

**TUGAS AKHIR – KI141502**

**Implementasi Metode *K-Nearest Neighbor* Untuk Penentuan Lokasi Pos Hujan Terdekat dengan Titik Rute Perjalanan Pada Aplikasi Clearroute.**

**Anwar Rosyidi**

**NRP 5113100180**

**Dosen Pembimbing I**

**Dr.tech. Ir. R.V.Hari Ginardi., M.Sc.**

**Dosen Pembimbing II**

**Abdul Munif, S.Kom., M.Sc.**

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2017**



***(Halaman ini sengaja dikosongkan)***

**TUGAS AKHIR – KI141502**

**Implementasi Metode *K-Nearest Neighbor* Untuk Penentuan Lokasi Pos Hujan Terdekat dengan Titik Rute Perjalanan Pada Aplikasi Clearroute.**

**Anwar Rosyidi**

**NRP 5113100180**

**Dosen Pembimbing I**

**Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.**

**Dosen Pembimbing II**

**Abdul Munif, S.Kom., M.Sc.**

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2017**

****

**UNDERGRADUATE THESES – KI141502**

**Implementation of K-Nearest Neighbor Method For Determining Nearby Weather Station With Travel Route Point on Clearroute Application**

**Anwar Rosyidi**

**NRP 5113100180**

**First Advisor**

**Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.**

**Second Advisor**

**Abdul Munif, S.Kom., M.Sc.**

**Department of Informatics**

**Faculty of Information Technology**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2017**

***(Halaman ini sengaja dikosongkan)***

# LEMBAR PENGESAHAN

**IMPLEMENTASI METODE *K-NEAREST NEIGHBOR* UNTUK PENENTUAN LOKASI POS HUJAN TERDEKAT DENGAN TITIK RUTE PERJALAN PADA APLIKASI CLEARROUTE**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

pada

Bidang Studi Manajemen Informasi

Program Studi S-1 Departemen Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Oleh:

**Anwar Rosyidi**

**NRP: 5113100180**

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr.tech. Ir. R.V.Hari Ginardi, M.Sc. .....................

NIP. 196505181992031003 (Pembimbing 1)

1. Abdul Munif, S.Kom., M.Sc. .....................

NIP. 198608232015041004 (Pembimbing 2)

**SURABAYA**

**Mei, 2017*(Halaman ini sengaja dikosongkan)***

**IMPLEMENTASI METODE *K-NEAREST NEIGHBOR* UNTUK PENENTUAN LOKASI POS HUJAN TERDEKAT DENGAN TITIK RUTE PERJALAN PADA APLIKASI CLEARROUTE**

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | **:** | **Anwar Rosyidi** |
| **NRP** | **:** | **5113100180** |
| **Jurusan** | **:** | **Teknik Informatika FTIF-ITS** |
| **Dosen Pembimbing 1** | **:** | **Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.** |
| **Dosen Pembimbing 2** | **:** | **Abdul Munif, S.Kom., M.Sc.** |

# Abstrak

Clearroute adalah aplikasi yang dibangun untuk memudahkan seseorang mengetahui informasi cuaca pada rute perjalanan yang akan dilaluinya. Aplikasi ini dibangun untuk platform perangkat bergerak yang memiliki fungsi-fungsi untuk memudahkan seseorang mencari informasi cuaca untuk perjalanannya.

Pada aplikasi perangkat bergerak tersebut, diperlukan sebuah sistem web service yang dapat melakukan pengolahan data. Seperti pengolahan data untuk menentukan lokasi pos hujan terdekat dengan rute perjalanan pengguna. Pengolahan data untuk mendapatkan kondisi cuaca dan kondisi ramalan cuaca yang akan datang. Oleh karena itu dibutuhkanlah suatu sistem yang dapat memenuhi fungsi tersebut agar aplikasi yang dibangun dapat bekerja dengan baik,

Pada sistem yang dibangun ini digunakan algoritma K-Nearest Neighbor untuk menentukan klasifikasi cuaca yang dimiliki oleh rute yang akan dilalui oleh pengguna. Kemudian sistem ini memanfaatkan Laravel 5.4 sebagai kerangka kerja pemrograman.

Pengujian pada sistem ini dilakukan dengan cara melakukan permintaan informasi cuaca kepada sistem, mencoba melakukan ekstraksi data cuaca dari BMKG.Pengujian tersebut dilakukan untuk mengetahui keberhasilan sistem dalam menangani permintaan dan pengolahan data yang diminta oleh aplikasi perangkat bergerak Clearroute. Dari hasi pengujian, sistem yang telah dirancang dan diimplementasikan telah memenuhi segala kebutuhan pengolahan data pada aplikasi Clearroute

**Kata kunci:** **Clearroute, BMKG, K-Nearest Neighbor, Web Service, Pos Hujan, Cuaca.**

**IMPLEMENTATION OF *K-NEAREST NEIGHBOR* METHOD FOR DETERMINING NEARBY WEATHER STATION WITH TRAVEL ROUTE POINT ON CLEARROUTE APPLICATION**

|  |  |  |
| --- | --- | --- |
| **Student’s Name** | **:** | **Anwar Rosyidi** |
| **Student’s ID** | **:** | **5113100180** |
| **Department** | **:** | **Department of Informatics FTIF-ITS** |
| **First Advisor** | **:** | **Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.** |
| **Second Advisor** | **:** | **Abdul Munif, S.Kom., M.Sc.** |

# Abstract

*Clearroute is an application that is built in order to facilitate people to get the weather information for a spesific route that is going to be taken. This application is built for mobile device platforms that has the functions of making it easier for people to search about weather information for their trip.*

*In this mobile device application, a web service system that is able to perform data processing is required, such as data processing to determine the location of the nearest rain post to the user’s route. Data processing to get weather condition and its forecast condition. Therefore, a system that is able to fulfill the function is required in order to have the application built properly.*

*This system uses K-Nearest Neighbor algorithm to determine the weather classification of the route that is going to be passed by the user. Also, this system utilizes Laravel 5.4 as its programming framework. The testing of this system is done by requesting weather information to the system, trying to extract the weather data from BMKG. The testing is done to determine the success of the system in handling request and data processing requested by Clearroute mobile device platform. From the test result, the system that has been designed and implemented has met all the needs of data processing for Clearroute application.*

**Keyword: Clearroute, BMKG, K-Nearest Neighbor, Web Service, Rain Post, Weather.**

**(*Halaman ini sengaja dikosongkan)***

# KATA PENGANTAR



Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

***“*Implementasi Metode *K-Nearest Neighbor* Untuk Penentuan Lokasi Pos Hujan Terdekat Dengan Titik Rute Perjalan Pada Aplikasi Clearroute”**

Harapan dari penulis semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Suharsono, Ibu Pujiati Nur Farida, Misbahul Munir, Habib Burrahman, dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
2. Bapak Hari Ginardi dan Bapak Abdul Munif selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Andika Putra Wardana, Ariosa Fakhri, Varian Aditya yang selama ini memberikan semangat dan ilmu yang bermanfaat selama penulis menyelesaikan masa studi.
6. Teman-teman Kabinet BEM FTIF ledakan inovasi 2015/2016, HMTC Berkarya 2014/2015, HMTC Optimasi 2015/2016,BPH SCHEMATICS HMTC 2015 yang telah memberikan pengalaman berharga kepada penulis selama masa studi penulis.
7. Teman-teman penghuni Lab Algoritma dan Pemrograman (Alpro),dan teman-teman lain yang telah memberikan banyak dukungan dan semangat kepada penulis.
8. Teman-teman pemandu FTIF angkatan 2012, 2013 dan 2014 yang telah berbagi kisah seru dalam memberikan pelatihan LKMM selama 2 tahun kepengurusan.
9. Teman-teman Bakor Pemandu FTIF 2015/2016 Robithah, Nafiar R, Ghaly A, Risha Z Aditya, Rika Nurlaili yang telah memberikan warna dan kontribusi yang baik ketika bekerja sama di BEM FTIF
10. Teman-teman angkatan 2013 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.
11. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2017

Anwar Rosyidi

# DAFTAR ISI

LEMBAR PENGESAHAN v

Abstrak vii

Abstract ix

KATA PENGANTAR xi

DAFTAR ISI xiii

DAFTAR GAMBAR xvii

DAFTAR TABEL xix

DAFTAR KODE SUMBER xxi

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Permasalahan 2

1.4 Tujuan 2

1.5 Manfaat Tugas Akhir 3

1.6 Metodologi 3

1.6.1 Penyusunan Proposal Tugas Akhir 3

1.6.2 Studi Literatur 3

1.6.3 Implementasi Perangkat Lunak 4

1.6.4 Pengujian dan Evaluasi 4

1.6.5 Penyusunan Buku 4

1.7 Sistematika Penulisan Laporan 4

BAB II TINJAUAN PUSTAKA 7

2.1 Sistem Peramalan Cuaca 7

2.2 Badan Meteorologi Klimatologi dan Geofisika (BMKG) 8

2.3 Pos Hujan BMKG 11

2.4 *Application Program Interface* (API) 12

2.5 *Hypertext Preprocessor* (PHP) 12

2.6 *JavaScript Object Notation* (JSON) 13

2.7 *Extensible Markup Language* (XML) 14

2.8 PostgreSQL 15

2.9 *K-Nearest Neighbor* (KNN) 16

2.10 *Geography JSON (GeoJSON)* 18

2.11 RGB 19

BAB III PERANCANGAN MODUL 21

3.1 Deskripsi Umum 21

3.1.1 Arsitektur Sistem 21

3.2 Desain Umum Sistem 27

3.2.1 Mendapatkan Data Ramalan Cuaca 28

3.2.2 Mendapatkan Data Kondisi Cuaca Terkini 30

3.2.3 Perhitungan *K-Nearest Neighbor* 31

3.2.4 Menentukan Lokasi Pos Hujan Terdekat dan Kondisi Cuaca Pada Rute Perjalanan 34

BAB IV IMPLEMENTASI 37

4.1 Lingkungan Implementasi 37

4.1.1 Lingkungan Implementasi Perangkat Keras 37

4.1.2 Lingkungan Implementasi Perangkat Lunak 37

4.2 Implementasi Perangkat Lunak 38

4.2.1 Implementasi Proses Mendapatkan Data Ramalan 38

4.2.2 Implementasi Proses Mendapatkan Data Kondisi Cuaca Terkini 42

4.2.3 Implementasi Proses Menentukan Lokasi Pos Hujan Terdekat dan Kondisi Cuaca Pada Rute Perjalanan 43

4.2.4 Implementasi Proses Perhitungan *K-Nearest Neighbor* 45

BAB V UJI COBA DAN EVALUASI 47

5.1 Lingkungan Pengujian 47

5.2 Skenario Uji coba 48

5.2.1 Skenario Uji Coba 1 48

5.2.2 Skenario Uji Coba 2 56

5.2.3 Skenario Uji Coba 3 58

5.2.4 Skenario Uji Coba 4 59

5.2.5 Skenario Uji Coba 5 62

5.3 Evaluasi 64

5.3.1 Evaluasi Skenario Uji Coba 1 64

5.3.2 Evaluasi Skenario Uji Coba 2 64

5.3.3 Evaluasi Skenario Uji Coba 3 65

5.3.4 Evaluasi Skenario Uji Coba 4 65

5.3.5 Evaluasi Skenario Uji Coba 5 65

BAB VI KESIMPULAN DAN SARAN 67

6.1 Kesimpulan 67

6.2 Saran 68

LAMPIRAN A. LOKASI POS HUJAN 69

LAMPIRAN B. Basis Data Clearroute 81

LAMPIRAN C. Hasil Uji Coba Skenario 5 83

DAFTAR PUSTAKA 91

BIODATA PENULIS 93

***(Halaman ini sengaja dikosongkan)***

# DAFTAR GAMBAR

[Gambar 2.1 Contoh Model Ramalan Cuaca WRF 8](#_Toc484351202)

[Gambar 2.2 Peta Jaringan Pos Hujan BMKG 11](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351203)

[Gambar 2.3 Contoh format XML 15](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351204)

[Gambar 2.4 Contoh penerapan K-Nearest Neighbor 17](#_Toc484351205)

[Gambar 2.5 Spektrum Warna RGB 19](#_Toc484351206)

[Gambar 3.1 Diagram Arsitektur Aplikasi Clearroute 22](#_Toc484351207)

[Gambar 3.2 Diagram Komponen Backend-Services 23](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351208)

[Gambar 3.3 Data Citra Ramalan Cuaca BMKG 25](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351209)

[Gambar 3.4 Keterangan Kondisi Cuaca 26](#_Toc484351210)

[Gambar 3.5 External Data Resources 27](#_Toc484351211)

[Gambar 3.6 Diagram Alir Mendapatkan Ramalan Cuaca 29](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351212)

[Gambar 3.7 Diagram Alir Mendapatkan Kondisi Cuaca terkini 30](#_Toc484351213)

[Gambar 3.8 Diagram Alir K-Nearest Neighbor 31](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351214)

[Gambar 3.9 Diagram Alir Menentukan Lokasi Pos Hujan Dan Kondisi Cuaca Pada Rute Perjalanan 34](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351215)

[Gambar 5.1 Hasil Uji Coba Response Time 15 Pengguna Rute A 50](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351216)

[Gambar 5.2 Hasil Uji Coba Response Time 15 Pengguna Rute B 51](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351217)

[Gambar 5.3 Hasil Uji Coba Response Time 15 Pengguna Rute C 51](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351218)

[Gambar 5.4 Hasil Uji Coba Response Time 25 Pengguna Rute B 52](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351219)

[Gambar 5.5 Hasil Uji Coba Response Time 25 Pengguna Rute A 52](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351220)

[Gambar 5.6 Hasil Uji Coba Response Time 35 Pengguna Rute A 53](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351221)

[Gambar 5.7 Hasil Uji Coba Response Time 25 Pengguna Rute C 53](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351222)

[Gambar 5.8 Hasil Uji Coba Response Time 35 Pengguna Rute B 54](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351223)

[Gambar 5.9 Hasil Uji Coba Response Time 35 Pengguna Rute B 54](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351224)

[Gambar 5.10 Hasil Uji Coba Perhitungan Response Time K-Nearest Neighbor 15 Pengguna 55](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351225)

[Gambar 5.11 Hasil Uji Coba Perhitungan Response Time K-Nearest Neighbor 35 Pengguna 56](#_Toc484351226)

[Gambar 5.12 Hasil Uji Coba Perhitungan Response Time K-Nearest Neighbor 25 Pengguna 56](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351227)

[Gambar 5.13 Lokasi Pos Hujan Berdasarkan Perhitungan KNN K=3 58](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351228)

[Gambar 5.14 Data BMKG yang Telah Diunduh 58](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351229)

[Gambar 5.15 Data XML 03:50:37pm.xml 59](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351230)

[Gambar 5.16 Hasil Uji Coba Rute A 60](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351231)

[Gambar 5.17 Hasil Uji Coba Rute B 60](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351232)

[Gambar 5.18 Hasil Uji Coba Rute C 61](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351233)

[Gambar 5.19 Data Uji Coba Ramalan Cuaca 62](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351234)

[Gambar 5.20 Garis Hitam yang Memotong Peta 66](#_Toc484351235)

[Gambar 5.21 Peta Dengan Kondisi Cuaca Cerah 66](#_Toc484351236)

# DAFTAR TABEL

[Tabel 2.1 Lokasi Pos Hujan Kota Surabaya 12](#_Toc484351254)

[Tabel 3.1 Data Kondisi Cuaca yang Sedang Terjadi 24](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351255)

[Tabel 3.2 Skala Keterangan Kondisi Cuaca 26](#_Toc484351256)

[Tabel 3.3 Kode RGB Keterangan Kondisi Cuaca 26](#_Toc484351257)

[Tabel 3.5 Hasil Perhitungan Koordinat Cartesian 32](#_Toc484351258)

[Tabel 3.6 Hasil Perhitungan K-Nearest Neighbor 33](#_Toc484351259)

[Tabel 3.7 Nilai Bobot Berdasarkan Kondisi Cuaca 35](#_Toc484351260)

[Tabel 5.1 Spesifikasi Perangkat Keras Uji Coba 47](#_Toc484351261)

[Tabel 5.2 Response Time 15 Pengguna 49](#_Toc484351262)

[Tabel 5.3 Response Time 25 Pengguna 50](#_Toc484351263)

[Tabel 5.4 Response Time 35 Pengguna 50](#_Toc484351264)

[Tabel 5.5 Uji Coba Response Time Perhitungan K-Nearest Neighbor 55](#_Toc484351265)

[Tabel 5.6 Hasil K=1 57](#_Toc484351266)

[Tabel 5.7 Hasil K=2 57](#_Toc484351267)

[Tabel 5.8 Hasil K=3 57](#_Toc484351268)

[Tabel 5.9 Daftar Ukuran file XML 59](#_Toc484351269)

[Tabel 5.10 Kondisi Cuaca yang Tidak Terdeteksi 63](#_Toc484351270)

***(Halaman ini sengaja dikosongkan)***

# DAFTAR KODE SUMBER

[Kode Sumber 4.1 Proses Mendapatkan Data Ramalan 38](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351271)

[Kode Sumber 4.2 Mendapatkan RGB Dari Gambar 39](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351272)

[Kode Sumber 4.3 Klasifikasi Jenis Cuaca 41](file:///C:\Users\Anwar%20Rosyidi\Google%20Drive\ClearRoute%20menuju%20SW116\Buku\buku%20ta%20anwar2.docx#_Toc484351273)

[Kode Sumber 4.4 Mendapatkan Kondisi Cuaca Terkini 43](#_Toc484351274)

[Kode Sumber 4.5 Menentukan Lokasi Pos Hujan Terdekat dan Kondisi Cuaca 45](#_Toc484351275)

[Kode Sumber 4.6 K-Nearest Neighbor 46](#_Toc484351276)

***(Halaman sengaja dikosongkan)***

# BAB I PENDAHULUAN

## Latar Belakang

Setiap orang pasti pernah melakukan perjalanan. Dalam setiap perjalanan tentu banyak pertimbangan yang akan diperhitungkan untuk melaksanakan perjalanan tersebut, beberapa diantaranya adalah rute yang dilalui dan cuaca. Rute merupakan alur dan arah yang akan dilalui untuk mencapai tempat tujuan, dan cuaca adalah keadaan udara pada saat tertentu dan di wilayah tertentu yang relatif sempit (tidak luas) dan pada jangka waktu yang singkat.

Dewasa kini pengguna tidak perlu repot-repot memikirkan rute mana yang harus diambil, karena hal tersebut sudah dapat dibuat secara langsung oleh komputer. Komputer menentukan rute berdasarkan pilihan-pilihan yang sudah diatur oleh pengguna atau pengembangnya, dan semua pilihan itu diolah menggunakan suatu Algoritma yang dirasa pengembang merupakan algoritma terbaik untuk menentukan rute yang diinginkan.

Di Indonesia, terdapat badan yang secara resmi memiliki tugas untuk mengawasi dan memberikan berita mengenai cuaca yang sedang berlangsung dan akan terjadi, yaitu Badan Meteorologi Klimatologi dan Geofisika (BMKG). BMKG secara *real time* selalu memperbaharui berita cuaca yang ada, agar dapat dilihat oleh masyarakat umum. Dan untuk mendapatkan data infrormasi cuaca BMKG kota Surabaya menggunakan acuan beberapa Pos Hujan yang biasanya merupakan kantor desa atau tempat umum besar lainnya untuk memberikan informasi mengenai cuaca di sekitar pos hujan tersebut.

Dengan menggabungkan rute yang dibuat dan data cuaca yang didapat dari BMKG, masyarakat tidak perlu lagi mencari informasi cuaca dan menentukan rute dari sumber atau aplikasi yang berbeda, dan hal tersebut akan membuat masyarakat menjadi lebih mudah menjalankan aktivitasnya.

## Rumusan Masalah

Tugas Akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana implementasi metode *K-Nearest Neighbor* untuk menentukan pos hujan terdekat dengan beberapa titik rute yang telah ditentukan?
2. Bagaimana menentukan rute terbaik berdasarkan bobot curah hujan terendah yang didapat dari pos hujan terdekat?
3. Bagaimana cara mendapatkan data kondisi cuaca secara *real time* dari BMKG?
4. Bagaimana cara mengirimkan informasi cuaca yang telah diolah ke aplikasi Clearroute?

## Batasan Permasalahan

Permasalahan yang dibahas pada Tugas Akhir ini memiliki batasan sebagai berikut:

1. Metode KNN hanya digunakan pada pos hujan wilayah kota Surabaya.
2. Sumber data yang digunakan hanya berasal dari data cuaca BMKG.
3. Daerah yang bisa dipetakan hanya wilayah kota Surabaya.

## Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Mengimplementasikan metode *K-Nearest Neighbor* pada penentuan beberapa titik rute yang terdekat dengan lokasi pos hujan.
2. Mendapatkan data kondisi cuaca saat ini dan data kondisi ramalan cuaca sesuai waktu yang diinginkan.
3. Memberikan bobot pada rute yang diberikan berdasarkan keadaan cuaca yang didapat dari Pos Hujan.
4. Memilih rute terbaik yang diberikan berdasarkan bobot cuaca yang sedang terjadi

.

## Manfaat Tugas Akhir

Manfaat yang diperoleh dari pembuatan Tugas Akhir ini adalah dapat membantu masyarakat untuk mengetahui kondisi cuaca pada jalur perjalanan yang akan dilaluinya. Sehingga pengguna dapat mempersiapkan segala kebutuhan perjalananannya sesuai dengan kondisi cuaca yang akan dilalui.

## Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

### Penyusunan Proposal Tugas Akhir

Tahapan awal dari Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir berisi pendahuluan, deskripsi dan gagasan metode-metode yang dibuat dalam Tugas Akhir ini. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir ini. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Terdapat pula subbab jadwal kegitan yang menjelaskan jadwal pengerjaan Tugas Akhir.

### Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi menggunakan literatur *paper* yang berasal dari jurnal internasional yaitu IEEE untuk mencari informasi dan teori-teori yang berhubungan dengan metode *K-Nearest Neighbor* yang dapat digunakan sebagai acuan dalam mengerjakan Tugas Akhir ini. Kemudian digunakan juga pustaka pendukung lainnya yaitu informasi mengenai sumber data cuaca yang berasal dari BMKG. Selain itu juga dilakukan perumusan awal mengenai proses distribusi data yang akan diimplementasikan pada sistem yang akan dibuat.

### Implementasi Perangkat Lunak

Pada tahap ini dilakukan tahap untuk membangun metode-metode yang telah diajukan pada proposal Tugas Akhir. Implementasi Tugas Akhir ini menggunakan bahasa pemrograman PHP dengan kerangka kerangka kerja Laravel versi 5.4 dan basis data yang digunakan ialah basis data spasial dari postgreSQL.

### Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap sistem yang dibangun. Pengujian dan evaluasi sistem dilakukan untuk mencari masalah yang mungkin timbul dan melakukan perbaikan jika ditemukan kesalahan pada sistem.

### Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

## Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

* + 1. Bab I. Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

* + 1. Bab II. Tinjauan Pustaka

Bab ini berisi penjelasan mengenai dasar-dasar penunjang yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

* + 1. Bab III. Perancangan Modul

Bab ini berisi pembahasan mengenai perancangan dari metode *K-Nearest Neighbor* dan penjelasan mengenai sumber data yang digunakan serta alur kerja dari sistem yang dibangun untuk Tugas Akhir ini.

* + 1. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari proses pengelolaan data cauca, dan metode *K-Nearest Neighbor* untuk klasifikasi.

* + 1. Bab V. Hasil Uji Coba dan Evaluasi

Bab ini berisikan hasil uji coba dari metode *K-Nearest Neighbor* untuk menentukan lokasi pos hujan terdekat yang sudah diimplementasikan pada kode sumber.

* + 1. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

* + 1. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

* + 1. Lampiran

Dalam lampiran terdapat tabel-tabel data hasil uji coba dan tabel-tabel sumber data secara keseluruhan.

***(Halaman ini sengaja dikosongkan)***

# BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan pustaka penunjang yang berkaitan dengan pembuatan Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap aplikasi dan berguna sebagai penunjang dalam pengembangan sistem.

## Sistem Peramalan Cuaca

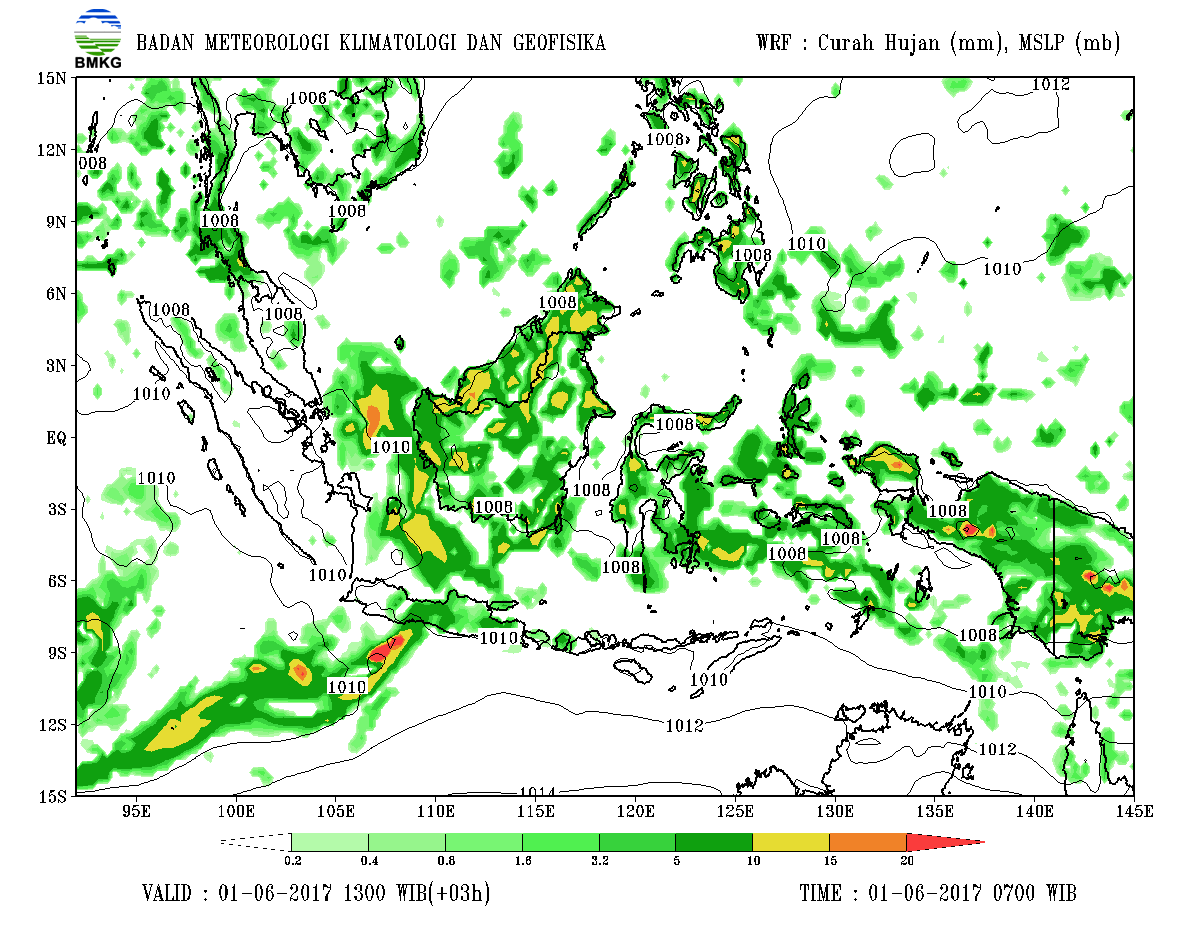
Peramalan cuaca adalah suatu cabang aplikasi sains dan teknologi untuk memprediksi kondisi atmosfer pada waktu yang akan datang pada suatu lokasi tertentu. Dalam menentukan prediksi suatu ramalan cuaca terdapat beberapa unsur yang perlu diamati yaitu:

1. Tekanan udara.
2. Suhu udara
3. Arah angin
4. Kelembaban udara

Dalam melakukan peramalan cuaca dapat dilakukan secara objektif dan subjektif. Metode subjektif adalah prediksi yang dibuat berdasarakan pertimbangan atau penilaian dari seseorang yang melakukan pengamatan cuaca atau yang disebut sebagai seorang prakirawan.

Ramalan cuaca yang dikeluarkan selama ini dibuat berdasarkan hasil pengamatan cuaca terakhir, analisis data dari citra radar cuaca dan pengalaman dari seorang prakirawan dalam menentukan ramalan cuaca. Data hasil ramalan cuaca yang dilakukan oleh prakirawan akan dibuat suatu model numerik yang bisa diamati oleh masyarakat umum.

Saat ini jenis model ramalan cuaca telah banyak digunakan untuk menghasilkan informasi ramalan cuaca salah satu model cuaca yang banyak digunakan ialah model WRF (*Weather Research Forecasting*). Model WRF ini menggambarkan kondisi cuaca berdasarkan kondisi atmosfer yang sedang terjadi. Penggambaran kondisi atmosfer oleh WRF langsung digambarkan dalam suatu peta wilayah yang kondisi ramalan cuacanya sedang diamati. Saat ini lembaga resmi pemerintah Indonesia yang menangani pengamatan cuaca telah menggunakan model cuaca WRF untuk menampilkan hasil pengamatan ramalan cuaca. Data model WRF yang dikeluarkan memiliki masa berlaku 3 jam sehingga setiap jeda waktu 3 jam model pengamatan WRF akan diperbarui [1]. Contoh data model pengamatan WRF dapat dilihat pada Gambar 2.1.



**Gambar 2.1 Contoh Model Ramalan Cuaca WRF**

## Badan Meteorologi Klimatologi dan Geofisika (BMKG)

BMKG merupakan suatu lembaga pemerintah non Departemen yang memiliki tugas untuk melaksanakan tugas pemerintahan di bidang meteorologi, klimatologi dan geofisika. Sejarah berdirinya BMKG dimulai pada tahun 1841 diawali dengan pengamatan yang dilakukan secara perorangan oleh Dr. Onnen, Kepala Rumah Sakit di Bogor. Pada tahun 1866, kegiatan pengamatan perorangan tersebut oleh Pemerintah Hindia Belanda diresmikan menjadi instansi pemerintah dengan nama Observatorium Magnetik, dan Meteorologi yang dipimpin oleh Dr. Bergsma.

Pada tahun 1949, setelah penyerahan kedaulatan negara Republik Indonesia dari Belanda,  diubah menjadi Jawatan Meteorologi dan Geofisika di bawah Departemen Perhubungan, dan Pekerjaan Umum.

Selanjutnya pada tahun 1950, Indonesia secara resmi masuk sebagai anggota Organisasi Meteorologi Dunia (World Meteorological Organization atau WMO) dan Kepala Jawatan Meteorologi, dan Geofisika menjadi Permanent Representative of Indonesia with WMO.

Pada tahun 1972, Direktorat Meteorologi, dan Geofisika diganti namanya menjadi Pusat Meteorologi dan Geofisika, suatu instansi setingkat eselon II di bawah Departemen Perhubungan, yang pada tahun 1980 statusnya dinaikkan menjadi suatu instansi setingkat eselon I dengan nama Badan Meteorologi, dan Geofisika, dengan kedudukan tetap berada dibawah Departemen Perhubungan.

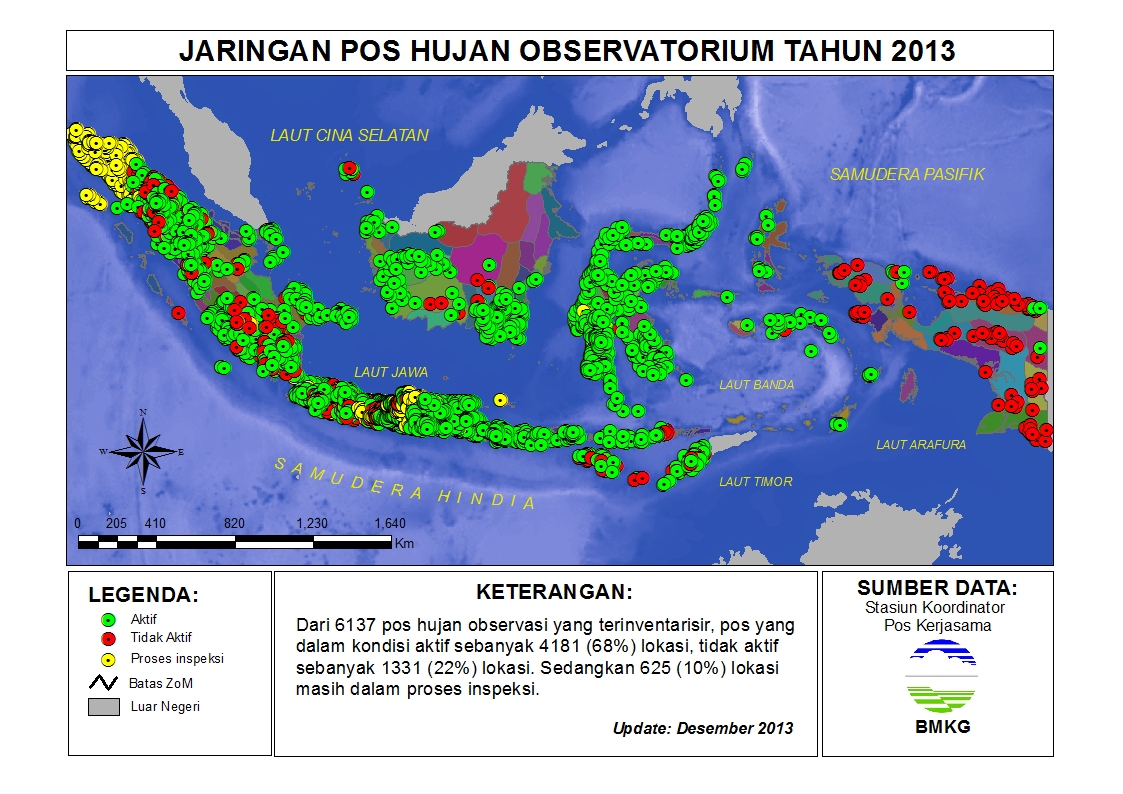
Pada tahun 2002, melalui Keputusan Presiden RI Nomor 46, dan 48 tahun 2002, struktur organisasinya diubah menjadi Lembaga Pemerintah Non Departemen (LPND) dengan nama tetap Badan Meteorologi dan Geofisika[2].

Terakhir, melalui Peraturan Presiden Nomor 61 Tahun 2008, BMG berganti nama menjadi Badan Meteorologi, Klimatologi*,* dan Geofisika dengan status tetap sebagai Lembaga Pemerintah Non Departemen

Dalam melaksanakan tugasnya BMKG menyelenggarakan fungsi berupa:

* Perumusan kebijakan nasional dan kebijakan umum di bidang meteorologi, klimatologi, dan geofisika.
* Perumusan kebijakan teknis di bidang meteorologi, klimatologi, dan geofisika.
* Koordinasi kebijakan, perencanaan dan program di bidang meteorologi, klimatologi, dan geofisika.
* Pelaksanaan, pembinaan dan pengendalian observasi, dan pengolahan data dan informasi di bidang meteorologi, klimatologi, dan geofisika.
* Pelayanan data dan informasi di bidang meteorologi, klimatologi, dan geofisika.
* Penyampaian informasi kepada instansi dan pihak terkait serta masyarakat berkenaan dengan perubahan iklim.
* Penyampaian informasi dan peringatan dini kepada instansi dan pihak terkait serta masyarakat berkenaan dengan bencana karena factor meteorologi, klimatologi, dan geofisika.
* Pelaksanaan kerja sama internasional di bidang meteorologi, klimatologi, dan geofisika.
* Pelaksanaan penelitian, pengkajian, dan pengembangan di bidang meteorologi, klimatologi, dan geofisika
* Pelaksanaan, pembinaan, dan pengendalian instrumentasi, kalibrasi, dan jaringan komunikasi di bidang meteorologi, klimatologi, dan geofisika.
* Koordinasi dan kerja sama instrumentasi, kalibrasi, dan jaringan komunikasi di bidang meteorologi, klimatologi, dan geofisika.
* Pelaksanaan pendidikan dan pelatihan keahlian dan manajemen pemerintahan di bidang meteorologi, klimatologi, dan geofisika.
* Pelaksanaan pendidikan profesional di bidang meteorologi, klimatologi, dan geofisika.
* Pelaksanaan manajemen data di bidang meteorologi, klimatologi, dan geofisika.
* Pembinaan dan koordinasi pelaksanaan tugas administrasi di lingkungan BMKG.
* Pengelolaan barang milik/kekayaan negara yang menjadi tanggung jawab BMKG.
* Pengawasan atas pelaksanaan tugas di lingkungan BMKG[3].

## Pos Hujan BMKG

Dalam melakukan pengamatan curah hujan BMKG memiliki pos-pos pengamatan curah hujan yang disebut sebagai pos hujan. Untuk kota Surabaya pos hujan yang dimiliki tersebar pada setiap kelurahan atau desa. Fungsi utama dari pos hujan ini ialah merekam curah hujan yang terjadi secara *real time*. Pos hujan akan melakukan perekaman data setiap 10 menit sekali yang kemudian dikirimkan kepada stasiun pengamatan cuaca induk. Untuk kota Surabaya sendiri stasiun pengamatan induk berada di kawansan bandara Juanda. Persebaran lokasi pos hujan di kota Surabaya dapat dilihat pada Tabel 2.1 dan terlampir pada lampiran A.1[4]. Gambar 2.2 menunjukan persebaran titik lokasi pos hujan yang dimiliki oleh BMKG pada tahun 2013.

**Gambar 2.2 Peta Jaringan Pos Hujan BMKG**

**Tabel 2.1 Lokasi Pos Hujan Kota Surabaya**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Pos** | **ydesimal** | **xdesimal** | **y*pixel*** | **y*pixel*** |
| Kelurahan Asemrowo | -7.252 | 112.715 | 280 | 438 |
| Kelurahan Genting | -7.24407 | 112.7166 | 269 | 440 |
| Kelurahan Greges | -7.2295 | 112.684 | 248 | 395 |
| Kelurahan Kalianak | -7.2289 | 112.702 | 248 | 420 |
| Kelurahan Tambak Langon | -7.225 | 112.664 | 242 | 367 |

## *Application Program Interface* (API)

*Application Program Interface* (API) adalah kumpulan dari rutinitas, protocol dan alat untuk membangun aplikasi perangkat lunak. Sebuah API menspesifikasikan cara komponen Aplikasi saling berinteraksi. API memungkinkan seorang pengembang aplikasi untuk menggunakan fungsi standar untuk melakukan interaksi dengan sistem operasi. API menyediakan fungsi dan perintah dengan bahasa yang lebih terstruktur dan lebih mudah dipahami oleh pengembang aplikasi[5].

## *Hypertext Preprocessor* (PHP)

PHP: *Hypertext Prepocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak digunakan untuk memrogram web dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Pada awalnya PHP merupakan kependekan dari *personal home page*. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada saat itu PHP masih berwujud sekumpulan skrip yang digunakan untuk mengolah formulir dari halaman *web*.

Selanjutnya Rasmus Lerdorf merilis kode sumber PHP tersebut dan perilisan kode ini dibuat *open source* sehingga banyak pengembang yang tertarik untuk ikut mengembangkan PHP. Pada tahun 1997 sebuah perusahaan bernama Zend menulis ulang interpeter PHP menjadi lebih baik dan lebih cepat. Kemudian pada tahun 1998 perusahaan merilis interpeter baru tersebut dan diberi nama PHP 3.0. Selanjutnya pada tahun 1999 perusahaan tersebut merilis PHP 4.0 dan hingga saat ini versi PHP yang telah dirilis sudah mencapai versi PHP 7.0[6].

Kelebihan PHP dari bahasa pemrograman web yang lain:

* Bahasa pemrograman PHP adalah sebuah bahasa skrip yang tidak melakukan sebuah kompilasi dalam penggunaannya.
* *Web Server* yang mendukung PHP mudah didapatkan seperti apache, IIS, Lighttpd, dan Xitami dengan konfigurasi yang relative murah.
* Dalam sisi pemahaman, PHP adalah bahasa skrip yang paling mudah karena memiliki referensi yang banyak.
* PHP adalah bahasa *open source*  yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat menjalankan perintah-perintah sistem.

## *JavaScript Object Notation* (JSON)

JSON adalah format pertukaran data yang sangat ringan. Data mudah dibaca dan ditulis oleh manusia. Data mudah dipecah dan dibuat oleh mesin. JSON dibuat berdasarkan subset dari Bahasa Pemrograman JavaScript. JSON adalah format tulisan yang independen tetapi masih menggunakan konvensi yang familiar untuk programmer Bahasa C beserta keluarganya, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan banyak lagi. Sifat tersebut menyebabkan JSON sebagai jenis pertukaran data yang ideal antar bahasa[7]. JSON sendiri terdiri dari dua struktur yaitu:

* Kumpulan pasangan nama/nilai yang biasanya dinyatakan sebagai objek, *record*, struktur,*dictionary*, tabel hash, dan *key list* atau *associative array*.
* Daftar nilai yang terurutkan dalam hal ini biasanya dinyatakan dalam *array, vector, list* dan *sequence*.

## *Extensible Markup Language* (XML)

XML adalah bahasa *markup* untuk keperluan umum yang disarankan oleh W3C (sebuah konsorsium yang bekerja untuk mengembangkan standar-standar untuk *world wide web*). Berfungsi sebagai sarana pertukaran data antar sistem yang tidak saling berkesesuaian seperti pertukaran data pada sistem komputer dengan basis data yang memiliki format penyimpanan yang berbeda. XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Selanjutnya XML menyediakan suatu cara terstandarisasi namun dapat dimodifikasi untuk menggambarkan isi dari dokumen[8].

Dalam praktiknya XML memiliki beberapa keunggulan yaitu:

* XML dapat menangani berbagai tingkat kompleksitas data
* Pemeliharaan yang mudah
* Sederhana
* Memiliki *portability* yang baik.

Di dalam XML kita dapat bebas mementukan format penyimpanan yang digunakan. Seperti dalam mendefinisikan jenis *tag* yang digunakan pada contoh Gambar 2.1 dapat dilihat contoh sederhana dari format XML. Pada contoh tersebut terdapat sebuah *tag* yang ditandai dengan tanda < > kemudian di dalam *tag* tersebut terdapat sebuah elemen yang bisa dimodifikasi seusai dengan kebutuhan data seperti halnya contoh yang ditunjukan dengan elemen resep. Kemudian kata yang terdapat di antara dua *tag* disebut dengan *attribute* yang menyatakan isi dari dokumen XML yang dibuat seperti halnya contoh pada *tag* <judul> terdapat *attribute* roti tawar.



**Gambar 2.3 Contoh format XML**

## PostgreSQL

PostgreSQL adalah sebuah basis data relasional yang termasuk dalam kategori *open source software*. Basis data ini dapat bekerja pada semua sistem operasi yang tersedia seperti Linux, Unix, dan Windows. PostgreSQL memiliki kemampuan penuh untuk mendukung perintah-perintah SQL seperti:

• *Foreign key*

*• Joins*

*• Views*

*• Triggers*

*• Stored procedures*

dan juga memiliki dukungan penuh pada tipe data SQL:2008. Basis data ini juga memiliki kemampuan untuk menyimpan *object* data biner yang besar seperti gambar, video, dan suara.

PostgreSQL pertama kali dikembangkan oleh universitas California pada tahun 1982 dibawah arahan professor Michael Stonebraker. Awalnya basis data ini diberi nama postgres tetapi karena banyak tambahan fungsi-fungsi SQL maka pada tahun 1995 namanya diganti menjadi Postgres95, dan semenjak 1996 hingga sekarang nama yang digunakan ialah PostgreSQL.

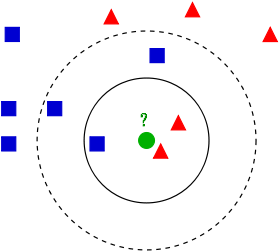
Perbedaan yang mendasar antara PostgreSQL dengan sistem relasional basis data yang lain ialah kemampuan PostgreSQL yang memungkinkan pengguna untuk mendefinisikan SQL-nya sendiri, terutama dalam pembuatan fungsi[7][8].

Saat ini basis data PostgreSQL banyak digunakan oleh perusahaan-perusaahan besar[11] seperti:

* Apple
* Fujitsu
* Cisco
* Sun Microsystems
* dll

## *K-Nearest Neighbor* (KNN)

*K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap suatu objek berdasarkan data pembelajaran yang memiliki jarak paling dekat dengan objek tersebut. Data pembelajaran akan diproyeksikan ke dalam K ruang berdimensi banyak, yang masing-masing dimensi merepresentasikan fitur dari data. Ruang ini akan dibagi-bagi menjadi suatu bagian berdasarkan klasifikasi yang telah dilakukan terhadap data pembelajaran. Contoh dari penerapan metode *K-Nearest Neighbor* dijelaskan pada Gambar 2.4.



**Gambar 2.4 Contoh penerapan K-Nearest Neighbor**

Pada Gambar 2.2 terdapat sebuah lingkaran berwarna hijau yang akan diuji ke dalam kelas manakah lingkaran tersebut? Apakah termasuk kedalam kelas segitiga berwarna merah atau masuk ke dalam kelas persegi berwarna biru? Jika nilai K=3 (digambarkan pada lingkaran garis pertama) lingkaran berwarna hijau akan masuk ke dalam kelas segitiga berwarna merah karena di dalam lingkaran tersebut terdapat 2 buah segitiga dan 1 buah persegi. Jika nilai K=5 (digambarkan pada lingkaran dengan garis putus-putus) maka lingkaran berwarna hijau akan masuk ke dalam kelas persegi berwarna biru karena di sana terdapat 3 buah persegi dan hanya ada 2 segitiga[10][11].

Langkah-langkah yang dilakukan untuk melakukan metode *K-Nearest Neighbor* ada sebagai berikut:

1. *Generate* data sampel yang akan digunakan sebagai data *training*
2. Inisialisasi K titik sebagai titik-titik pusat (*centroids*) awal
3. Hitung jarak setiap objek data set dengan data *training* menggunakan perhitungan *Euclidean* distance. Rumus untuk menghitung jarak antar dua titik sebagai titik data set dengan sebagai titik data *training* ditunjukkan pada Persamaan 2.1:

(2.1)

Apabila titik koordinat (x,y) dalam satuan koordinat *latitude* dan *longitude* maka terlebih dahulu nilai (x,y) tersebut diubah ke dalam satuan koordinat *Cartesian* seperti yang ditunjukkan pada Persamaan 2.2 dan Persamaan 2.3.

(2.2)

(2.3)

Keterangan:

* R = Jari-jari bumi (6371 km).
* lat = Nilai *latitude*.
* lon = Nilai *longitude*.

## *Geography JSON (GeoJSON)*

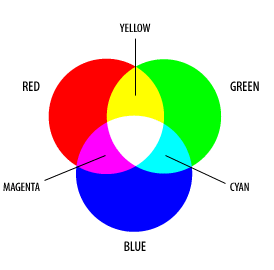
GeoJSON adalah format untuk berbagai jenis struktur data geografi. Sebuah objek GeoJSON dapat merepresentasikan geometri, fitur, dan koleksi dari fitur-fitur. GeoJSON mendukung tipe geometri *Point*, *LineString*, *Polygon*, *MultiPoint*, *MultiLineString*, *MultiPolygon*, dan *GeometryCollection*. Fitur di dalam GeoJSON berisi sebuah objek geometri dan properti tambahan, serta sebuah koleksi fitur yang merepresentasikan sebuah daftar dari banyak fitur.

Sebuah GeoJSON yang utuh selalu dalam bentuk objek. Di dalam GeoJSON, sebuah objek terdiri atas sebuah koleksi dari nama/nilai yang sepasang, yang juga disebut sebagai anggota. Dalam setiap anggota, nama akan selalu dalam tipe *string*. Nilai dari anggota dapat berupa *string*, *number*, *object*, *array*, dan salah satu dari: *true*, *false*, dan *null*[14]*.*

## RGB

RGB adalah suatu model warna yang terdiri atas 3 buah warna yaitu *red, green* dan *blue*, yang ditambahkan beberapa teknik pencampuran untuk menghasilkan warna yang beraneka ragam. RGB merupakan model warna *additive*, yaitu ketiga berkas cahaya yang ditambahkan bersama-sama, dengan menambahkan panjang gelombang, untuk membuat spektrum warna akhir.

Sebuah warna dalam RGB digambarkan dengan menentukan seberapa banyak masing-masing warna merah, biru dan hijau yang dicampurkan dan direpresentasikan dalam bentuk numerik di dalam triplet (r, g, b). nilai yang digunakan disimpan dalam angka *integer* antara 0 sampai 255 kemudia disimpan dalam bentuk hexadesimal setiap nilai mendefinisikan besaran warna masing-masing yang dicampurkan[15]. Spektrum warna RGB dapat dilihat pada Gambar 2.5.



**Gambar 2.5 Spektrum Warna RGB**

***(Halaman ini sengaja dikosongkan)***

# BAB III PERANCANGAN MODUL

Bab ini menjelaskan tentang perancangan dan pembuatan modul penentuan lokasi pos hujan terdekat dengan rute perjalanan pada aplikasi Clearroute. Tugas Akhir ini diawali dari persiapan data yang akan digunakan yaitu data lokasi pos hujan, data kondisi cuaca yang sedang terjadi, data ramalan cuaca dan data rute perjalanan yang akan dilalui. Data-data tersebut akan diproses sehingga dapat menghasilkan informasi kondisi cuaca yang sedang atau akan terjadi pada rute perjalanan yang akan dilewati.

## Deskripsi Umum

Tugas akhir ini merupakan sub-bagian dari aplikasi berbasis perangkat bergerak Clearroute yang secara khusus menangani pemrosesan data dan *transfer* data antara sistem Clearroute dengan sistem informasi cuaca yang dimiliki oleh BMKG.

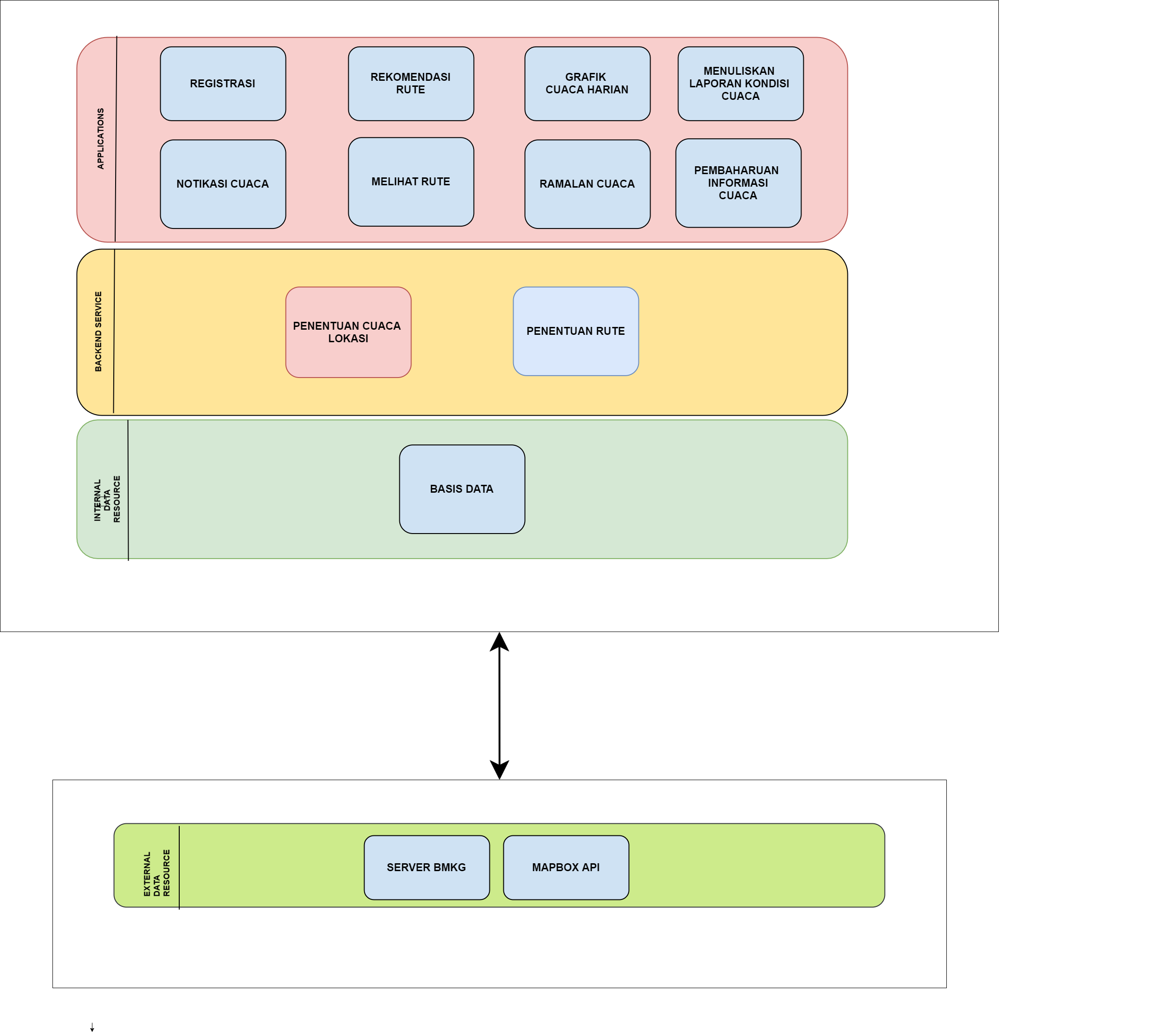
### Arsitektur Sistem

Pada Gambar 3.1 ditunjukkan rancangan arsitektur aplikasi Clearroute secara umum. Sistem yang akan dibangun terdiri dari 4 bagian yaitu:

1. *Applications*
2. *Backend-Services*
3. *Internal data resources*
4. *External data resources*

Keempat bagian tersebut saling berhubungan satu sama lainnya agar aplikasi Clearroute yang dibangun dapat bekerja dengan baik.

Pada *external data resources* terdapat sumber data yang berasal dari BMKG sehingga sistem yang dibangun tidak menangani pengolahan data BMKG dan hanya menerima informasi cuaca yang sudah jadi dari pusat data BMKG.



**Gambar 3.1 Diagram Arsitektur Aplikasi Clearroute**

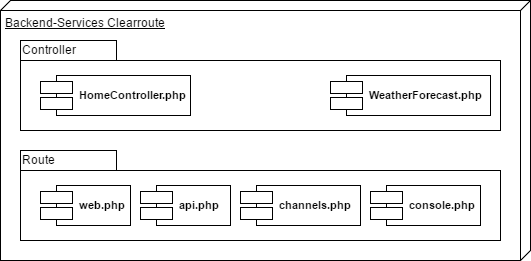
#### *Applications*

Bagian ini merupakan fitur-fitur yang terdapat pada aplikasi Clearroute itu sendiri. Pada aplikasi sendiri terdapat beberapa fitur yang dimiliki antara lain:

* Registrasi
* Rekomendasi rute
* Grafik cuaca harian
* Menuliskan laporan cuaca
* Notifikasi cuaca
* Pembaruan informasi cuaca

Pada Tugas Akhir ini akan dibangun sebuah sistem pengolahan data agar seluruh fitur dari aplikasi Clearroute dapat berjalan dengan baik. Fokus utama dari Tugas Akhir berada pada pengembangan *backend services* dan pengolahan data pada *internal data resources* dan *external data resources* untuk menunjang jalannya aplikasi.

#### *Backend services*

Bagian ini merupakan sistem yang akan menangani dua hal utama pada *server* aplikasi Clearroute, yaitu penentuan kondisi cuaca pada rute perjalanan, dan penentuan rute perjalanan.

**Gambar 3.2 Diagram Komponen Backend-Services**

Gambar 3.2 menjelaskan bagian-bagian dari *backend-services* pada aplikasi aplikasi Clearroute. Pada modul penentuan kondisi cuaca semua fungsi terdapat pada *file* HomeController.php dan untuk modul penentuan ramalan cuaca semua fungsi terdapat pada WeatherForecast.php. Kedua *file* tersebut terdapat pada komponen *controller* pada kerangka kerja Laravel. Kemudian alamat URL untuk melakukan akses pada dua *file* tersebut diatur dalam *file* web.php dalam komponen *Route*.

#### *Internal Data Resources*

Bagian ini bertanggung jawab utnuk mengatur basis data yang akan digunakan oleh aplikasi Clearroute. Jenis basis data yang digunakan adalah PostgreSQL dengan penambahan ekstensi PostGIS dan pgRouting sehingga basis data tersebut dapat mengolah data spasial yang akan digunakan pada tugas akhir ini.

Lampiran B memperlihatkan seluruh isi basis data yang digunakan untuk menyelesaikan Tugas Akhir ini. Semua tabel yang terdapat pada lampiran B kecuali datapos, rekaman dan backup\_ways adalah tabel yang otomatis dibuat ketika melakukan *import* data dari Openstreetmap ke dalam basis data PostgreSQL.

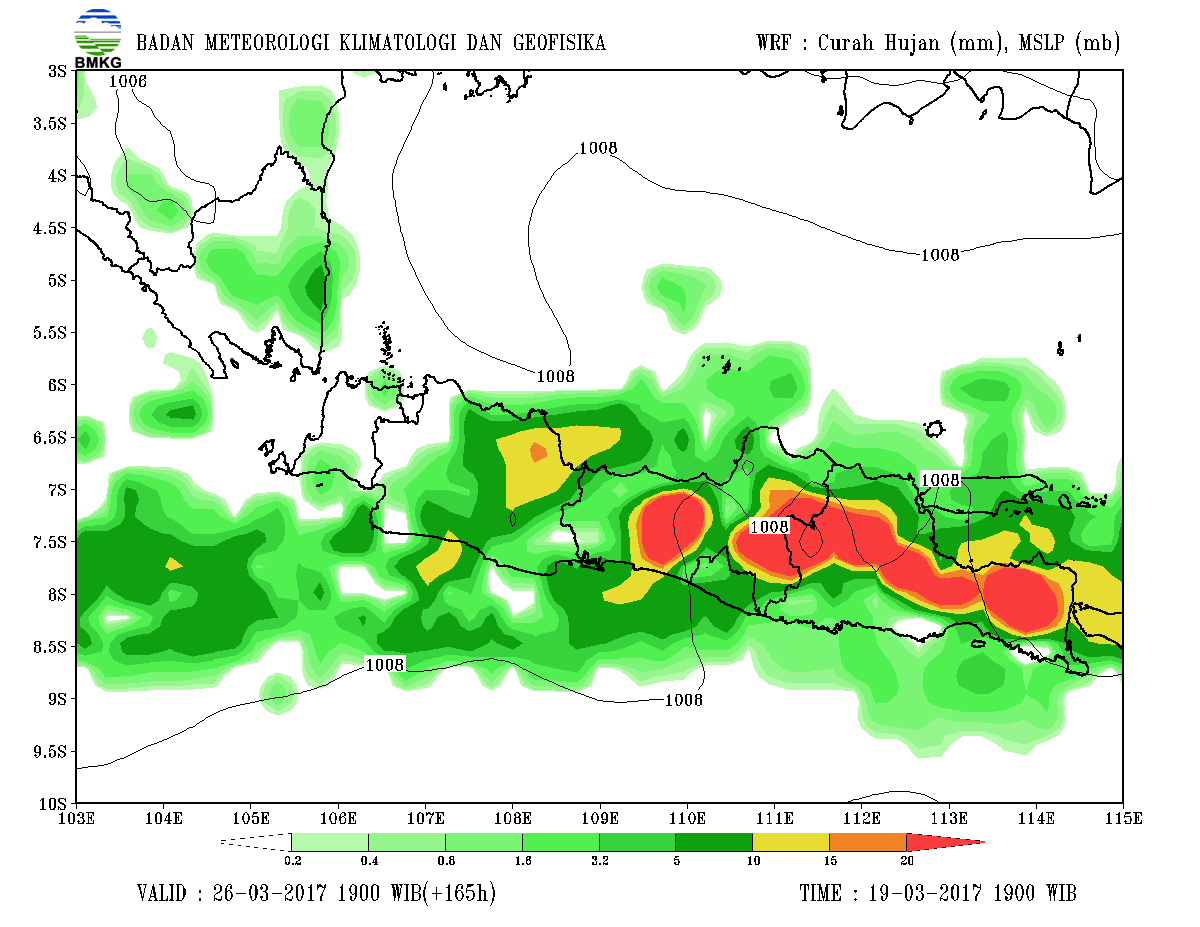
#### *External Data Resource*

**Tabel 3.1 Data Kondisi Cuaca yang Sedang Terjadi**

Bagian ini merupakan sumber data luar yang digunakan oleh aplikasi Clearroute. Pada pengerjaan tugas akhir ini sumber data yang digunakan ialah data cuaca yang dimiliki oleh BMKG. Terdapat dua jenis data yang akan digunakan yaitu data kondisi cuaca yang sedang terjadi dan data ramalan cuaca yang akan terjadi seperti yang ditunjukan oleh Tabel 3.1. Pada pengumpulan data kondisi cuaca yang sedang terjadi. BMKG menggunakan alat-alat yang terdapat pada pos hujan di mana pos hujan tersebut ditempatkan di seluruh keluarahan pada kota Surabaya. Data-data tersebut kemudian diolah oleh pihak BMKG dan informasi mengenai kondisi cuaca yang sedang terjadi akan dikirimkan melalui protokol HTTP *Request* menggunakan format data XML.

Untuk data ramalan cuaca BMKG mengeluarkan informasinya setiap pukul 00.00 WIB dan data ramalan BMKG tersebut berlaku setiap 3 jam sehingga untuk data ramalan selama 1 hari BMKG akan mengeluarkan 8 data. Data yang dikeluarkan BMKG untuk menginformasikan ramalan cuaca adalah data citra sehingga diperlukan suatu proses untuk mengolah data citra tersebut untuk mendapatkan informasi yang diinginkan.

Gambar 3.3 menunjukan kondisi ramalan cuaca yang akan terjadi pada tanggal 26 Maret 2017. Pada Gambar 3.4, Tabel 3.2 dan Tabel 3.3 ditunjukkan keterangan kondisi cuaca yang digambarkan pada peta ramalan cuaca yang dikeluarkan oleh BMKG.



**Gambar 3.3 Data Citra Ramalan Cuaca BMKG**



**Gambar 3.4 Keterangan Kondisi Cuaca**

**Tabel 3.2 Skala Keterangan Kondisi Cuaca**

|  |  |
| --- | --- |
| **Skala** | **Kondisi cuaca** |
| 0.2-5 | Berawan |
| 5-10 | Hujan ringan |
| 10-15 | Hujan sedang |
| 15-20 | Hujan deras |
| 20 > | Hujan badai |

**Tabel 3.3 Kode RGB Keterangan Kondisi Cuaca**

|  |  |
| --- | --- |
| **Kode RGB** | **Kondisi Cuaca** |
| #b4faaa | Berawan |
| #96f58c | Berawan |
| #78f573 | Berawan |
| #50f050 | Berawan |
| #37d23c | Berawan |
| #0fa00f | Hujan Ringan |
| #fa0f | Hujan Ringan |
| #e6dc32 | Hujan Sedang |
| #f08228 | Hujan Deras |
| #fa3c3c | Hujan Badai |

Secara umum sumber data yang digunakan pada aplikasi Clearroute dapat digambarkan seperti Gambar 3.5 berikut:

C:\Users\RIDHO\AppData\Local\Microsoft\Windows\INetCache\Content.Word\external-resources.png

**Gambar 3.5 External Data Resources**

Gambar 3.5 menjelaskan agar Aplikasi Clearroute dapat berjalan, diperlukan 3 sumber data, sumber data internal yaitu Clearroute API, dan 2 sumber data eksternal yaitu dari *Server* BMKG untuk ramalan dan kondisi cuaca, serta Mapbox API untuk menggambarkan peta dan rute yang akan digunakan pada tugas akhir ini.

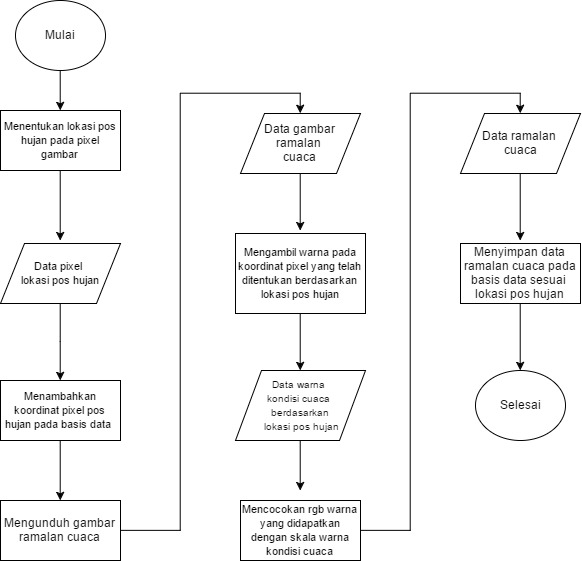
## Desain Umum Sistem

Pada sub bab ini akan dijelaskan mengenai desain proses pada modul yang dikerjakan untuk tugas akhir ini.

### Mendapatkan Data Ramalan Cuaca

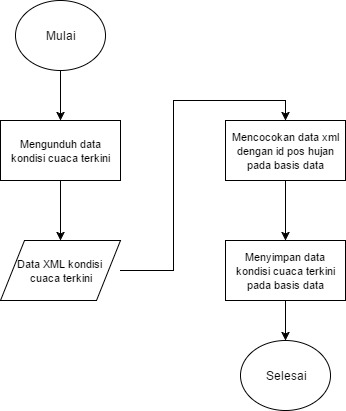
Seperti yang ditunjukkan pada Gambar 3.3 data ramalan cuaca yang dimiliki oleh BMKG masih berbentuk data citra sehingga diperlukan suatu proses ekstraksi warna gambar untuk mendapatkan data cuaca yang bisa digunakan untuk sistem. Proses untuk melakukan proses tersebut dapat digambarkan melalui diagram alir pada Gambar 3.6. Pada Gambar 3.6 terdapat beberapa langkah yang dilakukan pada tahap ini yaitu:

1. Tahap pertama ialah menentukan lokasi pos hujan pada *pixel* yang terdapat pada gambar. Pada tahap ini BMKG sudah menentukan lokasi *pixel* dari pos hujan yang ada di surabaya sehingga data tersebut bisa digunakan dan disimpan pada basis data yang menyimpan informasi mengenai lokasi pos hujan.
2. Tahap kedua ialah mengunduh data gambar ramalan cuaca dari *server* BMKG. Data tersebut dapat diunduh melalui alamat <http://diseminasi.meteo.bmkg.go.id/wrf/>. Pada tugas akhir ini telah dibuat sistem untuk mengunduh data tersebut secara otomatis setiap jam 00.00 W.I.B.
3. Tahap ketiga ialah mengambil RGB warna pada gambar ramalan cuaca yang telah diunduh sebelumnya. Data RGB warna diambil merupakan data RGB yang terletak pada *pixel* yang telah ditentukan pada tahap pertama.
4. Tahap keempat ialah mencocokan RGB warna yang telah diambil berdasarkan RGB pada skala warna yang menggambarkan kondisi cuaca untuk mendapatkan kondisi cuaca yang dipetakan apakah cerah, berawan atau hujan.
5. Tahap terakhir ialah menyimpan data kondisi cuaca yang telah didapatkan pada tahap empat ke dalam basis data.



**Gambar 3.6 Diagram Alir Mendapatkan Ramalan Cuaca**

### Mendapatkan Data Kondisi Cuaca Terkini



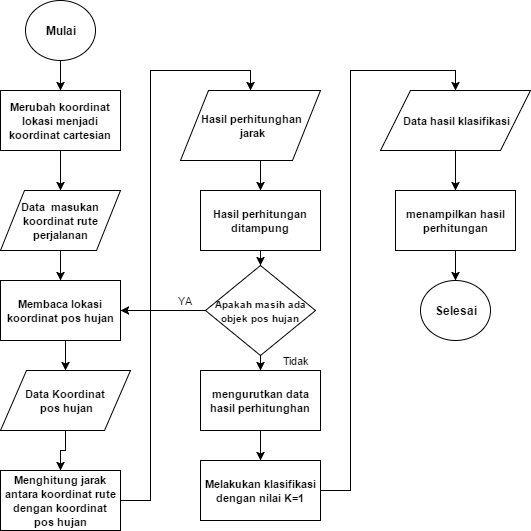
**Gambar 3.7 Diagram Alir Mendapatkan Kondisi Cuaca terkini**

Pada Gambar 3.7 terdapat diagram alir yang menjelaskan langkah-langkah untuk mendapatkan kondisi cuaca terkini berdasarkan lokasi pos hujan. Proses ini memiliki beberapa tahap yaitu:

1. Tahap pertama ialah mengunduh data XML kondisi cuaca terkini dari *server* BMKG. Data tersebut dapat diunduh melalui alamat [www.bmkgjuanda.ip-dynamic.com:81/sby/rekaman\_tenminute.xml](http://www.bmkgjuanda.ip-dynamic.com:81/sby/rekaman_tenminute.xml).
2. Tahap kedua ialah dari data XML yang telah diunduh kemudian dicocokan dengan id pos hujan yang terdapat pada basis data yang menyimpan data lokasi pos hujan untuk mendapatkan data kondisi cuaca yang sedang terjadi.
3. Terakhir data yang telah didapatkan pada tahap ketiga disimpan dan ke dalam basis data yang menyimpan informasi kondisi cuaca terkini.

Proses untuk mendapatkan kondisi cuaca ini dilakukan secara otomatis setiap 10 menit sekali dikarenakan setiap 10 menit BMKG akan melakukan pembaruan data kondisi cuaca pada setiap pos hujan yang dimilikinya.

### Perhitungan *K-Nearest Neighbor*



**Gambar 3.8 Diagram Alir K-Nearest Neighbor**

Seperti yang dijelaskan pada bab 2.8 *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap suatu objek berdasarkan data pembelajaran yang memiliki jarak paling dekat dengan objek tersebut. Pada subbab ini akan dijelaskan mekanisme perhitungan *K-Nearest Neighbor* yang digunakan. Sebagai contoh akan dilakukan klasifikasi pos hujan mana yang sesuai dengan koordinat Departemen Teknik Informatika ITS yang memiliki koordinat (-7.279457, 112.797828). Mekanisme perhitungan metode *K-Nearest Neighbor* dijelaskan melalui diagram alir pada Gambar 3.7.

Sesuai pada Gambar 3.7 ada beberapa tahapan untuk melakukan metode *K-Nearest Neighbor* yaitu:

1. Pada tahap pertama yaitu merubah koordinat lokasi rute dan pos hujan menjadi koordinat *Cartesian* dengan cara menggunakan Persamaan 2.2 dan 2.3 yang telah dijelaskan pada bab 2. Sehingga data koordinat Departemen Informatika ITS yang semula ( 112.797828, -7.279457) menjadi (3308.097667, -1021.529765) hasil perhitungannya ditunjukkan pada Tabel 3.5.

**Tabel 3.4 Hasil Perhitungan Koordinat Cartesian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id  pos | Namapos | X | Y | X  Cartersian | Y  *Cartesian* |
| 1 | Kelurahan Asemrowo | 112.715 | -7.252 | 3347.255309 | -1346.0095 |
| 1 | Kelurahan Genting | 112.716646 | -7.24407 | 3388.010771 | -1355.9241 |
| 1 | Kelurahan Greges | 112.684 | -7.2295 | 3411.281442 | -1496.1969 |
| 1 | Kelurahan Kalianak | 112.702 | -7.2289 | 3440.520247 | -1435.7488 |
| 1 | Kelurahan Tambak Langon | 112.664 | -7.225 | 3401.751776 | -1573.8691 |
| 2 | Kantor Kecamatan | 112.608 | -7.235 | 3262.692152 | -1737.4975 |

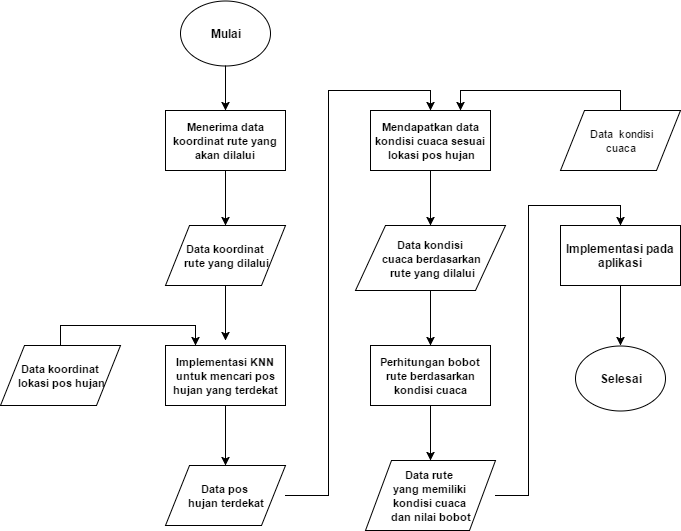
1. Pada tahap kedua akan diambil nilai x dan y yang telah dirubah menjadi koordinat *Cartesian* dari seluruh data pos hujan yang ada.
2. Pada tahap ketiga akan dihitung jarak antara semua koordinat pos hujan dengan koordinat titik rute perjalanan dengan menggunakan Persamaan *Euclidean distance* yang telah dijelaskan pada subbab 2.8.
3. Pada tahap keempat data hasil perhitungan jarak antara satu pos hujan dengan titik rute akan ditampung pada suatu variabel dan kemudian dilakukan perulangan perhitungan jarak sampai semua lokasi pos hujan sudah terhitung.
4. Pada tahap kelima data hasil perhitungan tahap empat diurutkan berdasarkan nilai jarak yang paling kecil. Hasilnya ditunjukkan pada Tabel 3.6.

**Tabel 3.5 Hasil Perhitungan K-Nearest Neighbor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Idpos** | **Idkecamatan** | **Namapos** | **Jarak** |
| 69 | 15 | Kelurahan Kejawan Putih Tambak | 21.5014255 |
| 218 | 38 | PENS | 23.10310988 |
| 191 | 38 | Kampus ITS | 24.23763984 |
| 106 | 23 | Kelurahan Gebang Putih | 48.18287087 |
| 179 | 36 | Mall GalaxyMall | 59.31752884 |
| 70 | 15 | Kelurahan Manyar Sabrangan | 63.08822148 |
| 108 | 23 | Kelurahan Klampis Ngasem | 69.84983801 |

1. Pada tahap ini dilakukan klasifikasi dengan menentukan nilai K=1 hingga hanya diambil urutan pertama dari hasil perhtiungan yang dilakukan pada tahap kelima.
2. Terakhir data hasil perhitungan yang ditampilkan ialah hasil dari perhitungan tahap keenam sehingga didapatkan pos hujan terdekat dari titik koordinat Departemen Teknik Informatika ialah pos hujan pada kelurahan Kejawan Putih Tambak.

### Menentukan Lokasi Pos Hujan Terdekat dan Kondisi Cuaca Pada Rute Perjalanan



**Gambar 3.9 Diagram Alir Menentukan Lokasi Pos Hujan Dan Kondisi Cuaca Pada Rute Perjalanan**

Gambar 3.9 menjelaskan mekanisme dari pengolahan seluruh data dari data rute dan data kondisi cuaca untuk mendapatkan hasil keluaran yang sesuai dengan tujuan pengerjaan Tugas Akhir ini. Mekanisme ini terdiri dari berbagai macam tahap yaitu:

1. Pada tahap pertama sistem akan menerima data berupa GeoJson yang berisi koordinat titik-titik rute yang akan dilalui oleh pengguna.
2. Pada tahap yang kedua akan dilakukan klasifikasi *K-Nearest Neighbor* untuk menentukan lokasi pos hujan mana yang sesuai dengan titik-titik rute perjalanan menggunakan data koordinat yang didapat pada tahap pertama dan data koordinat lokasi pos hujan pada basis data. Proses perhitungan telah dijelaskan pada subbab 3.2.3
3. Pada tahap ketiga dilakukan proses *query* data untuk mendapatkan data kondisi cuaca dengan parameter pos hujan yang didapatkan dari perhitungan sebelumnya.
4. Pada tahap keempat dilakukan proses perhitungan bobot rute tersebut berdasarkan kondisi cuaca yang dimiliki.

**Tabel 3.6 Nilai Bobot Berdasarkan Kondisi Cuaca**

|  |  |  |
| --- | --- | --- |
| **No.** | **Jenis Hujan** | **Bobot** |
| 1 | Sangat lebat | 5 |
| 2 | Lebat | 4 |
| 3 | Sedang | 3 |
| 4 | Ringan | 2 |
| 5 | Sangat Ringan | 1 |

Perhitungan total bobot pada rute menggunakan Persamaan yang ditunjukkan pada Persamaan 3.1:

Keterangan:

* = bobot rute
* = bobot rute ke-
* = bobot hujan ke-
* k = jumlah titik rute yang dilewati

1. Pada tahap ini data rute yang telah memiliki parameter kondisi cuaca dan nilai bobot dikirimkan menuju aplikasi untuk selanjutnya digambarkan pada peta yang terdapat di aplikasi Clearroute.

# BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun sistem. Sebelum masuk ke penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.



## Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak.

### Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam membangun sistem dalam Tugas Akhir ini adalah sebagai berikut:

Jenis : Komputer *Desktop*

*Processor* : Intel(R) Core(TM) i3-2120 CPU @ 3.30GHz

Memori : 4 GB 1600 MHz

Sistem Operasi : Ubuntu Xenial 16.04

### Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun sistem pada Tugas Akhir ini adalah sebagai berikut:

* Sublime Text 3 digunakan untuk menyunting kode sumber.
* PgAdmin IV digunakan untuk antarmuka pengelola basis data.
* Apache digunakan untuk *web server*.

## Implementasi Perangkat Lunak

Pada bagian ini akan dijelaskan mengenai implementasi proses yang telah dijabarkan pada perancangan modul di bab III ke dalam sebuah kode pemrograman.

### Implementasi Proses Mendapatkan Data Ramalan

Seperti yang telah dijelaskan pada bab perancangan sistem, bahwa hal pertama yang dilakukan untuk mendapatkan data ramalan cuaca ialah melakukan pengunduhan data dari *server* BMKG. Data yang diunduh haruslah data yang baru dikeluarkan BMKG sesuai dengan tanggal saat data tersebut diunduh. Pada kode Sumber 4.1 dapat dilihat sturuktur kode yang dibangun. Pada baris ketiga dilakukan sebuah *query* untuk mendapatkan semua id dan lokasi x,y koordinat lokasi pos hujan. Kemudian dilakukan pengunduhan data dari *server* BMKG. Judul dari data yang akan diunduh memiliki suatu pola yang sama yaitu terdiri dari tanggal data tersebut dikeluarkan dan waktu berlaku dari data yang dikeluarkan. Untuk itu pada baris ke 4 sampai ke 11 dilakukan deklarasi variabel untuk mendapatkan waktu saat ini. Dan kemudian pada baris 14 dilakukan pengunduhan data menggunakan fungsi bawaan dari *library* PHP yaitu imagecreatefromgif() yang berfungsi untuk mendapatkan data citra dan segala atribut yang ada di dalamnya.

1. **public** **function** pengumpulandata(){
2. //get *pixel* coordinates
3. $query=DB::table('datapos')->select('idpos','xramalan','yramalan','namapos')->get();
5. $tahun=date('Y');
6. $bulan= strtoupper(date('M'));
7. $hari=11;
8. $harisaatini=date('d');
9. $haha=date('m');
10. $tanggal=$tahun.$haha;
11. $tanggal2=$tahun.$haha.$hari;
12. $number=0;
13. //Download forecast data
14. $gambar=imagecreatefromgif('http://diseminasi.meteo.bmkg.go.id/wrf/indo\_'
15. .$tanggal.'/'.$tanggal2.'0000/jawa/HUJAN.03-JAM.'.$tanggal2.'0000/prec03-'.$tahun.''.$bulan.'
16. '.$hari.''.'21'.'.gif');

**Kode Sumber 4.1 Proses Mendapatkan Data Ramalan**

Selanjutnya ialah dilakukan proses pengambilan RGB dari data yang telah diunduh. Lokasi *pixel* yang kode warna RGB-nya diambil merupakan lokasi pos hujan yang telah terdapat pada basis data. Proses ini dilakukan secara berulang sebanyak jumlah data pos hujan yang ada.

Pada Kode Sumber 4.2 ditunjukkan struktur kode yang dibuat. Pada baris pertama dilakukan perulangan sebanyak data yang didapatkan dari hasil *query* lokasi pos hujan yang terdapat pada basis data. Kemudian pada baris ke 9 dilakukan pengambilan kode RGB dari gambar yang digunakan dengan memakai fungsi *imagecolorat()* yang merupakan bagian dari *library* bawaan yang terdapat pada PHP keluaran dari fungsi tersebut menghasilkan kode RGB warna dari gambar yang kemudian kode tersebut dipecah kembali untuk mendapatkan nilai *hexadecimal*-nya yang ditunjukkan pada baris 12-15 pada kode sumber 4.2.

**Kode Sumber 4.2 Mendapatkan RGB Dari Gambar**

1. **foreach** ($query **as** $key => $value) {
2. $xx = 4;
3. $jam1 = sprintf("%02.2d", $xx);
5. **if** ($value->xramalan != null) {
6. //set valid date
7. $valid = date('Y-m-d');
8. //get RGB
9. $RGB = imagecolorat($gambar, $value->xramalan, $value->yramalan);
10. $cols = imagecolorsforindex($gambar, $RGB);
11. //clasifying RGB code
12. $r = dechex($cols['red']);
13. $g = dechex($cols['green']);
14. $b = dechex($cols['blue']);
15. $warna = "#" . $r . $g . $b;

Masih dalam proses yang sama setelah mendapatkan kode RGB dilakukan klasifikasi jenis cuaca berdasarkan RGB tersebut dengan menggunakan perintah *if statement* yang kemudian data yang telah didapatkan langsung disimpan pada basis data. Kode Sumber 4.3 menunjukan keseluruhan dari struktur kode yang dibuat. Pada baris pertama ditunjukkan *if statement* berdasarkan jenis RGB yang didapatkan kemudian dilakukan proses *insert* ke dalam basis data yang ditunjukkan pada baris 6 sampai 10 dari Kode Sumber 4.3. Proses *insert* basis data sendiri menggunakan *syntax* dari Laravel sendiri.

1. **if** ($warna == '#b4faaa' || $warna == '#96f58c' || $warna == '#78f573'
2. || $warna == '#50f050' || $warna == '#37d23c'
3. ) {
4. echo "Berawan";
5. DB::table('rekaman')->insert([
6. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
7. 'idpos' => $value->idpos,
8. 'kategoriramalan' => 1,
9. 'validtime' => '04:00:00',
10. 'validdate' => $valid,
11. ]);
12. } **elseif** ($warna == '#0fa00f' || $warna == '#fa0f') {
13. echo "Hujan Ringan";
14. DB::table('rekaman')->insert([
15. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
16. 'idpos' => $value->idpos,
17. 'kategoriramalan' => 2,
18. 'validtime' => '04:00:00',
19. 'validdate' => $valid,
20. ]);
21. } **elseif** ($warna == '#e6dc32') {
22. echo "Hujan Sedang";
23. DB::table('rekaman')->insert([
24. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
25. 'idpos' => $value->idpos,
26. 'kategoriramalan' => 3,
27. 'validtime' => '04:00:00',
28. 'validdate' => $valid,
29. ]);
30. } **elseif** ($warna == '#f08228') {
31. echo "Hujan deras";
32. DB::table('rekaman')->insert([
33. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
34. 'idpos' => $value->idpos,
35. 'kategoriramalan' => 4,
36. 'validtime' => '04:00:00',
37. 'validdate' => $valid,
38. ]);
39. } **elseif** ($warna == '#fa3c3c') {
40. echo "Hujan sangat deras";
41. DB::table('rekaman')->insert([
42. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
43. 'idpos' => $value->idpos,
44. 'kategoriramalan' => 5,
45. 'validtime' => '04:00:00',
46. 'validdate' => $valid,
47. ]);
48. } **elseif** ($warna == '#ffffff') {
49. DB::table('rekaman')->insert([
50. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
51. 'idpos' => $value->idpos,
52. 'kategoriramalan' => 0,
53. 'validtime' => '04:00:00',
54. 'validdate' => $valid,
55. ]);
56. echo "Cerah";
57. } **else** {
58. DB::table('rekaman')->insert([
59. 'idrekaman' => $tahun . $bulan . $harisaatini . $jam1,
60. 'idpos' => $value->idpos,
61. 'kategoriramalan' => 6,
62. 'validtime' => '04:00:00',
63. 'validdate' => $valid,
64. ]);
65. }
66. echo "<br>";
67. echo $warna;
68. echo "<br>";
69. echo "<br>";
71. }

**Kode Sumber 4.3 Klasifikasi Jenis Cuaca**

### Implementasi Proses Mendapatkan Data Kondisi Cuaca Terkini

Proses ini dilakukan untuk mendapatkan data *real time* kondisi cuaca yang sedang terjadi. Data tersebut didapatkan dari hasil rekaman pos hujan yang dimiliki oleh BMKG. Data dari pos hujan tesebut kemudian diolah oleh BMKG yang selanjutnya dikirimkan kepada sistem Clearroute menggunakan metode pengiriman *Http Request* dalam bentuk XML. Kemudian data tersebut akan diolah kembali oleh sistem Clearroute agar didapatkan kategori cuaca yang sedang terjadi. Setelah itu data kemudian disimpan ke dalam basis data Clearroute.

Pada Kode Sumber 4.4 ditunjukkan struktur kode yang dibangun. Pertama pada baris ketiga dari Kode Sumber 4.4 dilakukan proses pengunduhan data XML menggunakan fungsi *file\_get\_contents()* yang kemudian disimpan pada variabel $xml selanjutnya data XML yang didapat dipecah berdasarkan *tag* yang terdapat didalam variabel XML dengan cara melakukan deklarasi fungsi *SimpleXMLElement()* yang disimpan pada variabel $data selanjutnya dilakukan proses perulangan sebanyak *tag* yang terdapat pada data XML dan kemudian diambil data dari *tag* yang bernama rekaman yang selanjutnya data tersebut akan disimpan ke dalam basis data menggunakan *syntax* yang tersedia pada kerangka kerja Laravel.

1. **public** **function** handle()
2. {
3. $xml=*file*\_get\_contents("bmkgjuanda.ip-dynamic.com:81/sby/rekaman\_tenminute.xml");
4. // $data->loadHTML($xml);
6. $data=**new** SimpleXMLElement($xml);
7. **foreach** ($data->rekaman **as** $key ) {
8. **if**($key->kategori >=1 && $key->kategori <=2){
9. DB::table('datapos')->where('idpos',$key->idpos)->update(['kategori'=>1]); //berawan
10. }
12. **if**($key->kategori >=3 && $key->kategori <=8){
13. DB::table('datapos')->where('idpos',$key->idpos)->update(['kategori'=>2]);//hujan ringan
14. }
16. **if**($key->kategori >=9 && $key->kategori <=11){
17. DB::table('datapos')->where('idpos',$key->idpos)->update(['kategori'=>3]); //hujan sedang
18. }
19. **if**($key->kategori >=12 && $key->kategori <=13){
20. DB::table('datapos')->where('idpos',$key->idpos)->update(['kategori'=>4]); //deras
21. }
22. **if**($key->kategori >13 ){
23. DB::table('datapos')->where('idpos',$key->idpos)->update(['kategori'=>5]); //sangat deras
24. }
25. **if**($key->kategori ==1){
26. DB::table('datapos')->where('idpos',$key->idpos)->update(['kategori'=>0]); //cerah
27. }

30. }
32. }

**Kode Sumber 4.4 Mendapatkan Kondisi Cuaca Terkini**

### Implementasi Proses Menentukan Lokasi Pos Hujan Terdekat dan Kondisi Cuaca Pada Rute Perjalanan

Proses ini dilakukan untuk menentukan kondisi cuaca berdasarkan titik rute perjalanan yang diterima oleh fungsi yang kemudian hasil pengolahan dari proses ini akan dikirimkan ke aplikasi Clearroute.

Langkah pertama yang dilakukan ialah dengan menerima data rute perjalanan dari modul penentuan perjalanan dan kemudian data yang telah diterima dalam bentuk Json diambil koordinat rute perjalanannya. Selanjutnya data koordinat yang telah diambil dikirimkan ke fungsi perhitungan *K-Nearest Neighbor* untuk didapatkan kondisi cuaca dari pos hujan yang terdekat dengan rute perjalanan. Setelah itu langsung dilakukan perhitungan bobot berdasarkan kondisi cuaca yang didapat.

Kode Sumber 4.5 akan memperlihatkan struktur kode dari proses ini hal pertama yang ditunjukkan pada Kode Sumber 4.5 ialah mendapatkan data pos hujan dan data cuaca dengan menggunakan metode *inner join* pada *query* yang digunakan proses *inner* *join* tersebut menggabungkan tabel datapos dan tabel rekaman cuaca. Selanjutnya dilakukan deklarasi variabel untuk proses pehitungan bobot rute. Setelah itu dilakukan perulangan sebanyak jumlah data titik rute perjalanan yang didapat. Perulangan ini dimaksudkan untuk mendapatkan titik koordinat dari rute perjalanan yang akan dilalui. Data koordinat yang didapat akan ditampung pada variabel $x dan $y yang kemudian variabel tersebut digunakan sebagai parameter untuk memanggil fungsi *K-Nearest Neighbor*. Hasil keluaran dari fungsi *K-Nearest Neighbor* yang dibuat langsung ditampung dalam sebuah *array* yang disebut $output.

Selanjutnya di dalam *array* tersebut terdapat nilai dari kondisi cuaca yang kemudian nilai tersebut ditotal untuk menghitung bobot yang ada. Perhitungan bobot tersebut disimpan ke dalam variabel yang bernama $finalcurr untuk kondisi cuaca terkini dan $finalfrcst untuk kondisi ramalan cuaca. Hasil dari perhitungan bobot tersebut kembali dimasukan ke dalam *array* $output yang selanjutnya variabel $output akan menjadi *return value*.

1. **public** function cuaca($waktu, $x1)
2. {
3. $datenow = date('Y-m-d ');
4. //get pos hujan data
5. $query = DB::select("select \* from datapos inner join rekaman on datapos.idpos = rekaman.idpos
6. where rekaman.validtime > '" . $waktu . "' and validdate = '" . $datenow . "'  order by validtime limit 208");
7. $count = 0;
8. $weightcur = 0;
9. $weightfrcst = 0;
10. $jsonrute = $x1;
11. $output = array();
12. //read all json data
13. foreach ($jsonrute as $key => $value) {
14. $datakoor = json\_decode($value->json);
15. //get coordinates from GeoJson
16. $x = $datakoor->geometry->coordinates[0][0];
17. $y = $datakoor->geometry->coordinates[0][1];
19. //call KNN
20. array\_push($output, $**this**->knn($query, $x, $y));
21. //weight count
22. $weightcur += $output[$count]['kategori'];
23. $weightfrcst += $output[$count]['ramalan'];
24. $count++;
25. }
26. $finalcurr = $weightcur / $count;
27. $finalfrcst = $weightfrcst / $count;
28. $bobot = array('weightcurr' => $finalcurr, 'weightfrcst' => $finalfrcst);
29. array\_push($output, $bobot);
30. **return** $output;

**Kode Sumber 4.5 Menentukan Lokasi Pos Hujan Terdekat dan Kondisi Cuaca**

### Implementasi Proses Perhitungan *K-Nearest Neighbor*

Proses ini dilakukan ketika menerima data titik rute perjalanan yang akan dilalui. Pada Tugas Akhir ini digunakan Persamaan yang telah dijelaskan pada subbab 2.8 yaitu menggunakan persamaan *Euclidean distance* dan persamaan untuk mengubah koordinat *latitude longitude* menjadi koordinat *Cartesian*. Parameter input yang digunakan pada fungsi ini ialah data kooordinat seluruh pos hujan dan data koordinat rute perjalanan. Kode sumber 4.6 akan menunjukan implementasi proses ini dalam sturuktur kode. Hal pertama yang dilakukan ialah menerima isi dari parameter yang diberikan untuk selanjutnya dilakukan perulangan sebanyak isi dari paramter yang diberikan. Kemudian dilakukan proses perhitungan yang disimpan pada variabel *array* $knn perhitungan *K-Nearest Neighbor* ini dilakukan sebanyak jumlah data pos hujan yang tersedia. Kemudian setelah melakukan perhitungan dan seluruh hasilnya disimpan pada variabel $knn dilakukan proses pengurutan data berdasarkan nilai perhitungan jarak paling kecil dengan menggunakan fungsi *sort*(). Kemudan dipilih nilai *array* pertama untuk diambil dan dikirim kembali sebagai *return value*.

1. **public** function knn($query, $x, $y)
2. {
3. $i = 0;
5. foreach ($query as $key => $hasil) {
6. //change to *Cartesian* and count *Euclidean* distance
7. $knn[$i] = array('value' => sqrt(pow((6371 \* cos($hasil->ydesimal) \* cos($hasil->xdesimal))
8. - (6371 \* cos($y) \* cos($x)), 2) + pow((6371 \* cos($hasil->ydesimal) \* sin($hasil->xdesimal))
9. - (6371 \* cos($y) \* sin($x)), 2)), "idpos" => $hasil->idpos, "namapos" => $hasil->namapos,
10. 'x' => $x, 'y' => $y, 'kategori' => $hasil->kategori, 'ramalan' => $hasil->kategoriramalan);
11. $i++;
12. }
13. //sort array ascending
14. sort($knn);
15. //retrurn first array K=1
16. **return** $knn[0];
18. }

**Kode Sumber 4.6 K-Nearest Neighbor**

Hasil dari kode sumber 4.6 ialah data cuaca berdasarkan pos hujan terdekat yang telah diklasifikasi menggunkan *K-Nearest Neighbor* berdasarkan titik koordinat rute perjalanan.

# BAB V UJI COBA DAN EVALUASI

Bab ini akan menjelaskan mengenai skenario uji coba pada perangkat lunak yang telah dibangun. Setelah itu, hasil uji coba akan dievaluasi kinerjanya sehingga dapat diputuskan apakah perangkat lunak ini mampu menyelesaikan permasalahan yang telah dirumuskan diawal. Secara garis besar, bab ini berisikan pembahasan mengenai lingkungan pengujian,dan uji kinerja.



## Lingkungan Pengujian

Lingkungan pengujian pada uji coba Tugas Akhir ini menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Perangkat Keras Uji Coba**

|  |  |  |
| --- | --- | --- |
| **Perangkat** | **Jenis Perangkat** | **Spesifikasi** |
| **Perangkat Keras** | Prosesor | Intel(R) Core(TM) i3-2120 CPU @ 3.30GHz |
| Memori | 4 GB 1600 |
| **Perangkat Lunak** | Sistem Operasi | Ubuntu Xenial 16.04 LTS |
| Perangkat Pengembang | Apache *web server* |

## Skenario Uji coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario uji coba ini sistem akan diuji apakah sudah berjalan dengan benar, dan bagaimana hasil performa pada masing-masing skenario uji coba. Pada Tugas Akhir ini, terdapat beberapa jenis skenario uji coba, yaitu:

1. Perhitungan *response time* pengiriman data json hasil penetuan rute dan penentuan kondisi cuaca ke aplikasi Clearroute.
2. Perbandingan hasil metode *K-Nearest Neighbor* dengan variasi nilai K.
3. Mendapatkan data cuaca BMKG secara *real time* dalam interval waktu yang telah ditentukan untuk melihat kesesuaian data yang diterima dengan data dari *server* BMKG.
4. Menentukan beberapa titik uji dan melihat hasil rekomendasi rute berdasarkan nilai bobot yang dimiliki.
5. Perhitungan akurasi pengambilan data cuaca dari citra ramalan cuaca BMKG antara menggunakan *script* dengan penglihatan manual.

### Skenario Uji Coba 1

Skenario uji coba pertama adalah perhitungan *response time* pengiriman data API JSON hasil penentuan rute dan penentuan kondisi cuaca ke aplikasi Clearroute. Untuk menghitung *response time* penulis menggunakan apli1kasi berbasis *web* dengan alamat <http://loader.io>. Aplikasi tersebut dapat melakukan perhitungan *response time* berdasarkan kriteria yang telah ditentukan.

Pada uji coba perhitungan *response time* kali ini dibuat kriteria pengujian mulai dari jarak perjalanan dan jumlah pengguna yang mengakses. Kriteria pengujian skenario uji coba satu dibuat sebagai berikut:

1. Rute Perjalanan
   1. Departemen Informatika - Carls’jr Kertajaya (3.4 KM).
   2. Departemen Informatika - Tunjungan Plasa (8.5 KM).
   3. Departemen Informatika - Kampus Unesa (17.7 KM).
2. Jumlah Pengguna
3. 15 pengguna.
4. 25 pengguna.
5. 35 pengguna.

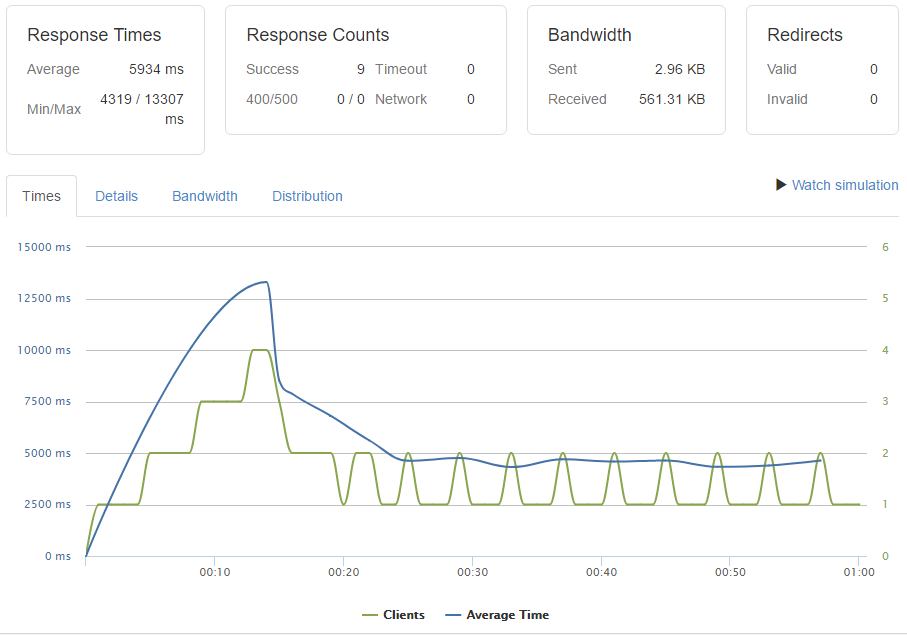
Uji coba ini dijalankan selama 1 menit dan dalam waktu tersebut terdapat jumlah pengguna yang telah ditentukan pada kriteria melakukan *request* data secara bersamaan. Hasil uji coba skenario 1 ini dapat dilihat pada Tabel 5.2, Tabel 5.3 dan Tabel 5.4.

**Tabel 5.2 Response Time 15 Pengguna**

|  |  |  |  |
| --- | --- | --- | --- |
| Rute | Waktu minimum | Waktu Maksimum | Rata-Rata |
| A | 4319 ms | 13307 ms | 5934 ms |
| B | 5304 ms | 9354 ms | 5790 ms |
| C | 5609 ms | 7632 ms | 6305 ms |

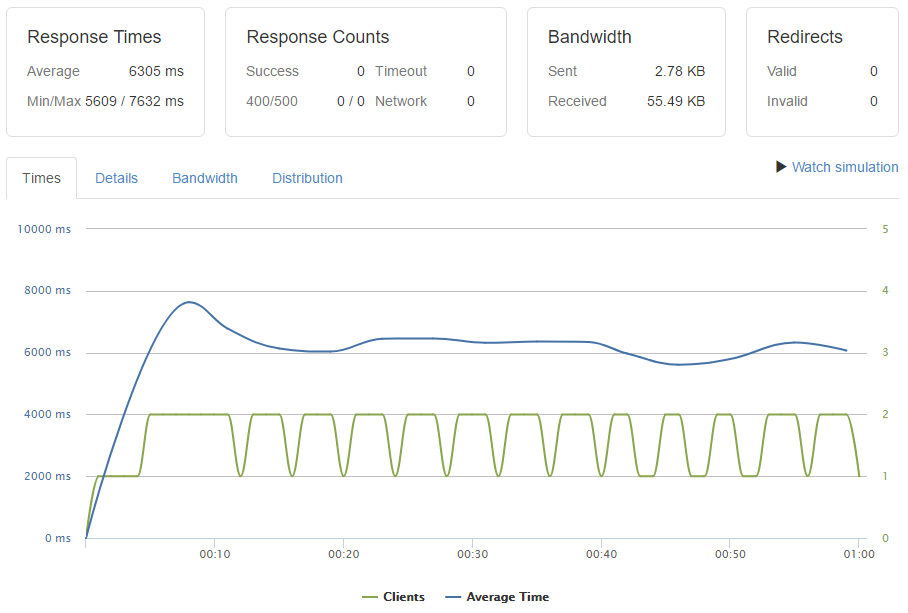
**Tabel 5.3 Response Time 25 Pengguna**

|  |  |  |  |
| --- | --- | --- | --- |
| Rute | Waktu minimum | Waktu Maksimum | Rata-Rata |
| A | 5967 ms | 13062 ms | 8186 ms |
| B | 6509 ms | 18989 ms | 11179 ms |
| C | 6524 ms | 15256 ms | 10724 ms |

**Tabel 5.4 Response Time 35 Pengguna**

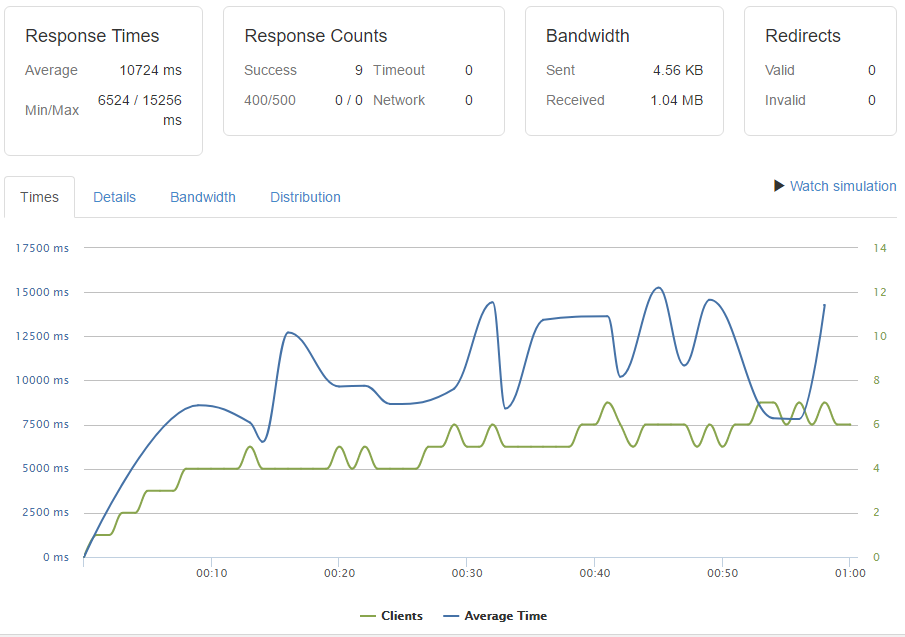
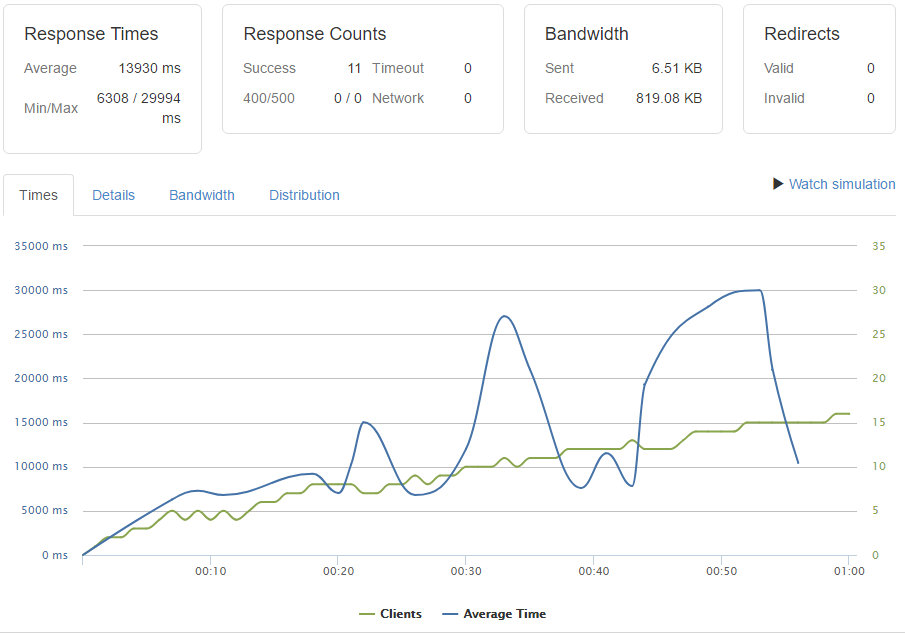
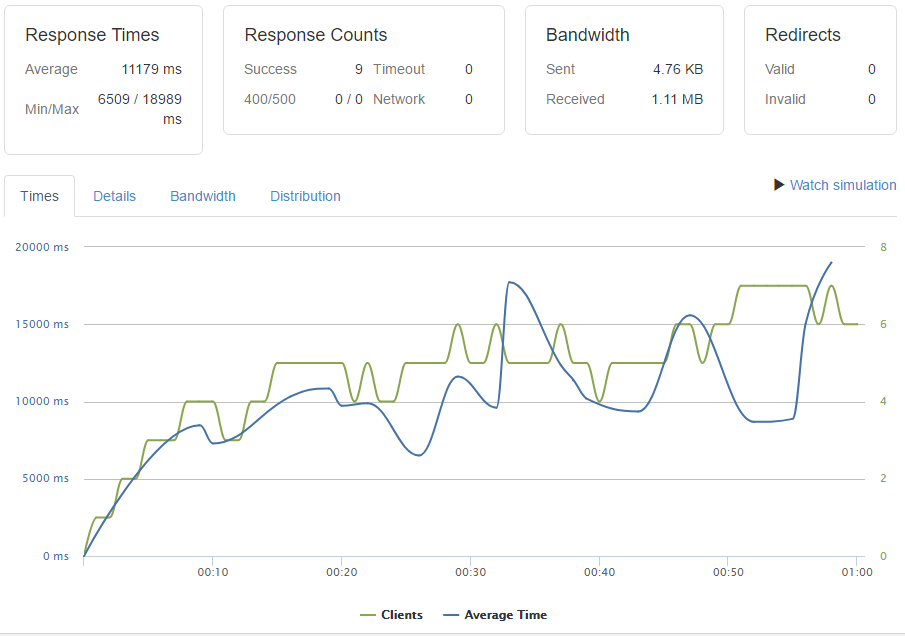
**Gambar 5.1** **Hasil Uji Coba Response Time 15 Pengguna Rute A**

|  |  |  |  |
| --- | --- | --- | --- |
| Rute | Waktu minimum | Waktu Maksimum | Rata-Rata |
| A | 6308 ms | 29994 ms | 13930 ms |
| B | 8299 ms | 31284 ms | 16835 ms |
| C | 9288 ms | 39123 ms | 19355 ms |



**Gambar 5.2 Hasil Uji Coba Response Time 15 Pengguna Rute B**

**Gambar 5.3 Hasil Uji Coba Response Time 15 Pengguna Rute C**

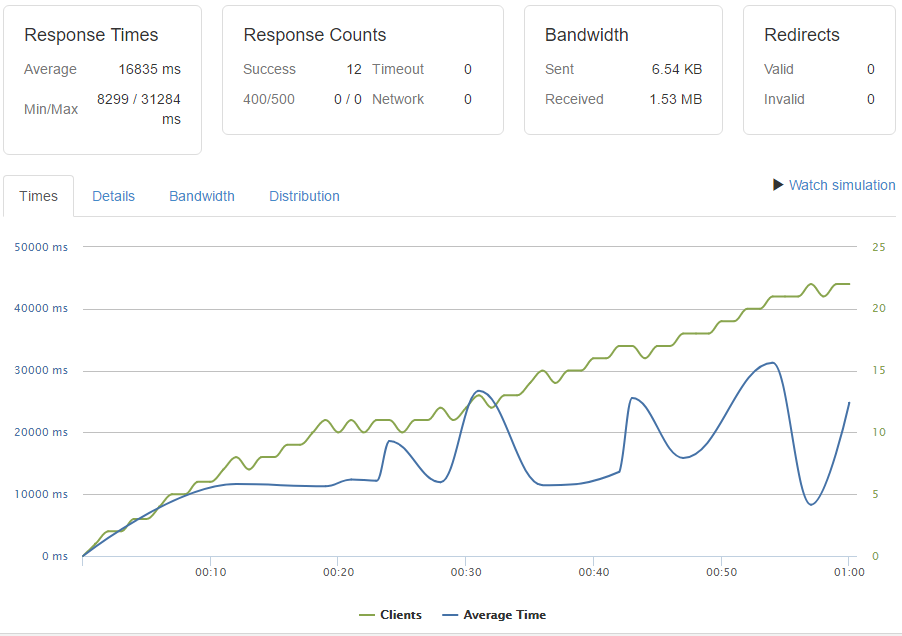
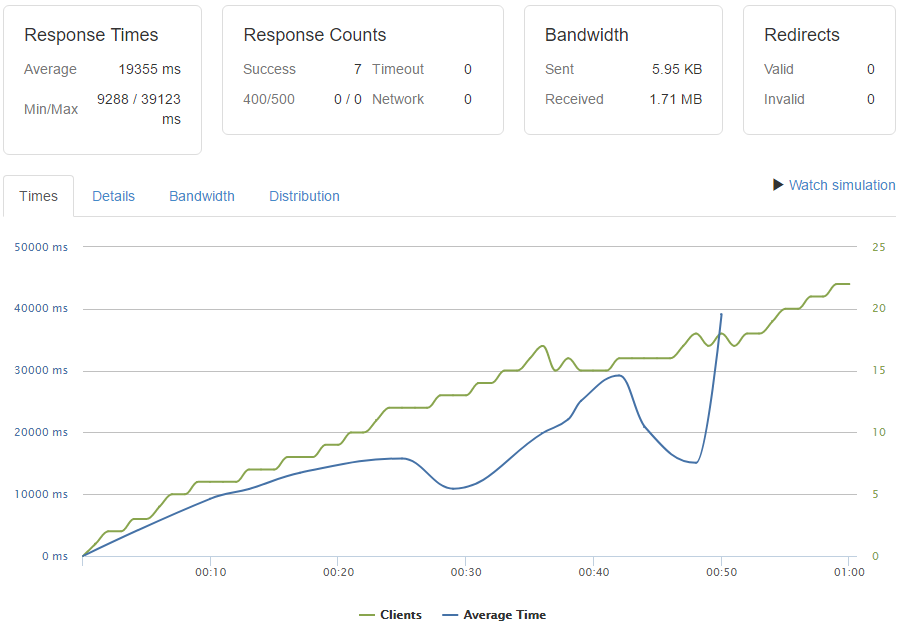


**Gambar 5.4 Hasil Uji Coba Response Time 25 Pengguna Rute B**

**Gambar 5.5 Hasil Uji Coba Response Time 25 Pengguna Rute A**

**Gambar 5.6 Hasil Uji Coba Response Time 35 Pengguna Rute A**

**Gambar 5.7 Hasil Uji Coba Response Time 25 Pengguna Rute C**



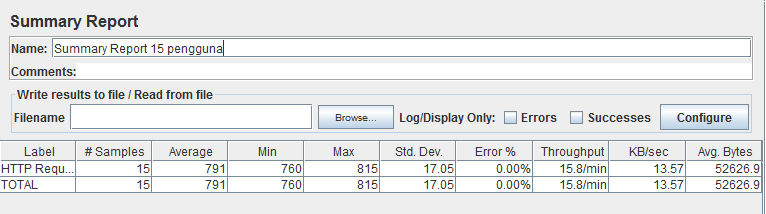
**Gambar 5.8 Hasil Uji Coba Response Time 35 Pengguna Rute B**

**Gambar 5.9 Hasil Uji Coba Response Time 35 Pengguna Rute B**

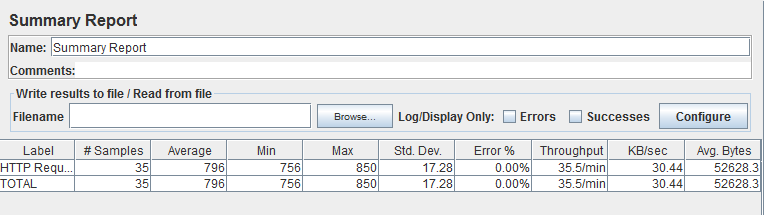
Kemudian dilakukan kembali pengujian *Response Time* perhitungan *K-Nearest Neighbor* menggunakan titik rute acak tanpa melakukan pencarian rute perjalanan pada basis data peta dan diakses secara bersamaan sebanyak 15, 25 dan 35 pengguna. Guna mengetahui kecepatan algoritma sendiri dalam melakukan perhitungan hasil dari pengujian ini dapat dilihat pada Tabel 5.7, Gambar 5.19, Gambar 5.20 dan gambar 5.21

**Tabel 5.5 Uji Coba Response Time Perhitungan K-Nearest Neighbor**

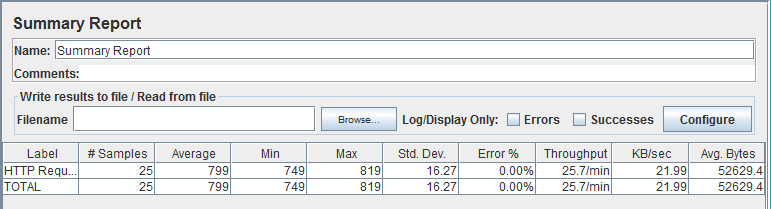
|  |  |  |  |
| --- | --- | --- | --- |
| Pengguna | Waktu minimum | Waktu Maksimum | Rata-Rata |
| 15 | 760 ms | 815 ms | 791 ms |
| 25 | 749 ms | 819 ms | 799 ms |
| 35 | 756 ms | 850 ms | 796 ms |



**Gambar 5.10 Hasil Uji Coba Perhitungan Response Time K-Nearest Neighbor 15 Pengguna**



**Gambar 5.11 Hasil Uji Coba Perhitungan Response Time K-Nearest Neighbor 35 Pengguna**



**Gambar 5.12 Hasil Uji Coba Perhitungan Response Time K-Nearest Neighbor 25 Pengguna**

### Skenario Uji Coba 2

Skenario uji coba kedua ialah menguji *output* yang didapat dari algoritma *K-Nearest Neighbor* dengan berbagai macam variasi nilai K yang digunakan sebagai masukan yang kemudian akan ditentukan nilai K berapa yang memiliki akurasi klasifikasi cuaca paling tinggi. Variasi nilai K yang akan digunakan ialah K=1 , K=2, K=3 dan sampel data yang digunakan ialah titik Departemen Informatika ITS dengan koordinat *latitude*: -7.279586 dan *longitude*:112.797836. Hasil uji coba skenario 2 dapat dilihat sebagai berikut:

Pada uji coba K=1 didapatkan titik sebagai berikut:

**Tabel 5.6 Hasil K=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama pos | *Latitude* | *Longitude* | Jarak | Kategori Cuaca |
| Kelurahan Kejawan putih tambak | -7.276 | 112.801 | 528 m | 2 |

Pada uji coba K=2 didapatkan titik-titik sebagai berikut:

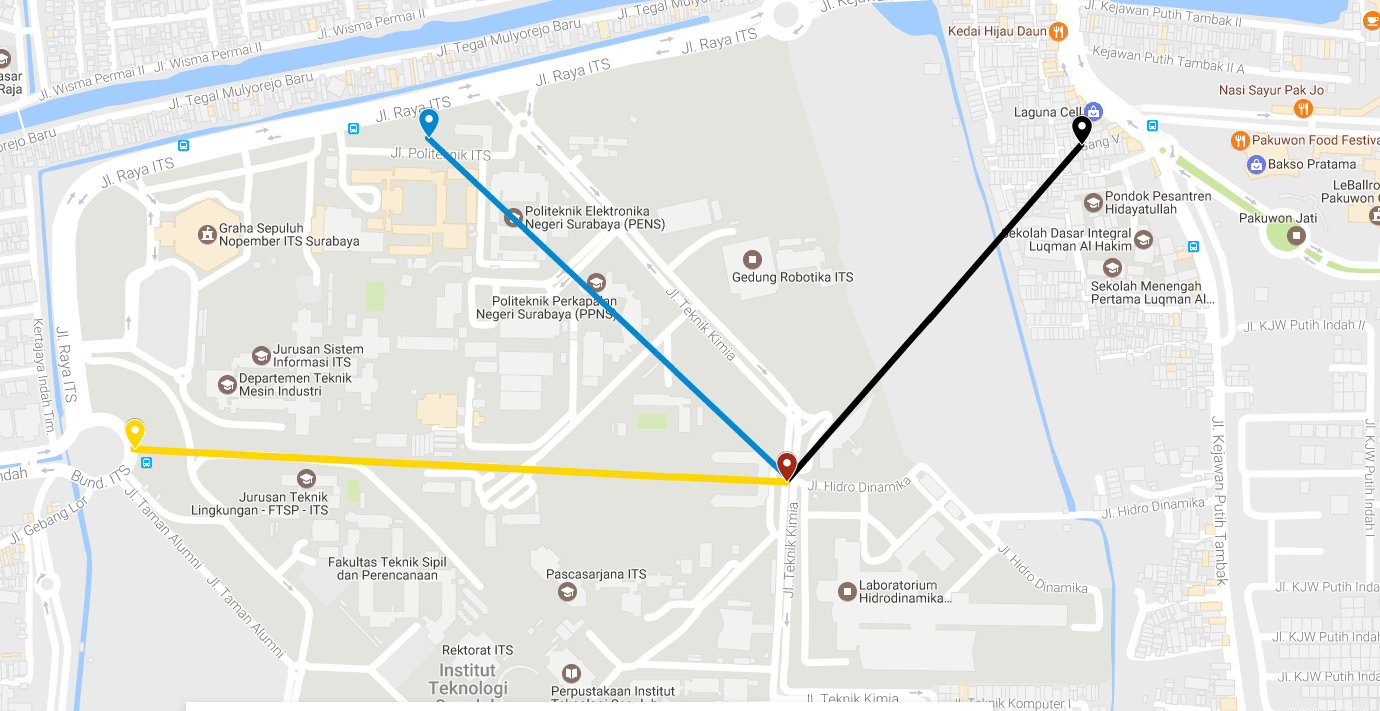
**Tabel 5.7 Hasil K=2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama pos | *Latitude* | *Longitude* | Jarak | Kategori Cuaca |
| Kelurahan Kejawan putih tambak | -7.276 | 112.801 | 528 m | 2 |
| PENS | -7.275923 | 112.793992 | 591 m | 1 |

Pada uji coba K=3 didapatkan titik-titik sebagai berikut:

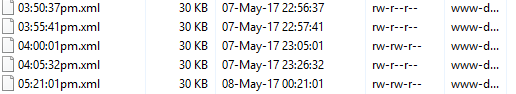
**Tabel 5.8 Hasil K=3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama pos | *Latitude* | *Longitude* | jarak | Kategori Cuaca |
| Kelurahan Kejawan putih tambak | -7.276 | 112.801 | 528 m | 2 |
| PENS | -7.275923 | 112.793992 | 591 m | 1 |
| Kampus ITS | -7.279235 | 112.790837 | 776.62 m | 0 |



**Gambar 5.13 Lokasi Pos Hujan Berdasarkan Perhitungan KNN K=3**

### Skenario Uji Coba 3

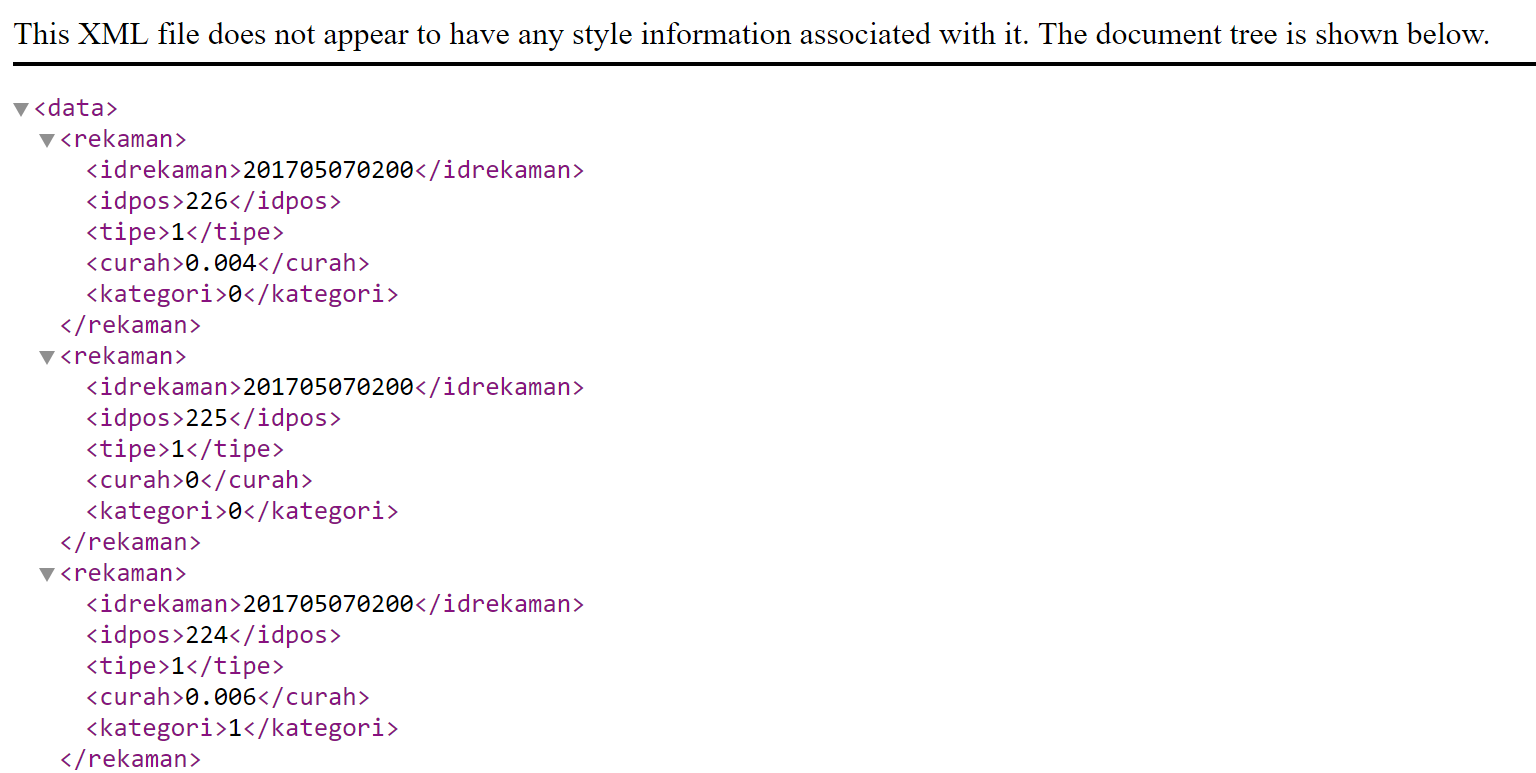
Pada skenario uji coba ketiga akan dilakukan uji coba mendapatkan data cuaca dari *server* BMKG secara *realtime*. Pada uji coba ini akan menguji *script* untuk mengunduh data dari BMKG secara otomatis apakah berhasil atau gagal. Parameter berhasil atau gagal ditentukan dari data yang telah diunduh apakah data tersebut rusak atau tidak. Pada uji coba kali ini *script* yang telah dibuat diatur untuk melakukan pengunduhan data secara otomatis setiap interval 5 menit sekali. Hasil uji coba dapat dilihat pada Gambar 5.23, Gambar 5.24 dan Tabel 5.11

**Gambar 5.14 Data BMKG yang Telah Diunduh**

**Tabel 5.9 Daftar Ukuran file XML**

**Gambar 5.15 DataXML03:50:37pm.xml**

|  |  |
| --- | --- |
| Nama *file* | Ukuran |
| 03:50:37pm.xml | 30 KB |
| 03:55:41pm.xml | 30 KB |
| 04:00:01pm.xml | 30 KB |
| 04:05:32pm.xml | 30 KB |

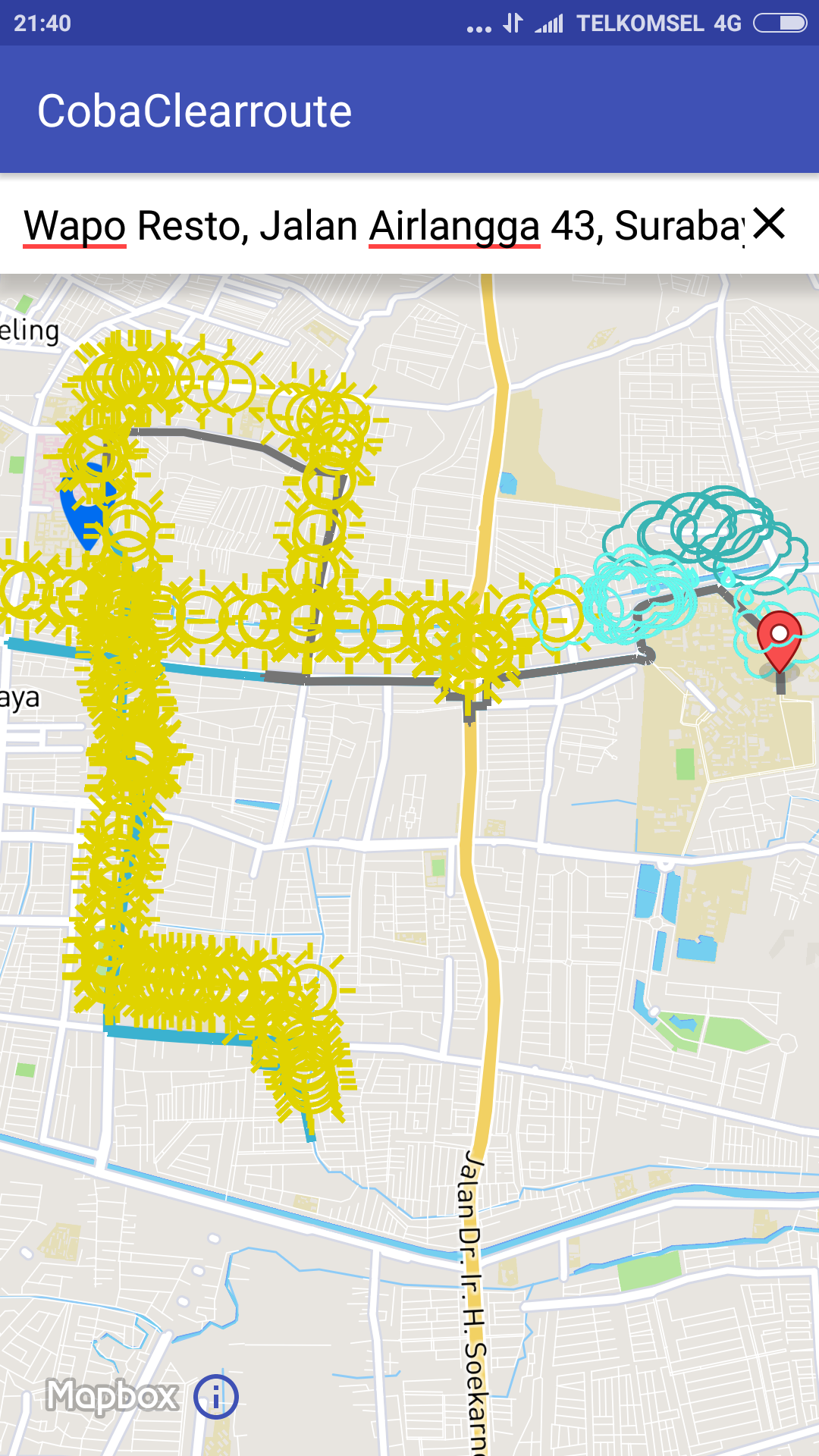
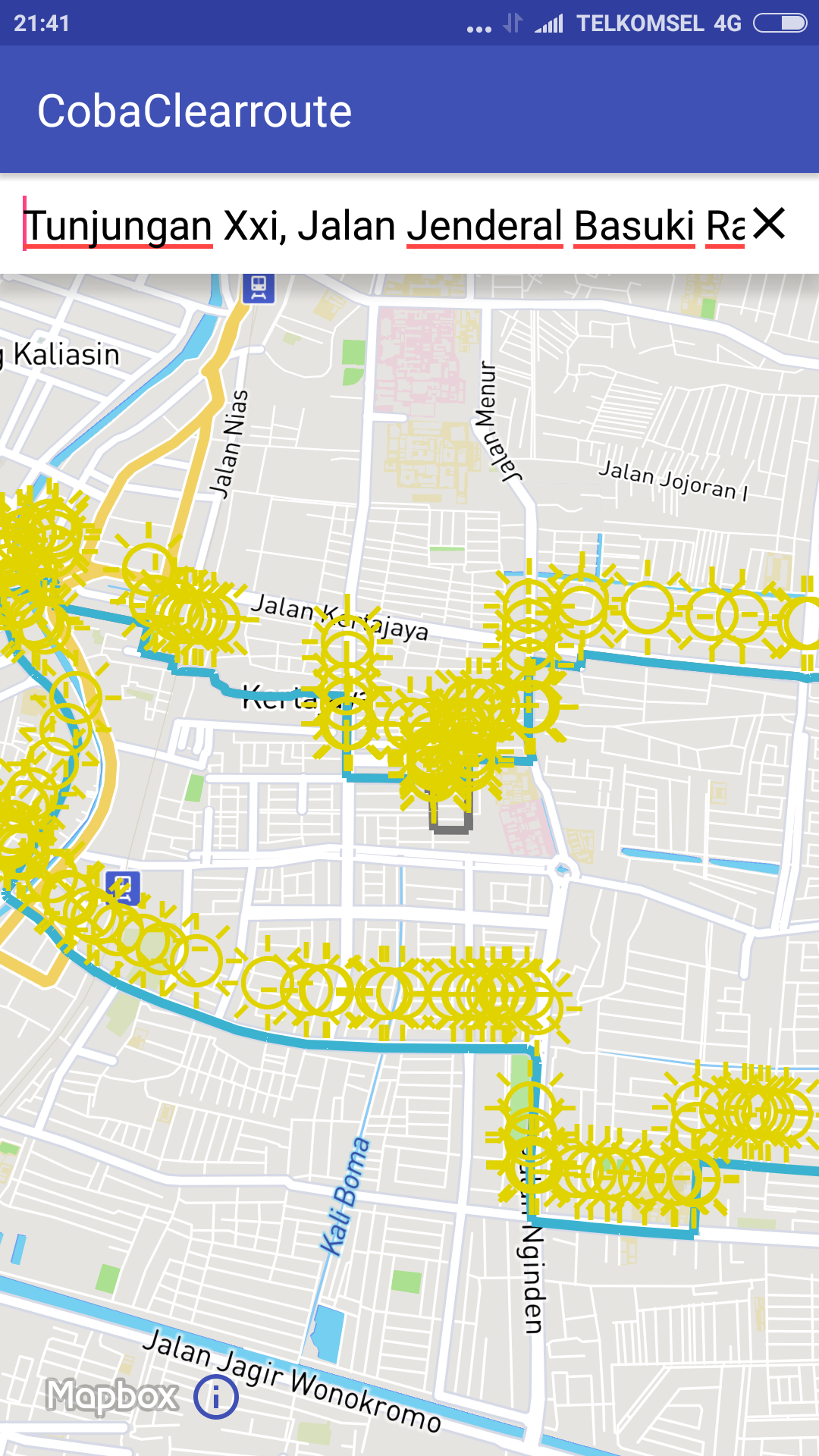


### Skenario Uji Coba 4

Pada Skenario Uji Coba keempat akan dilakukan pengujian rekomendasi titik rute berdasarkan bobot yang diberikan. Pada uji coba ini akan dicoba tiga jenis rute yaitu:

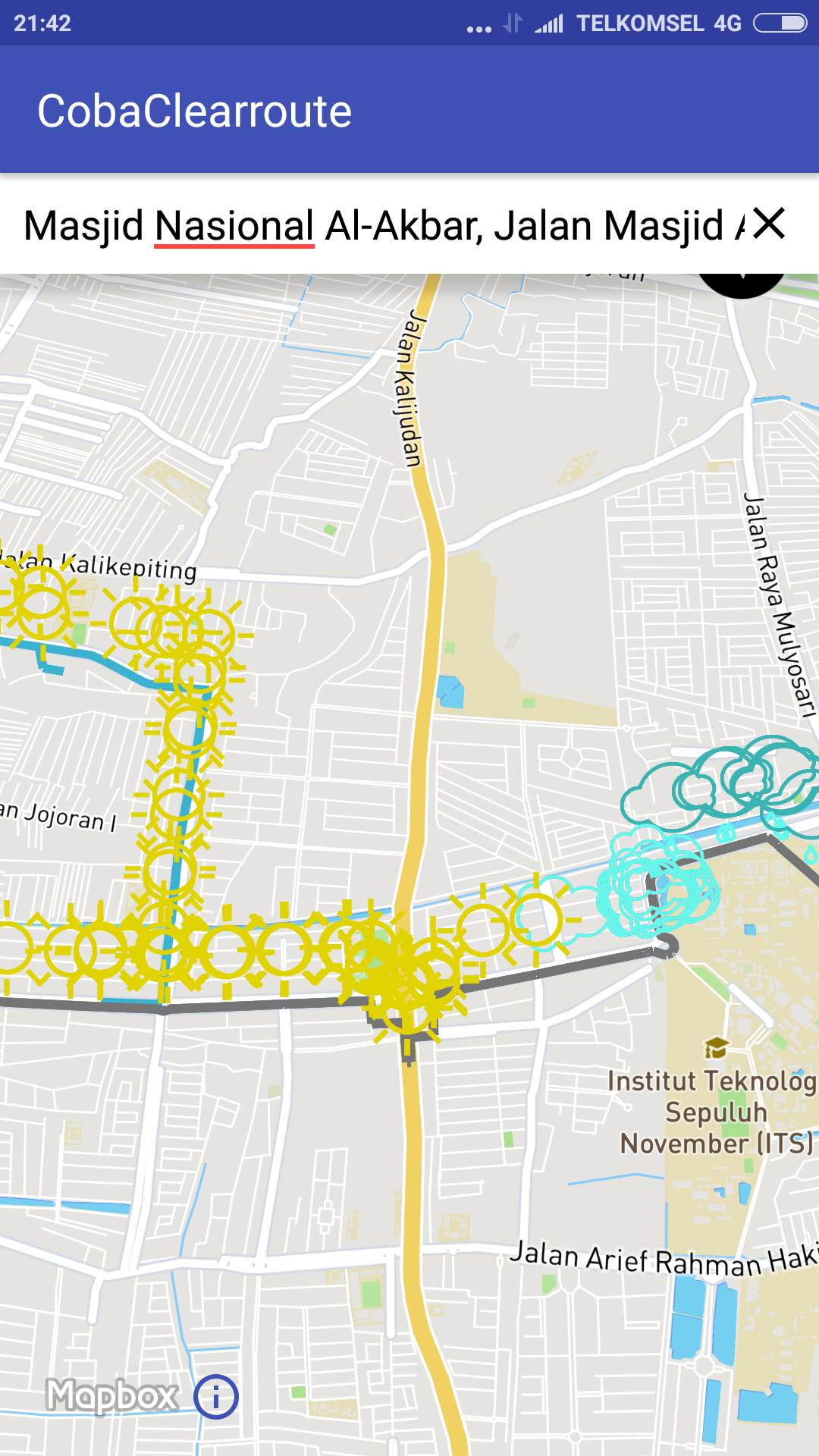
1. Departemen Informatika – Restoran Wapo
2. Departemen Informatika – Tunjungan Plaza
3. Departemen Informatika – Masjid Nasional Al-Akbar

Berdasarkan kriteria rute tersebut akan dilihat rekomendasi rute mana yang akan diberikan oleh sistem. Rute yang menjadi rekemondasi akan diwarnai dengan warna biru. Hasil uji coba skenario ini dapat dilihat pada Gambar 5.16, Gambar 5.17, Gambar 5.18.



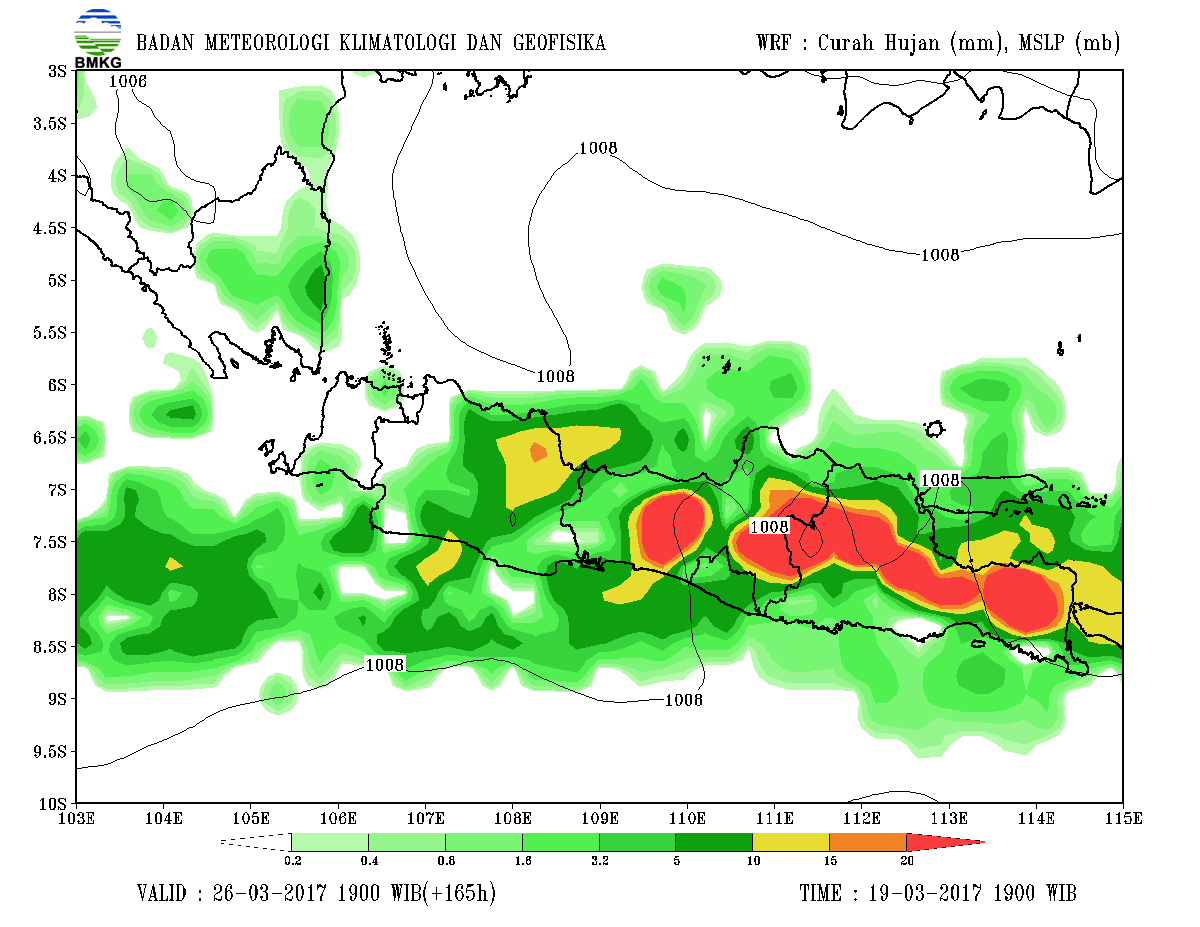
**Gambar 5.16 Hasil Uji Coba Rute A**

**Gambar 5.17 Hasil Uji Coba Rute B**



**Gambar 5.18 Hasil Uji Coba Rute C**

### Skenario Uji Coba 5

Pada uji coba ini akan dilakukan perbandingan hasil dari pengambilan data citra radar antara pengamatan manual dengan menggunakan *script* yang telah dibuat pada tugas akhir ini. Kemudian hasil dari perbandingan ini akan digunakan untuk menghitung akurasi data yang didapat dari *script* yang digunakan. Pada uji coba ini akan digunakan data gambar ramalan cuaca yang dikeluarkan oleh BMKG pada tanggal 26 maret 2017 yang ditunjukkan pada Gambar 5.28 dan digunakan juga titik koordinat pos hujan kota Surabaya sebanyak 208 titik.

**Gambar 5.19 Data Uji Coba Ramalan Cuaca**

Dari hasil uji coba ini didapatkan sebanyak 12 titik pos hujan yang data cuacanya tidak bisa terdeteksi yang ditunjukkan pada Tabel 5.10.

**Tabel 5.10 Kondisi Cuaca yang Tidak Terdeteksi**

|  |  |  |
| --- | --- | --- |
| **Idpos** | **RGB** | **Cuaca** |
| 66 | #000 | Tidak Terdeteksi |
| 67 | #000 | Tidak Terdeteksi |
| 68 | #000 | Tidak Terdeteksi |
| 69 | #000 | Tidak Terdeteksi |
| 70 | #000 | Tidak Terdeteksi |
| 71 | #000 | Tidak Terdeteksi |
| 82 | #000 | Tidak Terdeteksi |
| 83 | #000 | Tidak Terdeteksi |
| 84 | #000 | Tidak Terdeteksi |
| 85 | #000 | Tidak Terdeteksi |
| 86 | #000 | Tidak Terdeteksi |
| 87 | #000 | Tidak Terdeteksi |

Kemudian untuk menghitung persentase akurasi pengambilan data digunakan Persamaan 5.1 berikut.

(5.1)

Keterangan:

* n = banyaknya sampel data (titik pos hujan)
* X = banyaknya sampel data yang salah

Dengan menggunakan Persamaan 5.1 maka didapatkan nilai akurasi sebagai berikut

Keseluruhan data hasil uji coba ini ditunjukkan pada lampiran C.

## Evaluasi

Pada subbab evaluasi akan dibahas mengenai hasil ujicoba yang dilakukan pada subbab uji coba. Hasil dari evaluasi ini akan dimasukan ke bagian kesimpulan dan saran pada bab selanjutnya.

### Evaluasi Skenario Uji Coba 1

Pada uji coba yang telah dilakukan dapat diketahui bahwa rata-rata *response time* yang didapat untuk masing-masing kategori jumlah pengguna berada diatas 5 detik. Ada beberapa hal yang bisa menyebabkan nilai *response time* yang didapat menjadi lama yaitu:

1. Pertama adalah tidak dilakukannya optimasi *server* yang digunakan seperti menggunakan *load balancer* untuk mengatur jumlah *traffic* akses yang melakukan permintaan data.
2. Kedua adalah algoritma yang digunakan untuk mencari rute kurang terlalu efisien untuk digunakan pada sistem yang dibangun.

Dalam uji coba ini dapat diketahui juga bahwa *response time* untuk algoritma *K-Nearest Neighbor* sendiri menggunakan titik koordinat lokasi acak tanpa melakukan pencarian rute berada dibawah 1 detik hal ini ditunjukkan pada Tabel 5.5.

### Evaluasi Skenario Uji Coba 2

Pada uji coba yang telah dilakukan dapat diketahui hasil keluaran lokasi pos hujan yang diberikan berdasarkan algoritma KNN yang digunakan. Pada hasil pengujian seperti yang ditunjukkan oleh Tabel 5.6, Tabel 5.7, Tabel 5.8 setiap pos hujan menunjukan kategori jenis cuaca yang berbeda dikarenakan persebaran curah hujan yang berbeda pada setiap pos sehingga apabila digunakan nilai K lebih dari 1 bisa terjadi suatu kasus saat kondisi cuaca tidak dapat diklasifikasikan karena kondisi cuaca yang saling berbeda pada tiap titik. Sehingga pada Tugas Akhir ini ditetapkan nilai K=1 yang merupakan nilai K terbaik yang dapat digunakan pada algortima *K-Nearest Neighbor* yang digunakan.

### Evaluasi Skenario Uji Coba 3

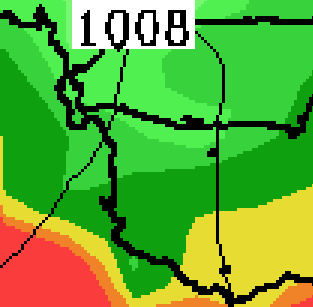
Dari uji coba yang telah dilakukan untuk skenario 3 adalah sistem dapat melakukan pengunduhan secara otomatis berdasarkan waktu yang telah ditentukan. Dari 4 data yang diunduh dari *server* BMKG semua datanya tidak ada yang rusak maupun *corrupt* seperti yang ditunjukkan Gambar 5.14dan Tabel 5.9 semua *file* yang diunduh memiliki ukuran yang sama yaitu 30 KBsehingga dapat dikatakan proses ini telah berjalan dengan baik.

### Evaluasi Skenario Uji Coba 4

Dari uji coba yang telah dilakukan pada skenario 4 dapat terlihat bahwa rekomendasi rute yang diberikan berdasarkan bobot dapat digambarkan pada peta hal ini ditunjukkan pada Gambar 5.16, Gambar 5.17, Gambar 5.18 di mana rute rekomendasi ditandai dengan warna biru. Berdasarkan hasil uji coba tersebut dapat diketahui bahwa rute yang menjadi rekomendasi bisa jadi bukan rute yang terdekat ini dikarenakan pemilihan rekomendasi rute berdasarkan nilai bobot dari kondisi cuaca yang terdapat pada rute sehingga rute yang menjadi rekomendasi tidak selalu rute yang terdekat.

### Evaluasi Skenario Uji Coba 5

Dari uji coba ini dapat diketahui bahwa *script* yang digunakan telah dapat mengambil RGB warna peta berdasarkan koordinat *pixel* lokasi pos hujan. Dari uji coba ini dapat diketahui ada beberapa titik pos hujan yang datanya tidak dapat ditentukam kondisi cuacanya. Hal ini disebabkan terdapat garis berwarna hitam yang memotong peta seperti yang ditunjukkan pada Gambar 5.20.



**Gambar 5.20 Garis Hitam yang Memotong Peta**

Sehingga kode RGB yang didapatkan ialah kode RGB dari garis yang memotong peta tersebut sehingga kondisi cuaca yang terjadi tidak dapat dikenali oleh proses yang sedang berlangsung. Garis hitam yang memotong citra ramalan tersebut bisa berbeda pada setiap data citra yang diunduh dari BMKG seperti Gambar 5.21.



**Gambar 5.21 Peta Dengan Kondisi Cuaca Cerah**

Garis hitam yang memotong gambar peta tersebut ialah garis yang menandakan arah jalannya awan dan garis hitam tersebut tidak akan tergambarkan pada peta dengan kondisi cuaca yang cerah. Didapatkan juga bahwa nilai akurasi yang didapatkan dengan cara membandingkan hasil dari *script* dan pengamatan manual berada pada nilai 94%. Nilai akurasi ini bisa jadi akan berbeda pada setiap data citra yang digunakan hal ini dikarenakan pada setiap peta memiliki kondisi cuaca penggambaran arah angin yang berbeda.

# BAB VI KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.



## Kesimpulan

Setelah melakukan proses perancangan, implementasi dan uji coba diperoleh kesimpulan sebagai berikut:

1. Sistem *web service* yang dibangun untuk aplikasi Clearroute mampu menangani pengolahan data yang diperlukan oleh aplikasi untuk mendapatkan data kondisi cuaca.
2. Sistem telah dapat mengirimkan data rute perjalanan dan cuaca yang telah diolah kepada aplikasi, walau terdapat evaluasi terhadap *response time* pengiriman data.
3. Nilai K dalam algoritma *K-Nearest Neighbor* yang optimal untuk digunakan pada sistem yang dibangun adalah 1.
4. Sistem berhasil melakukan pengunduhan data pos hujan BMKG secara *real time* tanpa adanya data yang rusak.
5. Pilihan rute terbaik berdasarkan nilai bobot rute terhadap cuaca tidak selalu merupakan rute yang terdekat terhadap lokasi tujuan.
6. Sistem dapat melakukan ekstraksi data warna pada citra peta ramalan cuaca yang dikeluarkan oleh BMKG, terdapat evaluasi terhadap beberapa lokasi pos hujan yang datanya tidak terdeteksi dikarenakan adanya suatu garis berwarna hitam yang bisa saja memotong gambar peta walau tidak semua peta memiliki garis hitam yang memotong tergantung kondisi cuaca yang dialaminya.

## Saran

Berikut ini beberapa saran mengenai pengembangan lebih lanjut berdasarkan hasil perancangan implementasi dan uji coba yang dilakukan:

1. Pada data citra peta ramalan cuaca BMKG perlu dilakukan pengolahan lebih lanjut dengan menggunakan proses *image stitching* yaitu menggabungkan beberapa gambar peta ramalan cuaca yang terjadi pada satu hari untuk mendapatkan simulasi gambar pergerakan awan agar data ramalan cuaca bisa lebih akurat memberikan informasi ramalan setiap 1 jam.
2. Diperlukan optimalisasi *server* yang digunakan agar sistem bisa lebih baik dalam menangani permintaan data dari banyak pengguna.

# LAMPIRAN A. LOKASI POS HUJAN

**Tabel A.1 Lokasi Pos Hujan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Pos** | **ydesimal** | **xdesimal** | **y*pixel*** | **y*pixel*** |
| Kelurahan Asemrowo | -7.252 | 112.715 | 280 | 438 |
| Kelurahan Genting | -7.24407 | 112.7166 | 269 | 440 |
| Kelurahan Greges | -7.2295 | 112.684 | 248 | 395 |
| Kelurahan Kalianak | -7.2289 | 112.702 | 248 | 420 |
| Kelurahan Tambak Langon | -7.225 | 112.664 | 242 | 367 |
| Kantor Kecamatan | -7.235 | 112.608 | 256 | 290 |
| Kelurahan Tambak Oso | -7.21192 | 112.6519 | 224 | 351 |
| Kelurahan RomoKalisari | -7.198 | 112.647 | 205 | 344 |
| Kelurahan Klakahrejo | -7.25244 | 112.6496 | 280 | 347 |
| Kelurahan Sememi | -7.249 | 112.647 | 275 | 344 |
| Kelurahan Kandangan | -7.252 | 112.653 | 280 | 352 |
| Kelurahan Jepara | -7.239 | 112.7208 | 262 | 446 |
| Kelurahan Gundih | -7.247 | 112.725 | 273 | 451 |
| Kelurahan Tembok Dukuh | -7.25308 | 112.7202 | 281 | 445 |
| Kelurahan Alon-alon contong | -7.2502 | 112.736 | 277 | 467 |
| Keluarahan Bubutan | -7.253 | 112.733 | 281 | 462 |
| Kelurahan Komplek Kenjeran | -7.247 | 113 | 273 | 544 |
| Kelurahan Sukolilo | -7.241 | 112.792 | 264 | 544 |
| Kelurahan Kenjeran | -7.226 | 112.775 | 243 | 520 |
| Kelurahan Bulak | -7.227 | 112.734 | 245 | 464 |
| Kelurahan Kedungcowek | -7.226 | 112.775 | 243 | 520 |
| Kelurahan Gunungsari | -7.299 | 112.719 | 345 | 443 |
| Kelurahan Dukuh Kupang | -7.27988 | 112.7151 | 318 | 438 |
| Kelurahan Dukuh Pakis | -7.292 | 112.712 | 335 | 433 |
| Kelurahan Pradah Kali Kendal | -7.283 | 112.688 | 323 | 400 |
| Kelurahan Ketintang | -7.309 | 112.725 | 359 | 451 |
| Kelurahan Dukuh Menanggal | -7.343 | 112.723 | 406 | 449 |
| Kelurahan Menanggal | -7.338 | 112.726 | 399 | 453 |
| Kelurahan Gayungan | -7.333 | 112.728 | 392 | 455 |
| Kelurahan Embong Kaliasin | -7.264 | 112.742 | 296 | 475 |
| Kelurahan Ketabang | -7.254 | 112.748 | 282 | 483 |
| Kelurahan Kapasari | -7.246 | 112.751 | 271 | 487 |
| Kelurahan Peneleh | -7.251 | 112.743 | 278 | 476 |
| Kelurahan Genteng | -7.25874 | 112.7423 | 289 | 475 |
| Kelurahan Gubeng | -7.268 | 112.751 | 302 | 487 |
| Kelurahan Kertajaya | -7.282 | 112.758 | 321 | 497 |
| Kelurahan Pucang Sewu | -7.284 | 112.7508 | 324 | 487 |
| Kelurahan Barata Jaya | -7.296 | 112.761 | 341 | 501 |
| Kelurahan Mojo | -7.274 | 112.764 | 310 | 505 |
| Kelurahan Airlangga | -7.276 | 112.758 | 313 | 497 |
| Kelurahan Rungkut Menanggal | -7.334 | 112.768 | 394 | 511 |
| Kelurahan Rungkut Tengah | -7.331 | 112.771 | 389 | 515 |
| Kelurahan Gununganyar | -7.336 | 112.802 | 396 | 557 |
| Kelurahan Gununganyar Tambak | -7.335 | 112.801 | 395 | 556 |
| Kelurahan Karah | -7.309 | 112.713 | 359 | 435 |
| Kelurahan Kebonsari | -7.328 | 112.714 | 385 | 436 |
| Kelurahan Jambangan | -7.321 | 112.712 | 376 | 433 |
| Kelurahan Pagesangan | -7.334 | 112.708 | 394 | 428 |
| Kelurahan Karangpilang | -7.3409 | 112.694 | 403 | 409 |
| Kelurahan Warugunung | -7.35 | 112.673 | 416 | 380 |
| Kelurahan Kebraon | -7.333 | 112.698 | 392 | 414 |
| Kelurahan Kedurus | -7.315 | 112.709 | 367 | 429 |
| Kelurahan Bulak Banteng | -7.224 | 112.775 | 241 | 520 |
| Kelurahan Sidotopo Wetan | -7.231 | 112.757 | 250 | 495 |
| Kelurahan Tambak Wedi | -7.208 | 112.774 | 218 | 519 |
| Kelurahan Tanah Kali Kedinding | -7.225 | 112.775 | 242 | 520 |
| Kelurahan Kemayoran | -7.237 | 112.729 | 259 | 457 |
| Kelurahan Krembangan Selatan | -7.232 | 112.734 | 252 | 464 |
| Kelurahan Moro Krembangan | -7.233 | 112.722 | 253 | 447 |
| Kelurahan Perak Barat | -7.23031 | 112.7267 | 249 | 454 |
| Kelurahan Jeruk | -7.305 | 112.644 | 353 | 340 |
| Kelurahan Lakarsantri | -7.303 | 112.632 | 350 | 323 |
| Kelurahan Lidah Kulon | -7.305 | 112.65 | 353 | 348 |
| Kelurahan Lidah Wetan | -7.307 | 112.665 | 356 | 369 |
| Kelurahan Sumur Welut | -7.325 | 112.672 | 381 | 378 |
| Kelurahan Dukuh Sutorejo | -7.259 | 112.791 | 289 | 542 |
| Kelurahan Kalijudan | -7.257 | 112.777 | 287 | 523 |
| Kelurahan Kalisari | -7.264 | 112.798 | 296 | 552 |
| Kelurahan Kejawan Putih Tambak | -7.276 | 112.801 | 313 | 556 |
| Kelurahan Manyar Sabrangan | -7.282 | 112.78 | 321 | 527 |
| Kelurahan Mulyorejo | -7.26 | 112.785 | 291 | 534 |
| Kelurahan Bongkaran | -7.24029 | 112.7398 | 263 | 472 |
| Kelurahan Krembangan Utara | -7.232 | 112.734 | 252 | 464 |
| Kelurahan Nyamplungan | -7.226 | 112.74 | 243 | 472 |
| Kelurahan Perak Timur | -7.2235 | 112.7338 | 240 | 464 |
| Kelurahan Perak Utara | -7.21779 | 112.7325 | 232 | 462 |
| Kelurahan Babat Jerawat | -7.242 | 112.632 | 266 | 323 |
| Kelurahan Benowo | -7.235 | 112.608 | 256 | 290 |
| Kelurahan Pakal | -7.237 | 112.617 | 259 | 302 |
| Kelurahan Sumber Rejo | -7.2258 | 112.605 | 287 | 243 |
| Kelurahan Tambak Dono | -7.217 | 112.616 | 231 | 301 |
| Kelurahan Kalirungkut | -7.322 | 112.77 | 377 | 513 |
| Kelurahan Kedung Baruk | -7.321 | 112.777 | 376 | 523 |
| Kelurahan Medoan Ayu | -7.324 | 112.797 | 380 | 551 |
| Kelurahan Penjaringan Sari | -7.314 | 112.784 | 366 | 533 |
| Kelurahan Rungkut Kidul | -7.329 | 112.772 | 387 | 516 |
| Kelurahan Wonorejo | -7.31 | 112.796 | 360 | 549 |
| Kelurahan Lontar | -7.284 | 112.662 | 324 | 365 |
| Kelurahan Made | -7.276 | 112.637 | 313 | 330 |
| Kelurahan Sambikerep | -7.274 | 112.653 | 310 | 352 |
| Kelurahan Bringin | -7.26 | 112.646 | 291 | 342 |
| Kelurahan Sawahan | -7.26573 | 112.7275 | 299 | 455 |
| Kelurahan Kupang Krajan | -7.266 | 112.721 | 299 | 446 |
| Kelurahan Petemon | -7.259 | 112.723 | 289 | 449 |
| Kelurahan Banyu Urip | -7.265 | 112.727 | 298 | 454 |
| Kelurahan Pakis | -7.28691 | 112.7151 | 328 | 438 |
| Kelurahan Ampel | -7.232 | 112.742 | 252 | 475 |
| Kelurahan Pegirian | -7.225 | 112.752 | 242 | 489 |
| Keluarahan Ujung | -7.222 | 112.74 | 238 | 472 |
| Kelurahan Wonokusumo | -7.226 | 112.775 | 243 | 520 |
| Kelurahan Kapasan | -7.238 | 112.748 | 260 | 483 |
| Kelurahan Sidodadi | -7.235 | 112.747 | 256 | 482 |
| Kelurahan Simokerto | -7.234 | 112.755 | 255 | 493 |
| Kelurahan Simolawang | -7.232 | 112.748 | 252 | 483 |
| Kelurahan Tambak Rejo | -7.246 | 112.752 | 271 | 489 |
| Kelurahan Gebang Putih | -7.283 | 112.785 | 323 | 534 |
| Kelurahan Keputih | -7.295 | 112.801 | 339 | 556 |
| Kelurahan Klampis Ngasem | -7.289 | 112.784 | 331 | 533 |
| Kelurahan Medoan Semampir | -7.306 | 112.781 | 355 | 529 |
| Kelurahan Menur Prumpungan | -7.294 | 112.77 | 338 | 513 |
| Kelurahan Nginden Jangkungan | -7.301 | 112.769 | 348 | 512 |
| Kelurahan Semolowaru | -7.3 | 112.779 | 346 | 526 |
| Kelurahan Putat Gede | -7.285 | 112.698 | 325 | 414 |
| Kelurahan Simo Mulyo | -7.265 | 112.711 | 298 | 432 |
| Kelurahan Sonok Wijen | -7.276 | 112.704 | 313 | 422 |
| Kelurahan Suko Manunggal | -7.266 | 112.699 | 299 | 416 |
| Kelurahan Tanjung Sari | -7.26 | 112.692 | 291 | 406 |
| Kelurahan Pacar Keling | -7.25746 | 112.7624 | 287 | 503 |
| Kelurahan Pacar Kembang | -7.2555 | 112.7663 | 284 | 508 |
| Kelurahan Ploso | -7.25217 | 112.7615 | 280 | 502 |
| Kelurahan Gading | -7.24808 | 112.7778 | 274 | 524 |
| Kelurahan Rangkah | -7.24843 | 112.766 | 275 | 507 |
| Kelurahan Tambak Sari | -7.25348 | 112.7541 | 282 | 491 |
| Kelurahan Balongsari | -7.258 | 112.679 | 288 | 388 |
| Kelurahan Buntaran | -7.24971 | 112.673 | 276 | 380 |
| Kelurahan Banjar Sugihan | -7.254 | 112.657 | 282 | 358 |
| Kelurahan Karang Poh | -7.262 | 112.681 | 294 | 391 |
| Kelurahan Manukan Kulon | -7.256 | 112.666 | 285 | 370 |
| Kelurahan Manukan Wetan | -7.258 | 112.671 | 288 | 377 |
| Kelurahan Bibis | -7.26048 | 112.6746 | 291 | 382 |
| Kelurahan Gedang Asin | -7.25963 | 112.6882 | 290 | 401 |
| Kelurahan Karang Poh | -7.26255 | 112.6816 | 294 | 391 |
| Kelurahan Tandes | -7.26202 | 112.6841 | 294 | 395 |
| Kelurahan Gadel | -7.27024 | 112.6759 | 305 | 384 |
| Kelurahan Tubanan | -7.27567 | 112.6841 | 313 | 395 |
| Kelurahan Kedungdoro | -7.263 | 112.734 | 295 | 464 |
| Kelurahan Tegalsari | -7.286 | 112.741 | 327 | 473 |
| Kelurahan Wonorejo | -7.27 | 112.732 | 305 | 461 |
| Kelurahan Dr.Sutomo | -7.28 | 112.739 | 319 | 471 |
| Kelurahan Keputran | -7.282 | 112.741 | 321 | 473 |
| Kelurahan Kendangsari | -7.324 | 112.743 | 380 | 476 |
| Kelurahan Kutisari | -7.334 | 112.748 | 394 | 483 |
| Kelurahan Panjangjiwo | -7.31 | 112.765 | 360 | 506 |
| Kelurahan Prapen | -7.313 | 112.756 | 364 | 494 |
| Kelurahan Tenggilis Mejoyo | -7.318 | 112.761 | 371 | 501 |
| Kelurahan Babatan | -7.309 | 112.674 | 359 | 381 |
| Kelurahan Balas Klumprik | -7.331 | 112.688 | 389 | 400 |
| Kelurahan Jajar Tunggal | -7.314 | 112.699 | 366 | 416 |
| Kelurahan Wiyung | -7.314 | 112.697 | 366 | 413 |
| Kelurahan Bendul Merisi | -7.309 | 112.745 | 359 | 479 |
| Kelurahan Jemur Wonosari | -7.32 | 112.739 | 374 | 471 |
| Kelurahan Margorejo | -7.316 | 112.737 | 369 | 468 |
| Kelurahan Sidosermo | -7.306 | 112.753 | 355 | 490 |
| Kelurahan Siwalankerto | -7.339 | 112.736 | 401 | 467 |
| Kelurahan Darmo | -7.291 | 112.731 | 334 | 460 |
| Kelurahan Jagir | -7.306 | 112.741 | 355 | 473 |
| Kelurahan Ngagel | -7.277 | 112.745 | 314 | 479 |
| Kelurahan Ngagel Rejo | -7.297 | 112.743 | 342 | 476 |
| Kelurahan Sawunggaling | -7.293 | 112.727 | 337 | 454 |
| Kelurahan Wonokromo | -7.30325 | 112.7281 | 351 | 456 |
| Balaikota Surabaya | -7.26063 | 112.7467 | 292 | 481 |
| Bandara Juanda Terminal 1 | -7.3728 | 112.7951 | 448 | 548 |
| Bandara Juanda Terminal 2 | -7.3835 | 112.776 | 462 | 522 |
| Terminal Bungurasih | -7.35117 | 112.7245 | 417 | 451 |
| Terminal Joyoboyo | -7.29892 | 112.7366 | 345 | 467 |
| Terminal Osowilangun | -7.20938 | 112.6518 | 220 | 350 |
| Terminal Bratang | -7.29885 | 112.7614 | 345 | 501 |
| Jembatan Suramadu sisi SBY | -7.21128 | 112.7786 | 223 | 525 |
| Jembatan Suramadu sisi BKL | -7.15838 | 112.782 | 149 | 530 |
| Monumen Kapal Selam | -7.26548 | 112.7503 | 298 | 486 |
| Stasiun Wonokromo | -7.302 | 112.7392 | 349 | 471 |
| Stasiun Pasar Turi | -7.24816 | 112.7311 | 274 | 460 |
| Stasiun Semut | -7.24285 | 112.7404 | 267 | 473 |
| Stasun Gubeng | -7.26552 | 112.7521 | 298 | 489 |
| Mall CITO | -7.34434 | 112.7285 | 408 | 456 |
| Mall Tunjungan Plasa | -7.26247 | 112.7395 | 294 | 471 |
| Mall Royal | -7.30961 | 112.7349 | 360 | 465 |
| Mall DTC | -7.30217 | 112.738 | 349 | 469 |
| Mall GalaxyMall | -7.27602 | 112.7816 | 313 | 529 |
| Tugu pahlawan | -7.24585 | 112.7378 | 271 | 469 |
| Jembatan Merah | -7.23664 | 112.7383 | 258 | 470 |
| House of Sampoerna | -7.23058 | 112.7343 | 250 | 464 |
| Ciputra Waterpark | -7.28546 | 112.633 | 326 | 325 |
| Masjid Al Akbar | -7.33637 | 112.715 | 397 | 438 |
| Museum Mpu Tantular | -7.43349 | 112.7203 | 532 | 445 |
| Museum WR Supratman | -7.24336 | 112.7609 | 268 | 501 |
| Kebun Binatang Surabaya | -7.2966 | 112.7385 | 342 | 470 |
| Hutan Manggrove Wonorejo | -7.30736 | 112.8233 | 357 | 587 |
| Rumah Batik | -7.31537 | 112.7404 | 368 | 473 |
| Pantai Kenjeran | -7.23177 | 112.7932 | 252 | 545 |
| Kampus ITS | -7.27924 | 112.7908 | 317 | 542 |
| Unair Kampus | -7.27246 | 112.7573 | 308 | 496 |
| Pelabuhan Tanjung Perak | -7.19971 | 112.734 | 207 | 464 |
| Pelabuhan Teluk Lamong | -7.20438 | 112.6693 | 213 | 375 |
| Kebun Bibit Wonorejo | -7.31112 | 112.7892 | 362 | 540 |
| Taman Apsari | -7.26374 | 112.7428 | 296 | 476 |
| Taman Bungkul | -7.29115 | 112.7395 | 334 | 471 |
| Taman Ekspresi | -7.25612 | 112.7416 | 285 | 474 |
| Taman Jayengrono | -7.23636 | 112.7377 | 258 | 469 |
| Museum Bank Indonesia | -7.29449 | 112.7388 | 339 | 470 |
| Masjid Ampel | -7.22947 | 112.7426 | 248 | 476 |
| Gwalk Market | -7.29025 | 112.6546 | 333 | 354 |
| Surabaya Town Square | -7.29456 | 112.7302 | 339 | 459 |
| Ciputra World | -7.293 | 112.7194 | 337 | 444 |
| Maspion Square | -7.31612 | 112.7361 | 369 | 467 |
| Gogo Mall | -7.29494 | 112.7605 | 339 | 500 |
| Pasar Atum Mall | -7.2418 | 112.7443 | 265 | 478 |
| BG Junction Mall | -7.2549 | 112.7329 | 284 | 462 |
| Lenmarck Mall | -7.28556 | 112.6814 | 326 | 391 |
| Golden City Mall | -7.29167 | 112.706 | 335 | 425 |
| Plasa Surabaya | -7.26506 | 112.7487 | 298 | 484 |
| Unitomo | -7.29883 | 112.7658 | 345 | 508 |
| ITATS | -7.28997 | 112.7791 | 332 | 526 |
| UBHARA | -7.32139 | 112.7327 | 376 | 462 |
| UNNESA | -7.31413 | 112.7268 | 366 | 454 |
| Univ Narotama | -7.28803 | 112.7765 | 330 | 522 |
| Untag | -7.29961 | 112.767 | 346 | 509 |
| PENS | -7.27592 | 112.794 | 313 | 546 |
| Univ Petra | -7.33894 | 112.7366 | 400 | 467 |
| Plaza Marina | -7.31627 | 112.7487 | 369 | 484 |
| IAIN Sunan Ampel | -7.32166 | 112.7349 | 376 | 465 |
| Univ Hang Tuah | -7.30919 | 112.7371 | 359 | 468 |
| Pakuwon Trade Centre | -7.28525 | 112.6719 | 326 | 378 |
| PT. TPS | -7.21879 | 112.7205 | 233 | 445 |
| Akademi Angkatan Laut | -7.22316 | 112.7144 | 240 | 437 |
| PT. PAL | -7.20554 | 112.7416 | 215 | 474 |

***(Halaman ini sengaja dikosongkan)***

# C:\Users\RIDHO\AppData\Local\Microsoft\Windows\INetCache\Content.Word\basis_data_clearroute.pngLAMPIRAN B. Basis Data Clearroute

***(Halaman ini sengaja dikosongkan)***

# LAMPIRAN C. Hasil Uji Coba Skenario 5

**Tabel C.1 Hasil Uji Coba Skenario 5**

| idpos | RGB | cuaca |
| --- | --- | --- |
| 1 | #37d23c | berawan |
| 2 | #37d23c | berawan |
| 3 | #37d23c | berawan |
| 4 | #37d23c | berawan |
| 5 | #37d23c | berawan |
| 6 | #fa0f | hujan ringan |
| 7 | #fa0f | hujan ringan |
| 8 | #fa0f | hujan ringan |
| 9 | #fa0f | hujan ringan |
| 10 | #fa0f | hujan ringan |
| 11 | #fa0f | hujan ringan |
| 12 | #37d23c | berawan |
| 13 | #37d23c | berawan |
| 14 | #37d23c | berawan |
| 15 | #37d23c | berawan |
| 16 | #37d23c | berawan |
| 17 | #37d23c | berawan |
| 18 | #37d23c | berawan |
| 19 | #37d23c | berawan |
| 20 | #37d23c | berawan |
| 21 | #37d23c | berawan |
| 22 | #37d23c | berawan |
| 23 | #37d23c | berawan |
| 24 | #37d23c | berawan |
| 25 | #37d23c | berawan |
| 26 | #fa0f | hujan ringan |
| 27 | #fa0f | hujan ringan |
| 28 | #fa0f | hujan ringan |
| 29 | #fa0f | hujan ringan |
| 30 | #37d23c | berawan |
| 31 | #37d23c | berawan |
| 32 | #37d23c | berawan |
| 33 | #37d23c | berawan |
| 34 | #37d23c | berawan |
| 35 | #37d23c | berawan |
| 36 | #37d23c | berawan |
| 37 | #37d23c | berawan |
| 38 | #37d23c | berawan |
| 39 | #37d23c | berawan |
| 40 | #37d23c | berawan |
| 41 | #37d23c | berawan |
| 42 | #37d23c | berawan |
| 43 | #37d23c | berawan |
| 44 | #37d23c | berawan |
| 45 | #37d23c | berawan |
| 46 | #37d23c | berawan |
| 47 | #37d23c | berawan |
| 48 | #37d23c | berawan |
| 49 | #37d23c | berawan |
| 50 | #37d23c | berawan |
| 51 | #37d23c | berawan |
| 52 | #37d23c | berawan |
| 53 | #37d23c | berawan |
| 54 | #37d23c | berawan |
| 55 | #37d23c | berawan |
| 56 | #37d23c | berawan |
| 57 | #37d23c | berawan |
| 58 | #37d23c | berawan |
| 59 | #37d23c | berawan |
| 60 | #37d23c | berawan |
| 61 | #fa0f | hujan ringan |
| 62 | #fa0f | hujan ringan |
| 63 | #fa0f | hujan ringan |
| 64 | #fa0f | hujan ringan |
| 65 | #fa0f | hujan ringan |
| 66 | #000 | Tidak Terdeteksi |
| 67 | #000 | Tidak Terdeteksi |
| 68 | #000 | Tidak Terdeteksi |
| 69 | #000 | Tidak Terdeteksi |
| 70 | #000 | Tidak Terdeteksi |
| 71 | #000 | Tidak Terdeteksi |
| 72 | #37d23c | berawan |
| 73 | #37d23c | berawan |
| 74 | #37d23c | berawan |
| 75 | #37d23c | berawan |
| 76 | #37d23c | berawan |
| 77 | #37d23c | berawan |
| 78 | #37d23c | berawan |
| 79 | #37d23c | berawan |
| 80 | #37d23c | berawan |
| 81 | #37d23c | berawan |
| 82 | #000 | Tidak Terdeteksi |
| 83 | #000 | Tidak Terdeteksi |
| 84 | #000 | Tidak Terdeteksi |
| 85 | #000 | Tidak Terdeteksi |
| 86 | #000 | Tidak Terdeteksi |
| 87 | #000 | Tidak Terdeteksi |
| 88 | #fa0f | hujan ringan |
| 89 | #fa0f | hujan ringan |
| 90 | #fa0f | hujan ringan |
| 91 | #fa0f | hujan ringan |
| 92 | #37d23c | berawan |
| 93 | #37d23c | berawan |
| 94 | #37d23c | berawan |
| 95 | #37d23c | berawan |
| 96 | #37d23c | berawan |
| 97 | #37d23c | berawan |
| 98 | #37d23c | berawan |
| 99 | #37d23c | berawan |
| 100 | #37d23c | berawan |
| 101 | #37d23c | berawan |
| 102 | #37d23c | berawan |
| 103 | #37d23c | berawan |
| 104 | #37d23c | berawan |
| 105 | #37d23c | berawan |
| 106 | #37d23c | berawan |
| 107 | #37d23c | berawan |
| 108 | #37d23c | berawan |
| 109 | #37d23c | berawan |
| 110 | #37d23c | berawan |
| 111 | #37d23c | berawan |
| 112 | #37d23c | berawan |
| 113 | #37d23c | berawan |
| 114 | #37d23c | berawan |
| 115 | #37d23c | berawan |
| 116 | #37d23c | berawan |
| 117 | #37d23c | berawan |
| 118 | #37d23c | berawan |
| 119 | #37d23c | berawan |
| 120 | #37d23c | berawan |
| 121 | #37d23c | berawan |
| 122 | #37d23c | berawan |
| 123 | #37d23c | berawan |
| 124 | #37d23c | berawan |
| 125 | #37d23c | berawan |
| 126 | #37d23c | berawan |
| 127 | #37d23c | berawan |
| 128 | #37d23c | berawan |
| 129 | #37d23c | berawan |
| 130 | #37d23c | berawan |
| 131 | #37d23c | berawan |
| 132 | #37d23c | berawan |
| 133 | #37d23c | berawan |
| 134 | #37d23c | berawan |
| 135 | #37d23c | berawan |
| 136 | #37d23c | berawan |
| 137 | #37d23c | berawan |
| 138 | #37d23c | berawan |
| 139 | #37d23c | berawan |
| 140 | #37d23c | berawan |
| 141 | #37d23c | berawan |
| 142 | #37d23c | berawan |
| 143 | #37d23c | berawan |
| 144 | #37d23c | berawan |
| 145 | #37d23c | berawan |
| 146 | #37d23c | berawan |
| 147 | #37d23c | berawan |
| 148 | #37d23c | berawan |
| 149 | #37d23c | berawan |
| 150 | #37d23c | berawan |
| 151 | #37d23c | berawan |
| 152 | #37d23c | berawan |
| 153 | #37d23c | berawan |
| 154 | #37d23c | berawan |
| 155 | #37d23c | berawan |
| 156 | #37d23c | berawan |
| 157 | #37d23c | berawan |
| 158 | #37d23c | berawan |
| 159 | #37d23c | berawan |
| 160 | #37d23c | berawan |
| 161 | #37d23c | berawan |
| 164 | #fa0f | hujan ringan |
| 165 | #37d23c | berawan |
| 166 | #fa0f | hujan ringan |
| 167 | #37d23c | berawan |
| 168 | #37d23c | berawan |
| 169 | #37d23c | berawan |
| 171 | #37d23c | berawan |
| 172 | #37d23c | berawan |
| 173 | #37d23c | berawan |
| 174 | #37d23c | berawan |
| 175 | #fa0f | hujan ringan |
| 176 | #37d23c | berawan |
| 177 | #fa0f | hujan ringan |
| 178 | #37d23c | berawan |
| 179 | #37d23c | berawan |
| 180 | #37d23c | berawan |
| 181 | #37d23c | berawan |
| 182 | #37d23c | berawan |
| 184 | #37d23c | berawan |
| 187 | #37d23c | berawan |
| 188 | #37d23c | berawan |
| 190 | #37d23c | berawan |
| 191 | #37d23c | berawan |
| 192 | #37d23c | berawan |
| 193 | #37d23c | berawan |
| 194 | #37d23c | berawan |
| 197 | #37d23c | berawan |
| 198 | #37d23c | berawan |
| 200 | #37d23c | berawan |
| 201 | #37d23c | berawan |
| 202 | #fa0f | hujan ringan |
| 204 | #fa0f | hujan ringan |
| 205 | #fa0f | hujan ringan |
| 206 | #37d23c | berawan |
| 207 | #37d23c | berawan |
| 208 | #37d23c | berawan |
| 211 | #37d23c | berawan |
| 212 | #37d23c | berawan |
| 213 | #37d23c | berawan |
| 214 | #fa0f | hujan ringan |
| 215 | #fa0f | hujan ringan |
| 216 | #37d23c | berawan |
| 217 | #37d23c | berawan |
| 218 | #37d23c | berawan |
| 222 | #37d23c | berawan |
| 223 | #37d23c | berawan |
| 225 | #37d23c | berawan |

# 

# DAFTAR PUSTAKA

[1] I. Gustari, T. W. Hadi, S. Hadi, dan F. Renggono, “AKURASI PREDIKSI CURAH HUJAN HARIAN OPERASIONAL DI JABODETABEK : PERBANDINGAN DENGAN MODEL WRF,” *J. Meteorol. Dan Geofis.*, vol. 13, no. 2, 2012.

[2] BMKG 2017 dan B. M. Geofisika Klimatologi dan, “Sejarah | BMKG,” *BMKG | Badan Meteorologi, Klimatologi, dan Geofisika*. [Daring]. Tersedia pada: ?p=sejarah&lang=ID. [Diakses: 29-Mei-2017].

[3] “Tugas dan Fungsi | BMKG.” [Daring]. Tersedia pada: http://www.bmkg.go.id/profil/?p=tugas-fungsi. [Diakses: 04-Jan-2017].

[4] “.: Jaringan Pos Hujan - Climate Early Warning System - CEWS BMKG:.” [Daring]. Tersedia pada: http://cews.bmkg.go.id/Sistem\_Observasi/Jaringan\_Pos\_Hujan.bmkg. [Diakses: 01-Mei-2017].

[5] “Application programming interface - Wikipedia.” [Daring]. Tersedia pada: https://en.wikipedia.org/wiki/Application\_programming\_interface. [Diakses: 04-Jan-2017].

[6] “PHP - Wikipedia.” [Daring]. Tersedia pada: https://en.wikipedia.org/wiki/PHP. [Diakses: 04-Jan-2017].

[7] “JSON.” [Daring]. Tersedia pada: http://www.json.org/json-id.html. [Diakses: 29-Mei-2017].

[8] “XML Introduction.” [Daring]. Tersedia pada: http://www.w3schools.com/xml/xml\_whatis.asp. [Diakses: 04-Jan-2017].

[9] “PostgreSQL: About.” [Daring]. Tersedia pada: https://www.postgresql.org/about/. [Diakses: 04-Jan-2017].

[10] “PostgreSQL,” *Wikipedia*. 23-Mei-2017.

[11] “PostgreSQL: PostgreSQL Featured Users.” [Daring]. Tersedia pada: https://www.postgresql.org/about/users/. [Diakses: 29-Mei-2017].

[12] A. Chunming dan L. Zongsen, “Studies on kNN query of moving objects for location management in spatial database,” in *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, 2013, hal. 2428–2432.

[13] “k-nearest neighbors algorithm - Wikipedia.” [Daring]. Tersedia pada: https://en.wikipedia.org/wiki/K-nearest\_neighbors\_algorithm. [Diakses: 04-Jan-2017].

[14] “GeoJson Specification.” [Daring]. Tersedia pada: http://geojson.org/geojson-spec.html#introduction. [Diakses: 26-Apr-2017].

[15] “The RGB (CMY) Color Model - Color Models - Technical Guides.” [Daring]. Tersedia pada: http://dba.med.sc.edu/price/irf/Adobe\_tg/models/rgbcmy.html. [Diakses: 23-Mei-2017].

# BIODATA PENULIS

Anwar Rosyidi atau yang biasa dipanggil Anwar di lingkungan kampus. Saat ini sedang menempuh pendidikan S1 Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya. Penulis lahir pada tanggal 30 Januari 1995. Penulis mulai mempelajari ilmu dasar pemrograman ketika masuk dunia perkuliahan. Pada semester 3 penulis sudah mulai membuat aplikasi berbasis website dan mengerjakan permintaan-permintaan untuk membuat apliasi berbasis website dari masyarakat umum. Selama berkuliah penulis juga aktif dibeberpa organisasi seperti himpunan mahasiswa dan BEM Fakultas**.** Penulis dapat dihubungi melalui alamat *email* **anwarrosyidi.5@gmail.com**