

**PEMBANGUNAN ARSITEKTUR WEBGIS UNTUK PEMETAAN
DAERAH YANG TERDAMPAK ERUPSI GUNUNG MERAPI
(STUDI KASUS: KABUPATEN SLEMAN)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Ahmad Arif Saputra
NIM: 135150401111047



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PEMBANGUNAN ARSITEKTUR WEBGIS UNTUK PEMETAAN DAERAH YANG TERDAMPAK ERUPSI GUNUNG MERAPI (STUDI KASUS: KABUPATEN SLEMAN)

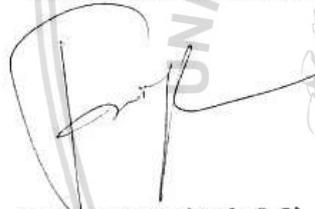
SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ahmad Arif Saputra
NIM: 135150401111047

Skripsi ini telah diuji dan dinyatakan lulus pada
11 Januari 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



D.Sc. Fatwa Ramdani, S.Si., M.Sc.

NIK: 2016118506191001

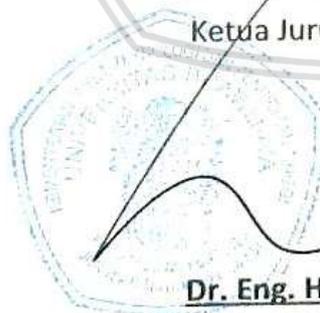
Dosen Pembimbing II



Suprpto, S.T., M.T

NIP: 19710727 199603 1 001

Mengetahui
Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T, M.T

NIP: 19740823 200012 1 001

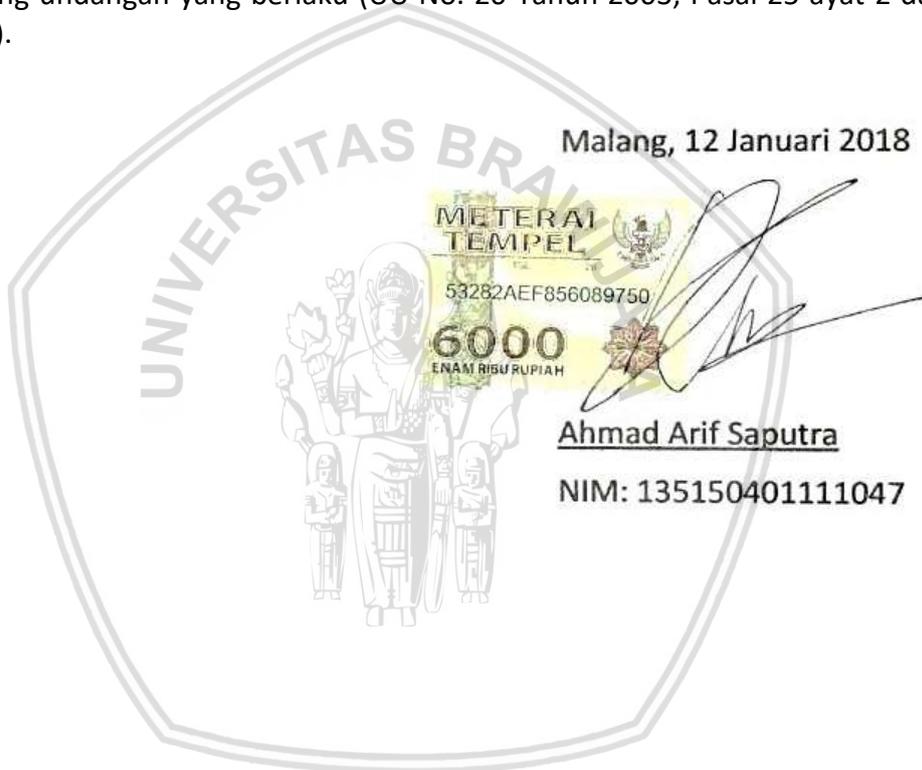


PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Januari 2018



METERAI
TEMPEL

53282AEF856089750

6000
ENAM RIBU RUPIAH

Ahmad Arif Saputra

NIM: 135150401111047

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT, atas segala rahmat, hidayah, dan bimbingan-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pembangunan Arsitektur WebGIS untuk Pemetaan Daerah yang Terdampak Erupsi Gunung Merapi (Studi Kasus : Kabupaten Sleman)” dengan baik. Skripsi ini merupakan sebagian persyaratan untuk memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya. Dalam menyelesaikan skripsi ini, penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Maka dari itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada :

1. Kedua orang tua dan seluruh keluarga tercinta yang telah memberikan doa, motivasi, kasih sayang serta dukungan baik secara moril ataupun materil dalam penyelesaian skripsi ini.
2. Bapak D.Sc. Fatwa Ramdani, S.Si., M.Sc. selaku dosen pembimbing I yang telah bersedia menerima ide penulis sebagai penelitian skripsi dan bersedia menjadi pembimbing, sabar dalam membimbing dan mengingatkan penulis, memahami keadaan penulis, serta selalu memberikan ilmu, motivasi, saran, arahan, dan doa sehingga akhirnya skripsi ini dapat terselesaikan dengan baik.
3. Bapak Suprpto, S.T., M.T. selaku dosen pembimbing II dan ketua program studi Sistem Informasi yang telah bersedia menjadi pembimbing penulis serta banyak memberikan ilmu, arahan, saran, motivasi, dan doa untuk penyelesaian skripsi ini.
4. Bapak dan Ibu Dosen Fakultas Ilmu Komputer yang telah mendidik dan memberikan ilmunya kepada penulis baik di dalam ataupun di luar perkuliahan.
5. Pihak Badan Penanggulangan Bencana Daerah Kabupaten Sleman yang telah memperbolehkan wawancara dan membantu dalam penelitian, sehingga penelitian dapat berjalan dengan baik dan akhirnya terselesaikan.
6. Para anggota dan teman-teman di *Geoinformatics Research Center* Fakultas Ilmu Komputer yang tidak dapat saya sebutkan satu per satu.
7. Para sahabat tercinta, seluruh teman-teman Eksekutif Mahasiswa Sistem Informasi, dan seluruh teman-teman Sistem Informasi yang telah banyak menemani, membantu, memberi masukan, mengingatkan, memberi dukungan, dan motivasi selama masa perkuliahan.
8. Amaliah Khoirun Nisyak yang telah banyak menemani, mendukung dan membantu selama proses pengerjaan skripsi ini.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dalam penyelesaian skripsi ini.

Semoga segala kebaikan, kesabaran, dan pertolongan yang telah diberikan kepada penulis mendapatkan balasan dari Allah SWT. Selama penyelesaian skripsi ini, penulis mendapatkan banyak ilmu dan pengalaman, khususnya di bidang Geoinformatika. Penulis berharap pengalaman dan ilmu ini akan semakin berkembang dan bertambah di kemudiah hari.

Dalam penulisan skripsi ini, penulis menyadari bahwa masih terdapat banyak kekurangan, oleh karena itu saran dan kritik yang membangun sangat diharapkan oleh penulis. Dan terakhir, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi semua pihak.

Malang, 12 Januari 2018



Penulis

ahmadarifsaputra@gmail.com

ABSTRAK

Gunung Merapi merupakan salah satu gunung api dengan erupsi paling aktif di Indonesia. Aktivitas Gunung Merapi pada tahun 2010 merupakan letusan dengan erupsi yang cukup besar dibandingkan erupsi yang terjadi pada tahun 1870. Kabupaten Sleman merupakan salah satu daerah yang terkena dampak letusan Gunung Merapi paling besar. Beberapa daerah di Sleman kini telah berubah menjadi dataran yang dipenuhi dengan debu – debu akibat letusan Gunung Merapi. Menurut hasil wawancara yang dilakukan dengan Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Sleman, Gunung Merapi diperkirakan dapat meletus kapan saja dalam jangka waktu dekat. Hal tersebut dikarenakan aktifitas lava yang masih tinggi dari Gunung Merapi dan lava tersebut seiring bertambahnya waktu semakin mencapai permukaan. Pipa-pipa lava yang digunakan sebagai jalan keluar lava dan uap panas yang mencegah terjadinya erupsi, saat ini juga telah tertutup oleh lava yang mengeras. Keadaan-keadaan tersebut mendukung Gunung Merapi untuk meletus kembali dalam jangka waktu dekat. Saat ini BPBD belum memiliki data *history* terkait daerah-daerah yang terdampak erupsi pada tahun-tahun kejadian sebelumnya dalam bentuk peta. Data tersebut dapat digunakan sebagai data dalam melakukan mitigasi bencana dan memprediksi luasan daerah yang mungkin akan terkena dampak apabila terjadi erupsi kembali. Untuk mendapatkan data tersebut, akan digunakan teknik klasifikasi terbimbing dengan menggunakan *Semi Automatic Classification Plug-In* (SCP) dari aplikasi Quantum GIS. Teknik ini digunakan untuk memetakan daerah-daerah yang terkena erupsi di Kabupaten Sleman dengan mendefinisikan data latih berupa *polygon*. Kemudian hasil dari pemetaan akan ditampilkan dengan menggunakan Sistem Informasi Geografis berbasis web (WebGIS), sehingga diharapkan dapat memberikan informasi hasil pemetaan erupsi Gunung Merapi dengan baik dan jelas kepada pengguna. WebGIS dibuat berdasarkan perancangan arsitektur yang dibagi kedalam 3 *layer*, yaitu *Client Layer*, *Service Layer*, dan *Support Layer*. Berdasarkan hasil pemetaan, didapatkan luasan dari daerah yang terdampak erupsi untuk setiap tahun kejadian sebagai berikut tahun 1992 dengan luas 281.94 ha dan persentase 4,1 %, tahun 1994 dengan luas 1058.61 ha dan persentase 15,2 %, tahun 1997 dengan luas 741.11 ha dan persentase 10,7 %, tahun 1998 dengan luas 278.6 ha dan persentase 4%, tahun 2001 dengan luas 499.97 ha dan persentase 7,2 %, tahun 2006 dengan luas 783.94 ha dan persentase 11,3 %, dan tahun 2010 dengan luas 3309.55 ha dan persentase 47,6 %. Sedangkan dari hasil pengujian akurasi didapatkan tingkat akurasi 92% dan presisi 86,2%, yang berarti bahwa hasil pemetaan memiliki akurasi dan presisi yang tinggi.

Kata kunci: Sistem Informasi Geografis, SCP, Arsitektur, Gunung Merapi, Kabupaten Sleman

ABSTRACT

Merapi Mountain is one of volcano with the most active eruption in Indonesia. The eruption of Merapi Mountain in 2010 was a considerable eruption compared to the eruption that occurred in 1870. Sleman District is one of the largest areas affected by the eruption of Merapi Mountain. Some areas in Sleman have turned into a plateau filled with dust from the eruption of Merapi Mountain. According to interviews with Badan Penanggulangan Bencana Daerah (BPBD) of Sleman Regency, Merapi Mountain is predicted to erupt anytime in the near future. This is because the activity of lava is still high from Merapi Mountain and the lava is increasingly reaching the surface. The lava pipe used as a lava exit and hot steam that prevents eruption has been enclosed by hardened lava. These conditions support Merapi Mountain to erupt in the near future. Currently, BPBD does not yet have any historical data related to areas affected by eruption in previous years in the form of maps. This data can be used as data for disaster mitigation and predict the large of areas that may be affected in the event of another eruption. To obtain the data, guided classification technique will be used by using Semi Automatic Classification Plug-In (SCP) from Quantum GIS application. This technique is used for mapping erupted areas in Sleman District by defining polygon training data. Then the results of the mapping will be displayed by using Web-based Geographic Information System (WebGIS), so it can provide the mapping results properly and clearly for users. WebGIS is based on architectural design that is divided into 3 layers, i.e. Client Layer, Service Layer, and Support Layer. Based on the mapping results, the large of area that affected by eruption for each year of the incident was obtained. Based on the mapping results, the large of eruption-affected areas for each year is as follows: 1992 with total area of 281.94 ha and percentage of 4.1%, in 1994 with total area of 1058.61 ha and percentage of 15.2%, in 1997 with total area of 741.11 ha and percentage of 10.7%, in 1998 with total area of 278.6 ha and percentage of 4%, in 2001 with total area of 499.97 ha and percentage of 7.2%, in 2006 with total area of 783.94 ha and percentage of 11.3%, and in 2010 with total area of 3309.55 ha and percentage of 47.6%. While the accuracy of the test results obtained accuracy is 92% and precision is 86.2%, which means that the mapping results have high accuracy and precision.

Key words: Geographic Information Systems, SCP, Architecture, Merapi Mountain, Sleman Regency

DAFTAR ISI

| | |
|--|----------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR..... | iv |
| ABSTRAK..... | vi |
| ABSTRACT | vii |
| DAFTAR ISI..... | viii |
| DAFTAR TABEL..... | xii |
| DAFTAR GAMBAR..... | xiv |
| DAFTAR LAMPIRAN | xviii |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar belakang..... | 1 |
| 1.2 Rumusan masalah..... | 3 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat | 3 |
| 1.5 Batasan masalah | 4 |
| 1.6 Sistematika pembahasan..... | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 6 |
| 2.1 Kajian Pustaka..... | 6 |
| 2.2 Kabupaten Sleman..... | 8 |
| 2.3 Gunung Merapi..... | 9 |
| 2.4 Sistem Informasi | 10 |
| 2.5 Sistem Informasi Geografis..... | 10 |
| 2.5.1 Jenis Data Sistem Informasi Geografis (SIG) | 11 |
| 2.5.2 Komponen Sistem Informasi Geografis | 13 |
| 2.6 Data Penginderaan Jarak Jauh..... | 13 |
| 2.7 Satelit Landsat..... | 14 |
| 2.8 Quantum GIS..... | 15 |
| 2.9 WebGIS | 15 |
| 2.9.1 Arsitektur WebGIS | 16 |
| 2.9.2 <i>Library Leaflet</i> | 17 |
| 2.9.3 <i>Semi-Automatic Classification Plug-in</i> | 18 |



| | |
|--|-----------|
| 2.9.4 CodeIgniter | 18 |
| 2.10 <i>Software Development Life Cycle (SDLC)</i> | 18 |
| 2.10.1 Model <i>Waterfall</i> | 19 |
| 2.11 <i>UML (Unified Modelling Language)</i> | 20 |
| 2.11.1 <i>Activity Diagram</i> | 20 |
| 2.11.2 <i>Diagram Use Case</i> | 21 |
| 2.11.3 <i>Class Diagram</i> | 22 |
| 2.11.4 <i>Sequence Diagram</i> | 24 |
| 2.12 Pengujian <i>Black-Box</i> | 25 |
| 2.13 Pengujian <i>Portability</i> | 25 |
| 2.14 Pengujian <i>Accuracy</i> | 26 |
| BAB 3 METODOLOGI PENELITIAN | 28 |
| 3.1 Pendekatan Penelitian | 28 |
| 3.2 Studi Pustaka | 28 |
| 3.3 Pengumpulan Data | 29 |
| 3.4 Analisis Kebutuhan Sistem | 29 |
| 3.5 Perancangan Sistem | 30 |
| 3.6 <i>Pre-processing Data dan Penerapan Semi-Automatic Classification Plug-in</i> | 30 |
| 3.7 Implementasi WebGIS | 31 |
| 3.8 Pengujian dan Analisa Hasil | 31 |
| 3.9 Kesimpulan | 31 |
| BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN | 32 |
| 4.1 Analisis Kebutuhan | 32 |
| 4.1.1 Gambaran Umum | 32 |
| 4.1.2 Identifikasi Aktor | 33 |
| 4.1.3 Kebutuhan Fungsional | 34 |
| 4.1.4 Kebutuhan Non-Fungsional | 35 |
| 4.1.5 <i>Diagram Use Case</i> | 36 |
| 4.1.6 <i>Use Case Scenario</i> | 36 |
| 4.2 Perancangan | 47 |
| 4.2.1 Perancangan Arsitektur | 47 |
| 4.2.2 <i>Diagram Aktivitas</i> | 48 |



| | |
|--|-----|
| 4.2.3 Class Diagram | 57 |
| 4.2.4 Sequence Diagram | 58 |
| 4.2.5 Perancangan Basis Data | 69 |
| 4.2.6 Perancangan Antarmuka..... | 71 |
| BAB 5 IMPLEMENTASI | 81 |
| 5.1 <i>Pre-processing</i> Data dan Penerapan <i>Semi-Automatic Classification Plug-in</i> (SCP)..... | 81 |
| 5.1.1 Mengunduh Data Landsat..... | 81 |
| 5.1.2 Menginstall <i>Semi Automatic Classification Plugin</i> (SCP) | 83 |
| 5.1.3 <i>Pre-Processing</i> Data Landsat..... | 84 |
| 5.1.4 Melakukan Klasifikasi Tutupan Lahan..... | 89 |
| 5.1.5 Hasil..... | 96 |
| 5.2 Implementasi WebGIS | 100 |
| 5.2.1 Implementasi Class | 100 |
| 5.2.2 Implementasi <i>Source Code</i> | 101 |
| 5.2.3 Implementasi Antarmuka | 113 |
| BAB 6 PENGUJIAN | 120 |
| 6.1 Pengujian <i>Black-Box</i> (Uji Validasi) | 120 |
| 6.1.1 Tujuan Pengujian..... | 120 |
| 6.1.2 Mekanisme Pengujian..... | 120 |
| 6.1.3 Hasil Pengujian | 132 |
| 6.1.4 Analisis Pengujian | 136 |
| 6.2 Pengujian <i>Portability</i> | 136 |
| 6.2.1 Tujuan Pengujian..... | 136 |
| 6.2.2 Mekanisme Pengujian..... | 136 |
| 6.2.3 Hasil Pengujian | 137 |
| 6.2.4 Analisis Pengujian | 141 |
| 6.3 Pengujian <i>Accuracy</i> | 141 |
| 6.3.1 Tujuan Pengujian..... | 141 |
| 6.3.2 Mekanisme Pengujian..... | 142 |
| 6.3.3 Hasil Pengujian | 142 |
| 6.3.4 Analisis Pengujian | 142 |
| BAB 7 PENUTUP | 143 |



| | |
|----------------------|-----|
| 7.1 Kesimpulan | 143 |
| 7.2 Saran | 144 |
| DAFTAR PUSTAKA..... | 145 |



DAFTAR TABEL

| | |
|--|-----|
| Tabel 2.1 Perbedaan penelitian kedua dan penelitian ini | 7 |
| Tabel 2.2 Perbedaan penelitian ketiga dan penelitian ini | 8 |
| Tabel 2.3 Simbol dan deskripsi diagram aktivitas | 20 |
| Tabel 2.4 Simbol dan deskripsi diagram <i>use case</i> | 21 |
| Tabel 2.5 Simbol dan deskripsi <i>class diagram</i> | 23 |
| Tabel 2.6 Simbol dan deskripsi <i>sequence diagram</i> | 24 |
| Tabel 2.7 Deskripsi matriks <i>confusion</i> | 26 |
| Tabel 4.1 Identifikasi dan karakteristik pengguna | 33 |
| Tabel 4.2 Kebutuhan fungsional <i>user</i> | 34 |
| Tabel 4.3 Kebutuhan fungsional Admin | 34 |
| Tabel 4.4 Kebutuhan Non-Fungsional | 35 |
| Tabel 4.5 <i>Use case scenario</i> melihat <i>history</i> erupsi | 36 |
| Tabel 4.6 <i>Use case scenario</i> melihat informasi <i>history</i> erupsi | 37 |
| Tabel 4.7 <i>Use case scenario</i> memilih <i>year layer</i> | 38 |
| Tabel 4.8 <i>Use case scenario</i> memilih <i>basemap</i> | 38 |
| Tabel 4.9 <i>Use case scenario</i> mencetak peta | 39 |
| Tabel 4.10 <i>Use case scenario</i> melihat diagram | 39 |
| Tabel 4.11 <i>Use case scenario login</i> | 40 |
| Tabel 4.12 <i>Use case scenario logout</i> | 41 |
| Tabel 4.13 <i>Use case scenario</i> melihat data <i>history</i> | 41 |
| Tabel 4.14 <i>Use case scenario</i> menambah data <i>history</i> | 42 |
| Tabel 4.15 <i>Use case scenario</i> menghapus data <i>history</i> | 43 |
| Tabel 4.16 <i>Use case scenario</i> melihat data info <i>history</i> | 44 |
| Tabel 4.17 <i>Use case scenario</i> menambah data info <i>history</i> | 44 |
| Tabel 4.18 <i>Use case scenario</i> menghapus data info <i>history</i> | 45 |
| Tabel 4.19 <i>Use case scenario</i> mengedit data info <i>history</i> | 46 |
| Tabel 4.20 Kamus data tabel <i>user</i> | 70 |
| Tabel 4.21 Kamus data tabel data | 70 |
| Tabel 4.22 Kamus data tabel info | 71 |
| Tabel 5.1 Kategori dan MC ID klasifikasi | 90 |
| Tabel 5.2 Implementasi <i>class controller</i> dan <i>model</i> | 100 |

| | |
|---|-----|
| Tabel 5.3 <i>Source code</i> untuk mengubah data JSON menjadi peta | 101 |
| Tabel 6.1 <i>Test case</i> melihat <i>history</i> erupsi | 120 |
| Tabel 6.2 <i>Test case</i> melihat informasi <i>history</i> erupsi | 121 |
| Tabel 6.3 <i>Test case</i> memilih <i>year layer</i> | 121 |
| Tabel 6.4 <i>Test case</i> memilih <i>basemap</i> | 122 |
| Tabel 6.5 <i>Test case</i> mencetak peta | 122 |
| Tabel 6.6 <i>Test case</i> melihat diagram | 123 |
| Tabel 6.7 <i>Test case login</i> [benar] | 123 |
| Tabel 6.8 <i>Test case login</i> [salah] | 124 |
| Tabel 6.9 <i>Test case logout</i> | 124 |
| Tabel 6.10 <i>Test case</i> melihat data <i>history</i> | 125 |
| Tabel 6.11 <i>Test case</i> menambah data <i>history</i> [benar] | 125 |
| Tabel 6.12 <i>Test case</i> menambah data <i>history</i> [salah] | 126 |
| Tabel 6.13 <i>Test case</i> menghapus data <i>history</i> [semua] | 127 |
| Tabel 6.14 <i>Test case</i> menghapus data <i>history</i> [baris tertentu] | 127 |
| Tabel 6.15 <i>Test case</i> melihat data info <i>history</i> | 128 |
| Tabel 6.16 <i>Test case</i> menambah data info <i>history</i> [benar] | 128 |
| Tabel 6.17 <i>Test case</i> menambah data info <i>history</i> [salah] | 129 |
| Tabel 6.18 <i>Test case</i> menghapus data info <i>history</i> [semua] | 130 |
| Tabel 6.19 <i>Test case</i> menghapus data info <i>history</i> [baris tertentu] | 130 |
| Tabel 6.20 <i>Test case</i> mengedit data info <i>history</i> [benar] | 131 |
| Tabel 6.21 <i>Test case</i> mengedit data info <i>history</i> [salah] | 132 |
| Tabel 6.22 Hasil pengujian <i>black-box</i> (uji validasi) | 133 |
| Tabel 6.23 Kasus uji pengujian <i>portability</i> | 137 |
| Tabel 6.24 Ringkasan hasil pengujian <i>portability</i> | 141 |
| Tabel 6.25 Hasil pengujian daerah terdampak erupsi tahun 2010 | 142 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 <i>Screenshot</i> WebGIS pemetaan tsunami di Itali..... | 6 |
| Gambar 2.2 Kategori tutupan lahan di Subcarpathian, Prahova..... | 7 |
| Gambar 2.3 Arsitektur WebGIS..... | 8 |
| Gambar 2.4 Peta Kabupaten Sleman | 9 |
| Gambar 2.5 Sumber data dalam SIG..... | 11 |
| Gambar 2.6 Tampilan data titik, garis, dan luasan | 12 |
| Gambar 2.7 Perbandingan data model vektor dan raster..... | 12 |
| Gambar 2.8 Komponen SIG..... | 13 |
| Gambar 2.9 Arsitektur <i>thin client</i> | 16 |
| Gambar 2.10 Arsitektur <i>thick client</i> | 16 |
| Gambar 2.11 Fungsionalitas <i>thin client</i> , <i>medium client</i> , dan <i>thick client</i> | 17 |
| Gambar 2.12 SDLC model <i>waterfall</i> | 19 |
| Gambar 2.13 Contoh <i>activity diagram</i> | 21 |
| Gambar 2.14 Contoh diagram <i>use case</i> | 22 |
| Gambar 2.15 Contoh <i>class diagram</i> | 23 |
| Gambar 2.16 Contoh <i>sequence diagram</i> | 25 |
| Gambar 2.17 Matriks <i>confusion</i> | 26 |
| Gambar 3.1 Metodologi penelitian..... | 28 |
| Gambar 4.1 Diagram <i>use case</i> sistem..... | 36 |
| Gambar 4.2 Perancangan arsitektural | 47 |
| Gambar 4.3 Diagram aktivitas melihat <i>history</i> erupsi | 48 |
| Gambar 4.4 Diagram aktivitas melihat informasi <i>history</i> erupsi | 49 |
| Gambar 4.5 Diagram aktivitas memilih <i>year layer</i> | 49 |
| Gambar 4.6 Diagram aktivitas memilih <i>basemap</i> | 50 |
| Gambar 4.7 Diagram aktivitas mencetak peta | 50 |
| Gambar 4.8 Diagram aktivitas melihat diagram | 51 |
| Gambar 4.9 Diagram aktivitas <i>login</i> | 51 |
| Gambar 4.10 Diagram aktivitas <i>logout</i> | 52 |
| Gambar 4.11 Diagram aktivitas melihat data <i>history</i> | 52 |
| Gambar 4.12 Diagram aktivitas menambah data <i>history</i> | 53 |
| Gambar 4.13 Diagram aktivitas menghapus data <i>history</i> | 54 |

| | |
|---|----|
| Gambar 4.14 Diagram aktivitas melihat data info <i>history</i> | 54 |
| Gambar 4.15 Diagram aktivitas menambah data info <i>history</i> | 55 |
| Gambar 4.16 Diagram aktivitas menghapus data info <i>history</i> | 56 |
| Gambar 4.17 Diagram aktivitas mengedit data info <i>history</i> | 57 |
| Gambar 4.18 <i>Class diagram</i> | 58 |
| Gambar 4.19 <i>Sequence diagram</i> melihat <i>history</i> erupsi | 59 |
| Gambar 4.20 <i>Sequence diagram</i> melihat informasi <i>history</i> erupsi | 59 |
| Gambar 4.21 <i>Sequence diagram</i> memilih <i>year layer</i> | 60 |
| Gambar 4.22 <i>Sequence diagram</i> memilih <i>basemap</i> | 60 |
| Gambar 4.23 <i>Sequence diagram</i> mencetak peta | 61 |
| Gambar 4.24 <i>Sequence diagram</i> melihat diagram | 61 |
| Gambar 4.25 <i>Sequence diagram</i> login | 62 |
| Gambar 4.26 <i>Sequence diagram</i> logout | 63 |
| Gambar 4.27 <i>Sequence diagram</i> melihat data <i>history</i> | 63 |
| Gambar 4.28 <i>Sequence diagram</i> menambah data <i>history</i> | 64 |
| Gambar 4.29 <i>Sequence diagram</i> menghapus data <i>history</i> | 65 |
| Gambar 4.30 <i>Sequence diagram</i> melihat data info <i>history</i> | 66 |
| Gambar 4.31 <i>Sequence diagram</i> menambah data info <i>history</i> | 67 |
| Gambar 4.32 <i>Sequence diagram</i> menghapus data info <i>history</i> | 68 |
| Gambar 4.33 <i>Sequence diagram</i> mengedit data info <i>history</i> | 69 |
| Gambar 4.34 Perancangan basis data | 70 |
| Gambar 4.35 Perancangan antarmuka halaman <i>history</i> | 72 |
| Gambar 4.36 Perancangan antarmuka halaman diagram..... | 73 |
| Gambar 4.37 Perancangan antarmuka halaman <i>basemap</i> | 74 |
| Gambar 4.38 Perancangan antarmuka halaman <i>sign in</i> | 75 |
| Gambar 4.39 Perancangan antarmuka halaman <i>Map Data</i> | 76 |
| Gambar 4.40 Perancangan antarmuka halaman form tambah data <i>history</i> | 77 |
| Gambar 4.41 Perancangan antarmuka halaman <i>Info Data</i> | 78 |
| Gambar 4.42 Perancangan antarmuka halaman <i>form</i> tambah info..... | 79 |
| Gambar 4.43 Perancangan antarmuka halaman <i>form edit</i> info | 80 |
| Gambar 5.1 Diagram alur <i>pre-processing</i> data dan penerapan SCP | 81 |
| Gambar 5.2 Langkah 1 mengundah data..... | 82 |



| | |
|---|----|
| Gambar 5.3 Langkah 2 mengunduh data..... | 82 |
| Gambar 5.4 Langkah 3 mengunduh data..... | 83 |
| Gambar 5.5 Langkah 4 mengunduh data..... | 83 |
| Gambar 5.6 Langkah 1 menginstall SCP..... | 84 |
| Gambar 5.7 Langkah 2 menginstall SCP..... | 84 |
| Gambar 5.8 Langkah 3 menginstall SCP..... | 84 |
| Gambar 5.9 Input data Landsat | 85 |
| Gambar 5.10 Input MTL..... | 85 |
| Gambar 5.11 Hasil input data Landsat dan MTL..... | 86 |
| Gambar 5.12 Hasil <i>pre-processing</i> | 86 |
| Gambar 5.13 Data SHP Kabupaten Sleman | 87 |
| Gambar 5.14 Proses pemotongan menggunakan <i>clip multiple raster</i> | 87 |
| Gambar 5.15 Hasil pemotongan daerah Kabupaten Sleman | 88 |
| Gambar 5.16 Tampilan pembuatan VRT..... | 88 |
| Gambar 5.17 Hasil VRT..... | 89 |
| Gambar 5.18 Tampilan SCP: <i>ROI Creation</i> | 89 |
| Gambar 5.19 Tampilan <i>training shapefile</i> | 90 |
| Gambar 5.20 Contoh pembuatan <i>polygon</i> | 90 |
| Gambar 5.21 Tampilan <i>ROI signature definition</i> | 91 |
| Gambar 5.22 Tampilan setelah pembuatan <i>polygon-polygon</i> | 91 |
| Gambar 5.23 Tampilan SCP: <i>Classification</i> | 92 |
| Gambar 5.24 Hasil klasifikasi | 92 |
| Gambar 5.25 Tampilan memilih daerah yang terkena erupsi | 93 |
| Gambar 5.26 Tampilan <i>save vector layer as</i> | 93 |
| Gambar 5.27 Tampilan daerah yang terkena erupsi | 94 |
| Gambar 5.28 Tampilan <i>open attribute table</i> | 94 |
| Gambar 5.29 Tampilan menu <i>dissolve</i> | 95 |
| Gambar 5.30 Tampilan <i>calculator field</i> | 95 |
| Gambar 5.31 Tampilan label luas daerah | 96 |
| Gambar 5.32 Hasil pemetaan daerah yang terdampak erupsi tahun 1992 | 97 |
| Gambar 5.33 Hasil pemetaan daerah yang terdampak erupsi tahun 1994 | 97 |
| Gambar 5.34 Hasil pemetaan daerah yang terdampak erupsi tahun 1997 | 98 |

| | |
|--|-----|
| Gambar 5.35 Hasil pemetaan daerah yang terdampak erupsi tahun 1998 | 98 |
| Gambar 5.36 Hasil pemetaan daerah yang terdampak erupsi tahun 2001 | 99 |
| Gambar 5.37 Hasil pemetaan daerah yang terdampak erupsi tahun 2006 | 99 |
| Gambar 5.38 Hasil pemetaan daerah yang terdampak erupsi tahun 2010 | 100 |
| Gambar 5.39 Antarmuka halaman <i>history</i> | 113 |
| Gambar 5.40 Antarmuka halaman diagram | 114 |
| Gambar 5.41 Antarmuka halaman <i>Basemap</i> | 115 |
| Gambar 5.42 Antarmuka halaman <i>sign in</i> | 115 |
| Gambar 5.43 Antarmuka halaman <i>map data</i> | 116 |
| Gambar 5.44 Antarmuka halaman <i>form</i> tambah data | 117 |
| Gambar 5.45 Antarmuka halaman <i>info data</i> | 118 |
| Gambar 5.46 Antarmuka halaman <i>form</i> tambah info | 119 |
| Gambar 5.47 Antarmuka halaman <i>form edit info</i> | 119 |
| Gambar 6.1 Hasil pengujian <i>Portability</i> pada halaman <i>History</i> | 138 |
| Gambar 6.2 Hasil pengujian <i>Portability</i> pada halaman diagram | 138 |
| Gambar 6.3 Hasil pengujian <i>Portability</i> pada halaman <i>basemap</i> | 138 |
| Gambar 6.4 Hasil pengujian <i>Portability</i> pada halaman <i>sign in</i> | 139 |
| Gambar 6.5 Hasil pengujian <i>Portability</i> pada halaman <i>map data</i> | 139 |
| Gambar 6.6 Hasil pengujian <i>Portability</i> pada halaman <i>form</i> tambah data..... | 140 |
| Gambar 6.7 Hasil pengujian <i>Portability</i> pada halaman <i>info data</i> | 140 |
| Gambar 6.8 Hasil pengujian <i>Portability</i> pada halaman <i>form</i> tambah info | 140 |
| Gambar 6.9 Hasil pengujian <i>Portability</i> pada halaman <i>edit info</i> | 141 |



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan negara yang masuk ke dalam jalur cincin api Pasifik yang biasanya di sebut *ring of fire* yang dimulai dari Sumatera – Jawa – Bali – Nusa Tenggara – Sulawesi – Banda- Maluku-Papua (Sutikno, et al, 1996). *Ring of fire* atau jalur cincin api adalah daerah yang sering mengalami gempa bumi dan letusan gunung berapi yang mengelilingi cekung Samudra Pasifik (Bemmelen, 1949). Salah satu gunung berapi yang aktif di Indonesia adalah Gunung Merapi yang terletak di perbatasan dua provinsi yaitu Provinsi D.I. Yogyakarta dan provinsi Jawa Tengah. Gunung Merapi adalah gunung api strato dengan kubah lava, memiliki elevasi ± 2.911 m dpl dan lebar ± 30 km (Katili dan Siswowidjojo, 1994). Gunung Merapi merupakan gunung api dengan erupsi paling aktif di Indonesia sehingga hal ini mendorong pemerintah dan masyarakat untuk memberikan perhatian khusus. Berdasarkan catatan sejarah Gunung Merapi, erupsi dapat diketahui berdasarkan umur batuan yang berasal dari endapan hasil erupsi, awan panas, dan endapan lahar di bagian utara, selatan dan barat (Widiyanto dan A. Rahman, 2008). Sejak abad XVI hingga abad XX erupsi Gunung Merapi mengalami perubahan waktu istirahat dari 71 tahun menjadi 8 tahun dengan jumlah kegiatan erupsi dari 7 kali menjadi 28 kali (Widiyanto dan A. Rahman, 2008). Aktivitas letusan Gunung Merapi 12 Oktober – 5 November 2010 merupakan letusan dengan erupsi yang cukup besar dibandingkan erupsi yang terjadi pada tahun 1870 (Badan Litbang Pertanian, 2011).

Sebagai objek penelitian, Kabupaten Sleman merupakan salah satu daerah yang terkena dampak letusan Gunung Merapi paling besar. Saat ini, kondisi alam di daerah kawasan Gunung Merapi berubah dengan cepat. Beberapa daerah di Sleman kini telah berubah menjadi dataran yang dipenuhi dengan debu – debu akibat letusan merapi (Wilson, et al., 2007). Menurut hasil wawancara yang dilakukan dengan Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Sleman, Gunung Merapi diperkirakan dapat meletus kapan saja dalam jangka waktu dekat. Hal tersebut dikarenakan aktifitas lava yang masih tinggi dari Gunung Merapi dan lava tersebut seiring bertambahnya waktu semakin mencapai permukaan. Selain itu, pihak lapangan BPBD Kabupaten Sleman telah melakukan pengukuran yang dilakukan dengan menjatuhkan batu ke dalam pintu lava dan menghitung waktu jatuh batu tersebut. Dari hasil pengukuran tersebut, lava sudah sangat dekat dengan permukaan. Pipa-pipa lava yang digunakan sebagai jalan keluar lava dan uap panas yang mencegah terjadinya erupsi, saat ini telah tertutup oleh lava yang mengeras. Keadaan-keadaan tersebut mendukung Gunung Merapi untuk meletus kembali dalam jangka waktu dekat. Saat ini BPBD belum memiliki data *history* terkait daerah-daerah yang terkena erupsi pada tahun-tahun kejadian sebelumnya dalam bentuk peta. Data tersebut dapat digunakan sebagai data dalam melakukan mitigasi bencana dan memprediksi luasan daerah yang mungkin akan terkena dampak apabila terjadi erupsi kembali (BPBD, 2017).

Penelitian ini mengacu pada tiga penelitian sebelumnya, penelitian pertama adalah penelitian yang dilakukan Fago et al (2014) di Itali menunjukkan bahwa kemajuan Sistem Informasi Geografis (SIG) dapat digunakan untuk melakukan pemetaan suatu lokasi atau daerah yang rawan terjadinya tsunami apabila terjadi gempa tektonik. Dari hasil pemetaan diketahui bahwa tsunami yang melanda Itali pada tahun 2013 telah melewati daerah pesisir selatan Italia (Fago, et al., 2014).

Penelitian kedua adalah yang dilakukan oleh Marina-Ramona dan Bogdan-Andrei (2016) tentang pemetaan tutupan lahan di daerah Subcarpathian, Prahova yang datanya didapatkan dari data penginderaan jarak jauh yaitu menggunakan Landsat 8 OLI. Untuk mendapatkan hasil kategori tutupan lahan, data tersebut akan diklasifikasi menggunakan teknik klasifikasi terbimbing dengan mendefinisikan satu set data latih dalam bentuk *polygon* yang terbagi dalam kategori hutan konifera, hutan gugur, padang rumput, lahan pertanian, area bangunan dan daerah yang tertutup air. Untuk menampilkan hasil klasifikasi digunakan 2 algoritma yaitu Algoritma *Minimum Distance* dan Algoritma *Maximum Likelihood*. Dari hasil klasifikasi, didapatkan beberapa kategori tutupan lahan antara lain hutan konifera, hutan gugur, rerumputan, kebun buah-buahan, vegetasi alami, tanaman pertanian, area bangunan, dasar sungai/dataran banjir dan badan air (Marina-Ramona dan Bogdan-Andrei, 2016).

Berdasarkan kedua penelitian tersebut, teknik klasifikasi terbimbing dapat digunakan untuk mendapatkan kategori tutupan lahan dari suatu daerah. Sehingga teknik ini dapat juga digunakan untuk memetakan daerah-daerah yang terkena erupsi di Kabupaten Sleman dengan mendefinisikan data latih berupa *polygon* dengan kategori tanah erupsi. Kemudian data daerah-daerah yang terkena erupsi di Kabupaten Sleman dapat ditampilkan dengan menggunakan Sistem Informasi Geografis berbasis web (WebGIS).

Penelitian ketiga yang dilakukan oleh Huang et al (2015) tentang pembuatan WebGIS untuk pemantauan longsor di daerah pegunungan Barat China, dengan menganalisis ambang batas curah hujan serta menunjukkan kriteria awal secara otomatis dan terus-menerus yang akan memberikan peringatan apabila akan terjadi longsor. WebGIS tersebut terdiri dari 3 *layer*, yaitu *Client Layer*, *Service Layer*, dan *Support Layer*. Setiap *layer* tersebut masing-masing memiliki fungsi dan peran yang saling terkait dan terhubung (Huang, et al., 2015).

Berdasarkan penelitian ketiga, WebGIS yang dibuat akan mengacu pada arsitektur yang terdiri dari tiga *layer* yaitu *Client Layer*, *Service Layer*, dan *Support Layer*. Ketiga *layer* tersebut dapat memberikan batasan yang jelas pada komponen-komponen yang ada dan digunakan pada WebGIS.

Dari latar belakang yang telah di uraikan diatas, kemudian penulis tertarik untuk mengkaji masalah tentang pemetaan daerah yang terkena erupsi Gunung Merapi di Kabupaten Sleman dengan menggabungkan konsep dari ketiga penelitian sebelumnya, dengan mengambil judul: "PEMBANGUNAN ARSITEKTUR

WEBGIS UNTUK PEMETAAN DAERAH YANG TERDAMPAK ERUPSI GUNUNG MERAPI (STUDI KASUS: KABUPATEN SLEMAN)".

1.2 Rumusan masalah

Terdapat beberapa rumusan masalah yang diperoleh berdasarkan latar belakang di atas, yaitu sebagai berikut :

1. Bagaimana luasan daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman selama periode penelitian?
2. Bagaimana arsitektur WebGIS yang digunakan untuk menampilkan hasil pemetaan daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman?
3. Bagaimana akurasi dari hasil analisis geo-spasial yang dilakukan?

1.3 Tujuan

Berdasarkan rumusan masalah diatas, dapat diperoleh tujuan dari penelitian antara lain sebagai berikut :

1. Mengetahui luasan daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman selama periode penelitian.
2. Membangun arsitektur WebGIS yang digunakan untuk menampilkan hasil pemetaan daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman.
3. Mengetahui akurasi dari hasil analisis geo-spasial yang dilakukan.

1.4 Manfaat

Apabila penelitian ini telah selesai dilakukan, maka diharapkan dapat memberikan manfaat untuk berbagai pihak sebagai berikut :

1. Bagi Jurusan Sistem Informasi
Hasil dari penelitian dapat menjadi salah satu sumber data dan sumber ide bagi penelitian selanjutnya.
2. Bagi Penulis
Menambah wawasan dan meningkatkan pemahaman konsep, serta meningkatkan pengaplikasian dari teori geo-informasi.
3. Bagi Pemerintah dan BPBD Kabupaten Sleman
Dapat digunakan sebagai data untuk melakukan mitigasi bencana erupsi Gunung Merapi dan memprediksi luasan daerah yang kemungkinan dapat terkena dampak di Kabupaten Sleman apabila kembali terjadi erupsi.

1.5 Batasan masalah

Terdapat beberapa batasan masalah dalam penelitian ini, sehingga tidak akan menyimpang dari permasalahan yang ada. Batasan-batasan masalah tersebut adalah sebagai berikut :

1. Kawasan yang dipetakan difokuskan pada daerah – daerah di Kabupaten Sleman yang terkena erupsi Gunung Merapi.
2. Data yang diambil merupakan data dengan rentan waktu 30 tahun dimulai dari tahun 1985 hingga tahun 2015, disesuaikan dengan waktu terjadinya erupsi Gunung Merapi.
3. Data yang digunakan adalah data yang tersedia di earthexplorer.usgs.gov dan dapat di olah.
4. Data satelit yang digunakan adalah Landsat 4-5 TM dan Landsat 7 ETM.
5. WebGIS yang dibuat hanya menampilkan data history dari dampak erupsi Gunung Merapi bagi Kabupaten Sleman.

1.6 Sistematika pembahasan

Terdapat 7 bab dalam sistematika pembahasan dari penelitian ini, yaitu sebagai berikut :

BAB I Pendahuluan

Pada bab ini terdapat 6 hal yang akan dijelaskan yaitu latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB II Landasan Kepustakaan

Bab ini bertujuan untuk menjelaskan seluruh teori-teori dan referensi yang digunakan dan berkaitan dalam melakukan penelitian.

BAB III Metodologi Penelitian

Pada bab ini berisi langkah-langkah yang akan dilakukan dalam penelitian, yaitu studi literatur, pengumpulan data, analisis kebutuhan sistem, perancangan sistem, pemrosesan data, implementasi WebGIS, pengujian dan analisa hasil, dan kesimpulan.

BAB IV Analisis Kebutuhan dan Perancangan

Pada bab ini menjelaskan proses analisis dan perancangan dari sistem/perangkat lunak yang akan dibuat.

BAB V Implementasi

Pada bab ini menjelaskan tentang hasil implementasi dari analisis kebutuhan dan perancangan yang telah dibuat. Serta menguraikan

bagaimana proses dalam pengolahan data dan klasifikasi yang digunakan untuk mendapatkan daerah yang terkena erupsi Gunung Merapi bagi Kabupaten Sleman. Selain itu, juga akan ditunjukkan hasil dari proses pengolahan data dan klasifikasi yang dilakukan.

BAB VI Pengujian

Pada bab ini berisi tentang dokumentasi dari pengujian perangkat lunak yang dilakukan. Terdapat 3 jenis pengujian yang akan dilakukan, yaitu pengujian *black-box* (uji validasi), pengujian *portability*, dan pengujian *accuracy*.

BAB VII Penutup

Pada bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini berdasarkan pada tiga penelitian sebelumnya. Penelitian yang pertama adalah penelitian yang dilakukan Fago et al (2014) di Italia. Dari penelitian menunjukkan bahwa kemajuan Sistem Informasi Geografis (SIG) dapat digunakan untuk melakukan pemetaan suatu lokasi atau daerah yang rawan terjadinya tsunami apabila terjadi gempa tektonik. Dari hasil pemetaan diketahui bahwa tsunami yang melanda Itali pada tahun 2013 telah melewati daerah pesisir selatan Italia yang ditunjukkan pada Gambar 2.1 berikut (Fago, et al., 2014).



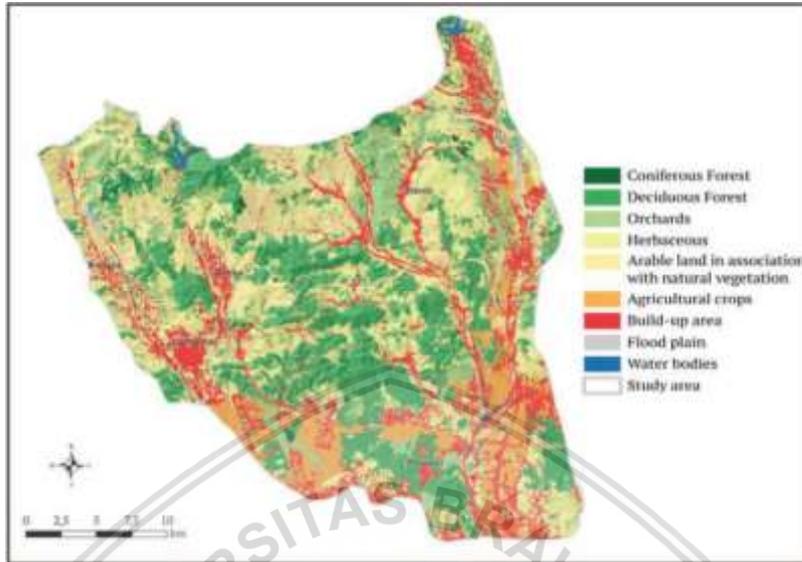
Gambar 2.1 Screenshot WebGIS pemetaan tsunami di Itali

Sumber: Fago, et al (2014)

Perbedaan antara penelitian pertama dengan penelitian ini adalah subjek dari daerah yang akan dipetakan dan jenis bencana pada daerah yang akan dipetakan. Pada penelitian ini daerah yang akan dipetakan adalah daerah yang terkena erupsi Gunung Merapi di Kabupaten Sleman dari tahun-tahun kejadian erupsi sebelumnya.

Penelitian yang kedua adalah penelitian yang dilakukan oleh Marina-Ramona dan Bogdan-Andrei (2016) tentang pemetaan tutupan lahan di daerah Subcarpathian, Prahova yang datanya didapatkan dari data penginderaan jarak jauh yaitu menggunakan Landsat 8 OLI. Untuk mendapatkan hasil kategori tutupan lahan, data tersebut akan diklasifikasi menggunakan teknik klasifikasi terbimbing dengan mendefinisikan satu set data latih dalam bentuk *polygon* yang terbagi dalam kategori hutan konifera, hutan gugur, padang rumput, lahan pertanian, area bangunan dan daerah yang tertutup air. Untuk menampilkan hasil klasifikasi digunakan 2 algoritma yaitu Algoritma *Minimum Distance* dan Algoritma *Maximum Likelihood*. Dari hasil klasifikasi, didapatkan beberapa kategori tutupan lahan antara lain hutan konifera, hutan gugur, rerumputan,

kebun buah-buahan, vegetasi alami, tanaman pertanian, area bangunan, dasar sungai/dataran banjir dan badan air yang dapat dilihat pada Gambar 2.2 berikut (Marina-Ramona dan Bogdan-Andrei, 2016).



Gambar 2.2 Kategori tutupan lahan di Subcarpathian, Prahova

Sumber: Marina-Ramona dan Bogdan-Andrei (2016)

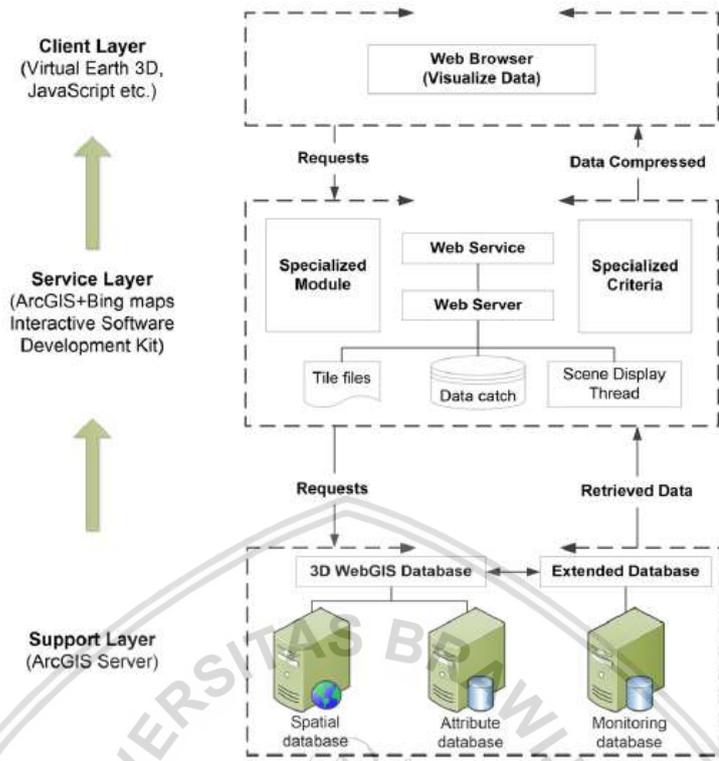
Perbedaan penelitian yang kedua dengan penelitian ini dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Perbedaan penelitian kedua dan penelitian ini

| No. | Perbedaan | Penelitian Sebelumnya | Penelitian Ini |
|-----|-----------------------|---|---|
| 1 | Data yang digunakan | Landsat 8 OLI | Landsat 4-5 TM dan Landsat 7 ETM. |
| 2 | Algoritma klasifikasi | Minimum Distance dan Maximum Likelihood | Maximum Likelihood dan Spectral Angle Mapping |

Penelitian yang ketiga dilakukan oleh Huang et al (2015) tentang pembuatan Web GIS untuk pemantauan longsor di daerah pegunungan Barat China, dengan menganalisis ambang batas curah hujan serta menunjukkan kriteria awal secara otomatis dan terus-menerus yang akan memberikan peringatan apabila akan terjadi longsor. WebGIS tersebut terdiri dari 3 *layer*, yaitu *Client Layer*, *Service Layer*, dan *Support Layer*. Setiap *layer* tersebut masing-masing memiliki fungsi dan peran yang saling terkait dan terhubung (Huang, et al., 2015).





Gambar 2.3 Arsitektur WebGIS
 Sumber: Huang, et al (2015)

Perbedaan antara penelitian yang ketiga dengan penelitian ini dapat dilihat pada Tabel 2.2 berikut.

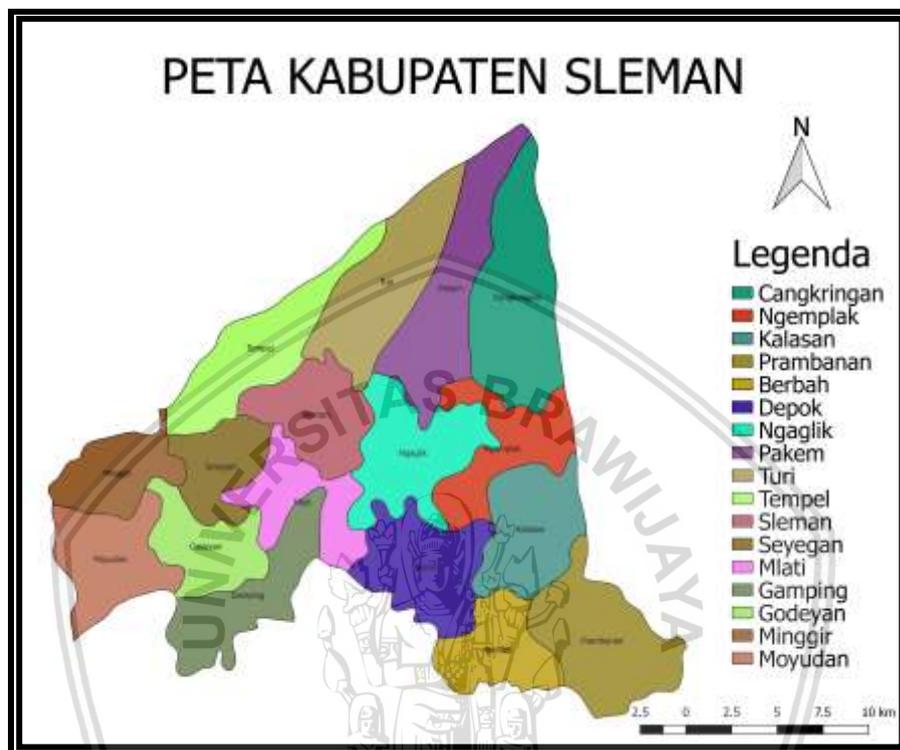
Tabel 2.2 Perbedaan penelitian ketiga dan penelitian ini

| Layer | Penelitian sebelumnya | Penelitian ini |
|---------------|---|--|
| Support Layer | ArcGIS Server | MySQL |
| Service Layer | ArcGIS + Bing maps Interactive Software Development Kit | Apache Web Server dengan menggunakan framework CI (CodeIgniter) dan PHP |
| Client Layer | Web Browser | Web Browser yang didukung oleh js, HTML, CSS, library leaflet, open layers, bootstrap, OSM |

2.2 Kabupaten Sleman

Kabupaten Sleman adalah bagian dari Daerah Istimewa Yogyakarta (DIY). Secara geografis, Kabupaten Sleman berada di bagian utara DIY, yang berbatasan dengan wilayah Kabupaten Bantul dan Kota Yogyakarta di bagian selatan, Kabupaten Klaten di bagian timur, Kabupaten Boyolali di bagian utara, dan Kabupaten Magelang serta Kabupaten Kulonprogo di bagian barat. Secara

astronomis, Kabupaten Sleman terletak antara $7^{\circ}.34'$ - $7^{\circ}.47'$ Lintang Selatan dan $110^{\circ}.13'$ - $110^{\circ}.33'$ Bujur Timur. Gunung Merapi terletak di bagian utara Kabupaten Sleman, yang merupakan salah satu gunung teraktif di dunia. Jarak linier Kabupaten Sleman menuju Ibukota Propinsi DIY sekitar 9 km (BPS, 2015). Berikut pada Gambar 2.4 merupakan peta dari Kabupaten Sleman beserta kecamatan-kecamatan yang terdapat di Kabupaten Sleman.



Gambar 2.4 Peta Kabupaten Sleman

Kabupaten Sleman mempunyai wilayah terluas ketiga setelah Gunung kidul dan Kulonprogo yaitu luas $574,82 \text{ km}^2$. Luas Kabupaten Sleman sekitar 18,04 persendari luas seluruh wilayah DIY (BPS, 2015).

2.3 Gunung Merapi

Gunung Merapi merupakan gunung berapi aktif yang terletak diantara Daerah Istimewa Yogyakarta dan Jawa Tengah. Gunung Merapi terbentuk karena aktivitas di zona subduksi Lempeng Indo-Australia yang bergerak di bawah Lempeng Eurasia. Gunung Merapi mengalami erupsi setiap dua sampai lima tahun sekali. Sejak tahun 1548, Gunung Merapi sudah meletus sebanyak 68 kali (BPPTK, 2000). Berikut adalah waktu-waktu kejadian erupsi Gunung Merapi dalam kurun waktu 30 tahun terakhir (Voight, et al., 1979) :

1. 2 Februari 1992
2. 22 November 1994
3. 14, 17 Januari 1997
4. 11, 19 Juli 1998

5. 10 Februari 2001
6. 8 Juni 2006
7. 5 November 2010

2.4 Sistem Informasi

Sistem informasi adalah sekumpulan data yang telah dipilih dan terorganisir disertai tatacara penggunaan yang cakupannya lebih jauh dari pada sekedar penyajian. Keberhasilan dari sistem informasi dengan mengukur berdasarkan maksud dari pembuatnya bergantung pada tiga faktor utama, yaitu : keserasian dan kualitas data, pengorganisasian data, dan tata cara penggunaan. Sistem Informasi memiliki struktur dan cara kerja yang berbeda-beda untuk memenuhi suatu permintaan pengguna, yang tergantung pada jenis keperluan dan permintaan yang harus dipenuhi. Data yang terdapat pada sebuah sistem informasi berasal dari berbagai sumber dan dengan berbagai jenis data. Sehingga agar bisa menggabungkan data-data tersebut, sistem informasi harus mengubah atau mentransformasi data menjadi data yang tergabung (*compatible*) (Al Fatta, 2009).

Dari pengertian sistem informasi diatas, dapat disimpulkan bahwa sistem informasi merupakan sebuah sarana yang berisikan kerangka kerja untuk pemrosesan sekumpulan data inputan yang nantinya akan menghasilkan sebuah output berupa informasi yang berguna untuk mencapai sasaran perusahaan dengan menggunakan tenaga yang berasal dari manusia, komputer, dan teknologi informasi.

2.5 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau biasa disebut *Geographic Information System* (GIS) merupakan sebuah sistem yang mampu mengelola data spasial dan dapat mengintergerasikan data spasial dengan karakteristik non spasial ataupun dengan atribut data yang mampu mendukung pengambilan keputusan. Pengembangan SIG terdiri dari metodologi dan teknologi yaitu berupa perangkat keras, perangkat lunak, data spasial dan struktur organisasi. SIG di gunakan untuk mengelola, menganalisis dan memetakan informasi spasial dan data atributnya dengan akurasi kartografi yang merupakan perpaduan antara perangkat keras dan perangkat lunak (Prahasta, 2009). Sedangkan aplikasi SIG adalah alat yang mambantu pengguna untuk melakukan analisis informasi spasial, *query* interaktif, mengedit data dalam peta, dan mempresentasikan hasil dari semua operasi tersebut (Maliene, et al., 2011).

Pengembangan SIG telah dimulai sejak tahun 1950-an, namun perangkat lunak SIG baru muncul sekitar tahun 1970-an dari laboratorium ESRI. Perkembangan SIG ditandai dengan kemampuannya untuk menggabungkan data dari berbagai sumber, evaluasi keruangan terhadap pengaruh pembangunan, dan analisa kecenderungan dengan interval waktu yang panjang. Sebelum mengimplementasikan SIG, sebaiknya perlu dipertimbangkan apa tujuan dan

sasarannya, baik untuk jangka waktu pendek atau jangka waktu panjang (Indarto, 2013).

2.5.1 Jenis Data Sistem Informasi Geografis (SIG)

Pada dasarnya, data geografis disusun oleh 2 komponen penting yaitu data spasial dan data atribut. Perbedaan antara dua jenis data tersebut adalah sebagai berikut (Ekadinata, et al., 2008):

1. Data Spasial,

Data spasial merupakan data yang bereferensi geografis atas representasi objek yang ada di bumi. Pada umumnya, data spasial berdasarkan peta yang berisikan interpretasi dan proyeksi seluruh fenomena yang berada di bumi. Berdasarkan perkembangan saat ini, sebuah peta tidak hanya merepresentasikan objek-objek yang terdapat di muka bumi, namun juga merepresentasikan objek yang terdapat di atas muka bumi yaitu di udara dan yang terdapat di bawah muka bumi.

Sumber data spasial berasal dari berbagai sumber dalam berbagai format. Sumber data spasial dapat dilihat pada Gambar 2.5, antara lain meliputi data grafis peta analog, foto udara, citra satelit, survei lapangan, pengukuran theodolit, pengukuran dengan menggunakan *Global Positioning Systems* (GPS) dan lain-lain.



Gambar 2.5 Sumber data dalam SIG

Sumber: Ekadinata, et al (2008)

Data spasial memiliki dua macam penyajian, yaitu:

a. Model vektor

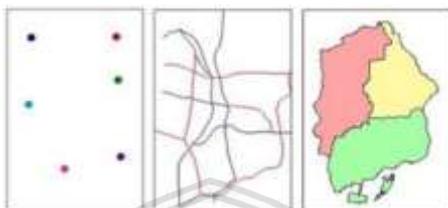
Data dalam model vektor akan merepresentasikan, menempatkan, serta menyimpan data spasial dalam tiga bentuk dasar yaitu titik, garis, dan poligon seperti yang ditunjukkan pada Gambar 2.6. Bentuk dasar dari model vektor didefinisikan dengan menggunakan sistem koordinat Kartesius dua dimensi (x,y).

Masing-masing bentuk dasar pada model vektor mewakili tipe objek tertentu, yaitu sebagai berikut :

1. Titik (*point*) : digunakan untuk merepresentasikan objek spasial yang tidak memiliki dimensi panjang dan/atau luas. Fitur spasial

direpresentasikan dalam satu pasangan koordinat x,y. Contohnya stasiun curah hujan, titik ketinggian, observasi lapangan, titik-titik sampel.

2. *Garis (line/segment)* : merepresentasikan objek yang memiliki dimensi panjang namun tidak mempunyai dimensi area, misalnya jaringan jalan, pola aliran, garis kontur.
3. *Poligon* : merepresentasikan fitur spasial yang memiliki area, contohnya adalah unit administrasi, unit tanah, zona penggunaan lahan.

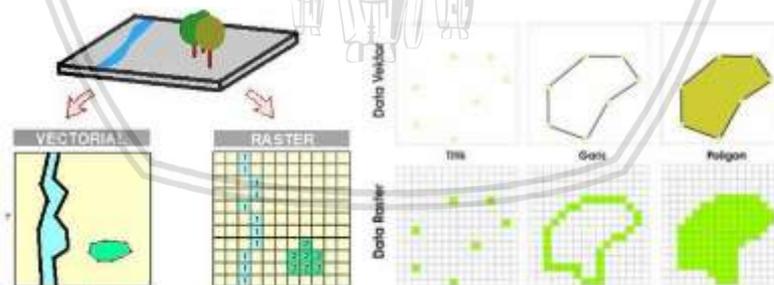


Gambar 2.6 Tampilan data titik, garis, dan luasan

Sumber: Ekadinata, et al (2008)

b. Model raster

Data dalam model raster akan merepresentasikan, menempatkan, dan menyimpan data spasial dengan struktur matriks atau piksel-piksel yang membentuk grid (bidang referensi horizontal dan vertikal yang terbagi menjadi kotak-kotak). Piksel merupakan unit dasar yang digunakan untuk menyimpan informasi secara eksplisit. Setiap piksel memiliki atribut tersendiri, termasuk koordinatnya yang unik. Akurasi data dari model raster sangat tergantung pada resolusi atau ukuran piksel suatu gambar. Semakin kecil ukuran piksel suatu gambar, maka akurasi dari data akan semakin besar. Berikut pada Gambar 2.7 merupakan perbandingan untuk data vektor dan data raster.



Gambar 2.7 Perbandingan data model vektor dan raster

Sumber: Ekadinata, et al (2008)

2. Data Atribut

Data atribut merupakan data yang menjelaskan karakteristik atau fenomena yang dikandung dari suatu objek data yang terdapat di peta. Data atribut tidak memiliki hubungan dengan posisi geografi dari suatu lokasi. Yang termasuk data atribut dapat berupa informasi numerik, foto, narasi, dan lain sebagainya, yang diperoleh dari data statistik, pengukuran lapangan dan sensus, dan lain-lain.

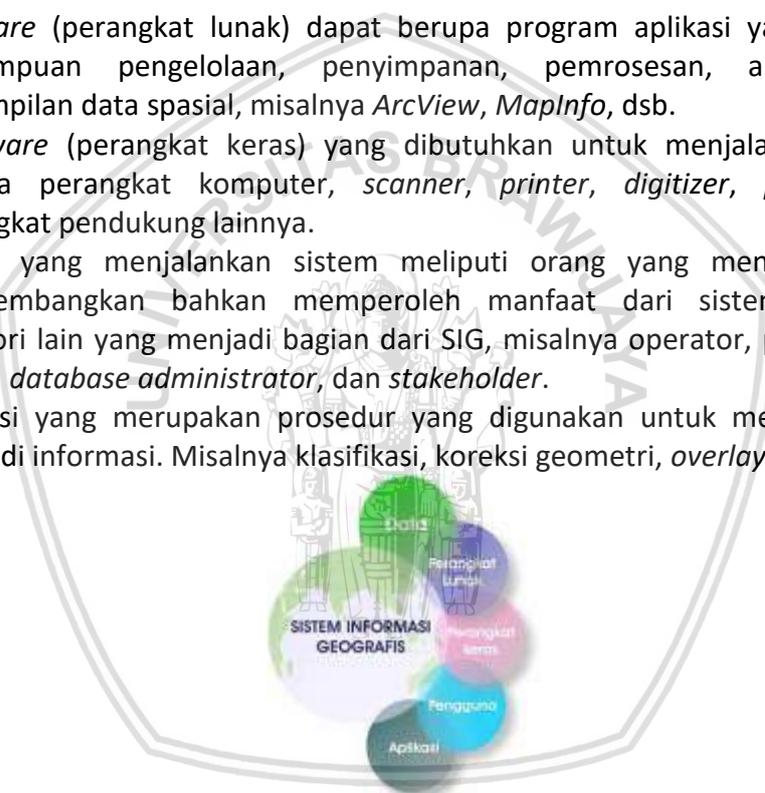


Atribut dari data dapat dideskripsikan secara kualitatif ataupun kuantitatif. Secara kualitatif, atribut akan mendeskripsikan tipe, klasifikasi, label suatu objek agar dapat dikenal dan dibedakan dengan objek lain. Sedangkan secara kuantitatif, atribut dari data mendeskripsikan suatu nilai dengan skala ordinat atau tingkatan, interval atau selang, dan rasio atau perbandingan dari suatu titik tertentu.

2.5.2 Komponen Sistem Informasi Geografis

Sistem Informasi Geografis terdiri dari beberapa komponen yang dapat dilihat pada Gambar 2.8, antara lain (John, et al., 2003) :

1. Data yang digunakan dalam Sistem Informasi Geografis, dapat berupa data grafis dan data atribut.
2. *Software* (perangkat lunak) dapat berupa program aplikasi yang memiliki kemampuan pengelolaan, penyimpanan, pemrosesan, analisis, dan penampilan data spasial, misalnya *ArcView*, *MapInfo*, dsb.
3. *Hardware* (perangkat keras) yang dibutuhkan untuk menjalankan sistem berupa perangkat komputer, *scanner*, *printer*, *digitizer*, *plotter*, dan perangkat pendukung lainnya.
4. Orang yang menjalankan sistem meliputi orang yang mengoperasikan, mengembangkan bahkan memperoleh manfaat dari sistem. Terdapat kategori lain yang menjadi bagian dari SIG, misalnya operator, programmer, analis, *database administrator*, dan *stakeholder*.
5. Aplikasi yang merupakan prosedur yang digunakan untuk mengolah data menjadi informasi. Misalnya klasifikasi, koreksi geometri, *overlay*, *query*, dsb.



Gambar 2.8 Komponen SIG

Sumber: Ekadinata, et al (2008)

2.6 Data Penginderaan Jarak Jauh

Penginderaan jarak jauh atau biasa disebut *remote sensing* adalah ilmu untuk memperoleh informasi tentang suatu obyek, fenomena, atau luasan yang dilakukan melalui analisa data yang diperoleh dari sensor. Dimana sensor berada di jarak yang jauh atau tanpa menyentuh obyek yang menjadi target (Indarto, 2013).

Dibandingkan dengan data konvensional, data penginderaan jarak jauh memiliki banyak kelebihan, namun juga memiliki kekurangan. Salah satu

kekurangan data penginderaan jarak jauh adalah bahwa data-data tersebut tidak diukur secara langsung. Data tersebut diperoleh dari hasil interpretasi spektrum gelombang elektromagnetik. Sehingga untuk menggunakan data penginderaan jarak jauh ke dalam suatu bidang permasalahan tertentu, data tersebut harus ditransformasikan ke informasi yang relevan. Hal ini membutuhkan pengembangan dan aplikasi yang lebih lanjut mengenai metodologi dan algoritme yang tepat (Indarto, 2013).

2.7 Satelit Landsat

Satelit Landsat merupakan salah satu satelit yang digunakan untuk mengamati permukaan bumi selain itu juga bertujuan memantau sumber daya lahan yang dikembangkan oleh NASA dan Departemen Dalam Negeri Amerika Serikat. Satelit Landsat sebagai Satelit sumber daya alam berfungsi dalam memetakan potensi sumber daya alam dan memantau kondisi lingkungan. Instrumen satelit-satelit Landsat telah menghasilkan jutaan citra. Citra Landsat merupakan citra yang paling banyak digunakan dalam pembuatan peta tutupan lahan, karena resolusi spasial dari citra Landsat cukup baik (30 m), dan ditambah dengan kombinasi sensor radiometrik yang cukup tinggi (mempunyai 9 kanal). Resolusi tempotal dari Landsat adalah 16 hari, karena jangka waktu pengoperasian cukup lama sehingga Landsat memiliki kelengkapan data historis amat baik (Ekadinata, et al., 2008).

Satelit Landsat senantiasa berkembang di tiap generasi. Pada penelitian monitoring dampak erupsi, peneliti menggunakan Landsat 4-5 TM dan Landsat 7 ETM+.

1. Landsat 4

Landsat 4 merupakan satelit keempat dari program Landsat. Satelit ini diluncurkan pada tanggal 16 Juli 1982, dengan tujuan utama menyediakan arsip global citra satelit. Meskipun Program Landsat dikelola oleh NASA, data dari Landsat 4 dikumpulkan dan didistribusikan oleh USGS. Landsat 4 operasi ilmu yang berakhir pada 14 Desember 1993 ketika satelit kehilangan kemampuannya untuk mengirimkan data ilmu pengetahuan, jauh melampaui harapan hidup yang dirancang dari lima tahun. Rumah tangga satelit telemetri dan pelacakan terus dipertahankan oleh NASA itu sampai dinonaktifkan pada tanggal 15 Juni 2001 (USGS, 2016).

2. Landsat 5

Landsat 5 adalah satelit kelima dari program Landsat. Satelit ini diluncurkan pada tanggal 1 Maret 1984 dengan tujuan utama menyediakan arsip global foto satelit. Program Landsat dikelola oleh USGS dan data dari Landsat 5 dikumpulkan serta didistribusikan dari USGS's Center untuk Earth Resources Observation and Science. Pada tanggal 2 Maret 2009, Landsat 5 merayakan 25 tahun keberhasilannya beroperasi. Landsat 5 telah melampaui harapan sejak pertama kali dirancang. Satelit ini memiliki bandwidth transmisi maksimum sebesar 85 Mbit/s dan ditempatkan pada ketinggian

705,3 km (438,3 mil). Dibutuhkan sekitar 16 hari untuk memindai seluruh bumi. Satelit ini adalah salinan identik dari Landsat 4 dan pada awalnya dimaksudkan sebagai backup Landsat 4 karena membawa instrumen yang sama, termasuk instrumen TM dan MSS. Instrumen MSS ini dimatikan pada tahun 1995 (Riebeek, 2010).

3. Landsat 7

Landsat 7 adalah satelit paling akhir dari program Landsat. Satelit ini diluncurkan pada tanggal 15 April 1999. Tujuan utama Landsat 7 adalah untuk memperbaharui arsip citra satelit, menyediakan citra yang update dan bebas awan. Meski program Landsat dikelola oleh NASA, data dari Landsat 7 dikumpulkan dan didistribusikan oleh USGS. Proyek NASA World Wind memungkinkan gambar tiga dimensi dari Landsat 7 dan sumber-sumber lainnya untuk dapat dengan mudah dinavigasi dan dilihat dari berbagai sudut. Landsat 7 dirancang untuk dapat bertahan 5 tahun dan memiliki kapasitas untuk mengumpulkan dan mentransmisikan hingga 532 citra setiap harinya. Orbit dari satelit ini adalah polar, orbit yang sinkron terhadap matahari, dalam arti dapat memindai seluruh permukaan bumi, yakni selama 232 orbit atau 15 hari. Massa satelit tersebut 1973 kg, memiliki panjang 4,04 meter dan diameter 2,74 meter. Tak seperti pendahulunya, Landsat 7 memiliki memori 378 gigabits (kira-kira 100 citra). Instrumen utama Landsat 7 adalah Enhanced Thematic Mapper Plus (ETM+) (USGS, 2016).

2.8 Quantum GIS

Quantum GIS (QGIS) merupakan sebuah aplikasi perangkat lunak gratis dan *open source* dengan berbasis *desktop* yang menyediakan tampilan data, mengedit data, serta analisis data geografis. Aplikasi QGIS adalah aplikasi *multiplatform* yang dapat berjalan pada berbagai sistem operasi antara lain Linux, Unix, Mac OSX, Windows dan Android. Selain itu, aplikasi QGIS juga mendukung banyak format data yaitu vektor, raster, format *database* dan fungsionalitas (QGIS, 2015).

Aplikasi QGIS dapat melakukan berbagai proses pengolahan data spasial ataupun data non-spasial, dengan menyediakan berbagai fitur yang tidak dapat dilakukan oleh perangkat lunak sejenis lainnya. Aplikasi QGIS memiliki fitur-fitur yang diantaranya dapat melakukan proses georeferensing, menghitung luasan suatu daerah, membuat peta tematik, dan proses pengolahan pemetaan lainnya (OSGeo, 2011).

2.9 WebGIS

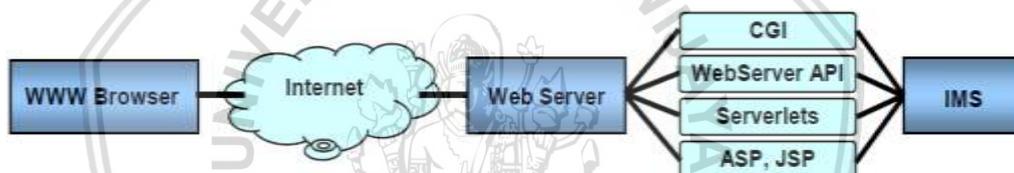
WebGIS merupakan suatu sistem yang dapat terhubung kedalam jaringan internet yang dapat digunakan untuk mengumpulkan, menyimpan, serta menampilkan data informasi bergeoreferensi atau data yang mengidentifikasi lokasi objek (Painho, 2001).

2.9.1 Arsitektur WebGIS

Dalam melakukan tugas analisis data geografis, WebGIS memiliki arsitektur *three-tier* dengan klien atau *server*. Biasanya pada sisi klien terdapat *web browser*, dan pada sisi *server* terdapat *web server*, *software* WebGIS dan basis data (Helali, 2001). Untuk mempermudah dalam komunikasi antara klien dan *server*, maka terdapat antarmuka (*user interface*) pada sisi klien dan terdapat *middleware* pada sisi *server* (Alesheikh, et al., 2002). Terdapat beberapa jenis arsitektur WebGIS, yaitu sebagai berikut (Alesheikh, et al., 2002) :

1. *Thin Client Architecture* (Aplikasi Pada Sisi Server)

Pada tipe arsitektur ini, klien hanya memiliki *user interface* untuk berkomunikasi dengan *server* dan untuk menampilkan hasil. Keseluruhan dari proses akan dilakukan oleh *server*, dimana sisi *server* biasanya memiliki kekuatan lebih dibandingkan sisi klien untuk mengelola sumber daya secara terpusat. Selain itu, fungsional utama dari arsitektur ini adalah pada sisi *server* juga terdapat kemungkinan bahwa program-program utilitas yang ada dapat dihubungkan dengan perangkat lunak *server*. Berikut pada Gambar 2.9 merupakan penggambaran arsitektur *Thin Client*.

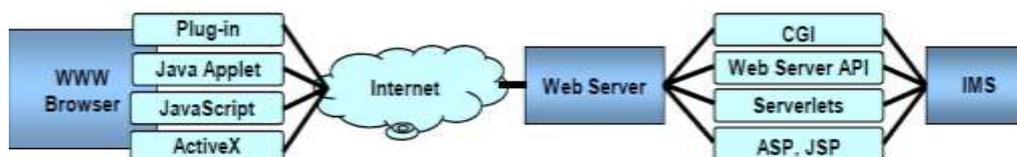


Gambar 2.9 Arsitektur *thin client*

Sumber : Alesheikh, et al (2002)

2. *Thick Client Architecture* (Aplikasi Pada Sisi Klien)

Pada umumnya, *web browser* dapat mengatasi dokumen HTML dan gambar raster dalam format standar. Namun untuk dapat mengatasi format data lain seperti data vektor, video, atau *file* musik, *browser* yang digunakan harus ditingkatkan. Sedangkan jika menggunakan komunikasi antara klien dan *server* seperti pada *Thin Client Architecture*, permasalahan tersebut belum dapat diatasi. Sehingga banyak aplikasi *browser* yang menawarkan mekanisme yang memungkinkan pihak ketiga lain untuk bekerja sama dengan *browser* sebagai *Plug-in*. Pada Gambar 2.10 menunjukkan arsitektur dimana pada bagian sisi klien bekerja sama dengan pihak ketiga.



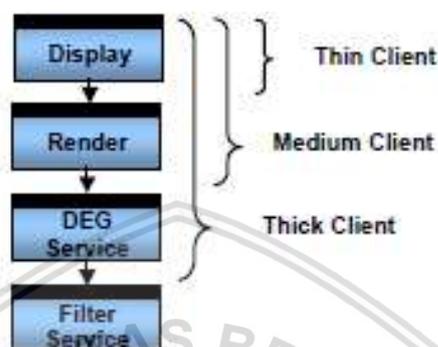
Gambar 2.10 Arsitektur *thick client*

Sumber : Alesheikh, et al (2002)



3. *Medium Client Architecture*

Untuk menghindari penggunaan data vektor pada sisi klien dan mengatasi permasalahan pada arsitektur sebelumnya, pada arsitektur ini menggunakan ekstensi pada kedua sisi server dan klien, dimana sisi klien mungkin memiliki lebih banyak fungsionalitas dibandingkan dengan *Thin Client Architecture*. Pada Gambar 2.11 berikut menunjukkan perbedaan antara *Thin Client*, *Medium Client*, dan *Thick Client*.



Gambar 2.11 Fungsionalitas *thin client*, *medium client*, dan *thick client*

Sumber : Aleseikh, et al (2002)

4. *Distributed Architecture*

Ide utama dari jenis arsitektur ini adalah bahwa sebuah program klien, yaitu *browser* atau aplikasi mandiri, harus dapat mengakses sumber data yang terdistribusi pada keseluruhan jaringan. Sumber data yang dimaksud disini adalah komponen *geodata* dan *geo-processing* yang tersedia pada jaringan. Arsitektur GIS yang terdistribusi dikatakan ideal apabila dapat melakukan *geodata* dan *geo-processing* dimanapun, yang berarti bahwa alat untuk melakukan *geodata* dan *geo-processing* harus dapat didistribusikan dimanapun pada jaringan tersebut.

2.9.2 *Library Leaflet*

Leaflet merupakan *library javascript* yang bersifat *open source*. *Library* ini dapat digunakan untuk membangun sebuah aplikasi peta berbasis web. Saat ini *leaflet* sudah didukung dengan teknologi HTML5 dan CSS3, serta dapat bekerja secara efisien pada berbagai jenis *platform* desktop dan mobile (Leaflet, 2015). *Library* ini tergolong ringan untuk dimuat pada halaman web karena hanya berukuran sekitar 34Kb *Javascript file*.

Selain berukuran ringan, *leaflet* juga menyediakan fungsi-fungsi standar untuk membangun WebGIS, misalnya menambahkan popups, penanda, *overlay* garis, *overlay shapes*, *zoom*, dan *pan*. Salah satu kelebihan paling signifikan dari *library* ini adalah kemampuan untuk memperluas fungsionalitas dengan plugin dari pihak ketiga (Maclean, 2012).

2.9.3 Semi-Automatic Classification Plug-in

Semi-Automatic Classification Plug-in (SCP) adalah *plug-in open source* gratis untuk QGIS. SCP memungkinkan untuk melakukan klasifikasi secara semi otomatis pada gambar hasil penginderaan jarak jauh. Selain itu, *plug-in* ini juga menyediakan beberapa alat untuk download, pemrosesan gambar awal, pemrosesan setelah klasifikasi, dan perhitungan raster. Beberapa data yang dapat di download melalui SCP antara lain adalah gambar ASTER, Landsat, MODIS, dan Sentinel-2. Beberapa algoritma juga tersedia untuk melakukan klasifikasi tutupan lahan (QGIS, 2017).

Beberapa algoritma yang digunakan untuk klasifikasi antara lain :

1. Algoritma *Maximum Likelihood*
Merupakan algoritma klasifikasi yang berpedoman pada nilai piksel yang sudah dikategorikan obyeknya atau dibuat dalam *training sample* untuk setiap obyek penutup lahan. Hasil klasifikasi yang baik dipengaruhi oleh *training sample*, semakin baik *training sample* maka hasil klasifikasi semakin memiliki akurasi yang baik.
2. Algoritma *Minimum Distance*
Merupakan algoritma klasifikasi menggunakan *vector* rata-rata *end member* masing-masing dan menghitung jarak Euclidean dari setiap piksel yang diketahui oleh vektor rata-rata untuk masing-masing kelas. Beberapa piksel memiliki kemungkinan tidak terklasifikasi apabila tidak memenuhi kriteria yang dipilih.
3. Algoritma *Spectral Angle Mapper*
Merupakan algoritma klasifikasi *spectral* berbasis fisik yang menggunakan sudut n-dimensi untuk mencocokkan piksel untuk *spectra* acuan.

2.9.4 CodeIgniter

CodeIgniter merupakan sebuah *framework* PHP yang membantu pengembang dalam mengembangkan aplikasi web berbasis PHP. Framework ini dapat dikatakan lebih efektif dan efisien, jika dibandingkan harus menulis semua kode program dari awal (Hakim, 2010). *CodeIgniter* bersifat *open source* dan banyak digunakan dalam pembuatan aplikasi php yang dinamis. Dalam *CodeIgniter* terdapat banyak *library* yang dapat dimanfaatkan untuk mempermudah kinerja pengembang. *CodeIgniter* dikembangkan dengan menggunakan konsep *Model-View-Controller* (MVC). Konsep ini menerapkan teknik dalam pemrograman yang memisahkan antara bisnis *logic* (alur pikir), data *logic* (penyimpanan data), dan *presentation logic* (antarmuka aplikasi) atau biasanya secara sederhana disebut dengan pemisah antara desain, data, dan proses (Daqiqil, 2011).

2.10 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan metode untuk melakukan pengembangan sistem secara tradisional yang sampai saat ini masih digunakan oleh sebagian besar organisasi. SDLC merupakan

framework terstruktur yang berisi proses-proses berurut dari sistem informasi yang dikembangkan (A.S. & Shalahuddin, 2014).

2.10.1 Model *Waterfall*

Model *waterfall* sering juga disebut sebagai model sekuensial linear atau alur hidup klasik, dikarenakan model ini menyediakan alur pengembangan perangkat lunak secara terurut yang dimulai dari tahap analisis, desain, implementasi, pengujian dan tahapan pendukung (A.S. & Shalahuddin, 2014). Berikut ini adalah tahapan dari SDLC menurut model *waterfall* yang ditunjukkan pada Gambar 2.12.



Gambar 2.12 SDLC model *waterfall*

Sumber : diadaptasi dari A.S. & Shalahuddin (2014)

1. Analisis

Tahap analisis merupakan tahap mengumpulkan kebutuhan yang dilakukan secara intensif untuk menentukan spesifikasi dari perangkat lunak yang akan dibuat. Pada tahap ini, spesifikasi kebutuhan perangkat lunak yang sudah ditentukan harus didokumentasikan.

2. Desain

Tahap desain merupakan langkah untuk mendesain perangkat lunak yang akan dibangun, yang meliputi mendesain prosedur pengkodean, struktur data, arsitektur perangkat lunak, dan antarmuka perangkat lunak. Tahap ini menerjemahkan kebutuhan yang dispesifikasikan pada tahap analisis, sehingga dapat diimplementasikan menjadi program pada tahap selanjutnya. Hasil dari tahap desain juga perlu untuk didokumentasikan.

3. Pengkodean

Dalam tahap pengkodean, semua hasil pada tahap desain akan diubah menjadi program jadi perangkat lunak. Hasil dari tahap ini adalah sistem atau program komputer yang sesuai dengan hasil dari tahap desain yang dilakukan sebelumnya.

4. Pengujian

Tahap pengujian dilakukan pada sistem yang telah selesai dibuat. Pengujian dilakukan dari sisi kebutuhan fungsional dan kebutuhan non-fungsional, untuk memastikan bahwa semua kebutuhan fungsional dan non-fungsional sudah terpenuhi pada sistem.

2.11 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan bahasa pemodelan standar untuk memodelkan perancangan dari pembangunan perangkat lunak yang berorientasi objek. UML dapat digunakan untuk memodelkan kebutuhan, dengan menggambarkan, membangun, dan mendokumentasikan perangkat lunak (A.S. & Shalahuddin, 2014). UML yang akan dibuat dalam penelitian ini antara lain *activity diagram*, *diagram use case*, *class diagram*, dan *sequence diagram*.

2.11.1 *Activity Diagram*

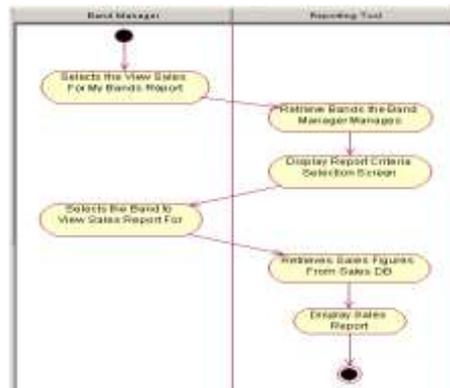
Activity diagram atau diagram aktivitas menggambarkan tentang proses, aliran, dan menu-menu yang terlibat dalam sebuah aktivitas yang akan dilakukan dalam sistem ketika terdapat suatu permintaan dari pengguna (A.S. & Shalahuddin, 2014). Dalam pembuatan diagram aktivitas, harus memperhatikan bahwa apa yang digambarkan pada diagram aktivitas adalah aliran aktivitas dari sistem, bukan apa yang dilakukan oleh aktor terhadap sistem. Pada Tabel 2.3 berikut, merupakan simbol-simbol yang digunakan dalam menggambarkan diagram aktivitas.

Tabel 2.3 Simbol dan deskripsi diagram aktivitas

| No | Simbol | Nama | Deskripsi |
|----|---|--------------------------|---|
| 1 |  | Aktivitas | Merupakan simbol untuk mendeskripsikan aktivitas yang dilakukan. Sebuah aktivitas dideskripsikan dengan menggunakan awalan kata kerja. |
| 2 |  | Status Awal | Merupakan simbol untuk menunjukkan status awal dari aktivitas. Setiap diagram aktivitas selalu diawali dengan sebuah status awal. |
| 3 |  | Decision/ percabangan | Simbol ini digunakan jika ada kondisi komunikasi yang harus bercabang untuk menentukan suatu pilihan, namun tetap berakhir dalam satu arah. |
| 4 |  | Join/ Penggabungan | Simbol ini digunakan untuk menggabungkan aktivitas yang lebih dari satu menjadi satu. |
| 5 |  | <i>Swimlane</i> | Simbol ini digunakan untuk memberikan batasan antara satu organisasi bisnis dengan organisasi bisnis lainnya, yang terlibat dengan aktivitas sistem yang terjadi. |
| 6 |  | Status Akhir | Merupakan simbol untuk menunjukkan status akhir dari sebuah diagram aktivitas. |

Sumber : diadaptasi dari A.S. & Shalahuddin (2014)

Berikut pada Gambar 2.13 merupakan contoh penggunaan simbol-simbol untuk diagram aktivitas yang telah dijelaskan diatas.



Gambar 2.13 Contoh *activity diagram*

Sumber : Bell (2003)

2.11.2 Diagram *Use Case*

Diagram *use case* digunakan untuk memodelkan aktivitas apa saja yang dapat dilakukan oleh aktor pada sistem. Pada diagram ini juga akan dideskripsikan hubungan antara aktor dengan sistem, siapa saja yang berhak menggunakan sistem, serta apa saja fungsi yang terdapat dalam sistem. Terdapat dua hal utama yang harus diperhatikan dalam pembuatan diagram *use case* yaitu *use case* yang merupakan fungsionalitas yang ada pada sistem dan aktor yang merupakan orang, proses, atau sistem lain yang dapat berinteraksi dengan sistem yang dibuat (A.S. & Shalahuddin, 2014).

Dalam pembuatan diagram *use case* digunakan beberapa simbol yang ditunjukkan pada Tabel 2.4 berikut.

Tabel 2.4 Simbol dan deskripsi diagram *use case*

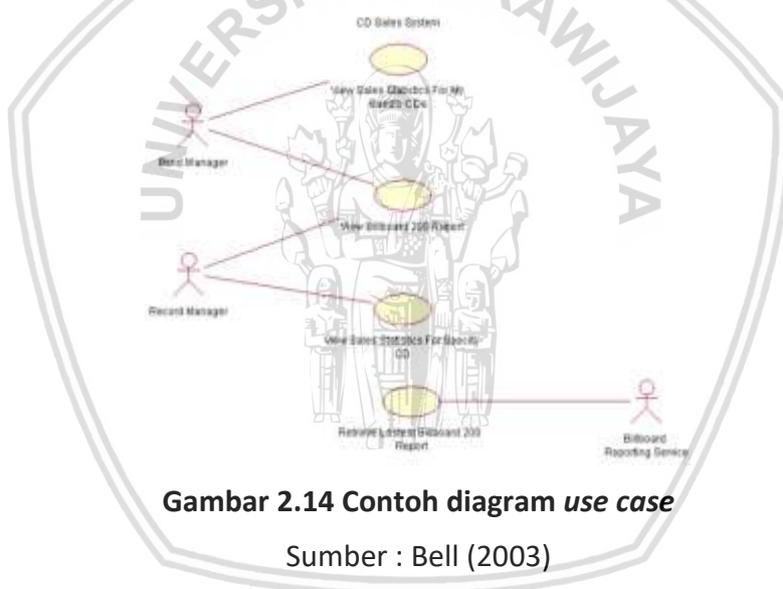
| No | Simbol | Nama | Deskripsi |
|----|--------|---------------------------------|--|
| 1 | | Actor/ Aktor | Simbol untuk pengguna yang dapat berinteraksi dengan sistem melalui <i>use case</i> . |
| 2 | | <i>Use Case</i> | Merupakan simbol untuk menunjukkan fungsionalitas yang ada pada sistem. Sebuah <i>use case</i> dapat saling bertukar pesan dengan aktor maupun dengan <i>use case</i> lain. |
| 3 | | Asosiasi/ <i>Association</i> | Simbol ini digunakan untuk menggambarkan komunikasi yang terjadi antara aktor dengan <i>use case</i> ataupun sebaliknya. |
| 4 | | Ekstensi/ <i>Extend</i> | Simbol untuk menggambarkan relasi antara <i>use case</i> dengan <i>use case</i> tambahan, dimana <i>use case</i> dapat berdiri sendiri tanpa adanya <i>use case</i> tambahan tersebut. |

Tabel 2.4 Simbol dan deskripsi diagram *use case* (lanjutan)

| No | Simbol | Nama | Deskripsi |
|----|---|----------------|---|
| 5 | <code><<include>></code> → | <i>Include</i> | Simbol yang digunakan untuk menggambarkan relasi antara <i>use case</i> dengan <i>use case</i> tambahan, namun <i>use case</i> memerlukan <i>use case</i> tambahan tersebut sebagai syarat untuk bisa melaksanakan fungsinya. |
| 6 | → | Generalisasi | Merupakan hubungan generalisasi dan spesialisasi antara dua <i>use case</i> dimana salah satu fungsi merupakan fungsi yang lebih umum daripada fungsi lainnya. |

Sumber : diadaptasi dari A.S. & Shalahuddin (2014)

Berikut pada Gambar 2.14 merupakan contoh penggunaan simbol-simbol pada diagram *use case*.



Gambar 2.14 Contoh diagram *use case*

Sumber : Bell (2003)

2.11.3 Class Diagram

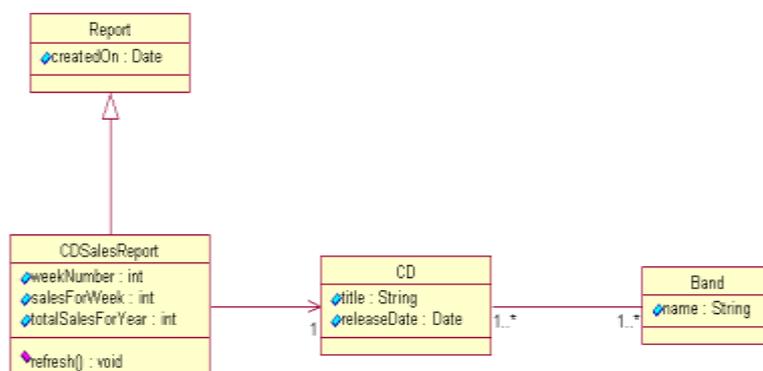
Class diagram atau diagram kelas merupakan diagram yang digunakan untuk memberikan gambaran dari struktur sistem dengan cara mendefinisikan kelas-kelas yang akan terlibat dalam kode program. Kelas-kelas dibuat dalam tahap perancangan yang didasarkan pada fungsi apa saja yang harus ada pada sistem. Sebuah kelas terdiri dari metode dan atribut. Metode adalah fungsi-fungsi yang terdapat pada suatu kelas. Sedangkan atribut adalah variabel-variabel yang dimiliki oleh suatu kelas, sehingga dapat dibedakan antara satu kelas dengan kelas lainnya (A.S. & Shalahuddin, 2014). Simbol-simbol yang digunakan dalam pembuatan diagram kelas dapat dilihat pada Tabel 2.5 berikut.

Tabel 2.5 Simbol dan deskripsi *class diagram*

| No | Simbol | Nama | Deskripsi |
|----|--------|---|---|
| 1 | | Kelas | Simbol yang digunakan untuk mendeskripsikan kelas-kelas yang terdapat pada sistem. |
| 2 | | Interface/ antarmuka | Interface yang memiliki konsep sama dengan pemrograman berorientasi objek. |
| 3 | | Asosiasi/ Association | Simbol ini merepresentasikan hubungan antar kelas secara umum. |
| 4 | | Directed Association/ Asosiasi berarah | Simbol ini digunakan untuk merepresentasikan hubungan antar kelas, dimana kelas yang satu digunakan oleh kelas yang lain. |
| 5 | | Generalisasi | Simbol ini digunakan untuk merepresentasikan hubungan generalisasi dan spesialisasi, dimana kelas yang satu berisi fungsi-fungsi yang lebih umum dibandingkan dengan kelas lainnya. |
| 6 | | Dependency/ Ketergantungan | Simbol ini digunakan untuk menunjukkan adanya ketergantungan suatu kelas terhadap kelas lainnya. |
| 7 | | Agregasi/ Agregasi | Simbol ini digunakan menunjukkan hubungan antar kelas dengan makna semua-bagian (<i>whole-part</i>). |

Sumber : diadaptasi dari A.S. & Shalahuddin (2014)

Berikut pada Gambar 2.15 merupakan contoh dari penggunaan simbol-simbol pada diagram kelas.



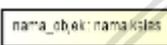
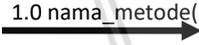
Gambar 2.15 Contoh *class diagram*

Sumber : Bell (2003)

2.11.4 Sequence Diagram

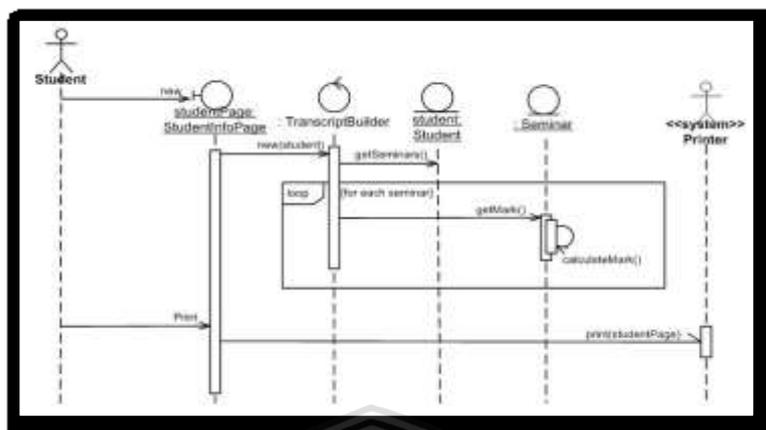
Sequence diagram merupakan diagram untuk mendeskripsikan pesan yang dikirim atau diterima antar objek serta waktu hidup dari objek. Pesan-pesan yang terdapat dalam diagram *sequence*, dikirimkan dalam bentuk metode-metode yang telah didefinisikan pada *class diagram* (A.S. & Shalahuddin, 2014). Simbol-simbol yang digunakan untuk membuat *sequence diagram* ditunjukkan pada Tabel 2.6 berikut.

Tabel 2.6 Simbol dan deskripsi *sequence diagram*

| No | Simbol | Nama | Deskripsi |
|----|---|--|--|
| 1 |  | <i>Actor/</i> Aktor | Digunakan untuk menunjukkan orang, proses, atau sistem lain yang terlibat. |
| 2 |  | <i>Lifeline/</i> Garis hidup | Digunakan untuk menunjukkan kehidupan dari objek yang terlibat. |
| 3 |  | Objek | Digunakan untuk menunjukkan objek yang terlibat dalam pertukaran pesan atau <i>message</i> . |
| 4 |  | <i>Activation Bar /</i> Waktu Aktif | Digunakan untuk menunjukkan bahwa objek sedang aktif dan sedang berinteraksi. Semua yang terhubung dengan <i>activation bar</i> ini menggambarkan tahapan yang sedang dilakukan didalamnya. |
| 5 |  | Pesan tipe <i>call</i> | Digunakan untuk menunjukkan bahwa objek sedang memanggil fungsi yang terdapat pada objek lain atau pada objek itu sendiri. Arah panah akan mengarah pada objek yang memiliki fungsi yang sedang dipanggil. Sehingga dalam penulisan di <i>class diagram</i> , fungsi yang dipanggil harus terdapat pada kelas atau objek yang dipanggil. |
| 6 |  | Pesan tipe <i>send</i> | Digunakan untuk menunjukkan bahwa suatu objek sedang mengirimkan data/inputan/informasi ke objek lainnya yang ditunjuk oleh arah panah. |
| 7 |  | Pesan tipe <i>return</i> | Digunakan untuk menunjukkan bahwa suatu objek mengembalikan suatu nilai tertentu setelah menjalankan suatu fungsi. |

Sumber : diadaptasi dari A.S. & Shalahuddin (2014)

Berikut pada Gambar 2.16 merupakan contoh pembuatan diagram *sequence* menggunakan simbol-simbol yang telah dijelaskan diatas.



Gambar 2.16 Contoh *sequence* diagram

Sumber : Ambler (2004)

2.12 Pengujian *Black-Box*

Pengujian *Black-box* atau pengujian perilaku bertujuan untuk memastikan kebutuhan fungsional yang didefinisikan pada tahap analisis dan perancangan telah terpenuhi pada sistem yang dibuat. Dalam melakukan pengujian ini, akan dibuat uji kasus atau *test case* untuk setiap kebutuhan fungsional. Dalam pengujian ini, perancang uji akan memilih input yang valid dan tidak valid pada kasus uji dan menentukan *output* yang benar untuk setiap pilihan yang valid dan tidak valid. Metode uji ini dapat diterapkan pada semua tingkat pengujian perangkat lunak baik pada tingkat unit, integrasi, fungsional, sistem, dan penerimaan. Pengujian ini memungkinkan pengembang sistem untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Pressman, 2010). Beberapa kesalahan yang ingin ditemuka pada pengujian ini antara lain kesalahan antarmuka, adanya fungsi yang salah atau hilang, kesalahan kinerja, kesalahan dalam struktur data atau akses eksternal basis data.

2.13 Pengujian *Portability*

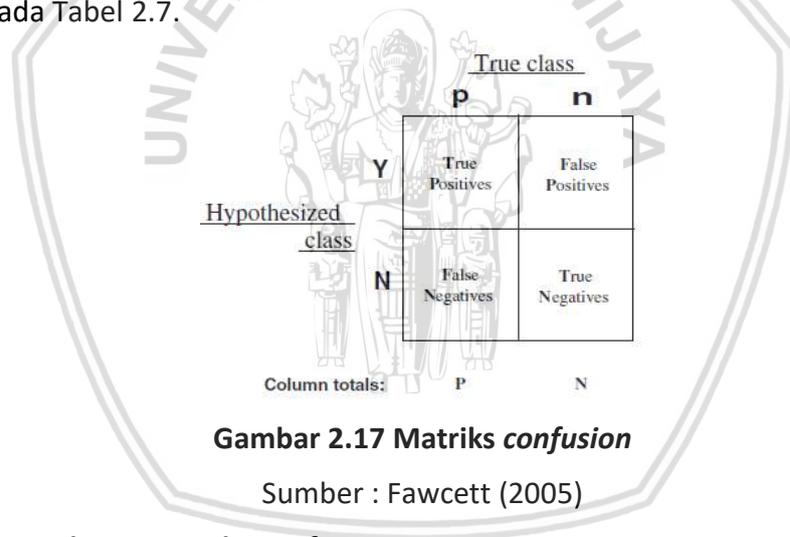
Pengujian *portability* merupakan pengujian yang dilakukan untuk mengetahui *portability* dari sistem apabila dibuka pada sistem perangkat lunak atau lingkungan perangkat lunak yang berbeda (Pressman, 2010). Pengujian *portability* sebuah perangkat lunak berbasis web dianggap memiliki *portability* yang tinggi apabila dari keseleruhan kasus uji memiliki nilai valid >80% terhadap *desktop browser* ataupun *mobile browser*. *Browser* yang akan diuji adalah *browser* yang banyak digunakan oleh sebagian besar masyarakat di dunia (Salonen, 2012).

Untuk melakukan pengujian tersebut akan menggunakan bantuan aplikasi SortSite yang menyediakan *tools* untuk pengujian pada dua belas jenis *desktop*,

terdiri dari tujuh *desktop browser* dan lima *mobile browser*. Hasil dari pengujian menggunakan *tools* ini adalah dengan menampilkan *error (missing file)* dan *warning (major dan minor performance)* dari *user interface* yang diuji. Apabila tidak terdapat *error* ataupun *warning* dari sistem yang diuji, maka hal tersebut menunjukkan bahwa sistem yang diuji telah memenuhi karakteristik *portability* (PowerMapper, 1996).

2.14 Pengujian Accuracy

Dalam melaksanakan pengujian *accuracy*, akan digunakan metode ROC (*Receiver Operating Characteristics*). ROC merupakan sebuah teknik untuk memvisualisasikan, mengatur, dan memilih *classifier* berdasarkan performansinya. Sebuah *classifier* akan dipetakan dari contoh terhadap kelas yang diprediksi. Beberapa model klasifikasi akan menghasilkan *output* yang kontinyu dengan nilai *threshold* yang berbeda-beda (Stefano, et al., 2007). Terdapat kemungkinan yang dapat dihasilkan dari model ini yang dapat dilihat pada Gambar 2.17, dimana *p* adalah jumlah data positif dan *n* adalah jumlah data negatif. Sedangkan *P* adalah jumlah total positif dan *N* adalah jumlah total negative (Fawcett, 2005). Kemungkinan-kemungkinan tersebut antara lain dapat dilihat pada Tabel 2.7.



Gambar 2.17 Matriks *confusion*

Sumber : Fawcett (2005)

Tabel 2.7 Deskripsi matriks *confusion*

| No. | Kelas Kategori | Deskripsi |
|-----|-----------------------|---|
| 1 | <i>True Positive</i> | Hasil yang diprediksi dan keadaan yang sebenarnya sama-sama bernilai positif. |
| 2 | <i>False Positive</i> | Hasil yang diprediksi bernilai positif, namun keadaan yang sebenarnya bernilai negatif. |
| 3 | <i>False Negative</i> | Hasil yang diprediksi bernilai negatif, namun keadaan yang sebenarnya bernilai positif. |
| 4 | <i>True Negative</i> | Hasil yang diprediksi dan keadaan yang sebenarnya sama-sama bernilai negatif. |

Formulasi untuk menghitung variabel-variabel diatas dapat dilihat pada Persamaan 2.1 sampai Persamaan 2.5 berikut (Fawcett, 2005).

$$TPR = \frac{TP}{P} \quad (2.1)$$

$$FPR = \frac{FP}{N} \quad (2.2)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2.3)$$

$$\text{Recall} = \frac{TP}{P} \quad (2.4)$$

$$\text{Akurasi} = \frac{TP+TN}{P+N} \quad (2.5)$$

Keterangan :

TPR = *True Positive Rate*

FPR = *False Positive Rate*

TP = jumlah *True Positive*

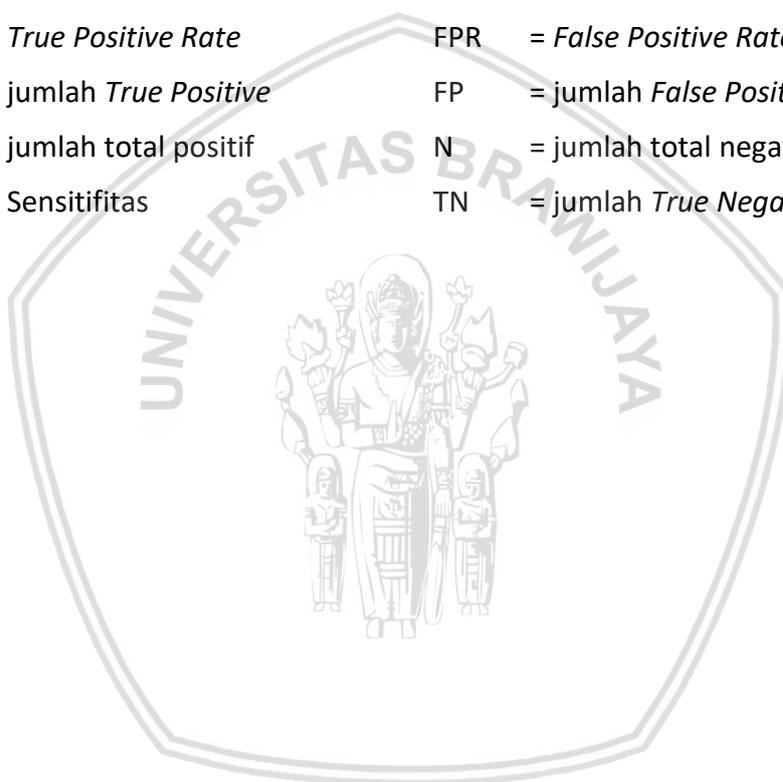
FP = jumlah *False Positive*

P = jumlah total positif

N = jumlah total negatif

Recall = Sensitifitas

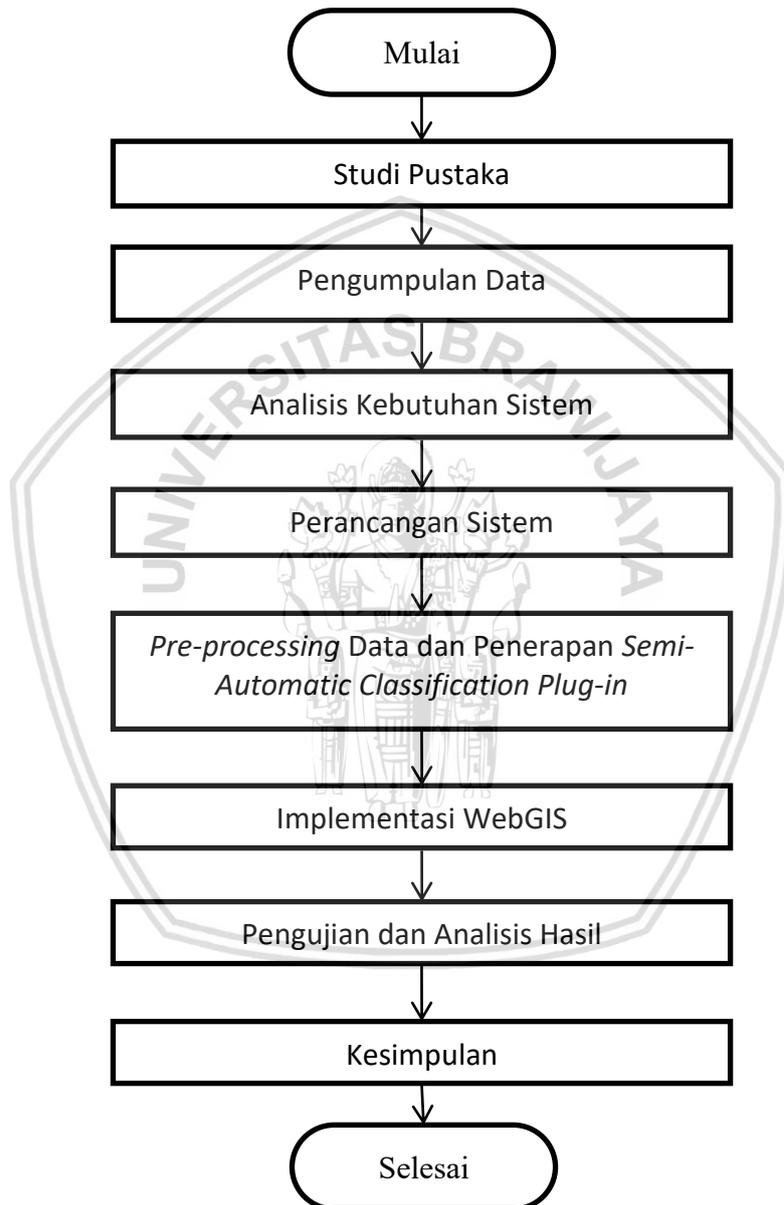
TN = jumlah *True Negative*



BAB 3 METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Metodologi penelitian merupakan prosedur yang akan dilakukan penulis demi tercapainya tujuan penelitian yang dapat dilihat pada Gambar 3.1 berikut :



Gambar 3.1 Metodologi penelitian

3.2 Studi Pustaka

Pada tahap studi literatur akan dilakukan pembelajaran pustaka dan studi literatur terhadap semua yang berhubungan dengan penelitian. Studi literatur dilakukan dengan mencari dan menggali referensi yang diambil dari buku,



laporan penelitian sebelumnya, skripsi, jurnal, dan artikel. Terdapat beberapa teori yang dipelajari untuk mendukung penelitian, antara lain :

1. Pemrograman web menggunakan bahasa pemrograman PHP.
2. Penggunaan aplikasi Quantum GIS.
3. Erupsi Gunung Merapi Yogyakarta
4. *Semi Automatic Classification*
5. Tahapan SDLC
6. *Unified Modelling Language* (UML) yang terdiri dari *activity diagram*, *use case diagram*, *sequence diagram*, dan *class diagram*.
7. *Framework CodeIgniter*
8. Pengujian perangkat lunak yang terdiri dari Pengujian *Black Box* dan Pengujian *Portability*.
9. Pengujian *accuracy*

3.3 Pengumpulan Data

Terdapat dua metode pengumpulan data-data serta informasi yang digunakan dalam penelitian ini, yaitu :

1. Metode Wawancara
Dalam metode ini, peneliti melakukan wawancara langsung terkait erupsi Gunung Merapi di Yogyakarta beserta dampaknya bagi Kabupaten Sleman. Wawancara dilakukan dengan salah satu narasumber yang bekerja di Badan Penanggulangan Bencana Daerah (BPDBD) Kabupaten Sleman.
2. Metode Kepustakaan
Merupakan teknik pengumpulan data dengan cara mengadakan studi pembelajaran terhadap literatur-literatur, buku-buku, catatan-catatan, serta laporan-laporan yang memiliki hubungan dengan masalah yang diangkat (Nazir, 1988). Metode ini digunakan untuk membantu menentukan lokasi-lokasi yang terkena dampak erupsi Gunung Merapi dan tahun-tahun kejadian erupsi Gunung Merapi di Kabupaten Sleman.

3.4 Analisis Kebutuhan Sistem

Analisis kebutuhan merupakan tahap untuk mendapatkan semua kebutuhan yang harus ada pada sistem. Adapun hal-hal yang akan dilakukan pada tahap analisis kebutuhan antara lain :

1. Melakukan analisis kebutuhan fungsional dan non-fungsional, mengidentifikasi aktor yang akan terlibat dalam sistem dan mendeskripsikan karakteristik masing-masing pengguna sistem. Kebutuhan fungsional merupakan kebutuhan yang berkaitan dengan fitur-fitur apa saja yang dibutuhkan oleh pengguna dan harus tersedia pada sistem. Sedangkan Kebutuhan non-fungsional merupakan kebutuhan yang berkaitan dengan

kualitas dan batasan sistem saat diaplikasikan ke dalam program jadi. Kebutuhan non-fungsional bukan merupakan kebutuhan utama, namun harus ada sehingga dapat mendukung kebutuhan fungsional.

2. Membuat daftar kebutuhan pengguna menggunakan pemodelan diagram *use case*.

3.5 Perancangan Sistem

Tahap perancangan merupakan tahap untuk membuat desain sistem berdasarkan kebutuhan yang didapatkan dari tahap sebelumnya. Perancangan dibuat dengan berorientasi objek dan menggunakan pemodelan UML. Perancangan yang akan dibuat antara lain sebagai berikut :

1. Perancangan arsitektur.
2. Perancangan *activity diagram* untuk memodelkan perilaku dari sistem.
3. Perancangan *sequence diagram* untuk memodelkan objek-objek dan metode-metode dari sebuah fungsionalitas.
4. Perancangan *class diagram* untuk memodelkan kelas yang terdapat pada sistem dan hubungannya.
5. Perancangan struktur basis data untuk memodelkan tabel-tabel untuk penyimpanan data yang dibutuhkan oleh sistem. Perancangan ini dibuat berdasarkan kelas model yang terdapat pada *class diagram*.
6. Perancangan *user interface* atau antarmuka sistem.

3.6 Pre-processing Data dan Penerapan *Semi-Automatic Classification Plug-in*

Tahap ini berperan dalam pengolahan awal terhadap data yang dibutuhkan dalam penelitian, sehingga data tersebut dapat digunakan pada implementasi WebGIS. Data awal yang didapatkan dari *website* earthexplorer.usgs.gov akan di *pre-processing* terlebih dahulu, baru kemudian data tersebut akan diolah menggunakan *Semi-Automatic Classification Plug-in* untuk mendapatkan daerah-daerah yang terkena dampak erupsi Gunung Merapi di Kabupaten Sleman. Dalam melakukan tahap ini, terdapat beberapa kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*) yang meliputi :

1. Kebutuhan *Hardware*
Hardware yang dibutuhkan adalah PC dengan spesifikasi RAM minimal 4 GB dan *HardDisk* minimal 500 GB.
2. Kebutuhan *Software*
Software yang dibutuhkan adalah Quantum GIS (QGIS) 2.14.3.

3.7 Implementasi WebGIS

Dalam tahap ini merupakan implementasi dari apa yang dilakukan pada tahap perancangan sebelumnya. Pada tahap ini akan dilakukan beberapa hal sehingga dapat menciptakan sebuah sistem, yaitu sebagai berikut :

- a. Pembuatan antarmuka pengguna.
- b. Input data yang dibutuhkan kedalam *database*.
- c. Menampilkan hasil pemetaan dampak erupsi Gunung Merapi di Kabupaten Sleman serta menampilkan informasi dari peta yang divisualisasikan.

Untuk dapat mengimplementasikan ke dalam WebGIS, terdapat beberapa kebutuhan perangkat keras dan kebutuhan perangkat lunak yang meliputi :

1. Kebutuhan *Hardware*

Hardware yang dibutuhkan adalah PC dengan spesifikasi RAM minimal 4 GB dan HardDisk minimal 500 GB.

2. Kebutuhan *Software*

1. Sublime Text / Notepad ++
2. *Web Browser*
3. XAMPP
4. Mysql

3.8 Pengujian dan Analisa Hasil

Tahap pengujian dilakukan untuk menemukan berbagai error atau kesalahan yang tidak dapat ditemukan ketika melakukan tahap-tahap sebelumnya. Terdapat 3 jenis pengujian yang akan dilakukan, antara lain pengujian *black-box* yaitu uji validasi, pengujian *portability*, dan pengujian *accuracy*. Pengujian validasi dan pengujian *portability* digunakan untuk menguji sistem yang telah selesai dibuat. Sedangkan pengujian *accuracy* digunakan untuk menguji ketepatan dari hasil pemetaan daerah yang terkena erupsi.

3.9 Kesimpulan

Pada tahap ini berisi kesimpulan mengenai semua tahapan yang telah dilalui pada tahap sebelumnya serta saran terhadap hasil yang telah dicapai. Kesimpulan merupakan ringkasan jawaban dari rumusan masalah dan harus sesuai dengan tujuan dari penelitian ini. Sedangkan saran berfungsi untuk memperbaiki kesalahan atau kekurangan yang diharapkan dapat diperbaiki dalam pengembangan lebih lanjut.

BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN

Dalam bab ini berisi tentang analisis kebutuhan dan perancangan dari sistem yang akan dibuat. Dalam analisis dan perancangan terdapat beberapa tahap yang akan dilakukan sebagai berikut :

1. Analisis

Dalam menganalisis kebutuhan pada sistem, terdapat beberapa tahap yang dilakukan antara lain penjabaran gambaran umum, mengidentifikasi aktor, mendedinisikan kebutuhan fungsional dan kebutuhan non-fungsional, dan membuat diagram *use case*.

2. Perancangan

Pada perancangan, terdapat beberapa jenis perancangan yang akan dibuat untuk memberikan gambaran dari sistem yang akan dibuat. Perancangan yang akan dilakukan antara lain perancangan arsitektural, perancangan diagram aktivitas, perancangan diagram *sequence*, perancangan diagram *class*, perancangan basis data, dan perancangan antarmuka (*user interface*).

4.1 Analisis Kebutuhan

Proses dalam melakukan analisis kebutuhan antara lain penjabaran gambaran umum, mengidentifikasi aktor, mendedinisikan kebutuhan fungsional dan kebutuhan non-fungsional, dan membuat diagram *use case*.

4.1.1 Gambaran Umum

Terdapat beberapa fitur utama yang ada pada sistem yang akan dibuat, yaitu sebagai berikut :

1. *History* Erupsi

History erupsi merupakan fitur utama yang digunakan untuk menampilkan peta yang menunjukkan daerah-daerah yang terkena dampak erupsi Gunung Merapi di Kabupaten Sleman. Peta yang akan ditampilkan disesuaikan dengan permintaan dari *user* berdasarkan tahun kejadian dari erupsi Gunung Merapi selama kurun waktu 30 tahun terakhir. Peta yang ditampilkan juga akan disertai dengan *basemap* peta. Selain itu, juga akan ditampilkan informasi dari warna-warna yang terdapat pada peta serta informasi mengenai luasan lahan dari daerah-daerah yang terkena dampak dari erupsi Gunung Merapi di Kabupaten Sleman.

2. *Basemap*

Basemap merupakan fitur yang terdapat pada sistem untuk dapat merubah peta dasar dari peta yang divisualisasikan. *User* dapat memilih *basemap* yang diinginkan untuk peta berdasarkan pilihan *basemap* yang tersedia pada sistem.

3. Year Layer

Year Layer merupakan fitur yang terdapat pada sistem untuk menampilkan peta hasil pemetaan dampak erupsi Gunung Merapi berdasarkan pada tahun kejadian tertentu. *User* dapat memilih tahun ke berapa yang ingin ditampilkan pada sistem.

4. Informasi Histori Erupsi

Merupakan fitur yang menampilkan informasi mengenai peta hasil pemetaan yang divisualisasikan serta memberikan informasi mengenai luasan dari setiap daerah-daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman.

4.1.2 Identifikasi Aktor

Pada sistem yang akan dibuat, terdapat 2 aktor yang dapat menggunakan sistem yaitu *user* dan *admin*. Berikut pada Tabel 4.1 menjelaskan tentang karakteristik dari aktor yang ada pada sistem.

Tabel 4.1 Identifikasi dan karakteristik pengguna

| No | Identifikasi Pengguna | Karakteristik |
|----|-----------------------|--|
| 1 | Admin | Admin adalah salah satu aktor yang ada pada sistem. Admin memiliki hak akses paling tinggi dan memiliki tugas untuk manajemen serta memanipulasi data yang akan ditampilkan menjadi informasi pada sistem. Terdapat 2 karakteristik dari Admin yaitu : <ol style="list-style-type: none"> 1. Admin bisa mendapatkan otoritasnya sebagai Admin dengan masuk ke dalam sistem menggunakan <i>username</i> dan <i>password</i>. 2. Admin dapat melakukan fungsi CRUD (<i>Creat</i>, <i>Read</i>, <i>Update</i>, dan <i>Delete</i>) pada data yang ada di sistem. |
| 2 | <i>User</i> | Yang termasuk <i>user</i> adalah pengguna selain Admin, yang memiliki hak akses terbatas pada sistem. Terdapat 5 karakteristik dari <i>user</i> , yaitu : <ol style="list-style-type: none"> 1. Dapat memanfaatkan fitur <i>history</i> erupsi untuk memvisualisasikan seluruh hasil pemetaan daerah yang terkena erupsi di Kabupaten Sleman. 2. Dapat memilih <i>basemap</i> untuk memvisualisasikan peta dengan <i>basemap</i> yang dipilih. 3. Dapat memilih hasil pemetaan daerah yang terkena erupsi berdasarkan tahun kejadian tertentu. 4. Dapat melihat informasi dari peta yang divisualisasikan dan melihat diagram persentase luasan daerah erupsi. 5. Dapat mencetak peta beserta informasi peta pada sistem. |

4.1.3 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus dipenuhi oleh sistem. Kebutuhan fungsional ini juga menggambarkan fitur-fitur apa saja yang terdapat pada sistem. Berikut ini pada Tabel 4.2 dan Tabel 4.3 adalah penjelasan kebutuhan fungsional dari aktor *User* dan Admin.

1. *User*

Tabel 4.2 Kebutuhan fungsional *user*

| No | Kode Fungsi | Nama Fungsi | Deskripsi |
|----|-------------|---|---|
| 1 | MIS-1-001 | Melihat <i>history</i> erupsi | <i>User</i> dapat melihat peta daerah yang terkena erupsi Gunung Merapi di Sleman. |
| 2 | MIS-1-002 | Melihat informasi <i>history</i> erupsi | <i>User</i> dapat melihat informasi mengenai peta yang divisualisasikan. |
| 3 | MIS-1-003 | Memilih <i>year layer</i> | <i>User</i> dapat melihat hasil pemetaan daerah yang terdampak erupsi berdasarkan pada tahun kejadian tertentu. |
| 4 | MIS-1-004 | Memilih <i>Basemap</i> | <i>User</i> dapat memilih dan mengubah peta dasar untuk peta yang divisualisasikan. |
| 5 | MIS-1-005 | Mencetak Peta | <i>User</i> dapat mencetak hasil pemetaan daerah yang terdampak erupsi pada sistem. |
| 6 | MIS-1-006 | Melihat Diagram | <i>User</i> dapat melihat diagram yang menunjukkan persentase luasan daerah yang terdampak erupsi. |

2. Admin

Tabel 4.3 Kebutuhan fungsional Admin

| No | Kode Fungsi | Nama Fungsi | Deskripsi |
|----|-------------|---------------|---|
| 1 | MIS-1-007 | <i>Login</i> | Admin dapat masuk ke dalam sistem dengan menginputkan <i>username</i> dan <i>password</i> yang telah didaftarkan, sehingga Admin bisa mendapatkan otoritas sebagai Admin dan dapat memanajemen serta memanipulasi data. |
| 2 | MIS-1-008 | <i>Logout</i> | Admin dapat keluar dari sistem untuk menghapus otoritas sebagai Admin. |

Tabel 4.3 Kebutuhan fungsional Admin (lanjutan)

| No | Kode Fungsi | Nama Fungsi | Deskripsi |
|----|-------------|------------------------------------|--|
| 3 | MIS-1-009 | Melihat data <i>history</i> | Admin dapat melihat data JSON yang akan ditampilkan sebagai peta daerah yang terkena erupsi yang pernah ditambahkan ke sistem. |
| 4 | MIS-1-010 | Menambah data <i>history</i> | Admin dapat menambahkan data JSON dari daerah yang terkena erupsi ke sistem. |
| 5 | MIS-1-011 | Menghapus data <i>history</i> | Admin dapat menghapus data JSON dari daerah yang terkena erupsi yang pernah ditambahkan ke sistem. |
| 7 | MIS-1-012 | Melihat data info <i>history</i> | Admin dapat melihat data informasi peta yang pernah ditambahkan ke sistem. |
| 8 | MIS-1-013 | Menambah data info <i>history</i> | Admin dapat menambahkan data informasi peta baru ke dalam sistem. |
| 9 | MIS-1-014 | Menghapus data info <i>history</i> | Admin dapat menghapus informasi peta yang pernah ditambahkan ke sistem. |
| 10 | MIS-1-015 | Mengedit data info <i>history</i> | Admin dapat memperbaiki data informasi peta yang pernah ditambahkan ke sistem. |

4.1.4 Kebutuhan Non-Fungsional

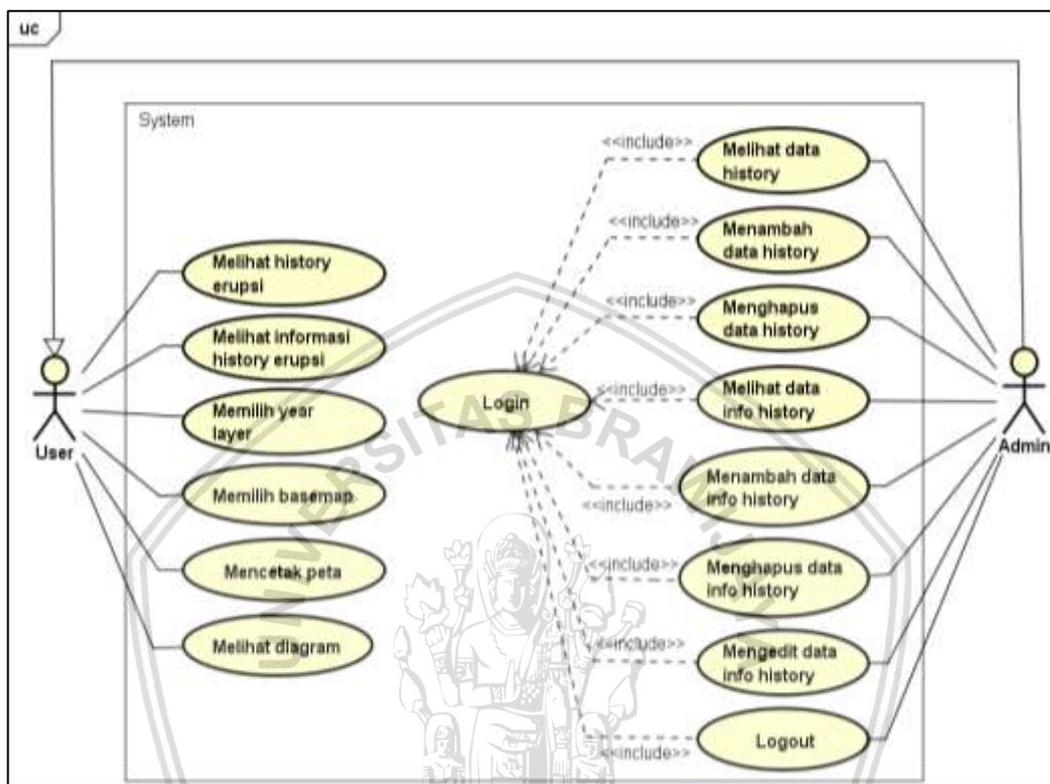
Kebutuhan non-fungsional merupakan kebutuhan yang secara tidak langsung harus dimiliki oleh sistem. Berikut ini pada Tabel 4.4 merupakan kebutuhan non-fungsional sistem secara umum.

Tabel 4.4 Kebutuhan Non-Fungsional

| No | Kode Fungsi | Nama Fungsi | Deskripsi |
|----|-------------|---------------------|--|
| 1 | MIS-2-001 | <i>Availability</i> | Sistem harus dapat beroperasi terus menerus selama 7 hari per minggu, 24 jam per hari tanpa berhenti. Sehingga dapat diakses kapanpun dan dimanapun. |
| 2 | MIS-2-002 | <i>Portability</i> | Sistem harus dapat dijalankan pada berbagai browser baik <i>desktop</i> ataupun <i>mobile</i> yang dapat mendukung penggunaan PHP, HTML, CSS dan JSON. |

4.1.5 Diagram Use Case

Pada diagram *use case* terdapat aktor beserta perilaku-perilaku yang dapat dilakukan oleh aktor pada sistem. Diagram *use case* dibuat berdasarkan kebutuhan fungsional yang telah didefinisikan sebelumnya. Berikut pada Gambar 4.1 merupakan diagram *use case* dari sistem.



Gambar 4.1 Diagram *use case* sistem

4.1.6 Use Case Scenario

Use case scenario digunakan untuk menjelaskan bagaimana aktor dalam melakukan kebutuhan fungsionalnya dan sistem dalam memberikan respon untuk memenuhi kebutuhan fungsional tersebut. Berikut merupakan *use case scenario* dari setiap kebutuhan fungsional yang telah dijelaskan sebelumnya.

1. Use Case Scenario Melihat History Erupsi

Pada Tabel 4.5 berikut merupakan skenario untuk *user* dan sistem ketika melakukan kebutuhan fungsional melihat *history* erupsi.

Tabel 4.5 *Use case scenario* melihat *history* erupsi

| Item | Deskripsi |
|---------------|---|
| Nama Use Case | Melihat <i>History</i> Erupsi |
| Deskripsi | <i>Use case scenario</i> ini menjelaskan alur <i>user</i> untuk dapat melihat peta daerah yang terkena erupsi Gunung Merapi di Sleman yang terdapat dalam sistem. |



Tabel 4.5 *Use case scenario* melihat *history* erupsi (lanjutan)

| Item | Deskripsi |
|--------------|--|
| Aktor | <i>User</i> |
| Pra-Kondisi | Sistem telah berjalan |
| Tindakan | <ol style="list-style-type: none"> 1. <i>User</i> memilih menu <i>History</i>. 2. Sistem menampilkan halaman <i>history</i> yang berisi peta daerah yang terkena dampak erupsi Gunung Merapi di Sleman untuk seluruh tahun kejadian. |
| Post-Kondisi | Sistem merespon <i>user</i> dengan menampilkan peta daerah yang terkena dampak erupsi Gunung Merapi di Sleman untuk seluruh tahun kejadian. |
| Alternatif | - |

2. *Use Case Scenario* Melihat Informasi *History* Erupsi

Pada Tabel 4.6 berikut merupakan skenario untuk *user* dan sistem ketika melakukan kebutuhan fungsional melihat informasi *history* erupsi, sehingga *user* dapat melihat informasi dari peta yang divisualisasikan.

Tabel 4.6 *Use case scenario* melihat informasi *history* erupsi

| Item | Deskripsi |
|----------------------|--|
| Nama <i>Use Case</i> | Melihat Informasi History Erupsi |
| Deskripsi | <i>Use case scenario</i> ini menjelaskan alur <i>user</i> untuk dapat melihat informasi mengenai peta yang divisualisasikan di sistem. |
| Aktor | <i>User</i> |
| Pra-Kondisi | Sistem telah berjalan |
| Tindakan | <ol style="list-style-type: none"> 1. <i>User</i> memilih menu <i>History</i>. 2. Sistem menampilkan seluruh peta daerah yang terkena erupsi dan menampilkan seluruh informasi dari peta tersebut. |
| Post-Kondisi | Sistem merespon <i>User</i> dengan menampilkan informasi mengenai seluruh peta hasil pemetaan yang divisualisasikan. |
| Alternatif | - |

3. *Use Case Scenario* Memilih *Year Layer*

Pada Tabel 4.7 berikut merupakan skenario untuk *user* dan sistem ketika melakukan kebutuhan fungsional memilih *year layer*, sehingga hasil pemetaan

daerah yang terdampak erupsi dapat ditampilkan berdasarkan tahun kejadian tertentu.

Tabel 4.7 Use case scenario memilih year layer

| Item | Deskripsi |
|---------------|---|
| Nama Use Case | Memilih year layer |
| Deskripsi | Use case ini menjelaskan alur user untuk dapat menampilkan hasil pemetaan daerah yang terdampak erupsi pada tahun kejadian tertentu. |
| Aktor | User |
| Pra-Kondisi | Sistem telah berjalan. |
| Tindakan | <ol style="list-style-type: none"> 1. User memilih menu <i>History</i>. 2. Sistem menampilkan pilihan tahun kejadian erupsi. 3. User memilih salah satu tahun kejadian erupsi. Kemudian menekan tombol <i>Choose</i>. 4. Sistem menampilkan peta dan informasi dari daerah yang terkena erupsi pada tahun yang dipilih. |
| Post-Kondisi | Sistem merespon user dengan memvisualisasikan peta dan informasi berdasarkan tahun kejadian tertentu yang telah dipilih. |
| Alternatif | - |

4. Use Case Scenario Memilih Basemap

Pada Tabel 4.8 berikut merupakan skenario untuk user dan sistem ketika melakukan kebutuhan fungsional memilih *basemap*, sehingga peta dapat divisualisasikan dengan menggunakan *basemap* yang dipilih.

Tabel 4.8 Use case scenario memilih basemap

| Item | Deskripsi |
|---------------|---|
| Nama Use Case | Memilih <i>basemap</i> |
| Deskripsi | Use case ini menjelaskan alur user untuk memilih <i>basemap</i> yang disediakan oleh sistem. |
| Aktor | User |
| Pra-Kondisi | Sistem telah berjalan |
| Tindakan | <ol style="list-style-type: none"> 1. User memilih menu <i>Basemap</i>. 2. Sistem merespon dengan menampilkan pilihan <i>basemap</i> yang dapat dipilih user. 3. User memilih salah satu <i>basemap</i>. |

Tabel 4.8 Use case scenario memilih basemap (lanjutan)

| Item | Deskripsi |
|--------------|---|
| Tindakan | 4. Sistem menampilkan peta hasil pemetaan dengan menggunakan <i>basemap</i> yang telah dipilih. |
| Post-Kondisi | Sistem merespon <i>user</i> dengan menampilkan <i>basemap</i> yang dipilih sebagai peta dasar dari peta hasil pemetaan. |
| Alternatif | Apabila tidak tersambung dengan koneksi internet, maka sistem tidak dapat menampilkan <i>basemap</i> yang dipilih. |

5. Use Case Scenario Mencetak Peta

Pada Tabel 4.9 berikut merupakan skenario untuk *user* dan sistem ketika melakukan kebutuhan fungsional mencetak peta dan informasi dari peta yang terdapat pada sistem.

Tabel 4.9 Use case scenario mencetak peta

| Item | Deskripsi |
|---------------|---|
| Nama Use Case | Mencetak peta |
| Deskripsi | <i>Use case</i> ini menjelaskan alur <i>user</i> untuk dapat mencetak peta hasil pemetaan daerah yang terkena erupsi dan informasi dari peta yang ditampilkan di sistem. |
| Aktor | <i>User</i> |
| Pra-Kondisi | Sistem telah menampilkan halaman <i>History</i> |
| Tindakan | <ol style="list-style-type: none"> 1. <i>User</i> menekan tombol <i>Print</i>. 2. Sistem merespon dengan menyimpan peta dan informasi peta hasil pemetaan dalam bentuk PDF. |
| Post-Kondisi | Sistem akan mengambil data dari database, kemudian menyimpan data peta dan informasi dalam bentuk PDF. |
| Alternatif | - |

6. Use Case Scenario Melihat Diagram

Pada Tabel 4.10 berikut merupakan skenario untuk *user* dan sistem ketika melakukan kebutuhan fungsional melihat diagram yang menunjukkan persentase luasan daerah yang terdampak erupsi dari setiap tahun kejadian.

Tabel 4.10 Use case scenario melihat diagram

| Item | Deskripsi |
|---------------|-----------------|
| Nama Use Case | Melihat diagram |

Tabel 4.10 *Use case scenario* melihat diagram (lanjutan)

| Item | Deskripsi |
|--------------|--|
| Deskripsi | <i>Use case</i> ini menjelaskan alur <i>user</i> untuk dapat melihat diagram yang menunjukkan luasan daerah yang terkena dampak erupsi Gunung Merapi di Sleman. |
| Aktor | <i>User</i> |
| Pra-Kondisi | Sistem telah berjalan. |
| Tindakan | <ol style="list-style-type: none"> 1. <i>User</i> memilih menu <i>History</i>. 2. Sistem merespon dengan menampilkan halaman <i>history</i> yang berisi peta daerah yang terkena dampak erupsi untuk seluruh tahun kejadian. 3. <i>User</i> menekan tombol <i>See Diagram</i>. 4. Sistem menampilkan diagram yang menunjukkan luasan daerah yang terdampak erupsi. |
| Post-Kondisi | Sistem merespon <i>user</i> dengan menampilkan diagram luasan daerah yang terkena dampak erupsi Gunung Merapi di Sleman untuk seluruh tahun kejadian. |
| Alternatif | - |

7. *Use Case Scenario Login*

Pada Tabel 4.11 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional *login*, sehingga Admin dapat masuk ke sistem dan mendapatkan otoritasnya sebagai Admin.

Tabel 4.11 *Use case scenario login*

| Item | Deskripsi |
|----------------------|---|
| Nama <i>Use Case</i> | <i>Login</i> |
| Deskripsi | <i>Use case</i> ini menjelaskan alur Admin untuk dapat masuk ke dalam sistem. |
| Aktor | Admin |
| Pra-Kondisi | Sistem menampilkan halaman <i>History</i> |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Sign In</i>. 2. Sistem merespon dengan menampilkan form <i>Sign In</i> yaitu <i>username</i> dan <i>password</i>. 3. Admin menginputkan <i>username</i> dan <i>password</i> yang dibutuhkan, kemudian menekan tombol <i>Sign In</i>. 4. Sistem menampilkan halaman <i>Map Data</i>. |

Tabel 4.11 Use case scenario login (lanjutan)

| Item | Deskripsi |
|--------------|--|
| Post-Kondisi | Sistem merespon Admin dengan mengecek <i>username</i> dan <i>password</i> yang diinputkan, apabila <i>username</i> dan <i>password</i> ada di basis data maka sistem akan menampilkan halaman <i>Map Data</i> . |
| Alternatif | Apabila <i>username</i> dan <i>password</i> belum diisi atau belum terdaftar di basis data, maka setelah Admin menekan tombol <i>Sign In</i> , sistem akan menampilkan halaman <i>Sign In</i> dengan notifikasi " <i>Login failed! Please make sure your username and password are correct!</i> ". |

8. Use Case Scenario Logout

Pada Tabel 4.12 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional *Logout*, sehingga Admin dapat keluar dari sistem.

Tabel 4.12 Use case scenario logout

| Item | Deskripsi |
|---------------|---|
| Nama Use Case | <i>Logout</i> |
| Deskripsi | <i>Use case</i> ini menjelaskan alur Admin untuk dapat keluar dari sistem. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah masuk ke dalam sistem. |
| Tindakan | <ol style="list-style-type: none"> Admin menekan tombol <i>Sign Out</i>. Sistem merespon dengan menampilkan halaman <i>History</i> dari sistem. |
| Post-Kondisi | Sistem merespon Admin dengan menghancurkan <i>session</i> dari <i>username</i> Admin, kemudian menampilkan halaman <i>History</i> sistem. |
| Alternatif | - |

9. Use Case Scenario Melihat Data History

Pada Tabel 4.13 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional melihat data *history*.

Tabel 4.13 Use case scenario melihat data history

| Item | Deskripsi |
|---------------|----------------------|
| Nama Use Case | Melihat data history |

Tabel 4.13 Use case scenario melihat data *history* (lanjutan)

| Item | Deskripsi |
|--------------|---|
| Deskripsi | <i>Use case</i> ini menjelaskan alur Admin untuk dapat melihat seluruh data JSON yang akan ditampilkan sebagai peta hasil pemetaan, yang pernah ditambahkan ke sistem. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah berhasil masuk ke dalam sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Map Data</i> pada halaman Admin. 2. Sistem merespon dengan menampilkan seluruh data JSON dari hasil pemetaan daerah yang terdampak erupsi, yang pernah ditambahkan ke sistem. |
| Post-Kondisi | Sistem merespon Admin dengan mengambil seluruh data JSON dari basis data, kemudian menampilkan seluruh data tersebut di halaman <i>Map Data</i> sistem. |
| Alternatif | - |

10. Use Case Scenario Menambah Data *History*

Pada Tabel 4.14 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional menambah data *history* berupa data JSON.

Tabel 4.14 Use case scenario menambah data *history*

| Item | Deskripsi |
|----------------------|--|
| Nama <i>Use Case</i> | Menambah Data <i>History</i> |
| Deskripsi | <i>Use case</i> ini menjelaskan alur Admin untuk menambahkan data JSON baru ke sistem yang nantinya akan divisualisasikan ke halaman <i>History</i> . |
| Aktor | Admin |
| Pra-Kondisi | Admin telah masuk (<i>login</i>) ke dalam sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Map Data</i> pada halaman Admin. 2. Sistem menampilkan seluruh data JSON yang pernah ditambahkan pada halaman <i>Map Data</i> sistem. 3. Admin menekan tanda <i>plus</i>. 4. Sistem menampilkan <i>form</i> tambah data. 5. Admin menginputkan data sesuai dengan kebutuhan <i>form</i>. Kemudian menekan tombol <i>Save</i>. |

Tabel 4.14 Use case scenario menambah data *history* (lanjutan)

| Item | Deskripsi |
|--------------|--|
| Post-Kondisi | Sistem merespon Admin dengan menampilkan notifikasi " <i>Successfully Add Data!</i> " dan menampilkan data yang baru ditambahkan ke halaman <i>Map Data</i> . |
| Alternatif | Apabila data yang diinputkan pada <i>form</i> tidak lengkap, maka sistem akan menampilkan notifikasi " <i>Please fill out this field</i> ". Dan apabila data yang diinputkan tidak sesuai format angka, sistem akan menampilkan notifikasi " <i>Please enter a number</i> ". |

11. Use Case Scenario Menghapus Data History

Pada Tabel 4.15 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional menghapus data *history*.

Tabel 4.15 Use case scenario menghapus data *history*

| Item | Deskripsi |
|---------------|--|
| Nama Use Case | Menghapus data <i>history</i> |
| Deskripsi | <i>Use case</i> ini menjelaskan alur Admin untuk dapat menghapus data JSON yang pernah ditambahkan. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah berhasil masuk ke dalam sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Map Data</i> pada halaman Admin. 2. Sistem menampilkan seluruh data JSON yang pernah ditambahkan ke sistem. 3. Admin menekan tanda hapus. 4. Sistem menampilkan notifikasi "<i>Are you sure to delete all data?</i>". 5. Admin menekan tombol "OK". 6. Sistem menampilkan halaman <i>Map Data</i>, dan data yang telah dihapus tidak akan ditampilkan. |
| Post-Kondisi | Sistem merespon dengan menghapus seluruh data dan data informasi dari database, kemudian data yang telah dihapus tidak akan ditampilkan. |
| Alternatif | Admin dapat menghapus baris data tertentu dengan menekan tombol <i>Delete</i> pada setiap baris data. Kemudian sistem akan menampilkan notifikasi " <i>Are you sure to delete this data?</i> ". Jika Admin menekan tombol OK, sistem akan menghapus data dan data informasi dari tahun yang dipilih. |

12. Use Case Scenario Melihat Data Info History

Pada Tabel 4.16 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional melihat data info *history*.

Tabel 4.16 Use case scenario melihat data info history

| Item | Deskripsi |
|---------------|--|
| Nama Use Case | Melihat data info <i>history</i> |
| Deskripsi | Use case ini menjelaskan alur Admin untuk dapat melihat seluruh data informasi dari peta yang pernah ditambahkan ke sistem. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah berhasil masuk ke sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Info Data</i> pada halaman Admin. 2. Sistem merespon dengan menampilkan seluruh data informasi peta yang pernah ditambahkan ke sistem. |
| Post-Kondisi | Sistem merespon Admin dengan mengambil seluruh data informasi peta dari basis data, kemudian ditampilkan pada halaman <i>Info Data</i> . |
| Alternatif | - |

13. Use Case Scenario Menambah Data Info History

Pada Tabel 4.17 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional menambah data info *history* untuk masing-masing tahun kejadian erupsi yang pernah ditambahkan ke sistem.

Tabel 4.17 Use case scenario menambah data info history

| Item | Deskripsi |
|---------------|---|
| Nama Use Case | Menambah data info <i>history</i> |
| Deskripsi | Use case ini menjelaskan alur Admin untuk dapat menambahkan data informasi peta baru ke sistem. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah berhasil masuk ke dalam sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Info Data</i> pada halaman Admin. 2. Sistem menampilkan seluruh data informasi peta yang pernah ditambahkan ke sistem. 3. Admin menekan tanda <i>plus</i>. |

Tabel 4.17 *Use case scenario* menambah data info *history* (lanjutan)

| Item | Deskripsi |
|--------------|--|
| Tindakan | <ol style="list-style-type: none"> 4. Sistem menampilkan <i>form</i> tambah data informasi. 5. Admin menginputkan data sesuai dengan kebutuhan <i>form</i>. Kemudian menekan tombol <i>Save</i>. 6. Sistem menampilkan notifikasi dan menampilkan data yang baru ditambahkan pada <i>Info Data</i>. |
| Post-Kondisi | Sistem merespon Admin dengan menyimpan data baru pada database, kemudian menampilkan notifikasi " <i>Successfully Add Data!</i> " dan menampilkan data yang baru ditambahkan ke halaman <i>Info Data</i> . |
| Alternatif | Apabila data yang diinputkan pada <i>form</i> tidak lengkap, maka sistem akan menampilkan notifikasi " <i>Please fill out this field</i> ". Dan apabila data yang diinputkan tidak sesuai format angka, maka sistem akan menampilkan notifikasi " <i>Please enter a number</i> ". |

14. Use Case Scenario Menghapus Data Info History

Pada Tabel 4.18 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional menghapus data info *history*.

Tabel 4.18 *Use case scenario* menghapus data info *history*

| Item | Deskripsi |
|---------------|--|
| Nama Use Case | Menghapus data info <i>history</i> |
| Deskripsi | <i>Use case</i> ini menjelaskan alur bagi Admin untuk dapat menghapus data informasi peta yang pernah ditambahkan. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah berhasil masuk ke sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Info Data</i> pada halaman Admin. 2. Sistem menampilkan seluruh data informasi peta. 3. Admin menekan tanda hapus. 4. Sistem menampilkan notifikasi "<i>Are you sure to delete all data?</i>". 5. Admin menekan tombol "OK". 6. Sistem menampilkan halaman <i>Info Data</i>, dan data yang telah dihapus tidak akan ditampilkan. |

Tabel 4.18 Use case scenario menghapus data info *history* (lanjutan)

| Item | Deskripsi |
|--------------|--|
| Post-Kondisi | Sistem merespon dengan menghapus seluruh data informasi dari database, dan kemudian data yang telah dihapus tidak akan ditampilkan pada halaman Info Data. |
| Alternatif | Admin dapat menghapus baris data tertentu dengan menekan tombol <i>Delete</i> pada setiap baris data. Kemudian sistem akan menampilkan notifikasi "Are you sure to delete this data?". Jika Admin menekan tombol OK, maka sistem akan menghapus data informasi yang dipilih. |

15. Use Case Scenario Mengedit Data Info *History*

Pada Tabel 4.19 merupakan skenario untuk Admin dan sistem ketika melakukan kebutuhan fungsional mengedit data info *history*, sehingga dapat memperbarui data informasi yang terdapat di sistem.

Tabel 4.19 Use case scenario mengedit data info *history*

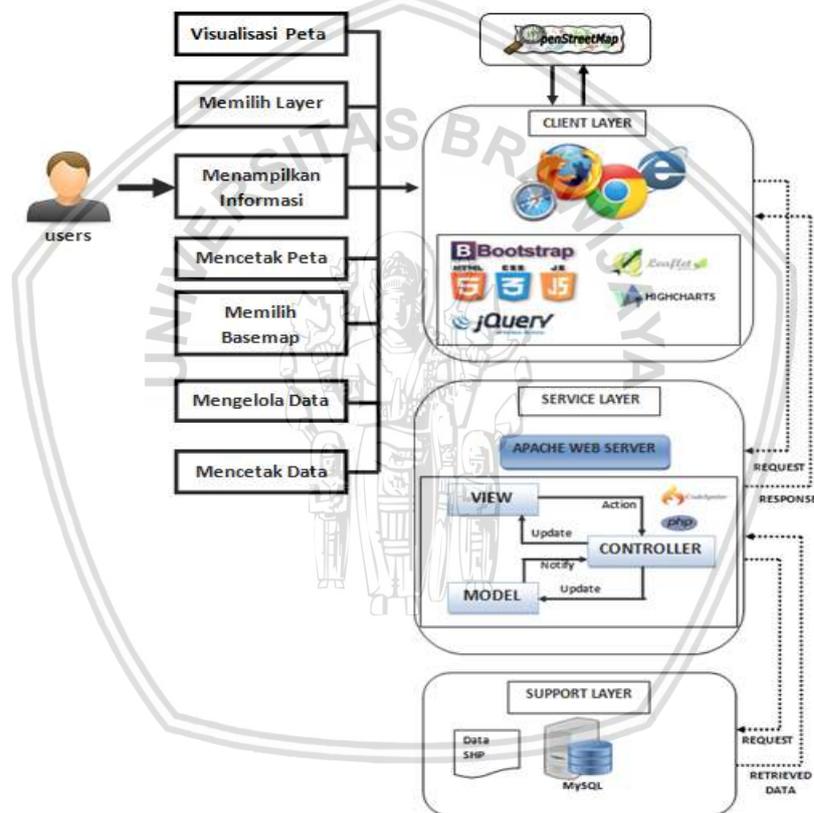
| Item | Deskripsi |
|---------------|--|
| Nama Use Case | Mengedit data info <i>history</i> |
| Deskripsi | Use case ini menjelaskan alur bagi Admin untuk dapat memperbarui data informasi peta yang pernah ditambahkan ke sistem. |
| Aktor | Admin |
| Pra-Kondisi | Admin telah berhasil masuk ke sistem |
| Tindakan | <ol style="list-style-type: none"> 1. Admin memilih menu <i>Info Data</i> pada halaman Admin. 2. Sistem menampilkan seluruh data informasi peta yang pernah ditambahkan ke sistem. 3. Admin menekan tombol <i>Edit</i> pada data tertentu. 4. Sistem menampilkan <i>form edit</i> data informasi. 5. Admin menginputkan data baru sesuai dengan kebutuhan <i>form</i>. Kemudian menekan tombol <i>Save</i>. |
| Post-Kondisi | Sistem merespon kepada Admin dengan menampilkan notifikasi "Successfully Update Data!" dan menampilkan data yang telah diperbarui ke halaman <i>Info Data</i> . |
| Alternatif | Apabila data yang diinputkan pada <i>form</i> tidak lengkap, sistem akan menampilkan notifikasi "Please fill out this field". Apabila data yang diinputkan tidak sesuai format angka, sistem akan menampilkan notifikasi "Please enter a number". |

4.2 Perancangan

Perancangan sistem yang dibuat menggunakan pendekatan berorientasi objek (OOP), dan akan dimodelkan dengan menggunakan *Unified Model Language* (UML). Perancangan yang akan dibuat antara lain perancangan arsitektural, perancangan diagram aktivitas, perancangan diagram sequence, perancangan diagram kelas, perancangan basis data, dan perancangan antarmuka.

4.2.1 Perancangan Arsitektur

Perancangan arsitektur merupakan pembuatan arsitektur dari sistem untuk menggambarkan kinerja dan peran dari setiap komponen yang terdapat pada sistem. Berikut pada Gambar 4.2 merupakan perancangan arsitektur dari sistem.



Gambar 4.2 Perancangan arsitektural

Berdasarkan gambar diatas, arsitektur yang dibuat terdiri dari 3 *layer* yaitu *Client Layer*, *Service Layer*, dan *Support Layer*. Berikut di bawah ini merupakan penjelasan dari setiap *layer* yang terdapat pada perancangan arsitektural.

1. *Client Layer*

Berisi *web browser* yang digunakan sebagai perantara untuk menampilkan antarmuka pengguna, sehingga pengguna dan sistem dapat berinteraksi. Dalam menampilkan *user interface*, sistem didukung dengan beberapa komponen antara lain *Bootstrap*, HTML, CSS, JS, *jQuery*, *Leaflet*, *Highcharts*, dan *OpenStreetMap*.



2. *Service Layer*

Pada *service layer* terdapat beberapa komponen yaitu *Apache* sebagai *web server*, *framework CodeIgniter*, dan bahasa pemrograman PHP. Dalam *framework CodeIgniter*, kinerja dari sistem menggunakan konsep MVC (*Model-View-Controller*). *View* berkaitan dengan *user interface* dari sistem, dimana permintaan dari *user* muncul. *Controller* berfungsi untuk menghubungkan antara *View* dan *Model*. Sehingga pada *Controller* juga berisi fungsi-fungsi yang menghubungkan antara permintaan pada *View* dengan *Model*. Sedangkan *Model* berfungsi untuk mengakses basis data dengan melakukan aktivitas seperti *SELECT*, *DELETE*, *INSERT*, dan *UPDATE*. Data yang didapatkan oleh *Model* akan dikembalikan ke *Controller* untuk ditampilkan pada *View*.

3. *Support Layer*

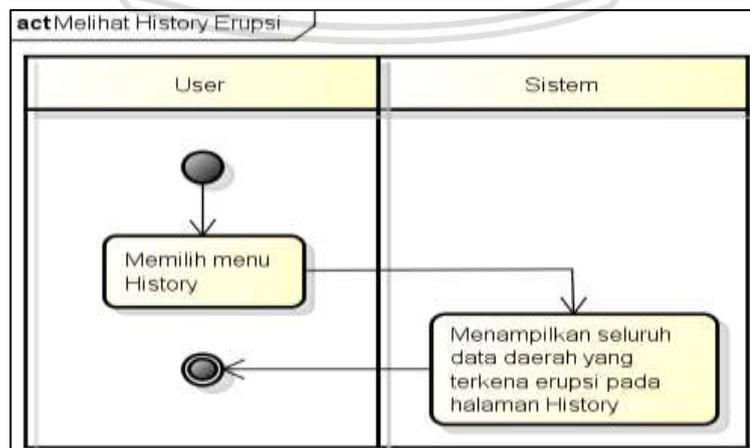
Support Layer terdiri dari *MySQL* yang merupakan DBMS atau *database server* yang digunakan untuk menyimpan data-data yang akan digunakan oleh sistem dan dipanggil oleh bagian *Model*. Data yang disimpan pada basis data disimpan dalam tabel-tabel yang dikelompokkan berdasarkan jenis kebutuhan data. Terdapat 3 tabel yang digunakan pada sistem yaitu tabel data, tabel info, dan tabel *user*.

4.2.2 Diagram Aktivitas

Diagram aktivitas merupakan diagram untuk memberikan gambaran dari aktivitas yang terdapat pada sistem. Diagram aktivitas dibuat berdasarkan skenario yang telah dibuat pada tahap sebelumnya.

1. **Diagram Aktivitas Melihat *History* Erupsi**

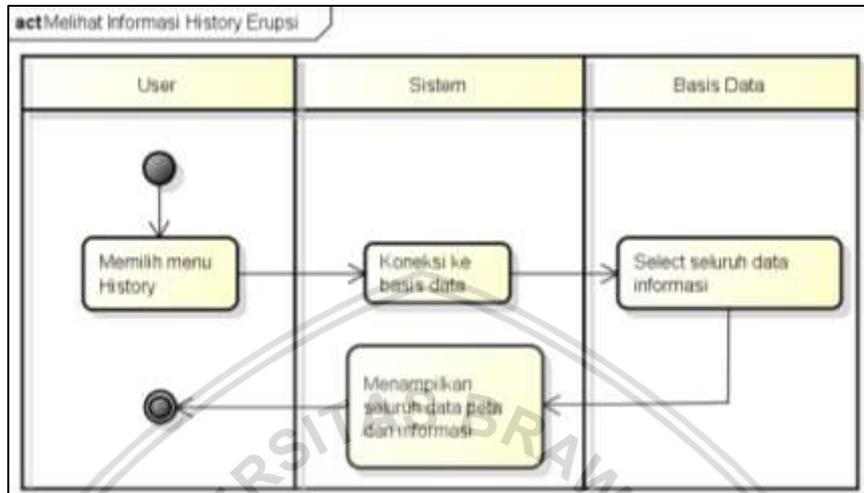
Pada Gambar 4.3 merupakan diagram aktivitas dari sistem ketika terdapat permintaan dari *user* untuk melihat *history* erupsi Gunung Merapi. Aktivitas sistem akan dimulai ketika *user* menekan menu *History* pada sistem. Dan diakhiri dengan sistem yang menampilkan seluruh data pada halaman *History*.



Gambar 4.3 Diagram aktivitas melihat *history* erupsi

2. Diagram Aktivitas Melihat Informasi *History* Erupsi

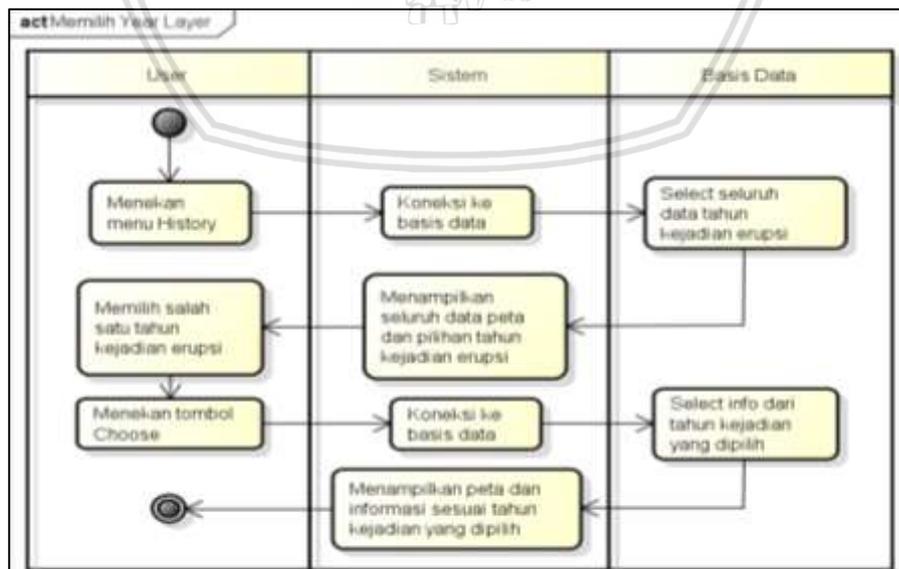
Pada Gambar 4.4 merupakan diagram aktivitas untuk melihat informasi *history* erupsi. Aktivitas akan dimulai ketika *user* memilih menu *History* dan diakhiri dengan aktivitas sistem yang menampilkan data serta informasi dari tahun kejadian yang dipilih.



Gambar 4.4 Diagram aktivitas melihat informasi *history* erupsi

3. Diagram Aktivitas Memilih *Year Layer*

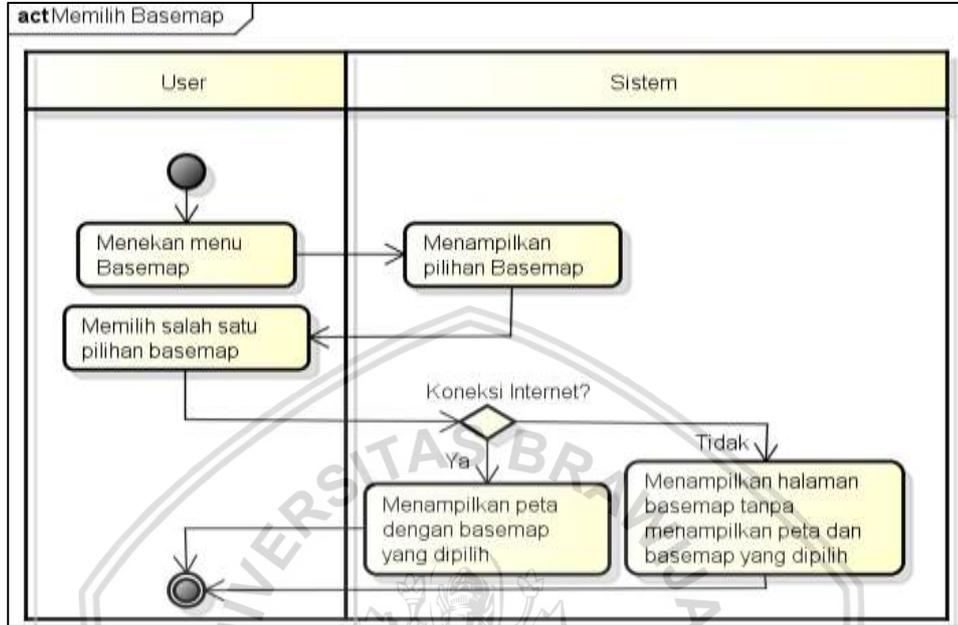
Berikut ini pada Gambar 4.5 adalah diagram aktivitas dari memilih *year layer*. Diagram aktivitas ini digunakan untuk menjelaskan apa saja yang harus dilakukan untuk memilih peta berdasarkan kejadian pada tahun tertentu sesuai dengan pilihan *user*. Aktivitas akan dimulai ketika *user* menekan menu *History*, kemudian diakhiri dengan aktivitas sistem yang menampilkan peta hasil pemetaan sesuai dengan tahun kejadian yang dipilih oleh *user*.



Gambar 4.5 Diagram aktivitas memilih *year layer*

4. Diagram Aktivitas Memilih *Basemap*

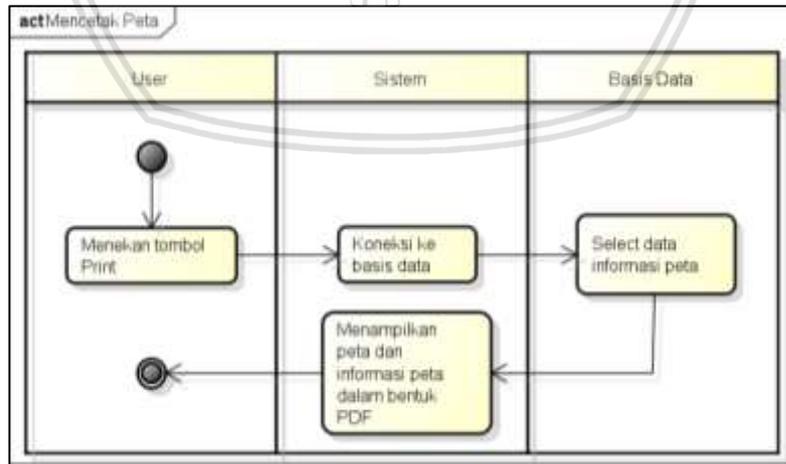
Berikut ini pada Gambar 4.6 merupakan diagram aktivitas yang menjelaskan apa saja yang harus dilakukan untuk menampilkan peta hasil pemetaan sesuai dengan *basemap* yang dipilih oleh *user*.



Gambar 4.6 Diagram aktivitas memilih *basemap*

5. Diagram Aktivitas Mencetak Peta

Berikut pada Gambar 4.7 merupakan diagram aktivitas dari kebutuhan fungsional mencetak peta. Diagram ini menggambarkan aktivitas untuk mencetak peta hasil pemetaan daerah yang terkena erupsi Gunung Merapi yang ditampilkan pada sistem.

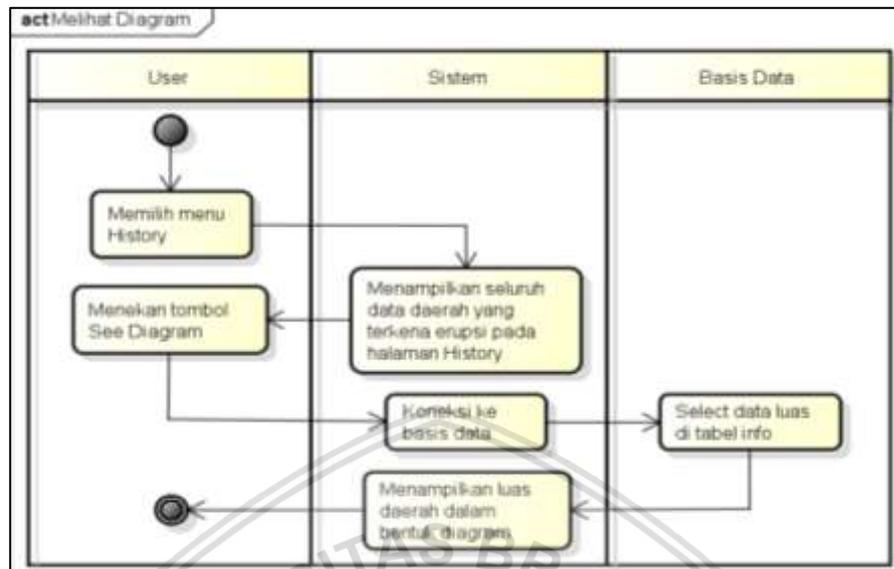


Gambar 4.7 Diagram aktivitas mencetak peta

6. Diagram Aktivitas Melihat Diagram

Pada Gambar 4.8 merupakan diagram aktivitas untuk melihat diagram yang menunjukkan persentase luasan daerah hasil pemetaan. Aktivitas dimulai

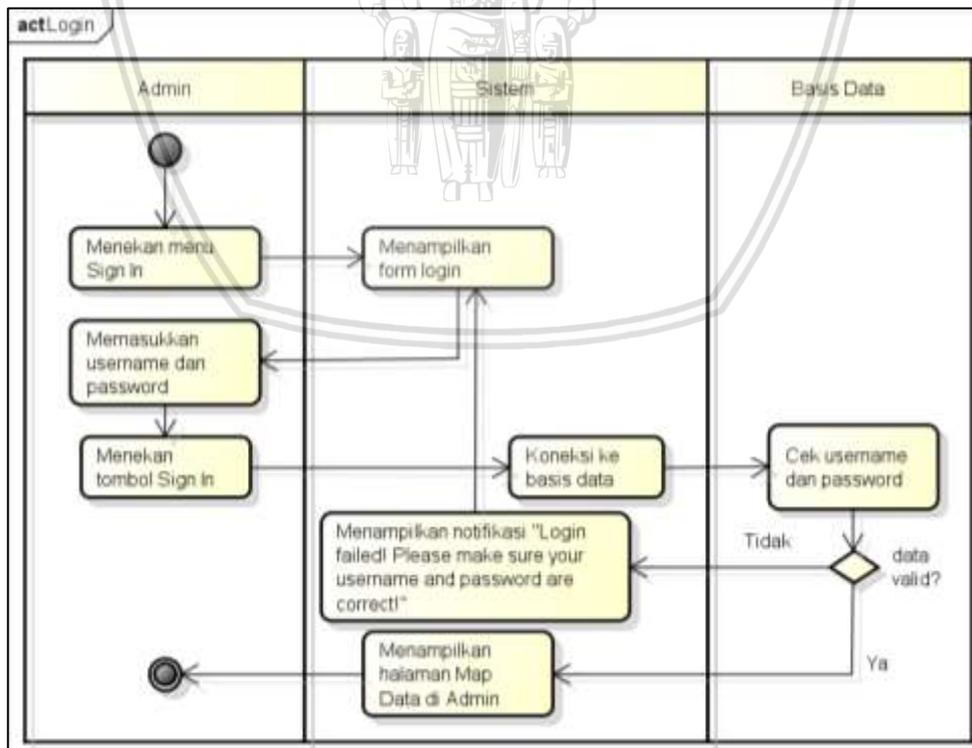
dengan *user* ketika memilih menu *History*. Kemudian diakhiri dengan aktivitas sistem yang menampilkan luasan daerah dalam bentuk diagram.



Gambar 4.8 Diagram aktivitas melihat diagram

7. Diagram Aktivitas Login

Diagram aktivitas dari kebutuhan fungsional *login* ditunjukkan pada Gambar 4.9 berikut. Diagram ini menjelaskan apa saja yang dilakukan oleh Admin agar bisa masuk ke sistem.

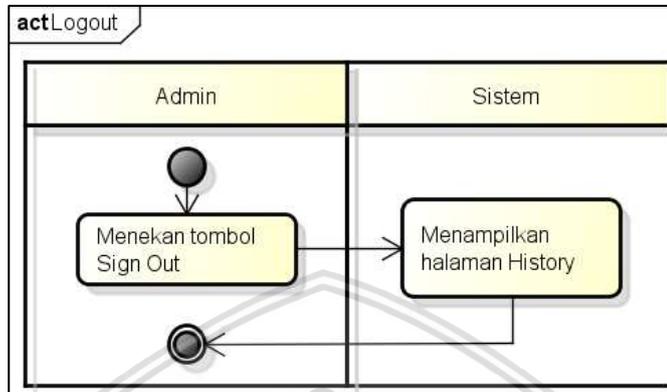


Gambar 4.9 Diagram aktivitas login



8. Diagram Aktivitas Logout

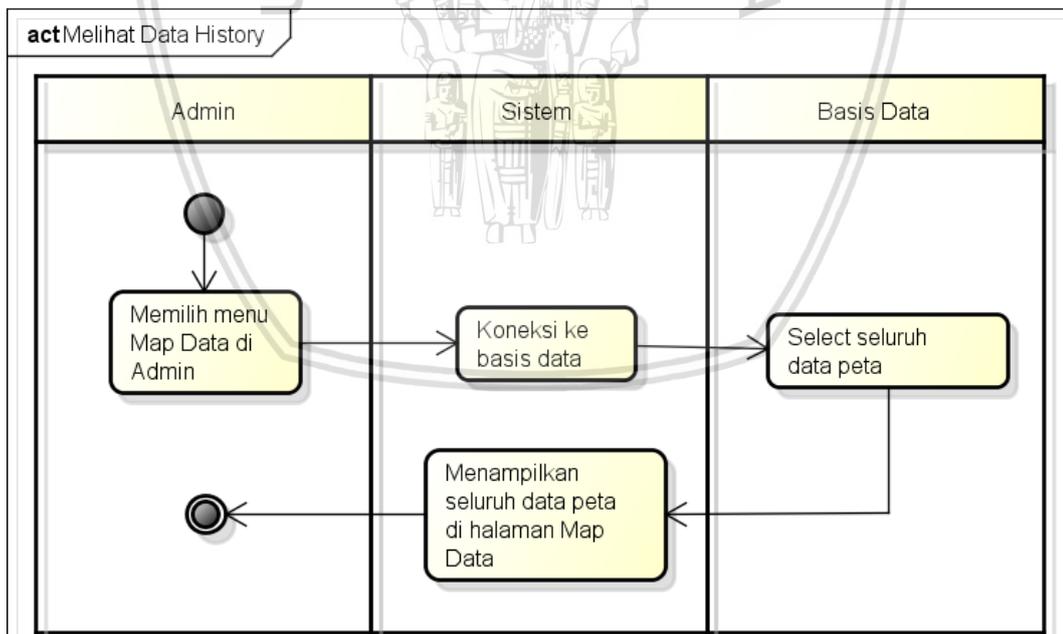
Berikut pada Gambar 4.10 menggambarkan aktivitas yang dilakukan untuk keluar (*logout*) dari sistem. Aktivitas diawali ketika Admin menekan tombol *Sign Out*, kemudian sistem akan menampilkan halaman *History* ketika aktivitas ini berhasil.



Gambar 4.10 Diagram aktivitas *logout*

9. Diagram Aktivitas Melihat Data History

Berikut pada Gambar 4.11 merupakan diagram yang menjelaskan apa saja yang dilakukan oleh Admin, sistem, dan basis data untuk menampilkan data JSON dari hasil pemetaan yang pernah ditambahkan ke sistem.

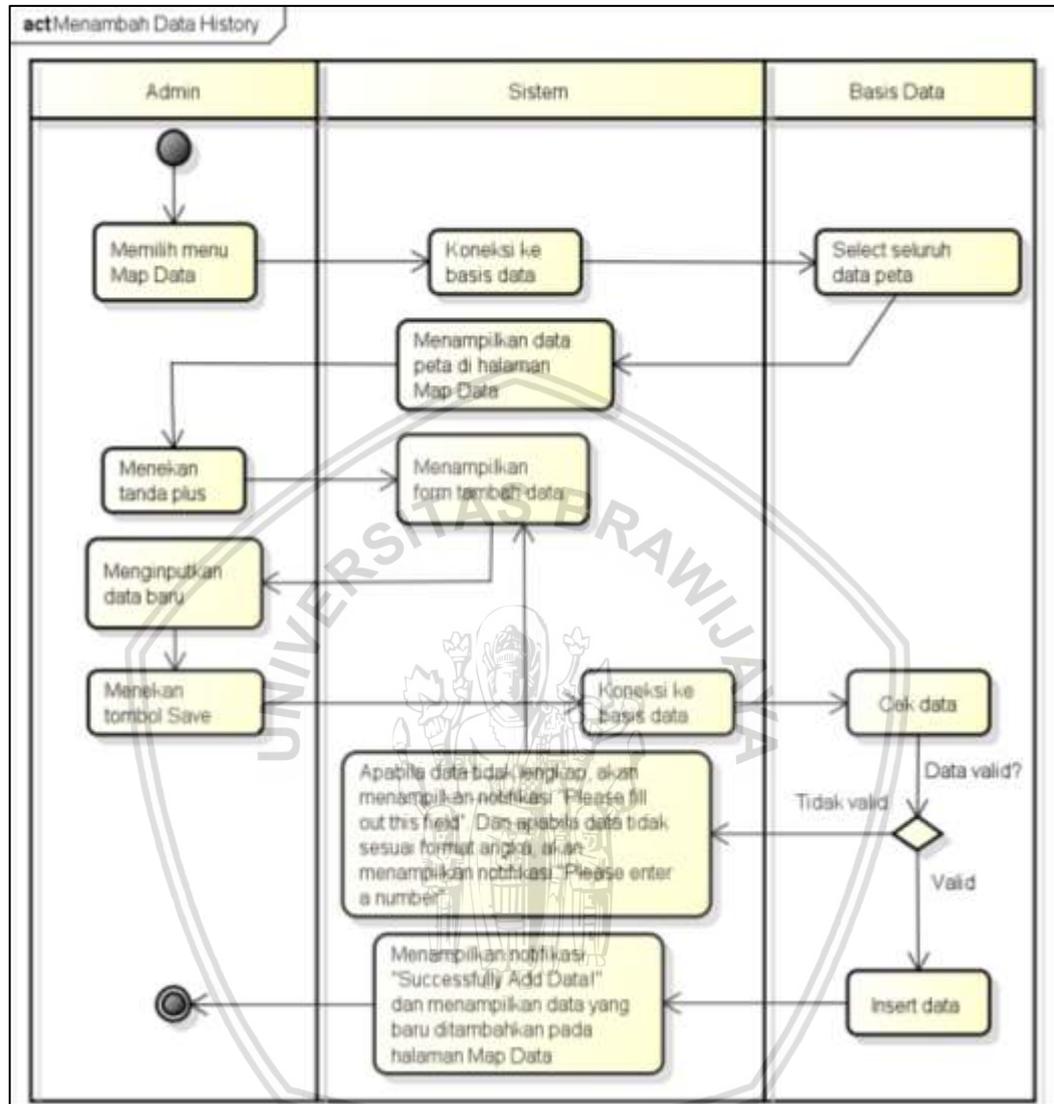


Gambar 4.11 Diagram aktivitas melihat data *history*

10. Diagram Aktivitas Menambah Data History

Berikut pada Gambar 4.12 merupakan diagram yang menggambarkan aktivitas untuk menambah data JSON dari hasil pemetaan yang dapat digunakan untuk memvisualisasikan peta pada halaman *History*. Aktivitas dimulai oleh

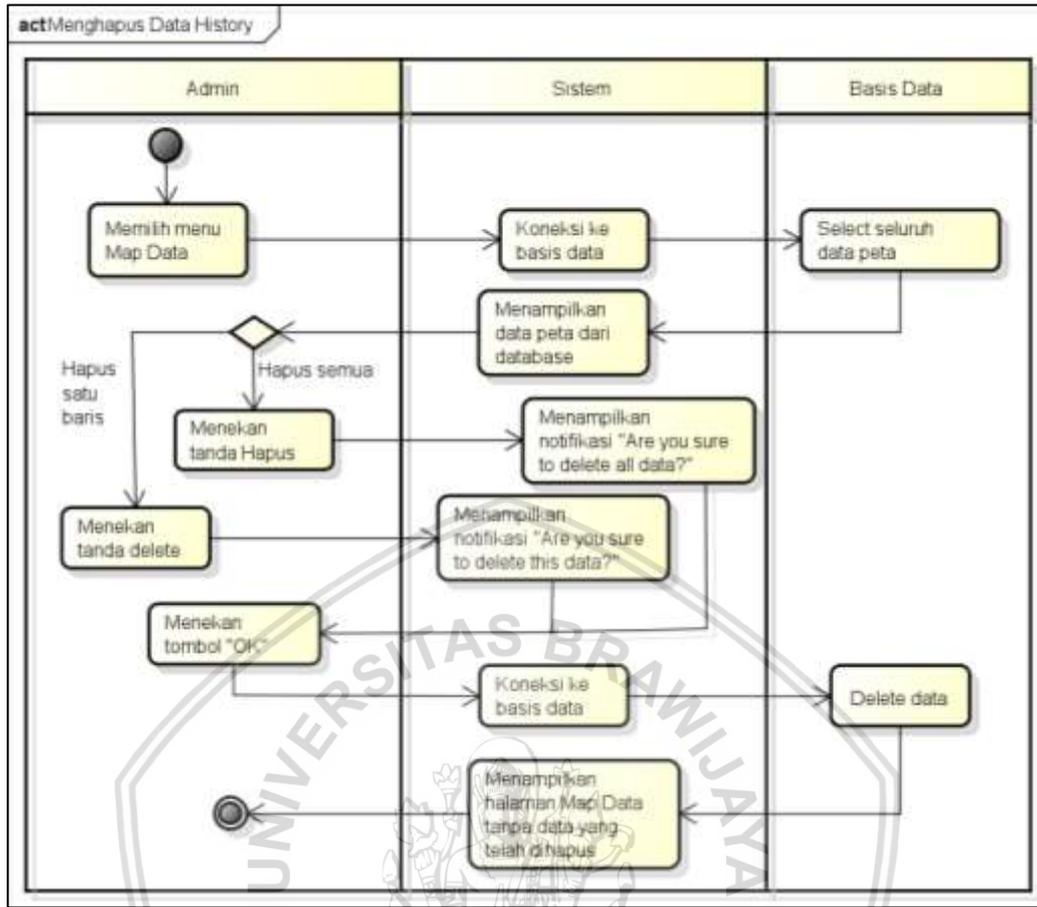
Admin yang menekan menu *Map Data*. Dan diakhiri dengan sistem yang menampilkan notifikasi "*Successfully add data!*" serta menampilkan data yang ditambahkan pada halaman *Map Data*.



Gambar 4.12 Diagram aktivitas menambah data *history*

11. Diagram Aktivitas Menghapus Data *History*

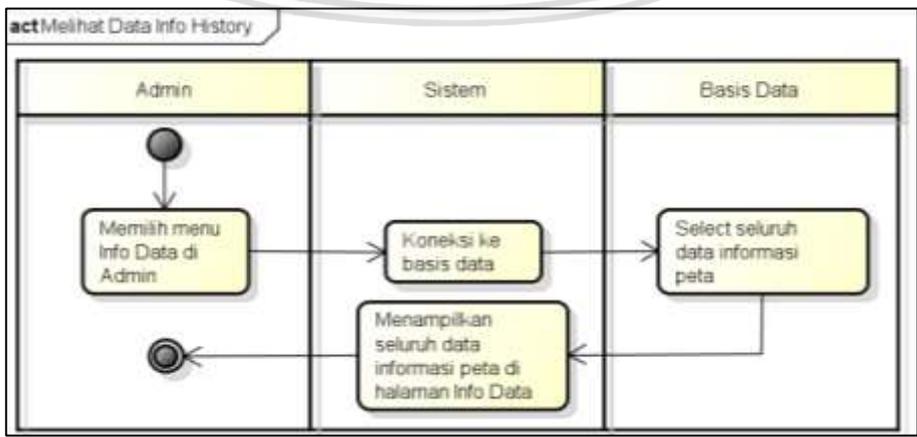
Berikut pada Gambar 4.13 merupakan diagram yang menjelaskan aktivitas apa saja yang dilakukan untuk menghapus data JSON yang ada di sistem. Aktivitas diawali ketika Admin memilih menu *Map Data*. Kemudian sistem akan menampilkan seluruh data JSON yang diambil dari basis data ke halaman *Map Data*. Untuk menghapus data, Admin akan menekan tanda hapus. Kemudian sistem akan menampilkan notifikasi "*Are you sure to delete this data?*". Jika Admin menekan tombol OK, maka data akan dihapus dari basis data. Dan data yang sudah dihapus tidak ditampilkan lagi di halaman *Map Data*.



Gambar 4.13 Diagram aktivitas menghapus data *history*

12. Diagram Aktivitas Melihat Data Info *History*

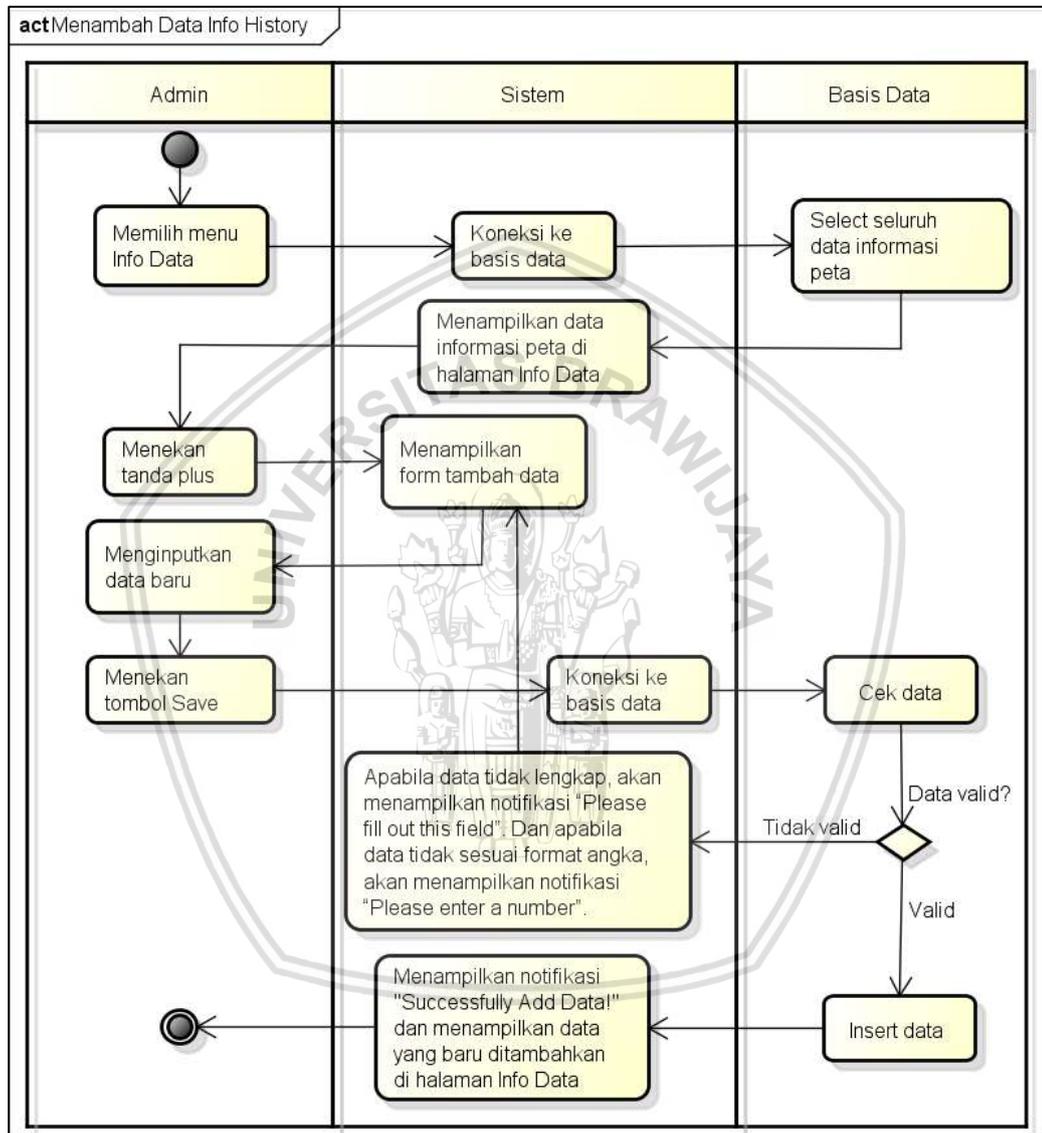
Berikut pada Gambar 4.14 merupakan diagram yang menggambarkan aktivitas untuk menampilkan data informasi peta yang pernah ditambahkan ke sistem. Aktivitas akan dimulai ketika Admin memilih menu *Info Data*, dan diakhiri dengan sistem yang menampilkan seluruh informasi peta pada halaman *Info Data*.



Gambar 4.14 Diagram aktivitas melihat data info *history*

13. Diagram Aktivitas Menambah Data Info History

Berikut pada Gambar 4.15 merupakan diagram yang menggambarkan aktivitas untuk menambah data informasi peta pada halaman *Info Data*. Aktivitas diawali ketika Admin menekan menu *Info Data*. Dan diakhiri dengan sistem yang menampilkan notifikasi “*Successfully add data!*” serta menampilkan data yang ditambahkan pada halaman *Info Data*.

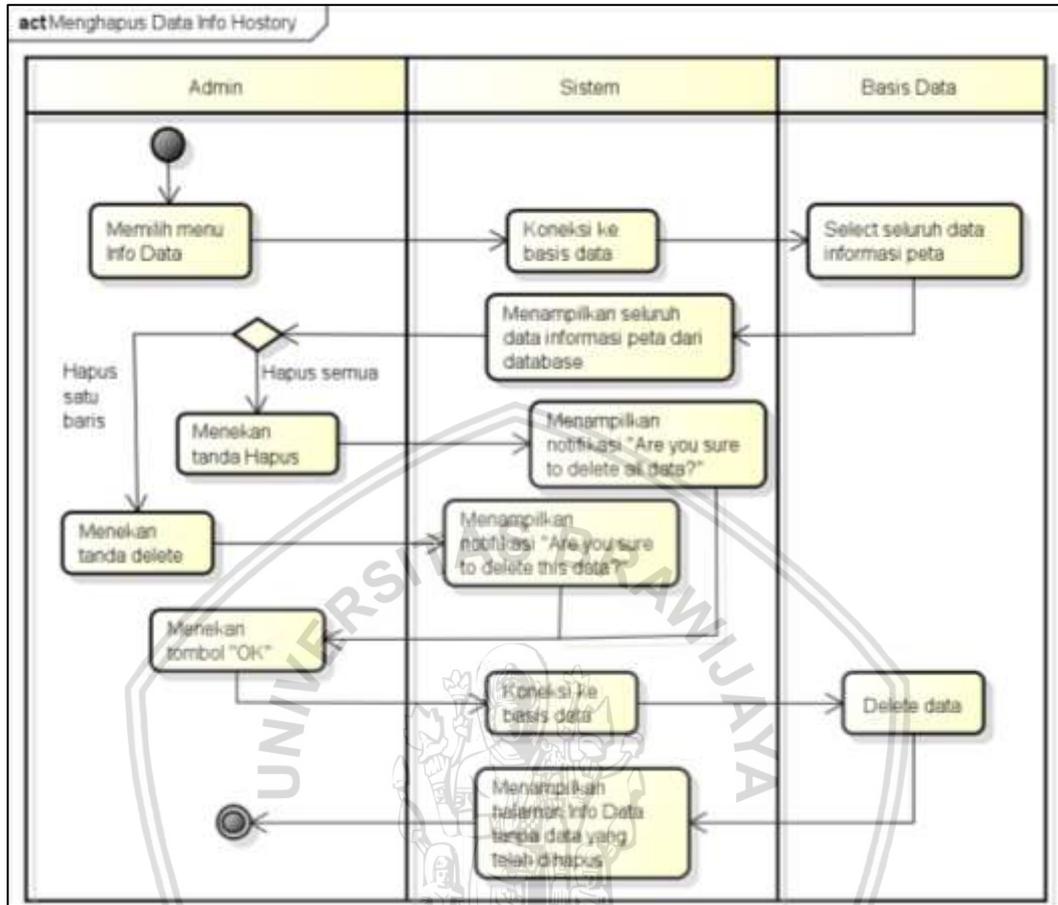


Gambar 4.15 Diagram aktivitas menambah data info history

14. Diagram Aktivitas Menghapus Data Info History

Berikut pada Gambar 4.16 merupakan diagram yang menggambarkan aktivitas untuk menghapus data informasi peta yang sudah pernah ditambahkan ke sistem. Aktivitas dimulai ketika Admin memilih menu *Info Data*. Kemudian sistem akan menampilkan seluruh data informasi peta yang diambil dari basis data ke halaman *Info Data*. Untuk menghapus data, Admin akan menekan tanda hapus. Kemudian sistem akan menampilkan notifikasi “*Are you sure to delete this*

data?”. Jika Admin menekan tombol OK, maka data akan dihapus dari basis data. Dan data yang sudah dihapus tidak akan ditampilkan lagi di halaman *Info Data*.

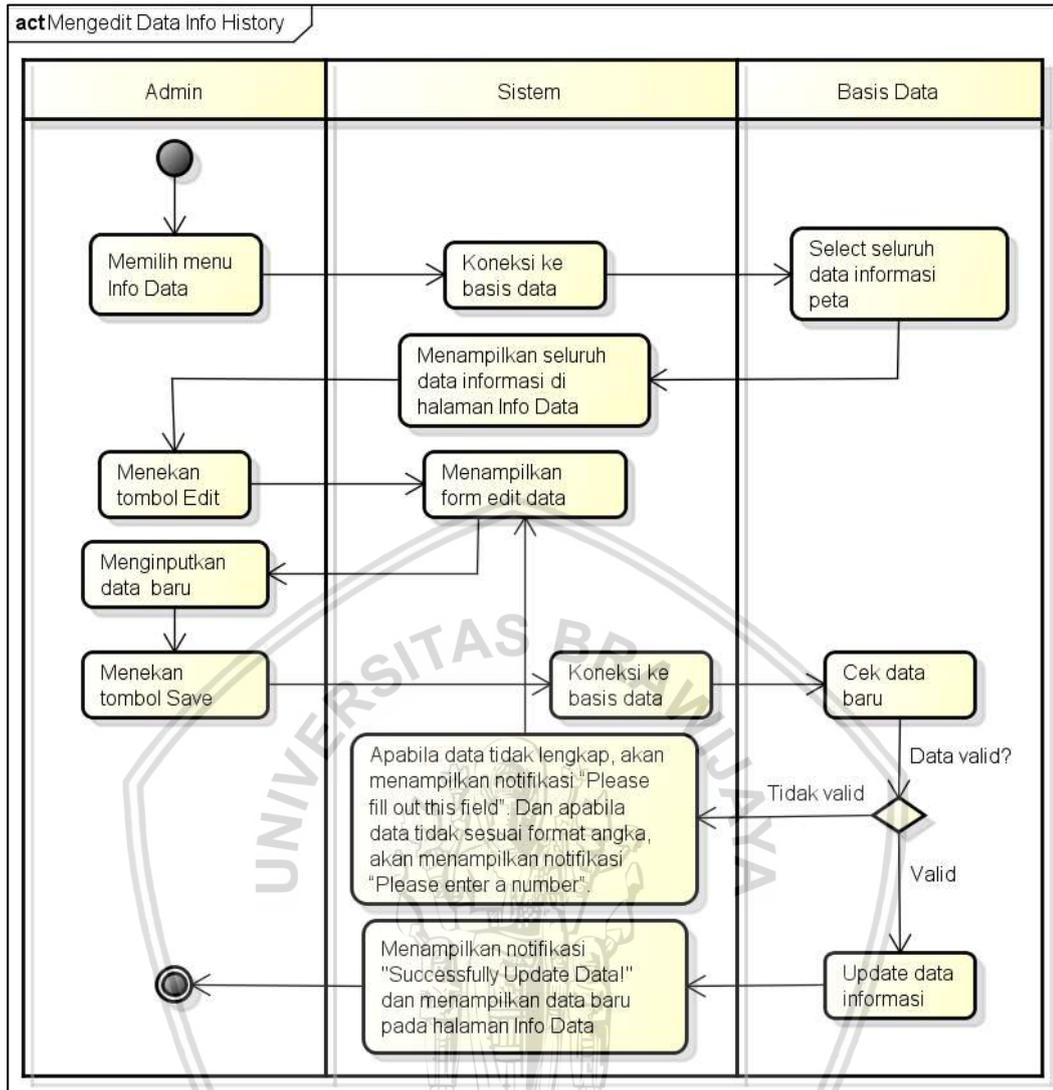


Gambar 4.16 Diagram aktivitas menghapus data info *history*

15. Diagram Aktivitas Mengedit Data Info *History*

Berikut pada Gambar 4.17 merupakan diagram yang menggambarkan aktivitas untuk mengedit data informasi peta yang sudah pernah ditambahkan ke sistem. Aktivitas diawali ketika Admin memilih menu *Info Data*. Kemudian sistem akan menampilkan seluruh data informasi peta yang diambil dari basis data ke halaman *Info Data*. Untuk mengedit data, Admin akan menekan tombol *edit*. Kemudian sistem akan menampilkan *input form* untuk mengedit data. Admin akan menginputkan data baru ke form, dan menekan tombol simpan. Aktivitas diakhiri oleh sistem yang menampilkan data yang telah diperbarui di halaman *Info Data*.





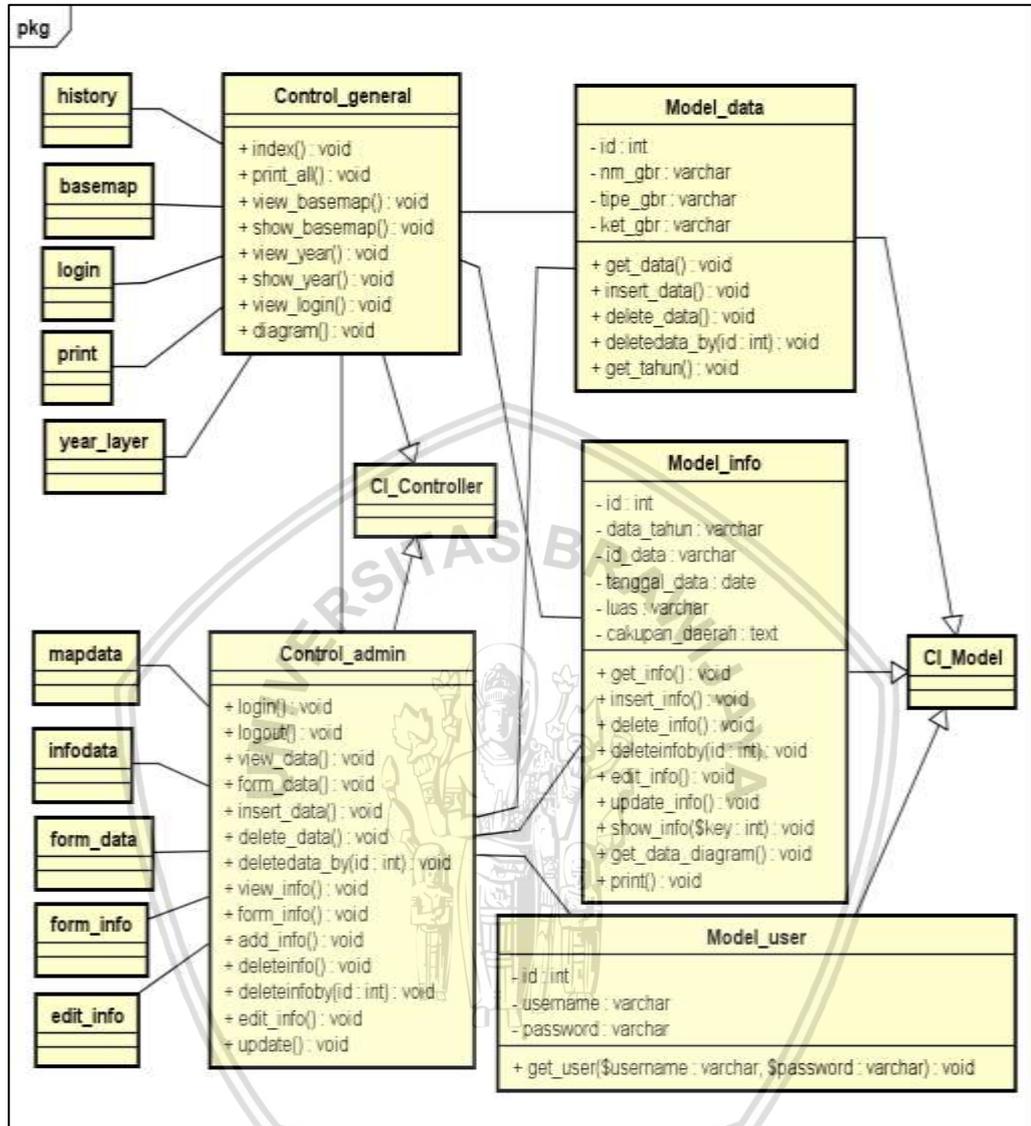
Gambar 4.17 Diagram aktivitas mengedit data info history

4.2.3 Class Diagram

Perancangan *class diagram* dibuat untuk mengetahui komponen yang menyusun sistem dan mengetahui bagaimana interaksi antar komponen. Diagram kelas dari sistem ditunjukkan pada Gambar 4.18. Pada diagram kelas yang dibuat, terdiri dari kelas-kelas Model dan Controller. Terdapat 3 kelas Model, yaitu kelas Model_data, Model_info, dan Model_user. Kelas Model_data berisi fungsi-fungsi untuk mengakses basis data yang berkaitan dengan penyimpanan data JSON dari hasil pemetaan. Kelas Model_info berisi fungsi-fungsi untuk mengakses basis data yang berkaitan dengan info dari data. Sedangkan kelas Model_user berisi fungsi-fungsi untuk mengakses basis data yang berkaitan dengan penyimpanan data Admin.

Pada Controller, terdapat 2 kelas Controller yaitu Control_admin dan Control_general. Kelas Control_admin digunakan untuk meletakkan fungsi-fungsi yang berkaitan dengan aktivitas-aktivitas yang dapat dilakukan oleh Admin saja.

Sedangkan kelas `Control_general` berisi fungsi-fungsi yang berkaitan dengan aktivitas-aktivitas yang dapat dilakukan oleh pengguna umum sistem.



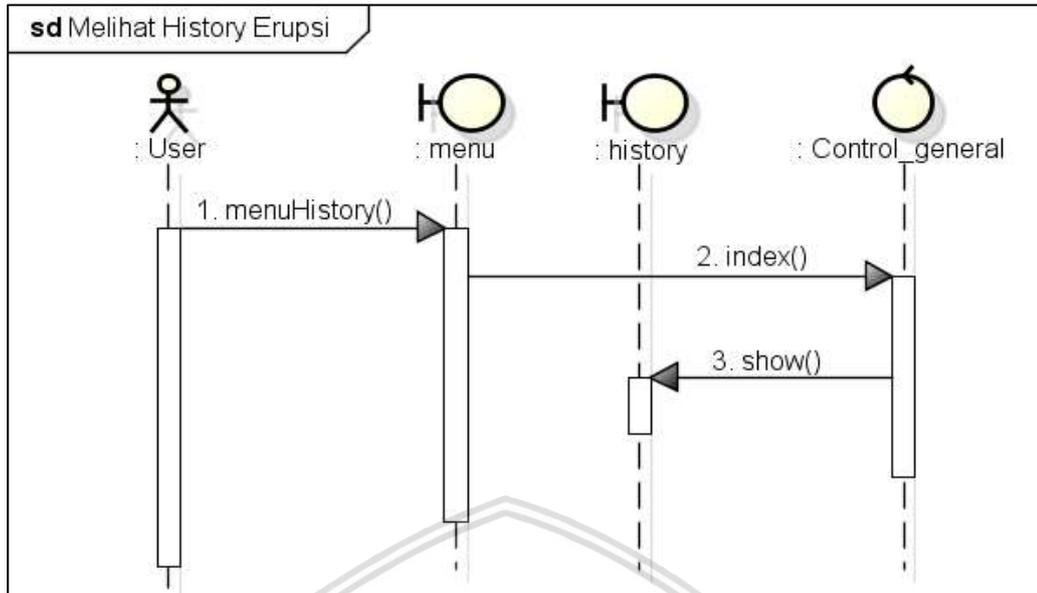
Gambar 4.18 Class diagram

4.2.4 Sequence Diagram

Sequence diagram digunakan untuk memberikan gambaran tentang interaksi antar komponen dari kode program yang akan dibuat. Pada diagram ini juga akan digambarkan perilaku dari objek melalui pesan yang dikirimkan dan diterima antar objek. Berikut ini merupakan perancangan diagram *sequence* dari setiap *use case*.

1. Sequence Diagram Melihat History Erupsi

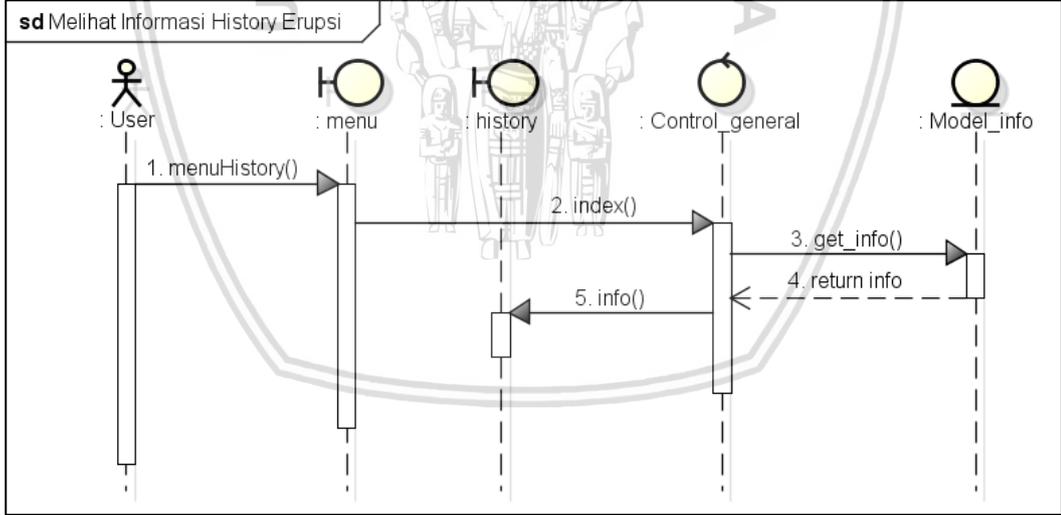
Pada Gambar 4.19 merupakan *sequence diagram* dari *use case* melihat *history* erupsi. Diagram ini menunjukkan aliran kode program untuk menampilkan seluruh peta hasil pemetaan daerah yang terkena erupsi Gunung Merapi di Kabupaten Sleman.



Gambar 4.19 Sequence diagram melihat history erupsi

2. Sequence Diagram Melihat Informasi History Erupsi

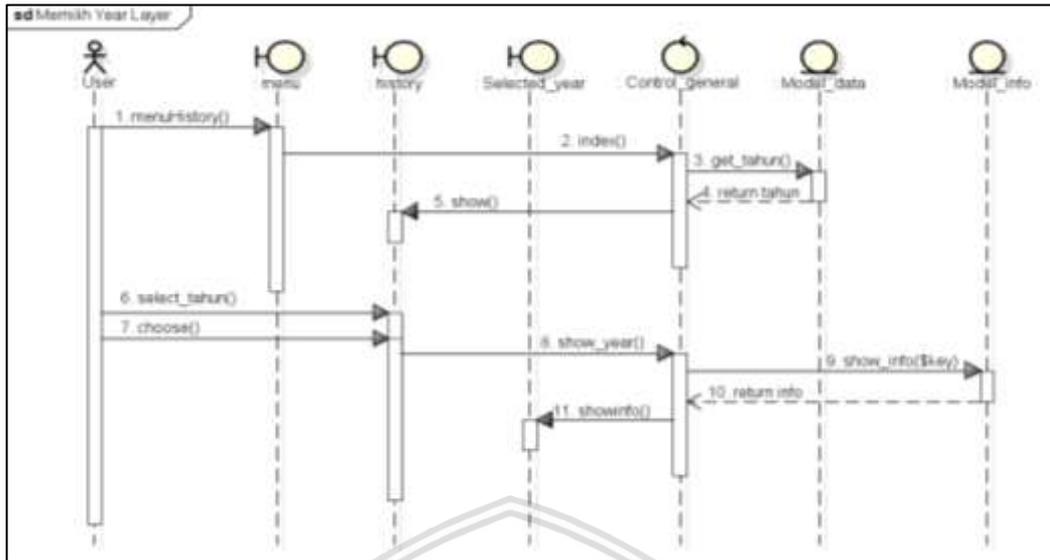
Pada Gambar 4.20 merupakan sequence diagram dari use case melihat informasi history erupsi. Diagram ini menunjukkan aliran kode program untuk menampilkan seluruh informasi peta yang ditampilkan.



Gambar 4.20 Sequence diagram melihat informasi history erupsi

3. Sequence Diagram Memilih Year Layer

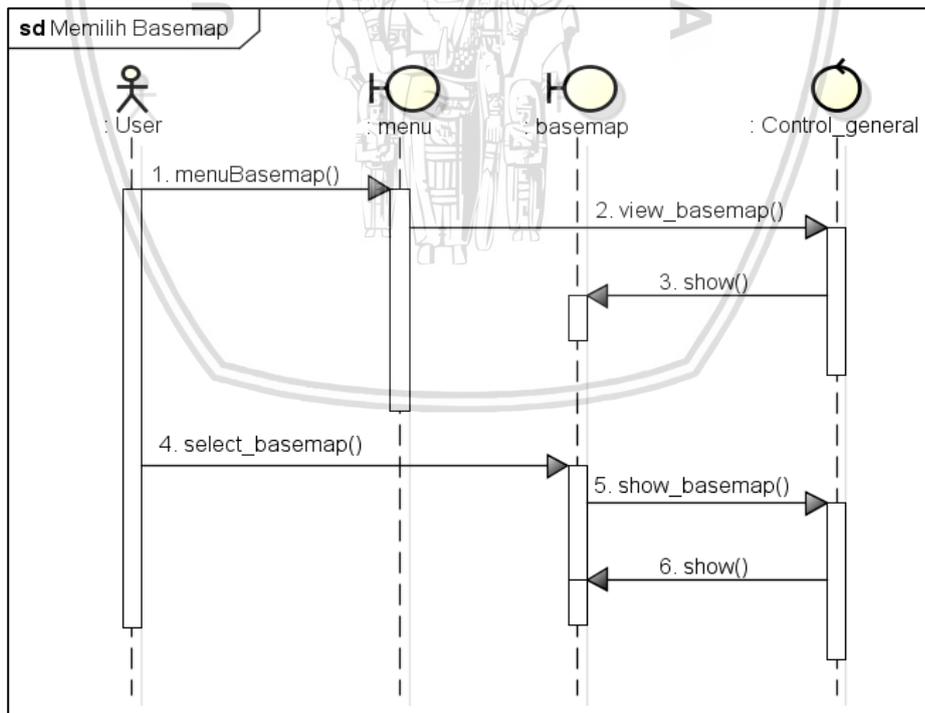
Pada Gambar 4.21 merupakan sequence diagram dari use case memilih year layer. Diagram ini menunjukkan aliran kode program untuk memilih peta hasil pemetaan berdasarkan pada tahun kejadian tertentu.



Gambar 4.21 Sequence diagram memilih year layer

4. Sequence Diagram Memilih Basemap

Pada Gambar 4.22 merupakan *sequence diagram* dari *use case* memilih *basemap*. Diagram ini menunjukkan aliran kode program untuk memvisualisasikan hasil pemetaan daerah yang terkena erupsi sesuai dengan *basemap* yang dipilih user.

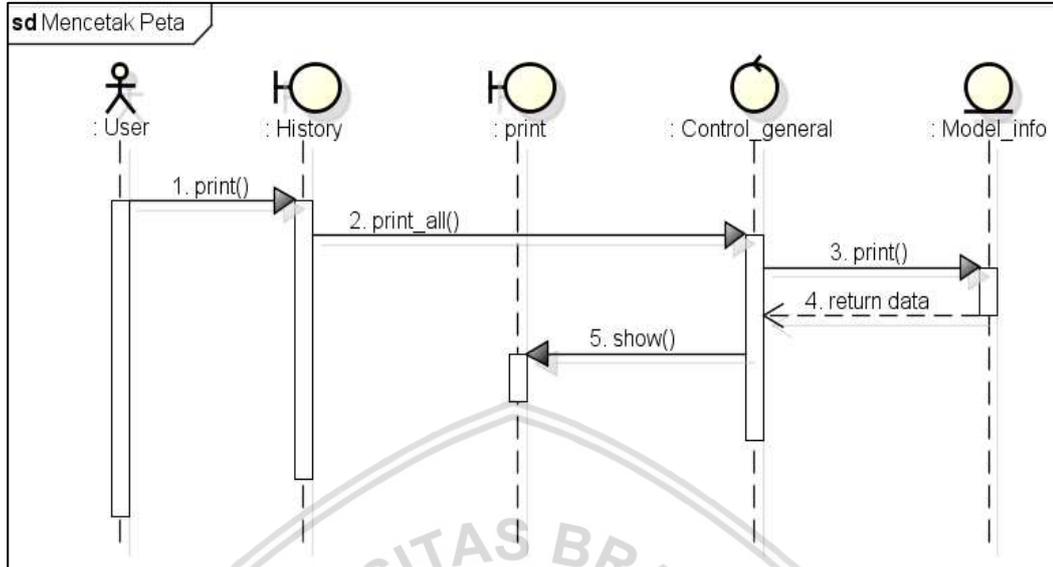


Gambar 4.22 Sequence diagram memilih basemap

5. Sequence Diagram Mencetak Peta

Pada Gambar 4.23 merupakan *sequence diagram* dari *use case* mencetak peta. Diagram ini menunjukkan aliran kode program untuk mencetak seluruh

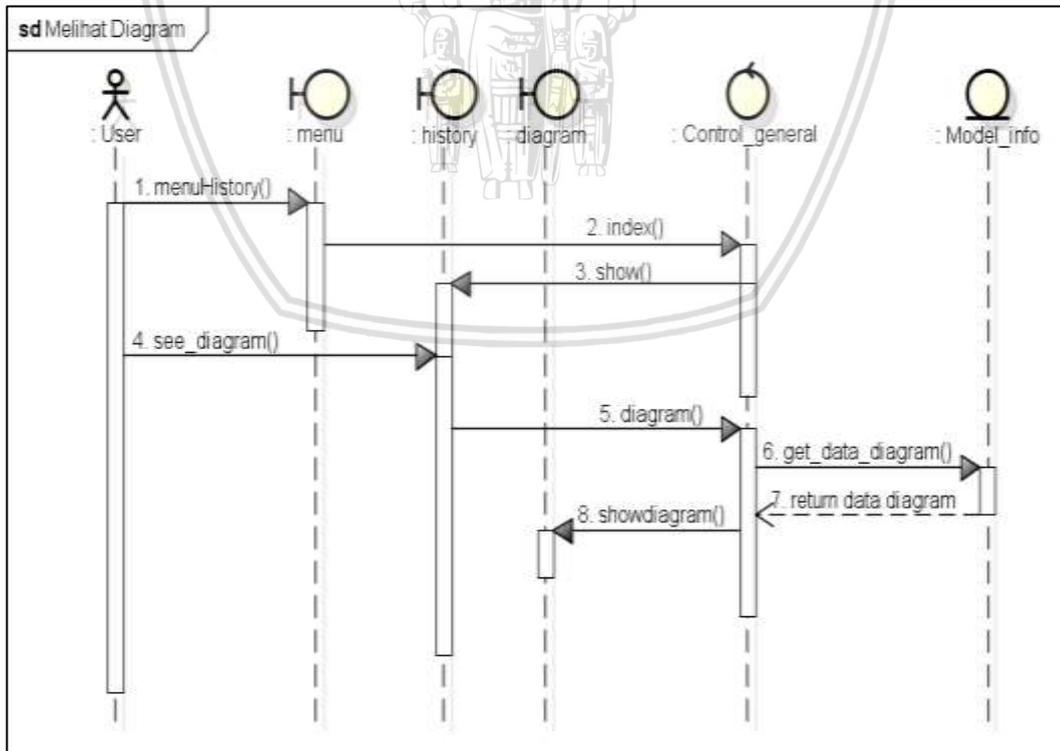
hasil pemetaan daerah yang terkena erupsi beserta informasi dari peta yang terdapat pada sistem.



Gambar 4.23 Sequence diagram mencetak peta

6. Sequence Diagram Melihat Diagram

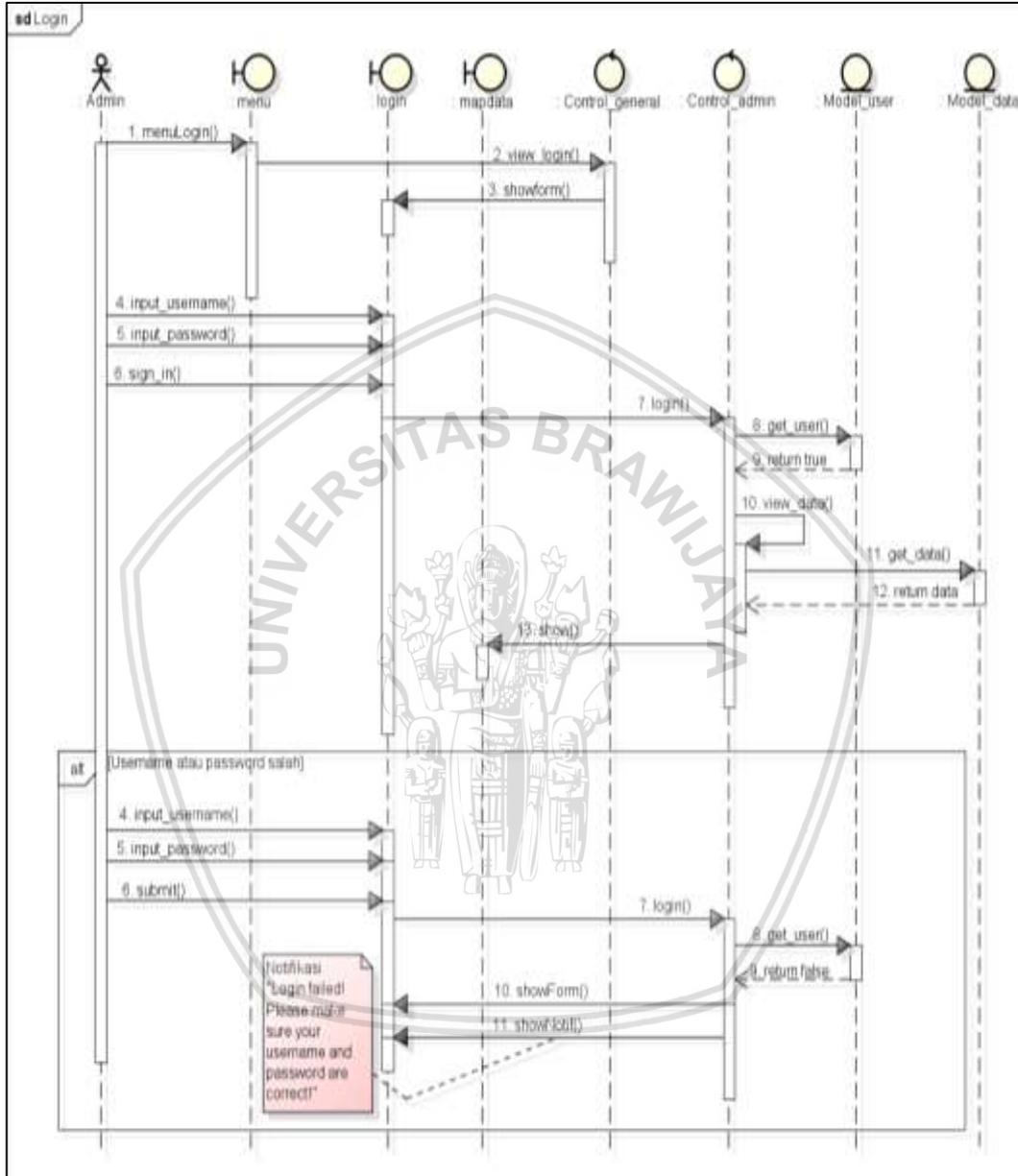
Pada Gambar 4.24 merupakan *sequence diagram* dari *use case* melihat diagram. Diagram ini menunjukkan aliran kode program untuk menampilkan diagram yang menunjukkan luasan daerah hasil pemetaan erupsi.



Gambar 4.24 Sequence diagram melihat diagram

7. Sequence Diagram Login

Pada Gambar 4.25 merupakan *sequence diagram* dari *use case login*. Diagram ini menunjukkan aliran kode program untuk Admin yang ingin masuk ke sistem.

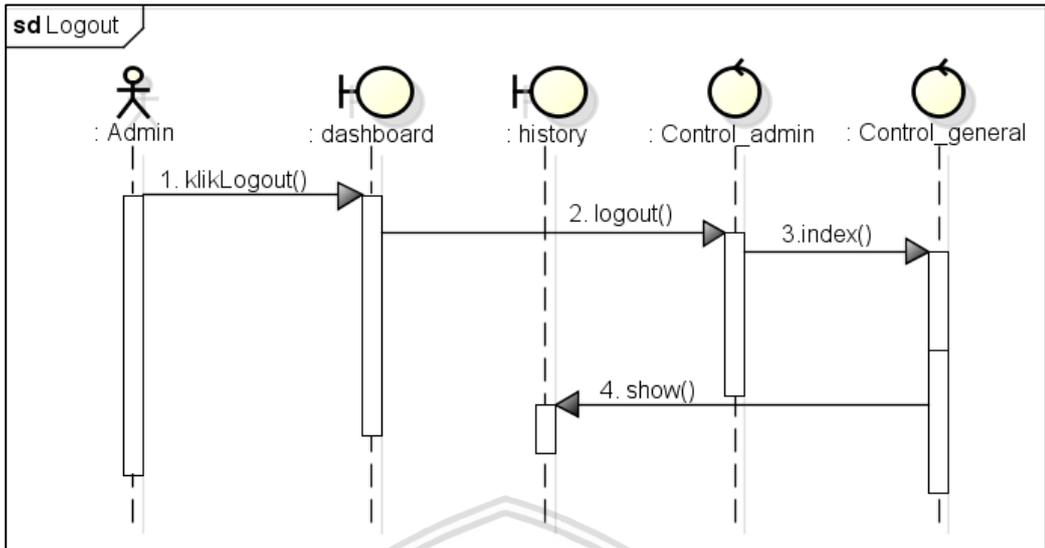


Gambar 4.25 Sequence diagram login

8. Sequence Diagram Logout

Pada Gambar 4.26 merupakan *sequence diagram* dari *use case logout*. Diagram ini menunjukkan aliran kode program untuk Admin ketika ingin keluar dari akun sistemnya.

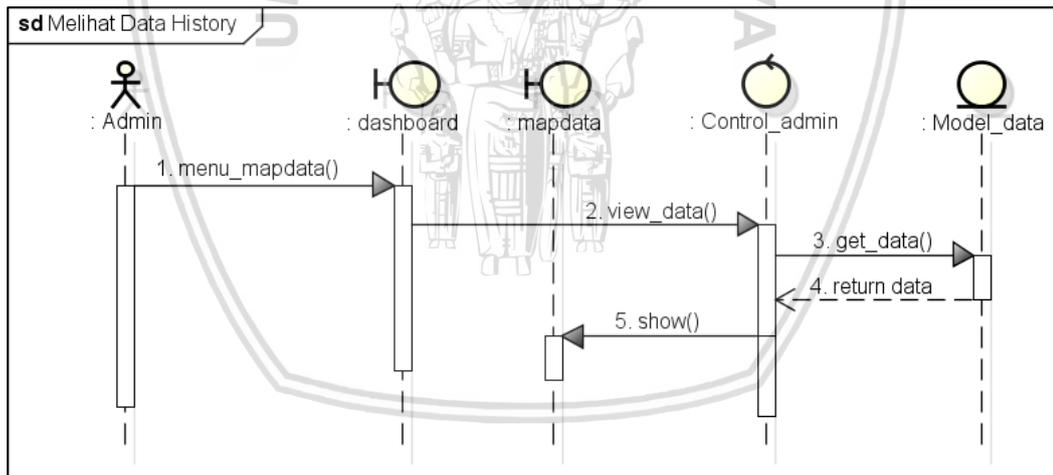




Gambar 4.26 Sequence diagram logout

9. Sequence Diagram Melihat Data History

Pada Gambar 4.27 merupakan *sequence diagram* dari *use case* melihat data *history*. Diagram ini menunjukkan aliran kode program untuk melihat data yang akan ditampilkan sebagai peta hasil pemetaan yang pernah ditambahkan ke sistem.

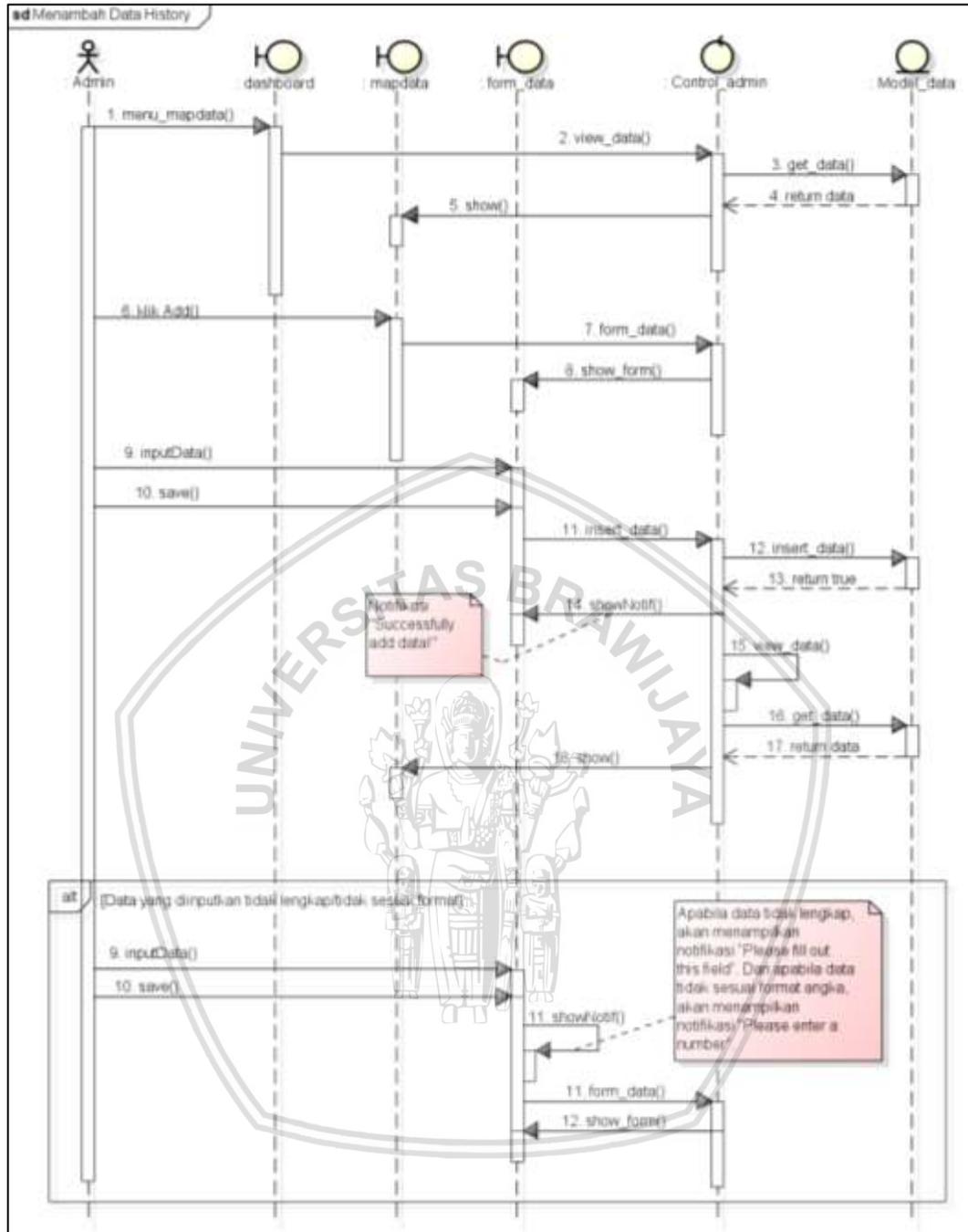


Gambar 4.27 Sequence diagram melihat data history

10. Sequence Diagram Menambah Data History

Pada Gambar 4.28 merupakan *sequence diagram* dari *use case* menambah data *history*. Diagram ini menunjukkan aliran kode program untuk Admin ketika ingin menambahkan data JSON baru ke sistem.

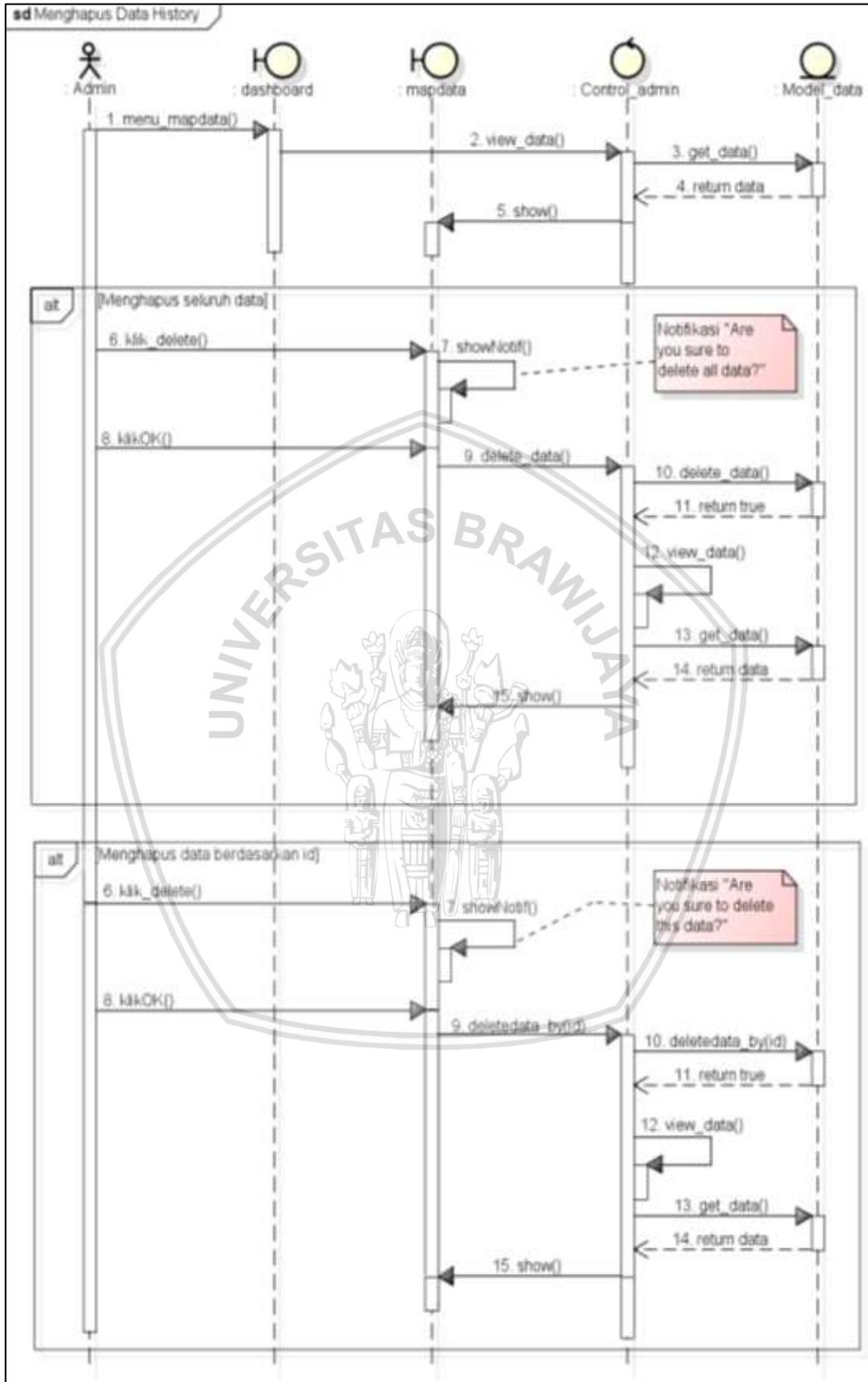




Gambar 4.28 Sequence diagram menambah data history

11. Sequence Diagram Menghapus Data History

Pada Gambar 4.29 merupakan *sequence diagram* dari *use case* menghapus data *history*. Diagram ini menunjukkan aliran kode program untuk menghapus data JSON yang pernah ditambahkan ke sistem.

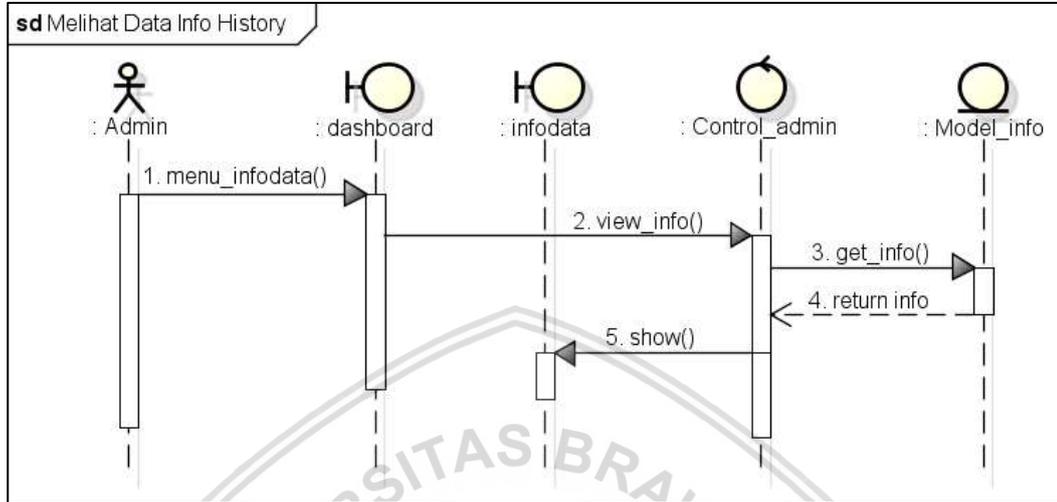


Gambar 4.29 Sequence diagram menghapus data history



12. Sequence Diagram Melihat Data Info History

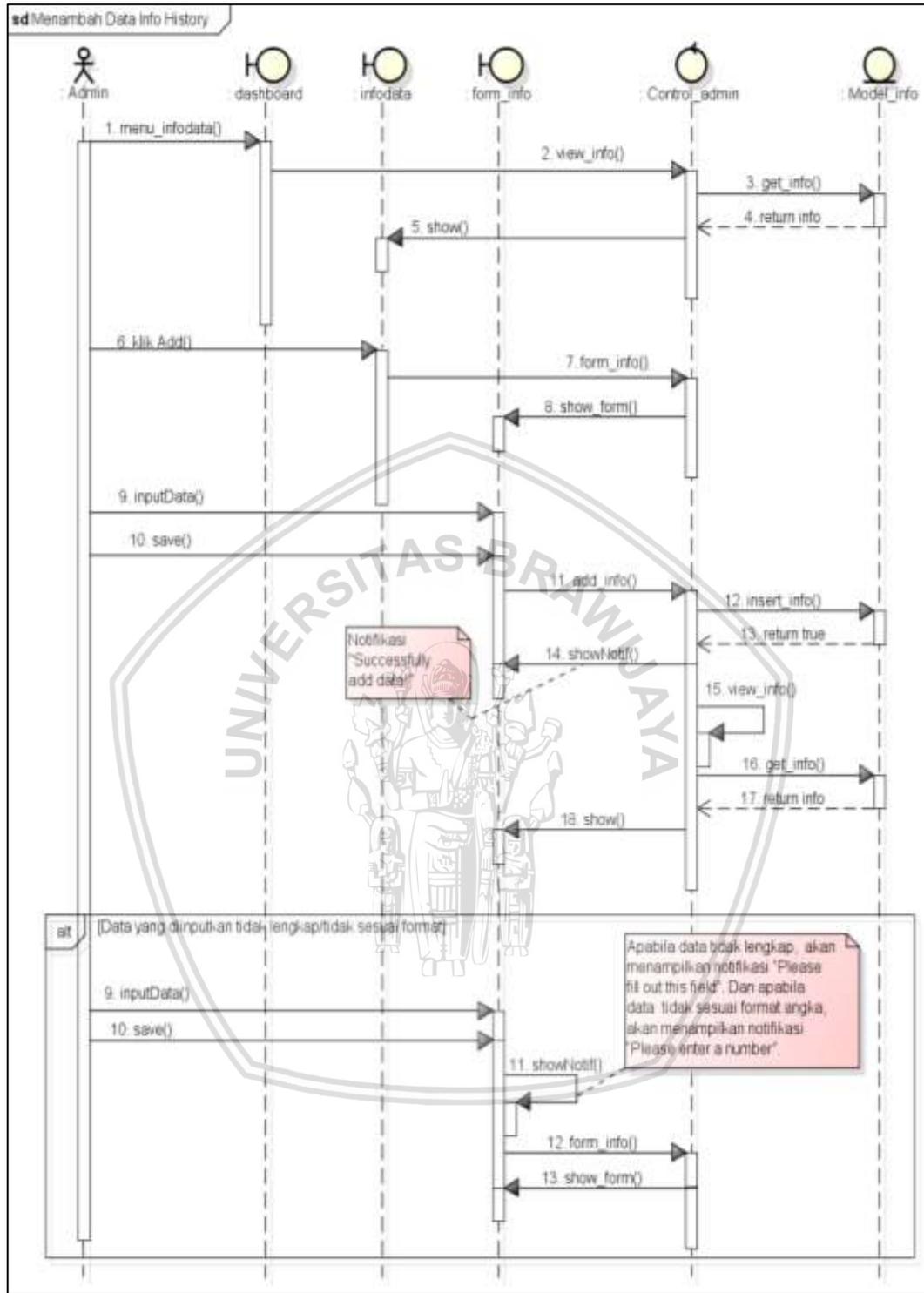
Pada Gambar 4.30 merupakan *sequence diagram* dari *use case* melihat data info *history*. Diagram ini menunjukkan aliran kode program untuk melihat data informasi peta yang pernah ditambahkan ke sistem.



Gambar 4.30 Sequence diagram melihat data info history

13. Sequence Diagram Menambah Data Info History

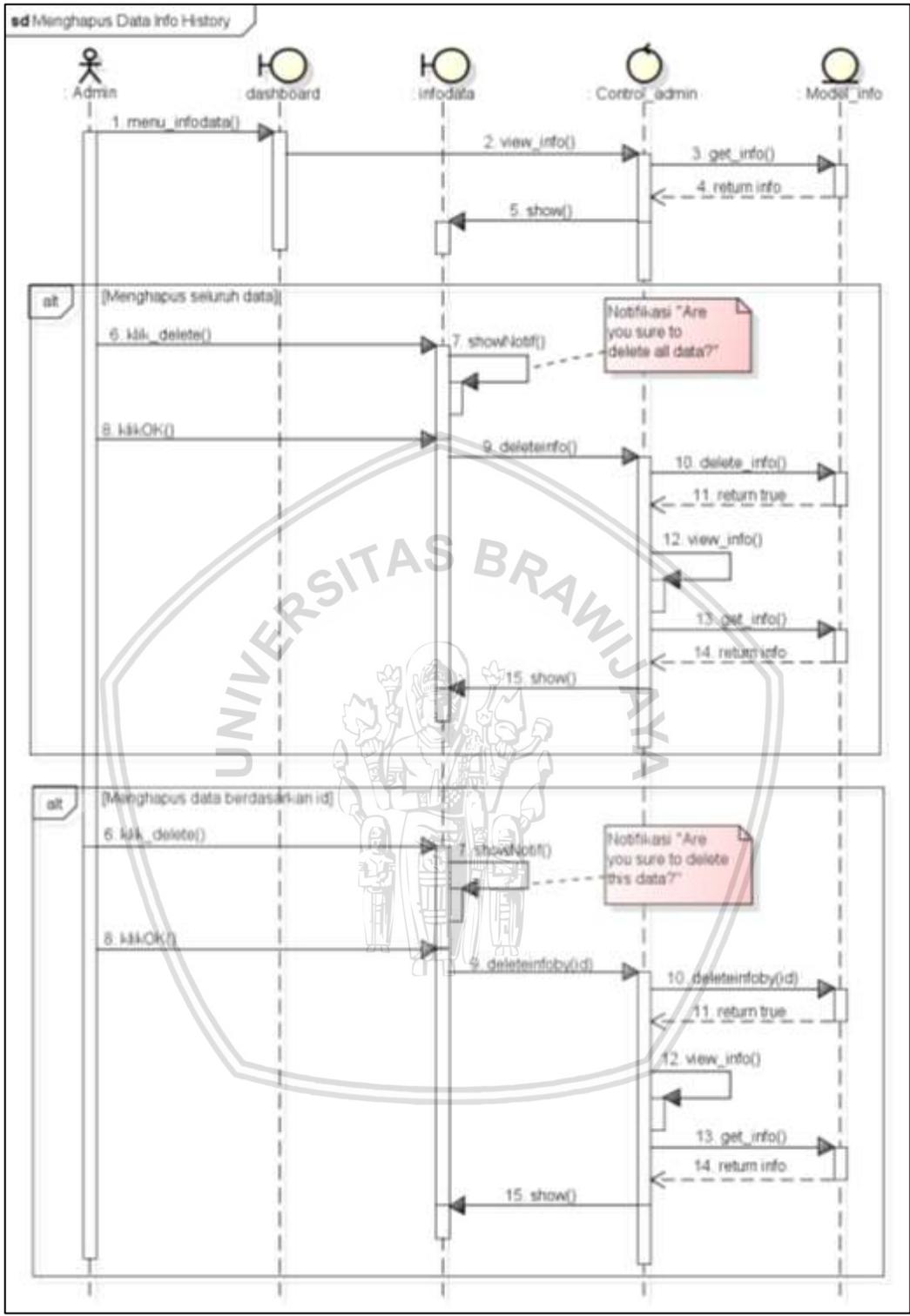
Pada Gambar 4.31 merupakan *sequence diagram* dari *use case* menambah data info *history*. Diagram ini menunjukkan aliran kode program untuk menambahkan data informasi peta baru ke sistem. Terdapat beberapa komponen dalam diagram *sequence* dari *use case* ini, antara lain Admin, *dashboard*, *view infodata*, *view form_info*, *Control_admin*, dan *Model_info*.



Gambar 4.31 Sequence diagram menambah data info history

14. Sequence Diagram Menghapus Data Info History

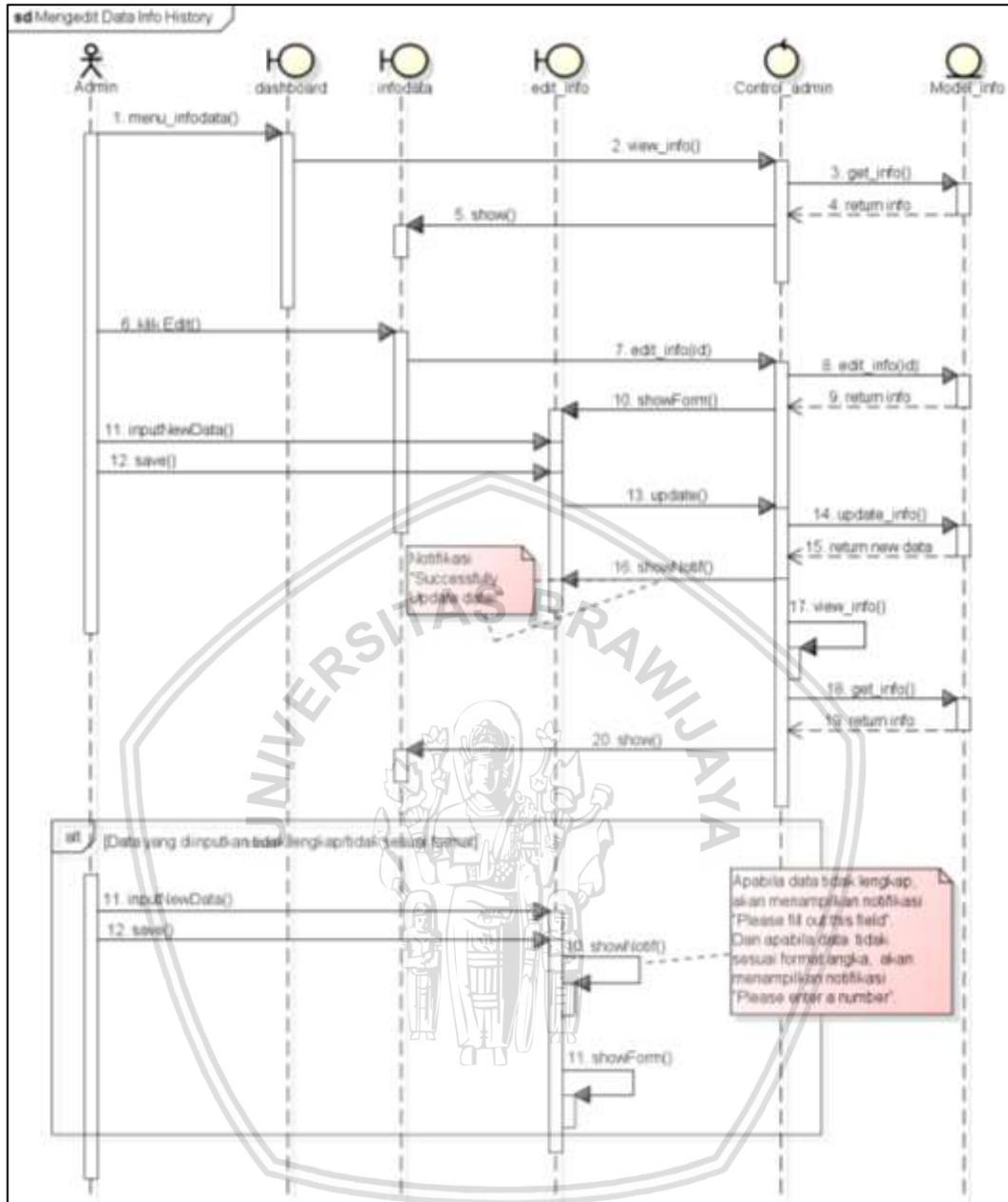
Pada Gambar 4.32 merupakan sequence diagram dari use case menghapus data info history. Diagram ini menunjukkan aliran kode program untuk menghapus informasi peta yang pernah ditambahkan ke sistem.



Gambar 4.32 Sequence diagram menghapus data info history

15. Sequence Diagram Mengedit Data Info History

Pada Gambar 4.33 merupakan sequence diagram dari use case mengedit data info history. Diagram ini menunjukkan aliran kode program untuk memperbarui data informasi peta yang pernah ditambahkan ke sistem.

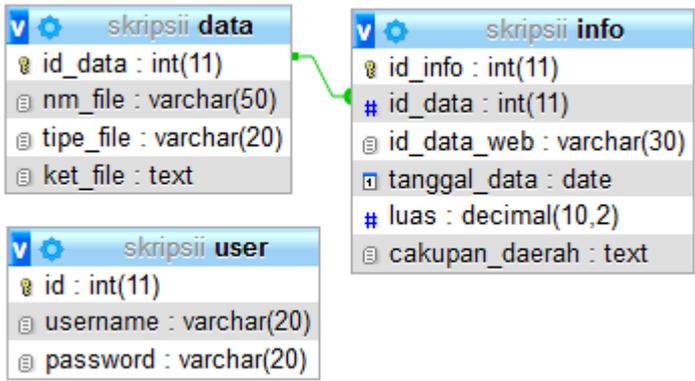


Gambar 4.33 Sequence diagram mengedit data info history

4.2.5 Perancangan Basis Data

Perancangan basis data digunakan untuk menyimpan seluruh data yang akan ditampilkan pada sistem dalam bentuk tabel. Perancangan basis data dapat dilihat pada Gambar 4.34 yang menunjukkan relasi antar entitas. Terdapat 3 tabel pada database, antara lain tabel data, tabel info, dan tabel user.





Gambar 4.34 Perancangan basis data

4.2.5.1 Kamus Data

Berikut ini merupakan perancangan setiap tabel yang terdapat pada basis data.

1. Tabel User

Nama tabel : user
 Jumlah field : 3
 Fungsi : untuk menyimpan seluruh data pengguna yaitu Admin

Tabel 4.20 Kamus data tabel user

| No. | Nama Field | Tipe | Lebar | Keterangan |
|-----|------------|-------------------|-------|-------------------|
| 1 | id | Integer | 11 | Id Admin |
| 2 | username | Character varying | 20 | Username pengguna |
| 3 | password | Character varying | 20 | Password pengguna |

2. Tabel Data

Nama tabel : data
 Jumlah field : 4
 Fungsi : untuk menyimpan seluruh data JSON hasil pemetaan daerah yang terkena erupsi

Tabel 4.21 Kamus data tabel data

| No. | Nama Field | Tipe | Lebar | Keterangan |
|-----|------------|-------------------|-------|---|
| 1 | id_data | Integer | 11 | Id dari data berupa tahun dari kejadian erupsi. |
| 2 | nm_file | Character varying | 50 | Nama dari file data json. |

Tabel 4.21 Kamus data tabel data (lanjutan)

| No. | Nama <i>Field</i> | Tipe | Lebar | Keterangan |
|-----|-------------------|--------------------------|-------|----------------------------|
| 3 | tipe_file | <i>Character varying</i> | 20 | Tipe dari file data JSON. |
| 4 | ket_file | <i>Text</i> | - | Keterangan file data JSON. |

3. Tabel Info

Nama tabel : info

Jumlah *field* : 7

Fungsi : untuk menyimpan seluruh data informasi hasil pemetaan daerah yang terkena erupsi

Tabel 4.22 Kamus data tabel info

| No. | Nama <i>Field</i> | Tipe | Lebar | Keterangan |
|-----|-------------------|--------------------------|-------|--|
| 1 | id_info | <i>Integer</i> | 11 | Id info |
| 2 | id_data | <i>Integer</i> | 11 | Id data yang merupakan foreign key dari <i>reference</i> tabel data. Dan sebagai keterangan tahun kejadian erupsi dari informasi yang ditambahkan. |
| 3 | id_data_web | <i>Character varying</i> | 30 | Id data asli dari web USGS. |
| 4 | tanggal_data | <i>Date</i> | | Tanggal data asli yang digunakan, yang diambil dari web USGS. |
| 5 | luas | <i>Decimal</i> | 10,2 | Luas daerah yang terkena erupsi berdasarkan tahun kejadian. |
| 6 | cakupan_daerah | <i>Text</i> | | Cakupan daerah di Sleman yang terkena erupsi. |

4.2.6 Perancangan Antarmuka

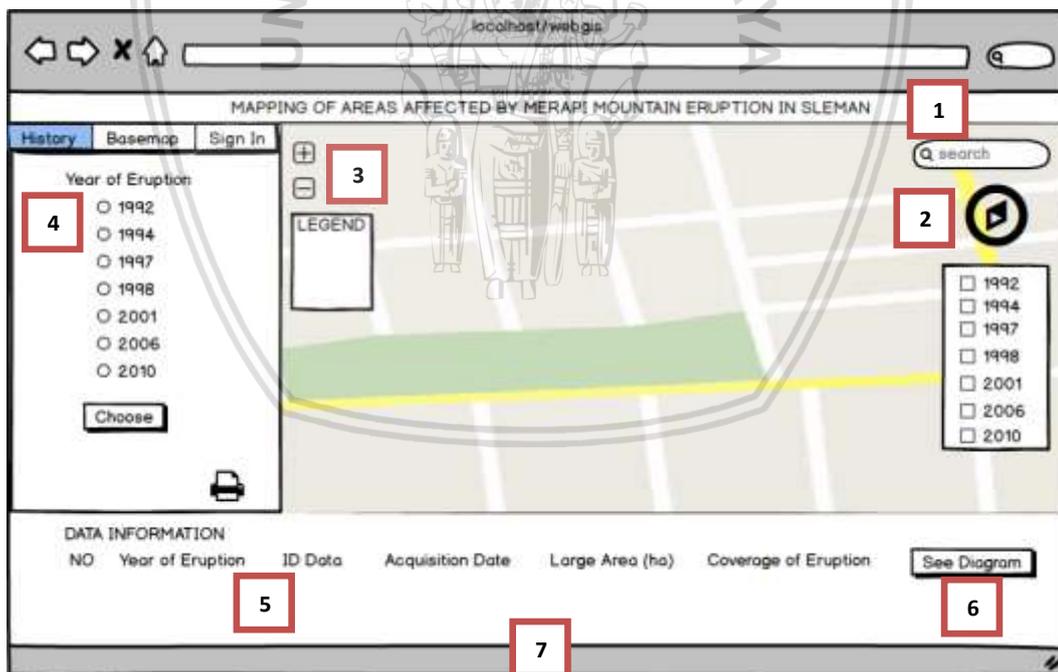
Perancangan antarmuka dibuat untuk memberikan gambaran tentang *user interface* yang menghubungkan antara aktor dengan sistem. Perancangan antarmuka dari sistem dibuat berdasarkan halaman-halaman yang ada pada sistem.

1. Perancangan Antarmuka Halaman *History*

Pada Gambar 4.35 merupakan perancangan antarmuka dari halaman *history* yang sekaligus menjadi halaman utama sistem. Halaman ini adalah halaman yang

akan tampil ketika pertama kali *user* mengakses sistem. Halaman tersebut menampilkan berbagai fitur yang dimiliki oleh sistem. Dibawah ini merupakan keterangan dari bagian-bagian pada halaman *history* :

1. *Header* sistem yang berisi nama dari sistem.
2. Kolom *Search*, arah mata angin, dan kolom yang menampilkan informasi tahun kejadian erupsi.
3. Visualisasi peta hasil pemetaan serta tombol *Zoom in*, *Zoom out* dan *Legend* dari *basemap* yang ditampilkan.
4. Menu navigasi dari sistem antara lain *History*, *Basemap*, dan *Sign In*. Pada halaman *History* juga akan ditampilkan pilihan tahun kejadian erupsi yang dapat dipilih oleh *user*. Selain itu, terdapat tombol *print* yang digunakan untuk mencetak peta hasil pemetaan yang divisualisasikan.
5. Menampilkan informasi detail dari peta yang divisualisasikan, yaitu *Year of Eruption*, *ID Data*, *Acquisition Date*, *Large Area* dalam satuan ha, dan *Coverage of Eruption*.
6. Tombol *See Diagram* digunakan untuk melihat diagram yang menunjukkan luasan daerah hasil pemetaan.
7. *Footer* dari sistem.

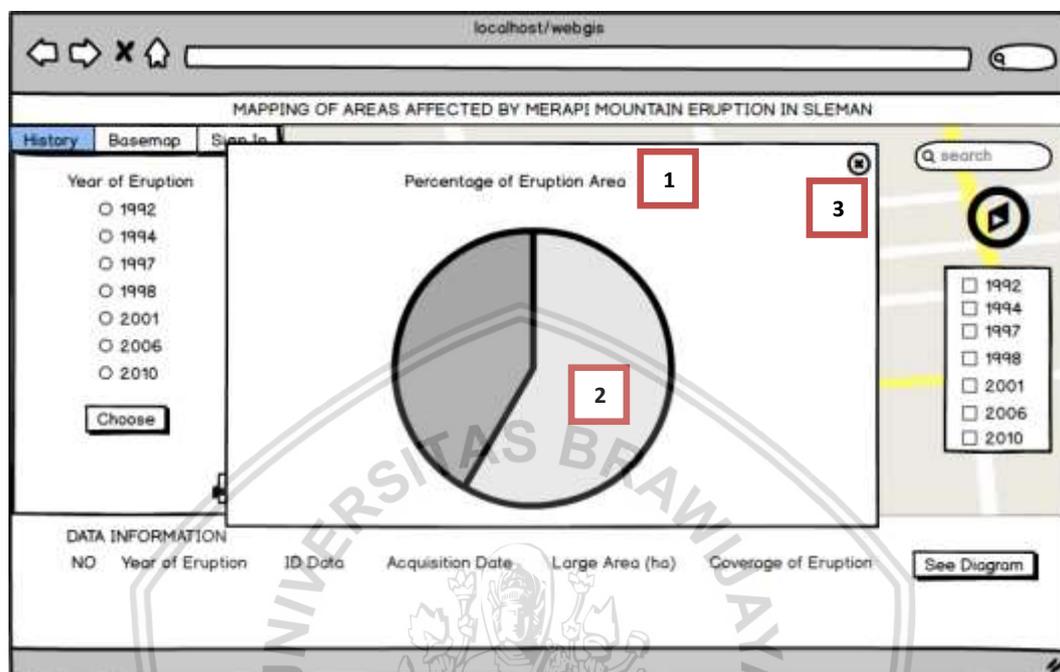


Gambar 4.35 Perancangan antarmuka halaman *history*

2. Perancangan Antarmuka Halaman Diagram

Pada Gambar 4.36 merupakan perancangan antarmuka dari halaman diagram yang menampilkan diagram luasan daerah hasil pemetaan. Dibawah ini merupakan keterangan dari bagian-bagian pada halaman diagram :

1. Judul dari *popup* diagram yang ditampilkan.
2. *Pie Diagram* yang menunjukkan luasan daerah yang terdampak erupsi pada setiap tahun kejadian yang ditambahkan ke sistem.
3. Tombol *Close* untuk menutup *popup*.



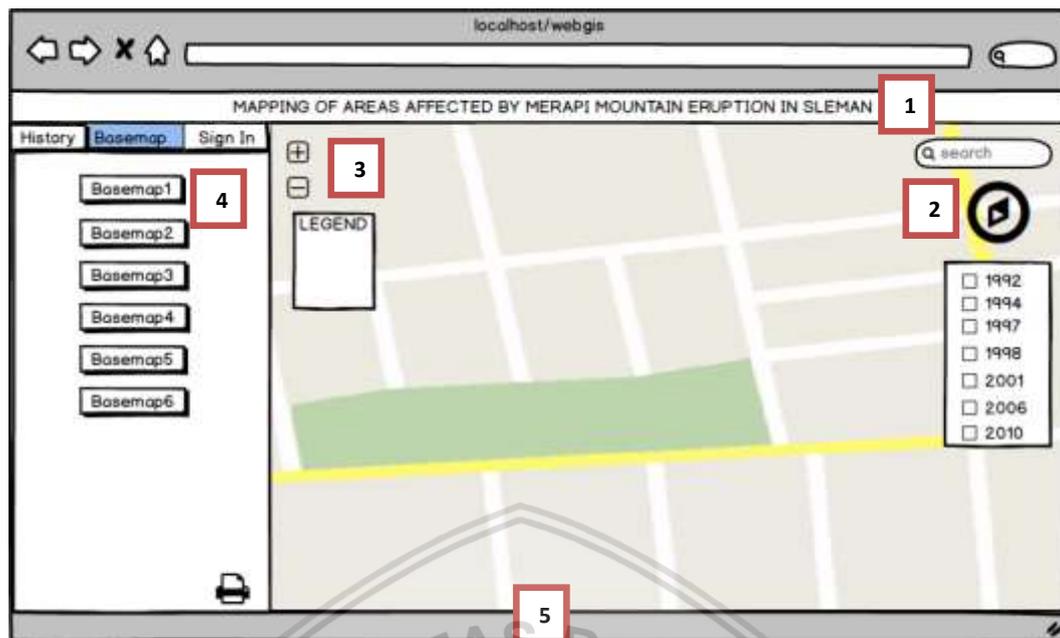
Gambar 4.36 Perancangan antarmuka halaman diagram

3. Perancangan Antarmuka Halaman *Basemap*

Pada Gambar 4.37 merupakan antarmuka dari menu *Basemap*, yang digunakan untuk menampilkan pilihan *basemap* yang terdapat pada sistem. *User* dapat memilih *basemap* untuk peta yang divisualisasikan.

Dibawah ini merupakan keterangan dari bagian-bagian yang terdapat pada halaman *Basemap* :

1. *Header* sistem dimana terdapat judul dari sistem.
2. Kolom *Search*, arah mata angin, dan kolom yang menampilkan informasi tahun kejadian erupsi.
3. Visualisasi peta hasil pemetaan serta tombol *Zoom in*, *Zoom out* dan *Legend* dari *basemap* yang ditampilkan.
4. Pilihan-pilihan *basemap* yang dapat dipilih dan tombol *print* yang digunakan untuk mencetak peta dengan *basemap* yang telah dipilih.
5. *Footer* sistem.

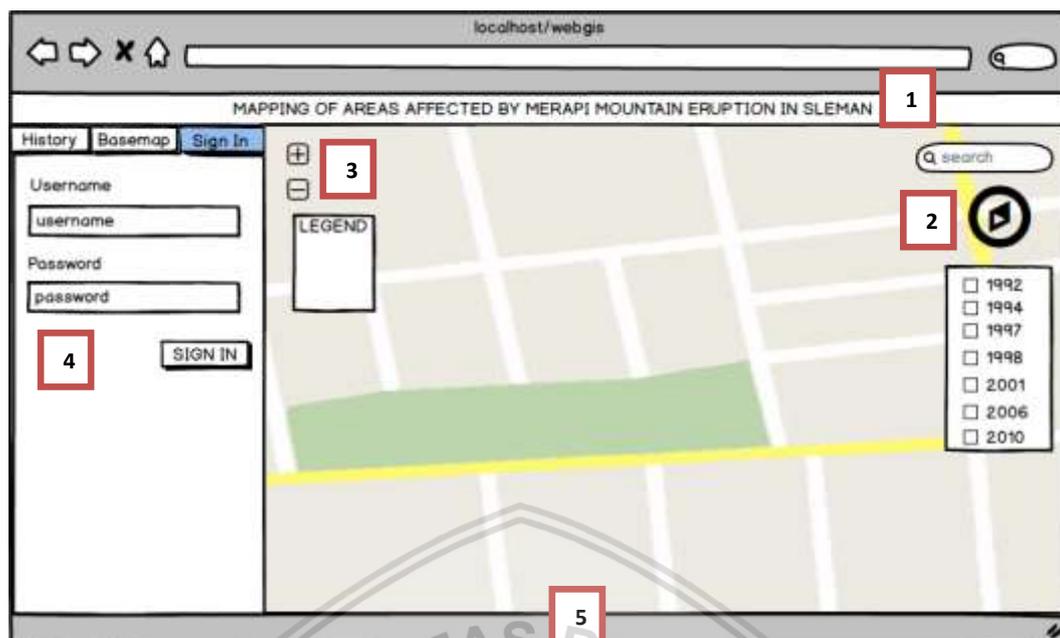


Gambar 4.37 Perancangan antarmuka halaman *basemap*

4. Perancangan Antarmuka Halaman *Sign In*

Gambar 4.38 merupakan tampilan menu *Sign In* dari sistem. Halaman *Sign In* digunakan oleh Admin ketika akan masuk ke dalam sistem. Di bawah ini merupakan keterangan dari bagian-bagian yang terdapat pada halaman *Sign In* :

1. Header sistem dimana terdapat judul dari sistem.
2. Kolom *Search*, arah mata angin, dan kolom yang menampilkan informasi tahun kejadian erupsi.
3. Visualisasi hasil pemetaan dalam bentuk peta, serta tombol *Zoom in*, *Zoom out* dan *Legend* dari *basemap* yang ditampilkan.
4. Kolom *username* yang digunakan untuk menginputkan *Username* Admin, kolom *password* yang digunakan untuk menginputkan *password*, dan tombol *Sign In* yang digunakan untuk masuk ke dalam sistem.
5. *Footer* sistem.



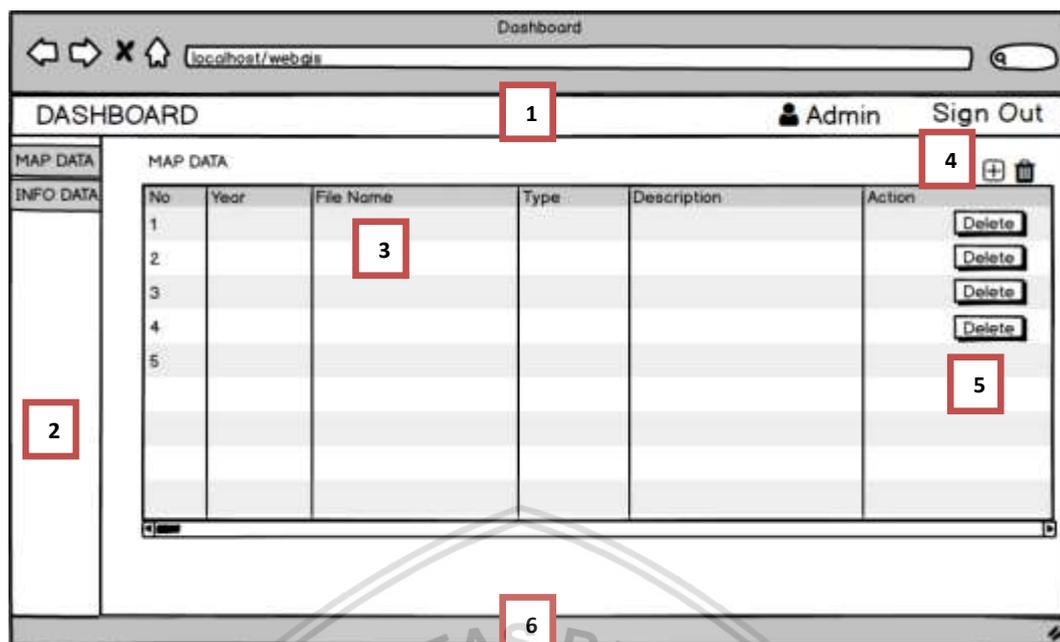
Gambar 4.38 Perancangan antarmuka halaman *sign in*

5. Perancangan Antarmuka Halaman *Map Data*

Gambar 4.39 merupakan antarmuka menu *Map Data* yang akan tampil ketika Admin telah berhasil masuk ke sistem. Seluruh data JSON yang pernah ditambahkan ke sistem akan ditampilkan pada halaman *Map Data*. Halaman ini juga digunakan untuk mengelola data JSON pada sistem.

Dibawah ini merupakan keterangan dari bagian-bagian yang terdapat pada halaman *Map Data* :

1. *Header dashboard* dimana terdapat menu *Sign Out*, keterangan dari admin, dan keterangan yang menjelaskan bahwa halaman ini bagian dari *dashboard*.
2. Menu-menu yang ada pada dashboard Admin yaitu menu *Map Data* dan *Info Data*.
3. Tabel untuk menampilkan data JSON yang pernah ditambahkan ke sistem sebelumnya.
4. Tombol *Add* digunakan untuk menambahkan data JSON baru, sedangkan tombol *Delete* digunakan untuk menghapus seluruh data JSON yang terdapat pada halaman *Map Data*.
5. Tombol *Delete* pada setiap baris data digunakan untuk menghapus data per baris.
6. *Footer dashboard* Admin.



Gambar 4.39 Perancangan antarmuka halaman *Map Data*

6. Perancangan Antarmuka Halaman Form Tambah Data *History*

Gambar 4.40 merupakan antarmuka dari *form* untuk menambahkan data JSON baru ke database dan sistem. Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *form data history* :

1. Header dashboard Admin dimana terdapat menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini bagian dari dashboard.
2. Menu-menu yang ada pada dashboard yaitu menu *Map Data* serta *Info Data*.
3. Tombol *Browse* digunakan untuk mencari dan memilih peta baru yang ingin diinputkan.
4. Kolom *Year of Eruption* digunakan untuk menambahkan tahun dari data yang akan ditambahkan.
5. Kolom *Description* yang digunakan untuk menambahkan keterangan tentang data JSON yang ingin ditambahkan.
6. Tombol *Save* yang digunakan untuk menyimpan data yang diinputkan ke *database* dan tombol *Reset* yang digunakan untuk mereset kolom data.
7. Bagian *footer* dashboard.

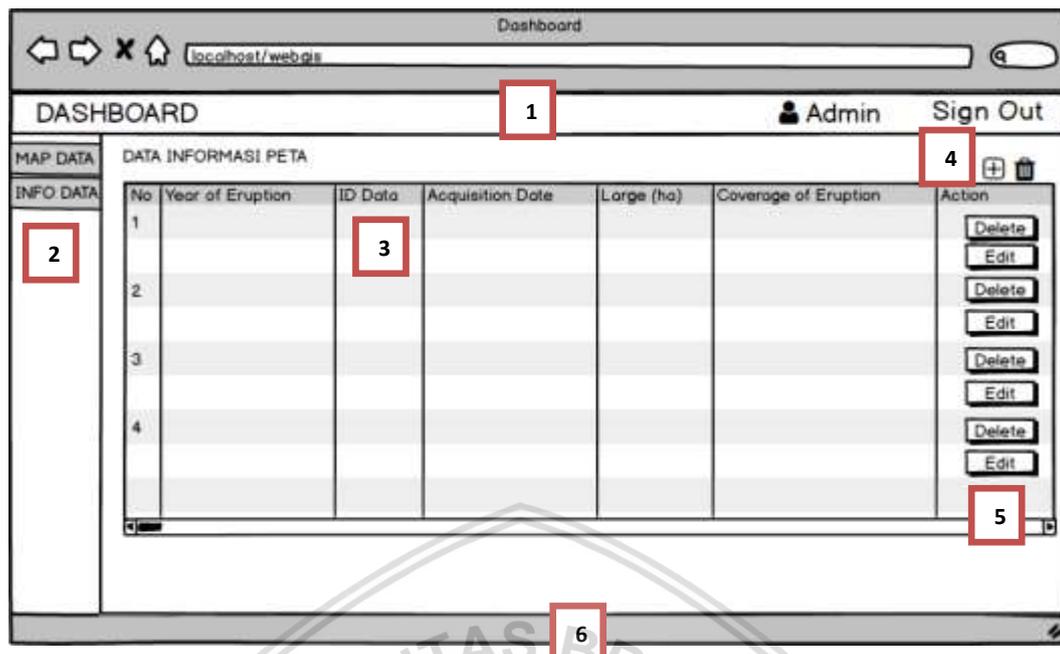
Gambar 4.40 Perancangan antarmuka halaman form tambah data *history*

7. Perancangan Antarmuka Halaman *Info Data*

Gambar 4.41 merupakan antarmuka menu *Info Data* yang digunakan untuk menampilkan semua informasi peta yang pernah ditambahkan serta digunakan untuk mengelola data informasi peta pada sistem.

Dibawah ini merupakan keterangan dari bagian-bagian yang terdapat pada halaman *Info Data* :

1. *Header* dari *dashboard* Admin dimana terdapat menu *Sign Out*, keterangan *admin*, dan keterangan yang menjelaskan bahwa halaman ini adalah bagian dari *dashboard*.
2. Menu-menu yang ada pada dashboard yaitu menu *Map Data* dan *Info Data*.
3. Tabel yang digunakan untuk menampilkan informasi peta yang pernah ditambahkan.
4. Tombol *Add* digunakan untuk menambahkan informasi peta baru, sedangkan tombol *Delete* digunakan untuk menghapus seluruh data yang ada pada halaman *Info Data*.
5. Tombol *Delete* pada tiap baris digunakan untuk menghapus data per baris. Sedangkan tombol *Edit* digunakan untuk memperbarui data per baris.
6. *Footer* dari halaman *Info Data*.



Gambar 4.41 Perancangan antarmuka halaman *Info Data*

8. Perancangan Antarmuka Halaman *Form Tambah Info*

Pada Gambar 4.42 merupakan antarmuka dari *form* untuk menambahkan data informasi peta baru ke *database* dan sistem. Di bawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *form* info :

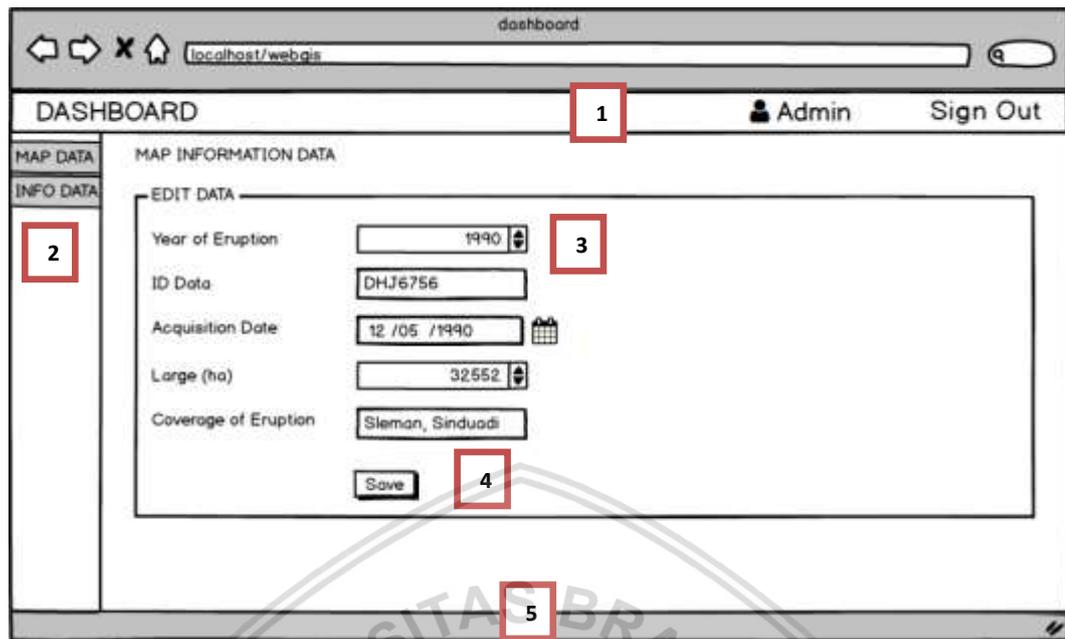
1. Header dari dashboard Admin, dimana terdapat menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini adalah bagian dari dashboard.
2. Menu-menu yang ada di dashboard Admin yaitu menu *Map Data* dan *Info Data*.
3. Kolom-kolom input yang harus dimasukkan ketika menambahkan data informasi baru, yaitu kolom *year of eruption*, *id data*, *acquisition date*, *large* dalam satuan ha, dan *coverage of eruption*.
4. Tombol *Save* yang digunakan untuk menyimpan data ke *database*, sedangkan tombol *Reset* digunakan untuk mereset data.
5. Bagian *footer* dari halaman tambah info.

Gambar 4.42 Perancangan antarmuka halaman *form* tambah info

9. Perancangan Antarmuka Halaman Form *Edit* Info

Gambar 4.43 merupakan antarmuka dari *form* untuk memperbaiki data informasi peta yang pernah ditambahkan. Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *form edit* info :

1. Header dari dashboard Admin, dimana terdapat menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini adalah bagian dari dashboard.
2. Menu-menu yang ada di dashboard yaitu menu *Map Data* dan *Info Data*.
3. Kolom-kolom input yang harus dimasukkan ketika memperbaiki data informasi, yaitu kolom *year of eruption*, *id data*, *acquisition date*, *large* dalam satuan ha, dan *coverage of eruption*.
4. Tombol *Save* digunakan untuk menyimpan perubahan data ke basis data.
5. Bagian *footer* dari halaman *edit* info.



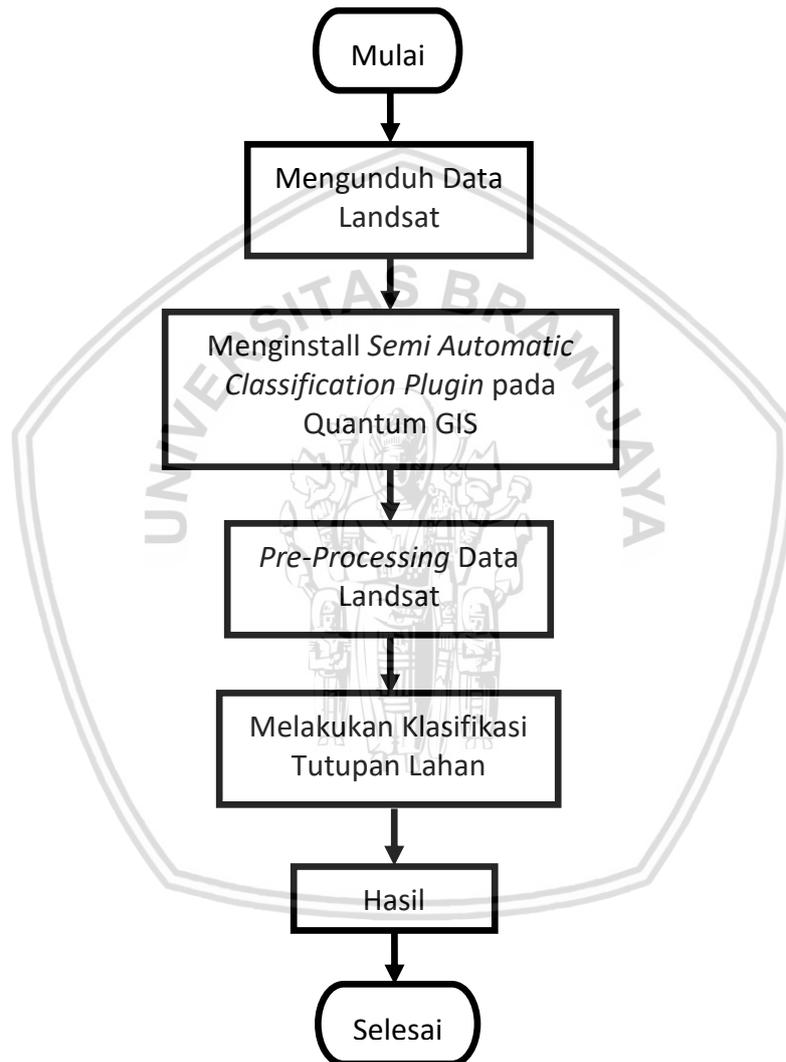
Gambar 4.43 Perancangan antarmuka halaman *form edit info*



BAB 5 IMPLEMENTASI

5.1 *Pre-processing* Data dan Penerapan *Semi-Automatic Classification Plug-in (SCP)*

Gambar 5.1 merupakan proses yang akan dilakukan penulis dalam melakukan pengolahan data yang digunakan untuk menghasilkan pemetaan daerah yang terkena erupsi.



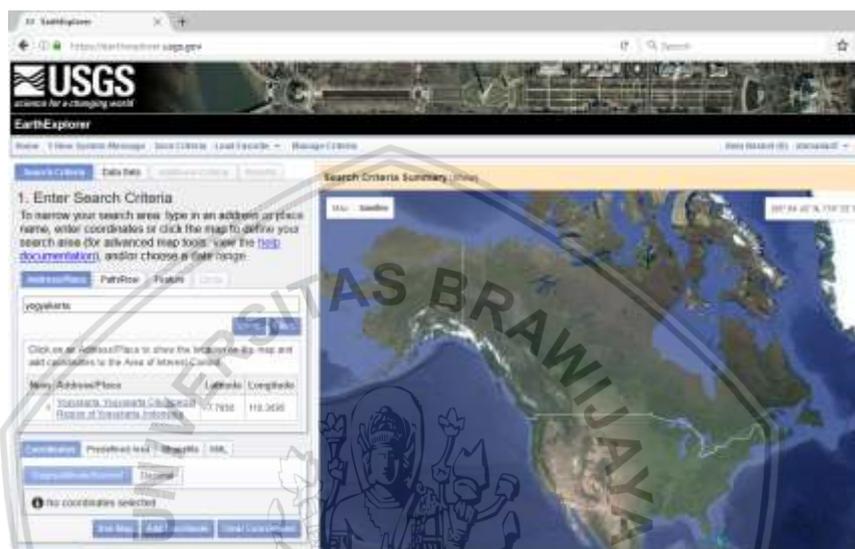
Gambar 5.1 Diagram alur *pre-processing* data dan penerapan SCP

5.1.1 Mengunduh Data Landsat

Langkah pertama yang harus dilakukan adalah mengunduh data Landsat yang dibutuhkan pada website earthexplorer.usgs.gov. Data Landsat yang akan diunduh pada penelitian ini adalah data Landsat untuk Daerah Istimewa Yogyakarta, yaitu di Kabupaten Sleman yang terletak pada *Path* : 120 dan *Row* : 65.

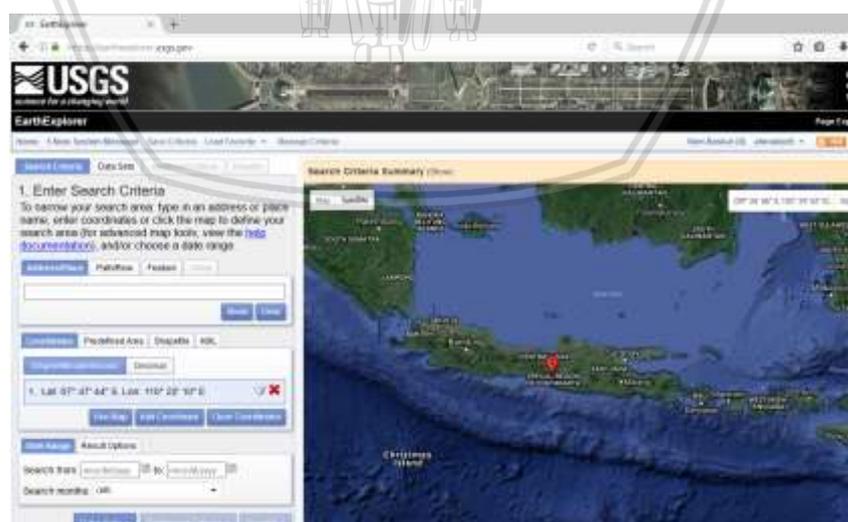
Untuk mendapatkan data yang dibutuhkan, berikut adalah langkah-langkah yang harus dilakukan :

1. Buka halaman website www.earthexplorer.usgs.gov, maka akan muncul tampilan seperti pada Gambar 5.2.
2. Masukkan kata kunci Yogyakarta pada kolom *Enter Search Criteria*, kemudian klik *Show*. Maka pada bawah kolom *Search* akan muncul nama tempat yang dicari beserta *latitude* dan *longitude* dari Yogyakarta seperti pada Gambar 5.2.



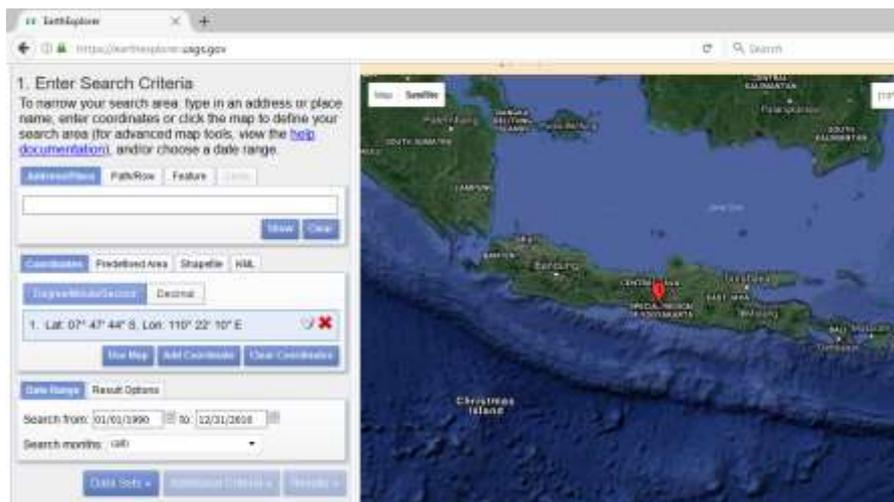
Gambar 5.2 Langkah 1 mengundah data

3. Klik pada bagian Yogyakarta tersebut, maka akan muncul *latitude* dan *longitude* secara lengkap dari Yogyakarta seperti pada Gambar 5.3.



Gambar 5.3 Langkah 2 mengundah data

4. Pada kolom *Date Range* masukkan waktu pencarian dari data yang dibutuhkan yaitu mulai tahun 1990 sampai tahun 2016 seperti pada Gambar 5.4. Kemudian klik *Data Sets*.



Gambar 5.4 Langkah 3 mengunduh data

- Pilih *Landsat Archive* – pilih *Pre-Collection Level-1* – centang pada bagian L4-5 TM untuk bisa mendapatkan data dari tahun 1990 juga. Kemudian pilih *Results* seperti pada Gambar 5.5.



Gambar 5.5 Langkah 4 mengunduh data

- Maka akan muncul pilihan data yang dibutuhkan, dan kita hanya perlu memilih data mana yang ingin diunduh sesuai dengan tahun kejadian erupsi Gunung Merapi.

5.1.2 Menginstall *Semi Automatic Classification Plugin (SCP)*

Langkah kedua yang harus dilakukan setelah seluruh data Landsat yang dibutuhkan telah diunduh adalah menginstall *Semi Automatic Classification Plugin (SCP)* pada aplikasi Quantum GIS. Langkah-langkah yang harus dilakukan untuk instalasi SCP adalah sebagai berikut :

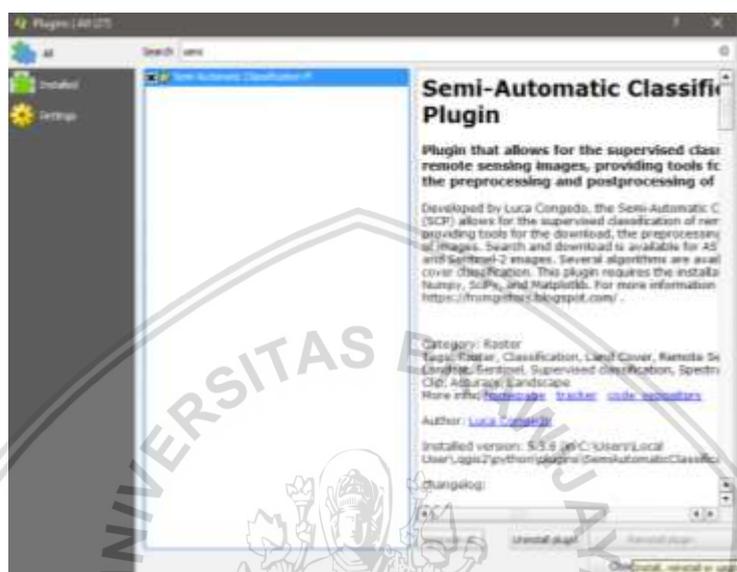
- Buka Quantum GIS, kemudian pilih menu *Plugins* – pilih *Manage and Install Plugins* seperti pada Gambar 5.6.





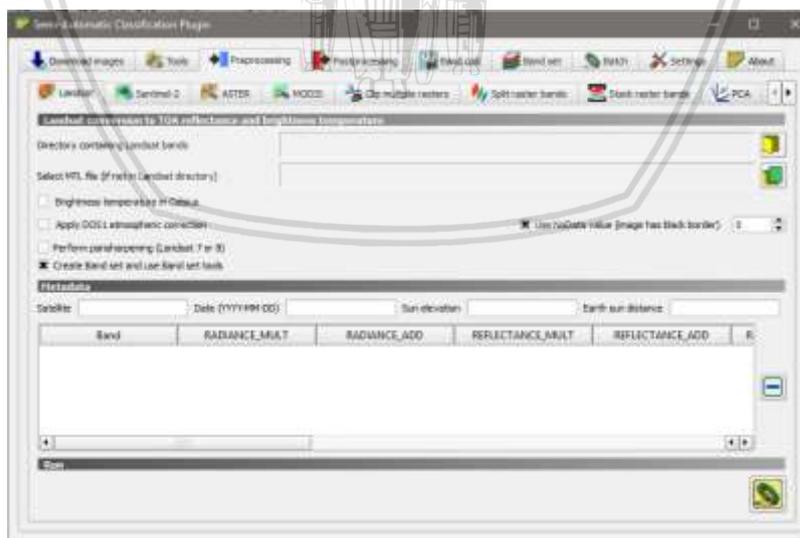
Gambar 5.6 Langkah 1 menginstall SCP

2. Pada kolom *Search* cari kata kunci *Semi Automatic Classification*, maka akan muncul *plugins* tersebut seperti yang ditunjukkan pada Gambar 5.7. Kemudian klik *Install Plugin* dan tunggu hingga proses instalasi selesai.



Gambar 5.7 Langkah 2 menginstall SCP

3. Apabila telah berhasil diinstall, maka dapat dilihat detail tampilan dari *Semi Automatic Classification Plugin* seperti yang ditunjukkan pada Gambar 5.8.



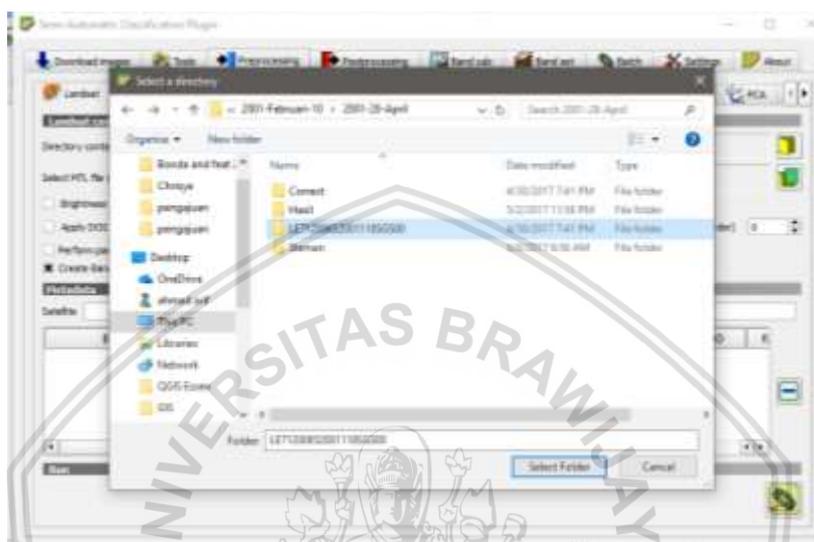
Gambar 5.8 Langkah 3 menginstall SCP

5.1.3 Pre-Processing Data Landsat

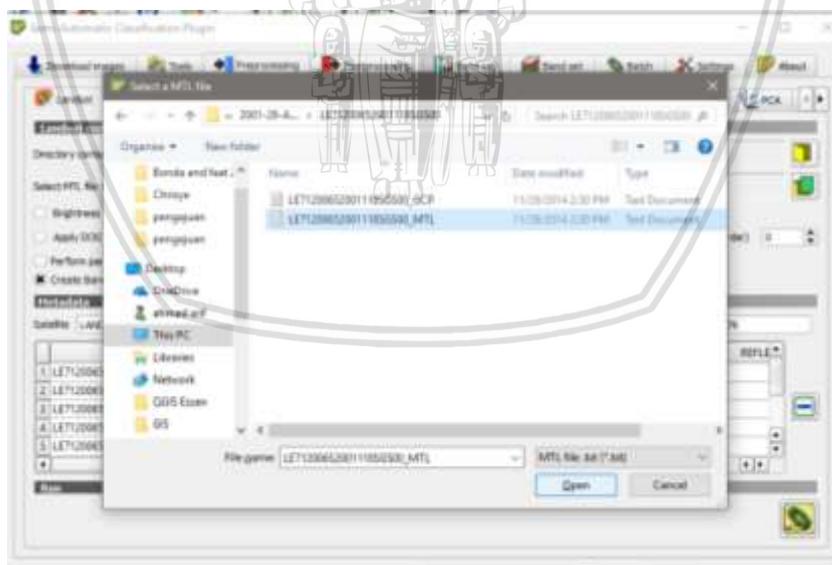
Setelah *plugin* telah berhasil diinstal, langkah selanjutnya adalah melakukan *pre-processing* data sebagai langkah awal dalam pengolahan data. Pre-

processing dilakukan untuk menghilangkan *noise*, meningkatkan akurasi, memperjelas fitur data, memperbesar atau memperkecil ukuran data, serta mengkonversi data asli untuk mendapatkan data yang sesuai dengan kebutuhan. Langkah-langkah untuk *pre-processing* yang harus dilakukan adalah antara lain :

1. Buka *Semi Automatic Classification Plugin* seperti pada langkah sebelumnya.
2. Inputkan data Landsat dan MTL dari data Landsat yang akan dilakukan *pre-processing* seperti pada Gambar 5.9 dan 5.10.



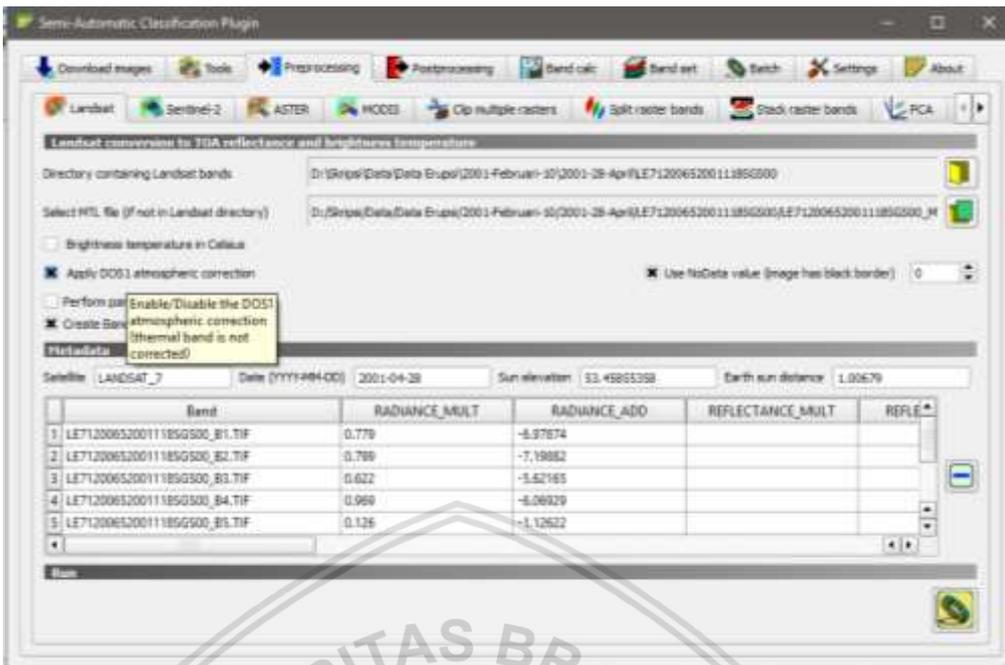
Gambar 5.9 Input data Landsat



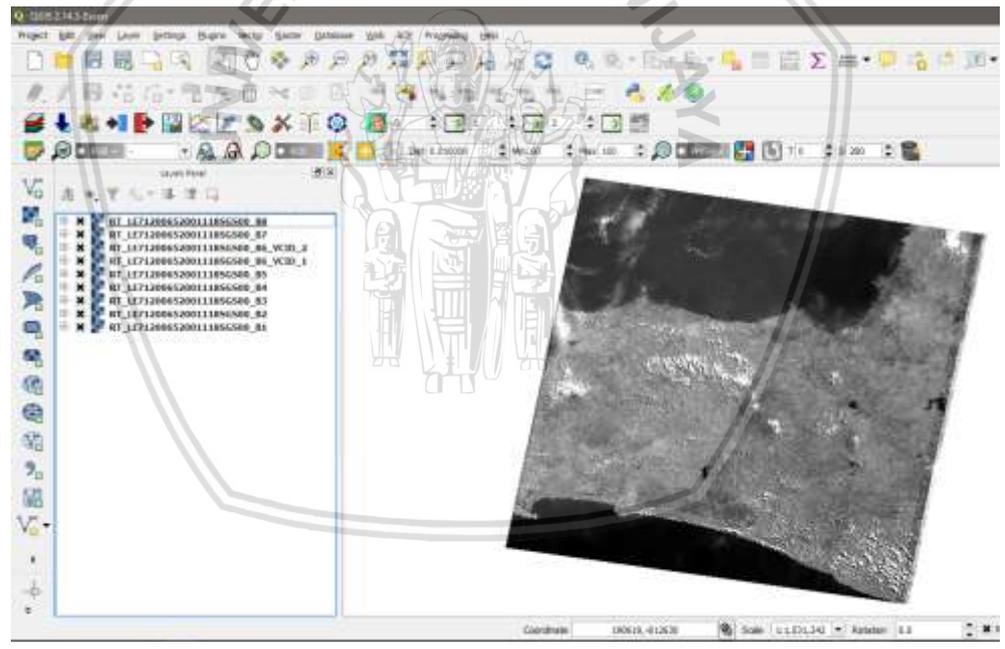
Gambar 5.10 Input MTL

3. Apabila data Landsat dan MTL telah diinputkan, maka tampilannya akan seperti pada Gambar 5.11. Kemudian centang pada pilihan *Apply DOS1 atmospheric correction* dan klik *Run* untuk melakukan *pre-processing*. Tunggu hingga proses *pre-processing* selesai. Dan hasilnya akan seperti yang ditunjukkan pada Gambar 5.12.



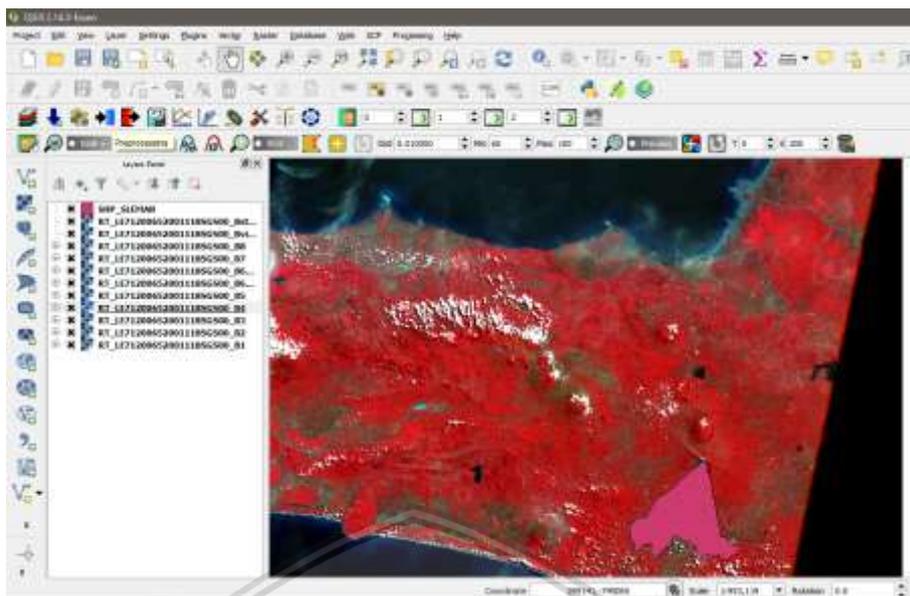


Gambar 5.11 Hasil input data Landsat dan MTL



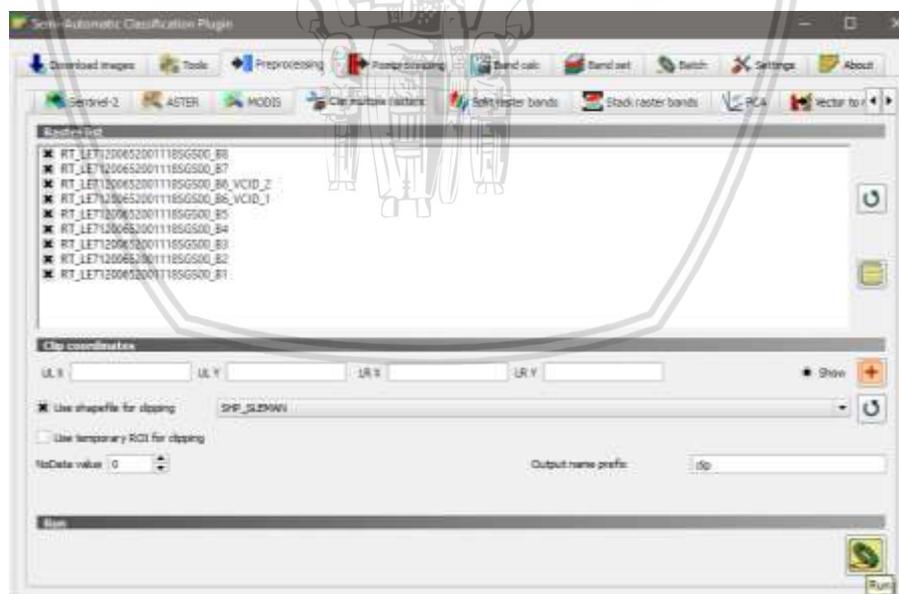
Gambar 5.12 Hasil pre-processing

- Untuk mendapatkan data pada daerah Kabupaten Sleman saja, perlu untuk melakukan pemotongan data dengan menggunakan *tool Clipper*. Sebelum dilakukan pemotongan, masukkan terlebih dahulu data SHP yang berbentuk Kabupaten Sleman seperti pada Gambar 5.13. Kemudian buka kembali *Semi Automatic Classification Plugin*.



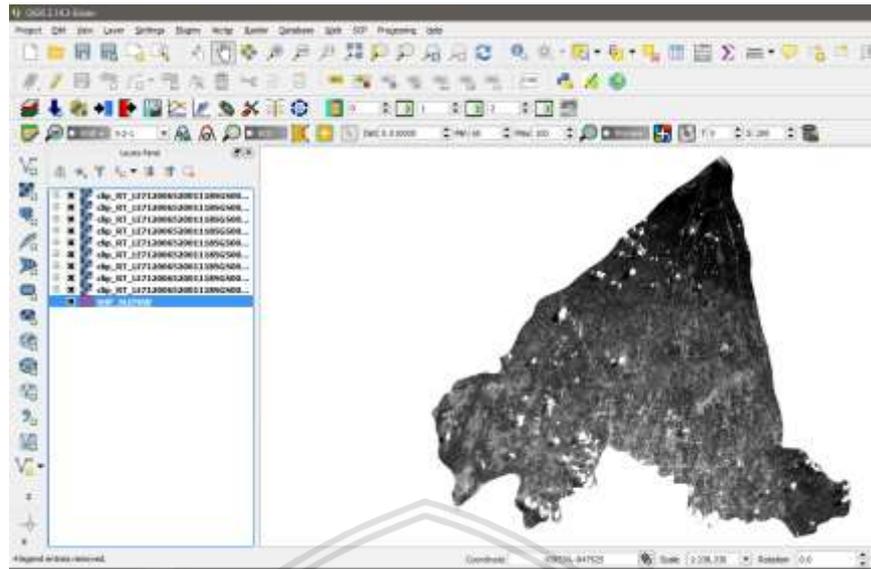
Gambar 5.13 Data SHP Kabupaten Sleman

5. Pilih sub menu *Clip Multiple Raster* seperti pada Gambar 5.14 dan klik *select all* sehingga seluruh band hasil pre-processing dapat dipotong bersamaan. Centang pada *pilihan Use shapefile for clipping* dan pilih file SHP yang akan digunakan untuk memotong berdasarkan bentuk Kabupaten Sleman. Kemudian klik *Run* untuk menjalankan perintah pemotongan dan pilih direktori untuk menyimpan data hasil pemotongan.



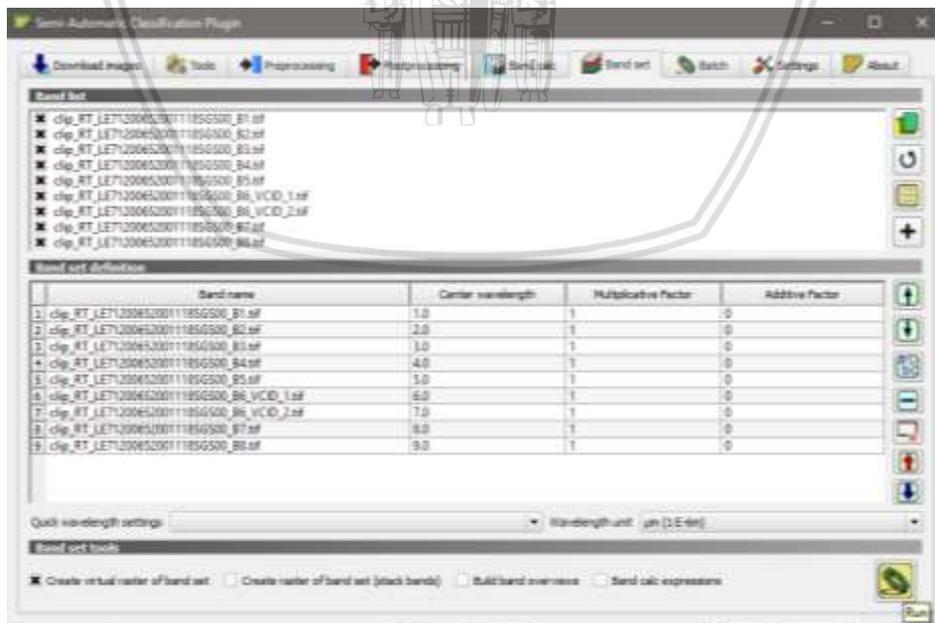
Gambar 5.14 Proses pemotongan menggunakan *clip multiple raster*

6. Apabila proses pemotongan telah selesai, maka akan tampil hasilnya seperti pada Gambar 5.15.



Gambar 5.15 Hasil pemotongan daerah Kabupaten Sleman

- Setelah data berhasil dipotong sesuai bentuk Kabupaten Sleman, selanjutnya adalah membuat *Virtual Raster of Band Set* (VRT) yang akan digunakan untuk melakukan klasifikasi data pada tahapan berikutnya. Untuk membuat VRT, buka kembali *Semi Automatic Classification Plugin* – pilih menu *Band set* – klik *Select All* untuk memilih seluruh data band – klik tanda *Add band to band set* untuk menambahkan data yang telah dipilih – centang pada *Create Virtual raster of band set* – kemudian klik run dan pilih direktori untuk menyimpan file VRT. Langkah-langkah tersebut dapat dilihat pada Gambar 5.16.

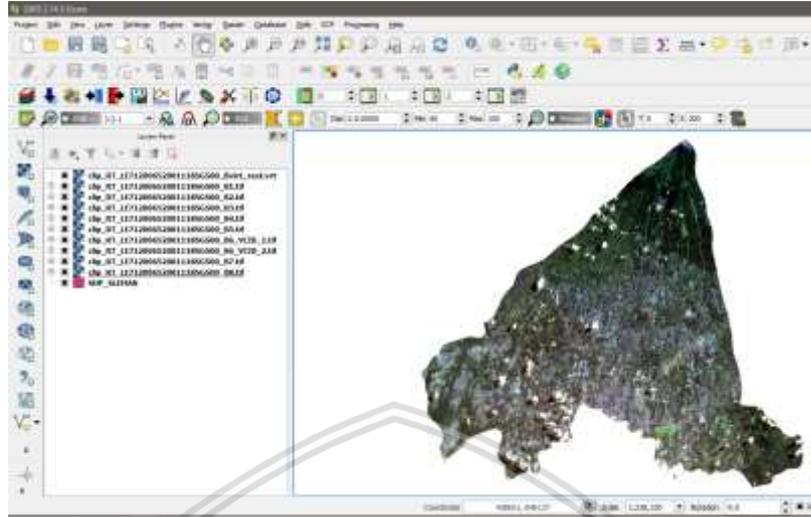


Gambar 5.16 Tampilan pembuatan VRT

- Apabila proses telah selesai, maka akan tampil hasil VRT seperti pada Gambar 5.17. VRT tersebut yang akan digunakan untuk melakukan



klasifikasi. Untuk mempermudah dalam membedakan setiap objek yang ada pada peta, kita dapat mengubah RGB dari VRT tersebut.

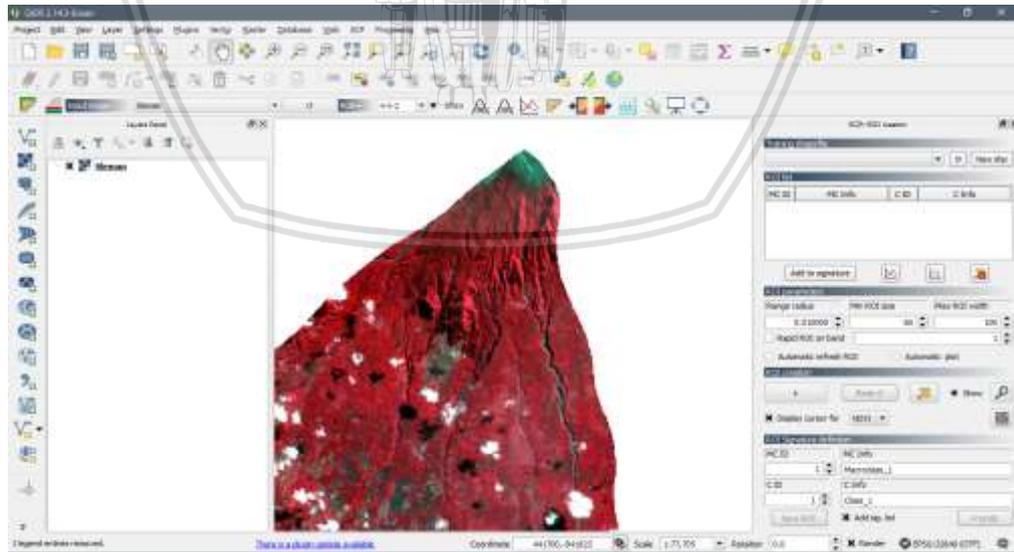


Gambar 5.17 Hasil VRT

5.1.4 Melakukan Klasifikasi Tutupan Lahan

Setelah kita mendapatkan data dalam bentuk VRT, langkah selanjutnya adalah melakukan klasifikasi untuk membedakan daerah yang terkena erupsi dan tidak. Langkah-langkah untuk melakukan klasifikasi adalah sebagai berikut :

1. Buka *Semi Automatic Classification plug-in* dengan menekan menu  pada QGIS. Maka akan muncul tampilan SCP: ROI Creation pada QGIS seperti yang ditunjukkan pada Gambar 5.18.



Gambar 5.18 Tampilan SCP: ROI Creation

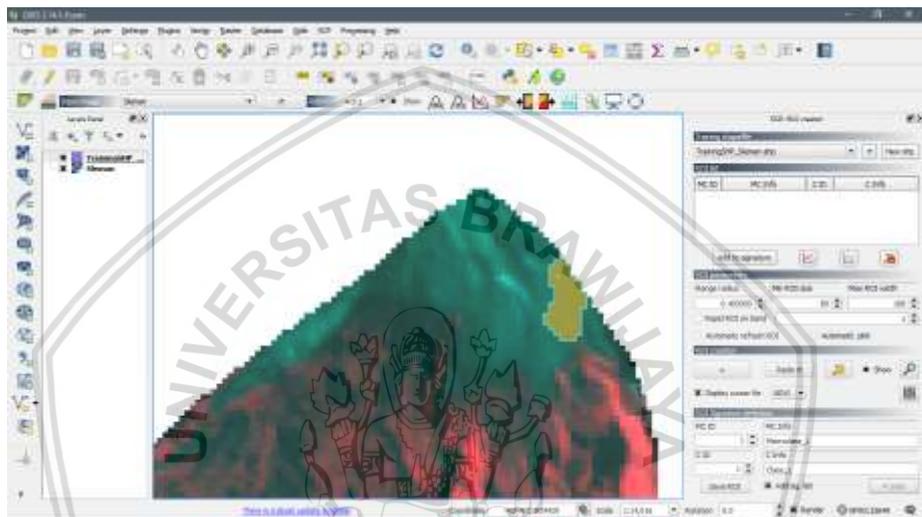
2. Untuk melakukan klasifikasi, terlebih dahulu harus membuat *training shapefile*. Klik pada new SHP seperti pada Gambar 5.19 dan beri nama untuk file tersebut.





Gambar 5.19 Tampilan *training shapefile*

- Setelah file terbuat, langkah selanjutnya adalah membuat ROI. ROI merupakan data latih yang dibuat dalam bentuk *polygon* untuk mendefinisikan jenis dari setiap lokasi. Untuk membuat ROI, gunakan tombol  untuk membuat *polygon* secara otomatis ketika di klik pada VRT, atau tombol  untuk membuat *polygon* secara manual sesuai keinginan kita. Berikut pada Gambar 5.20 merupakan contoh hasil pembuatan *polygon*.



Gambar 5.20 Contoh pembuatan *polygon*

- Setelah membuat sebuah *polygon*, langkah selanjutnya adalah mendefinisikan MC ID, C ID, MC Info, dan C Info pada kolom ROI *Signature definition*. Berikut pada Tabel 5.1 merupakan kategori-kategori dan MC ID yang akan digunakan dalam klasifikasi.

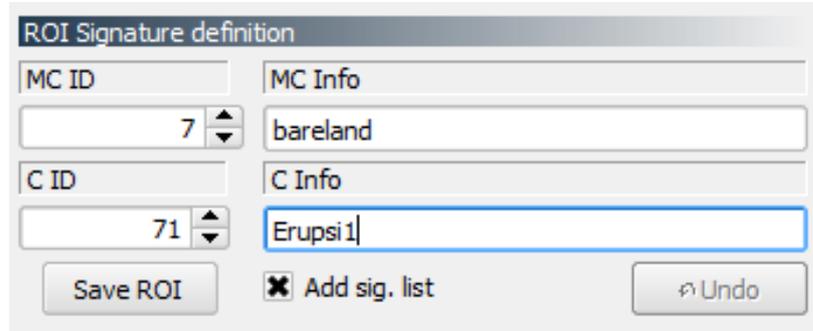
Tabel 5.1 Kategori dan MC ID klasifikasi

| No. | Kategori | MC ID |
|-----|---------------------|-------|
| 1 | <i>Water</i> | 1 |
| 2 | <i>Urban</i> | 2 |
| 3 | <i>Forest</i> | 3 |
| 4 | <i>Agriculture</i> | 4 |
| 5 | <i>Cloud</i> | 5 |
| 6 | <i>Cloud Shadow</i> | 6 |
| 7 | <i>Bareland</i> | 7 |

MC ID dan MC Info merupakan id dan nama kategori utama, misalnya MC ID 7 dengan kategori termasuk bareland. Sedangkan C ID dan C Info merupakan

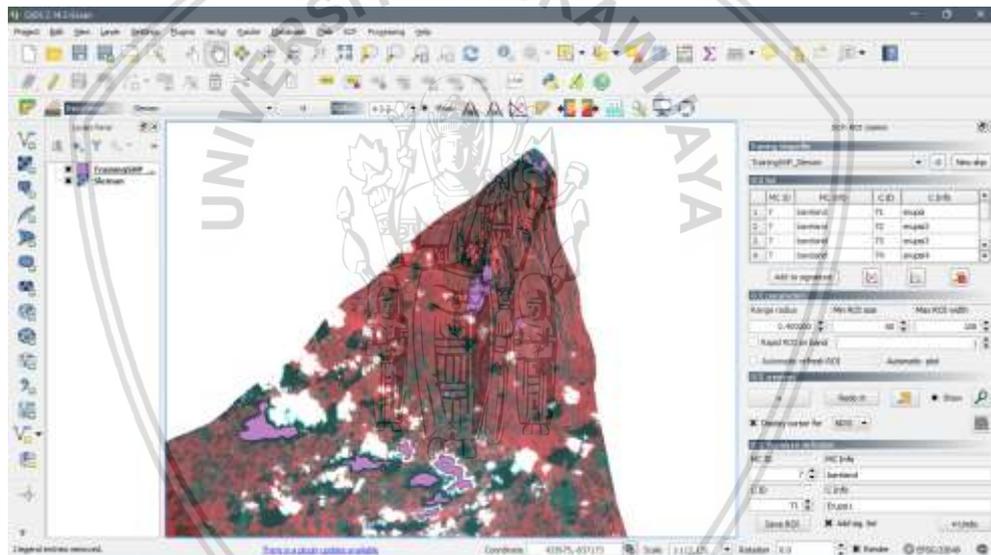


id dan nama sub kategori, misalnya MC ID 71 dengan sub kategori tanah erupsi. Kemudian klik *save ROI*. Contoh dari pembuatan ROI dapat dilihat pada Gambar 5.21.



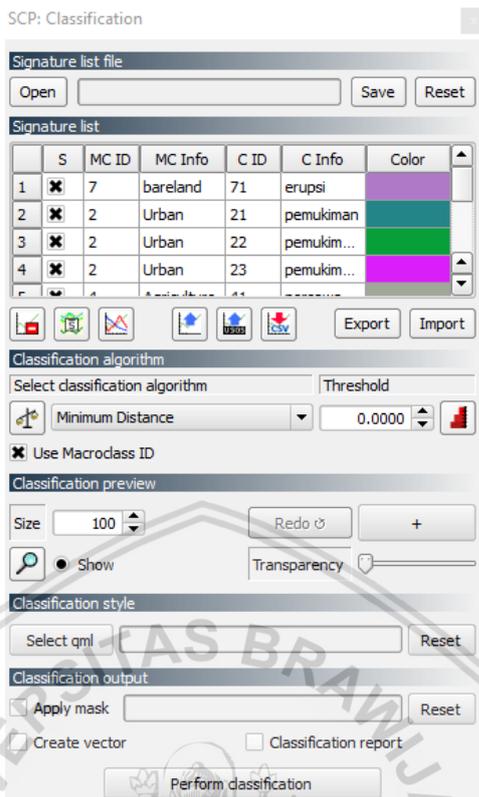
Gambar 5.21 Tampilan ROI *signature definition*

- Untuk mendapatkan hasil klasifikasi yang semakin akurat, ulangi langkah diatas untuk membuat *polygon* sebanyak mungkin seperti yang ditunjukkan pada Gambar 5.22.



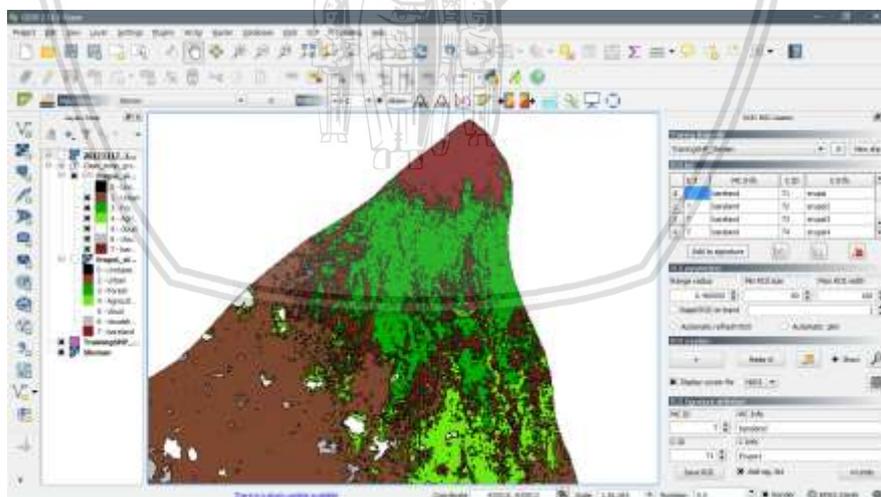
Gambar 5.22 Tampilan setelah pembuatan *polygon-polygon*

- Klik tanda  untuk membuka tampilan SCP: *Classification* seperti pada Gambar 5.23. Pada tampilan tersebut, kita dapat mengubah warna dari setiap kategori klasifikasi yang sudah kita buat dengan *double* klik pada warna. Selanjutnya klik *save*. Pilih algoritme klasifikasi yang akan digunakan yaitu *Maximum Likelihood*. Untuk melihat hasil klasifikasi sebelum dilakukan *perform*, kita dalam menekan *preview*. Untuk mendapatkan hasil klasifikasi, centang pada bagian *create vector* dan *classification report*, kemudian klik *perform classification* dan tunggu hingga proses selesai.



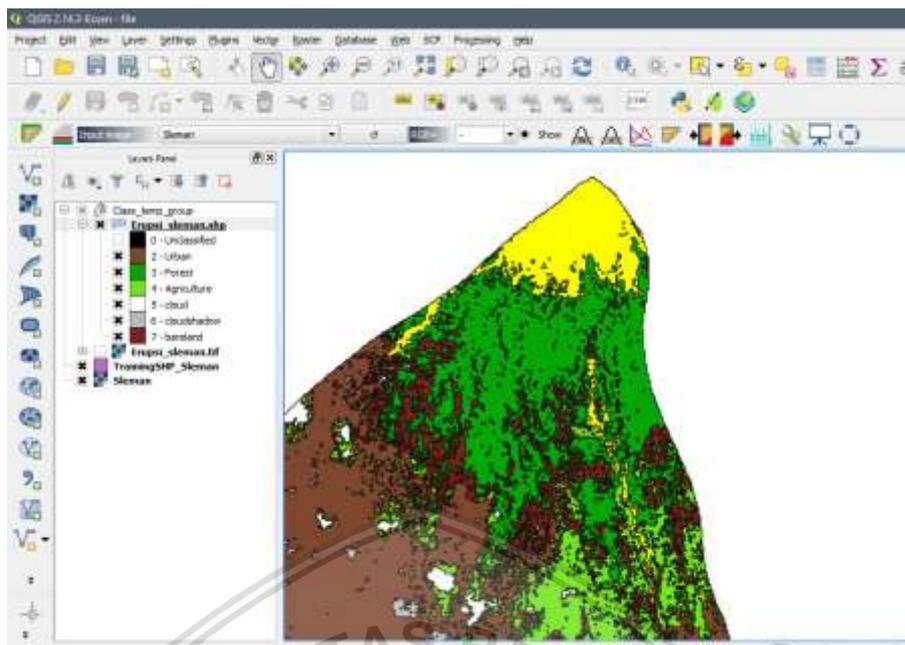
Gambar 5.23 Tampilan SCP: Classification

- Setelah *perform classification* selesai, maka hasil dari klasifikasi adalah seperti yang ditunjukkan pada Gambar 5.24.



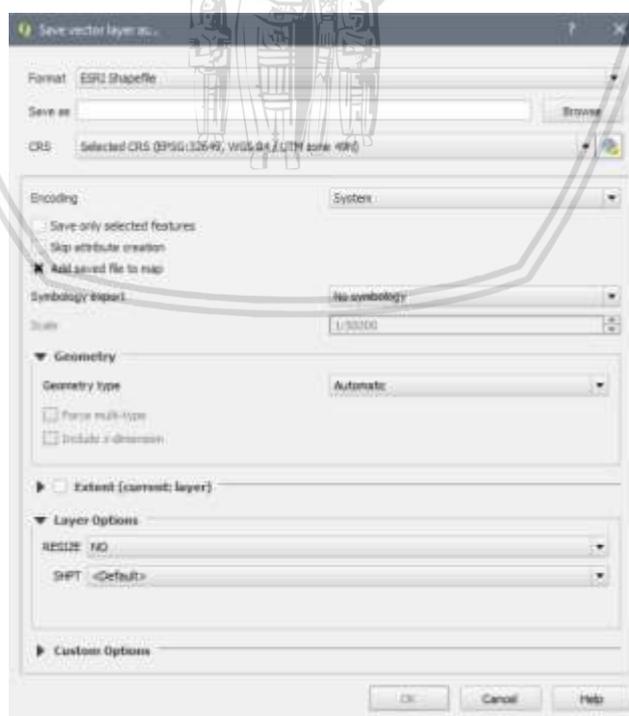
Gambar 5.24 Hasil klasifikasi

- Langkah selanjutnya yang harus dilakukan adalah hanya mengambil hasil klasifikasi yang merupakan daerah yang terkena erupsi. Untuk mengambil daerah yang terkena erupsi saja, pilih icon  kemudian klik pada hasil klasifikasi. Agar bisa memilih banyak daerah erupsi pada hasil klasifikasi, tekan daerah yang terkena erupsi + Ctrl seperti pada Gambar 5.25.

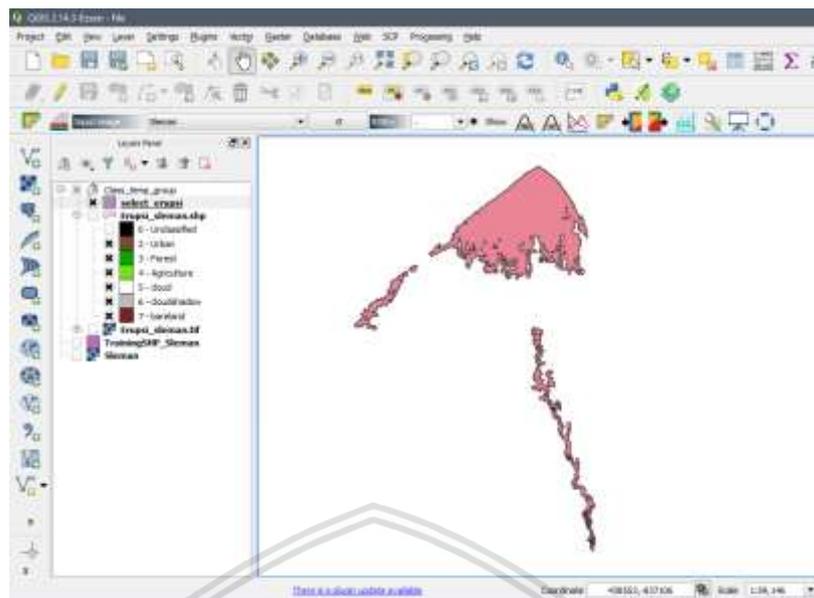


Gambar 5.25 Tampilan memilih daerah yang terkena erupsi

9. Selanjutnya simpan daerah yang terkena erupsi dengan memilih menu *Layer – Save as*, kemudian akan muncul tampilan seperti yang ditunjukkan pada Gambar 5.26. Pada bagian Format pilih *ESRI Shapefile*, pada *Save as* pilih lokasi untuk menyimpan *file SHP* dan nama dari *file* tersebut. Centang pada bagian *Save only selected features*, kemudian klik *OK*. Setelah berhasil disimpan, maka hasilnya akan seperti pada Gambar 5.27.

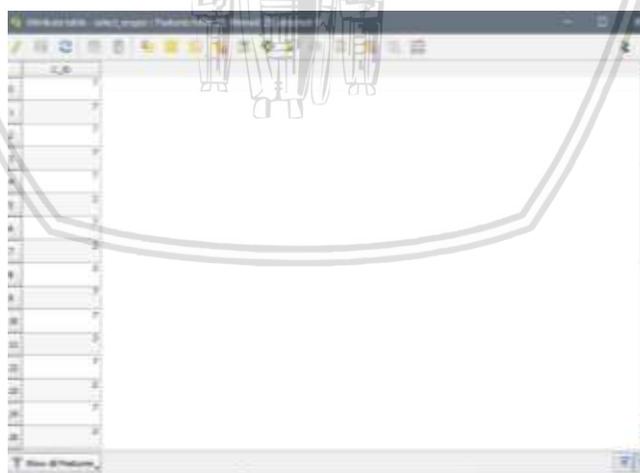


Gambar 5.26 Tampilan *save vector layer as*



Gambar 5.27 Tampilan daerah yang terkena erupsi

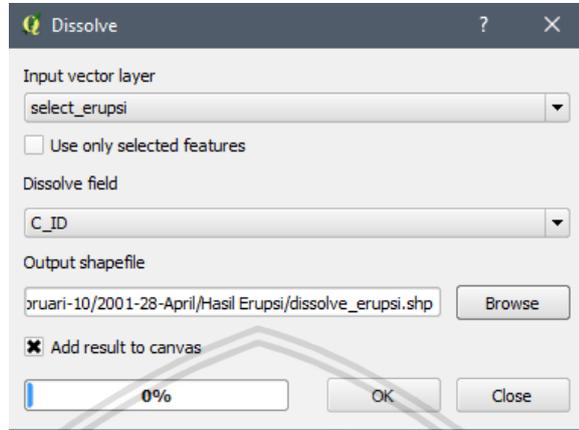
10. Data daerah tersebut belum memiliki C ID yang sama seluruhnya dikarenakan data hasil klasifikasi tidak seluruhnya benar, sehingga harus disamakan terlebih dahulu C ID dari daerah yang telah dipilih sebelumnya. Letakkan cursor pada layer yang hanya berisi daerah yang terkena erupsi, kemudian klik kanan – *Open attribute table*, maka akan tampil seperti pada Gambar 5.28. Urutkan C ID dengan menekan kolom C ID. Untuk mengedit data pada kolom C ID, tekan *icon*  dan ganti nilai C ID dengan angka 7 seluruhnya. Simpan C ID tersebut dengan kembali menekan *icon* .



Gambar 5.28 Tampilan *open attribute table*

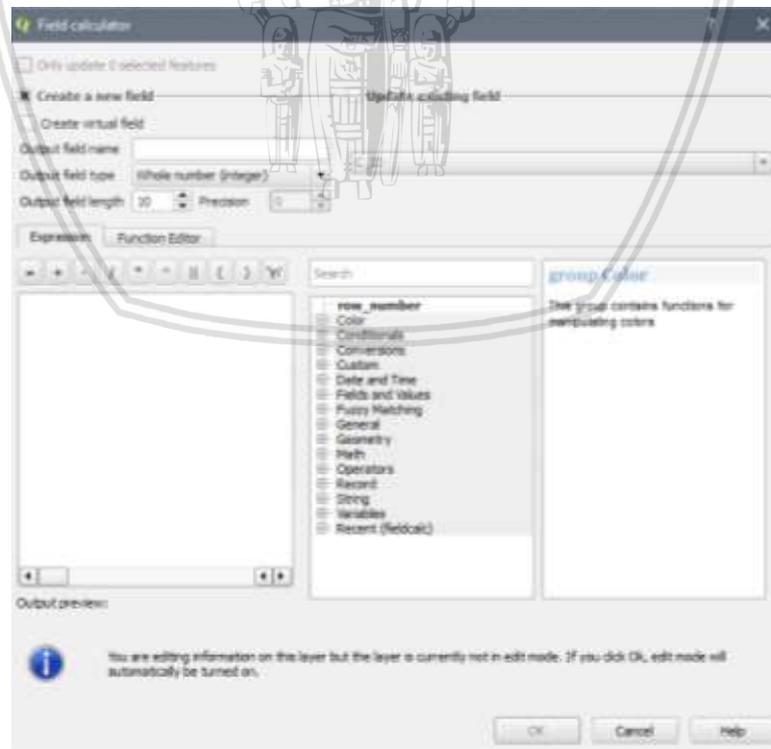
11. Untuk mendapatkan data luas dari daerah yang terkena erupsi, langkah yang harus dilakukan sebelumnya adalah menggabungkan seluruh *polygon* menjadi satu menggunakan *dissolve*. Tekan menu *vector – geoprocessing tools – dissolve*, dan akan muncul tampilan seperti yang ditunjukkan pada Gambar 5.29. Pada *input vector layer* pilih file SHP daerah yang terkena erupsi. Pada *dissolve field* pilih C ID untuk menggabungkan daerah

berdasarkan C ID. Kemudian pilih lokasi untuk menyimpan hasil penggabungan dan beri nama dari *file* tersebut. Klik OK untuk memproses *dissolve*. Apabila kita membuka *open attribute table* pada data hasil dissolve, maka hanya akan terdapat 1 C ID.



Gambar 5.29 Tampilan menu *dissolve*

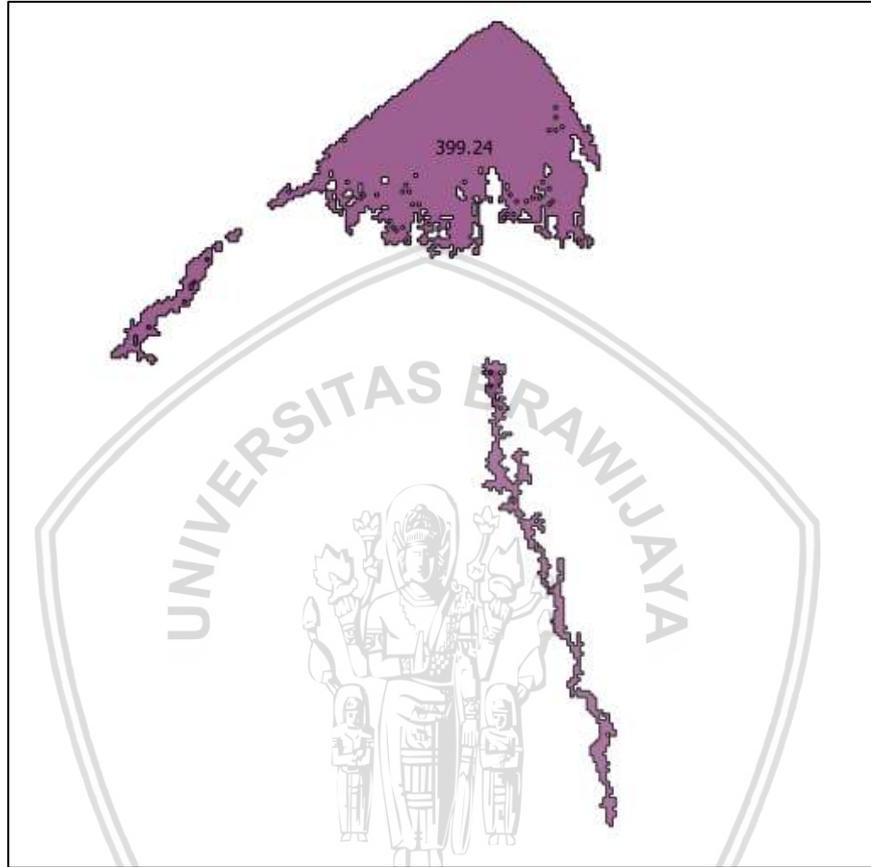
- Untuk menghitung luas daerah, tekan icon  sehingga akan muncul tampilan *calculator field* seperti yang ada pada Gambar 5.30. Isikan luas pada *Output field name*, isikan tipe data *decimal* pada *Output field type*, dan pada *Output field length* isikan 10 dengan *precision* 2. Sedangkan pada kolom *Expression* isikan rumus berikut “ $\$area/10000$ ” untuk mendapatkan luas dalam satuan ha. Kemudian klik OK.



Gambar 5.30 Tampilan *calculator field*



13. Untuk dapat melihat hasil perhitungan luas, buka *open attribute table*, maka akan muncul 2 kolom yaitu kolom C ID dan luas. Dan apabila ingin menampilkan label dari kolom luas pada peta, hanya perlu menampilkan label luas dari *properties*, maka akan tampil seperti yang terdapat pada Gambar 5.31. Pada Gambar tersebut dapat dilihat bahwa luasan daerah yang terdampak erupsi sebesar 399,24 ha.



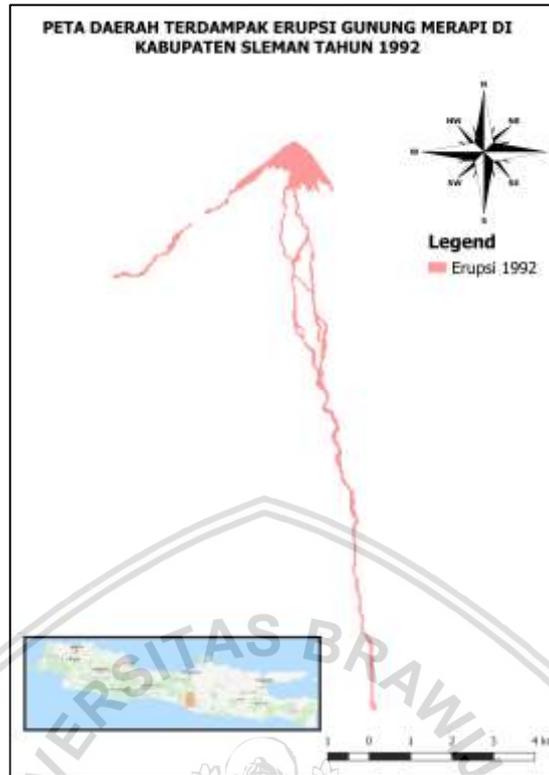
Gambar 5.31 Tampilan label luas daerah

5.1.5 Hasil

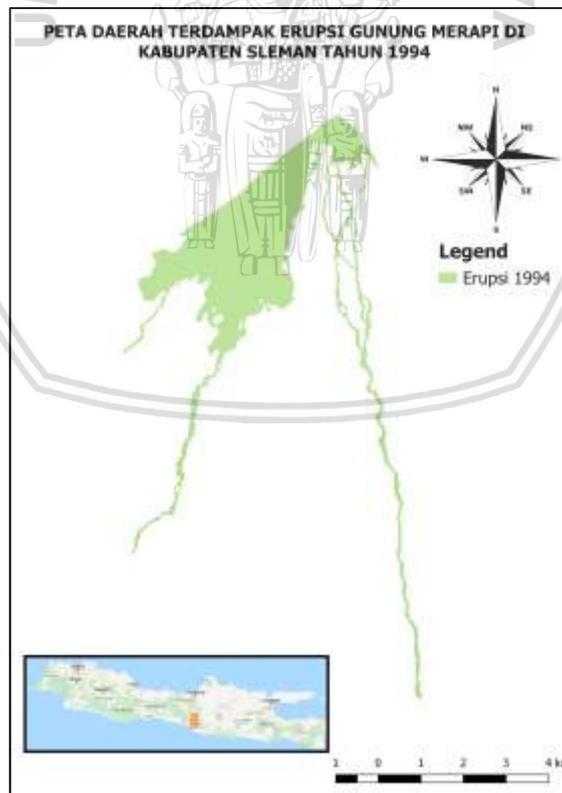
Setelah melakukan tahap klasifikasi tutupan lahan, maka akan didapatkan hasil akhir pemetaan daerah yang terkena erupsi Gunung Merapi di Kabupaten Sleman untuk setiap tahun kejadian yang dapat dilihat pada Gambar 5.32 sampai Gambar 5.38. Setelah dihitung luas dari daerah yang terdampak erupsi untuk pada aplikasi QGIS, selanjutnya adalah menghitung persentase dari luas daerah yang terdampak erupsi untuk setiap tahun kejadian. Untuk mendapatkan persentase tersebut menggunakan perhitungan yang ditunjukkan pada Persamaan (5.1). Namun, hasil persentase yang didapatkan pada sistem merupakan perhitungan persentase otomatis yang dilakukan sistem menggunakan bantuan *library HighCharts*.

$$Persentase = \frac{\text{Luas tahun } n}{\text{Total luas seluruh tahun kejadian}} \times 100\% \quad (5.1)$$

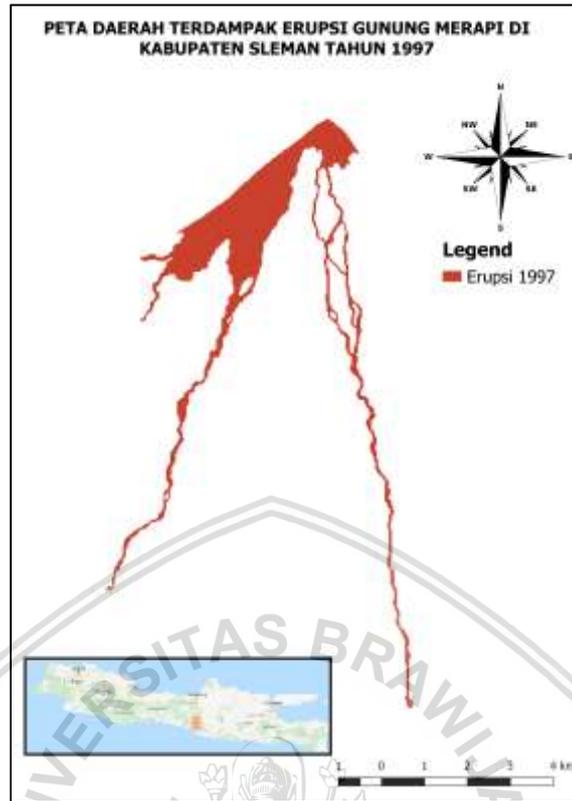




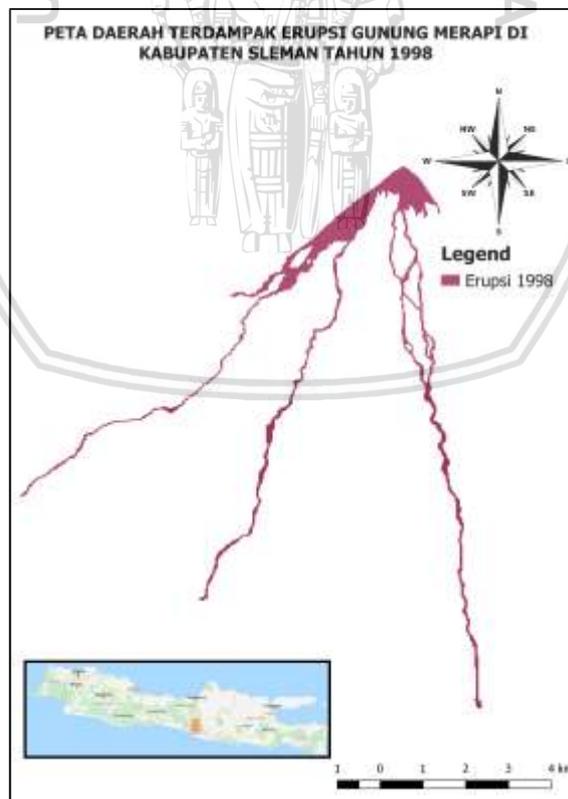
Gambar 5.32 Hasil pemetaan daerah yang terdampak erupsi tahun 1992



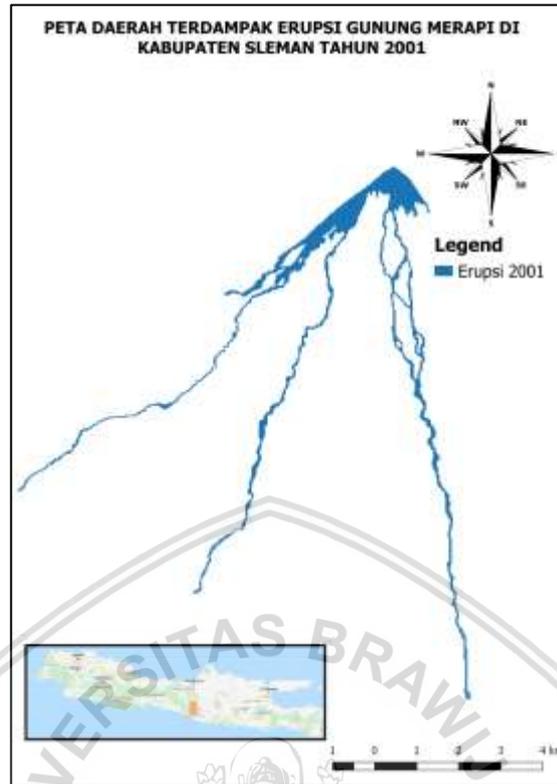
Gambar 5.33 Hasil pemetaan daerah yang terdampak erupsi tahun 1994



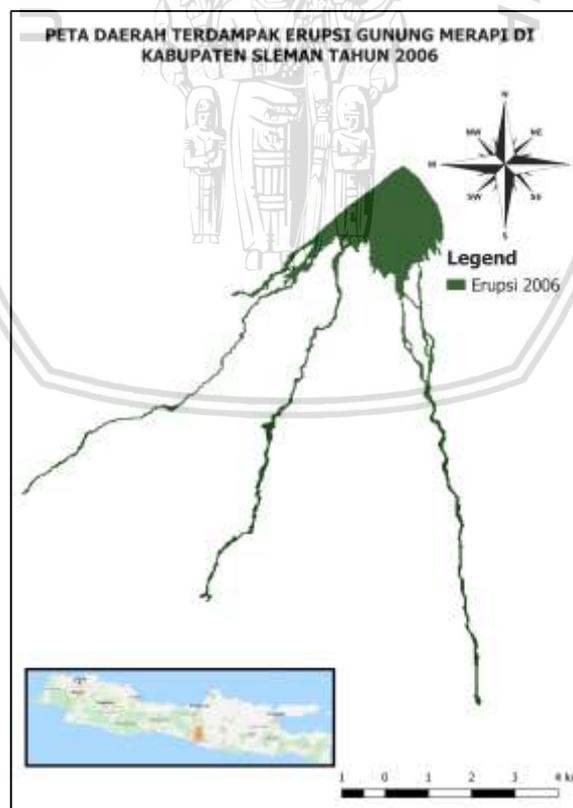
Gambar 5.34 Hasil pemetaan daerah yang terdampak erupsi tahun 1997



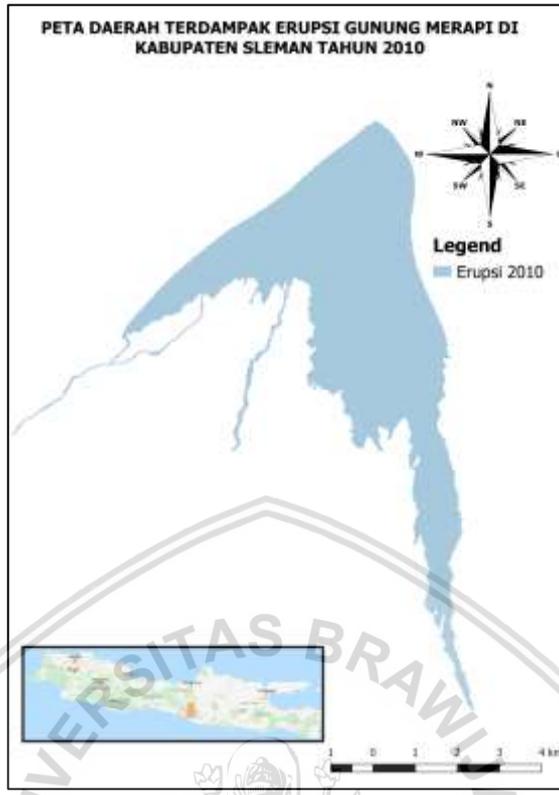
Gambar 5.35 Hasil pemetaan daerah yang terdampak erupsi tahun 1998



Gambar 5.36 Hasil pemetaan daerah yang terdampak erupsi tahun 2001



Gambar 5.37 Hasil pemetaan daerah yang terdampak erupsi tahun 2006



Gambar 5.38 Hasil pemetaan daerah yang terdampak erupsi tahun 2010

5.2 Implementasi WebGIS

Untuk dapat mewujudkan sebuah WebGIS, maka terdapat beberapa tahap yang harus dilakukan, yaitu mengimplementasikan *class* dalam kode program, mengimplementasikan kode program (*source code*), dan mengimplementasikan *user interface*. Dibawah ini akan dibahas mengenai masing-masing proses dalam implementasi WebGIS.

5.2.1 Implementasi Class

Implementasi *class* merupakan proses dimana semua *class* yang dirancang pada tahap perancangan *class diagram* akan diimplementasikan dalam kode program. Implementasi *class* beserta nama file yang menyimpan masing-masing *class* dapat dilihat pada Tabel 5.2.

Tabel 5.2 Implementasi *class controller* dan *model*

| No | Package | Nama Kelas | Nama File | Keterangan |
|----|---------------|---------------|-------------------|---|
| 1 | Cl_Controller | Control_admin | Control_admin.php | <i>Controller</i> ini digunakan untuk meletakkan seluruh fungsi yang mempunyai hubungan dengan kebutuhan fungsional dari Admin. |

Tabel 5.2 Implementasi *class controller* dan *model* (lanjutan)

| No | Package | Nama Kelas | Nama File | Keterangan |
|----|---------------|-----------------|---------------------|---|
| 2 | CI_Controller | Control_general | Control_general.php | <i>Controller</i> ini digunakan untuk meletakkan seluruh fungsi yang mempunyai hubungan dengan kebutuhan fungsional dari <i>user</i> secara umum. |
| 3 | CI_Model | Model_data | Model_data.php | <i>Model class</i> ini digunakan untuk meletakkan seluruh fungsi yang berkaitan dengan pengaksesan data dari tabel data. |
| 4 | CI_Model | Model_info | Model_info.php | <i>Model class</i> ini digunakan untuk meletakkan seluruh fungsi yang berkaitan dengan pengaksesan data dari tabel info. |
| 5 | CI_Model | Model_user | Model_user.php | <i>Model class</i> ini digunakan untuk meletakkan seluruh fungsi yang berkaitan dengan pengaksesan data dari tabel user. |

5.2.2 Implementasi *Source Code*

1. *Source Code* untuk Visualisasi Peta Hasil Pemetaan Daerah yang Terdampak Erupsi pada Halaman *History*

Tabel 5.3 menunjukkan bagaimana kode program pada PHP untuk mengubah data JSON menjadi peta daerah yang terdampak erupsi.

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta

| No | <i>Source Code</i> |
|----|--------------------------|
| 1 | <script> |
| 2 | L.ImageOverlay.include({ |
| 3 | getBounds: function () { |
| 4 | return this._bounds; |
| 5 | } |
| 6 | }); |
| 7 | var map = L.map('map', { |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|----|--|
| 8 | zoomControl:true, maxZoom:28, minZoom:1 |
| 9 | }).fitBounds([[[-7.68776754813,110.36437176],[- 7.48859052141,110.616059931]]]); |
| 10 | var hash = new L.Hash(map); |
| 11 | map.attributionControl.addAttribution('qgis2web'); |
| 12 | var bounds_group = new L.featureGroup([]); |
| 13 | var basemap0 = L.tileLayer('http://{s}.tile.openstreetmap.org/ {z}/{x}/{y}.png', { |
| 14 | attribution: '© OpenStreetMap contributors,CC-BY-SA', |
| 15 | maxZoom: 28 |
| 16 | }); |
| 17 | basemap0.addTo(map); |
| 18 | function setBounds() { |
| 19 | } |
| 20 | var osmGeocoder = new L.Control.OSMGeocoder({ |
| 21 | collapsed: false, |
| 22 | position: 'topright', |
| 23 | text: 'Search', |
| 24 | }); |
| 25 | osmGeocoder.addTo(map); |
| 26 | var baseMaps = {}; |
| 27 | var title = new L.Control(); |
| 28 | title.onAdd = function (map) { |
| 29 | this._div = L.DomUtil.create('div', 'detail'); |
| 30 | this.update(); |
| 31 | return this._div; |
| 32 | }; |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|----|--|
| 33 | title.update = function () { |
| 34 | this._div.innerHTML = '' |
| 35 | }; |
| 36 | title.addTo(map); |
| 37 | var legend = new L.Control({ |
| 38 | position: 'topleft', |
| 39 | }); |
| 40 | legend.onAdd = function (map) { |
| 41 | this._div = L.DomUtil.create('div', 'legend'); // create a div with a class "info" |
| 42 | this.update(); |
| 43 | return this._div; |
| 44 | }; |
| 45 | legend.update = function () { |
| 46 | this._div.innerHTML = '' |
| 47 | }; |
| 48 | legend.addTo(map); |
| 49 | function pop_1992_6(feature, layer) { |
| 50 | var popupContent = '<table class="table">\ |
| 51 | <tr>\ |
| 52 | <th scope="row">Large (ha)</th>\ |
| 53 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 54 | </tr>\ |
| 55 | </table>'; |
| 56 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 57 | } |
| 58 | function style_1992_6_0() { |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|----|--|
| 59 | return { |
| 60 | pane: 'pane_1992_6', |
| 61 | opacity: 1, |
| 62 | color: 'rgba(0,0,0,0.0)', |
| 63 | dashArray: "", |
| 64 | lineCap: 'butt', |
| 65 | lineJoin: 'miter', |
| 66 | weight: 1.0, |
| 67 | fill: true, |
| 68 | fillOpacity: 1, |
| 69 | fillColor: 'rgba(251,154,153,1.0)', |
| 70 | } |
| 71 | } |
| 72 | map.createPane('pane_1992_6'); |
| 73 | map.getPane('pane_1992_6').style.zIndex = 406; |
| 74 | map.getPane('pane_1992_6').style['mix-blend-mode'] = 'normal'; |
| 75 | var layer_1992_6 = new L.geoJson(json_1992, { |
| 76 | attribution: '', |
| 77 | pane: 'pane_1992_6', |
| 78 | onEachFeature: pop_1992_6, |
| 79 | style: style_1992_6_0, |
| 80 | }); |
| 81 | bounds_group.addLayer(layer_1992_6); |
| 82 | map.addLayer(layer_1992_6); |
| 83 | |
| 84 | L.control.layers(baseMaps,{bootstrap/legend/1992.png"/> layer_1992_6},{collapsed:false}).addTo(map); |
| 85 | function pop_1994_5(feature, layer) { |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|---|
| 86 | var popupContent = '<table class="table">\ |
| 87 | <tr>\ |
| 88 | <th scope="row">Large (ha)</th>\ |
| 89 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 90 | </tr>\ |
| 91 | </table>'; |
| 92 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 93 | } |
| 94 | |
| 95 | function style_1994_5_0() { |
| 96 | return { |
| 97 | pane: 'pane_1994_5', |
| 98 | opacity: 1, |
| 99 | color: 'rgba(0,0,0,0.0)', |
| 100 | dashArray: "", |
| 101 | lineCap: 'butt', |
| 102 | lineJoin: 'miter', |
| 103 | weight: 1.0, |
| 104 | fill: true, |
| 105 | fillOpacity: 1, |
| 106 | fillColor: 'rgba(178,223,138,1.0)', |
| 107 | } |
| 108 | } |
| 109 | map.createPane('pane_1994_5'); |
| 110 | map.getPane('pane_1994_5').style.zIndex = 405; |
| 111 | map.getPane('pane_1994_5').style['mix-blend-mode'] = 'normal'; |
| 112 | var layer_1994_5 = new L.geoJson(json_1994, { |
| 113 | attribution: '', |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|---|
| 114 | pane: 'pane_1994_5', |
| 115 | onEachFeature: pop_1994_5, |
| 116 | style: style_1994_5_0, |
| 117 | }); |
| 118 | bounds_group.addLayer(layer_1994_5); |
| 119 | map.addLayer(layer_1994_5); |
| 120 | L.control.layers(baseMaps,{ layer_1994_5},{collapsed:false}).addTo(map); |
| 121 | function pop_1997_4(feature, layer) { |
| 122 | var popupContent = '<table class="table">\ |
| 123 | <tr>\ |
| 124 | <th scope="row">Large (ha)</th>\ |
| 125 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 126 | </tr>\ |
| 127 | </table>; |
| 128 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 129 | } |
| 130 | function style_1997_4_0() { |
| 131 | return { |
| 132 | pane: 'pane_1997_4', |
| 133 | opacity: 1, |
| 134 | color: 'rgba(0,0,0,0.0)', |
| 135 | dashArray: '', |
| 136 | lineCap: 'butt', |
| 137 | lineJoin: 'miter', |
| 138 | weight: 1.0, |
| 139 | fill: true, |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|--|
| 140 | fillOpacity: 1, |
| 141 | fillColor: 'rgba(200,74,39,1.0)', |
| 142 | } |
| 143 | } |
| 144 | map.createPane('pane_1997_4'); |
| 145 | map.getPane('pane_1997_4').style.zIndex = 404; |
| 146 | map.getPane('pane_1997_4').style['mix-blend-mode'] = 'normal'; |
| 147 | var layer_1997_4 = new L.geoJson(json_1997, { |
| 148 | attribution: '', |
| 149 | pane: 'pane_1997_4', |
| 150 | onEachFeature: pop_1997_4, |
| 151 | style: style_1997_4_0, |
| 152 | }); |
| 153 | bounds_group.addLayer(layer_1997_4); |
| 154 | map.addLayer(layer_1997_4); |
| 155 | |
| 156 | L.control.layers(baseMaps,{bootstrap/legend/1997.png" /> 1997': layer_1997_4},{collapsed:false}).addTo(map); |
| 157 | function pop_1998_3(feature, layer) { |
| 158 | var popupContent = '<table class="table">\ |
| 159 | <tr>\ |
| 160 | <th scope="row">Large (ha)</th>\ |
| 161 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 162 | </tr>\ |
| 163 | </table>'; |
| 164 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 165 | } |
| 166 | function style_1998_3_0() { |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|---|
| 167 | return { |
| 168 | pane: 'pane_1998_3', |
| 169 | opacity: 1, |
| 170 | color: 'rgba(0,0,0,0.0)', |
| 171 | dashArray: "", |
| 172 | lineCap: 'butt', |
| 173 | lineJoin: 'miter', |
| 174 | weight: 1.0, |
| 175 | fill: true, |
| 176 | fillOpacity: 1, |
| 177 | fillColor: 'rgba(183,16,86,1.0)', |
| 178 | } |
| 179 | } |
| 180 | map.createPane('pane_1998_3'); |
| 181 | map.getPane('pane_1998_3').style.zIndex = 403; |
| 182 | map.getPane('pane_1998_3').style['mix-blend-mode'] = 'normal'; |
| 183 | var layer_1998_3 = new L.geoJson(json_1998, { |
| 184 | attribution: '', |
| 185 | pane: 'pane_1998_3', |
| 186 | onEachFeature: pop_1998_3, |
| 187 | style: style_1998_3_0, |
| 188 | }); |
| 189 | bounds_group.addLayer(layer_1998_3); |
| 190 | map.addLayer(layer_1998_3); |
| 191 | |
| 192 | L.control.layers(baseMaps,{bootstrap/legend/1998.png" /> 1998': layer_1998_3,},{collapsed:false}).addTo(map); |
| 193 | function pop_2001_2(feature, layer) { |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|---|
| 194 | var popupContent = '<table class="table">\ |
| 195 | <tr>\ |
| 196 | <th scope="row">Large (ha)</th>\ |
| 197 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 198 | </tr>\ |
| 199 | </table>'; |
| 200 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 201 | } |
| 202 | function style_2001_2_0() { |
| 203 | return { |
| 204 | pane: 'pane_2001_2', |
| 205 | opacity: 1, |
| 206 | color: 'rgba(0,0,0,0.0)', |
| 207 | dashArray: '', |
| 208 | lineCap: 'butt', |
| 209 | lineJoin: 'miter', |
| 210 | weight: 1.0, |
| 211 | fill: true, |
| 212 | fillOpacity: 1, |
| 213 | fillColor: 'rgba(31,120,180,1.0)', |
| 214 | } |
| 215 | } |
| 216 | map.createPane('pane_2001_2'); |
| 217 | map.getPane('pane_2001_2').style.zIndex = 402; |
| 218 | map.getPane('pane_2001_2').style['mix-blend-mode'] = 'normal'; |
| 219 | var layer_2001_2 = new L.geoJson(json_2001, { |
| 220 | attribution: '', |
| 221 | pane: 'pane_2001_2', |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|--|
| 222 | onEachFeature: pop_2001_2, |
| 223 | style: style_2001_2_0, |
| 224 | }); |
| 225 | bounds_group.addLayer(layer_2001_2); |
| 226 | map.addLayer(layer_2001_2); |
| 227 | L.control.layers(baseMaps,{ 2001': layer_2001_2},{collapsed:false}).addTo(map); |
| 228 | function pop_2006_1(feature, layer) { |
| 229 | var popupContent = '<table class="table">\ |
| 230 | <tr>\ |
| 231 | <th scope="row">Large (ha)</th>\ |
| 232 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 233 | </tr>\ |
| 234 | </table>'; |
| 235 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 236 | } |
| 237 | function style_2006_1_0() { |
| 238 | return { |
| 239 | pane: 'pane_2006_1', |
| 240 | opacity: 1, |
| 241 | color: 'rgba(0,0,0,0.0)', |
| 242 | dashArray: "", |
| 243 | lineCap: 'butt', |
| 244 | lineJoin: 'miter', |
| 245 | weight: 1.0, |
| 246 | fill: true, |
| 247 | fillOpacity: 1, |

Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|--|
| 248 | fillColor: 'rgba(23,78,15,1.0)', |
| 249 | } |
| 250 | } |
| 251 | map.createPane('pane_2006_1'); |
| 252 | map.getPane('pane_2006_1').style.zIndex = 401; |
| 253 | map.getPane('pane_2006_1').style['mix-blend-mode'] = 'normal'; |
| 254 | var layer_2006_1 = new L.geoJson(json_2006, { |
| 255 | attribution: '', |
| 256 | pane: 'pane_2006_1', |
| 257 | onEachFeature: pop_2006_1, |
| 258 | style: style_2006_1_0, |
| 259 | }); |
| 260 | bounds_group.addLayer(layer_2006_1); |
| 261 | map.addLayer(layer_2006_1); |
| 262 | |
| 263 | L.control.layers(baseMaps,{ 2006': layer_2006_1},{collapsed:false}).addTo(map); |
| 264 | function pop_2010_0(feature, layer) { |
| 265 | var popupContent = '<table class="table">\ |
| 266 | <tr>\ |
| 267 | <th scope="row">Large (ha)</th>\ |
| 268 | <td>' + (feature.properties['Luas (ha)'] !== null ? Autolinker.link(String(feature.properties['Luas (ha)'])) : '') + '</td>\ |
| 269 | </tr>\ |
| 270 | </table>'; |
| 271 | layer.bindPopup(popupContent, {maxHeight: 400}); |
| 272 | } |
| 273 | function style_2010_0_0() { |
| 274 | return { |

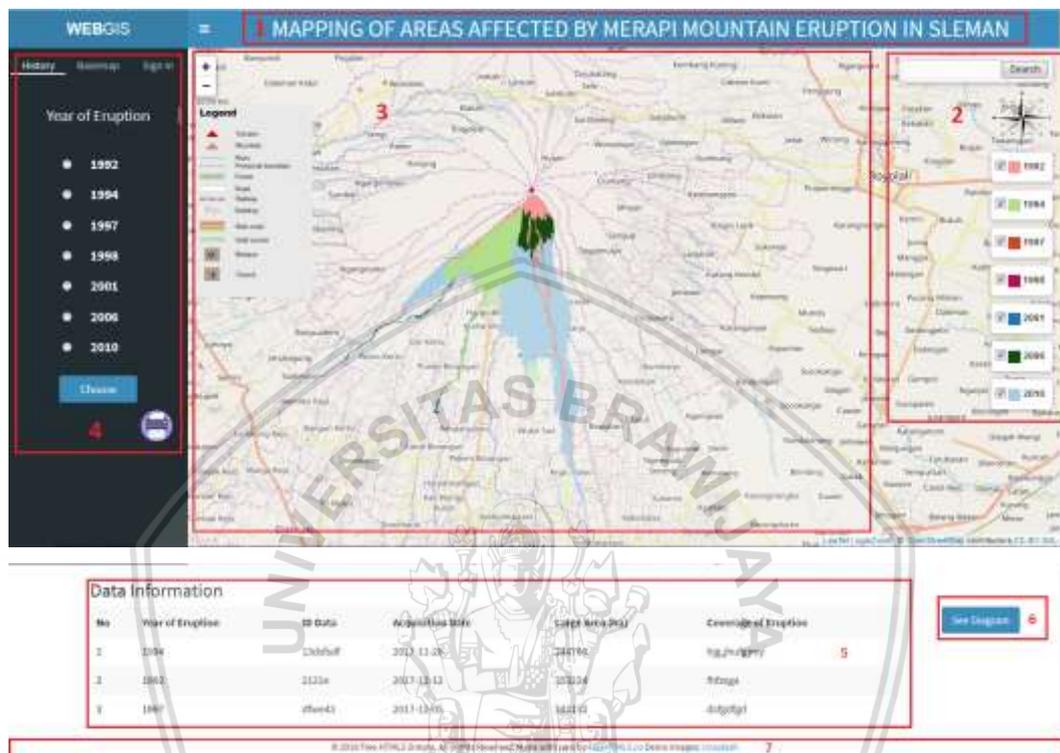
Tabel 5.3 *Source code* untuk mengubah data JSON menjadi peta (lanjutan)

| No | Source Code |
|-----|--|
| 275 | pane: 'pane_2010_0', |
| 276 | opacity: 1, |
| 277 | color: 'rgba(166,206,227,0.0)', |
| 278 | dashArray: '', |
| 279 | lineCap: 'butt', |
| 280 | lineJoin: 'miter', |
| 281 | weight: 1.0, |
| 282 | fill: true, |
| 283 | fillOpacity: 1, |
| 284 | fillColor: 'rgba(166,206,227,1.0)', |
| 285 | } |
| 286 | } |
| 287 | map.createPane('pane_2010_0'); |
| 288 | map.getPane('pane_2010_0').style.zIndex = 400; |
| 289 | map.getPane('pane_2010_0').style['mix-blend-mode'] = 'normal'; |
| 290 | var layer_2010_0 = new L.geoJson(json_2010, { |
| 291 | attribution: '', |
| 292 | pane: 'pane_2010_0', |
| 293 | onEachFeature: pop_2010_0, |
| 294 | style: style_2010_0_0, |
| 295 | }); |
| 296 | bounds_group.addLayer(layer_2010_0); |
| 297 | map.addLayer(layer_2010_0); |
| 298 | L.control.layers(baseMaps,{ 2010': layer_2010_0},{collapsed:false}).addTo(map); |
| 299 | setBounds(); |
| 300 | </script> |

5.2.3 Implementasi Antarmuka

1. Antarmuka Halaman *History*

Gambar 5.39 merupakan implementasi antarmuka untuk menampilkan halaman *History*. *User* dapat melihat hasil pemetaan daerah yang terdampak erupsi di Kabupaten Sleman pada halaman ini.



Gambar 5.39 Antarmuka halaman *history*

Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *History* :

1. *Header* dari sistem yang berisi nama dari sistem.
2. Arah mata angin untuk mempermudah mengetahui arah, kolom *Search* untuk mencari suatu lokasi, dan informasi tahun kejadian erupsi.
3. Visualisasi peta hasil pemetaan serta tombol *Zoom in*, *Zoom out*, dan *Legend* dari *basemap* yang ditampilkan.
4. Menu navigasi dari sistem antara lain *History*, *Basemap*, dan *Sign In*. Pada halaman *History* juga akan ditampilkan pilihan tahun kejadian erupsi yang dapat dipilih oleh *user*. *Selain itu*, terdapat tombol *print* yang digunakan untuk mencetak peta hasil pemetaan yang divisualisasikan.
5. Digunakan untuk menampilkan informasi detail dari peta yang divisualisasikan, yaitu *Year of Eruption*, *ID Data*, *Acquisition Date*, *Large Area* dalam satuan ha, dan *Coverage of Eruption*.

6. Tombol *See Diagram* digunakan untuk melihat diagram yang menunjukkan luasan daerah hasil pemetaan.

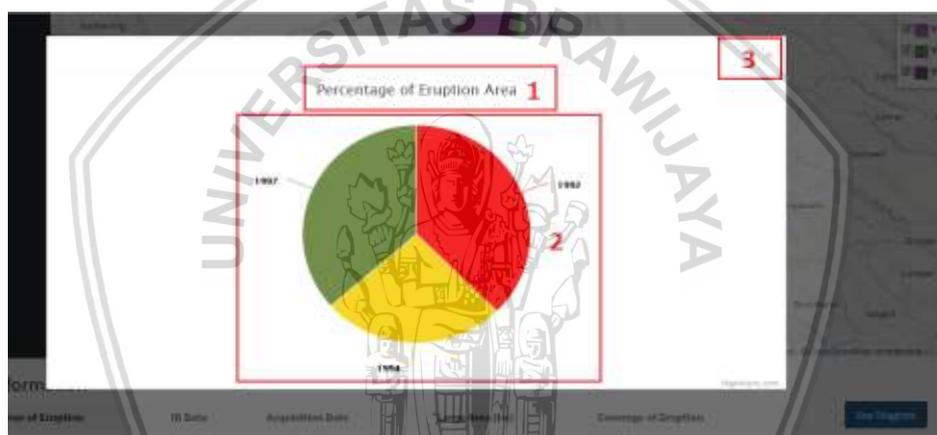
7. Footer dari sistem.

2. Antarmuka Halaman Diagram

Gambar 5.40 merupakan implementasi antarmuka untuk halaman diagram, yang menunjukkan persentase luasan daerah yang terdampak erupsi pada setiap tahun kejadian.

Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman diagram :

1. Judul *popup* diagram yang ditampilkan.
2. *Pie Diagram* yang menunjukkan luasan daerah yang terdampak erupsi pada setiap tahun kejadian yang ditambahkan ke sistem.
3. Tombol *Close* untuk menutup *popup*.

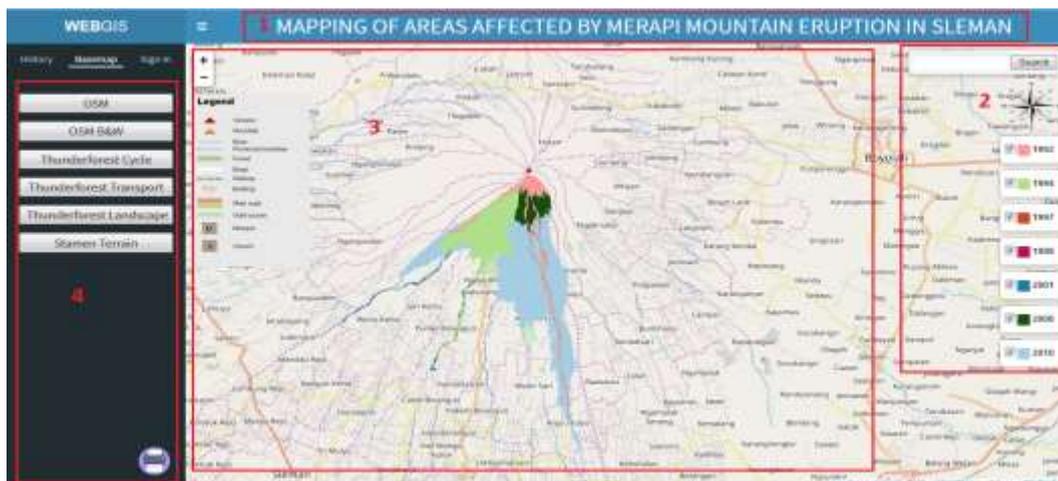


Gambar 5.40 Antarmuka halaman diagram

3. Antarmuka Halaman *Basemap*

Gambar 5.41 merupakan implementasi antarmuka untuk memvisualisasikan peta beserta *basemap* yang dipilih oleh *user*. Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *Basemap* :

1. *Header* sistem yang berisi nama dari sistem.
2. Arah mata angin untuk mempermudah mengetahui arah, kolom *Search* untuk mencari suatu lokasi, dan informasi tahun kejadian erupsi.
3. Visualisasi hasil pemetaan daerah yang terkena erupsi dalam bentuk peta, tombol *Zoom in*, *Zoom out*, dan *Legend* dari *basemap* yang ditampilkan.
4. Pilihan-pilihan *basemap* yang disediakan oleh sistem dan dapat dipilih oleh *user*. Sedangkan tombol *print* digunakan untuk mencetak peta yang divisualisasikan beserta *basemap* yang telah dipilih.



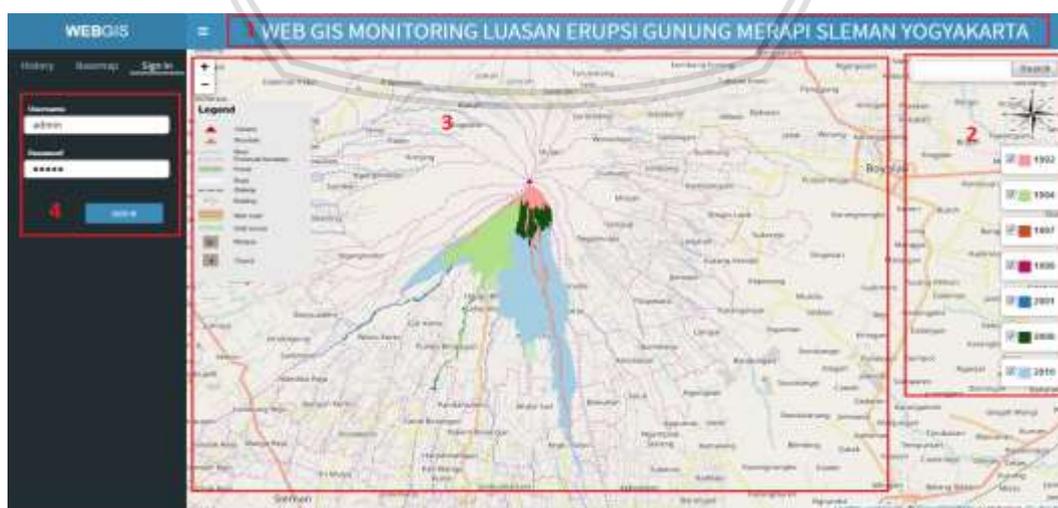
Gambar 5.41 Antarmuka halaman *Basemap*

4. Antarmuka Halaman *Sign In*

Pada Gambar 5.42 merupakan antarmuka yang menampilkan halaman *Sign In* dari sistem, sehingga Admin dapat masuk ke sistem.

Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *Sign In* :

1. Header dari sistem yang berisi nama dari sistem.
2. Arah mata angin untuk mempermudah mengetahui arah, kolom Search untuk mencari suatu lokasi, dan informasi tahun kejadian erupsi.
3. Visualisasi hasil pemetaan daerah yang terkena erupsi, tombol Zoom in, Zoom out, dan Legend dari basemap yang ditampilkan.
4. Kolom *username* yang digunakan untuk memasukkan *Username* dari Admin, kolom *password* digunakan untuk memasukkan *password*, dan tombol *Sign In* yang digunakan untuk masuk ke dalam sistem.



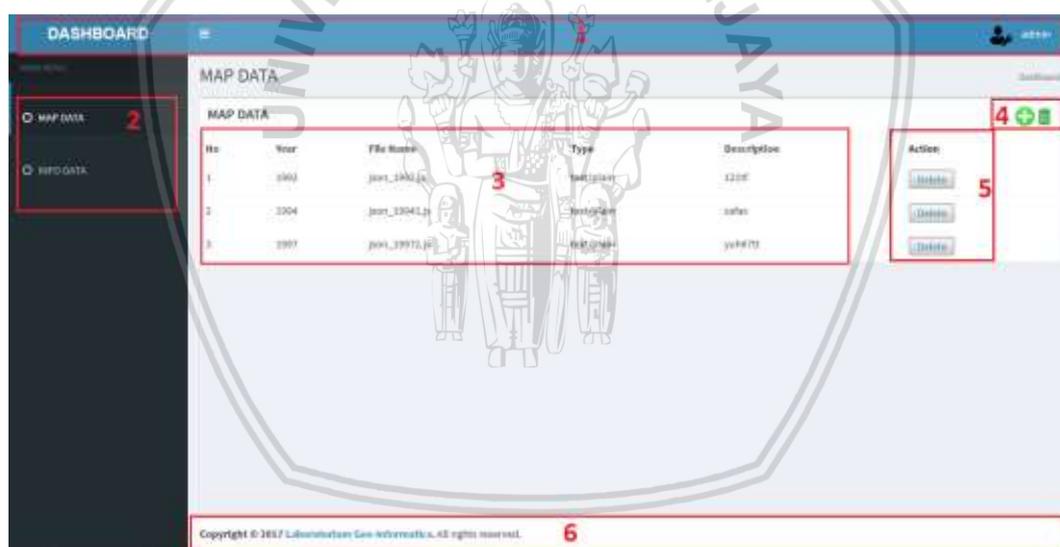
Gambar 5.42 Antarmuka halaman *sign in*

5. Antarmuka Halaman *Map Data*

Gambar 5.43 merupakan implementasi antarmuka untuk dapat melihat seluruh data JSON yang akan ditampilkan sebagai peta di halaman *History*.

Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada halaman *Map Data* :

1. *Header* dari *dashboard*, dimana terdapat menu *Sign Out*, keterangan admin, dan keterangan yang menunjukkan bahwa halaman ini adalah bagian dari *dashboard*.
2. Menu-menu yang ada pada dashboard, yaitu menu *Map Data* dan *Info Data*.
3. Tabel untuk menampilkan seluruh data JSON.
4. Tombol *Add* digunakan untuk menambahkan data JSON baru, sedangkan tombol *Delete* digunakan untuk menghapus semua data yang ada pada halaman *Map Data*.
5. Tombol *Delete* pada setiap baris data digunakan untuk menghapus data per baris.
6. *Footer dashboard* Admin.



Gambar 5.43 Antarmuka halaman *map data*

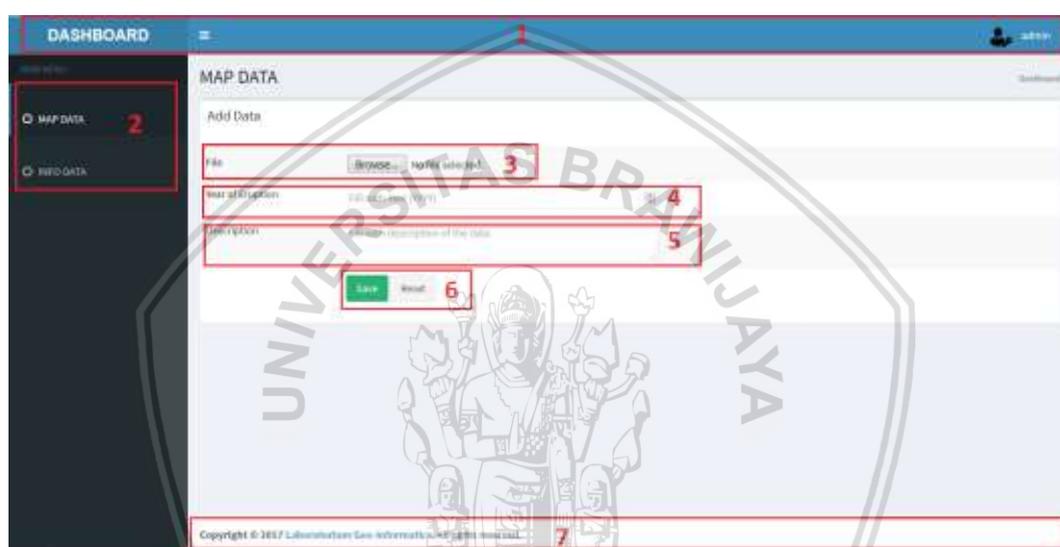
6. Antarmuka Halaman *Form Tambah Data*

Gambar 5.44 merupakan implementasi antarmuka untuk halaman tambah data JSON, sehingga Admin dapat menambahkan data JSON baru di halaman *Map Data*.

Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada antarmuka halaman *form* tambah data :

1. *Header* dari dashboard dimana terdapat menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini merupakan bagian dari dashboard.

2. Menu-menu yang ada pada dashboard, yaitu menu Map Data dan Info Data.
3. Tombol *Browse* digunakan untuk mencari dan memilih peta baru yang ingin diinputkan.
4. Kolom *Year of Eruption* digunakan untuk menginputkan tahun dari data yang akan ditambahkan.
5. Kolom *Description* digunakan untuk memasukkan keterangan tentang data JSON yang ingin ditambahkan.
6. Tombol *Save* digunakan untuk menyimpan data ke *database*, sedangkan tombol *Reset* untuk mereset data pada kolom.
7. *Footer* dashboard admin.



Gambar 5.44 Antarmuka halaman *form* tambah data

7. Antarmuka Halaman *Info Data*

Gambar 5.45 merupakan implementasi antarmuka untuk halaman *Info Data*. Halaman ini menampilkan seluruh data informasi yang pernah ditambahkan ke sistem.

Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada antarmuka halaman *Info Data* :

1. *Header* dari *dashboard* dimana terdapat menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini adalah bagian dari *dashboard*.
2. Menu-menu yang ada pada dashboard, yaitu menu *Map Data* dan *Info Data*.
3. Tabel yang digunakan untuk menampilkan seluruh data informasi peta yang telah ditambahkan sebelumnya.
4. Tombol *Add* digunakan untuk menambahkan data JSON baru, sedangkan tombol *Delete* digunakan untuk menghapus semua data yang ada pada halaman *Info Data*.

5. Tombol *Delete* pada tiap baris untuk menghapus data informasi per baris. Dan tombol *Edit* untuk memperbarui data informasi per baris.
6. *Footer* dari halaman *Info Data*.

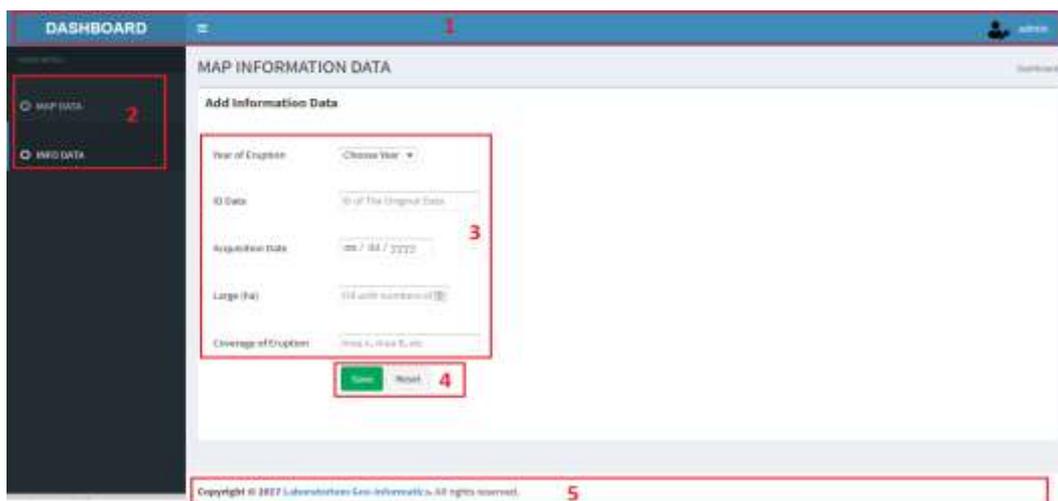


Gambar 5.45 Antarmuka halaman *info data*

8. Antarmuka Halaman *Form Tambah Info*

Gambar 5.46 merupakan implementasi antarmuka untuk halaman yang menampilkan form untuk menambah data informasi. Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada antarmuka halaman *form* tambah info :

1. Header dari dashboard, dimana terdapat beberapa menu yaitu menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini adalah bagian dari dashboard.
2. Menu-menu yang ada pada dashboard, yaitu menu *Map Data* dan *Info Data*.
3. Kolom-kolom input yang harus dimasukkan ketika menambahkan data informasi baru, yaitu kolom *year of eruption*, *id data*, *acquisition date*, *large* dalam satuan ha, dan *coverage of eruption*.
4. Tombol *Save* yang digunakan untuk menyimpan data ke *database*, sedangkan tombol *Reset* digunakan untuk mereset data.
5. Bagian *footer* dari halaman tambah info.

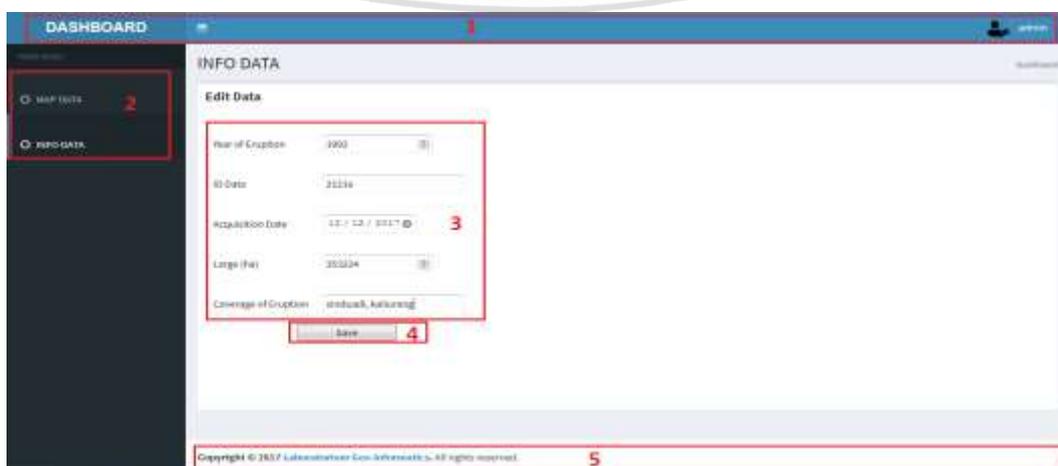


Gambar 5.46 Antarmuka halaman *form* tambah info

9. Antarmuka Antarmuka Halaman *Form Edit Info*

Gambar 5.47 merupakan implementasi antarmuka untuk halaman yang menampilkan form untuk memperbarui data informasi pada baris tertentu. Dibawah ini merupakan keterangan dari bagian-bagian yang ada pada antarmuka halaman *form edit info* :

1. Header dari dashboard, dimana terdapat beberapa menu yaitu menu *Sign Out*, keterangan admin, dan keterangan yang menjelaskan bahwa halaman ini adalah bagian dari dashboard.
2. Menu-menu yang ada pada dashboard, yaitu menu *Map Data* dan *Info Data*.
3. Kolom-kolom input yang harus dimasukkan ketika memperbarui data informasi, yaitu kolom *year of eruption*, *id data*, *acquisition date*, *large* dalam satuan ha, dan *coverage of eruption*.
4. Tombol *Save* yang digunakan untuk menyimpan perubahan data ke *database*.
5. Bagian *footer* dari halaman *edit info*.



Gambar 5.47 Antarmuka halaman *form edit info*

BAB 6 PENGUJIAN

Bab ini berisi tentang pengujian yang dilakukan terhadap sistem dan hasil pemetaan daerah yang terdampak erupsi. Terdapat 3 jenis pengujian yang dilakukan pada bab ini, yaitu Pengujian *Black-Box* (Uji Validasi), Pengujian *Portability*, dan Pengujian *Accuracy*. Berikut dibawah ini akan dijelaskan mengenai tujuan, mekanisme, hasil, dan analisis dari setiap pengujian yang dilakukan.

6.1 Pengujian *Black-Box* (Uji Validasi)

6.1.1 Tujuan Pengujian

Pengujian *Black-Box* yaitu uji validasi dilakukan untuk memastikan semua kebutuhan fungsional yang didefinisikan pada tahap sebelumnya telah terpenuhi oleh sistem.

6.1.2 Mekanisme Pengujian

Mekanisme dari pengujian ini adalah dengan membuat *test case* (kasus uji) berdasarkan kebutuhan fungsional yang didefinisikan pada tahap sebelumnya. Dalam setiap *test case* akan didefinisikan hasil yang diharapkan, kemudian hasil dari pengujian dengan hasil yang diharapkan akan dibandingkan. Apabila kedua hasil tersebut adalah sama, maka kebutuhan fungsional telah terpenuhi pada sistem dengan status valid.

1. *Test Case* Melihat *History* Erupsi

Kasus uji validasi untuk *use case* melihat *history* erupsi ditunjukkan pada Tabel 6.1 berikut.

Tabel 6.1 *Test case* melihat *history* erupsi

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-01 |
| Nama Kasus Uji | Melihat <i>history</i> erupsi |
| Kode Use Case | MIS-1-001 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan peta hasil pemetaan pada halaman <i>History</i> |
| Test Case | Pengujian menampilkan seluruh peta hasil pemetaan. |
| Prosedur Pengujian | 1. Penguji memilih menu <i>History</i> . |
| Hasil yang Diharapkan | Sistem menampilkan seluruh hasil pemetaan pada halaman <i>History</i> dalam bentuk peta. |

2. Test Case Melihat Informasi *History* Erupsi

Kasus uji validasi untuk *use case* melihat informasi *history* erupsi ditunjukkan pada Tabel 6.2 berikut.

Tabel 6.2 Test case melihat informasi *history* erupsi

| | |
|------------------------------|--|
| Nomor Kasus Uji | TEST-02 |
| Nama Kasus Uji | Melihat informasi <i>history</i> erupsi |
| Kode Use Case | MIS-1-002 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan informasi dari peta hasil pemetaan daerah yang terkena erupsi. |
| Test Case | Pengujian menampilkan informasi dari seluruh peta hasil pemetaan yang divisualisasikan. |
| Prosedur Pengujian | 1. Penguji memilih menu <i>History</i> . |
| Hasil yang Diharapkan | Sistem menampilkan informasi dari seluruh peta hasil pemetaan. |

3. Test Case Memilih *Year Layer*

Kasus uji validasi untuk *use case* memilih *year layer* ditunjukkan pada Tabel 6.3 berikut.

Tabel 6.3 Test case memilih *year layer*

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-03 |
| Nama Kasus Uji | Memilih <i>year layer</i> |
| Kode Use Case | MIS-1-003 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan hasil pemetaan daerah yang terkena erupsi di Kabupaten Sleman berdasarkan tahun kejadian erupsi yang dipilih dalam bentuk peta. |
| Test Case | Pengujian menampilkan peta hasil pemetaan berdasarkan tahun kejadian erupsi yang dipilih. |
| Prosedur Pengujian | 1. Penguji memilih menu <i>History</i> . 2. Penguji memilih satu tahun kejadian yang tersedia di sistem. |
| Hasil yang Diharapkan | Sistem menampilkan hasil pemetaan daerah yang terkena erupsi dalam bentuk peta dan menampilkan informasi dari peta, sesuai dengan tahun kejadian yang dipilih oleh <i>user</i> . |

4. Test Case Memilih Basemap

Kasus uji validasi untuk *use case* memilih *basemap* ditunjukkan pada Tabel 6.4 berikut.

Tabel 6.4 Test case memilih basemap

| | |
|------------------------------|--|
| Nomor Kasus Uji | TEST-04 |
| Nama Kasus Uji | Memilih <i>basemap</i> |
| Kode Use Case | MIS-1-004 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan hasil pemetaan dengan <i>basemap</i> yang dipilih oleh pengguna. |
| Test Case | Pengujian pada menu <i>Basemap</i> . |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Basemap</i>. 2. Penguji memilih satu <i>basemap</i> yang tersedia di sistem. |
| Hasil yang Diharapkan | Sistem menampilkan hasil pemetaan daerah yang terkena erupsi dengan peta dasar/ <i>basemap</i> yang telah dipilih. |

5. Test Case Mencetak Peta

Kasus uji validasi untuk *use case* mencetak peta dapat dilihat pada Tabel 6.5 berikut.

Tabel 6.5 Test case mencetak peta

| | |
|------------------------------|--|
| Nomor Kasus Uji | TEST-05 |
| Nama Kasus Uji | Mencetak peta |
| Kode Use Case | MIS-1-005 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat mencetak hasil pemetaan daerah yang terdampak erupsi dan informasi yang ada di sistem. |
| Test Case | Pengujian dengan menekan <i>icon Print</i> yang ada pada halaman <i>History</i> . |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>History</i>. 2. Penguji menekan tombol <i>Print</i>. |
| Hasil yang Diharapkan | Sistem akan menyimpan peta hasil pemetaan daerah yang terdampak erupsi dan informasi peta dalam format PDF sehingga dapat diunduh. |

6. Test Case Melihat Diagram

Kasus uji validasi untuk *use case* mencetak peta ditunjukkan pada Tabel 6.6 berikut.

Tabel 6.6 Test case melihat diagram

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-06 |
| Nama Kasus Uji | Melihat diagram |
| Kode Use Case | MIS-1-006 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan diagram yang menunjukkan persentase luasan daerah yang terdampak erupsi untuk seluruh tahun kejadian yang terdapat pada sistem. |
| Test Case | Pengujian menekan tombol <i>See Diagram</i> di halaman <i>History</i> . |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>History</i>. 2. Penguji menekan tombol <i>See Diagram</i>. |
| Hasil yang Diharapkan | Sistem menampilkan diagram yang menunjukkan persentase luasan daerah yang terdampak erupsi untuk seluruh tahun kejadian yang ada pada sistem. |

7. Test Case Login

Kasus uji validasi untuk *use case login* dengan masukan *username* dan *password* yang benar ditunjukkan pada Tabel 6.7 berikut.

Tabel 6.7 Test case login [benar]

| | |
|---------------------------|--|
| Nomor Kasus Uji | TEST-07 |
| Nama Kasus Uji | <i>Login</i> [benar] |
| Kode Use Case | MIS-1-007 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem bisa melakukan autentikasi pengguna saat proses <i>login</i> . |
| Test Case | Pengujian <i>login</i> dengan menginputkan <i>username</i> dan <i>password</i> yang telah terdaftar di <i>database</i> dan sistem. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Sign In</i>. 2. Penguji menginputkan <i>username</i> dengan nilai admin dan <i>password</i> dengan nilai admin, yang sudah terdaftar di sistem. 3. Penguji menekan tombol <i>Sign In</i>. |

Tabel 6.7 Test case login [benar] (lanjutan)

| | |
|-----------------------|---|
| Hasil yang Diharapkan | Sistem menampilkan halaman <i>Map Data</i> dengan status pengguna adalah admin. |
|-----------------------|---|

Sedangkan kasus uji untuk *use case login* dengan masukan *username* dan *password* yang salah ditunjukkan pada Tabel 6.8 berikut.

Tabel 6.8 Test case login [salah]

| | |
|-----------------------|--|
| Nomor Kasus Uji | TEST-08 |
| Nama Kasus Uji | <i>Login [salah]</i> |
| Kode Use Case | MIS-1-007 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem bisa melakukan autentikasi saat proses <i>login</i> . |
| Test Case | Pengujian <i>login</i> dengan menginputkan <i>username</i> dan <i>password</i> yang tidak terdaftar di sistem. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Sign In</i>. 2. Penguji menginputkan <i>username</i> dan <i>password</i> dengan nilai apapun yang belum terdaftar di <i>database</i>. Atau tidak mengisi kolom <i>username</i> dan <i>password</i>. 3. Penguji menekan tombol <i>Sign In</i>. |
| Hasil yang Diharapkan | Sistem menampilkan notifikasi “Login failed! <i>Please make sure your username and password are correct</i> ”, kemudian kembali menampilkan halaman <i>Sign In</i> . |

8. Test Case Logout

Kasus uji validasi untuk *use case logout* ditunjukkan pada Tabel 6.9 berikut.

Tabel 6.9 Test case logout

| | |
|-----------------------|---|
| Nomor Kasus Uji | TEST-09 |
| Nama Kasus Uji | <i>Logout</i> |
| Kode Use Case | MIS-1-008 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa admin dapat keluar dari dashboard admin ketika menekan <i>Sign Out</i> . |
| Test Case | Pengujian <i>Sign Out</i> |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji menekan tombol <i>Sign Out</i>. |
| Hasil yang Diharapkan | Sistem menampilkan halaman <i>History</i> . |

9. Test Case Melihat Data *History*

Kasus uji validasi untuk *use case* melihat data *history* ditunjukkan pada Tabel 6.10 berikut.

Tabel 6.10 Test case melihat data *history*

| | |
|------------------------------|--|
| Nomor Kasus Uji | TEST-10 |
| Nama Kasus Uji | Melihat data <i>history</i> |
| Kode Use Case | MIS-1-009 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan seluruh data JSON yang terdapat di sistem. |
| Test Case | Pengujian melihat data pada menu <i>Map Data</i> . |
| Prosedur Pengujian | 1. Penguji memilih menu <i>Map Data</i> . |
| Hasil yang Diharapkan | Sistem menampilkan seluruh data JSON yang pernah ditambahkan ke <i>database</i> di halaman <i>Map Data</i> . |

10. Test Case Menambah Data *History*

Kasus uji validasi untuk *use case* menambah data *history*, dimana semua kolom sudah terisi dengan benar dan sesuai format ditunjukkan pada Tabel 6.11 dibawah ini.

Tabel 6.11 Test case menambah data *history* [benar]

| | |
|---------------------------|---|
| Nomor Kasus Uji | TEST-11 |
| Nama Kasus Uji | Menambah data <i>history</i> [benar] |
| Kode Use Case | MIS-1-010 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan data JSON ke <i>database</i> dan sistem. |
| Test Case | Pengujian tambah data dengan inputan <i>form</i> data yang terisi lengkap dan sesuai format. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Map Data</i>. 2. Penguji menekan <i>icon</i> tambah data. 3. Penguji mengisi kolom <i>File</i> dengan <i>file</i> data json, kolom <i>Year of Eruption</i> dengan tahun kejadian dari data yang akan ditambahkan berupa angka 4 digit di atas tahun 1990, dan kolom <i>Description</i> dengan deskripsi dari data yang akan ditambahkan berupa teks. |

Tabel 6.11 *Test case* menambah data *history* [benar] (lanjutan)

| | |
|------------------------------|---|
| Prosedur Pengujian | 4. Penguji menekan tombol <i>Save</i> . |
| Hasil yang Diharapkan | Sistem akan menyimpan data JSON beserta keterangan dan tahun yang telah diinputkan ke <i>database</i> dan sistem, menampilkan notifikasi " <i>Successfully Add Data!</i> ", dan kemudian menampilkan data JSON yang baru ditambahkan pada halaman <i>Map Data</i> . |

Sedangkan kasus uji validasi untuk *use case* menambah data *history*, dimana terdapat kolom yang belum terisi atau tidak sesuai format ditunjukkan pada Tabel 6.12 berikut.

Tabel 6.12 *Test case* menambah data *history* [salah]

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-12 |
| Nama Kasus Uji | Menambah data <i>history</i> [salah] |
| Kode Use Case | MIS-1-010 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan notifikasi kesalahan dan tidak menyimpan data ke <i>database</i> , apabila terdapat kolom yang tidak terisi atau tidak sesuai format. |
| Test Case | Pengujian tambah data dengan inputan data yang tidak terisi lengkap atau tidak sesuai format. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Map Data</i>. 2. Penguji menekan <i>icon</i> tambah data. 3. Penguji tidak mengisi kolom <i>File</i> namun mengisi kolom lainnya, atau mengisi semua kolom dengan benar kecuali pada kolom <i>Year of Eruption</i> diisi dengan huruf. 4. Penguji menekan tombol <i>Save</i>. |
| Hasil yang Diharapkan | Sistem menampilkan beberapa notifikasi, yaitu notifikasi " <i>Please fill out this field</i> " jika terdapat kolom yang tidak diisi. Dan menampilkan notifikasi " <i>Please enter a number</i> ", jika kolom <i>Year of Eruption</i> diisi dengan nilai selain angka. |

11. *Test Case* Menghapus Data *History*

Kasus uji validasi untuk *use case* menghapus seluruh data *history* dapat dilihat pada Tabel 6.13 berikut.

Tabel 6.13 *Test case* menghapus data *history* [semua]

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-13 |
| Nama Kasus Uji | Menghapus data <i>history</i> [semua] |
| Kode Use Case | MIS-1-011 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus seluruh data JSON yang terdapat di sistem sekaligus menghapus seluruh informasi peta. |
| Test Case | Pengujian menghapus seluruh data JSON dengan menekan <i>icon</i> hapus di halaman <i>Map Data</i> . |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Map Data</i> pada halaman admin. 2. Penguji menekan <i>icon</i> hapus. 3. Penguji menekan tombol "OK" yang terdapat pada <i>popup</i>. |
| Hasil yang Diharapkan | Sistem akan menghapus seluruh data JSON dari sistem dan <i>database</i> , dan tidak akan menampilkan data yang telah dihapus pada halaman <i>Map Data</i> . |

Sedangkan kasus uji validasi untuk *use case* menghapus data *history* pada baris tertentu ditunjukkan pada Tabel 6.14 berikut.

Tabel 6.14 *Test case* menghapus data *history* [baris tertentu]

| | |
|---------------------------|--|
| Nomor Kasus Uji | TEST-14 |
| Nama Kasus Uji | Menghapus data <i>history</i> [baris tertentu] |
| Kode Use Case | MIS-1-011 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus data JSON tertentu sekaligus informasi dari peta yang dipilih. |
| Test Case | Pengujian menghapus data JSON dengan menekan tombol <i>Delete</i> pada salah satu data di halaman <i>Map Data</i> . |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Map Data</i>. 2. Penguji menekan tombol <i>Delete</i> pada salah satu baris data. 3. Penguji menekan tombol "OK" yang terdapat pada <i>popup</i>. |

Tabel 6.14 *Test case* menghapus data *history* [baris tertentu] (lanjutan)

| | |
|------------------------------|--|
| Hasil yang Diharapkan | Sistem akan menghapus data JSON dan informasi peta dari <i>database</i> sesuai tahun kejadian yang dipilih, dan tidak akan menampilkan data yang telah dihapus pada halaman <i>Map Data</i> dan <i>Info Data</i> . |
|------------------------------|--|

12. Test Case Melihat Data Info History

Kasus uji validasi untuk *use case* melihat data info *history* dapat dilihat pada Tabel 6.15 berikut.

Tabel 6.15 *Test case* melihat data info *history*

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-15 |
| Nama Kasus Uji | Melihat data info <i>history</i> |
| Kode Use Case | MIS-1-012 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan seluruh data informasi peta yang telah ditambahkan ke sistem. |
| Test Case | Pengujian melihat data informasi pada menu <i>Info Data</i> . |
| Prosedur Pengujian | 1. Penguji memilih menu <i>Info Data</i> pada halaman admin. |
| Hasil yang Diharapkan | Sistem menampilkan semua informasi yang terdapat di <i>database</i> pada halaman <i>Info Data</i> . |

13. Test Case Menambah Data Info History

Kasus uji validasi untuk *use case* menambah data info *history*, dimana semua kolom sudah terisi dengan benar dan sesuai format ditunjukkan pada Tabel 6.16 dibawah ini.

Tabel 6.16 *Test case* menambah data info *history* [benar]

| | |
|-------------------------|--|
| Nomor Kasus Uji | TEST-16 |
| Nama Kasus Uji | Menambah data info <i>history</i> [benar] |
| Kode Use Case | MIS-1-013 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan informasi ke <i>database</i> dan informasi tersebut akan ditampilkan pada halaman <i>Info Data</i> . |
| Test Case | Pengujian tambah data dengan inputan data informasi yang terisi lengkap dan sesuai format. |

Tabel 6.16 *Test case* menambah data info *history* [benar] (lanjutan)

| | |
|------------------------------|--|
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Info Data</i>. 2. Penguji menekan <i>icon</i> tambah data. 3. Penguji mengisi kolom <i>Year of Eruption</i> dengan tahun kejadian dari data yang akan ditambahkan berupa angka 4 digit di atas tahun 1990, kolom ID Data dengan ID dari data asli berupa teks atau angka, kolom <i>Acquisition Date</i> dengan tanggal sesuai urutan (mm/dd/yyyy), kolom <i>Large</i> dengan nilai berupa angka, dan kolom <i>Coverage of Eruption</i> dengan daerah yang terdampak erupsi pada tahun kejadian itu dalam bentuk teks. 4. Penguji menekan tombol <i>Save</i>. |
| Hasil yang Diharapkan | Sistem akan menyimpan informasi ke basis data, menampilkan notifikasi " <i>Successfully Add Data!</i> ", dan menampilkan informasi tersebut pada halaman <i>Info Data</i> . |

Sedangkan kasus uji validasi untuk *use case* menambah data info *history*, dimana terdapat kolom yang belum terisi atau tidak sesuai format ditunjukkan pada Tabel 6.17 berikut.

Tabel 6.17 *Test case* menambah data info *history* [salah]

| | |
|---------------------------|--|
| Nomor Kasus Uji | TEST-17 |
| Nama Kasus Uji | Menambah data info <i>history</i> [salah] |
| Kode Use Case | MIS-1-013 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan notifikasi kesalahan dan tidak menyimpan data ke <i>database</i> , apabila terdapat kolom yang belum terisi atau tidak sesuai format. |
| Test Case | Pengujian tambah data dengan inputan data yang tidak terisi lengkap atau tidak sesuai format. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Info Data</i>. 2. Penguji menekan <i>icon</i> tambah data. 3. Penguji tidak mengisi salah satu kolom, atau mengisi semua kolom dengan benar kecuali pada kolom <i>Large</i> diisi dengan huruf. 4. Penguji menekan tombol <i>Save</i>. |

Tabel 6.17 *Test case* menambah data *history* [salah] (lanjutan)

| | |
|------------------------------|---|
| Hasil yang Diharapkan | Sistem menampilkan beberapa notifikasi, yaitu "Please fill out this field" jika terdapat kolom yang tidak diisi. Dan notifikasi "Please enter a number", jika kolom <i>Large</i> diisi dengan nilai selain angka. |
|------------------------------|---|

14. *Test Case* Menghapus Data Info *History*

Kasus uji validasi untuk *use case* menghapus semua data info *history* dapat dilihat pada Tabel 6.18 berikut.

Tabel 6.18 *Test case* menghapus data info *history* [semua]

| | |
|------------------------------|--|
| Nomor Kasus Uji | TEST-18 |
| Nama Kasus Uji | Menghapus data info <i>history</i> [semua] |
| Kode Use Case | MIS-1-014 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus seluruh informasi yang terdapat pada sistem. |
| Test Case | Pengujian menghapus semua informasi dengan menekan <i>icon</i> hapus di halaman <i>Info Data</i> . |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Info Data</i> pada halaman admin. 2. Penguji menekan <i>icon</i> hapus. 3. Penguji menekan tombol "OK" yang terdapat pada <i>popup</i>. |
| Hasil yang Diharapkan | Sistem akan menghapus seluruh informasi dari basis data, dan tidak akan menampilkan informasi yang telah dihapus pada halaman <i>Info Data</i> . |

Sedangkan kasus uji validasi untuk *use case* menghapus data info *history* pada baris tertentu dapat dilihat pada Tabel 6.19 berikut.

Tabel 6.19 *Test case* menghapus data info *history* [baris tertentu]

| | |
|-------------------------|--|
| Nomor Kasus Uji | TEST-19 |
| Nama Kasus Uji | Menghapus data info <i>history</i> [baris tertentu] |
| Kode Use Case | MIS-1-014 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus informasi pada baris tertentu. |
| Test Case | Pengujian menghapus informasi dengan menekan tombol <i>Delete</i> pada baris informasi tertentu. |

Tabel 6.19 *Test case* menghapus data info *history* [baris tertentu]

| | |
|------------------------------|---|
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Info Data</i>. 2. Penguji menekan tombol <i>Delete</i> pada salah satu informasi . 3. Penguji menekan tombol “OK” yang terdapat pada <i>popup</i>. |
| Hasil yang Diharapkan | Sistem akan menghapus informasi dari basis data sesuai tahun kejadian yang dipilih, dan tidak akan menampilkan informasi yang telah dihapus pada halaman <i>Info Data</i> . |

15. Test Case Mengedit Data Info History

Kasus uji validasi untuk *use case* mengedit data info *history*, dimana semua kolom sudah terisi dengan benar dan sesuai format ditunjukkan pada Tabel 6.20 dibawah ini.

Tabel 6.20 *Test case* mengedit data info *history* [benar]

| | |
|---------------------------|---|
| Nomor Kasus Uji | TEST-20 |
| Nama Kasus Uji | Mengedit data info <i>history</i> [benar] |
| Kode Use Case | MIS-1-015 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui informasi peta yang terdapat di sistem. |
| Test Case | Pengujian <i>edit</i> data dengan inputan informasi baru yang terisi lengkap dan sesuai format. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu <i>Info Data</i>. 2. Penguji menekan tanda <i>Edit</i> pada salah satu baris data. 3. Penguji mengubah isi seluruh kolom dan mengisi kolom sesuai dengan format yang ditentukan, yaitu mengisi kolom <i>Year of Eruption</i> dengan tahun diatas 1990 dalam 4 digit, mengisi kolom ID Data dengan teks, mengisi kolom <i>Acquisition Date</i> dengan tanggal sesuai urutan (mm/dd/yyyy), mengisi kolom <i>Large</i> dengan nilai berupa angka, dan mengisi kolom <i>Coverage of Eruption</i> dengan daerah yang terdampak erupsi pada tahun kejadian itu dalam bentuk teks. 4. Penguji menekan tombol <i>Save</i>. |

Tabel 6.20 Test case mengedit data info *history* [benar] (lanjutan)

| | |
|------------------------------|--|
| Hasil yang Diharapkan | Sistem akan menyimpan informasi baru ke basis data, menampilkan notifikasi “ <i>Successfully Update Data!</i> ” dan menampilkan informasi tersebut di halaman <i>Info Data</i> . |
|------------------------------|--|

Sedangkan kasus uji validasi untuk *use case* mengedit data info *history*, dimana ada kolom yang belum terisi atau tidak sesuai format ditunjukkan pada Tabel 6.21 berikut.

Tabel 6.21 Test case mengedit data info *history* [salah]

| | |
|------------------------------|---|
| Nomor Kasus Uji | TEST-21 |
| Nama Kasus Uji | Mengedit data info <i>history</i> [salah] |
| Kode Use Case | MIS-1-015 |
| Tujuan Pengujian | Pengujian untuk memastikan bahwa sistem bisa menampilkan notifikasi kesalahan dan tidak menyimpan data ke <i>database</i> , apabila terdapat kolom yang tidak terisi atau tidak sesuai format. |
| Test Case | Pengujian <i>edit</i> data dengan inputan informasi baru yang tidak terisi lengkap dan tidak sesuai format. |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Penguji memilih menu Info Data. 2. Penguji menekan tombol <i>Edit</i> pada salah satu informasi. 3. Penguji mengubah isi dari kolom <i>Large</i> dengan huruf. 4. Penguji menekan tombol <i>Save</i>. |
| Hasil yang Diharapkan | Sistem akan merespon dengan menampilkan beberapa notifikasi, yaitu notifikasi “Please fill out this field” jika terdapat kolom yang tidak diisi. Dan menampilkan notifikasi “Please enter a number”, jika kolom <i>Large</i> diisi dengan nilai selain angka. |

6.1.3 Hasil Pengujian

Setelah dilakukan pengujian, berikut pada tabel 6.22 merupakan perbandingan dari hasil yang diharapkan dengan hasil pengujian di atas. Apabila hasil yang diharapkan sama dengan hasil yang diperoleh dari pengujian, maka akan memiliki status valid yang berarti bahwa kebutuhan yang didefinisikan telah terpenuhi pada sistem.

Tabel 6.22 Hasil pengujian *black-box* (uji validasi)

| Nomor Kasus Uji | Hasil yang Diharapkan | Hasil yang Diperoleh | Status |
|-----------------|--|--|--------|
| TEST-01 | Sistem menampilkan seluruh hasil pemetaan pada halaman <i>History</i> dalam bentuk peta. | Sistem menampilkan seluruh hasil pemetaan pada halaman <i>History</i> dalam bentuk peta. | Valid |
| TEST-02 | Sistem menampilkan informasi dari seluruh peta hasil pemetaan. | Sistem menampilkan informasi dari seluruh peta hasil pemetaan. | Valid |
| TEST-03 | Sistem menampilkan hasil pemetaan daerah yang terkena erupsi dalam bentuk peta dan menampilkan informasi dari peta, sesuai dengan tahun kejadian yang dipilih oleh <i>user</i> . | Sistem menampilkan hasil pemetaan daerah yang terkena erupsi dalam bentuk peta dan menampilkan informasi dari peta, sesuai dengan tahun kejadian yang dipilih oleh <i>user</i> . | Valid |
| TEST-04 | Sistem menampilkan hasil pemetaan daerah yang terkena erupsi dengan peta dasar/ <i>basemap</i> yang telah dipilih. | Sistem menampilkan hasil pemetaan daerah yang terkena erupsi dengan peta dasar/ <i>basemap</i> yang telah dipilih. | Valid |
| TEST-05 | Sistem akan menyimpan peta hasil pemetaan daerah yang terdampak erupsi dan informasi peta dalam format PDF sehingga dapat diunduh. | Sistem akan menyimpan peta hasil pemetaan daerah yang terdampak erupsi dan informasi peta dalam format PDF sehingga dapat diunduh. | Valid |
| TEST-06 | Sistem menampilkan diagram yang menunjukkan persentase luasan daerah yang terdampak erupsi untuk seluruh tahun kejadian yang ada pada sistem. | Sistem menampilkan diagram yang menunjukkan persentase luasan daerah yang terdampak erupsi untuk seluruh tahun kejadian yang ada pada sistem. | Valid |
| TEST-07 | Sistem menampilkan halaman <i>Map Data</i> dengan status pengguna adalah admin. | Sistem menampilkan halaman <i>Map Data</i> dengan status pengguna adalah admin. | Valid |

Tabel 6.22 Hasil pengujian *black-box* (uji validasi) (lanjutan)

| Nomor Kasus Uji | Hasil yang Diharapkan | Hasil yang Diperoleh | Status |
|-----------------|--|--|--------|
| TEST-08 | Sistem menampilkan notifikasi "Login failed! <i>Enter your username and password correctly</i> ", kemudian kembali menampilkan halaman <i>Sign In</i> . | Sistem menampilkan notifikasi "Login failed! <i>Enter your username and password correctly</i> ", kemudian kembali menampilkan halaman <i>Sign In</i> . | Valid |
| TEST-09 | Sistem menampilkan halaman <i>History</i> . | Sistem menampilkan halaman <i>History</i> . | Valid |
| TEST-10 | Sistem menampilkan seluruh data JSON yang pernah ditambahkan ke <i>database</i> di halaman <i>Map Data</i> . | Sistem menampilkan seluruh data JSON yang pernah ditambahkan ke <i>database</i> di halaman <i>Map Data</i> . | Valid |
| TEST-11 | Sistem akan menyimpan data JSON beserta keterangan dan tahun yang telah diinputkan ke <i>database</i> dan sistem, menampilkan notifikasi "Successfully Add Data!", dan kemudian menampilkan data JSON yang baru ditambahkan pada halaman <i>Map Data</i> . | Sistem akan menyimpan data JSON beserta keterangan dan tahun yang telah diinputkan ke <i>database</i> dan sistem, menampilkan notifikasi "Successfully Add Data!", dan kemudian menampilkan data JSON yang baru ditambahkan pada halaman <i>Map Data</i> . | Valid |
| TEST-12 | Sistem menampilkan beberapa notifikasi, yaitu notifikasi "Please fill out this field" jika terdapat kolom yang tidak diisi. Dan menampilkan notifikasi "Please enter a number", jika kolom <i>Year of Eruption</i> diisi dengan nilai selain angka. | Sistem menampilkan beberapa notifikasi, yaitu notifikasi "Please fill out this field" jika terdapat kolom yang tidak diisi. Dan menampilkan notifikasi "Please enter a number", jika kolom <i>Year of Eruption</i> diisi dengan nilai selain angka. | Valid |
| TEST-13 | Sistem akan menghapus seluruh data JSON dari sistem dan <i>database</i> , dan tidak akan menampilkan data yang telah dihapus pada halaman <i>Map Data</i> . | Sistem akan menghapus seluruh data JSON dari sistem dan <i>database</i> , dan tidak akan menampilkan data yang telah dihapus pada halaman <i>Map Data</i> . | Valid |

Tabel 6.22 Hasil pengujian *Black-Box* (Uji Validasi) (lanjutan)

| Nomor Kasus Uji | Hasil yang Diharapkan | Hasil yang Diperoleh | Status |
|-----------------|---|---|--------|
| TEST-14 | Sistem akan menghapus data JSON dan informasi peta dari <i>database</i> sesuai tahun kejadian yang dipilih, dan tidak akan menampilkan data yang telah dihapus pada halaman <i>Map Data</i> dan <i>Info Data</i> . | Sistem akan menghapus data JSON dan informasi peta dari <i>database</i> sesuai tahun kejadian yang dipilih, dan tidak akan menampilkan data yang telah dihapus pada halaman <i>Map Data</i> dan <i>Info Data</i> . | Valid |
| TEST-15 | Sistem menampilkan semua informasi yang terdapat di <i>database</i> pada halaman <i>Info Data</i> . | Sistem menampilkan semua informasi yang terdapat di <i>database</i> pada halaman <i>Info Data</i> . | Valid |
| TEST-16 | Sistem akan menyimpan informasi ke basis data, menampilkan notifikasi " <i>Successfully Add Data!</i> ", dan menampilkan informasi tersebut pada halaman <i>Info Data</i> . | Sistem akan menyimpan informasi ke basis data, menampilkan notifikasi " <i>Successfully Add Data!</i> ", dan menampilkan informasi tersebut pada halaman <i>Info Data</i> . | Valid |
| TEST-17 | Sistem menampilkan beberapa notifikasi, yaitu " <i>Please fill out this field</i> " jika terdapat kolom yang tidak diisi. Dan notifikasi " <i>Please enter a number</i> ", jika kolom <i>Large</i> diisi dengan nilai selain angka. | Sistem menampilkan beberapa notifikasi, yaitu " <i>Please fill out this field</i> " jika terdapat kolom yang tidak diisi. Dan notifikasi " <i>Please enter a number</i> ", jika kolom <i>Large</i> diisi dengan nilai selain angka. | Valid |
| TEST-18 | Sistem akan menghapus seluruh informasi dari basis data, dan tidak akan menampilkan informasi yang telah dihapus pada halaman <i>Info Data</i> . | Sistem akan menghapus seluruh informasi dari basis data, dan tidak akan menampilkan informasi yang telah dihapus pada halaman <i>Info Data</i> . | Valid |

Tabel 6.22 Hasil pengujian *Black-Box* (Uji Validasi) (lanjutan)

| Nomor Kasus Uji | Hasil yang Diharapkan | Hasil yang Diperoleh | Status |
|-----------------|---|---|--------|
| TEST-19 | Sistem akan menghapus informasi dari basis data sesuai tahun kejadian yang dipilih, dan tidak akan menampilkan informasi yang telah dihapus pada halaman <i>Info Data</i> . | Sistem akan menghapus informasi dari basis data sesuai tahun kejadian yang dipilih, dan tidak akan menampilkan informasi yang telah dihapus pada halaman <i>Info Data</i> . | Valid |
| TEST-20 | Sistem akan menyimpan informasi baru ke basis data, menampilkan notifikasi " <i>Successfully Update Data!</i> " dan menampilkan informasi tersebut di halaman <i>Info Data</i> . | Sistem akan menyimpan informasi baru ke basis data, menampilkan notifikasi " <i>Successfully Update Data!</i> " dan menampilkan informasi tersebut di halaman <i>Info Data</i> . | Valid |
| TEST-21 | Sistem akan merespon dengan menampilkan beberapa notifikasi, yaitu notifikasi " <i>Please fill out this field</i> " jika terdapat kolom yang tidak diisi. Dan menampilkan notifikasi " <i>Please enter a number</i> ", jika kolom <i>Large</i> diisi dengan nilai selain angka. | Sistem akan merespon dengan menampilkan beberapa notifikasi, yaitu notifikasi " <i>Please fill out this field</i> " jika terdapat kolom yang tidak diisi. Dan menampilkan notifikasi " <i>Please enter a number</i> ", jika kolom <i>Large</i> diisi dengan nilai selain angka. | Valid |

6.1.4 Analisis Pengujian

Berdasarkan hasil pengujian validasi yang telah dilakukan diatas, dapat dilihat bahwa semua kasus uji memiliki status valid. Sehingga dapat disimpulkan bahwa sistem telah memenuhi seluruh kebutuhan fungsional yang didefinisikan.

6.2 Pengujian *Portability*

6.2.1 Tujuan Pengujian

Pengujian *portability* bertujuan untuk memastikan bahwa kebutuhan non-fungsional yaitu *portability*.

6.2.2 Mekanisme Pengujian

Pada pengujian ini, sistem yang dibuat akan diuji pada 12 *browser*, yang terdiri dari tujuh *desktop browser* (IE v11, Edge v15, Firefox v56, Safari v11.0,

Opera v46, Google Chrome v60, dan Google Chrome v61) dan *lima mobile* browser (iOS v9.0, iOS v10.0, iOS v11.0, Android v4.0, dan BlackBerry v10.0). Pengujian dilakukan dengan menjalankan setiap halaman yang terdapat pada sistem pada ke-12 *browser* tersebut. Untuk mempermudah dalam tahap pengujian, akan dibantu dengan menggunakan aplikasi SortSite. Setiap halaman dari sistem akan diuji menggunakan aplikasi ini, dan dianggap valid apabila tidak terdapat *error* ataupun *warning*. Sistem dianggap memiliki *portability* yang tinggi jika dari keseluruhan halaman yang diuji bernilai valid >80%.

Terdapat beberapa parameter dari aplikasi *SortSite* yang digunakan pada pengujian, yaitu :

1. *Critical Issues* digunakan untuk mengecek fungsionalitas dan kehilangan dari konten sistem yang diuji.
2. *Major Issues* digunakan untuk mengecek *layout* utama, seperti letak menu atau navigasi, ukuran gambar, dan bentuk tabel dari sistem yang diuji.
3. *Minor Issues* digunakan untuk mengecek *property* yang terdapat pada *layout* di halaman, misalnya fungsi *required* pada input dan properti css tampilan.

Berikut pada Tabel 6.23 merupakan kasus uji untuk pengujian *portability*, dimana terdapat nama halaman dan URL dari setiap halaman yang diuji.

Tabel 6.23 Kasus uji pengujian *portability*

| Nomor Kasus Uji | Nama Halaman | URL Halaman |
|-----------------|------------------|---|
| PT-01 | <i>History</i> | http://localhost/webgis/Control_general/index |
| PT-02 | Diagram | http://localhost/webgis/Control_general/diagram |
| PT-03 | <i>Basemap</i> | http://localhost/webgis/Control_general/view_base map |
| PT-04 | <i>Sign In</i> | http://localhost/webgis/Control_general/view_login |
| PT-05 | <i>Map Data</i> | http://localhost/webgis/Control_admin/view_data |
| PT-06 | Form Tambah Data | http://localhost/webgis/Control_admin/form_data |
| PT-07 | <i>Info Data</i> | http://localhost/webgis/Control_admin/view_info |
| PT-08 | Form Tambah Info | http://localhost/webgis/Control_admin/ form_info |
| PT-09 | <i>Edit Info</i> | http://localhost/webgis/Control_admin/edit_info |

6.2.3 Hasil Pengujian

Berdasarkan pengujian dengan menggunakan aplikasi SortSite, didapatkan hasil pengujian untuk setiap halaman sebagai berikut.

1. Kasus Uji Halaman *History*

Gambar 6.1 merupakan hasil pengujian dari aplikasi SortSite yang menunjukkan bahwa pada halaman *History* tidak menunjukkan adanya eror, sehingga halaman *History* memiliki *portability* terhadap seluruh *browser* yang telah ditentukan.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|---------|---------|---------|-----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.1 Hasil pengujian *Portability* pada halaman *History*

2. Kasus Uji Halaman Diagram

Pada Gambar 6.2 merupakan hasil pengujian dari halaman Diagram menggunakan aplikasi SortSite. Dari aplikasi tersebut, tidak terdapat eror sama sekali pada halaman Diagram. Hasil pengujian tersebut menunjukkan bahwa halaman tersebut memiliki *portability* terhadap seluruh *browser* yang telah ditentukan.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|---------|---------|---------|-----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.2 Hasil pengujian *Portability* pada halaman diagram

3. Kasus Uji Halaman *Basemap*

Pada Gambar 6.3 merupakan hasil dari pengujian dari halaman *Basemap*, yang menunjukkan bahwa halaman ini memiliki *portability* untuk dijalankan pada seluruh *browser* yang telah ditentukan. Hal tersebut diketahui dengan tidak adanya error pada hasil pengujian.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|---------|---------|---------|-----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.3 Hasil pengujian *Portability* pada halaman *basemap*



4. Kasus Uji Halaman *Sign In*

Hasil dari pengujian pada halaman *Sign In* ditunjukkan pada Gambar 6.4, yang menunjukkan bahwa halaman *Sign In* tidak *portable* untuk dijalankan pada Firefox v56, dan pada Google Chrome v60 dan v61. Error tersebut terjadi karena pada Firefox dan Google Chrome menampilkan beberapa peringatan keamanan untuk kata sandi yang tidak menggunakan HTTPS.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|--------|--------|--------|----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤10 11 | ≤46 47 | ≤60 61 | ≤9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ● | ✓ | ✓ | ● | ● | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Priority 1
1 issues on 1 pages.
● Chrome and Firefox display multiple security warnings for password fields that don't use HTTPS. [Chrome ≥ 58 Firefox ≥ 52](#) 1 pages

Gambar 6.4 Hasil pengujian *Portability* pada halaman *sign in*

5. Kasus Uji Halaman *Map Data*

Gambar 6.5 merupakan hasil pengujian pada halaman *Map Data*, yang menunjukkan tidak adanya error ketika diuji pada seluruh *browser* yang telah ditentukan. Berdasarkan hasil pengujian tersebut, halaman *Map Data* memiliki *portability* terhadap semua *browser* yang diuji.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|--------|--------|--------|----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤10 11 | ≤46 47 | ≤60 61 | ≤9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.5 Hasil pengujian *Portability* pada halaman *map data*

6. Kasus Uji Halaman *Form Tambah Data*

Pada Gambar 6.6 merupakan hasil pengujian pada halaman *form* tambah data yang memperlihatkan *portability* terhadap seluruh *browser* yang telah ditentukan, karena tidak menunjukkan adanya eror apapun.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|---------|---------|---------|-----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.6 Hasil pengujian *Portability* pada halaman *form* tambah data

7. Kasus Uji Halaman *Info Data*

Gambar 6.7 memperlihatkan hasil pengujian untuk halaman *Info Data*, dimana tidak terdapat error ketika halaman tersebut dijalankan pada seluruh *browser* yang ditentukan. Berdasarkan hasil pengujian tersebut, halaman *Info Data* memiliki *portability* terhadap seluruh *browser* yang ditentukan.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|---------|---------|---------|-----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.7 Hasil pengujian *Portability* pada halaman *info data*

8. Kasus Uji Halaman *Form Tambah Info*

Gambar 6.8 merupakan hasil pengujian untuk halaman *form* tambah info, yang menunjukkan bahwa halaman ini memiliki *portability* terhadap seluruh *browser* yang ditentukan. Hal tersebut ditunjukkan dengan tidak adanya *error* pada hasil pengujian.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|-------|------|---------|---------|---------|---------|-----------|---------|------------|
| Version | 10 11 | 15 | 56 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.8 Hasil pengujian *Portability* pada halaman *form* tambah info

9. Kasus Uji Halaman *Edit Info*

Pengujian pada halaman *edit info* ditunjukkan pada Gambar 6.9 dengan hasil tidak terdapat error apapun untuk seluruh *browser* yang diuji. Berdasarkan hasil pengujian tersebut, halaman ini memiliki *portability* terhadap seluruh *browser* yang telah ditentukan untuk pengujian.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | Android | BlackBerry |
|-----------------|----------|------|---------|---------|---------|-----------|-----|---------|------------|
| Version | 10 11 15 | 55 | ≤ 10 11 | ≤ 46 47 | ≤ 60 61 | ≤ 9 10 11 | 4* | 10.0 | |
| Critical Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Major Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Minor Issues | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.9 Hasil pengujian *Portability* pada halaman *edit info*

Berikut pada Tabel 6.24 merupakan ringkasan hasil secara keseluruhan untuk seluruh halaman yang diuji.

Tabel 6.24 Ringkasan hasil pengujian *portability*

| Nomor Kasus Uji | Status |
|-----------------|-------------|
| PT-01 | Valid |
| PT-02 | Valid |
| PT-03 | Valid |
| PT-04 | Tidak Valid |
| PT-05 | Valid |
| PT-06 | Valid |
| PT-07 | Valid |
| PT-08 | Valid |
| PT-09 | Valid |

6.2.4 Analisis Pengujian

Berdasarkan ringkasan hasil pengujian diatas, dari keseluruhan kasus uji, hanya pada halaman *Sign In* yang menunjukkan adanya *error*. Sehingga dapat disimpulkan bahwa sistem yang diuji masih memiliki kompatibilitas yang tinggi dengan nilai 88,9 %, karena hanya 1 kasus uji dari 9 kasus uji yang memiliki eror. Dikarenakan hasil pengujian memiliki nilai valid >80%, maka sistem yang dibuat dapat dikatakan telah memenuhi kebutuhan non-fungsional untuk parameter *Portability*.

6.3 Pengujian *Accuracy*

6.3.1 Tujuan Pengujian

Pengujian *accuracy* dilakukan untuk mengetahui tingkat akurasi dan presisi dari hasil pemetaan terhadap keadaan daerah yang sebenarnya.

6.3.2 Mekanisme Pengujian

Pada pengujian ini, akan dilakukan pengujian terhadap hasil pemetaan pada tahun 2010, karena data tersebut merupakan kejadian terbaru dari erupsi Gunung Merapi. Untuk melakukan pengujian ini, dibutuhkan data pembanding yaitu berupa titik-titik koordinat beserta nilai dari setiap lokasi. Nilai tersebut adalah apakah daerah yang diambil titik koordinat terkena erupsi atau tidak. Data pembanding tersebut merupakan data yang akan dimasukkan dalam *True Class*, sedangkan hasil dari pemetaan yang dilakukan dimasukkan dalam *Hypothesized Class*. Data pembanding yang dibutuhkan didapatkan dari survei langsung ke lokasi-lokasi daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman, dengan menggunakan aplikasi kompas pada *smarthphone* yang memiliki resolusi 15x15 m, sebagai alat pengambil koordinat. Dari hasil survei, didapatkan 50 data yang akan digunakan sebagai data pembanding yang dapat dilihat pada Lampiran.

6.3.3 Hasil Pengujian

Hasil dari pengujian *accuracy* dengan menggunakan ROC untuk daerah yang terdampak erupsi pada tahun 2010 dapat dilihat pada Tabel 6.25. Berikut merupakan hasil dari pengujian yang menunjukkan hasil bahwa yang diduga positif benar sejumlah 25 titik, yang diduga positif salah sejumlah 4 titik, yang diduga negatif salah adalah tidak ada, dan yang diduga negatif benar sejumlah 21 titik.

Sedangkan nilai presisi positif yang didapatkan adalah 86,2%, nilai presisi negatif yang didapatkan adalah 100%, nilai sensitifitas benar yang didapatkan adalah 100%, nilai sensitifitas salah yang didapatkan adalah 84% dan nilai akurasi adalah 92%. Detail perhitungan hasil pengujian akurasi dan data pembanding yang digunakan untuk pengujian dapat dilihat pada Lampiran B dan Lampiran C.

Tabel 6.25 Hasil pengujian daerah terdampak erupsi tahun 2010

| Akurasi = 92% | | True class | | Class Precision |
|----------------|----------------------------|----------------------|----------------------------|-----------------|
| | | p (terdampak erupsi) | n (bukan terdampak erupsi) | |
| Prediksi Hasil | Y (terdampak erupsi) | 25 | 4 | 86,2% |
| | N (bukan terdampak erupsi) | 0 | 21 | 100% |
| Class Recall | | 100% | 84% | |

6.3.4 Analisis Pengujian

Berdasarkan hasil pengujian *accuracy* dengan menggunakan ROC, dapat disimpulkan bahwa dari hasil pengujian didapatkan nilai akurasi 92% dan nilai presisi 86,2%. Hal ini berarti bahwa hasil pemetaan memiliki tingkat akurasi dan presisi yang tinggi dengan keadaan yang sebenarnya.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pada seluruh proses penelitian yang telah dilakukan, maka dapat disimpulkan beberapa hasil sebagai berikut :

1. Dari hasil klasifikasi, didapatkan hasil akhir pemetaan daerah yang terdampak erupsi untuk setiap tahun kejadian adalah sebagai berikut :
 - a. Tahun 1992 menghasilkan luasan sebesar 281.94 ha dengan persentase 4,1 % dari seluruh tahun kejadian.
 - b. Tahun 1994 menghasilkan luasan sebesar 1058.61 ha dengan persentase 15,2 % dari seluruh tahun kejadian.
 - c. Tahun 1997 menghasilkan luasan sebesar 741.11 ha dengan persentase 10,7 % dari seluruh tahun kejadian.
 - d. Tahun 1998 menghasilkan luasan sebesar 278.6 ha dengan persentase 4% dari seluruh tahun kejadian.
 - e. Tahun 2001 menghasilkan luasan sebesar 499.97 ha dengan persentase 7,2 % dari seluruh tahun kejadian.
 - f. Tahun 2006 menghasilkan luasan sebesar 783.94 ha dengan persentase 11,3 % dari seluruh tahun kejadian.
 - g. Tahun 2010 menghasilkan luasan sebesar 3309.55 ha dengan persentase 47,6 % dari seluruh tahun kejadian.
2. Arsitektur WebGIS yang dibuat terdiri dari 3 *layer*, yaitu *Client Layer*, *Service Layer*, dan *Support Layer*.
 - a. *Client Layer* digunakan untuk penghubung antara *user* dan sistem melalui *web browser*. Dalam menampilkan antarmuka, didukung oleh beberapa komponen yaitu *Bootstrap*, *HTML*, *CSS*, *JS*, *jQuery*, *Leaflet*, *Highcharts*, dan *OpenStreetMap*.
 - b. *Service Layer* digunakan sebagai penyedia layanan jaringan telekomunikasi. Pada *service layer* menggunakan *Apache* sebagai *web server*, *framework CodeIgniter*, dan bahasa pemrograman *PHP*. Dalam penerapan *framework CodeIgniter*, kinerja dari sistem menggunakan konsep *Model-View-Controller (MVC)*. *View* merupakan bagian untuk menampilkan antarmuka bagi *user*. *Controller* merupakan bagian yang berisi fungsi-fungsi untuk menghubungkan antara Model dan View. Sedangkan *Model* merupakan bagian yang berisi fungsi-fungsi untuk mengakses *database*.
 - c. *Support Layer* merupakan bagian yang berisi basis data yang digunakan untuk menyimpan data. Pada *Support Layer* terdapat *MySQL* yang berperan sebagai server penyimpanan dan penyedia data. Di dalam

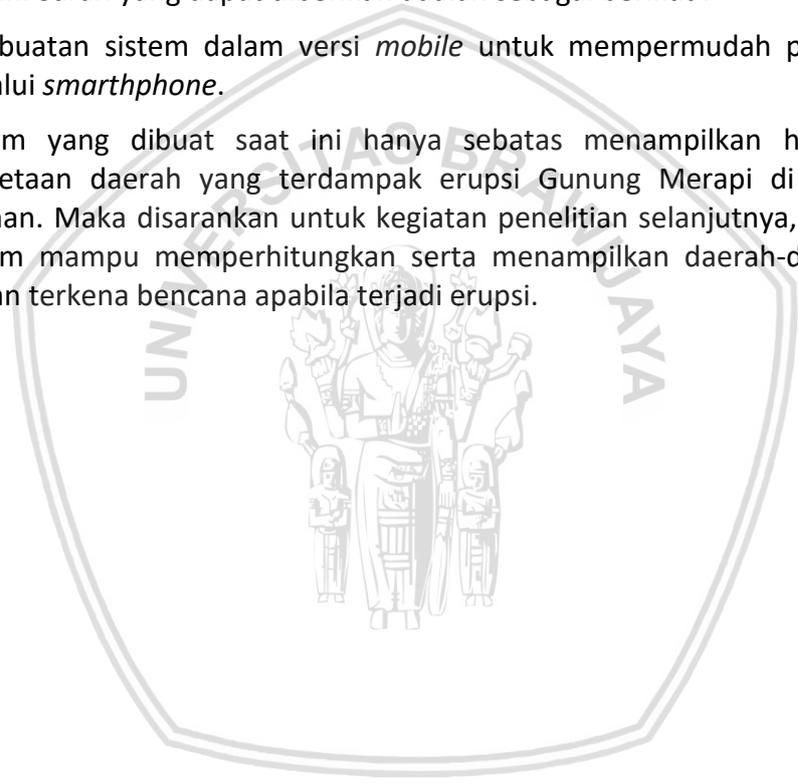
MySQL terdapat sebuah *database* dan table-tabel yang digunakan untuk menyimpan data yang dibutuhkan dalam WebGIS. Tabel-tabel yang terdapat dalam *database* antara lain tabel data, tabel info, dan tabel *user*.

3. Dari hasil pengujian *accuracy* pada data hasil pemetaan daerah yang terkena erupsi menggunakan ROC, maka didapatkan hasil nilai akurasi sebesar 92% dan nilai presisi sebesar 86,2%. Hasil pengujian tersebut menunjukkan bahwa hasil pemetaan memiliki akurasi dan presisi yang tinggi.

7.2 Saran

Saran diberikan sebagai referensi dalam pengembangan sistem selanjutnya, apabila ada yang berminat untuk mengembangkan sistem ini dalam sebuah penelitian. Saran yang dapat diberikan adalah sebagai berikut :

1. Pembuatan sistem dalam versi *mobile* untuk mempermudah pengaksesan melalui *smarthphone*.
2. Sistem yang dibuat saat ini hanya sebatas menampilkan history hasil pemetaan daerah yang terdampak erupsi Gunung Merapi di Kabupaten Sleman. Maka disarankan untuk kegiatan penelitian selanjutnya, diharapkan sistem mampu memperhitungkan serta menampilkan daerah-daerah yang rawan terkena bencana apabila terjadi erupsi.



DAFTAR PUSTAKA

- Alesheikh, A.A., Helali, H., Behroz, H.A., 2002. *Web GIS: Technologies and Its Applications*. Department of Geodesy and Geomatics Engineering K.N. Toosi University of Technology, Tehran, Iran.
- Al Fatta, Hanif, 2009. Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern. Yogyakarta: Penerbit Andi
- Ambler, Scott W., 2004. *UML 2 Sequence Diagramming Guidelines. Agile Model Driven Development with UML 2*. [online] Ambyssoft Inc. Tersedia di: <http://agilemodeling.com/style/sequenceDiagram.htm> [Diakses 5 April 2017]
- A.S., R. & Shalahuddin, M., 2014. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Sleman, 2017. Dampak Gunung Merapi di Kabupaten Sleman [Wawancara] [Juni 2017]
- Badan Penelitian dan Pengembangan Pertanian, 2011. Laporan Hasil Kajian Singkat (Quick Assessment) Dampak Erupsi Gunung Merapi di Sektor Pertanian. Jakarta.
- Badan Pusat Statistik (BPS), 2015. Statistik Daerah Istimewa Yogyakarta 2015. [online] Tersedia di: [https:// http://yogyakarta.bps.go.id/website/pdf_publicasi/Daerah-Istimewa-Yogyakarta-Dalam-Angka-2015.pdf](https://http://yogyakarta.bps.go.id/website/pdf_publicasi/Daerah-Istimewa-Yogyakarta-Dalam-Angka-2015.pdf) [Diakses 10 Oktober 2016]
- Balai Penyelidikan dan Pengembangan Teknologi Kegunungapian (BPPTK), 2000. *Karakteristik Gunung Api*. Yogyakarta.
- Bell, D., 2003. *UML Basics: An Introduction to the Unified Modelling Language. Rational Software*. [online] Tersedia di: <https://www.ibm.com/developerworks/rational/library/769.html> [Diakses 5 April 2017]
- Bemmelen, van, R.W., 1949. *The Geology of Indonesia Vol. IA. General Geology of Indonesia and Adjacent Archipelagoes*. The Haque, Nederland.
- Daqiqil, I., 2011. *Framework Codelgniter Sebuah Panduan dan Best Practice*. Pekanbaru: s.n.
- Ekadinata, A., Dewi, S., Hadi, D.P., Nugroho, D.K., dan Johana, F., 2008. *Sistem Informasi Geografis dan Penginderaan Jauh Menggunakan ILWIS Open Source*. Bogor: World Agroforestry Centre ICRAF South East Asia Regional Office.
- Fago, P., Pignatelli, C., Piscitelli, A., Milella, M., Venerito, M., Sanso, P., Mastronuzzi, G., 2014. *WebGIS for Italian Tsunami: A Useful Tool For*

- Coastal Planners*. [online] Tersedia di: www.sciencedirect.com [Diakses 10 Mei 2017]
- Fawcett, Tom, 2005. *An Introduction to ROC analysis*. Elsevier on Pattern Recognition.
- Fu, P., Sun, J., 2011. *Web Gis principles and applications*. [online] ESRI Press. Tersedia di: <https://www.gislounge.com/difference-web-gis-internet-gis/> [Diakses 29 September 2016]
- Hakim, L., 2010, *Membangun Web Berbasis PHP dengan Framework CodeIgniter*. Yogyakarta: Lokomedia.
- Helali, H. 2001. *Design and Implementation of a Web GIS for the City of Tehran*. MSc Thesis, Department of Geodesy And Geomatics Engineering K.N. Toosi University of Technology, Tehran, Iran.
- Huang, J., R, Huang., N, Ju., Q, Xu., C, He. 2015. *3D WebGIS-based platform for debris flow early warning: A case study*. Tersedia di: www.elsevier.com/locate/compag [Diakses 02 November 2016]
- Indarto, 2013. *Sistem Informasi Geografis*. Yogyakarta: Graha Ilmu
- John E. Harmon, Steven J. Anderson. 2003. *Design and Implementation of Geographic Information Systems*. John Wiley and Sons: New Jersey.
- Katili, J., A., dan Siswiwidjojo, S., S., 1994. *Pemantauan gunung api di Filipina dan Indonesia*. IA G1, Bandung.
- Leaflet, 2015. [online] Tersedia di: <http://leafletjs.com/index.html> [Diakses 9 Oktober 2016]
- Maclean, M., 2012. *Leaflet Tips and Tricks – Interactive Maps Made Easy*. [e-book] Tersedia di: Leanpub <https://leanpub.com/leaflet-tips-and-tricks/read/> [Diakses 9 Oktober 2016]
- Maliene, V., Grigonis, V., Palevicius, V. & Griffiths, S., 2011. *Geographic information sistem: Old principles with new capabilities*. *Urban Design International*. [online] Tersedia di: <http://link.springer.com/article/10.1057%2Fudi.2010.25> [Diakses 8 Oktober 2016]
- Marina-Ramona, R., Bogdan-Andrei, M., 2016. *Mapping Land Cover Using Remote Sensing Data and GIS Techniques: A Case Study of Prahova Subcarpathians*. [online] Tersedia di: www.sciencedirect.com [Diakses 10 Mei 2017]
- Nazir, M., 1988. *Metode Penelitian*. Jakarta: Ghalia Indonesia.
- OSGeo, Universitas Gadjah Mada, 2011. [online] Tersedia di: <http://osgeo.ft.ugm.ac.id/quantum-gis/> [Diakses 9 Oktober 2016]
- Painho, M., 2001. *WebGIS as a Teaching Tool*. San Diego, California.
- PowerMapper, 1996. *SortSite – Browser Compatibility Tests*. [online] Tersedia di: <http://www.powermapper.com/products/sortsite/checks/browser-compatibility/> [Diakses 16 Februari 2017]



- Prahasta, E., 2009. Sistem Informasi Geografis: konsep-konsep Dasar (Perpektif Geodesi & Geomatika). Bandung: Informasi.
- Pressman, R.S., 2010. *Software Engineering: A Practitioner'S Approach*. SEVENTH penyunt. Newyork: McGraw-Hill.
- QGIS, 2015. [online] Tersedia di: <http://www.qgis.org/id/site/about/index.html> [Diakses 9 Oktober 2016]
- QGIS, 2017. [online] Tersedia di: <https://plugins.qgis.org/plugins/SemiAutomaticClassificationPlugin/> [Diakses 4 Oktober 2016]
- Riebeek, H., 2013. *Historic Landsat 5 Mission Ends. NASA: Landsat Science*. [online] Tersedia di: <https://Landsat.gsfc.nasa.gov/historic-Landsat-5-mission-ends/> [Diakses 8 Februari 2017].
- Salonen, V., 2012. *Automatic Portability Testing*. Jyvaskyla: University of Jyvaskyla.
- Stefano, A., Paola, C., Rafaella, L., 2007. *A Face Recognition System Based on Local Feature Analysis, Face Recognition*. Book edited by: Kresimir Delac and Mislav Grgic, ISBN 978-3-902613-03-5.
- Sutikno, B., D., Sayuti dan G., Hartono, 1996. Variasi Luncuran Awan Panas Gunung Merapi dan Bahayanya. *Proceedings of the 25th Annual Convention of the Indonesian Association of Geologist: STTN dan Akademi IP Yogyakarta*.
- USGS, 2016. *Landsat 4 History. USGS: Science For A Changing World*. [online] Tersedia di: <https://Landsat.usgs.gov/Landsat-4-history> [Diakses 8 Februari 2017].
- USGS, 2016. *Landsat 7 History. USGS: Science For A Changing World*. [online] Tersedia di: <https://Landsat.usgs.gov/Landsat-7-history> [Diakses 8 Februari 2017].
- Voight, B., Sukhyar, R., Wirakusumah, A.D., 1979. Data Dasar Gunung Api Indonesia. Direktorat Vulkanologi.
- Widiyanto dan Rachman, A., 2008. Aspek Morfologi terhadap Bahaya Gunung Merapi. *Jurnal Kebencanaan Indonesia* vol. II No. 5, Nopember 2008.
- Wilson, T., Kaye, G., Stewart, C. and Cole, J., 2007. *Impacts of the 2006 eruption of Merapi volcano, Indonesia, on agriculture and infrastructure*. GNS Science Report 2007/07 69p.