BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Sebelum dilakukannya penelitian ini, penelitian sejenis mengenai layanan berbasis lokasi juga pernah dilakukan oleh Siprianus Tago dari STMIK Akakom Yogyakarta. Penelitian tersebut mengambil kasus mengenai pencarian asrama mahasiswa Kalimantan di Yogyakarta dengan menggunakan Google Maps API, PHP, AJAX, dan MySql yang berbasis web. Kasus yang diambil tersebut menggunakan data yang kurang dinamis dalam kehidupan sehari-hari.

Yudi Rahman yang juga selaku alumni mahasiswa STMIK Akakom Yogyakarta juga pernah melakukan penelitian yang serupa. Kasus yang diangkat yaitu mengenai informasi bangunan, informasi wisata, dan informasi kuliner di Kabupaten Paser. Kasus ini hampir mirip dengan kasus yang diambil oleh Siprianus Tago.

Untuk kasus yang berbeda dilakukan oleh Andri Permana Wicaksono, Arna Fariza, dan Arif Basofi dari Universitas Pekalongan dan Universitas Diponegoro. Mereka mengambil kasus mengenai Sistem Informasi Geografis Pemetaan Jalur Kereta Api Dan Analisa Traffic. Kasus yang diambil ini merupakan suatu contoh kasus yang dinamis. Sedangkan penelitian yang sedang saya lakukan ini merupakan contoh kasus yang juga dinamis dengan mengambil tema mengenai gangguan perjalanan kereta api di wilayah DAOP 6.

No	Penulis	Metode	*Tools*	Studi Kasus
1	Siprianus Tago	*Layanan berbasis lokasi*	Google Maps Api, PHP, AJAX, MySQL	Pencarian Asrama Mahasiswa Kalimantan di Yogyakarta
2	Yudi Rahman	*Layanan berbasis lokasi*	PHP, XML, MySQL, Google Maps API.	Informasi Bangunan, Wisata, Kuliner di Kabupaten Paser
3	Andri Permana Wicaksono, Arna Fariza, Arif Basofi	*Layanan berbasis lokasi*	ArcView, MapServer, QuantumGIS, PostgrSQL, PostGIS	Pemetaan Jalur Kereta Api dan Analisa Traffic
4	Sandra Al Komari	*Layanan berbasis lokasi*	HTML, PHP, MySQL, Google Maps API	Pengelolaan Gangguan Kereta Api Di DAOP 6

Tabel 2.1 Perbandingan penelitian sejenis yang pernah dilakukan.

2.2 Dasar Teori

2.2.1 Sistem Layanan Berbasis Lokasi

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

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

Ada dua tipe layanan yang bisa digunakan dalam LBS untuk memperoleh posisi pengguna, yaitu dengan menggunakan posisi sel jaringan atau GPS maupun aGPS. Dari kedua cara ini akan didapatkan posisi pengguna dalam bentuk koordinat *latitude* dan *longitude*. *Latitude* adalah representasi dari arah Utara-Selatan, sedangkan *longitude* adalah representasi dari arah Timur-Barat.

Selain dari sisi posisi pengguna, LBS juga bisa dilihat dari sisi layanan yang diberikan. Dari sisi layanan yang diberikan, LBS bisa dibagi menjadi dua yaitu *reactive* LBS dan *proactive* LBS. *Reactive* LBS adalah layanan yang hanya aktif jika ada aksi yang dilakukan pengguna. Layanan yang hanya akan memberi jika ada permintaan dari pengguna. Sedangkan *proactive* LBS merupakan layanan yang akan selalu memberi informasi kepada pengguna walaupun pengguna tidak melakukan permintaan terhadap layanan.

2.2.2 Google Maps API

API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan *programmer* untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan *system function*. Proses ini dikelola

melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh Google Maps yang terdiri dari HTML, Javascript, dan AJAX serta XML, memungkinkan untuk menampilkan peta Google Maps di *website* lain.

Google juga menyediakan layanan Google Maps API yang memungkinkan para pengembang untuk mengintegrasikan Google Maps ke dalam *website* masing-masing dengan menambahkan data poin sendiri. Dengan menggunakan Google Maps API, Google Maps dapat ditampilkan pada *website eksternal*. Agar aplikasi Google Maps dapat muncul di *website* tertentu, diperlukan kode unik yang digenerasikan oleh Google untuk suatu *website* tertentu, agar *server* Google Maps dapat mengenali.

Pada Google Maps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

a. *ROADMAP*, untuk menampilkan peta biasa 2 dimensi.
b. *SATELLITE*, untuk menampilkan foto satelit.
c. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
d. *HYBRID*, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota).

2.2.3 Pengertian PHP

PHP adalah kependekan dari *PHP Hypertext Preprocessor*, bahasa interpreter yang mempunyai kemiripan dengan bahasa C dan Perl yang mempunyai kesederhanaan dalam perintah, yang digunakan untuk pembuatan aplikasi web. (Sidik, 2004, h:3)

PHP/F1 merupakan nama awal dari PHP (*Personal Home Page* / *Form Interface*). Dibuat pertama kali oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web. Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses *update* dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.

PHP merupakan *script* untuk pemrograman *script* web *server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

Setiap program PHP disebut dengan *script*. *Script* berupa file teks, yang dapat dibuat dengan menggunakan program editor file teks biasa seperti notepad, edit, vi (dalam lingkungan Unix/linux), atau lainnya. Editor teks yang digunakan sebaiknya editor teks yang memungkinkan membuat program PHP lebih mudah.

Script PHP diawali dengan tag (<?) dan diakhiri dengan tag (?>). Setiap baris perintah / statement harus diakhiri dengan menggunakan tanda titik koma (;). Umumnya setiap statement dituliskan dalam satu baris. *Script* PHP merupakan *script* yang digunakan untuk menghasilkan halaman-halaman web.

2.2.4 *Database Server* dengan *MySQL*

MySQL adalah sebuah sistem manajemen database relasi (*relational database management system*) yang bersifat *open source*. (Arbie, 2004: 5) MySQL merupakan buah pikiran dari Michael "Monty" Widenius, David Axmark dan Allan Larson yang di mulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia.

Pengertian MySQL menurut MySQL manual adalah sebuah *open source software database SQL* (*Search Query Language*) yang menangani sistem manajemen *database* dan sistem manajemen *database relational*. MySQL adalah *open source software* yang dibuat oleh sebuah perusahaan Swedia yaitu MySQL AB. MySQL mempunyai fitur-fitur yang sangat mudah dipelajari bagi para penggunanya dan dikembangkan untuk menangani *database* yang besar dengan waktu yang relatif lebih singkat. Kecepatan, konektivitas dan keamanannya yang lebih baik membuat MySQL sangat dibutuhkan untuk mengakses *database* di internet.

MySQL versi 1.0 di rilis pada Mei 1996 dan penggunaannya hanya terbatas di kalangan perusahaan saja. Barulah pada bulan Oktober 1996, MySQL versi 3.11.0 di rilis ke masyarakat luas. MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. MySQL memiliki kinerja, kecepatan proses dan ketangguhan yang tidak kalah dibanding database-database besar lainnya yang komersil seperti ORACLE, Sybase, Unify dan sebagainya. MySQL dapat berjalan di atas banyak sistem operasi seperti Linux, Windows, Solaris, FreeBSD, Mac OS X, dan lain sebagainya.

2.2.5 Diagram Alir Data (DAD)

Menurut Pressman (2002:364), Diagram Aliran data atau Data Flow Diagram adalah sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output. DFD dapat digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap tingkat abstraksi. Kenyataannya, DFD dapat dipartisi ke dalam tingkat-tingkat yang merepresentasikan aliran informasi yang bertambah dan fungsi ideal. DFD tingkat 0 yang disebut juga dengan model sistem fundamentasi atau model konteks, merepresentasikan seluruh elemen sistem sebagai sebuah bubble tungal dan keluar secara berurutan. Proses tambahan (bubble dan jalur aliran informasi direpresentasikan pada saat DFD tingkat 0 dipartisi untuk mengungkap detail yang lebih. Contohnya DFD tingkat 1 dapat berisi 5 atau 6 bubble dengan anak panah yang saling menghubungkan. Setiap proses yang direpresentasikan pada tingkat 1 merupakan subfungsi dari seluruh sistem yang digambarkan di dalam model konteks.

Simbol yang akan pada diagram aliran data atau data flow diagram antara lain :

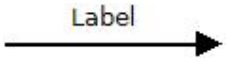
a. Kesatuan Luar (*External Entity*)

   Kesatuan Luar (*External Entity*) merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain. Lingkungan luar dan batas sistem dikonotasikan dalam simbol yang sama.

Gambar 2.1 Notasi entitas

b. Arus Data (*Data Flow*)

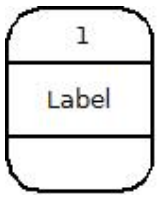
Arus Data (*Data Flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpan data (*data store*) dan kesatuan luar (*external entity*).



Gambar 2.2 Notasi Arus Data

c. Proses (*Process*)

Proses adalah kegiatan oleh manusia, mesin atau komputer yang bertujuan mengolah data yang dimasukan menjadi keluaran yang diinginkan.



Gambar 2.3 Notasi Proses

d. Simpanan Data

Simpanan data disimbolkan dengan sepasang garis horizontal paralel. Simpanan data adalah tempat menyimpan atau untuk mengakses data yang diperlukan oleh sistem.

Gambar 2.4 Notasi Simpanan Data

2.2.6 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) atau disebut dengan R-R secara grafis menggambarkan isi secara database. Database merupakan kumpulan file yang saling berhubungan. Diagram ini memiliki dua komponen utama yaitu entity dan relasi. Pada model *relation* antara file direlasikan dengan kunci relasi yang merupakan kunci uatama dari masing-masing file. Untuk melambangkan fungsi diatas maka digunakan simbol-simbol yang bisa dilihat pada daftar simbol.

Komponen utama pembentuk model *Entity Relationship* adalah entitas (*entity*) dan relasi (*relation*). *Entity* merupakn individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain, sedangkan karekteristik dari entitas didefinisikan oleh suatu atribut / property. Relasi menunjukan adanya hubungan diantara sebuah entitas yang berasa dari himpunan entitas yang berbeda.