BAB II
TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah (Irma Januarti, 2012).

Menurut Dwi Iskandar (2015) Komputer adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasar urutan instruksi ataupun program yang diberikan kepadanya.

Menurut Hamacher komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan output berupa informasi.

Jadi, komputer adalah serangkaian alat elektronik yang digunakan untuk mengolah data menjadi suatu sistem informasi.

2.2. Teori Judul

2.2.1. Pengertian Aplikasi

Menurut Jogiyanto (1999:12), Aplikasi adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

2.2.2. Pengertian *Mobile*

Mobile dapat diartikan sebagai perpindahan yang mudah 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 ketempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA.

2.2.3. Pengertian *E-Commerce*

Menurut Laudon (1998), *E-commerce* adalah suatu proses membeli dan menjual produk-produk secara elektronik oleh konsumen dan dari perusahaan ke perusahaan dengan komputer sebagai perantara transaksi bisnis.

2.2.4. Pengertian Android

Menurut Hermawan (2011 : 1), Android merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, i-*Phone* OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

2.2.5. Pengertian Perusahaan

Perusahaan adalah setiap bentuk badan usaha yang menjalankan setiap jenis usaha yang bersifat tetap dan terus menerus dan didirikan, bekerja, serta berkedudukan dalam wilayah negara indonesia untuk tujuan memperoleh keuntungan dan atau laba (Kansil, 2001).

2.3. Teori Khusus

2.3.1. Metode Jacobson OOSE (*Object Oriented Software Engineering*)

Suatu rekayasa perangkat lunak yang digunakan untuk membangun software dengan melalui serangkaian proses terlebih dahulu dari Ivar Jacobson, dengan tahapan sebagai berikut:

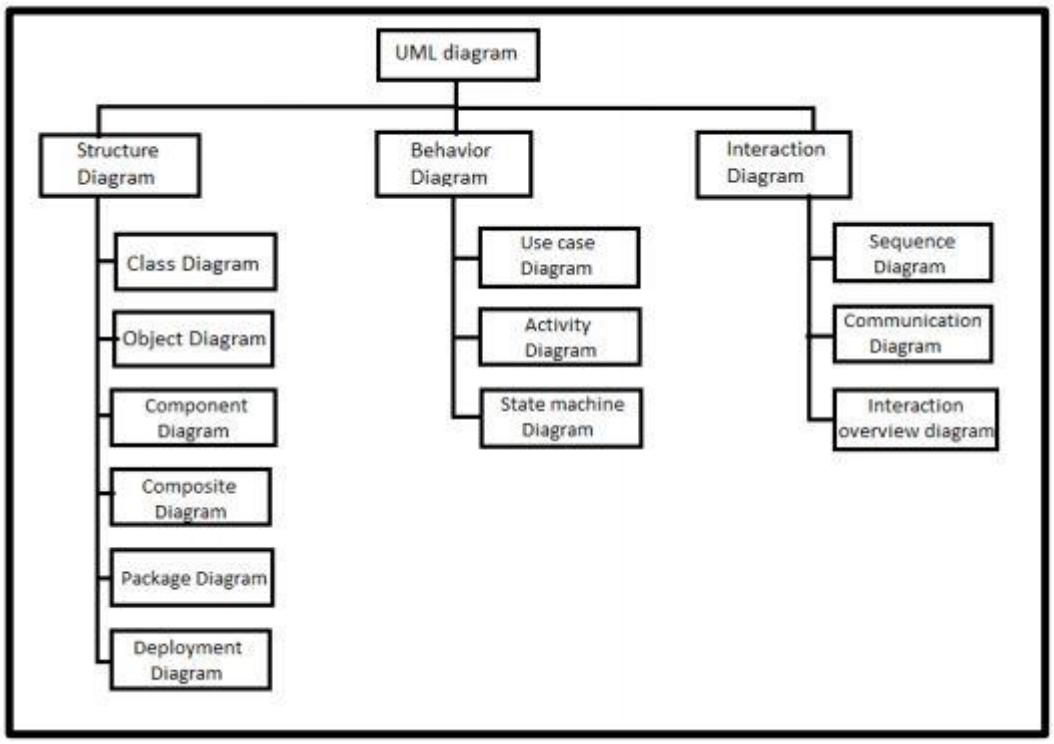
1. **Requirement Model**, yaitu membuat batasan sistem dan mendefinisikannya secara fungsional. Dalam tahapan ini meliputi definisi kebutuhan sistem, diagram *use case*, deskripsi *interface*, dan identifikasi objek.
2. **Analysis Model**, yaitu menganalisis perilaku/tingkah laku pada *use case*. Tahap ini merupakan pondasi untuk tahapan *design*.
3. **Design Model**, tahapan ini akan mempresentasikan analisis model dan mengadaptasinya ke tahapan implementasi.
4. **Implementasi Model**, berisi *source code* yang telah ditetapkan pada tahap *design* model.
5. **Test Model**, yaitu menguji kebenaran seluruh struktur sistem.

2.3.2. UML (*Unified Modeling Language*)

Menurut Nugroho (2009:4), UML (*Unified Modeling Language*) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Enggineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP).

2.3.3. Macam-macam Diagram UML (*Unified Modeling Language*)

Sukamto dan Shalahuddin (2013: 137) menjelaskan bahwa UML terdiri dari 13 (tiga belas) macam diagram yang dikelompokkan dalam 3 (tiga) kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.1 Macam-macam Diagram UML

Sumber: Sukamto dan Shalahuddin (2013)

Berikut ini penjelasan singkat dari pembagian kategori tersebut:

a. *Stucture diagrams,* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
b. *Behavior diagrams,* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction diagrams,* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.4. Diagram Use Case (*Use Case Diagram*)

Menurut Sukamto dan Shalahuddin (2013: 155), "Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Menurut Al Fatta (2007: 91), "Use Case Diagram adalah metode berbasis teks untuk menggambarkan dan mendokumentasikan proses yang kompleks." Sukamto dan Shalahuddin juga menjelaskan bahwa terdapat simbol-simbol yang digunakan di dalam Diagram Use Case, yaitu:

GAMBAR	NAMA	KETERANGAN
	*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	*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 menghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*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.

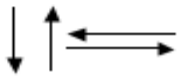
Gambar 2.2 Simbol-simbol dalam *Use Case* Diagram

Sumber: Sukamto dan Shalahuddin (2013)

2.3.5. Diagram Aktivitas (*Activity Diagram*)

Menurut Sukamto dan Shalahuddin (2013: 161), "Diagram Aktivitas atau Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak".

Berikut adalah simbol-simbol yang digunakan di dalam Diagram Aktivitas atau Activity Diagram:

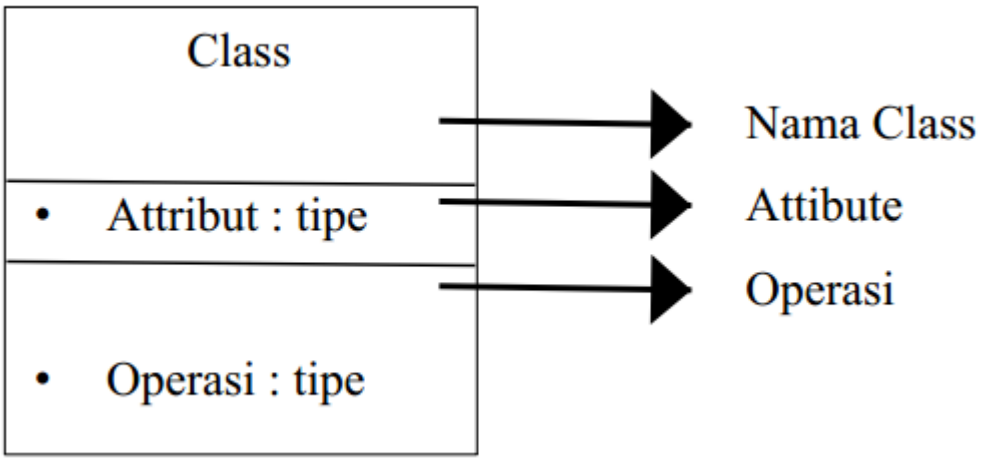
NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan diakhiri
5		*Decision*	Diguanakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		*Line Connector*	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Gambar 2.3. Simbol-simbol dalam Diagram Aktivitas (*Activity Diagram*)

Sumber: Sukamto dan Shalahuddin (2013)

2.3.6. Diagram Kelas (*Class Diagram*)

Menurut Sukamto dan Shalahuddin (2013: 141), "Diagram Kelas atau Class Diagram menggambarkan struktur sistem dari segi pendefinisiam kelas-kelas yang akan dibuat untuk membangun sistem."
Kelas diagram memiliki yang disebut atribut dan metode atau operasi. Berikut ini adalah struktur dan diagram kelas:



Gambar 2.4 Struktur Diagram Kelas (*Class Diagram*)

Sumber: Sukamto dan Shalahuddin (2013)

2.3.7. Diagram Sekuen (*Sequence Diagram*)

Menurut Rosa & Shalahuddin (2013: 165), Diagram Sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus ketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

No.	Simbol	Fungsi
1		Merepresentasikan entitas yang berada di luar sistem,mereka bisa berupa manusia, atau perangkat sistem lain.
2	 *Lifeline*	Merepresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak. Entitas ini memilki nama, *stereotype* atau berupa *instance*.
3	 *Self Message*	Relasi ini menunjukkan bahwa suatu objek hendak memanggil dirinya sendiri.
4	 *Message*	Relasi ini digunakan untuk memanggil operasi atau metode yang dimiliki oleh suatu objek. *Message* mengharuskan kita menyelesaikan proses baru kemudian memanggil proses berikutnya.

Gambar 2.5 Simbol-simbol Diagram Sekuen (*Sequence Diagram*)

Sumber: Rosa dan Shalahuddin (2013)

2.4. Teori Pemrograman

2.4.1. Pengertian Android Studio

Menurut Falker (2013), Android Studio adalah sebuah IDE dari Google yang diperkenalkan saat event Google I/O pada bulan Mei tahun 2013 dan merupakan IDE. Alternative selain IDE Eclipse. Dalam website resminya dikatakan bahwa Android Studio adalah IDE resmi untuk mengembangkan aplikasi android, yang berbasis intellij IDEA. Pada proses pembuatan aplikasi, penulis menggunakan software Android Studio 2.3.3, Java Development Kit (JDK) versi 7, Standard Development Kit (SDK) versi 14.

2.4.2. Struktur Android Studio

2.4.2.1. Java

Menurut Gosling et al. (2005, p1), Java adalah bahasa yang konkuren, berbasis kelas, dan bahasa yang berorientasi objek, yang dirancang sederhana yang banyak programmer dapat mencapai kefasihan dalam bahasa.

2.4.2.2. *Resources*

Stuktur Resources adalah struktur dalam android studio yang berfungsi sebagai tempat desain sebuah aplikasi yang akan dibuat. Folder resources dapat ditemukan di app / res directory, yang berisikan :

- *Drawable* folder sebagai tempat meletakan gambar-gambar yang kita pakai.
- *Layout* folder sebagai tempat xml untuk desain yang dipakai.
- *Menu* folder sebagai tempat xml untuk desain actionbar.
- *Values* folder sebagai tempat untuk meletakan konfigurasi statis, seperti strings, colors, dimensions dan styles.

2.4.2.3. Android Manifest.xml

Android Manifest adalah xml yang mengatur semua yang terjadi pada aplikasi anda. Semua *activity* harus dilaporkan kepada stuktur utama, yaitu androidmanifest.xml. *Title*, *icon*, dan *version* juga dilaporkan disini.

2.4.2.4. Intent

Intent merupakan suatu pesan yang digunakan untuk mengaktifkan tiga komponen dasar pada aplikasi Android yaitu *Activity*, *Service*, dan *Broadcast Receiver*. Aktifasi pada komponen-komponen tersebut bisa terjadi pada aplikasi yang sama atau berbeda, seperti menjalankan *Activity*, inisiasi Service, atau pengiriman pesan kepada *Broadcast Receiver*. Pada saat terjadi komunikasi antar komponen, Intent menyimpan paket informasi yang digunakan pada proses tersebut

2.4.3. Pengertian Extensible Markup Language (XML)

Menurut Hunter et al. (2007, p3), Extensible Markup Language (XML) merupakan teknologi dengan aplikasi dunia nyata, khususnya untuk manajemen, tampilan, dan organisasi data. XML bekerja dengan tujuan markup dari setiap jenis data tetapi dengan kompleksitas yang di eliminasi, XML tidak benar – benar merupakan bahasa, tetapi lebih pada sintaks yang digunakan untuk menjelaskan markup lain.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur, serta menyediakan format tertentu untuk dokumen – dokumen yang mempunyai data terstruktur.

2.4.4. Pengertian Basis Data (*Database*)

Menurut Asrianda (2008) database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Menurut Meiska, Arief, dan Bandi (2014: 101) Basis data merupakan kumpulan data non redundant yang dapat digunakan secara bersamaan (share) untuk aplikasi yang berbeda-beda. Untuk mendapatkan informasi yang berguna dari kumpulan data maka diperlukan suatu perangkat lunak (software) untuk memanipulasi data sehingga mendapatkan informasi yang berguna.

Database juga merupakan kumpulan data yang umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi. Sistem database merupakan sistem komputer yang digunakan untuk menyimpan dan mengelola data tersebut (Nugroho, Yuliandri Priyo, 2012).

Jadi, database adalah sekumpulan tabel-tabel yang saling berhubungan.

2.4.5. Pengertian Firebase

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan Mobile Apps Developer. Dengan adanya Firebase, apps developer bisa fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend. Beberapa fitur yang dimiliki oleh Firebase adalah sebagai berikut :

1. Firebase Analytics.
2. Firebase Cloud Messaging dan Notifications.
3. Firebase Authentication.
4. Firebase Remote Config.
5.Firebase Real Time Database.
6. Firebase Crash Reporting.

Dua fitur yang menarik adalah Firebase Remote Config dan Firebase Real Time Database. Secara sederhananya, Remote Config adalah fitur yang memungkinkan developer mengganti / mengubah beberapa konfigurasi aplikasi Android / iOS tanpa harus memberikan update aplikasi via Play Store / App Store. Salah satu konfigurasi yang bisa dimanipulasi adalah seperti warna / tema aplikasi.