

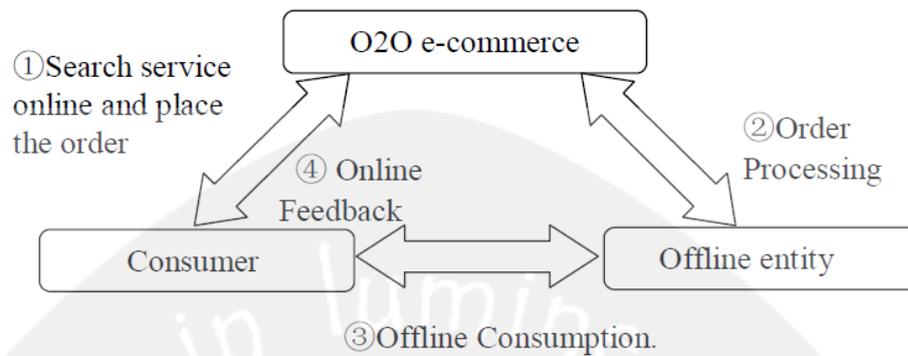
BAB 3

LANDASAN TEORI

3.1 M-Commerce

M-commerce (mobile commerce) merupakan bagian dari *e-commerce*, mencakup semua transaksi *e-commerce* dan dilakukan menggunakan *mobile device* (Mirzaie & Asadollahi, 2011). *Mobile commerce* tidak harus dilihat sebagai *e-commerce* dengan keterbatasan, melainkan sebagai bentuk unik dari *e-commerce* (Scornavacca & Barnes, 2006). Kemudian untuk pemanfaatan *mobile commerce* dalam dunia industri dapat diterapkan pada bidang *finances, services, retails, telecommunication and information technology services* (Niranjanamurthy, dkk., 2013).

Dengan perkembangan dunia perdagangan elektronik. Muncul model *e-commerce* yang dapat diterapkan pada *mobile commerce* bernama *O2O (Online to Offline)*. Model *e-commerce O2O* adalah kombinasi dari bisnis *offline* dengan membuka peluang usaha dengan *internet*. *E-commerce O2O* memungkinkan pelayanan melakukan pemesanan secara *online*, dan melakukan pengambilan barang secara *offline* (Yingseng Du & Youchun Tang, 2014). Gambar 3.1. menunjukkan aliran operasional terhadap model *O2O*.



Gambar 3.1. Aliran Operasional O2O e-Commerce

(Sumber: Yingseng Du & Youchun Tang, 2014)

3.2 Pamella Swalayan

Pamella Swalayan adalah supermarket yang telah memiliki delapan cabang di Yogyakarta. Menjual berbagai macam barang kebutuhan sehari-hari seperti makanan, minuman, pakaian, barang elektronik, dsb. Setiap cabang pamella swalayan melakukan transaksi jual beli secara *offline* dan jumlah yang sangat banyak setiap harinya. Barang yang akan dijual diberikan kode barang, diletakkan pada tempat yang disediakan selanjutnya calon pembeli melakukan pembelian dengan mengambil barang dan membayarnya dikasir. Melalui proses pembayaran, pembeli mendapatkan bukti pembayaran. Setiap proses yang berlangsung baik itu pencatatan detail produk sebelum siap di jual, dan proses pembelian hingga pembayaran disimpan di database pada masing-masing cabang. Dan dilakukan *back up* terpusat di satu *database* di salah satu cabang pamella.

3.3 Market Basket Analysis

Market Basket Analysis atau biasa dikenal *Association Rule Mining* adalah teknik *data mining* yang berasal dari bidang pemasaran (Aguinis, dkk., 2013). Teknik ini digunakan untuk menentukan produk-produk manakah yang akan dibeli oleh pelanggan secara bersamaan dengan melakukan analisis terhadap daftar transaksi pelanggan (Ngatimin, 2013). Dalam prosesnya, *Market Basket Analysis* akan menganalisis kebiasaan membeli (*buying habits*) dari konsumen dengan menemukan asosiasi antar produk-produk yang berbeda yang diletakkan konsumen dalam keranjang belanja (*shopping basket*).

3.3.1 Metode Dasar Association Rule

Metode ini terbagi menjadi dua bagian yaitu:

1. Analisa Pola Frekuensi Tinggi

Tahap ini adalah mencari kombinasi item yang memenuhi syarat minimum nilai support dengan rumus, seperti yang ditunjukkan pada rumus (1).

$$\text{Support}(A,B) = P(A \cap B) = \frac{\text{Jumlah transaksi mengandung } A \text{ dan } B}{\text{Total Transaksi}} \quad (1)$$

2. Pembentukan Aturan Asosiatif

Semua pola frekuensi tinggi ditemukan, dicari aturan asosiatif yang memenuhi syarat minimum untuk confidence dengan rumus, seperti yang ditunjukkan pada rumus (2).

$$\text{Confidence} = P(B | A) = \frac{\text{Jumlah transaksi mengandung } A \text{ dan } B}{\text{Jumlah Transaksi mengandung } A} \quad (2)$$

3.3.2 Algoritma Apriori

Algoritma apriori digunakan untuk mencari frekuensi *itemset* dari data yang terdapat di dalam *database*. Algoritma ini mengendalikan data yang akan dikombinasikan dengan data yang lain. Kemudian selanjutnya akan ditentukan nilai *support* dan *confidence* dengan metode *association rule*.

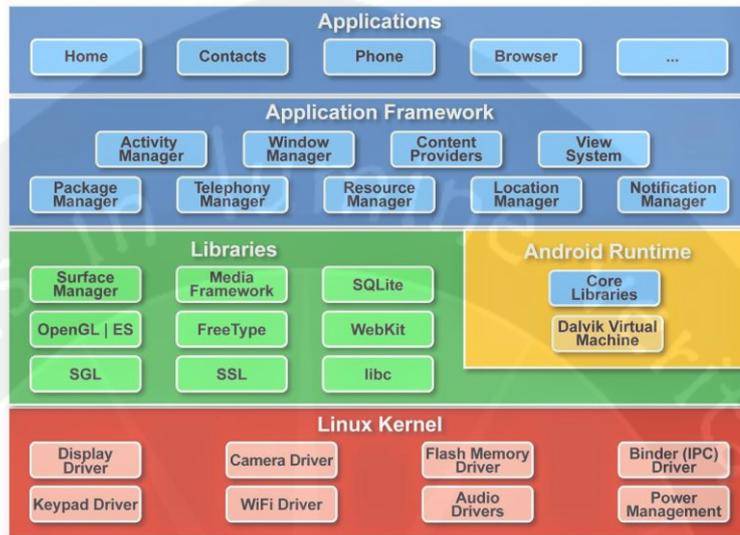
3.4 Ionic Framework, Mobile Hybrid Application

Ionic Framework adalah kerangka pembangunan aplikasi *mobile HTML5* yang ditargetkan untuk membangun aplikasi *mobile hybrid*. Aplikasi *hybrid* pada dasarnya adalah *website* yang berjalan dalam *browser* sebuah aplikasi yang memiliki akses ke lapisan *platform native*. Aplikasi *hybrid* memiliki banyak manfaat lebih dari aplikasi asli murni, khususnya dalam hal mendukung *multiplatform*, kecepatan dalam pembuatannya, dan akses ke kode pihak ketiga (Boedijono, dkk., 2015).

3.5 Android

Di akuisisi oleh Google pada agustus 2015. OS Android dibangun berbasis platform Linux yang bersifat *open source*, senada dengan Linux, Android juga bersifat *open source*. Dengan nama besar Google dan konsep *open source* pada OS Android, tidak membutuhkan waktu lama bagi android untuk bersaing dan menyisihkan Mobile OS lainnya seperti Symbian, Windos Mobile, Blackberry dan iOS. Kini siapa yang tak kenal Android yang telah menjelma menjadi penguasa Operating System bagi Smartphone (Lengkong, dkk., 2015). Pada Gambar 3.2. di

bawah ini menunjukkan bentuk arsitektur pada *platform* Android.



Gambar 3.2. Arsitektur Android

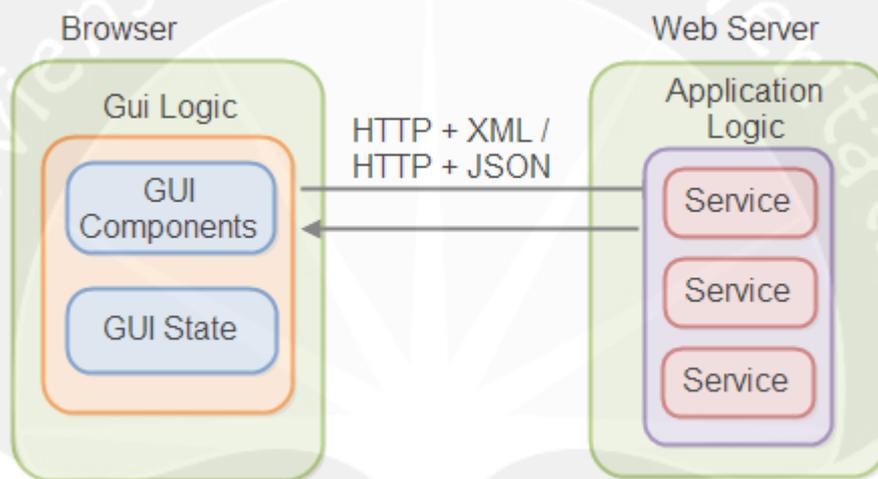
3.6 *Web*

Web server adalah *software* yang menjadi tulang belakang dari *world wide web* (*www*) yang pertama kali tercipta sekitar tahun 1980an. *Web server*, untuk berkomunikasi dengan *client*-nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*). Dengan protokol ini, komunikasi antar *web server* dengan *client*-nya dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan di atas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari (Nurmiati, 2012).

3.7 JSON (Java Script Object Notation)

JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di website (Wijaya, dkk., 2012)

Contoh penggunaan JSON pada aplikasi MEPAMS dapat dilihat pada Gambar 3.3.



Gambar 3.3 Contoh Penggunaan JSON Rest

3.8 Codeigniter

CodeIgniter adalah sebuah framework PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibandingkan jika menulis semua kode program dari awal. CodeIgniter menyediakan banyak library untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi berbasis web. Selain itu, struktur dan susunan logis dari CodeIgniter membuat aplikasi yang dibuat menjadi semakin teratur dan rapi (Koepradono, dkk., 2013).