

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Rumah Kos**

Rumah kos adalah tempat yang disediakan untuk memfasilitasi wanita atau pria, pelajar, mahasiswa, dan pekerja yang tinggal dan menetap dalam kurun waktu tertentu dan dengan harga yang telah disepakati. Umumnya, pembayaran rumah kos dapat dilakukan per bulan, per beberapa bulan atau per tahun. Rumah kos berfungsi sebagai tempat tinggal. Rumah kos sering dijumpai di sekitar perkantoran, universitas, sekolah-sekolah, dan juga pusat-pusat kegiatan lainnya.

#### **2.2 Android**

Menurut Yuni Puspita Sari (Jurnal Informatika Vol.19, No.1 Juni 2019 ) *Android* adalah sebuah sistem operasi untuk *smartphone* dan *Tablet*. Sistem operasi dapat diilustrasikan sebagai 'jembatan' antara peranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. *Mobile phone* adalah salah satu perangkat yang bergerak seperti telepon seluler atau komputer bergerak yang digunakan untuk mengakses jasa jaringannya. Pada *mobile application* juga digunakan untuk mendeskripsikan aplikasi internet yang berjalan pada *smartphone* serta peranti *mobile* lainnya.

#### **2.3 MySQL**

*MySQL* adalah suatu sistem relational *database* yang menyimpan data pada tabel berbeda dan tidak meletakkannya pada satu tabel saja. Hal ini meningkatkan kecepatan dan fleksibilitas. Tabel tersebut dihubungkan dengan suatu relasi yang didefinisikan sehingga dapat mengkombinasikan data dari beberapa tabel pada satu saat. *MySQL* menggunakan bahasa standar *Structure Query Language*, yaitu bahasa standar yang paling banyak digunakan untuk mengakses *database*. *MySQL* sebagai

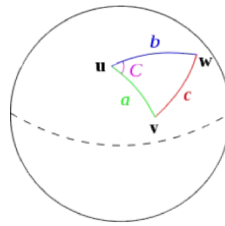
*database server* yang memiliki konsep *database* modern dan memiliki banyak sekali keistimewaan. Dukungan penggunaan banyak tersedia, ada banyak grup diskusi tentang *MySQL* yang ditawarkan di situs *MySQL*, atau anda dapat juga bertanya lewat *MySQL AB* dengan sedikit biaya. Berikut ini beberapa kelebihan *MySQL* sebagai *database server* antara lain:

1. *Source MySQL* dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit
3. Pengaksesan *database* dapat dilakukan dengan mudah.
4. *MySQL* merupakan program yang *multithreaded*, sehingga dapat dipasang pada *server* yang memiliki multi CPU
5. Didukung program-program umum seperti *C*, *C++*, *Java*, *Perl*, *PHP*, *Python*, dsb.
6. Bekerja pada berbagai *platform*. (tersedia berbagai versi untuk berbagai sistem operasi).
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
8. Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*
9. Mendukung ODBC untuk sistem operasi Windows.
10. Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

*MySQL* dan *PHP* merupakan sistem yang saling terintegrasi. Maksudnya adalah pembuatan *database* dengan menggunakan sintak *PHP* dapat di buat. Sedangkan *input* yang di masukkan melalui aplikasi web yang menggunakan *script serverside* seperti *PHP* dapat langsung dimasukkan ke *database MySQL* yang ada di *server* dan tentunya web tersebut berada di sebuah *web server*.

## 2.4 *Haversine Formula*

Teorema *Haversine Formula* adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dengan *longitude* dan *latitude* sebagai inputannya. Ini merupakan bentuk persamaan khusus dari trigonometri bola, *law of haversines*, mencari hubungan sisi dan sudut pada segitiga dalam bidang bola.



**Gambar : 2.1** *Segitiga bola hukum haversine formula*

*Haversine Formula* merupakan suatu metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan.

Penggunaan rumus ini mengasumsikan pengabaian efek *ellipsoidal*, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah dipermukaan bumi. *Haversine Formula* nantinya akan digunakan dalam perhitungan jarak antara dua titik *GPS*. Dalam hal ini adalah titik *GPS* pengguna dan titik *GPS* tujuan, titik *GPS* ini berisikan *latitude* dan *longitude*, sehingga dapat menjadi kunci utama dalam perbandingan jarak pada penentuan lokasi jasa jahit terdekat. Berikut rumus *Haversine* yang akan di implementasikan kedalam skripsi ini. Berikut adalah rumus *haversine* pada persamaan

$$\Delta lat = lat2 - lat1$$

$$\Delta long = long2 - long1$$

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2\left(\frac{\Delta long}{2}\right)$$

$$c = 2 a \tan 2\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

Dimana :

$\Delta lat$  = besaran perubahan *latitude*

$\Delta long$  = besaran perubahan *longitude*

$R$  = jari-jari bumi sebesar 6371(Km)

$c$  = kalkulasi perpotongan sumbu

$d$  = jarak (Km)

1 derajat = 0,0174532925 radian

Untuk memperjelas penggunaan rumus *Haversine* di atas, berikut penjabaran rumus yang nantinya akan digunakan pada perhitungan manual *Haversine Formula*. Dari rumus di atas untuk menghitung jarak antara dua titik dapat di tulis dengan persamaan berikut :

$$d = R.c$$

Untuk mendapatkan nilai  $c$  , digunakan rumus sebagai berikut :

$$c = 2 a \tan 2 (\sqrt{a}, \sqrt{1-a})$$

Dimana untuk mendapatkan nilai  $a$ , digunakan rumus sebagai berikut:

$$a = \sin^2 \left( \frac{\Delta lat}{2} \right) + \cos(lat1). \cos(lat2). \sin^2 \left( \frac{\Delta long}{2} \right)$$

$\Delta lat$  merupakan perhitungan dari  $\frac{\pi}{180}$  (*latitude tujuan* – *latitude pengguna*)

$lat1$  merupakan *latitude* pengguna sedangkan  $lat2$  adalah posisi *latitude* lokasi tujuan,

$\Delta long$  merupakan perhitungan dari  $\frac{\pi}{180}$  (*longitude tujuan* – *longitude pengguna*)

Dari perhitungan di atas, untuk menentukan jarak paling dekat dengan lokasi rumah kos harus terdapat pembanding. Jadi harus terdapat banyak titik lokasi dan jarak, barulah dibandingkan jarak terpendek. Untuk jalur jalan yang sebenarnya sesuai kondisi memanfaatkan *google maps*, lalu akan ditampilkan *marker* antara lokasi pengguna dan lokasi tujuan serta informasi jalan mana saja yang di lalui. Perhitungan *Haversine* dilakukan didalam *query sql* yang dikerjakan oleh *system*, dan hasilnya akan di jadikan sebuah JSON, sehingga aplikasi tidak berkerja dengan berat.

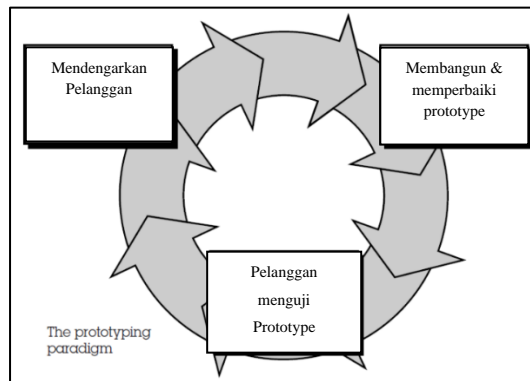
## 2.5 *Google Map API*

*Google Map API* adalah suatu *library* yang berbentuk *Javascript*. Cara membuat *Google Map* untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *Javascript*, serta koneksi Internet yang sangat stabil. Menggunakan *Google Map API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik *Google* sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia. Melalui *Google Map API* pengguna mendapatkan *tools* pemrograman yang kuat untuk membuat peta yang interaktif, akses ke *database Google Earth* dan kesempatan untuk pengembangan antarmuka pengguna yang akan digunakan untuk melakukan operasi pencarian melalui *address geocoding*, visualisasi data geografis, menerapkan analisis dan konsultasi.

## 2.6 Metodologi Terkait

Yuni Puspita Sari (2016) menguraikan bahwa model *Prototyping* paradigma dimulai dengan pengumpulan kebutuhan pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar dimana definisi-definisi lebih jauh

merupakan keharusan, kemudian dilakukan perancangan kilat, di akhiri dengan evaluasi *prototyping*, dapat dilihat pada gambar berikut ini.



**Gambar 2.2 Model Prototype**

Tahapan pada model *prototyping* dapat di jelaskan sebagai berikut ini :

1. Mendengarkan pelanggan

Tahap pertama dari model ini adalah mendengarkan pelanggan yaitu mengumpulkan data-data mengenai kebutuhan sistem yang akan dibuat. Tahap ini diawali dengan mengumpulkan kebutuhan yang dibutuhkan untuk melakukan proses penjadwalan kegiatan penelitian.

2. Membangun atau memperbaiki *Mock-up*

Setelah mendapat kebutuhan yang dibutuhkan untuk membuat *prototype*, tahap berikutnya yang dilakukan adalah membangun dan memperbaiki *prototype*, tahap ini dilakukan dengan pembuatan sistem secara keseluruhan hingga selesai berdasarkan pada analisis kebutuhan yang sudah dilakukan sebelumnya.

3. Pelanggan melihat atau menguji *Mock-up*

Tahap akhir dari model *prototype* ini adalah tahap pengujian (*testing*). Pengujian dilakukan oleh pengguna dari sistem. Tahap pengujian dilakukan untuk mendapatkan tanggapan atas sistem yang telah dibuat.

### 2.6.1 Kelebihan Model *Prototype*

Kelebihan dari metode *prototyping* ini sebagai berikut:

- 1) Adanya komunikasi yang baik antara pengembang dan pelanggan.
- 2) Pengembangan dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- 3) Lebih menghemat waktu dalam pengembangan sistem.
- 4) Penerapan menjadi lebih mudah karena pemakaian mengetahui apa yang diharapkannya.

### 2.6.2 Kekurangan Model *Prototype*

kekurangan dari model *prototype* ini adalah sebagai berikut:

- 1) Risiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
- 2) Interaksi pemakai penting. Sistem harus menyediakan dialog online antara pelanggan dan komputer.
- 3) Hubungan pelanggan dengan komputer yang disediakan mungkin tidak menceritakan teknik perancangan yang baik.

## 2.7 Penelitian Terdahulu

Ada beberapa penelitian berkaitan permasalahan yang penulis angkat, diantaranya:

**Tabel 2.1** *Penelitian Terdahulu*

No	Nama	Judul	Terbit/Tahun	Keterangan
1	Pratama Rezalendra Putra	Aplikasi Persebaran Rumah Kos Berbasis Android (Studi Kasus: Kota Malang)	Skripsi tesis, ITN Malang/2019	Penelitian ini membahas mengenai persebaran rumah kos berbasis android kota malang dirancang sebagai alat bantu untuk

				mempermudah masyarakat dalam memasarkan dan mencari informasi tentang rumah kos yang ada di kota malang.
2	Rika Oviani	Perancangan Aplikasi Sistem Informasi Geografis Pencarian Rumah Kos Dan Kontrakan Sekitaran Kampus Di Kota Jambi Berbasis Android	Skripsi tesis, STIKOM Dinamika Bangsa Jambi/2019	Aplikasi ini hanya membahas tentang mencari rumah kos dan rumah kontrakan diwilayah sekitar Kampus STIKOM Dinamika Bangsa dan UNBAR dirancang sebagai alat bantu untuk mempermudah masyarakat dan mahasiswa dalam memasarkan dan mencari informasi tentang rumah kos yang ada sekitar Kampus STIKOM Dinamika Bangsa dan UNBAR

Kelebihan penelitian di atas adalah :

1. Menggunakan Metode *Haversine Formula* sehingga penentuan jarak terdekat lebih akurat mencapai 98,66%.
2. Sudah ada fitur rekomendasi rumah kos yang ada disekitar pengguna.
3. Cakupan wilayah lebih luas.