

BAB 3

LANDASAN TEORI

Bab ini akan membahas hal-hal yang mendasari dibuatnya aplikasi JOGIFT, arsitektur, bahasa pemrograman dan *tools* yang digunakan dalam pembuatan aplikasi JOGIFT.

3.1 Produk

Pengertian produk (*product*) menurut Kotler & Armstrong, (2001: 346) adalah segala sesuatu yang dapat ditawarkan ke pasar untuk mendapatkan perhatian, dibeli, digunakan, atau dikonsumsi yang dapat memuaskan keinginan atau kebutuhan. Secara konseptual produk adalah pemahaman subyektif dari produsen atas sesuatu yang bisa ditawarkan sebagai usaha untuk mencapai tujuan organisasi melalui pemenuhan kebutuhan dan kegiatan konsumen, sesuai dengan kompetensi dan kapasitas organisasi serta daya beli pasar.

3.2 Arsitektur Client-Server

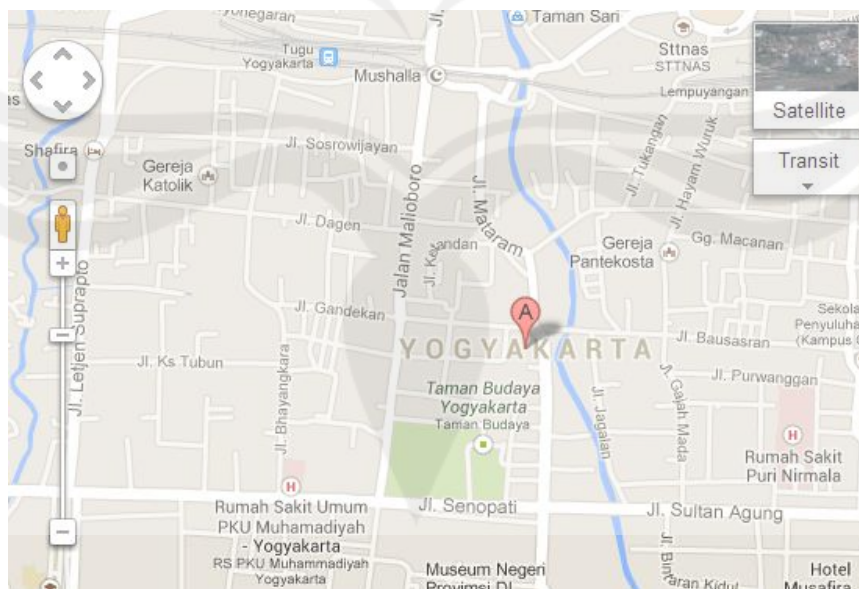
Arsitektur *client-server* adalah model komputasi terdistribusi di mana terdapat peminta (*client*) dan penyedia (*server*). Dalam arsitektur *client-server*, baik peminta maupun penyedia bekerja di komputer terpisah yang dihubungkan dengan suatu jaringan. Aplikasi yang dijalankan oleh *client* membutuhkan proses yang diminta dari *server* (Maffeis, 2005).

Klien adalah proses atau program yang mengirimkan pesan ke *server* melalui jaringan. Pesan tersebut berisi permintaan pada *server* untuk melakukan tugas tertentu.

Server adalah proses atau program yang mendengarkan permintaan klien, kemudian menjalankan tugas sesuai pesan yang dikirimkan (Maffeis, 2005).

3.3 Peta

Peta adalah gambar atau lukisan pada kertas dan sebagainya yang menunjukkan letak tanah (anonim, 2013). Peta bisa disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Istilah peta berasal dari bahasa Yunani *mappa* yang berarti taplak atau kain penutup meja. Namun secara umum pengertian peta adalah lembaran seluruh atau sebagian permukaan bumi pada bidang datar yang diperkecil dengan menggunakan skala tertentu. Sebuah peta adalah representasi dua dimensi dari suatu ruang tiga dimensi. Banyak peta mempunyai skala, yang menentukan seberapa besar objek pada peta dalam keadaan yang sebenarnya.



Gambar 3.1 Ilustrasi peta pada Google Map

3.4 Sistem Layanan Berbasis Lokasi

Sistem Layanan Berbasis Lokasi, atau lebih dikenal dengan *Location-based Services (LBS)*, menggabungkan antara proses dari layanan *mobile* dengan posisi geografis dari penggunanya. Poin pentingnya adalah ketika posisi target, di mana sebuah target bisa jadi adalah pengguna LBS itu sendiri atau entitas lain yang tergabung dalam suatu layanan (Kupper, 2005).

LBS mengarah pada aplikasi-aplikasi dan layanan-layanan yang menyediakan informasi lokasi mengenai suatu pengguna *mobile*, untuk menyediakan sebuah layanan atau informasi khusus pada pengguna.

Untuk menentukan lokasi geografis dari pengguna diantaranya dengan menggunakan *Global Positioning System (GPS)*, yang mana adalah sistem yang berfungsi sebagai sistem navigasi global yang dapat menerima informasi dari sistem satelit. Satelit GPS ini memancarkan sinyal yang memungkinkan penerima sinyal GPS untuk mendapatkan informasi berupa lokasi penerima, arah, dan kecepatan (Wicaksono, 2008).

Sistem GPS mempunyai 3 segmen utama (RDC, 2004), yaitu:

1. Segmen Kontrol

Segmen kontrol terdiri dari suatu sistem yang mengontrol kegiatan satelit GPS yang tersebar diseluruh dunia.

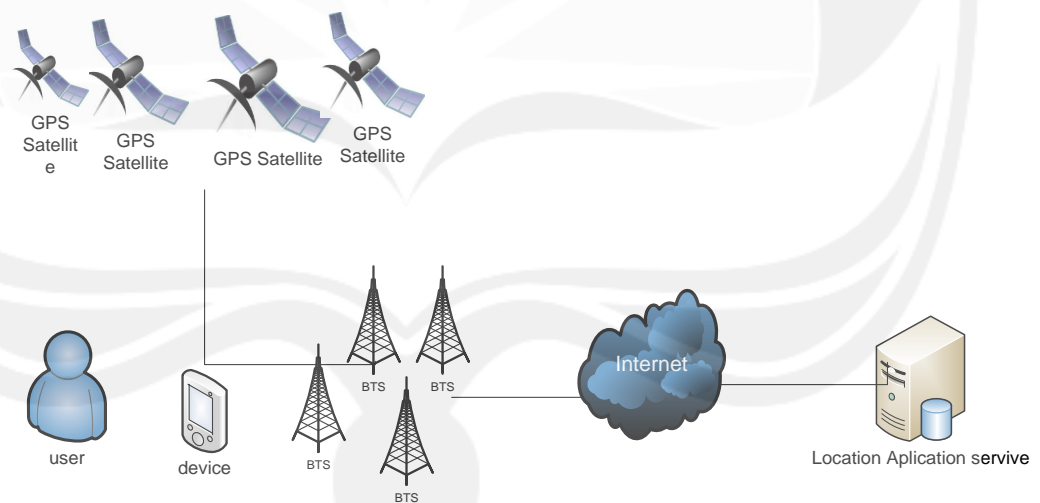
2. Segmen Satelit

Segmen satelit terdiri dari satelit-satelit GPS. Satelit bertugas untuk menerima dan menyimpan data yang ditransmisikan oleh segmen kontrol dan memancarkan data secara terus-menerus ke segmen

pengguna. Konstelasi GPS terdiri dari 24 satelit yang menempati 6 buah bidang orbit yang berbentuk mendekati lingkaran, dengan masing-masing bidang orbit ditempati 4 buah satelit. Kedudukan masing-masing satelit dalam tiap orbitnya diatur sedemikian rupa dengan jarak interval diantaranya tidak sama. Hal ini untuk meminimalkan pengaruh keadaan dimana ada satelit yang tidak berfungsi.

3. Segmen Pengguna

Segmen pengguna terdiri dari para pengguna yang tersebar diseluruh permukaan bumi dan alat penerima GPS. Segmen pengguna bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi, arah, jarak, dan waktu yang diperlukan.



Gambar 3.2 Ilustrasi Location Based Service (LBS)

3.5 Google Maps API

Google Maps API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan google Map di suatu aplikasi yang tidak dibuat oleh Google. Google

Maps API adalah suatu *library* yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan (Elian, 2012). Dalam perkembangannya Google Maps API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan *geocoding*, dan memberikan penutun arah. Google Maps API bersifat gratis untuk publik.

Penggunaan Google Maps API pada pengembangan aplikasi android dengan menggunakan Eclipse dan komputer menggunakan sitem operasi *windows*.

Kekurangan yang ada pada Google Maps API :

1. Jika ingin melakukan akses harus terdapat layanan *internet* pada perangkat yang digunakan.

Kelebihan yang ada pada Google Maps API:

1. Dukungan penuh yang dilakukan Google sehingga terjamin dan bervariasi fitur yang ada pada Google Maps API.
2. Banyak pengembang yang menggunakan Google Maps API sehingga mudah dalam mencari referensi dalam pengembangan aplikasi.

3.6 Android

Android adalah sistem operasi untuk perangkat *mobile* yang mana terdiri dari sebuah sistem operasi, dan aplikasi utama yang berbasis *Linux* (Belluccini et al, 2008). *Android* menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Aplikasi *android* ditulis dalam bahasa pemrograman *Java* dengan *Java Libraries* yang dikembangkan *Google*.

Android bukan sekedar hanya untuk perangkat *mobile* saja, *android* merupakan sebuah sistem operasi yang dikemas sedemikian rupa sehingga dapat digunakan untuk berbagai perangkat yang menggunakan layar (Simmonds, 2010).

Aplikasi berbasis *android* dapat dengan mudah dibuat dalam *IDE (Integrated Development Environment) Eclipse* dengan bantuan eksistensi *android* yang disebut *ADT (Android Development Tools)* yang fungsinya diantara lain untuk:

1. Memungkinkan akses ke *Android Development Tools* lainnya.
2. Memungkinkan akses *New Project Wizard* untuk aplikasi *android*.
3. Mengotomasi dan menyederhanakan proses pembangunan aplikasi *android*.
4. Menyediakan *Android Code Editor* untuk membantu pengguna menulis *file XML* dengan *valid* untuk *Android Manifest* dan *file resource* lainnya.
5. Mempublikasikan aplikasi menjadi *signed APK* agar dapat didistribusikan kepada pengguna lain. (Android Developer, 2010).

3.7 PHP

PHP adalah bahasa *scripting language* yang dirancang secara khusus untuk penggunaan pada *web*. PHP kependekan dari *HyperText Preprocessor*. Seperti bahasa pemrograman *web* lainnya PHP memproses seluruh perintah yang berada dalam skrip

PHP didalam *web browser* dan menampilkan *output*-nya ke dalam *web browser* klien. PHP adalah bahasa *scripting* yang menghasilkan *output* HTML ataupun *output* lain sesuai keinginan program yang dijalankan pada *server side* (Suryatiningsih, 2010).

3.8 Application Programming Interface

Application Programming Interface (API) adalah sekumpulan *commands*, fungsi, dan protokol yang digunakan oleh aplikasi untuk berkomunikasi dengan sistem operasi maupun program lain seperti *Database Management System* (DBMS) maupun protokol komunikasi (PcMag,2013). Pembangunan aplikasi untuk Android menggunakan 3 buah API, yakni Google Maps API v2, Google Location APIs dan Google Places API. Google Maps API v2 memungkinkan pengembang untuk menampilkan peta interaktif dan kaya fitur kepada pengguna di dalam aplikasi Android (Google Developer, 2013). Google Location APIs memudahkan pengembang untuk membangun aplikasi yang peka terhadap lokasi pengguna, tanpa memerlukan fokus pada detail teknologi yang ada dibaliknya, sementara Google Places Api adalah sebuah layanan yang mengembalikan informasi mengenai lokasi menggunakan HTTP *request* dan mengembalikan hasil dalam *format* JSON atau XML (Google Developer, 2013).

3.9 Arsitektur Android

Arsitektur android terdiri dari beberapa lapisan (Sariana, 2010), yaitu:

1. *Linux Kernel* :

Android bukan Linux, tetapi android dibangun diatas Linux Kernel versi 2.6.

2. *Libraries* :

Android menyertakan satu set *libraries* C atau C++ yang digunakan dalam berbagai komponen sistem android.

3. *Android Runtime* :

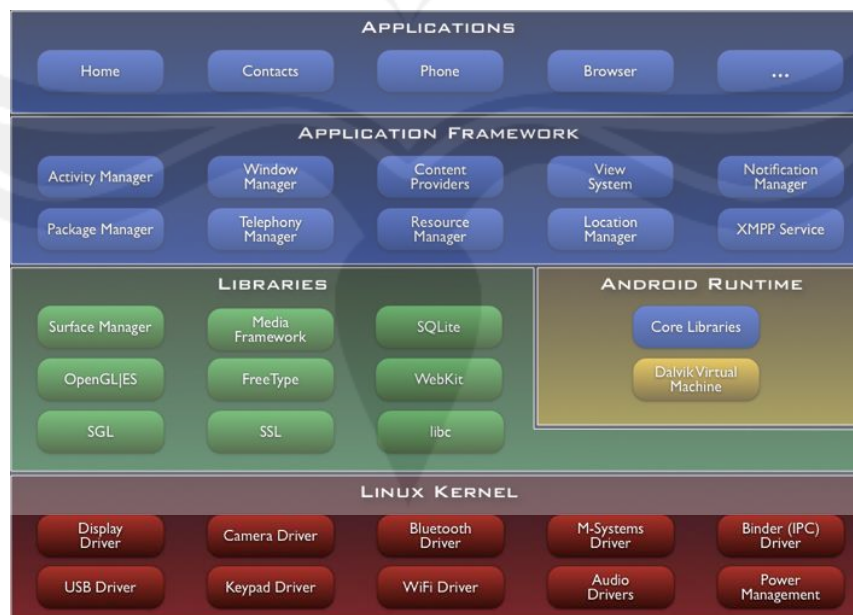
Android terdiri dari satu set perpustakaan inti (*core libraries*) yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat dalam perpustakaan inti dari bahasa pemrograman java.

4. *Application Framework* :

Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah.

5. *Application and Widget* :

Pada lapisan ini *developer* menempatkan aplikasi yang dibuat.

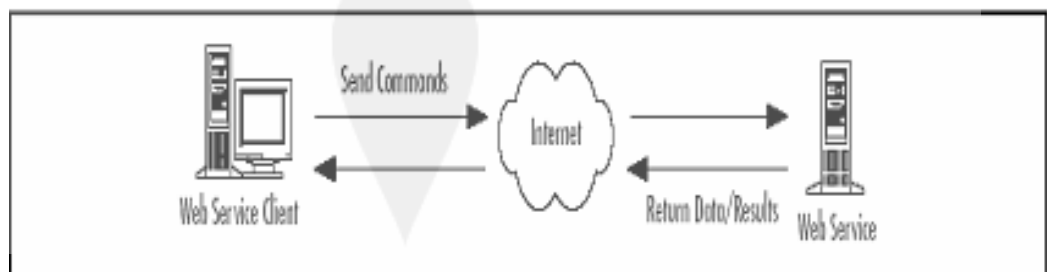


Gambar 3.1Ilustrasi Arsitektur Android

3.10 Web Service

Web Service adalah sebuah teknologi yang bisa digunakan untuk membuat sebuah aplikasi yang bersifat *platform-independent*. *Web Service* dapat menghubungkan antar *web* yang berbeda, aplikasi yang berbeda, bahkan sampai *device* yang berbeda dalam sistem operasi yang berbeda pula (Hendrawan, 2006).

Web Service dapat menjalankan operasi-operasi termasuk akses data, *update database*, mengatur integritas data, mengirim *respon* dari *request* yang dikirim oleh *user*. *Web service* tidak memiliki tampilan karena *web service* termasuk dalam *Bussiness-Service tier*. Artinya di dalam *web service* hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh aplikasi lainnya. Pada penggunaan pada sistem perancangan ini digunakan *web service* yang berupa file ber-ekstensi *.php* untuk dapat menjembatani antara aplikasi *mobile phone* dengan basis data. *Web service* diletakkan tempatnya sama dengan lokasi basis data. Penggunannya dengan menggunakan *HTTP Access* dimana hasil balikan dari *web service* adalah data dengan ekstensi *.json*.



Gambar 3.2 Cara Kerja Web Service

3.11 Database Management System

Database Management System berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. DBMS terdiri dari set program pengelola untuk menambahkan data, menghapus data, mengambil data, dan membaca data. *Database* adalah kumpulan *file* yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap *file* yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkupn perusahaan, instansi (Kristanto, 2004).

Sekian pembahasan hal-hal yang mendasari dibuatnya aplikasi JOGIFT, arsitektur, bahasa pemrograman dan *tools* yang digunakan dalam pembuatan aplikasi JOGIFT. Pada bab selanjutnya akan dibahas mengenai analisis dan perancangan aplikasi JOGIFT.