BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian yang dilakukan Nur Rahmah Andayani & Priskila Makian (2016) yang berjudul "Pengaruh Pelatihan Kerja dan Motivasi Kerja terhada Kinerja Karyawan Bagian PT. PCI Elektronik International (Studi pada Karyawan PT PCI Elektronik International)" Menghasilkan suatu kegiatan untuk memperbaiki kemampuan kerja seseorang dalam kaitannya dengan aktivitas ekonomi. Pelatihan membantu karyawan dalam memahami suatu pengetahuan praktis dan penerapannya, guna meningkatkan keterampilan, kecakapan, dan sikap yang diperlukan organisasi dalam usaha mencapai tujuan. Pelatihan menurut Desler (2009) adalah Proses mengajarkan karyawan baru atau yang ada sekarang, keterampilan dasar yang mereka butuhkan untuk menjalankan pekerjaan mereka". Pelatihan merupakan salah satu usaha dalam meningkatkan mutu sumber daya manusia dalam dunia kerja. Karyawan, baik yang baru ataupun yang sudah bekerja perlu mengikuti pelatihankarena adanya tuntutan pekerjaan yang dapat berubah akibat perubahanlingkungan kerja, strategi, dan lain sebagainya.

Penelitian lain yang dilakukan Rifa'atul Mahmudah & Unqiqi dan Rafsanjani (2018) yang berjudul "Strategi Pemasaran Melalui Media *Online* Pada Produk Usaha Rumahan Kerupuk Bawang Dan Kripik Sukun Di Desa Cendoro Kec.Palang Kab.Tuban" Menghasilkan sebuah program pelatihan produk usaha rumahan kruput bawang dan kripik sukun melalui media *online*. Untuk dapat

meningkatkan daya saing serta nilai tambah dari barang, dilakukan pelatihan untuk meningkatkan SDM dari kelompok mitra. Pelatihan manajemen pemasaran, bertujuan untuk memberikan wawasan dan pemahaman kepada mitra tentang upaya pemasaran melalui promosi dan distribusi produk. Tidak lanjur dari pelatihan tentang strategi promosi melalui media *online* adalah dengan membuat media promosi di media *online*.

Penelitian lain yang dilakukan oleh Apol Pribadi, Khakim Ghozali, dan kawan-kawan (2018) yang berjudul "Pelatihan Teknik Social Media Marketing sebagai Sarana Pemasaran Produk pada Konveksi Kerudung 'Al- Kattar' Di Kelurahan Merjosari, Malang" menghasilkan Sosialisasi dan pelatihan penggunaan tentang sosial media marketing dan aplikasi rekap tersebut berjalan dengan lancar. Dalam mendukung proses pelatihan yang ada di Konveksi AL-KATTAR, maka telah berhasil dibuat beberapa modul pelatihan, diantaranya:

a. Modul Pembuatan Akun di Media *Facebook, Instagram*, dan *Shopee*
b. Modul Cara Pemasaran *via Facebook*
c. Modul Cara Pemasaran *via Instagram*
d. Modul Cara Pemasaran *via Shopee*

II.2. *Android*

Android adalah sistem operasi untuk telepon seluler yang berbasis *linux* yang dikembangkan oleh *google*. *Google* mengibaratkan android sebagai sebuah tumpukan *software*, setiap lapisan dari tumpukan ini menghimpun beberapa

program yang mendukung fungsi-fungsi spesifikasi dari *system* operasi yang disebut arsitektur *android*.

a. *Linux Kernel*

Android dibangun diatas *kernel linux* 2.6. Namun secara keseluruhan *android* bukanlah *linux*, karena dalam *android* tidak terdapat paket standar yang dimiliki *linux* lainnya.

b. Android *Runtime*

Android *Runtime* merupakan mesin virtual yang membuat aplikasi android menjadi lebih tanggoh dengan paket pustaka yang telah ada. Paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari aslinya untuk tertanam pada *kernel linux*.

c. *Application Framework*

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangankan aplikasi android. Selain itu, juga menyediakan abstrak generic untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi.

d. *Application*

Puncak dari diagramarsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program (Yuda Putra Utama : 2016 : 73).

II.3. Versi Android

Banyak *Smartphone* dan *Tablet* yang menggunakan sistem operasi dengan versi berbeda. Semakin versi tinggi fiturnya, semakin canggih *Smartphone* atau *tablet* tersebut. Telepon pertama yang memakai siste operasi Android adalah HTC Dream yang dirilis pada tanggal 28 Oktober 2008 (Ir. Yuniar Supardi : 2017 : 5).

Beberapa uraian Versi Android, yaitu :

Tabel II.1. Versi Android

Versi	Nomor Versi	Nama Versi	Tanggal Rilis
1	(Belum memakai)	*Android Beta*	5 November 2017
2	1.0	*Android 1.0*	23 September 2008
3	1.1	*Android 1.1*	09 Februari 2008
4	1.5	*Cupcake*	27 April 2009
5	1.6	*Donut*	15 September 2009
6	2.0	*Eclair*	26 Oktober 2009
7	2.0.1	*Eclair*	3 Desember 2009
8	2.1	*Eclair*	12 Januari 2010
9	2.2	*Froyo*	20 Mei 2010
10	2.2.1	*Froyo*	18 Januari 2011
11	2.2.2	*Froyo*	22 Januari 2011
12	2.2.3	*Froyo*	21 November 2011
13	2.3	*Gingerbread*	6 Desember 2010
14	2.3.3	*Gingerbread*	9 Februari 2011
15	2.3.4	*Gingerbread*	28 April 2011
16	2.3.5	*Gingerbread*	25 juli 2011

17	2.3.6	*Gingerbread*	2 September 2011
18	2.3.7	*Gingerbread*	21 September 2011
19	3.0	*Honeycomb*	22 Februari 2011
20	3.1	*Honeycomb*	10 Mei 2011
21	3.2	*Honeycomb*	15 Juli 2011
22	3.2.1	*Honeycomb*	20 September 2011
23	3.2.2	*Honeycomb*	30 Agustus 2011
24	3.2.4	*Honeycomb*	Desember 2011
25	3.2.6	*Honeycomb*	Februari 2012
26	4.0.1	*ICS (Ice Cream Sandwich)*	19 Oktober 2011
27	4.0.2	*ICS*	28 November 2011
28	4.0.3	*ICS*	16 Desember 2011
29	4.0.4	*ICS*	29 Maret 2012
30	4.1	*Jelly Bean*	09 Juli 2012
31	4.4	*Kit Kat*	31 Oktober 2013
32	5.0	*Lollipop*	12 November 2014
33	5.1	*Lollipop*	25 Juni 2014
34	6.0	*Marshmallow*	5 Oktober 2015
35	7.0	*Nougat*	22 Agustus 2016

(Sumber: Ir. Yunlar Supardi ; 2017)

II.4. *Firebase*

Firebase sebagai *database realtime* dan *backend*. *Backend* sendiri adalah sebuah bagian dalam kode aplikasi yang berhubungan langsung dengan isi *database*. Dengan *Firebase*, pengembang aplikasi ini nantinya tidak perlu

membuat *backend* sendiri melainkan menakai API yang telah disediakan oleh *Firebase* sehingga pengembangan aplikasi dapat dipersingkat. *Firebase* dikembangkan dengan menggunakan *database MongoDB* sehingga *Firebase* menggunakan tipe *database NoSQL (Google, 2014)*. Karena menakai tipe *database NoSQL* maka struktur *database* dari *Firebase* bersifat fleksibel dan cepat sehingga cocok untuk digunakan pada aplikasi berbasis *mobile* seperti yang akan peneliti kembangkan. (Afifur Rozak ; 2018 : 668)

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

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). 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. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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

a. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara

satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *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 :

Tabel II.2. Simbol *Use Case*

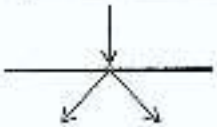
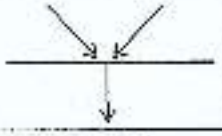
Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari *orang atau sistem* yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

b. Diagram Aktivitas (*Activity Diagram*)

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

Tabel II.3, Simbol *Activity Diagram*

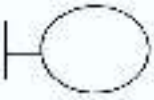
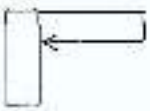
Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activities*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false*.
New Swimlane	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

c. Diagram Urutan (*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 yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry 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*, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	*Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Winda Gata ; 2013 : 7)

d. *Class Diagram* (Diagram Kelas)

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-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Association*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut, Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel **II.5.** ***Multiplicity Class Diagram***

Multiplicity	**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 atasan. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 8)