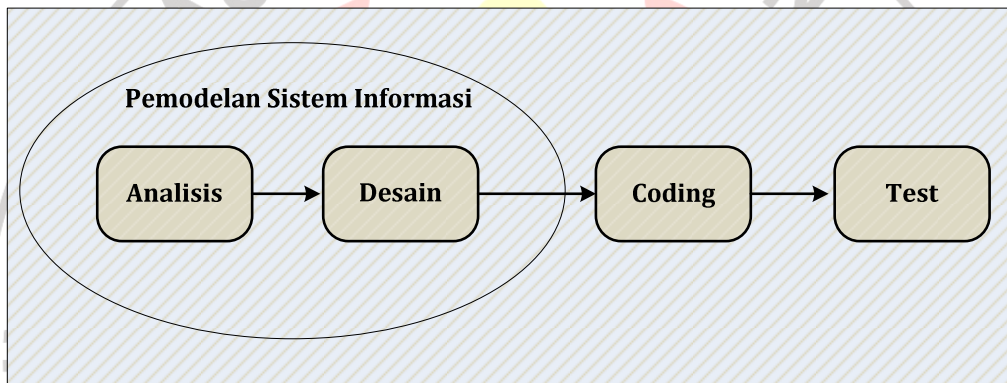


## BAB III METODOLOGI PENELITIAN

### 3.1. Desain Penelitian

Pengembangan Aplikasi Pencarian Rute Terpendek Menggunakan Algoritma A\* dan Dijkstra ini menggunakan model *waterfall*. Model *waterfall* penelitian untuk sistem ini dapat dilihat pada gambar berikut ini:



Gambar 3.1. Desain Penelitian

Model *waterfall* penelitian Aplikasi Pencarian Rute Terpendek Menggunakan Algoritma A\* dan Dijkstra ini melingkupi aktivitas-aktivitas berikut ini

1. Pemodelan Sistem Informasi

Pemodelan sistem informasi harus dilakukan terlebih dahulu sebelum mulai melakukan implementasi program atau pengkodean program. Pemodelan sistem informasi ini bertujuan untuk menemukan batasan-batasan masalah pada penerapan sistem

Pemodelan sistem informasi ini terdiri dari 2 tahap yaitu

a. Analisis Kebutuhan

Mencari semua kebutuhan yang diperlukan dalam pembuatan sistem informasi dan pembuatan dokumen teknis yang nantinya akan dibaca oleh pengguna sistem.

a. Desain Sistem

Proses desain ini bertujuan untuk menerjemahkan hasil analisis kebutuhan ke dalam representasi perangkat lunak. Empat atribut yang menjadi fokus desain sistem adalah: struktur data, arsitektur perangkat lunak, representasi *interface* dan detail atau spesifikasi proses.

2. Implementasi Program

Implementasi program adalah proses mengonversi desain sistem informasi ke dalam bentuk bahasa pemrograman yang dimengerti oleh mesin.

Implementasi program tidak boleh melebihi dari apa yang telah ditentukan dalam desain perangkat lunak.

3. Uji Coba dan Evaluasi

Uji coba dan evaluasi sistem berfokus pada logika internal sistem informasi. Proses uji coba sistem dilakukan dengan dua cara yaitu:

*blackbox testing* dan *whitebox testing*.

### 3.2. Fokus Penelitian

Fokus masalah dalam penelitian ini adalah proses pembuatan sistem *routing* yang akan dilakukan oleh sistem serta mengukur tingkat akurasi dari pencarian rute terpendek yang diimplementasikan pada sebuah peta digital berbasis *WEBGIS* yang pada kasus ini menggunakan peta lingkungan kampus UPI Bandung. Proses *routing* dibuat dengan cara menghitung jarak terpendek antar *node* awal menuju *node* tujuan dengan menggunakan dua cara, yaitu dengan Algoritma Dijkstra dan Algoritma A\* (*A Star*). Lalu dari pengujian kedua algoritma, akan diketahui perbedaan proses pencarian diantara kedua Algoritma tersebut.

### 3.3. Metode Penelitian

Metode pengumpulan data yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Studi Pustaka

Mempelajari literatur berupa artikel, *paper* maupun sumber lain yang mendukung yang berkaitan dengan objek penelitian.

2. Observasi

Melakukan pengumpulan data dan informasi serta mengamati aplikasi-aplikasi pencarian rute berbasis *web* yang sudah ada sebagai bahan referensi.

### 3. Forum *Online*

Untuk mendapatkan data seakurat mungkin, proses tanya jawab perlu dilakukan secara langsung dengan pihak/komunitas terkait yang berhubungan dengan aplikasi ini agar aplikasi yang dibuat sesuai dengan kebutuhan.

## 3.4. Alat dan Bahan Penelitian

### 3.4.1. Alat Penelitian

#### 1. Perangkat Keras (*Hardware*) SIG yang digunakan :

- a) CPU : Processor 2.26 Gb
- b) *Hardisk* : Kapasitas 20 Gb
- c) *Memory* : Minimal 512 Mb
- d) Monitor : Resolusi 1024 x 768 dengan 32 bit color

#### 2. Perangkat Lunak (*Software*) SIG yang digunakan :

- a) Sistem Operasi : Windows XP
- b) Pemrograman : HTML, PHP dan JavaScript
- c) *Web Server* : Apache
- d) *Database* : PostgreSQL, PostGIS, PgRouting
- e) *Tools* : QuantumGIS, *Mapserver*, Dreamweaver, Openlayers

### 3.4.2. Bahan Penelitian

#### 1. Data Spasial:

- Peta jaringan jalan di lingkungan kampus UPI
- Peta objek gedung di lingkungan kampus UPI
- Peta kumpulan *node* persimpangan dan pintu masuk gedung di lingkungan kampus UPI.

#### 2. Data Atribut:

- Data jaringan jalan di lingkungan kampus UPI ( nama, panjang, arah, kordinat )
- Data objek-objek gedung di lingkungan kampus UPI (nama, keterangan, kordinat)
- Data kumpulan *node* persimpangan dan pintu masuk gedung di lingkungan kampus UPI (nama, keterangan, kordinat)

### 3.5. Tahapan Penelitian

Dalam mengembangkan aplikasi pencarian rute terpendek ini, ada beberapa tahap yang harus dikerjakan, berikut tahapannya :

#### 1. Mempelajari teknik pendigitasian peta digital

*Software* Quantum GIS merupakan salah satu dari sekian banyak *tools* untuk membuat sebuah peta digital. Sumber belajar mengenai teknik ini mudah ditemukan di internet.

## 2. Menentukan algoritma pencarian

Dari sekian banyak algoritma pencarian yang ada, peneliti memilih Algoritma A\* dan Algoritma Dijkstra untuk melakukan proses *routing* pada aplikasi ini. Hal ini dikarenakan kedua algoritma ini mempunyai proses yang berbeda dalam melakukan sebuah pencarian rute, Namun pada kenyataannya, hasil pencarian dari algoritma A\* sama dengan hasil pencarian algoritma Dijkstra, dan sudah banyak artikel, paper dan tutorial yang membahas kedua algoritma ini. Dengan demikian, peneliti dapat memperoleh kemudahan dalam membuat proses *routing*. Seperti paper (<http://digilib.its.ac.id/public/ITS-Undergraduate-9123-2205100111-Paper.pdf>) membahas algoritma A\* dan *paper* yang membahas Dijkstra (<http://journal.uui.ac.id/index.php/Snati/article/viewFile/1926/1701>).

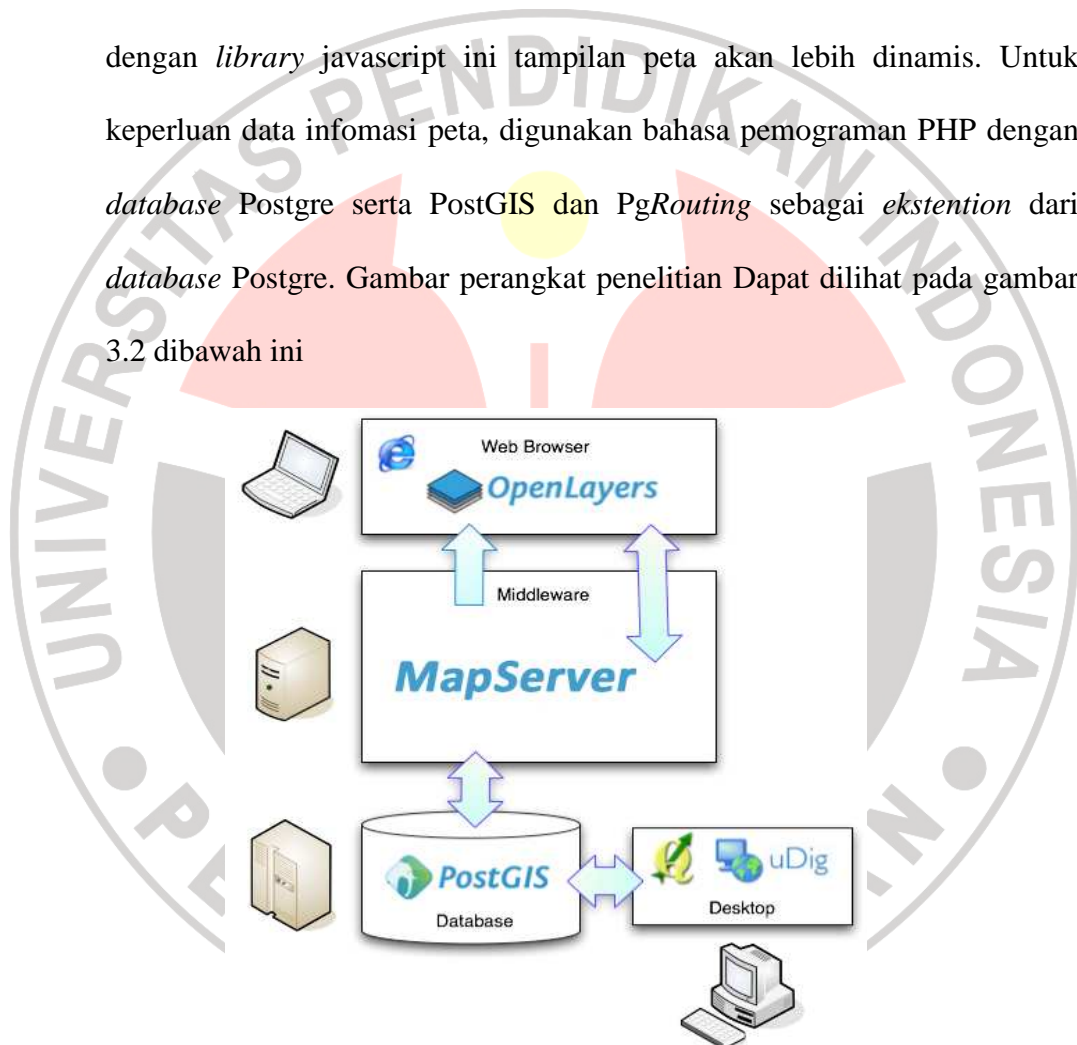
## 3. Mengumpulkan data spasial dan data atribut

Peneliti menggunakan gambar dari google maps untuk meng-*capture* tampilan wilayah lingkungan kampus UPI untuk data spasial yang nantinya gambar itu akan didigitasi menggunakan *tools* Quantum GIS. Sedangkan untuk data atribut, peneliti memperoleh data dari internet, segala informasi tentang objek yang ada di lingkungan kampus UPI untuk keperluan aplikasi.

## 4. Menyiapkan perangkat penelitian

Peneliti membuat suatu aplikasi pencarian rute berbasis *WEBGIS* sederhana untuk menerapkan Algoritma A\* dan Algoritma Dijkstra yang akan dibuat. Semua *tools* yang peneliti pakai untuk mengembangkan aplikasi ini bersifat

*open source*, ini merupakan salah satu kelebihan aplikasi yang peneliti kembangkan. Pertama Sistem ini dibuat menggunakan *Mapserver* sebagai *MapEngine* untuk menampilkan peta kampus UPI yang sudah didigitasi dalam sebuah *brower*, lalu untuk tampilan peta pada *browser* peneliti menggunakan *openlayers*, *Openlayers* merupakan sebuah *library javascript*, dengan *library javascript* ini tampilan peta akan lebih dinamis. Untuk keperluan data informasi peta, digunakan bahasa pemograman PHP dengan *database Postgre* serta *PostGIS* dan *PgRouting* sebagai *ekstention* dari *database Postgre*. Gambar perangkat penelitian Dapat dilihat pada gambar 3.2 dibawah ini



**Gambar 3.2.** Gambar perangkat penelitian

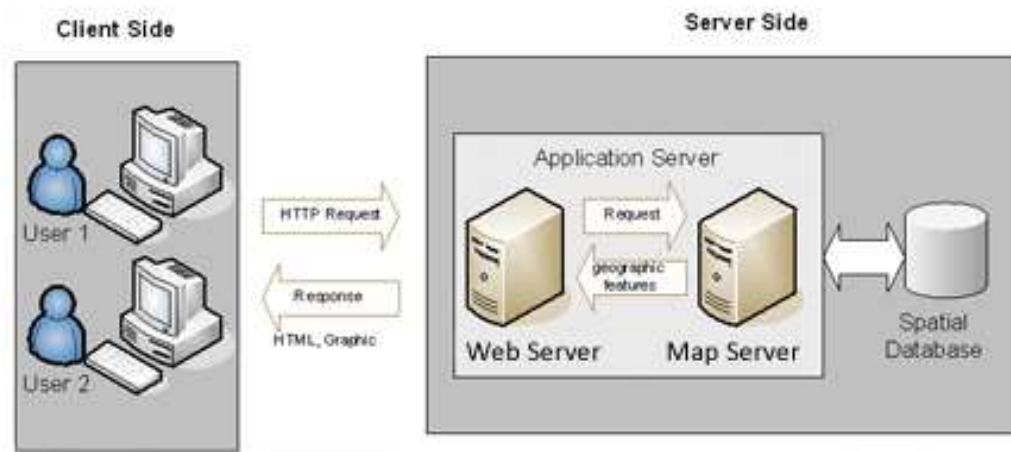
- Kelebihan dari aplikasi ini adalah semua *tools* yang dipakai untuk membuat aplikasi ini bersifat *open source*, tentunya untuk mengembangkan aplikasi ini kita tidak perlu menggunakan *software* yang berbayar.



## 5. Pengembangan Sistem *Routing*

Peneliti dalam mengembangkan sistem *routing* pada aplikasi ini, memanfaatkan *ekstention* PostGIS dari *database* Postgre, PostGIS memungkinkan peneliti untuk menyimpan data spasial sebuah peta dalam *database*, Sedangkan PgRouting mempunyai fungsi *routing* (penghitungan jarak terpendek dari data *polyline*) pada PostGIS. Aplikasi ini dibuat ini dibuat menggunakan fungsi *shortest path* yang dimiliki modul pgRouting, yang merupakan fungsi tambahan dari PostgreSQL/PostGIS untuk menangani masalah *routing* pada aplikasi pencarian rute berbasis *webgis*.

## 6. WebGIS System



**Gambar 3.3** Arsitektur WEBGIS

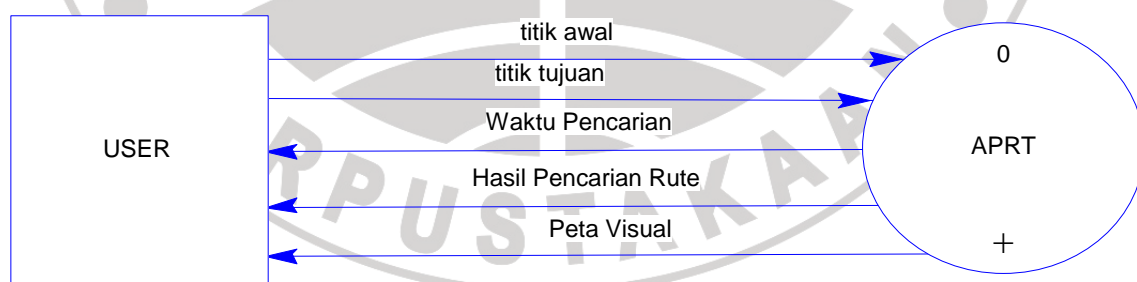
Gambar diatas menunjukkan arsitektur minimum sebuah sistem *WebGIS*. Aplikasi berada disisi *client* yang berkomunikasi dengan *Server* sebagai penyedia data melalui *web Protokol* seperti HTTP (*Hyper Text Transfer Protocol*). Aplikasi seperti ini bisa dikembangkan dengan *web browser* (Mozilla Firefox, Opera,



Internet Explorer, dll). *Web Server* bertanggung jawab terhadap proses permintaan dari *client* dan mengirimkan tanggapan terhadap respon tersebut. Dalam arsitektur *web*, sebuah *web server* juga mengatur komunikasi dengan *server side* GIS. Komponen yaitu Mapserver. MapServer bertindak sebagai perantara (*interface*) antara data dengan *server* aplikasi dan bertanggung jawab terhadap koneksi kepada *database* spasial seperti membuat representasi yang diteruskan ke *server*. Selain itu komponen ini berfungsi sebagai penerjemah dan pengakses data spasial (baik dalam format *raster* maupun *vector*), pelaku analisis spasial, me-render peta digital hingga siap dipublikasikan di *web user*. Komponen Mapserver inilah yang paling membedakan suatu produk *webbased GIS* dengan yang lain.

### 7. Implementasi

Deskripsi umum penerapan sistem dapat dilihat pada *context diagram* berikut Di bawah ini:



**Gambar 3.4** Contexts Diagram Aplikasi Pencarian Rute Terpendek

Dari Konteks diagram di atas dapat dilihat Aplikasi Pencarian Rute Terpendek ini akan memberikan *output* berupa rute terpendek, *visualisasi* rute terpendek, informasi objek dan perbandingan algoritma setelah *user* memberi *input* berupa titik awal dan titik tujuan.

Penelitian ini mencoba untuk membandingkan algoritma Dijkstra dan algoritma A\* yang diimplementasikan dalam sebuah peta digital berbasis *WEBGIS*. Mengembangkan sebuah sistem *routing* yang diterapkan untuk mencari rute tercepat pada jaringan jalan. Jaringan jalan yang digunakan untuk sistem ini berasal dari data spasial, pada kasus ini kita menggunakan jalan di lingkungan kampus UPI. Data spasial berubah menjadi format sql yang mendukung kemampuan akses sebagai *WEBGIS*. Setelah semua data telah disiapkan dan dikonversi ke format yang sesuai, kita mencoba untuk merancang antarmuka *web*. Dengan menggunakan Sistem Informasi Geografis (SIG) maka akan lebih mudah untuk memvisualisasikan hasil pencarian rute ini. Karena dengan adanya SIG maka akan digambarkan juga posisi penyebaran data pada kondisi sesungguhnya.

