serta laporan setiap transaksi yang terjadi dicatat secara otomatis oleh aplikasi sehingga proses bisnis UD Hoky Celluler Shop dapat menjadi lebih efektif dan efisien.

Pada penelitian yang dilakukan oleh penulis Aziz Firmansyah Lingga (2022) adalah sebagai aplikasi yang sudah berbasis *Android* dimana aplikasi ini dapat membantu dalam memberikan informasi produk-produk yang diawarkan oleh toko Usaha Kurma Kita. Serta menjadikan media promosi sekaligus pengenalan pada toko Usaha Kurma Kita. Dengan adanya aplikasi ini dapat membantu konsumen dalam mencari informasi tentang produk yang ditawarkan oleh Toko Usaha Kurma Kita. Fitur yang digunakan dalam sistem yang akan

dirancang adalah menu login, registrasi, tampilan menu utama, daftar pesanan, lokasi toko, dan informasi tentang toko.

II.2. Landasan Teori

II.2 Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu melayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018)

II.2.1 Software

Software merupakan komponen non fisik komputer berupa kumpulan program dan struktur datanya. Software dibagi menjadi dua kategori, yaitu sistem program dan aplikasi. Language Program (bahasa program) Bahasa program merupakan software yang dipergunakan untuk memprogram atau memberi instruksi kepada komputer agar menjalankan perintah yang dimasukkan (Korni Mufarola; 2019).

II.2.2. Firebase Realtime Database

Firebase Realtime Database sendiri merupakan basis data berbasis *cloud* NoSQL yang menyinkronkan data di semua klien secara *realtime*, dan menyediakan fungsionalitas *offline*. Data yang diinputkan disimpan ke dalam

database Realtime sebagai JSON. Semua klien yang terhubung dan saling berbagi dalam satu waktu, secara otomatis akan menerima pembaruan dengan data terbaru (Kurniawati dan Lukman Bacthiar; 2020: 59).

II.3 Android

Android adalah aplikasi sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak (Deny Adhar; 2019).

II.3.1 Versi Android

1. Android 1.5 Cupcake

Cupcake dirilis 30 April 2009. Cupcake menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5. Cupcake adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

2. Android 1.6 Donut

Android V1.6, codename Donut, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan reboot, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin. Bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan toping di atasnya.

3. Android 2.0/2.1 Eclair

Android 2.0/2.1 Eclair Dirilis 26 Oktober 2009. Eclair adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. Android 2.2 Froyo

Dirilis 20 Mei 2010. Menggunakan codename Froyo, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari Yoghurt. Froyo singkatan dari Frozen Yoghurt, Froyo adalah yoghurt yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim.

5. Android 2.3 Gingerbread

Android versi 2.3 Gingerbread dirilis resmi tanggal 6 Desember 2010. Gingerbread merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. Android 3.0 Honeycomb

Dirilis tanggal 22 February 2011. H adalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, Honeycomb/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. Android 4.0 Ice Cream Sandwich

Android 4.0-4.0.2 API Level 14 dan 4.0.3 API Level 15 pertama dirilis 19 Oktober 2011. Dinamai Ice Cream Sandwich. Ice Cream Sandwich es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. Android 4.1 Jelly bean

Android Jelly Bean diluncurkan pertama kali pada Juli 2012, dengan berbasis Linux Kernel dari Android 4.1 API Level 16, Android 4.2 API Level 17, Android 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. Android 4.4 KitKat

Android 4.4 Kitkat API level 19. Google mengumumkan Android KitKat (dinamai dengan izin Nestle dan Hershey) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. KitKat merupakan merk sebuah coklat yang dikeluarkan oleh Nestle. Rilis berikutnya setelah nama KitKat diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai 'Pie' (Hendra Nugraha Lengkong; 2016).

II.4 *Android Studio*

Android Studio adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi *Android* yang dikembangkan menggunakan *Java* dan mudah menyesuaikan ke platform baru. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*). *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai

konfigurasi perangkat keras. Sistem operasi yang mendasari *Android* merupakan lisensi di bawah naungan GNU, *General Public License Versi 2 (GPLv2)*, yang biasa dikenal dengan istilah *Copy left*. Istilah *copy left* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah terms. Distribusi *Android* berada di bawah *lisensi Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android* diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. (Ganda Yoga Swara, dkk; 2017) .

II.4.1 *Integrated Development Environment (IDE)*

Integrated Development Environment adalah aplikasi pengembang perangkat lunak dengan fungsi-fungsi terintegrasi yang dibutuhkan untuk membangun sebuah perangkat lunak seperti code editor, debugger, compiler, dan sebagainya (Efmi Maiyana; 2018).

II.4.2 *Android Software Development Kit (Android SDK)*

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Pada Android SDK ini terdiri dari debugger, libraries, handset emulator, dokumentasi, kode contoh dan tutorial. SDK memungkinkan pengembang membuat aplikasi untuk platform Android SDK, Android mencakup proyek sampel dengan kode sumber, perangkat pengembangan, emulator dan perpustakaan yang diperlukan untuk membangun aplikasi Android. Aplikasi yang ditulis dengan bahasa pemrograman Java dan

berjalan di Dalvik, mesin virtual yang dirancang khusus untuk penggunaan embedded yang berjalan diatas kernel Linux (Efmi Maiyana; 2018).

II.4.3 *Java Development Kit (JDK)*

JDK (Java Development Kit) adalah Paket fungsi API untuk bahasa pemrograman Java, meliputi Java Runtime Environment (JRE) dan Java Virtual Machine (JVM) (Efmi Maiyana; 2018).

II.4.4 *Android Virtual Device (AVD)*

Android Virtual Device merupakan emulator untuk menjalankan program aplikasi Android yang kita buat. AVD ini selanjutnya digunakan sebagai tempat untuk test dan menjalankan aplikasi Android tanpa harus menggunakan perangkat Android yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi Android, jenis dan ukuran layar dan besarnya memori. AVD bisa dibuat sebanyak yang kita inginkan (Efmi Maiyana; 2018).

II.5. *Unified Modelling Language (UML)*

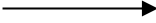
Menurut Rosa dan shalahuddin (2018) *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu sebagai berikut:

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Keterangan
	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
	Actor atau Aktor adalah Abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>Use Case</i>
	Asosiasi antara aktor dan <i>Use Case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan

	siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>Use Case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
 <<include>>	Include, merupakan di dalam <i>Use Case</i> lain (<i>required</i>) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
 <<extends>>	Extend, merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.

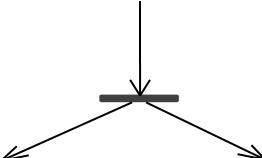
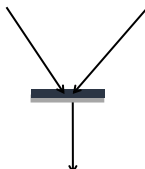
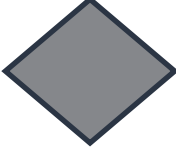
(Sumber: Rosa dan Shalahuddin, 2018)

2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu sebagai berikut :

Tabel II.2. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas

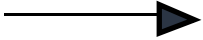
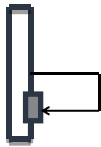
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, true atau false
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

(Sumber: Rosa dan Shalahuddin, 2018)

3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol- simbol *Sequence Diagram* dapat dilihat pada Tabel 4. Sebagai berikut:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal system dan menjadi landasan untuk menyusun basis data
	Boundary Class, berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak
	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	Message, simbol mengirim pesan antar class
	Recursive, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	Activation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi

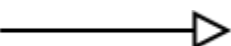
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation
---	---

(Sumber: Rosa dan Shalahuddin, 2018)

4. Class Diagram

Class Diagram Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, atribut, operasi dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel II.4. Simbol Class Diagram

Gambar	Keterangan
	Generalization, untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna khusus dapat menggunakan simbol ini.
	Class, untuk sebuah kelas pada struktur sistem. Penulisan

Nama_kelas +atribut +operasi	tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi.
	Interface, untuk simbol interface atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP).
	Association, digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya
	Directed Association, adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	Aggregation, adalah relasi antar kelas dengan makna semua bagian.
	Dependency, adalah relasi antar kelas dengan makna kebergantungan antar kelas.

(Sumber: Rosa dan Shalahuddin, 2018)

Tabel II.5. *Multiplicity Class Diagram*

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

(Sumber: Rosa dan Shalahuddin, 2018)