

## **BAB 1. PENDAHULUAN**

### **1.1 Latar Belakang**

Rumah sakit memiliki peranan penting dalam menangani masalah kesehatan masyarakat, mengingat kesehatan merupakan salah satu kebutuhan pokok manusia. Dengan banyaknya rumah sakit yang ada di kabupaten Jember, maka cukup membingungkan masyarakat yang berasal dari luar kabupaten Jember dalam memilih rumah sakit yang akan dituju agar sesuai dengan kebutuhan dan dapat menghemat waktu perjalanan. Oleh karena itu, dibutuhkan perantara cepat untuk sampai ke lokasi rumah sakit terdekat. Ada banyak cara untuk sampai ke lokasi rumah sakit, salah satunya adalah dengan melihat peta untuk mempermudah pencarian. Selain itu, jika terjadi hal – hal yang mendesak seperti sakit mendadak atau hal – hal lain yang berpotensi membahayakan jiwa seseorang, informasi pelayanan kesehatan yang dimiliki oleh rumah sakit yang akan dituju tentu sangat penting. Namun sebagian dari masyarakat yang berasal dari luar kabupaten Jember tidak mengetahui informasi tersebut, sehingga dibutuhkan pula sistem yang dapat menyarankan rumah sakit terdekat mana saja yang memiliki pelayanan kesehatan yang dibutuhkan oleh pengguna sekaligus rute perjalanan untuk menuju kesana.

Selain masyarakat yang berasal dari luar kabupaten Jember, tak jarang beberapa penduduk asli kabupaten Jember membutuhkan perantara cepat untuk sampai ke lokasi rumah sakit terdekat beserta informasi pelayanannya. Saat berada dalam keadaan tergesa – gesa ataupun panik, pola pikir manusia dalam mencari solusi akan terganggu sehingga tidak tercapai solusi yang maksimal. Seperti saat terjadi kecelakaan, sakit mendadak, ataupun keadaan mendesak lainnya yang mengganggu kesehatan seseorang, maka orang disekitarnya menjadi panik mencari solusi untuk menolong orang tersebut, misalnya mencari lokasi rumah sakit terdekat mana yang memiliki penanganan berupa pelayanan kesehatan yang tepat supaya pertolongan pertama segera

didapatkan. Mengetahui rumah sakit terdekat saja tidak cukup, karena apabila pasien dilarikan ke rumah sakit terdekat, namun di rumah sakit tersebut tidak memiliki pelayanan kesehatan yang dibutuhkan oleh pasien maka pasien harus dilarikan ke rumah sakit lain yang memiliki pelayanan kesehatan yang ia butuhkan, tentu saja hal ini lebih banyak membuang waktu sedangkan kondisi pasien akan semakin melemah.

Berdasarkan permasalahan diatas, maka dibuatlah aplikasi yang dapat membantu masyarakat di kabupaten Jember maupun yang berasal dari luar kabupaten Jember untuk menemukan titik lokasi rumah sakit terdekat yang memiliki pelayanan kesehatan sesuai kebutuhan mereka dan dilengkapi dengan fasilitas rute dengan waktu tercepat untuk menuju ke lokasi rumah sakit tersebut. Peta dan rute yang digunakan dalam aplikasi ini memanfaatkan peta milik *Google*, yakni *Google MAPS API*, sehingga dalam menggunakan sistem ini pengguna diharuskan terkoneksi dengan internet karena peta *Google* yang bersifat *online*. Aplikasi ini akan memudahkan penggunanya menemukan titik lokasi rumah sakit terdekat dengan pelayanan kesehatan yang tepat dari lokasinya. Aplikasi ini akan menyaring rumah sakit – rumah sakit yang memiliki pelayanan kesehatan yang dibutuhkan oleh pengguna, kemudian secara otomatis akan mendeteksi lokasi pengguna menggunakan sensor *Global Positioning System (GPS)* yang ada pada *Android*. Setelah itu, titik lokasi pengguna akan dikirimkan dari *Android* ke *Web View* yang menyediakan data – data yang dibutuhkan oleh aplikasi. Setelah itu, dilakukan perhitungan jarak antara titik lokasi pengguna dengan titik lokasi rumah sakit yang telah disaring sebelumnya sehingga mendapatkan hasil titik lokasi rumah sakit terdekat dari lokasi pengguna dengan pelayanan kesehatan yang dibutuhkan. Hasil tersebut ditampilkan dari *Web View* yang diakses melalui *Android* sehingga tampil di layar *Smartphone* pengguna. Ada banyak metode yang dapat digunakan untuk menyelesaikan perhitungan dalam masalah penentuan titik lokasi terdekat, salah satunya adalah metode *Haversine Formula*. Metode *Haversine Formula* digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*latitude*) sebagai variabel masukan. *Haversine formula* adalah persamaan penting pada

navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Seperti yang sudah dijalankan sebelumnya, aplikasi ini berbasis web yang dirancang untuk perangkat *mobile* dengan sistem operasi *Android* yang sudah familiar di kalangan masyarakat Indonesia, sehingga pengguna tidak repot untuk mengaksesnya. Dengan adanya aplikasi ini, diharapkan dapat mempermudah pengguna untuk mengetahui titik lokasi rumah sakit terdekat beserta informasi pelayanan kesehatan yang ada di rumah sakit tersebut.

## **1.2 Rumusan Masalah**

Berdasarkan penjabaran latar belakang, maka dapat disimpulkan rumusan masalah sebagai berikut :

- a. Bagaimana merancang dan membuat Aplikasi Penentuan Titik Lokasi Rumah Sakit Berdasarkan Pelayanan Kesehatan berbasis web.
- b. Bagaimana metode *Haversine Formula* dapat menentukan titik lokasi rumah sakit terdekat di kabupaten Jember.
- c. Bagaimana memanfaatkan fasilitas dari *Google Maps API* untuk menggambarkan rute perjalanan.

## **1.3 Batasan Masalah**

Adapun batasan masalah dari Tugas Akhir ini adalah sebagai berikut :

- a. Fasilitas kesehatan yang dimaksud adalah rumah sakit yang berada dalam kawasan Kabupaten Jember.
- b. Rute yang ditampilkan menuju rumah sakit berdasarkan pada *Google Direction API* yang diimplementasikan pada *Google MAPS API*.
- c. Aplikasi ini berbasis web dan dirancang untuk perangkat *mobile* dengan sistem operasi *Android* yang familiar di kalangan masyarakat, sehingga mudah untuk digunakan.
- d. Target sistem operasi *Android* yang digunakan adalah versi *Kit Kat* dan sistem operasi *Android* minimum yang dapat mengakses aplikasi ini adalah versi *Jelly Bean*.

- e. Karena situasi yang tidak menentu, maka aplikasi tidak memberikan informasi mengenai keadaan jalan.

#### **1.4 Tujuan**

Tujuan dibuatnya tugas akhir ini adalah sebagai berikut :

- a. Merancang dan membuat Aplikasi Penentuan Titik Lokasi Rumah Sakit Berdasarkan Pelayanan Kesehatan berbasis web yang dapat ditampilkan di *Android*.
- b. Mengimplementasikan metode *Haversine Formula* dalam menentukan titik lokasi rumah sakit terdekat berdasarkan pelayanan kesehatan di kabupaten Jember.
- c. Pemanfaatan fasilitas dari *Google Maps API* untuk menggambarkan rute perjalanan dari lokasi pengguna menuju lokasi rumah sakit yang dipilih.

#### **1.5 Manfaat**

Manfaat dibuatnya Tugas Akhir ini adalah sebagai berikut :

- a. Masyarakat di kabupaten Jember dan masyarakat yang berasal dari luar kabupaten Jember mengetahui titik lokasi rumah sakit terdekat dari lokasinya beserta informasi pelayanan kesehatan yang ada di rumah sakit tersebut.
- b. Mengetahui bagaimana metode *Haversine Formula* dalam menentukan titik lokasi rumah sakit terdekat di kabupaten Jember.

## **BAB 2. TINJAUAN PUSTAKA**

### **2.1 Definisi Lokasi**

Pada studi Geografi, lokasi merupakan variabel penting yang dapat mengungkapkan berbagai hal tentang gejala atau fenomena yang dipelajari. Sumaatmadja (1988:118) menjelaskan, bahwa : lokasi suatu benda dalam ruang dapat menjelaskan dan dapat memberikan kejelasan pada benda atau gejala geografi yang bersangkutan secara lebih jauh lagi. Mempelajari geografi sama artinya dengan mempelajari lokasi-lokasi di muka bumi, dan ketika orang atau individu berbicara tentang lokasi di permukaan bumi maka individu tersebut sedang berbicara mengenai fenomena di permukaan bumi. Lokasi sangat erat kaitannya dengan jarak di permukaan bumi. Suatu gejala akan sangat strategis dan mempunyai nilai guna yang tinggi jika terletak pada lokasi yang menguntungkan. (Definisi pengertian, 2015).

### **2.2 Jember**

#### **2.2.1 Selayang Pandang Jember**

Jember adalah sebuah wilayah kabupaten yang merupakan bagian dari wilayah Provinsi Jawa Timur. Kabupaten Jember berada di lereng Pegunungan Yang dan Gunung Argopuro membentang ke arah selatan sampai dengan Samudera Indonesia. Dalam konteks regional, Kabupaten Jember mempunyai kedudukan dan peran yang strategis sebagai salah satu Pusat Kegiatan Wilayah (PKW).

Provinsi Jawa Timur yang meliputi Wilayah Hinterland Kabupaten Jember, Kabupaten Bondowoso, dan Kabupaten Situbondo. Secara administratif, wilayah Kabupaten Jember berbatasan dengan Kabupaten Bondowoso dan Kabupaten Probolinggo di sebelah utara, Kabupaten Lumajang di sebelah barat, Kabupaten Banyuwangi di sebelah timur, dan di sebelah selatan dibatasi oleh Samudera Indonesia.

Keberadaan Kabupaten Jember secara geografis memiliki posisi yang sangat strategis dengan berbagai potensi sumber daya alam yang potensial, sehingga

banyak menyimpan peristiwa-peristiwa sejarah yang menarik untuk digali dan dikaji. Tentang nama Jember sendiri dan kapan wilayah ini diakui keberadaannya, hingga saat ini memang masih belum diperoleh kepastian fakta sejarahnya. Hari jadi bagi suatu daerah sangatlah penting dan mendasar, karena menandai suatu awal pemerintahan sehingga dapat dijadikan ukuran waktu bagi daerah kapan mulai berpemerintahan? Sementara ini untuk menentukan hari jadi Kabupaten Jember berpedoman pada sejarah pemerintahan kolonial Belanda, yaitu berdasarkan pada Staatsblad nomor 322 tanggal 9 Agustus 1928 yang mulai berlaku tanggal 1 Januari 1929 sebagai dasar hukumnya. (Pemerintah Kabupaten Jember, 2016).

### **2.2.2 Geografis dan Tipografi Jember**

Kabupaten Jember, adalah sebuah kabupaten di Provinsi Jawa Timur, Indonesia. Ibukotanya adalah Jember. Kabupaten Bondowoso di utara, kabupaten Banyuwangi di timur, Samudera Hindia di selatan, serta Kabupaten Bondowoso di utara, Kabupaten Lumajang di barat. Kabupaten Jember terdiri atas 31 kecamatan. Kota Jember dahulu merupakan kota administrative, namun sejak tahun 2001 istilah kota administratif dihapus, sehingga Kota Administratif Jember kembali menjadi bagian dari Kabupaten. Kabupaten Jember terletak pada posisi 6°27'9'' s/d 7°14'33'' Bujur Timur dan 7°59'6'' s/d 8°33'56'' Lintang Selatan.

Jember memiliki luas 3.333,94 Ha dengan ketinggian antara 0 – 3.330 mdpl. Iklim Kabupaten Jember adalah tropis dengan kisaran suhu antara 23°C – 32°C. Bagian selatan wilayah Kabupaten Jember adalah dataran rendah dengan titik terluarnya adalah Pulau Barong. Pada kawasan ini terdapat Taman Nasional Meru Betiri yang berbatasan dengan wilayah administratif Kabupaten Banyuwangi. Bagian barat laut (berbatasan dengan Kabupaten Probolinggo adalah pegunungan, bagian dari pegunungan Iyang, dengan puncaknya Gunung Argopuro (3.088 m). Bagian timur merupakan bagian dari rangkaian Dataran Tinggi Ijen. Jember memiliki beberapa sungai antara lain Sungai Bedadung yang bersumber dari Pegunungan Iyang di bagian tengah, Sungai Mayang yang bersumber dari

Pegunungan Raung di bagian timur, dan Sungani Bondouodo yang bersumber dari Pegunungan Semeru di bagian barat. (Arvita Agus Kurnisari, 2014).

## **2.3 Rumah Sakit**

### **2.3.1 Pengertian Rumah Sakit**

Menurut WHO (*World Health Organization*), rumah sakit adalah bagian integral dari suatu organisasi sosial dan kesehatan dengan fungsi menyediakan pelayanan paripurna (komprehensif), penyembuhan penyakit (kuratif) dan pencegahan penyakit (preventif) kepada masyarakat. Rumah sakit Juga merupakan pusat pelatihan bagi tenaga kesehatan dan pusat penelitian medik. (Samuel Halomoan, 2012).

Berdasarkan undang – undang No. 44 Tahun 2009 tentang rumah sakit, yang dimaksudkan dengan rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan dan gawat darurat. (Samuel Halomoan, 2012).

### **2.3.2 Rumah Sakit di Wilayah Kabupaten Jember**

Daftar rumah sakit yang ada di Kabupaten Jember menurut Mochammad Adibatuszackya (2016), yaitu :

- a. Rumah Sakit Umum Daerah Dr. Soebandi
- b. Rumah Sakit Jember Klinik
- c. Rumah Sakit Baladhika Husada (DKT)
- d. Rumah Sakit Paru Jember
- e. Rumah Sakit Bina Sehat
- f. Rumah Sakit Citra Husada
- g. Rumah Sakit Umum Daerah Balung
- h. Rumah Sakit Kaliwates
- i. Rumah Sakit Umum Daerah Kalisat

## 2.4 Haversine Formula

*Haversine* adalah persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. (Gus Pascima, 2015).

Berikut ini merupakan rumus dari metode *Haversine formula* menurut M. Rofiq dan Riza Fathul Uzzy (2014) dijelaskan pada Tabel 2.1 Rumus *Haversine Formula*, yaitu :

Tabel 2.1 Rumus *Haversine Formula*

Rumus Haversine Formula	
$r(\text{bumi}) = 6371 \text{ km}$	
$d = 1^\circ = 0.01745329251994 \text{ rad}$	
$\Delta\text{Lat} = (\text{Lat}2 - \text{Lat}1) \times d$	
$\Delta\text{Long} = (\text{Long}2 - \text{Long}1) \times d$	
$\sin(\Delta\text{Lat}/2)^2$	
$\sin(\Delta\text{Long}/2)^2$	
$\cos(\text{Lat}2 \times d)$	
$\cos(\text{Lat}1 \times d)$	
$\text{Jarak} = 2r \times \arcsin(\sqrt{\sin(\Delta\text{Lat}/2)^2 + \cos(\text{Lat}2 \times d) \times \cos(\text{Lat}1 \times d) \times \sin(\Delta\text{Long}/2)^2})$	
Keterangan :	
$r$	= Radius bumi (rata – rata radius = 6371 km)
$d$	= Radians
$\text{lat}$	= <i>Latitude</i> (lintang)
$\text{long}$	= <i>Longitude</i> (bujur)

## 2.5 Framework

### 2.5.1 Definisi Framework

*Framework* jika diterjemahkan secara harfiah berarti “kerangka kerja”. *Framework* juga dapat diartikan sebagai kumpulan *script* (terutama *class* dan fungsi) yang dapat membantu *developer/programmer* dalam menangani berbagai masalah



dalam pemrograman seperti koneksi ke *database*, pemanggilan variabel, berkas, dan lain – lain sehingga *developer* lebih fokus dan lebih cepat membangun aplikasi.

Secara sederhana bisa dijelaskan bahwa *framework* adalah kumpulan fungsi dari kumpulan *library*. Sehingga seorang *programmer* tidak perlu lagi membuat fungsi – fungsi dari awal. *Programmer* tinggal memanggil fungsi – fungsi yang sudah ada di dalam *framework*. Tentunya cara menggunakan fungsi – fungsi tersebut sudah ditentukan oleh *framework*. Beberapa contoh fungsi – fungsi standar yang telah tersedia dalam suatu *framework* adalah fungsi *paging*, enkripsi, email, SEO, *session*, *security*, kalender, bahasa, manipulasi, gambar, grafik, tabel bergaya zebra, validasi, *upload*, *captcha*, proteksi terhadap XSS (XSS filtering), *template*, kompresi, XML, dan lain – lain. (Bernadus Herdi Sirenden, dkk., 2012)

### 2.5.2 CodeIgniter

*CodeIgniter* ditulis oleh Rick Ellis, seorang musisi rock yang beralih profesi menjadi pemogram. Penulisan ini dilakukan Rick setelah melakukan suatu riset kecil – keciilan. Dalam riset tersebut dia menilai bahwa banyak framework PHP yang :

- a. Menggunakan banyak asumsi bahwa pemogram memiliki keterampilan tinggi dan pengetahuan luas;
- b. Mempersyaratkan ketergantunagn pada PEAR (*PHP Extension and Repository*) dan banyak aplikasi *Open Source* lain;
- c. Hanya kompatibel pada PHP 5;
- d. Berukuran terlalu besar atau terlalu minimalis untuk digunakan; dan
- e. Dokumentasinya kurang baik. Itupun jika memiliki dokumentasi.

Rick berniat membuat CI hanya berukuran kecil, dapat berjalan ringan, tetapi memenuhi fitur umum aplikasi PHP. Dengan itu, patut disadari bahwa CI sendiri belum tentu dapat memenuhi semua kebutuhan Anda. Situs berskala besar dan rumit tentunya kurang cocok jika dibangun dengan CI. (Antonius Nugraha Widhi Pratama, 2010)

*CodeIgniter* (CI) adalah salah satu framework php yang tangguh dengan ukuran kecil dan cukup mudah dikuasai. Untuk menggunakan *CodeIgniter* dengan maksimal, Anda akan memerlukan :

- a. PHP 4.3.2 atau versi di atasnya
- b. Web server, direkomendasikan *Apache*
- c. MySQL, MySQLi, MS SQL, Postgre, Oracle, SQLite, atau ODBC
- d. *CodeIgniter* php framework. CI dapat diperoleh secara gratis, dengan mendownloadnya di [www.codeigniter.com](http://www.codeigniter.com). (Bernadus Herdi Sirenden, dkk., 2012)

## 2.6 MySQL

MySQL tergolong sebagai DMBS (*Database Management System*). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut.

- a. Menyimpan data ke dalam tabel,
- b. Menghapus data dalam tabel,
- c. Mengubah data dalam tabel,
- d. Mengambil data yang tersimpan dalam tabel,
- e. Memungkinkan untuk melakukan pengaturan hak akses terhadap data.

MySQL banyak dipakai untuk kepentingan penanganan *database* karena selain handal juga bersifat *Open Source*. Konsekuensi dari *Open Source*, perangkat lunak ini dapat dipakai oleh siapa saja tanpa membayar dan *source code*-nya bisa diunduh oleh siapa saja. (Abdul Kadir, 2010)

MySQL adalah suatu perangkat lunak untuk mengatur *database* yang berhubungan (*Relational Database Management System* atau RDBMS) yang sangat terkenal di dunia dan merupakan *database* SQL yang *open source*. MySQL mendukung berbagai *platform*, serta kecepatan aksesnya cukup bisa diandalkan. Selain itu, kemudahannya dalam integrasi ke berbagai aplikasi.

MySQL dikembangkan oleh sebuah perusahaan Swedia yang bernama MySQL AB, yang pada saat itu bernama T.c.X *Data-Konsult* AB, dimulai sejak tahun 1994 – 1995. Tujuan mula – mula perusahaan tersebut membuat MySQL untuk mengembangkan aplikasi web yang dimiliki kliennya. (Bernadus Herdi Sirenden, dkk., 2012)

## 2.7 Google Maps

*Google Map* merupakan salah satu layanan yang dikeluarkan oleh *Google*. *Google Maps* menyediakan layanan berbasis peta yang sangat responsif dan mudah dalam penggunaannya. Dengan menggunakan *Google Map* ini, pengguna dapat dengan mudah mencari suatu lokasi serta dapat melakukan penelusuran route menuju lokasi yang diinginkan.

Ditingkat pemrograman, *Google Map* dapat dikembangkan dengan basis data, semua data yang terkait dengan titik lokasi disimpan dalam tabel dan dapat ditampilkan sesuai keinginan pengguna. Isi tabel yang berisi data posisi peta dapat ditampilkan dengan menyajikan informasi lokasi yang menggunakan *Google Map*, pengunjung *web* tentunya akan mendapatkan informasi yang lebih detail terutama informasi lokasi perusahaan atau instansi.

Melalui fitur *Google Maps*, pengguna internet dapat *browsing* informasi grafis berikut :

- a. *Satellite Map*
- b. Pengguna dapat menikmati gambar satelit planet bumi. Pengguna juga dapat menikmati foto satelit lebih detail lengkap dengan cara *zooming* pada bagian peta yang diinginkan.
- c. Hasil pencarian integrasi
- d. Mencari lokasi, bisnis, peta buatan pengguna dan *real estate*.
- e. *Draggable Maps*
- f. Peta digital mapping yg *draggable* (bisa digeser) dengan bantuan mouse.
- g. *Terrain Maps* (Peta Topografi)
- h. *Terrain Maps* menyediakan informasi fitur peta fisik atau peta topografi yang biasa disediakan buku peta Atlas. (Rani Ekasari, 2016)

## 2.8 Google Maps API

Menurut Rani Ekasari (2016), *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu *web* atau *blog* sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi internet yang sangat stabil. Dengan menggunakan *Google Maps 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.

Dalam pembuatan program *Google Map API* menggunakan urutan sebagai berikut:

- a. Memasukkan *Maps API JavaScript* ke dalam HTML.
- b. Membuat element *div* dengan nama *map\_canvas* untuk menampilkan peta.
- c. Membuat beberapa obyek literal untuk menyimpan properti-properti pada peta.
- d. Menuliskan fungsi *JavaScript* untuk membuat obyek peta.
- e. Meng-inisiasi peta dalam *tag body HTML* dengan *event onload*.

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, contoh nya akan menunjukkan gunung dan sungai.
- d. *HYBRID*, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota).

Fitur dalam *Google Maps API* sebagai berikut :

- a. Kerangka kerja aplikasi (application framework)  
Digunakan untuk menulis aplikasi di Android sehingga memungkinkan penggunaan kembali dan penggantian komponen.

Kerangka kerja ini didukung oleh berbagai open source libraries seperti openssl, sqlite, dan libc serta didukung oleh libraries utama Android. Kerangka kerja sistem operasi Android didasarkan pada UNIX file system permission yang menjamin bahwa aplikasi-aplikasi tersebut hanya memiliki kemampuan yang diberikan oleh pemilik ponsel pada waktu penginstalan.

b. Dalvik Virtual Machine (DVM)

*Dalvik Virtual Machine* (DVM) adalah sebuah mesin virtual yang menggunakan memori yang sangat rendah dan secara khusus dirancang untuk Android untuk dijalankan pada *embedded system*. DVM bekerja dengan baik pada situasi dengan tenaga yang rendah dan mengoptimalkan perangkat *mobile*. DVM juga mengatur atribut dari *Central Processing Unit* (CPU) serta membuat sebuah format *file* yang spesial (.DEX) yang dibuat selama *build time post processing*. DVM mengambil *file* yang dihasilkan oleh *class* Java dan menggabungkannya ke dalam satu atau lebih *Dalvik Executable* (.dex). DVM dapat menggunakan kembali salinan informasi dari beberapa *class file* dan secara efektif mengurangi kebutuhan penyimpanan oleh setengah dari *Java Archive* (.jar) *file* tradisional. Konversi antara kelas Java dan format (.dex) dilakukan dengan memasukkan “dx tool”. DVM menggunakan *assembly-code* yang berbeda dimana DVM menggunakan *register* sebagai unit utama dari penyimpanan data daripada menggunakan *stack*. Hasil akhir dari *executable-code* pada Android, merupakan hasil dari DVM yang didasarkan bukan pada Java *byte-code* melainkan pada *file* (.dex). Hal ini berarti bahwa Java *byte-code* tidak dieksekusi secara langsung melainkan dimulai dari Java *classfile* terlebih dahulu dan kemudian mengkonversikannya ke dalam *file* (.dex) yang berhubungan.

## 2.9 Unified Modeling Language (UML)

*Unified Modelling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa A,S dan M.Shalahuddin, 2013).

### a. *Use Case Diagram*

*Use case diagram* atau diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa saja yang disebut aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Rosa A,S dan M.Shalahuddin, 2013) .

### b. *Activity Diagram*

*Activity diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal seperti rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan, urutan atau pengelompokan tampilan dari sistem atau user interface dimana setiap aktivitas

dianggap memiliki sebuah rancangan antarmuka tampilan, dan rancangan menu yang ditampilkan pada perangkat lunak (Rosa A,S dan M.Shalahuddin, 2013).

c. *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek. Membuat diagram sekuen juga membutuhkan untuk melihat skenario yang ada pada use case. Banyaknya diagram sekuen yang harus digambarkan adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa A,S dan M.Shalahuddin, 2013).

d. *Class Diagram*

*Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas yang memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak sesuai dengan perancangan diagram kelas (Rosa A,S dan M.Shalahuddin, 2013).

## **2.10 Android**

*Android* merupakan sistem operasi berbasis *Linux* yang digunakan untuk telepon seluler (*mobile*), seperti telepon pintar (*Smartphone*) dan komputer tablet (PDA). Berikut beberapa pengertian lain dari *Android*, yaitu (Supardi, 2015):

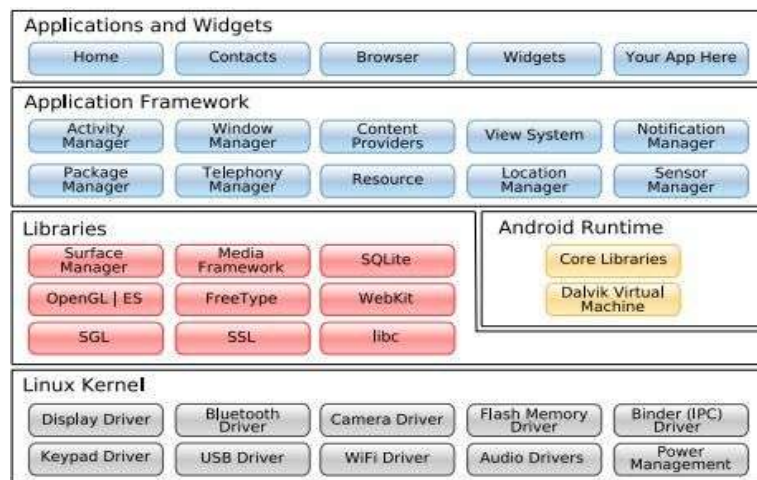
- a. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.

- b. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*
- c. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time environment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat atau *device* dengan sistem memori yang kecil.

Terdapat dua jenis distributor sistem operasi *Android*. Pertama yang mendapatkan dukungan penuh dari *Google* atau *GMS (Google Mail Service)*, dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari *Google*, atau dikenal sebagai *OHD ( Open Handset Distribution)*. Pada tahun 2012, *android* telah digunakan pada peranti-peranti : *smartphone, tablet*, peranti pembaca buku *elektronik, netbook, MP4 player*, dan *TV internet*.

### 2.11.1 Arsitektur *Android*

Secara garis besar, arsitektur android terdiri atas *Applications Widgets*, *Applications Framework*, *Libraries*, *Android Run Time*, dan *Linux Kernel*.



Gambar 2.1 Arsitektur Android (Supardi, 2015)

- a. *Appllications* dan *Widgets* merupakan layer (lapis), dimana kita berhubungan dengan aplikasi saja.
- b. *Applications Frameworks* merupakan *Open development Platform* yang ditawarkan android untuk dapat dikembangkan dalam membangun aplikasi.
- c. *Libraries* merupakan layer dimana fitur-fitur Android berada.



- d. *Android Run Time* merupakan layer yang membuat aplikasi Android dapat dijalankan, dimana dalam prosesnya menggunakan implementasi *Linux*.
- e. *Linux Kernel* merupakan layer inti dari sistem operasi Android berada.

### 2.10.2 Versi *Android*

Berikut uraian atau table dari versi *Android* (Supardi, 2015) :

Tabel 2.2 Versi *Android*

	Nomor Versi	Nama Versi	Tanggal rilis
1	(belum memakai)	Android Beta	5 November 2007
2	1.0	Android 1.0	23 september 2008
3	1.1	Android 1.1	9 Februari 2009
4	1.5	Cupcake	30 April 2009
5	1.6	Donut	15 September 2009
6	2.0	Éclair	26 Oktober 2009
7	2.0.1	Éclair	3 Desember 2009
8	2.1	Éclair	12 Januari 2010
9	2.2	Froyo	20 Mei 2010
10	2.2.1	Froyo	18 Januari 2011
11	2.2.2	Froyo	22 Januari 2011
12	2.2.3	Froyo	21 November 2011
13	2.3	GingerBread	6 Desember 2010
14	2.3.3	GingerBread	9 februari 2011
15	2.3.4	GingerBread	28 April 2011
16	2.3.5	GingerBread	25 Juli 2011
17	2.3.6	GingerBread	2 september 2011
18	2.3.7	GingerBread	21 September 2011
19	3.0	Honeycomb	22 Februari 2011
20	3.1	Honeycomb	10 Mei 2011
21	3.2	Honeycomb	15 Juli 2011

Lanjutan Tabel 2.2 Versi *Android*

	Nomor Versi	Nama Versi	Tanggal rilis
22	3.2.1	Honeycomb	20 September 2011
23	3.2.2	Honeycomb	30 Agustus 2011
24	3.2.4	Honeycomb	Desemebr 2011
25	3.2.6	Honeycomb	Februari 2011
26	4.0.1	ICS (Ice Cream Sandwich)	19 Oktober 2011
27	4.0.2	ICS	28 November 2011
28	4.0.3	ICS	16 Desember 2011
29	4.0.4	ICS	29 Maret 2012
30	4.1	Jelly Bean	27 Juni 2012
31	4.4	KitKat	14 Oktober 2013
32	5.0	Lollipop	25 Juni 2014
33	5.1	Lollipop	25 Juni 2014

### 2.10.3 *Android* Menurut API Level

Dalam membuat program biasanya yang banyak dipakai di dalam menentukan target versi *Android* adalah nomor API level (Supardi,2015).

Tabel 2.3 Versi *Android* berdasarkan API Level (Supardi,2015)

Versi Android Berdasarkan API Level
2.1 Android 1.0 (API Level 1)
2.2 Android 1.1 (API Level 2)
2.3 Android 1.5 Cupcake(API Level 3)
2.4 Android 1.6 Donut(API Level 4)
2.5 Android 2.0 Eclair(API Level 5)
2.6 Android 2.0.1 Eclair(API Level 6)
2.7 Android 2.1 Eclair(API Level 7)
2.8 Android 2.2-2.2.3 Froyo(API Level 8)
2.9 Android 2.3-2.3.2 GingerBread(API Level 9)

Lanjutan Tabel 2.3 Versi *Android* berdasarkan API Level (Supardi,2015)

Versi Android Berdasarkan API Level
2.10 Android 2.3.3-2.3.7 GingerBread(API Level 10)
2.11 Android 3.0 Honeycomb(API Level 11)
2.12 Android 3.1 Honeycomb(API Level 12)
2.13 Android 3.2 Honeycomb(API Level 13)
2.14 Android 4.0-4.0.2 Ice Cream Sandwich(API Level 14)
2.15 Android 4.0.3-4.0.4 Ice Cream Sandwich(API Level 15)
2.16 Android 4.1 Jelly Bean(API Level 16)
2.17 Android 4.2 Jelly Bean(API Level 17)
2.18 Android 4.3 Jelly Bean(API Level 18)
2.19 Android 4.4 KitKat(API Level 19)
2.20 Android 5.0 Lollipop(API Level 21)

## 2.12 Java

*Java* adalah bahasa yang dapat dijalankan di sembarang platform, di beragam lingkungan: internet, bahasa yang dapat dijalankan disembarang *platform*, di beragam lingkungan: *internet*, *consumer electronic products*, dan *computer applications*. Sebagian besar bahasa pemrograman modern berdiri diatas pustaka-pustaka kelas yang telah ada untuk mendukung fungsionalitas. Pada bahasa java, kelompok-kelompok kelas yang berkaitan erat dimasukkan disatu paket, bervariasi sesuai edisi *java*. Masing-masing paket untuk dimaksudkan tertentu: *applet*, aplikasi standar, skala *enterprise*, dan produk konsumen ( Yuliandari ,2015 ).

## 2.13 Java Development Kit

JDK (*Java Development Kit*) berisi sekumpulan kelas baris perintah (*sommand-line-tool*) untuk menciptakan program *java*. JDK berisi sekumpulan kakas, utilitas, dan dokumentasi serta kode applet contoh untuk pengemabngan program java. Berikut adalah daftar komponen utama JDK(Yuliandari, 2015) :

- e. Kompilator (*javac*), Kompilator menghasilkan file bytecode kelas dengan ekstensi *.class*. Kompilator menciptakan satu file *.class* untuk tiap kelas yang didefinisikan di file sumber.
- f. *Interpreter*, merupakan modul utama sistem Java yang digunakan aplikasi Java. *Interpreter* untuk menjalankan program *bytecode* Java. *Interpreter* bertindak sebagai kakas baris perintah untuk menjalankan program Java non-grafis.
- g. *Applet Viewer*, adalah kakas untuk pengujian Java *applet* secara minimal.
- h. *Java Debugger* (*jdb*) adalah utilitas baris perintah untuk melakukan *debugging* aplikasi Java.
- i. *Java Class File Disassembler* (*javap*), untuk melakukan penguraian (*disassemble*) terhadap *file class*. Keluaran *default* aktivitas *dissamble* berisi daftar dari data publik dan metode publik di kelas.
- j. *Java Header and Stub File Generator* (*javah*), untuk menghasilkan C *header* dan file kode sumber untuk implementasi metode Java dalam bahasa C.
- k. *Java Documentation Generator* (*javadoc*), adalah kakas untuk menghasilkan dokumentasi API secara langsung dari kode sumber Java, *Java documentation generator* melakukan parsing terhadap file sumber Java dan menghasilkan halaman HTML berdasarkan deklarasi dan komentar di file sumber.
- l. Demo, JDK berisi beragam contoh program Java, semuanya disertai kode sumber.
- m. Kode sumber Java API, JDK disertai kode sumber secara lengkap untuk semua kelas yang membentuk Java API. Kode sumber API secara otomatis dipasang di *hard drive* saat melakukan dekompres JDK tapi masih dalam bentuk yang file dikompresi.

## 2.14 Android SDK

*Android SDK* merupakan alat atau tool yang digunakan untuk membuat aplikasi *platform* Android menggunakan bahasa pemrograman *java*. Beberapa fitur android yang paling penting adalah (Supardi, 2015) :

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- b. Mesin *virtual* Dalvik dioptimalkan untuk perangkat mobile.
- c. *Integrated browser* berdasarkan *engine open source webkit*.
- d. Grafis yang dioptimalkan dan didukung oleh *libraries*, grafis 2D, grafis 3D berdasarkan spesifikasi *opengl* ES1,6 (opsional akselerasi hardware).
- e. *SQLite* untuk penyimpanan data.
- f. Media support yang mendukung audio, video, dan gambar (MPEG 4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM *telephony* (tergantung *hardware*)).
- g. *Bluetooth*, EDGE, 3G, dan *wifi* (tergantung *hardware*).
- h. Kamera, GPS, kompas, dan akselerometer (tergantung *hardware*).
- i. Lingkungan *development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk debugging, profil dan kinerja memori, dan *plug-in* untuk IDE *eclipse*.

## **2.15 Karya Tulis Yang Mendahului**

### **2.15.1 Sistem Informasi Geografis Daerah Rawan Banjir Kabupaten Jember (Arvita Agus Kurnisari, Polteknik Negeri Jember, 2014).**

Sistem informasi geografis daerah rawan banjir Kabupaten jember merupakan sistem yang dapat memberikan informasi mengenai daerah rawan banjir pada masing-masing kecamatan yang mana daerah rawan tersebut didapatkan berdasarkan perhitungna *fuzzy tsukamoto* sesuai dengan ketentuan yang telah ditentukan, selain dapat memberikan informasi daerah rawan juga dapat memberikan informasi mengenai sarana kesehatan dan kejadian banjir yang mendahului. Sistem ini mengimplementasikan informasi kedalam bentuk peta digital dengan tujuan agar mempermudah penyampaian informasi. Data yang digunakan diperoleh dari BPBD (Badan Penanggulangan Bencana Daerah) dan BPEKAP (Badan Perencanaan Pembangunan Kabupaten) Jember yang kemudian diolah menjadi sebuah sistem yang sesuai dan tepat guna. Pada hasil akhir sistem ini akan menampilkan informasi daerah rawan banjir, sarana kesehatan, serta data kejadian banjir mendahului yang diimplementasikan kedalam bentuk peta digital agar lebih mudah penyampaiannya.

### **2.15.2 Aplikasi Penentu Titik Lokasi Rumah Sakit Terdekat (Studi Kasus Rumah Sakit di Jember), (Mochammad Adibatuszackya, Politeknik Negeri Jember, 2016)**

Jember adalah kota terbesar ketiga di Jawa Timur setelah Surabaya dan Malang. Di wilayah ini banyak tempat-tempat yang strategis untuk berbagai macam sentra bidang sebagai penunjang pendapatan daerah, seperti sentra ekonomi, bisnis, hiburan, pendidikan, pariwisata, dan pemerintahan. Begitu banyak tempat-tempat strategis, ditambah lagi dengan tingginya tingkat kepadatan penduduk di kota Jember, mengakibatkan kemungkinan terjadinya hambatan atau hal-hal yang tidak diinginkan, seperti tindakan kejahatan, kemacetan, banjir dan lain sebagainya, tak terkecuali kesehatan dan kecelakaan lalu lintas. Tidak sedikit masyarakat yang panik dan menjadi bingung bila sedang dalam keadaan darurat seperti halnya kecelakaan lalu lintas atau kesehatannya terganggu dan butuh pelayanan medis dengan cepat.

Untuk mengatasi permasalahan ini, penggunaan peta digital pada smartphone yang sangat mudah dioperasikan dapat dijadikan solusi untuk mengetahui titik lokasi rumah sakit terdekat. Dengan memanfaatkan GPS yang dapat memberikan posisi suatu objek dimuka bumi dengan akurat dan cepat (koordinat 3 dimensi  $x,y,z$ ) dan memberikan informasi waktu serta kecepatan bergerak secara kontinyu di seluruh dunia.

Untuk menentukan titik lokasi terdekat antar dua buah titik koordinat pada Sistem Informasi Geografis (SIG) dapat menggunakan beberapa metode diantaranya Teorema Harvesine. Dalam hal ini digunakan metode Euclidean Distance. Metode ini merupakan perhitungan jarak dari 2 buah titik dalam Euclidean space. Euclidean space diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 SM. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Phytagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

Berdasarkan Uraian di atas Maka Akan Dibangun “Aplikasi Penentu Titik Lokasi Rumah Sakit Terdekat Di Jember”. Aplikasi Ini Nantinya Memberikan

Informasi titik lokasi terdekat dan peta letak rumah sakit yang ada di Kota Jember beserta informasi nomor telepon darurat dari rumah sakit yang ada di Kota Jember. Dalam pengembangan aplikasi ini akan dibuat menggunakan bahasa pemrograman java mobile (android) dan dengan memanfaatkan peta milik google, yakni Google MAPS API, sehingga dalam menggunakan sistem ini pengguna diharuskan terkoneksi dengan internet karena peta Google yang bersifat online. Dengan demikian maka diharapkan pengguna dapat mengetahui titik lokasi rumah sakit terdekat dari lokasi pengguna berada.

## 2.16 State Of The Art

Tabel 2.4 Perbedaan dengan Penelitian Terdahulu

No	Penulis	Arvita Kurniasari	Agus Mochammad Adibatuszackya	Almira Octaviani	Fitri
1.	Judul	Sistem Informasi Geografis Daerah Rawan Banjir Kabupaten Jember	Aplikasi Titik Rumah Terdekat Kasus Rumah Sakit di Jember)	Penentu Lokasi Rumah Sakit (Studi Pelayanan Kesehatan dengan Memanfaatkan <i>Google Maps API</i> (Studi Kasus Rumah Sakit di Jember)	Titik
2.	Topik	SIG	SIG	SIG	
3.	Objek	Daerah Rawan Banjir Kabupaten Jember	Titik Lokasi Rumah Sakit di Kabupaten Jember	Informasi Layanan Kesehatan dan Titik Lokasi Rumah Sakit di Kabupaten Jember	
4.	Metode	<i>Fuzzy Tsukamoto</i>	<i>Euclidean Space</i>	<i>Haversine Formula</i>	
5.	Platform	Web	Andrioid	Web	

Lanjutan Tabel 2.4 Perbedaan dengan Penelitian Terdahulu

6.	Tahun	2014	2016	2017
7.	Manfaat		Mempermudah pengguna untuk menemukan titik lokasi rumah sakit terdekat di Kabupaten Jember.	Mempermudah pengguna menemukan titik lokasi rumah sakit yang pelayanan kesehatannya sesuai dengan kebutuhan.

Berdasarkan isi dari tabel diatas, dapat disimpulkan bahwa ketiga karya tulis tersebut memiliki persamaan yaitu mengambil topik Sistem Informasi Geografis (SIG). Sedangkan yang membedakan adalah sebagai berikut :

- a. Objek yang digunakan dalam karya tulis Sistem Informasi Geografis Daerah Rawan Banjir Kabupaten Jember adalah Daerah Rawan Banjir Kabupaten Jember, sedangkan objek yang digunakan dalam karya tulis Aplikasi Penentu Titik Lokasi Rumah Sakit Terdekat (Studi Kasus Rumah Sakit di Jember) adalah Titik Lokasi Rumah Sakit di Kabupaten Jember, dan objek yang digunakan dalam karya tulis Penentuan Titik Lokasi Rumah Sakit Berdasarkan Pelayanan Kesehatan dengan Memanfaatkan *Google Maps API* (Studi Kasus Rumah Sakit di Jember) adalah Informasi Layanan Kesehatan dan Titik Lokasi Rumah Sakit di Kabupaten Jember.
- b. Metode yang digunakan dalam karya tulis Sistem Informasi Geografis Daerah Rawan Banjir Kabupaten Jember adalah *Fuzzy Tsukamoto*, sedangkan metode yang digunakan dalam karya tulis Aplikasi Penentu Titik Lokasi Rumah Sakit Terdekat (Studi Kasus Rumah Sakit di Jember) adalah *Euclidean Space*, dan objek yang digunakan dalam karya tulis Penentuan Titik Lokasi Rumah Sakit Berdasarkan Pelayanan Kesehatan dengan Memanfaatkan *Google Maps API* (Studi Kasus Rumah Sakit di Jember) adalah *Haversine Formula*.



- c. *Platform* yang digunakan dalam karya tulis Sistem Informasi Geografis Daerah Rawan Banjir Kabupaten Jember dan objek yang digunakan dalam karya tulis Penentuan Titik Lokasi Rumah Sakit Berdasarkan Pelayanan Kesehatan dengan Memanfaatkan *Google Maps API* (Studi Kasus Rumah Sakit di Jember) adalah web, sedangkan *platform* yang digunakan dalam karya tulis Aplikasi Penentu Titik Lokasi Rumah Sakit Terdekat (Studi Kasus Rumah Sakit di Jember) adalah Android.

## **BAB 3. METODE KEGIATAN**

### **3.1 Tempat dan Waktu Pelaksanaan**

#### **3.1.1 Tempat Pelaksanaan**

Tempat pelaksanaan tugas akhir ini yaitu sebagai berikut :

- a. Politeknik Negeri Jember
- b. *WifiId Corner* Jember dan Yogyakarta
- c. Rumah kerabat di Jalan Pajajaran VI Blok G-17 Bukit Permai, Jember
- d. Rumah orang tua di Kadipaten Wetan KP I/189 Yogyakarta

#### **3.1.2 Waktu Pelaksanaan**

Waktu pelaksanaan tugas akhir ini mulai dari bulan September 2016 sampai dengan bulan Februari 2017.

### **3.2 Obyek Kegiatan**

Dalam pengerjaan tugas akhir ini, yang menjadi obyek penelitian adalah rumah sakit – rumah sakit yang terdaftar di dalam situs resmi Dinas Kesehatan Kabupaten Jember yaitu :

- a. Rumah Sakit Umum Daerah Dr. Soebandi, dengan pelayanan kesehatan sebagai berikut :
  - 1) Poli umum
  - 2) Poli kemoterapi
  - 3) Poli eksekutif
  - 4) Poli gigi dan mulut
  - 5) Poli jiwa
  - 6) Poli kulit dan kelamin
  - 7) Poli mata
  - 8) Poli anak
  - 9) Poli THT (Telinga Hidung Tenggorokan)

- 10) Poli bedah umum
  - 11) Poli bedah syaraf
  - 12) Poli urologi
  - 13) Poli orthopedi dan traumatologi
  - 14) Poli gizi
  - 15) Poli kandungan
  - 16) Poli syaraf
  - 17) Poli jantung
  - 18) Poli interna (dalam)
  - 19) Poli paru
- b. Rumah Sakit Paru Jember, dengan pelayanan kesehatan sebagai berikut :
- 1) Poli umum
  - 2) Poli interna
  - 3) Poli Spesialis anak 2 jantung
  - 4) Poli TB
  - 5) Poli spesialis paru A
  - 6) Poli spesialis paru B
  - 7) Poli rehab medik
- c. Rumah Sakit Baladhika Husada (DKT), dengan pelayanan kesehatan sebagai berikut :
- 1) Poli bedah
  - 2) Poli bedah onkologi
  - 3) Poli kandungan dan kebidanan
  - 4) Poli mata
  - 5) Poli anak
  - 6) Poli penyakit dalam
  - 7) Poli jantung dan pembuluh darah
  - 8) Poli anaestesi dan reanimasi
  - 9) Poli radiologi
  - 10) Poli patologi klinik
  - 11) Poli umum

- d. Rumah Sakit Bina Sehat, dengan pelayanan kesehatan sebagai berikut :
- 1) Poli umum
  - 2) Poli gigi dan mulut
  - 3) Poli penyakit dalam
  - 4) Poli kesehatan anak
  - 5) Poli paru
  - 6) Poli jantung
  - 7) Poli kulit dan kelamin
  - 8) Poli syaraf
  - 9) Poli THT
  - 10) Poli konsultasi gizi
  - 11) Poli bedah umum
  - 12) Poli bedah urologi
  - 13) Poli bedah orthopedi
  - 14) Poli bedah syaraf
  - 15) Poli fisioterapi
  - 16) Poli psikologi
  - 17) Poli akupunktur
  - 18) Poli KB dan KIA
  - 19) Poli kesehatan jiwa
  - 20) Poli mata
  - 21) Poli kebidanan dan kandungan
- e. Rumah Sakit Jember Klinik, dengan pelayanan kesehatan sebagai berikut :
- 1) Poli jantung
  - 2) Poli bedah orthopedi
  - 3) Poli syaraf
  - 4) Poli kandungan
  - 5) Poli mata
  - 6) Poli anak
  - 7) Poli bedah syaraf
  - 8) Poli kulit dan kelamin

- 9) Poli bedah mulut
  - 10) Poli gigi anak
  - 11) Poli penyakit dalam
  - 12) Poli urologi
  - 13) Poli bedah
  - 14) Poli THT
  - 15) Poli rehab medik
  - 16) Poli kesehatan jiwa
- f. Rumah Sakit Citra Husada, dengan pelayanan kesehatan sebagai berikut :
- 1) Poli mata
  - 2) Poli gigi
  - 3) Poli syaraf
  - 4) Poli penyakit dalam
  - 5) Poli anak
  - 6) Poli bedah umum
  - 7) Poli bedah orthopedi
  - 8) Poli kandungan
  - 9) Poli KIA
- g. Rumah Sakit Umum Daerah Balung, dengan pelayanan kesehatan sebagai berikut :
- 1) Poli anak
  - 2) Poli mata
  - 3) Poli syaraf
  - 4) Poli gigi
  - 5) Poli penyakit dalam
  - 6) Poli THT
  - 7) Poli bedah
  - 8) Poli umum
  - 9) Poli anaestesi dan khusus nyeri
- h. Rumah Sakit Kaliwates, dengan pelayanan kesehatan sebagai berikut :
- 1) Poli penyakit dalam

- 2) Poli jantung
- 3) Poli paru
- 4) Poli syaraf
- 5) Poli kesehatan jiwa
- 6) Poli patologi klinik
- 7) Poli anak
- 8) Poli THT
- 9) Poli mata
- 10) Poli kulit dan kelamin
- 11) Poli rehabilitasi medik
- 12) Poli radiologi
- 13) Poli bedah
- 14) Poli bedah syaraf
- 15) Poli bedah orthopedi
- 16) Poli urologi
- 17) Poli bedah mulut
- 18) Poli kandungan
- 19) Poli anaestesi
- 20) Poli umum
- 21) Poli gigi

i. Rumah Sakit Kalisat, dengan pelayanan kesehatan sebagai berikut :

- 1) Poli kandungan
- 2) Poli interna
- 3) Poli bedah
- 4) Poli anak
- 5) Poli syaraf
- 6) Poli jantung
- 7) Poli gigi

### **3.3 Alat dan Bahan**

#### **3.3.1 Alat**

Terdapat dua jenis alat yang digunakan untuk mendukung pembuatan program tugas akhir ini yaitu :

- a. Perangkat keras (*Hardware*), terdiri dari :
  - 1) Satu unit laptop *merk* Compaq seri 420
  - 2) Satu unit laptop *merk* Asus
  - 3) Satu buah mouse beserta alasnya
  - 4) Satu buah buku kosong
  - 5) Satu buah bolpoint
  - 6) Satu buah buku Pedoman Penulisan Karya Ilmiah Lingkup Internal Politeknik Negeri Jember
- b. Perangkat lunak (*Software*), terdiri dari :
  - 1) *Microsoft Word 2007*
  - 2) *Adobe Reader*
  - 3) *Mozilla Firefox*
  - 4) *Operating System Windows 7*
  - 5) *MySQL*
  - 6) *Google Maps API*
  - 7) *Power Designer*
  - 8) *Notepad*
  - 9) *Internet Download Manager*
  - 10) *XAMPP*
  - 11) *Eclipse Moon*
  - 12) *Nox Emulator*
  - 13) *WinSCP-5.9.3*

#### **3.3.2 Bahan**

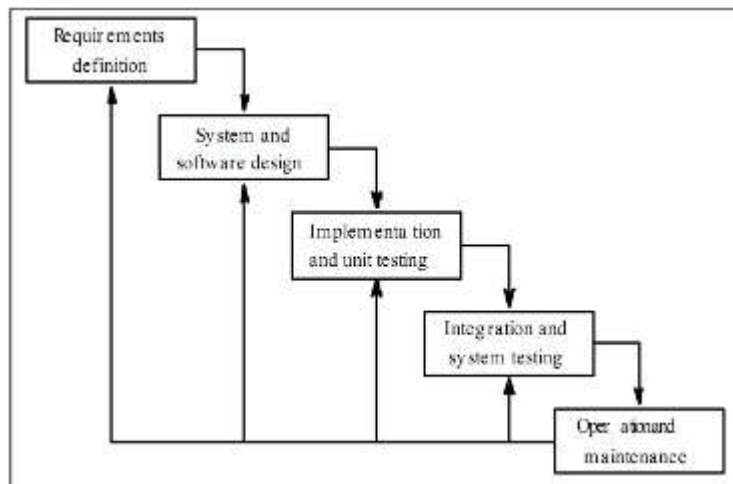
Bahan yang digunakan dalam pengerjaan tugas akhir ini yaitu data mengenai pelayanan kesehatan yang ada pada setiap rumah sakit di Jember yang dijadikan sebagai bahan untuk proses pemilihan rumah sakit yang tepat. Lalu data

lokasi pengguna berada dan data lokasi rumah sakit yang dituju setelah proses pemilihan selesai supaya dapat dihitung jaraknya menggunakan metode *Haversine Formula* dan diolah di *Google Maps API* untuk menentukan rute perjalanannya.

### 3.4 Metode Kegiatan

Metode kegiatan yang digunakan dalam Tugas akhir ini adalah metode *waterfall* yang tahapannya dimulai dari tahapan analisis, tahapan desain, tahapan pengkodean, tahapan pengujian dan tahapan supporting.

Urutan – urutan metode tersebut dapat digambarkan pada Gambar 3.1 Metode *Waterfall* sebagai berikut.



Gambar 3.1 Metode *Waterfall*

Rincian kegiatan yang dilakukan berdasarkan gambar 3.1 Metode *Waterfall* adalah sebagai berikut:

#### a. *Requirements Definition*

Pada tahap ini, peneliti menganalisis kebutuhan dengan melakukan pengumpulan data dan informasi mengenai tugas akhir ini melalui observasi, buku, jurnal, artikel maupun internet. Survey juga dilakukan ke rumah sakit – rumah sakit yang menjadi objek penelitian untuk mendapatkan data – data yang dibutuhkan.



b. *System and Software Design*

Tahapan berikutnya merupakan pembuatan desain sistem berdasarkan analisis yang telah di susun. Desain sistem yang dibuat tidak hanya desain *interface* saja namun juga desain alur dari sistem.

c. *Implementation and Unit Testing*

Dari desain sistem yang telah dibuat pada tahap sebelumnya kemudian diimplementasikan kedalam bahasa pemograman hingga menjadi sebuah produk yang siap digunakan.

d. *Integration and System Testing*

Setelah program yang dibuat selesai, maka dilakukan pengujian untuk melihat apakah program yang dibuat sesuai dengan rancangan pada tahap sebelumnya dan dapat berjalan dengan seharusnya.

e. *Operation and Maintenance*

Dalam tahap ini, setelah produk digunakan maka akan dilakukan pengembangan terhadap sistem yang tergantung dari kebutuhan – kebutuhan user. Selain itu juga dilakukan perbaikan sistem yang mungkin mengalami kerusakan.