

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 TINJAUAN PUSTAKA

Penelitian yang berhubungan dengan penelitian ini sebelumnya pernah dilakukan seperti ditunjukkan pada tabel 2.1.

Tabel 2.1 Perbandingan Tinjauan Pustaka

Parameter Penulis	Objek	Metode atau Teknologi	Bahasa Pemrograman
Abdika Assabri (2018)	Toko oleh-oleh	A* (star)	HTML, PHP, JavaScript
Anugerah Penembahan Atunggal (2016)	Bengkel Motor	GPS	Android
Faishal Abrari (2017)	Cafe	Mapbox	Android
Ridho Warno Saputro (2018)	Halte Trans Jogja	Google Maps dan Here Maps	Android
Ryanda Reza Nugraha (2015)	Wisata Belitung	LBS	Android
Yang dikembangkan, (2020)	Dealer Motor	LBS dan Manhattan	Android

Berdasarkan tabel 2.1 Abdika Assabri (2018) mengembangkan aplikasi pencarian toko oleh-oleh dengan metode A* (star) untuk menentukan lokasi alternatif menggunakan *platform* Web. Anugerah Penembahan Atunggal (2016)

mengembangkan aplikasi pencarian bengkel motor dengan teknologi GPS untuk menentukan rute menggunakan *platform* Android. Faishal Abrari (2017) mengembangkan aplikasi pencarian lokasi café terdekat dengan teknologi mapbox untuk menentukan lokasi dan rute menggunakan platform Android. Ridho Warno Saputro (2018) mengembangkan aplikasi pencarian halte trans jogja dengan membandingkan layanan *google maps* dan *here maps* menggunakan *platform* Android. Ryanda Reza Nugraha (2015) mengembangkan aplikasi wisata Belitung dengan metode LBS untuk menentukan lokasi wisata menggunakan *platform* android.

Penelitian yang dikembangkan, (2020) yaitu pencarian lokasi dealer motor. Yang membedakan dari yang dikembangkan sebelumnya adalah menggunakan metode LBS dan metode *Manhattan* untuk membandingkan jarak dengan menggunakan *platform* android. Dimana *output* dari aplikasi ini memberikan informasi dealer motor dan jarak terdekat dealer motor dari lokasi pengguna.

2.2 DASAR TEORI

2.2.1. Android

1. Pengertian Android

Menurut Nazruddin Safaat (2012), Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. Android juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak.

2. Arsitektur Android



Gambar 2.1 Arsitektur Android (Nazruddin Safaat 2012)

Menurut Nazruddin Safaat (2012), secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut:

a. *Application* dan *widgets*

Application dan *widgets* adalah layer yang berhubungan dengan aplikasi saja, biasanya dilakukan download kemudian di install dan aplikasi tersebut dijalankan. Di layer terdapat aplikasi inti termasuk klien email, program, sms, kalender, peta, browser, kontak dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java

b. *Application Framework*

Android adalah *Open Development Platform*, yang dimana android menawarkan kepada pengembang atau memberi kemampuan kepada

pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan *service background*, mengatur alarm, menambahkan status *notification*, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur dirancang agar dapat dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*).

Jadi Application Framework ini adalah layer dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa sms dan panggilan telepon.

c. *Libraries*

Libraries ini adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, Layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL

d. *Android Runtime*

Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan Implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android.

e. *Linux Kernel*

Tumpukan paling bawah pada arsitektur Android ini adalah *kernel*. Linux *kernel* yang digunakan Android adalah linux kernel *release 2.6*. *Kernel*

berperan sebagai abstraction layer antara *hardware* dan keseluruhan *software*. Namun secara keseluruhan Android bukanlah linux, karena dalam Android tidak terdapat paket standar yang dimiliki oleh linux lainnya.

3. Fundamental Aplikasi

Aplikasi android dirancang dengan menggunakan bahasa pemrograman java. Kode-kode yang digunakan untuk merancang aplikasi pada java di-*compile* bersamaan dengan *resource* melalui proses yang di *package* oleh *Tools* yang dinamakan “apt tools” dan dipaketkan sehingga menjadi satu *file* dengan format apk. *File* apk tersebut merupakan rancangan aplikasi yang sudah jadi dan siap untuk digunakan.

Ada 4 jenis komponen dalam aplikasi android, misalnya:

a. *Activities*

Suatu aktifitas yang menyajikan *User Interface* (UI) bagi pengguna, sehingga pengguna dapat dengan mudah berinteraksi dengan aplikasi. Aplikasi android biasanya memiliki banyak aktifitas, namun mungkin ada aplikasi android yang hanya memiliki satu aktifitas saja. Untuk menampilkan suatu aplikasi yang bertindak sebagai *User Interface* ketika aplikasi tersebut dijalankan dapat menggunakan satu aktifitas saja. Untuk memindahkan satu aktifitas ke aktifitas yang lain dapat menggunakan event, seperti contoh klik tombol button, memilih opsi, ataupun menggunakan trigger tertentu.

b. Service

Service tidak memiliki GUI (*Graphic User Interface*), namun service berjalan seperti background. Service harus berada dalam satu kelas induk untuk dapat dijalankan. Contoh service seperti memainkan atau menjalankan aplikasi *music* pada media player dengan memutar lagu pada list yang sudah ada dan pengguna dapat memilih lagu yang akan diputar, hal tersebut dikarenakan service menggunakan dua atau lebih aktifitas.

c. Broadcast Receiver

Broadcast Receiver dapat menerima dan menampilkan notifikasi. Broadcast Receiver memberikan informasi pengingat dalam suatu aplikasi android, contoh broadcast receiver ada pada suatu aplikasi android yang dapat menginisiasi broadcast misalnya memberikan informasi bahwa data yang diunduh telah selesai dan dapat digunakan.

d. Content Provider

Membuat sekumpulan data-data aplikasi yang spesifik dan dapat digunakan oleh aplikasi lainnya kemudian data disimpan dalam file sistem berupa database seperti pada database SQLite dan juga memberikan cara untuk mengakses data yang dibutuhkan oleh suatu aktifitas.

2.2.2. Location Based Service (LBS)

Location Based Services (LBS) suatu layanan yang memberikan informasi berdasarkan posisi pengguna (device) sebagai acuan utama. Menurut Nazarudin Safaat (2012), dua unsur utama dalam LBS yaitu :

1. *Location Manager (API Maps)*

Menyediakan perangkat bagi sumber atau *source* untuk LBS, *Application Programming Interface (API)* menyediakan fasilitas untuk menampilkan atau memanipulasi peta.

2. *Location Providers (Api Location)*

Menyediakan teknologi pencarian lokasi yang di gunakan oleh perangkat. *API Location* berhubungan dengan data GPS dan data lokasi *real-time*.

Data lokasi pengguna biasanya di dapatkan melalui jaringan telepon seluler ataupun menggunakan GPS. LBS memiliki komponen-komponen yang menunjang dalam prosesnya yaitu :

1. Perangkat mobile

Pengguna membutuhkan perangkat mobile untuk menggunakan layanan LBS ini. Sepertihalnya *smartphone, tablet* dan lain-lain.

2. Jaringan komunikasi

Jaringan komunikasi digunakan untuk menghubungkan perangkat mobile dengan perangkat lainnya.

3. Komponen pengambil posisi latitude dan longitude (satelit)

Satelit merupakan alat yang menentukan posisi pengguna. Seperti jarak, lokasi dan lain-lain.

4. Data dan provider content

Data yang di dapatkan akan di proses di *server* dan dikirim kembali ke pengguna berupa data yang telah akurat.

5. WMS (*web map server*)

Merupakan server dimana tempat pengumpulan dan pemrosesan data.

2.2.3. Manhattan Distance

Merupakan jarak yang diukur mengikuti jalur yang tegak lurus disebut dengan jarak manhattan. Pengukuran dengan jarak manhattan sering digunakan karena mudah perhitungannya, mudah dimengerti dan untuk beberapa masalah lebih sesuai, misalnya untuk menentukan jarak antar kota, jarak antar fasilitas dimana peralatan pemindahan bahan hanya dapat bergerak searah tegak lurus. (Setya Yaniar.Nimas.2012).

Dalam pengukuran jarak manhattan digunakan notasi sebagai berikut:

$$d(x,y) = |X_i - Y_i| \dots\dots\dots(2.1)$$

$d(x,y)$ = manhattan distance yaitu jarak antara data pada titik x dan titik y

x = Data

y = Pusat cluster

2.2.4. SIG

Sistem Informasi Georafis atau *Georaphic Information Sistem (GIS)* merupakan suatu sistem informasi yang berbasis komputer, dirancang untuk bekerja dengan menggunakan data yang memiliki informasi spasial (bereferensi keruangan). Sistem ini meng*capture*, mengecek, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang secara spasial mereferensikan kepada kondisi bumi. Teknologi SIG mengintegrasikan operasi-

operasi umum database, seperti *query* dan analisa statistik, dengan kemampuan visualisasi dan analisa yang unik yang dimiliki oleh pemetaan.

2.2.5. Google Map API

Google Maps *Application Programming Interface* (API) merupakan fitur aplikasi yang digunakan untuk memfasilitasi para pengguna yang ingin mengintegrasikan Google Maps ke dalam *website* masing-masing dengan menampilkan data point milik sendiri. Google Maps dapat di-*embed* pada *website* eksternal dengan menggunakan Google Maps API. Maps API Javascript harus dimasukkan terlebih dahulu sebelumnya agar aplikasi Google Maps dapat muncul di *website* tertentu.

2.2.6. Fungsi *Latitude* dan *Longitude*

Dalam tampilan google maps API terdapat cara menghitung koordinat pencarian lokasi yang menunjukkan garis *horizontal* dan *vertical*. Diawali dengan 2 koordinat letak bumi dari angka depan *latitude* (ketinggian) dan angka belakang (*longitude*). Fungsi *latitude* dan *longitude* adalah untuk mengukur jarak antara satu titik dengan titik yang lain. Titik koordinat menjadi hal yang penting saat membaca peta. Terdapat definisi dari *latitude* dan *longitude* sebagai berikut:

a. Latitude

Latitude atau garis lintang adalah garis maya yang melingkari bumi ditarik dari arah barat hingga ke timur atau sebaliknya, sejajar dengan *equator* (garis khatulistiwa). Garis lintang terus melingkari bumi, dari *equator* hingga ke bagian kutub utara dan kutub selatan bumi. Titik

yang dipakai dari 0 ke 90 derajat kearah kutub utara dan 0 ke -90 derajat ke kutub selatan.

b. Longitude

Longitude atau garis bujur adalah garis maya yang ditarik dari kutub utara hingga ke kutub selatan atau sebaliknya. Derajat antar garis bujur semakin melebar di daerah khatulistiwa dan makin menyempit di daerah kutub. Jika pada garis lintang, daerah yang dilalui garis khatulistiwa (equator) di anggap sebagai 0 derajat untuk garis bujur, tempat yang dianggap sebagai 0 derajat adalah garis dari kutub utara ke kutub selatan yang tepat melintasi kota Greenwich di Inggris.

2.2.7. JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus

(dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array.

2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.