

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Mengacu pada karya ilmiah yang sebelumnya sudah pernah dibuat, yaitu diantaranya :

Parameter Penulis	Objek	Metode	Bahasa Pemrograman	Interface
Hakim M (2007)	Kontrakan rumah dan Kost	Survei, Dokumentasi	PHP, MySQL	GUI
Irma (2015)	Rumah kost	Wawancara dan Kuisisioner	PHP, MySQL	GUI
El vera Rosliyanti (2015)	Kamar Hotel	Survei, Observasi, Dokumentasi	PHP, MySQL	GUI
Surji Bambang Haryanto (2015)	Kamar Hotel	Survei, Dokumentasi	PHP, MySQL	GUI
Chandra Arief Syuada (2016)	Rumah Kost	Surver, Observasi dan Dokumentasi	PHP, MySQL	GUI

2.2 Dasar Teori

2.2.1 Metode *Heversine*

Metode Heversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*latitude*) sebagai variable inputan. Heversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan garis bujur dan garis lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari – jari (R) = 6.371 km dan lokasi dari titik koordinat bola masing – masing adalah $lon1$, $lat1$, $lon2$, dan $lat2$, maka rumus heversine dapat dituliskan sebagai berikut :

Rumus *heversine* (sumber : *Gus Pascima*)

$$x = (lon2 - lon1) * \cos((lat1 + lat2) / 2) \dots \dots \dots \text{persamaan 1}$$

$$y = (lat2 - lat1) \dots \dots \dots \text{persamaan 2}$$

$$d = \sqrt{x^2 + y^2} * R \dots \dots \dots \text{persamaan 3}$$

Keterangan:

x = Longitude(Lintang)

y = Latitude(Bujur)

d = Jarak

R = RadiusBumi = 6.371 km

1 derajat= 0.0174532925 radian

Gambar 2.1 Rumus heversine

2.2.2 Sekilas Pengertian *Framework CodeIgniter*

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model *MVC* (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal. *CodeIgniter* dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.4.

2.2.3 *Framework*

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

Ada beberapa alasan mengapa menggunakan *Framework*:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada).
3. Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya *validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling*, dll).
4. Lebih bebas dalam pengembangan jika dibandingkan *CMS*.

2.2.4 *MVC (Model View Controller)*

Model View Controller (MVC) merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, *MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu *MVC* pattern dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diaturoleh controller. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert, update, delete, create*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip *MVC* suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer yang menangani bagian model dan controller, sedangkan designer yang menangani bagian *view*,

sehingga penggunaan arsitektur *MVC* dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan. Ada beberapa kelebihan *CodeIgniter* (CI) dibandingkan dengan *Framework PHP* lain.

1. Performa sangat cepat : salah satu alasan tidak menggunakan framework adalah karena eksekusinya yang lebih lambat dari pada PHP, tapi *Codeigniter* sangat cepat bahkan mungkin bias dibilang *codeigniter* merupakan *framework* yang paling cepat disbanding *framework* yang lain.
2. Konfigurasi yang sangat minim (*nearly zero configuration*) : tentu saja untuk menyesuaikan dengan database dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database. Php* atau *autoload.php*, namun untuk menggunakan *codeigniter* dengan setting standard, anda hanya perlu mengubah sedikit saja file pada folder *config*.
3. Banyak komunitas: dengan banyaknya komunitas *CI* ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau berbagi informasi teknologi terbaru.
4. Dokumentasi yang sangat lengkap : Setiap paket instalasi *codeigniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk di

jadikan permulaan, bahasanya pun mudah dipahami. (EllisLab 28
Februari 2006)