BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka ini dibuat berdasarkan kajian dari perbandingan penelitian aplikasi sebelumnya yang hanya berupa aplikasi berbasis *website.* Dari aplikasi tersebut kita dapat mengembangkan teknologi dengan menggunakan aplikasi berbasis *website* dan aplikasi berbasis *android* untuk pemasaran penjualan agar *customer* memesan barang lebih mudah dan cepat.

(Putra, 2012), dalam penelitiannya yang berjudul "Aplikasi Ponsel Berbasis Android Untuk Penjualan Pada Kios Eceran Q-Mono Flower". Fokus penelitian ini membahas kelebihan sistem aplikasi berbasis android yang memudahkan pemilik kios mendapatkan hasil informasi jumlah sisa stok barang, jumlah pembelian setiap hari, jumlah penjualan setiap hari. Selain itu penelitian ini juga membahas kelemahan dari sistem aplikasi seperti sangat menguras penyimpanan memori hp, tidak mampu memproses cetak nota jual dan beli, belum mampu mem*backup* data dan belum mampu meng*export* maupun meng*import file*.

(Evitarina, 2015) melakukan penelitian dengan judul "Rancang Bangun Aplikasi Pemesanan Barang Berbasis Android Pada Mini Market Faras Pangkalpinang". Penelitian tersebut bertujuan pada sistem aplikasi yang dirancang dengan menggunakan teknik yang tersusun dengan baik dan memudahkan *customer* dalam memenuhi kebutuhan sehari-hari.

(Isnanto & Putra, 2013) melakukan penelitian dengan judul "Rancang Bangun Aplikasi M-Commerce Berbasis Android Sebagai Media Pemesanan Pada Distro Online"Pengaplikasian *mobile commerce* bertujuan mempermudah penjualan dalam hal mengolah data pesanan barang, data konsumen dan melakukan pengecekan untuk pengiriman barang secara *online* yang menggunakan sistem implementasi dengan bahasa pemrograman Java Android, PHP dan relasi *database*.

Persamaan penelitian yang terdahulu dan sekarang adalah sama-sama membuat aplikasi berbasis *android* sebagai media pemasaran atau *e-commerce* untuk memperluas jangkauan pemasaran tetapi memiliki perbedaan berupa sistem

yang kebanyakan belum *online* dan tidak menggunakan *website* untuk lebih memasarkan penjualan. Selain itu pembanding penelitian yang terdahulu menggunakan konsep bahasa pemrograman PHP dan *database* yang berupa MySQL dan SQLite sebagai *web server*nya, sedangkan "APLIKASI PENJUALAN ONLINE PERKAKAS RUMAH TANGGA BERBASIS ANDROID" ini menggunakan konsep bahasa pemrograman PHP dan *database* yang terhubung dengan *domain hosting* agar dapat diakses *customer* dimana saja.

2.2 Landasan Teori

CLIENT SERVER

Jaringan *client server* diartikan sebagai suatu perancangan jaringan komputer yang mana perangkat *client* melakukan proses meminta data, dan server yang bertugas untuk memberikan respon dari *feedback* yang berupa data. Menurut (Refflan, 2012) *Client* adalah perorangan yang menghubungkan dengan *server* untuk meminta data atau layanan ke server sedangkan server adalah, perorangan yang menyediakan data atau layanan yang di diharapkan oleh *client*. *Client-Server* adalah pembagian tugas antara server dan client yang mempunyai akses menuju server di dalam suatu satu jaringan. Jadi arsitektur client-server ialah desain aplikasi yang berisi client dan server yang saling berkomunikasi ketika hendak mengakses server untuk suatu jaringan.

API

Seperti dikatakan oleh (Setiawan, 2013) API (*Application Programming Interface*) ialah semacam perintah, fungsi, komponen, dan protokol yang menyediakan sebuah sistem operasi ataupun bahasa pemrograman tertentu yang dapat digunakan oleh programmer saat membuat sebuah perangkat lunak.

ANDROID

Seperti yang dikatakan Oleh (Nazaruddin, 2012) *android* ialah sebuah sistem operasi untuk *handphone* yang berbasis *Linux*. *Android* menyediakan *platform* terbuka untuk para pengembang untuk membangun aplikasi mereka sendiri agar dapat digunakan oleh bermacam peranti bergerak. *Android* pada umumnya digunakan di *handphone* dan tablet PC. Fungsinya mirip layaknya sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS.

UML (Unified Modeling Language)

Berikut ini penjelasan mengenai *UML* menurut para ahli:

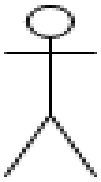
1. Seperti yang dikatakan oleh (Nugroho, 2010), UML ialah pemodelan untuk sebuah sistem atau perangkat lunak yang berorientasi objek. Pemodelan sesungguhnya dipakai untuk menyederhanakan masalah-masalah yang kompleksnya sedemikian rupa sehingga dapat dipahami dan mudah dipelajari.
2. Seperti yang dikatakan oleh (Rosa & Shalahuddin, 2013) UML ialah Salah satu acuan bahasa yang dipakai di dunia industri untuk menjelaskan requirement, membuat analisa & perancangan arsitektur dalam pemrograman berorientasi objek. didalam UML seperti *usecase diagram, activity diagram, class diagram, squence diagram.*

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa "*(UML)* adalah sebuah bahasa yang berdasarkan analisa dan perancangan untuk membangun dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)".

a. Use Case Diagram

Seperti yang dikatakan oleh (Rosa & Shalahuddin 2013) pengertian UML ialah Salah satu acuan bahasa yang dipakai di dunia industri untuk menjelaskan requirement, membuat analisa & perancangan arsitektur dalam pemrograman berorientasi objek. didalam UML seperti *usecase diagram, activity diagram, class diagram, squence diagram.* Tabel 2.1 menjelaskan simbol simbol yang digunakan.

Tabel 2.1 Keterangan simbolis *usecase*.

Lambang	Penjelasan
Use case 	Fungsional diberikan oleh sistem agar unit-unit saling bertukar pesan antar unit atau aktor; terkadang dinyatakan dengan menggunakan kata kerja diawal frase nama *use case*.
Aktor atau *actor* 	Orang dari sistem yang memiliki interaksi dengan sistem informasi yang hendak dibuat di luar sistem informasi yang hendak dibuat sendiri, walaupun lambang aktor ialah gambar orang, tapi aktor belum tentu merupakan orang.
Asosiasi atau *association*	Komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi atau *extend* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang di tambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan.
Generalisasi atau *generalization*	Hubungan generalisasi dan sepesialis (umum - khusus) antara dua buah *use case* dimana fungsi yang umum dari lainya.
Menggunakan atau *include* 	Relasi *use case* tambahan ke *use case* dimana *use case* yang ditambahkan membutuhkan *use case* ini untuk menjalankan fungsinya.

b. Activity diagram

Menurut (Sukamto & Shalahuddin, 2013) “Diagram aktivitas atau *activity* diagram mengambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnik atau menu yang ada pada perangkat lunak”. beikut adalah simbol yang digunakan.

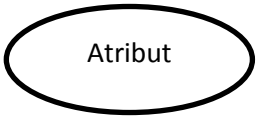
Tabel 2.2 Tabel keterangan simbolis *activity* diagram

Simbol	Deskripsi
Start state 	Suatu awal dari suatu aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Activity 	.Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Decition 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
End State 	Suatu akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

c. *Entity Relationship* Diagram *(ER Diagram)*

Seperti yang dikatakan oleh (Brady & Loonam, 2010), Entity Relationship diagram merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

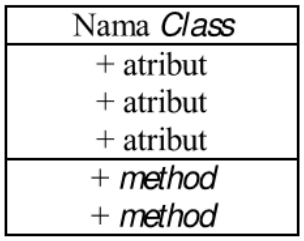
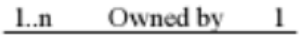
Tabel 2.3 Komponen penyusun ERD adalah sebagai berikut :

Komponen	Keterangan
	Persegi mewakili entias
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menhubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

d. *Class* Diagram

Class diagram menurut (Munawar, 2005) merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *atribut*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak dan memberikan. Rancangan arsitektur perangkat lunak menggambarkan desain sistem dari sistem yang akan dibuat.

Tabel 2.4 Tabel keterangan simbolis *class* diagram

Simbol	Deskripsi
Class 	*Class* blok-blok pembangunan pada pemrograman berorientasi Obyek, Sebuah *class* digambarkan sebagai sebuah kontak yang terbagi atas tiga bagian. Baian atas adalah bagian nama dari *class,* bagian tengah mendefinisikan *property* atau atribut *class,* bagian akhir mendefinisikan *method* dari sebuah *class*
Association 	Sebuah asosiasi merupakan sebuah *relationship* paling umum antara dua *class* dan dilambangkan oleh sebuah garis yang menghubungkan antara dua *class.*
Compotion 	Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *composition* terhadap tempat dia bergantung tersebut.
Dependency 	Kadangkala sebuah *class* menggunakan *class* lain, hal ini disebut *dependency.* Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada Sebuah *class* yang menggunkan *class* yang lain
Aggregation 	*Aggregation* mengindikasikan keseluruh bagian *relationship* dan biasanya disebut sebagai relasi.