



TUGAS AKHIR - IF184802

**RANCANG BANGUN APLIKASI BERBASIS WEB  
PENDETEKSI BORAKS PADA BAKSO DAN LOKASI  
PENJUALANNYA DI SURABAYA TIMUR  
MENGUNAKAN *ELECTRONIC NOSE***

DEWI SEKARINI  
NRP 05111640000004

Dosen Pembimbing  
Ir. Siti Rochimah, M.T., Ph.D  
Dwi Sunaryono, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020





TUGAS AKHIR - IF184802

**RANCANG BANGUN APLIKASI BERBASIS WEB  
PENDETEKSI BORAKS PADA BAKSO DAN  
LOKASI PENJUALANNYA DI SURABAYA TIMUR  
MENGUNAKAN *ELECTRONIC NOSE***

DEWI SEKARINI

NRP 05111640000004

Dosen Pembimbing

Ir. Siti Rochimah, M.T., Ph.D

Dwi Sunaryono, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESIS - IF184802**

**DESIGN AND IMPLEMENTATION OF WEB  
BASED APPLICATION AS BAKSO CONTAINING  
BORAX DETECTOR AND THE SALES LOCATION  
IN EAST SURABAYA WITH ELECTRONIC NOSE**

**DEWI SEKARINI  
NRP 05111640000004**

**Advisor  
Ir. Siti Rochimah, M.T., Ph.D  
Dwi Sunaryono, S.Kom., M.Kom.**

**INFORMATICS ENGINEERING DEPARTMENT  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### RANCANG BANGUN APLIKASI BERBASIS WEB PENDETEKSI BORAKS PADA BAKSO DAN LOKASI PENJUALANNYA DI SURABAYA TIMUR MENGUNAKAN *ELECTRONIC NOSE*

### TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer pada  
Rumpun Mata Kuliah Rekayasa Perangkat Lunak  
Program Studi S-1 Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh :

**DEWI SEKARINI**

NRP : 05111640000004

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Ir. Siti Rochimah, M.T., Ph.D

NIP: 196810021994032001



(pembimbing 1)

Dwi Sunaryono, S.Kom., M.Kom

NIP: 197205281997021001

(pembimbing 2)

**SURABAYA  
JANUARI 2020**

*[Halaman ini sengaja dikosongkan]*



**RANCANG BANGUN APLIKASI BERBASIS WEB  
PENDETEKSI BORAKS PADA BAKSO DAN LOKASI  
PENJUALANNYA DI SURABAYA TIMUR MENGGUNAKAN  
*ELECTRONIC NOSE***

Nama Mahasiswa : DEWI SEKARINI  
NRP : 0511154000004  
Jurusan : Teknik Informatika ITS  
Dosen Pembimbing I : Ir. Siti Rochimah, M.T., Ph.D  
Dosen Pembimbing II : Dwi Sunaryono, S.Kom., M.Kom.

**Abstrak**

*Isu penggunaan zat aditif buatan pada makanan menjadi salah satu isu yang marak dibicarakan akhir-akhir ini. Zat aditif atau bahan tambahan makanan adalah bahan yang ditambahkan dengan sengaja ke dalam makanan dalam jumlah tertentu, dengan bertujuan untuk memperbaiki penampilan, cita rasa, tekstur, dan memperpanjang daya simpan. Salah satu penggunaan zat aditif yang banyak dibicarakan saat ini adalah penggunaan boraks.*

*dan Penggunaan boraks bahan sebagai pengawet makanan dilarang untuk digunakan menurut Permenkes RI No 722/Menkes Per/IX/1988 tentang Bahan Tambahan Makanan. Penggunaan boraks dengan dosis tinggi dapat menyebabkan penyakit-penyakit berbahaya sehingga penggunaannya dilarang untuk panganan. Bakso merupakan salah satu makanan yang sering didapati mengandung boraks. Para penjual nakal menambahkan boraks sebagai bahan campuran agar bakso yang dibuat memiliki tekstur yang lebih kenyal, lebih awet, dan memiliki warna yang menarik. Sehingga, diperlukan sebuah media untuk memberikan informasi mengenai tempat-tempat penjualan bakso dengan kadar boraks yang tinggi maupun rendah.*

*Maka dari itu, Aplikasi Pendeteksi Lokasi Penjualan Bakso Ber kandungan Boraks diusulkan sebagai solusi dari permasalahan di atas. Dengan memanfaatkan teknologi electronic nose, yaitu sebuah perangkat yang dapat mengidentifikasi komponen-komponen dari bau yang diterima, kandungan boraks*

*pada diharapkan dapat terdeteksi dengan memanfaatkan salah satu ciri dari ciri-ciri bakso berboraks di atas. Selain itu, aplikasi ini memanfaatkan teknologi Android yang telah menjadi salah satu teknologi smartphome yang melekat di masyarakat dan fitur Geotagging yang dapat memberikan keterangan dimanakah lokasi penjualan bakso tersebut dari foto yang di ambil.*

*Tugas Akhir ini berhasil mengimplementasikan sebuah aplikasi yang dapat digunakan sebagai pemberi informasi terkait lokasi penjualan bakso berboraks pada studi kasus di wilayah Surabaya Timur dengan akurasi 90% dan tingkat kepercayaan 91%.*

***Kata kunci: Geotagging, Boraks, Android, Klasifikasi, Electronic nose***

**DESIGN AND IMPLEMENTATION OF WEB BASED  
APPLICATION AS BAKSO CONTAINING BORAX DETECTOR  
AND THE SALES LOCATION IN EAST SURABAYA WITH  
ELECTRONIC NOSE**

Name : DEWI SEKARINI  
NRP : 05111640000004  
Major : Informatics Engineering ITS  
Supervisor I : Ir. Siti Rochimah, M.T., Ph.D  
Supervisor II : Dwi Sunaryono, S.Kom., M.Kom.

**Abstract**

*The issue of the use of artificial additives in food has become one of the most common issues to be discussed these days. Food additives or food additives are ingredients that are added intentionally to food in a certain amount, with the aim of improving the appearance, taste, texture, and extend shelf life. One use of additives which is widely discussed today is the use of borax*

*The use of borax ingredients as food preservatives is prohibited to be used according to the Minister of Health's Regulation No 722/ Menkes Per /IX/1988 concerning Food Additives. The use of high doses of borax can cause dangerous diseases so that its use is prohibited for handling. Meatballs are one of the foods that are often found to contain borax. Naughty sellers add borax as a mixture so that the meatballs are made to have a more resilient texture, more durable, and have an attractive color. So, we need media to provide information about places to sell meatballs with high or low borax levels.*

*Therefore, an application as detector of bakso containing borax sales location is proposed as a solution to problem mentioned above. By utilizing electronic nose technology, which is a device that can identify the components of the odor received, the content of borax is expected to be detected by utilizing one of the characteristics of the meatball meatballs above. In addition, this application utilizes Android technology which has become one of the smartphone technologies inherent in the community and*

*Geotagging features that can provide information on where the meatball sales location from photos taken.*

*This Final Project successfully implemented an application that can be used to give information about location for selling meatball containing borax in East Surabaya with an accuracy of 90% and a confidence level of 91%.*

***Keywords: Geotagging, Borax, Android, Classification, Electronic nose***

## KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Aplikasi Berbasis Web Pendeteksi Boraks Pada Bakso dan Lokasi Penjualannya di Surabaya Timur Menggunakan Electronic Nose”**.

Dalam pelaksanaan Tugas Akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini serta senantiasa menemani dan memberi kemudahan kepada penulis dalam menyelesaikan Tugas Akhir.
2. Kedua Orang Tua Alm. Ayah, Bapak Budi Wahyono dan Ibu penulis, Ibu Nuryani Widi Widayati tersayang yang telah membesarkan penulis, memberikan doa dan dukungan selama ini.
3. Ibu Ir. Siti Rochimah, M.T., Ph.D selaku dosen wali dan dosen pembimbing Tugas Akhir pertama yang telah membimbing, memotivasi dan memberikan banyak masukan dalam pengerjaan Tugas Akhir ini.
4. Bapak Dwi Sunaryono, S.Kom., M.Kom. selaku dosen pembimbing Tugas Akhir kedua yang selalu memberikan koreksi serta masukan-masukan yang dapat penulis kembangkan dari Tugas Akhir ini.
5. Bapak dan Ibu dosen Departemen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Departemen Teknik Informatika ITS.

6. Mbak Shoffi Izza Sabilla yang telah sangat banyak membantu dan membimbing penulis pada pengerjaan Tugas Akhir ini.
7. Diana Hudani Kisyono, teman seperjuangan Tugas Akhir yang saling menyemangati, menghibur dan saling menguatkan selama mengerjakan Tugas Akhir.
8. Ismail Syarief, teman penulis yang tidak pelit ilmu dan selalu mau membantu penulis jika mengalami kesulitan selama pengerjaan Tugas Akhir.
9. Multichat berlima; Almas Aqmarina, Diana Hudani, Denise Sonia Rahmadina, dan Isye Putri Roselin yang telah memberikan warna pada kehidupan perkuliahan penulis selama di Departemen Teknik Informatika ITS.
10. Teman-teman penulis; Maile, Vincent, Alam, Deka, dan Falah yang telah membantu mengisi kehabutan waktu penulis selama waktu perkuliahan.
11. Adinda Mega Agustiarra yang selalu mengajak penulis untuk mencari hiburan ketika lelah selama pengerjaan Tugas Akhir.
12. Teman-teman PH HMTC Garang; Kupon, Vincent, Deka, Nish, Frandita, Fandy, Tamtam, Almas, Hima, Denise, Ganen, Titut, Maile, Diana, Hafid, Nila, Nopal, Kewang, Indra, Zepi, Oe, Azki, dan Izzan yang telah memberikan kesempatan penulis untuk mendapatkan pengalaman kurang lebih satu tahun kepengurusan.
13. Teman-teman Kabinet UKM Sepakbola ITS 2018/2019 yang memberi penulis pengalaman berharga yang tidak bisa dilupakan.
14. Teman-teman pejuang *fast track*; Fadilla, Zevi, Akbar yang berjuang bersama dalam melalui perkuliahan S2 yang melelahkan.
15. Teman-teman Administrator Laboratorium RPL yang telah memberikan tempat kepada penulis selama pengerjaan Tugas Akhir.
16. Teman-teman Administrator Laboratorium Algoritma dan Pemrograman yang telah memberikan tempat bermain selama perkuliahan di Departemen Teknik Informatika ITS.

17. Teman-teman satu angkatan Teknik Informatika ITS 2016 yang saling menyemangati satu sama lain.
18. Pihak-pihak lain yang tidak bisa penulis sebutkan satu per satu.

Penulis sangat berharap bahwa apa yang dihasilkan dari tugas akhir ini bisa memberikan manfaat bagi semua pihak. Tidak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan Tugas Akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Januari 2020

Dewi Sekarini

*[Halaman ini sengaja dikosongkan]*



## DAFTAR ISI

<b>Abstrak .....</b>	<b>vii</b>
<b>Abstract.....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xix</b>
<b>DAFTAR TABEL .....</b>	<b>xxiii</b>
<b>DAFTAR KODE SUMBER.....</b>	<b>xxvii</b>
<b>1 BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah .....	4
1.4. Tujuan .....	4
1.5. Manfaat .....	4
1.6. Metodologi.....	5
1.7. Sistematika Penulisan .....	6
<b>2 BAB II TINJAUAN PUSTAKA.....</b>	<b>9</b>
2.1. Penelitian Sebelumnya.....	9
2.2. Electronic Nose .....	9
2.3. Boraks .....	11
2.4. Klasifikasi .....	12
2.5. K-Nearest Neighbor .....	12
2.6. Support Vector Machine .....	13
2.7. Multilayer Perceptron .....	15

2.8.	PHP.....	16
2.9.	Laravel.....	17
2.10.	Android.....	18
2.11.	Geotagging .....	18
<b>3</b>	<b>BAB III ANALISIS DAN PERANCANGAN.....</b>	<b>21</b>
3.1.	Analisis.....	21
3.1.1.	Analisis Permasalahan.....	21
3.1.2.	Deskripsi Umum Perangkat Lunak.....	22
3.1.3.	Spesifikasi Kebutuhan Perangkat Lunak .....	23
3.2.	Perancangan.....	37
3.2.1.	Perancangan Arsitektur Sistem.....	37
3.2.2.	Perancangan Antarmuka Pengguna .....	38
3.2.3.	Perancangan Data .....	54
3.2.4.	Perancangan Proses Klasifikasi Data.....	69
3.2.5.	Perancangan API Endpoint.....	70
<b>4</b>	<b>BAB IV IMPLEMENTASI.....</b>	<b>75</b>
4.1.	Lingkungan Implementasi .....	75
4.2.	Implementasi Antarmuka Sistem.....	75
4.2.1.	Halaman Melihat Lokasi Penjualan.....	76
4.2.2.	Halaman Mengelola Lokasi Penjualan .....	78
4.2.3.	Halaman Antarmuka Menambah Data Sampel ..	84
4.2.4.	Halaman Antarmuka Mengelola Data Sampel ...	87
4.2.5.	Halaman Antarmuka Melihat Laporan Mengenai Lokasi Penjualan .....	100
4.2.6.	Halaman Antarmuka Melihat Notifikasi .....	101

4.3.	Implementasi Proses Pengumpulan dan Klasifikasi Data	102
4.3.1.	Implementasi Proses Pengumpulan Data .....	102
4.3.2.	Implementasi Proses Latihan Data.....	103
4.3.3.	Implementasi Proses Klasifikasi Data Sampel..	108
4.4.	Implementasi API .....	112
4.4.1.	Implementasi API Mengambil Lokasi Penjualan	112
4.4.2.	Implementasi API Menambah Lokasi Penjualan	115
4.4.3.	Implementasi API Menambah Laporan Mengenai Lokasi Penjualan .....	118
<b>5</b>	<b>BAB V UJI COBA DAN EVALUASI .....</b>	<b>121</b>
5.1.	Lingkungan Uji Coba.....	121
5.2.	Skenario Pengujian Kasus Penggunaan .....	121
5.2.1.	Pengujian Melihat Lokasi Penjualan.....	122
5.2.2.	Pengujian Mengelola Lokasi Penjualan.....	122
5.2.3.	Pengujian Menambah Data Sampel .....	128
5.2.4.	Pengujian Mengelola Data Sampel .....	131
5.2.5.	Pengujian Melihat Laporan Mengenai Lokasi Penjualan	136
5.2.6.	Pengujian Melihat Notifikasi .....	137
5.3.	Skenario Pengujian API.....	138
5.3.1.	Pengujian API Mengambil Lokasi Penjualan ...	138
5.3.2.	Pengujian API Menambah Lokasi Penjualan....	139
5.3.3.	Pengujian API Menambah Laporan Mengenai Lokasi Penjualan .....	141

5.4.	Pengujian Klasifikasi Sampel.....	142
5.5.	Pengujian Usabilitas .....	144
5.6.	Rekapitulasi Pengujian Kasus Penggunaan dan API	145
<b>6</b>	<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>147</b>
6.1.	Kesimpulan.....	147
6.2.	Saran.....	147
	<b>DAFTAR PUSTAKA .....</b>	<b>149</b>
	<b>LAMPIRAN .....</b>	<b>153</b>
	<b>BIODATA PENULIS .....</b>	<b>161</b>

## DAFTAR GAMBAR

Gambar 1.1 Lingkup Pengerjaan Tugas Akhir .....	3
Gambar 2.1 Diagram cara kerja <i>electronic nose</i> .....	10
Gambar 2.2 Proses Reverse <i>GeoCoding</i> .....	19
Gambar 3.1 Proses Bisnis Sistem .....	22
Gambar 3.2 Deskripsi Umum Sistem .....	23
Gambar 3.3 Diagram Kasus Penggunaan Sistem Keseluruhan ..	25
Gambar 3.4 Diagram Kasus Penggunaan Aplikasi Web .....	26
Gambar 3.5 Diagram Kasus Penggunaan Aplikasi Android.....	26
Gambar 3.6 Diagram Aktivitas UC001.....	27
Gambar 3.7 Diagram Aktivitas UC002.....	30
Gambar 3.8 Diagram Aktivitas UC003.....	32
Gambar 3.9 Diagram Aktivitas UC004.....	34
Gambar 3.10 Diagram Aktivitas UC005.....	35
Gambar 3.11 Diagram Aktivitas Melihat Notifikasi.....	36
Gambar 3.12 Ilustrasi Arsitektur Sistem Web .....	37
Gambar 3.13 Rancangan Antarmuka Melihat Lokasi Penjualan	38
Gambar 3.14 Rancangan Halaman Antarmuka Mengelola Data Lokasi Penjualan.....	40
Gambar 3.15 Rancangan Halaman Antarmuka Mengubah Data Lokasi Penjualan.....	41
Gambar 3.16 Rancangan Antarmuka Menghapus Lokasi Penjualan .....	42
Gambar 3.17 Rancangan Antarmuka Menambah Data Sampel..	43
Gambar 3.18 Rancangan Antarmuka Melihat Data Sampel - 1..	45
Gambar 3.19 Rancangan Antarmuka Melihat Data Sampel - 2..	46
Gambar 3.20 Perancangan Antarmuka Modal Melihat Peta Lokasi Penjualan.....	48
Gambar 3.21 Perancangan Antarmuka Melihat Modal Foto Lokasi Penjualan.....	49
Gambar 3.22 Perancangan Antarmuka Dialog Konfirmasi Penghapusan Data Sampel.....	50
Gambar 3.23 Perancangan Antarmuka Dialog Konfirmasi Pengubahan Data Sampel .....	50

Gambar 3.24 Perancangan Antarmuka Modal Perubahan Data Sampel.....	51
Gambar 3.25 Perancangan Antarmuka Melihat Laporan Mengenai Lokasi Penjualan .....	52
Gambar 3.26 Perancangan Antarmuka Melihat Notifikasi.....	53
Gambar 3.27 Electronic Nose Enosika.....	55
Gambar 3.28 Contoh File CSV <i>Electronic Nose</i> Kolom 1 – 7...	56
Gambar 3.29 Contoh File CSV <i>Electronic Nose</i> Kolom 8 – 13.	56
Gambar 3.30 Contoh File CSV <i>Electronic Nose</i> Kolom 14 – 2457	
Gambar 3.31 Contoh File CSV <i>Electronic Nose</i> Kolom 25 – 3157	
Gambar 3.32 Contoh File CSV <i>Electronic Nose</i> Kolom 32 – 3957	
Gambar 3.33 Contoh File CSV <i>Electronic Nose</i> Kolom 40 – 48 .....	58
Gambar 3.34 CDM Sistem .....	58
Gambar 3.35 PDM Sistem.....	59
Gambar 4.1 Implementasi Antarmuka Melihat Lokasi Penjualan .....	77
Gambar 4.2 Implementasi Antarmuka Melihat Daftar Lokasi Penjualan Kolom 1 – 4 .....	79
Gambar 4.3 Implementasi Antarmuka Melihat Daftar Lokasi Penjualan Kolom 5 – 7 .....	79
Gambar 4.4 Implementasi Antarmuka Mengubah Data Lokasi Penjualan .....	80
Gambar 4.5 Implementasi Antarmuka Menghapus Data Lokasi Penjualan .....	84
Gambar 4.6 Implementasi Antarmuka Menambah Data Sampel	84
Gambar 4.7 Implementasi Antarmuka Melihat Data Sampel Bagian Tabel Data Sampel .....	88
Gambar 4.8 Implementasi Antarmuka Melihat Data Sampel Bagian Grafik Sensor MQ2 – MQ5.....	89
Gambar 4.9 Implementasi Antarmuka Melihat Data Sampel Bagian Grafik Sensor MQ6 – MQ9.....	90
Gambar 4.10 Implementasi Antarmuka Melihat Data Sampel Bagian Grafik Sensor MQ135 .....	91

Gambar 4.11 Implementasi Antarmuka Melihat Peta Lokasi Penjualan.....	92
Gambar 4.12 Implementasi Antarmuka Melihat Foto Lokasi Penjualan.....	93
Gambar 4.13 Implementasi Antarmuka Menghapus Data Sampel .....	93
Gambar 4.14 Implementasi Antarmuka Konfirmasi Mengubah Data Sampel.....	96
Gambar 4.15 Implementasi Antarmuka Modal Mengubah Data Sampel .....	96
Gambar 4.16 Implementasi Antarmuka Melihat Laporan Mengenai Lokasi Penjualan.....	101
Gambar 4.17 Implementasi Antarmuka Melihat Notifikasi.....	101
Gambar 4.18 Hasil Latihan Data Latih .....	103
Gambar 4.19 Confusion Matrix, Precision, Recall, Fscore, Support, dan Accuracy dari Proses Training Data.....	107
Gambar 5.1 Tampilan Hasil Pengujian Melihat Lokasi Penjualan .....	123
Gambar 5.2 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 1.....	127
Gambar 5.3 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 2.....	127
Gambar 5.4 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 3.....	127
Gambar 5.5 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 4.....	128
Gambar 5.6 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 5.....	128
Gambar 5.7 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 6.....	128
Gambar 5.8 Tampilan Hasil Pengujian Menambah Data Sampel Skenario 1 .....	130
Gambar 5.9 Tampilan Hasil Pengujian Menambah Data Sampel Skenario 2 .....	131

Gambar 5.10 Tampilan Hasil Pengujian Menambah Data Sampel Skenario 3.....	131
Gambar 5.11 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 1.....	134
Gambar 5.12 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 2.....	134
Gambar 5.13 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 3.....	135
Gambar 5.14 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 4.....	135
Gambar 5.15 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 5.....	135
Gambar 5.16 Tampilan Hasil Pengujian Melihat Laporan Mengenai Lokasi Penjualan.....	136
Gambar 5.17 Tampilan Hasil Pengujian Pengujian Melihat Notifikasi.....	137
Gambar 5.18 Hasil Pengujian API Mengambil Lokasi Penjualan .....	138
Gambar 5.19 Hasil Pengujian API Menambah Lokasi Penjualan Skenario 1.....	140
Gambar 5.20 Hasil Pengujian API Menambah Lokasi Penjualan Skenario 2.....	140
Gambar 5.21 Hasil Pengujian API Menambah Lokasi Penjualan Skenario 3.....	141
Gambar 5.22 Hasil Pengujian API Menambah Laporan Mengenai Lokasi Penjualan Skenario 1 .....	142
Gambar 5.23 Hasil Pengujian API Menambah Laporan Mengenai Lokasi Penjualan Skenario 2 .....	142



## DAFTAR TABEL

Tabel 3.1 Spesifikasi Karakteristik Aktor pada Sistem .....	24
Tabel 3.2 Kebutuhan Fungsional Sistem .....	24
Tabel 3.3 Spesifikasi Kasus Penggunaan Melihat Lokasi Penjualan .....	27
Tabel 3.4 Spesifikasi Kasus Penggunaan Mengelola Lokasi Penjualan.....	28
Tabel 3.5 Spesifikasi Kasus Penggunaan Menambah Data Sampel .....	30
Tabel 3.6 Spesifikasi Kasus Penggunaan Mengolah Data Sampel .....	32
Tabel 3.7 Spesifikasi Kasus Penggunaan Melihat Laporan Mengenai Lokasi Penjualan .....	35
Tabel 3.8 Spesifikasi Kasus Penggunaan Melihat Notifikasi .....	36
Tabel 3.9 Penjelasan Antarmuka Melihat Lokasi Penjualan .....	39
Tabel 3.10 Penjelasan Antarmuka Mengelola Lokasi Penjualan .....	39
Tabel 3.11 Penjelasan Antarmuka Mengubah Data Lokasi Penjualan.....	41
Tabel 3.12 Penjelasan Antarmuka Menghapus Lokasi Kegiatan.....	42
Tabel 3.13 Penjelasan Antarmuka Menambah Data Sampel .....	43
Tabel 3.14 Penjelasan Antarmuka Melihat Data Sampel.....	46
Tabel 3.15 Penjelasan Antarmuka Modal Melihat Peta Lokasi Penjualan.....	48
Tabel 3.16 Penjelasan Antarmuka Modal Melihat Foto Lokasi Penjualan.....	49
Tabel 3.17 Penjelasan Antarmuka Dialog Konfirmasi Penghapusan Data Sampel .....	49
Tabel 3.18 Penjelasan Antarmuka Dialog Konfirmasi Perubahan Data Sampel .....	51
Tabel 3.19 Penjelasan Antarmuka Modal Perubahan Data Sampel .....	51
Tabel 3.20 Penjelasan Antarmuka Melihat Laporan Mengenai Lokasi Penjualan.....	53
Tabel 3.21 Penjelasan Antarmuka Melihat Notifikasi .....	54

Tabel 3.22 Kolom Jenis Bau Sebagai Parameter Sensor Untuk Nilai Data Input .....	55
Tabel 3.23 Deskripsi Rinci Tabel Users .....	59
Tabel 3.24 Deskripsi Rinci Tabel Jenis Sampel .....	60
Tabel 3.25 Deskripsi Rinci Tabel Lokasi Penjualan .....	61
Tabel 3.26 Deskripsi Rinci Tabel Data Sampel .....	63
Tabel 3.27 Deskripsi Rinci Tabel Laporan.....	68
Tabel 3.28 Metode Klasifikasi yang Digunakan .....	69
Tabel 3.29 Deskripsi Atribut Keluaran Proses Klasifikasi.....	70
Tabel 3.30 Deskripsi Perancangan API Endpoint .....	70
Tabel 3.31 Struktur JSON Response API Endpoint Mengambil Lokasi Penjualan .....	71
Tabel 3.32 Struktur JSON Request API Endpoint Menambah Lokasi Penjualan .....	72
Tabel 3.33 Struktur JSON Response API Endpoint Menambah Lokasi Penjualan .....	73
Tabel 3.34 Struktur JSON Request API Endpoint Menambah Laporan.....	73
Tabel 3.35 Struktur JSON Response API Menambah Laporan..	74
Tabel 4.1 Penjelasan Warna Marker Pada Peta Melihat Lokasi Penjualan .....	78
Tabel 4.2 Penjelasan Kode Sumber Melihat Lokasi Penjualan..	78
Tabel 4.3 Penjelasan Kode Sumber Menampilkan Lokasi Penjualan Sesuai Jenis Sampel.....	80
Tabel 4.4 Penjelasan Fungsi Mengubah Data Lokasi Penjualan	82
Tabel 4.5 Penjelasan Fungsi Menghapus Data Lokasi Penjualan .....	83
Tabel 4.6 Penjelasan Fungsi Menambah Data Sampel.....	87
Tabel 4.7 Penjelasan Fungsi Menampilkan Data Sampel .....	93
Tabel 4.8 Penjelasan Fungsi Menghapus Data Sampel .....	95
Tabel 4.9 Penjelasan Fungsi Mengubah Data Sampel.....	99
Tabel 4.10 Penjelasan Fungsi Melihat Laporan Mengenai Lokasi Penjualan .....	101
Tabel 4.11 Contoh Hasil Pengambilan Data Sampel Dengan Electronic Nose dan Arduino .....	102

Tabel 4.12 Penjelasan Kode Sumber Implementasi Pembuatan Model Data Sampel dengan Metode Multilayer Perceptron.....	106
Tabel 4.13 Penjelasan Kode Sumber Implementasi Klasifikasi Data Sampel.....	111
Tabel 4.14 Penjelasan Kode Sumber Implementasi API Mengambil Lokasi Penjualan .....	114
Tabel 4.15 Penjelasan Kode Sumber Implementasi API Menambah Lokasi Penjualan.....	117
Tabel 4.16 Penjelasan Kode Sumber Implementasi API Menambah Laporan Mengenai Lokasi Penjualan.....	120
Tabel 5.1 Lingkungan Pengujian Aplikasi.....	121
Tabel 5.2 Lingkungan Server Aplikasi Web.....	121
Tabel 5.3 Kasus Pengujian Melihat Lokasi Penjualan.....	122
Tabel 5.4 Kasus Pengujian Mengelola Lokasi Penjualan .....	124
Tabel 5.5 Kasus Pengujian Menambah Data Sampel .....	129
Tabel 5.6 Kasus Mengelola Data Sampel .....	131
Tabel 5.7 Kasus Pengujian Melihat Laporan Mengenai Lokasi Penjualan.....	136
Tabel 5.8 Kasus Pengujian Melihat Notifikasi .....	137
Tabel 5.9 Pengujian API Mengambil Lokasi Penjualan .....	138
Tabel 5.10 Pengujian API Menambah Lokasi Penjualan.....	139
Tabel 5.11 Pengujian API Menambah Laporan Mengenai Lokasi Penjualan.....	141
Tabel 5.12 Hasil Klasifikasi Data Sampel yang Sudah Diketahui Kandungan Boraksnya .....	143
Tabel 5.13 Hasil Klasifikasi Data Sampel yang Belum Diketahui Kandungan Boraksnya .....	143
Tabel 5.14 Hasil Kuisisioner Usabilitas .....	144
Tabel 5.15 Rekapitulasi Pengujian Kasus Penggunaan Aplikasi Berbasis Web .....	145
Tabel 5.16 Rekapitulasi Pengujian API Endpoint .....	146

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi Melihat Lokasi Penjualan.....	76
Kode Sumber 4.2 Fungsi Menampilkan Lokasi Penjualan Sesuai Jenis Sampel.....	80
Kode Sumber 4.3 Fungsi Mengubah Data Lokasi Penjualan ....	82
Kode Sumber 4.4 Fungsi Menghapus Data Lokasi Penjualan ...	83
Kode Sumber 4.5 Fungsi Menambah Data Sampel .....	86
Kode Sumber 4.6 Fungsi Menampilkan Data Sampel.....	92
Kode Sumber 4.7 Fungsi Menghapus Data Sampel.....	95
Kode Sumber 4.8 Fungsi Mengubah Data Sampel .....	99
Kode Sumber 4.9 Fungsi Melihat Laporan Mengenai Lokasi Penjualan.....	100
Kode Sumber 4.10 Implementasi Pembuatan Model Data Sampel dengan Metode Multilayer Perceptron.....	106
Kode Sumber 4.11 Implementasi Klasifikasi Data Sampel .....	111
Kode Sumber 4.12 Implementasi API Mengambil Lokasi Penjualan.....	114
Kode Sumber 4.13 Implementasi API Menambah Lokasi Penjualan.....	117
Kode Sumber 4.14 Implementasi API Menambah Laporan Mengenai Lokasi Penjualan.....	120

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

Bab pendahuluan membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

### **1.1. Latar Belakang**

Isu penggunaan zat aditif buatan pada makanan menjadi salah satu isu yang marak dibicarakan akhir-akhir ini. Zat aditif atau bahan tambahan makanan adalah bahan yang ditambahkan dengan sengaja ke dalam makanan dalam jumlah tertentu, dengan bertujuan untuk memperbaiki penampakan, cita rasa, tekstur, dan memperpanjang daya simpan. Salah satu penggunaan zat aditif yang banyak dibicarakan saat ini adalah penggunaan boraks.

Boraks atau sodium tetraborate merupakan campuran garam mineral konsentrasi tinggi dengan nama kimia natrium tetraborat ( $\text{Na}_2\text{B}_4\text{O}_7 \cdot 10\text{H}_2\text{O}$ ). Dalam industri, boraks biasa dipakai sebagai bahan pembuat solder, pengawet kayu, antiseptik kayu, dan pengontrol kecoa. Namun, terkadang boraks disalahgunakan sebagai bahan campuran makanan. Boraks biasa digunakan sebagai bahan pengawet dan pengental pada makanan seperti bakso, tahu, mie basah, lontong, dan siomay.

Penggunaan boraks bahan sebagai pengawet makanan dilarang untuk digunakan menurut Permenkes RI No 722/Menkes Per/IX/1988 tentang Bahan Tambahan Makanan. Penggunaan boraks dilarang karena penggunaan boraks pada dosis tinggi, yaitu 5 – 10 gram pada orang dewasa dan 3 – 6 gram pada anak-anak, menimbulkan gejala-gejala yang tertunda meliputi badan terasa tidak nyaman (malaise), mual, nyeri hebat pada perut bagian atas (epigastrik), pendarahan gastroenteritis disertai muntah darah, diare, lemah, mengantuk, demam, rasa sakit kepala, dan menyebabkan penumpukan zat kimia pada otak, hati, dan ginjal yang dapat mengakibatkan kanker[1].

Ada beberapa ciri dari makanan yang mengandung boraks. Ciri tersebut meliputi bertekstur sangat kenyal, tidak mudah hancur, atau sangat renyah, berwarna sangat mencolok dibandingkan dengan warna aslinya, tidak rusak meskipun disimpan di suhu ruang selama sehari-hari, dan memiliki bau yang menyengat.

Bakso merupakan salah satu makanan yang sering didapati mengandung boraks. Para penjual nakal menambahkan boraks sebagai bahan campuran agar bakso yang dibuat memiliki tekstur yang lebih kenyal, lebih awet, dan memiliki warna yang menarik. Padahal, bakso merupakan salah satu makanan yang sering dikonsumsi oleh masyarakat Indonesia. Sehingga, diperlukan sebuah media untuk memberikan informasi mengenai tempat-tempat penjualan yang terindikasi menjual bakso berkandungan boraks maupun tidak.

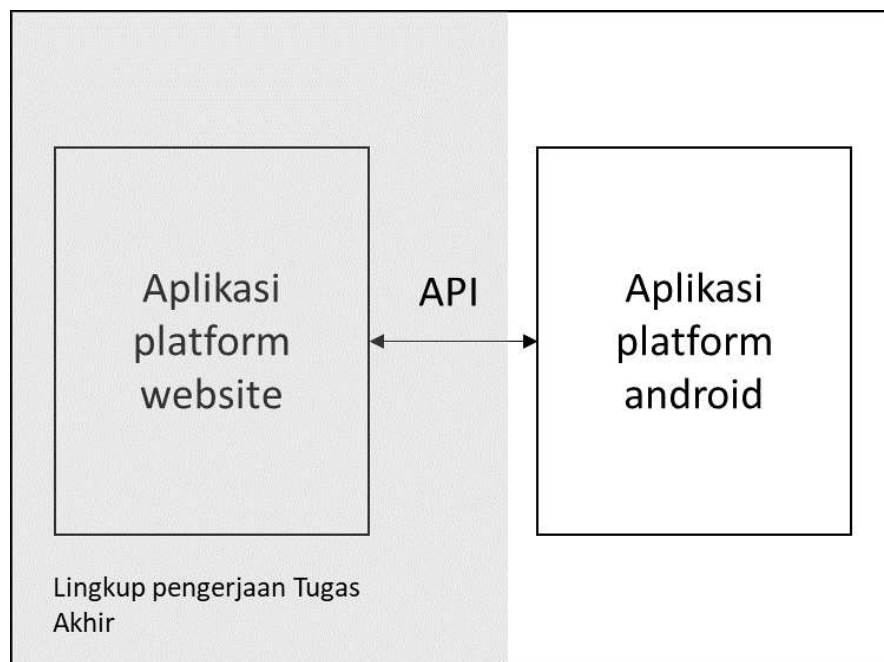
Maka dari itu, Aplikasi Pendeteksi Boraks pada Bakso dan Lokasi Penjualannya diusulkan sebagai solusi dari permasalahan di atas. Dengan memanfaatkan teknologi *electronic nose*, yaitu sebuah perangkat yang dapat mengidentifikasi komponen-komponen dari bau yang diterima, kandungan boraks pada diharapkan dapat terdeteksi dengan memanfaatkan salah satu ciri dari ciri-ciri bakso berboraks di atas. Selain itu, aplikasi ini memanfaatkan teknologi Android yang telah menjadi salah satu teknologi smartphone yang melekat di masyarakat dan fitur Geotagging yang dapat memberikan keterangan dimanakah lokasi penjualan bakso tersebut dari foto yang di ambil.

Aplikasi yang diusulkan untuk dibangun pada Tugas Akhir ini terdiri dari dua *platform*, yaitu website dan Android. *Platform web* akan digunakan sebagai penyedia layanan API yang akan digunakan oleh aplikasi *platform* Android, media untuk mengelola data-data lokasi penjualan dan proses klasifikasi dari data sampel. Sedangkan, platform Android akan digunakan sebagai media untuk memberi informasi mengenai lokasi-lokasi penjualan dari data sampel yang akan diteliti dan sebagai media untuk menyampaikan laporan atau informasi mengenai lokasi penjualan



baik yang diusulkan untuk diteliti maupun informasi perubahan pada lokasi penjualan yang sudah ada sebelumnya pada sistem.

Tugas Akhir ini akan berfokus pada perancangan dan implementasi pada platform web dan perancangan API yang akan digunakan oleh platform Android. Selanjutnya, pembuatan platform Android pada aplikasi ini dilakukan pada penelitian berjudul “Rancang Bangun Aplikasi Berbasis Android Pengindikasi Formalin Pada Tahu dan Lokasi Penjualannya di Surabaya Timur Menggunakan Identifikasi Geografis[2]”. Untuk bagan lingkup pengerjaan, dapat dilihat pada Gambar 1.1.



**Gambar 1.1 Lingkup Pengerjaan Tugas Akhir**

Dengan dibangunnya aplikasi tersebut, diharapkan masyarakat dapat terbantu untuk terhindar dari mengonsumsi bakso ber kandungan boraks.

## **1.2. Rumusan Masalah**

Perumusan masalah yang terdapat pada Tugas Akhir ini, antara lain adalah:

1. Bagaimanakah cara mengumpulkan sampel dan mengklasifikasikan data kandungan boraks pada bakso?

2. Bagaimanakah cara untuk mengimplementasikan pembuatan aplikasi pendeteksi kandungan boraks pada bakso?
3. Bagaimanakah cara untuk menyajikan data pasar-pasar mana yang menjual bakso dengan kandungan boraks?

### 1.3. Batasan Masalah

Batasan masalah yang terdapat pada Tugas Akhir ini, sebagai berikut:

1. Teknologi yang digunakan dalam pembuatan aplikasi ini adalah bahasa pemrograman web (PHP, HTML, CSS JavaScript), Java, kerangka kerja Laravel.
2. Sampel bakso yang digunakan sebagai data dalam aplikasi ini adalah sampel yang didapatkan dari pasar-pasar yang berada di wilayah Surabaya Timur.
3. Mesin *electronic nose* yang digunakan untuk mengumpulkan data merupakan mesin *e-nose* yang menggunakan sensor MQ2, MQ3, MQ4, MQ5, MQ6, MQ7, MQ8, MQ9, dan MQ135.

### 1.4. Tujuan

Tugas Akhir ini adalah membuat sebuah aplikasi berbasis Android dan *Geotagging* yang dapat menampilkan lokasi-lokasi penjualan bakso berkandungan boraks di pasar-pasar wilayah Surabaya Timur sesuai dengan hasil klasifikasi kadar kandungan boraks dengan menggunakan *e-nose* dan fitur *geotagging*

### 1.5. Manfaat

Manfaat yang diharapkan dari pembuatan Tugas Akhir ini antara lain:

1. Diharapkan dengan terciptanya aplikasi pendeteksi lokasi penjualan bakso di pasar-pasar pada wilayah Surabaya Timur ini masyarakat dapat terbantu untuk menemukan lokasi penjualan bakso dengan kadar boraks yang aman untuk dikonsumsi.

2. Bagi badan hukum nasional terkait seperti Badan Pengawas Obat dan Makanan, aplikasi ini diharapkan dapat mempermudah pengecekan kelayakan dan pendataan tahu yang dijual di pasaran.

## **1.6. Metodologi**

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

### **1. Penyusunan Proposal Tugas Akhir**

Proposal Tugas Akhir ini berisi tentang deskripsi pendahuluan dari Tugas Akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, tujuan dari pembuatan Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan Tugas Akhir mulai dari penyusunan proposal hingga penyusunan buku Tugas Akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

### **2. Studi literatur**

Pada tahap ini dilakukan studi literatur pada penelitian-penelitian sebelumnya menggunakan *electronic nose*. Pada tahap ini juga dilakukan mengenai studi literatur terkait metode dalam pembuatan aplikasi berbasis web dan Android.

### **3. Implementasi**

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

#### **4. Uji Coba dan Evaluasi**

Pada tahapan ini dilakukan uji coba pada perangkat lunak dengan pengujian berdasarkan *usability* dan *maintenance*.

#### **5. Penyusunan Buku Tugas Akhir**

Pada tahapan ini disusun buku yang membuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

### **1.7. Sistematika Penulisan**

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

#### **1. Bab I. Pendahuluan**

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan Tugas Akhir.

#### **2. Bab II. Tinjauan Pustaka**

Bab tinjauan pustaka berisi penjelasan mengenai dasar teori yang mendukung pengerjaan Tugas Akhir.

#### **3. Bab III. Analisis dan Perancangan**

Bab ini berisi tentang desain sistem, rancangan basis data, diagram kasus penggunaan, diagram aktivitas dan rancangan antarmuka pengguna.

#### **4. Bab IV. Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa tampilan antarmuka yang telah dibuat dan dapat berfungsi untuk mengakomodir kebutuhan fungsional yang ada.

#### **5. Bab V. Uji Coba dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

## **6. Bab VI. Kesimpulan dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan perangkat lunak.

#### **2.1. Penelitian Sebelumnya**

Beberapa studi sebelumnya telah memperlihatkan kapabilitas *electronic nose* dalam membedakan beberapa tipe daging. Sensor semikonduktor digunakan untuk pengambilan data dari bau daging. Studi oleh Nurjuliana et al menggunakan *principal component analysis* (PCA) untuk membedakan antara lemak babi, lemak ayam, lemak sapi, lemak domba, dan sampel campuran dari lemak babi dan lemak ayam. Eksperimen dilakukan pada temperature 50°C to 200°C [3].

Terdapat juga studi lainnya yang terkait dengan mendeteksi kemurnian daging dengan hasil penelitian menunjukkan bahwa sistem ini dapat membedakan daging sapi yang dicampur dengan daging babi dengan perbandingan 0%, 10%, 25%, 50%, 75%, 90%, dan 100% dengan akurasi 97,86% menggunakan algoritma kNN [4].

#### **2.2. Electronic Nose**

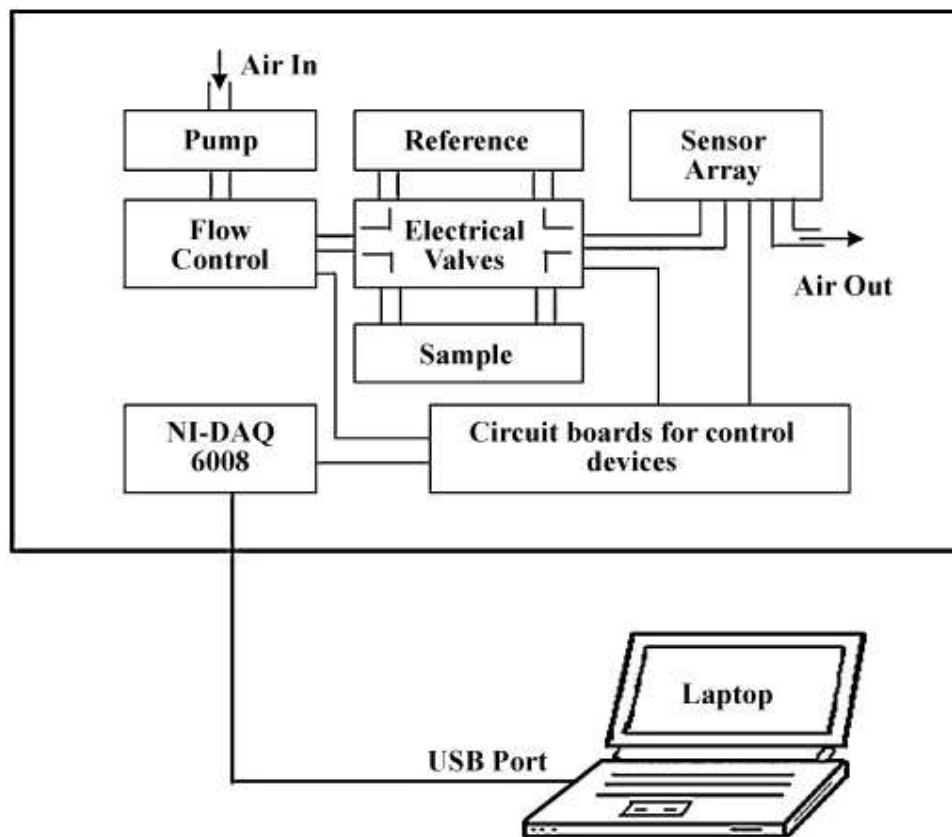
*Electronic nose* merupakan sebuah sistem yang dikembangkan sebagai sistem yang dibangun menyerupai cara kerja indera penciuman manusia untuk mendeteksi bau, uap, dan gas secara otomatis. Komponen utama dari sebuah *e-nose* adalah sistem deteksi dan sistem pengenalan pola secara otomatis. Sistem deteksi bau dapat berupa sekumpulan sensor kimia yang setiap elemennya dapat mendeteksi properti yang berbeda, atau dapat merupakan perangkat yang mendeteksi satu jenis properti saja [5].

*E-nose* dikembangkan untuk meniru cara kerja indera penciuman manusia yang mana mekanisme dari fungsinya tidak

bisa dipisahkan antara satu dengan yang lain. Pada dasarnya, cara kerja *e-nose* dibagi menjadi tiga bagian sebagai berikut:

1) Sistem pengantar sampel

Sistem pengantar sampel memungkinkan pembentukan senyawa volatil yang dihasilkan dari proses *headspace sampling*. Sistem ini nantinya akan mengirimkan partikel-partikel tersebut ke sistem pendeteksi dari sebuah *electronic nose*[6].



**Gambar 2.1** Diagram cara kerja *electronic nose*

2) Sistem pendeteksi

Sistem pendeteksi merupakan bagian yang terdiri dari sekumpulan sensor. Pada dasarnya, bau disusun dari molekul molekul yang memiliki ukuran dan bentuk yang spesifik. Setiap molekul memiliki ukuran yang sesuai dan bentuk dengan reseptor pada hidung manusia. Ketika sebuah reseptor menerima sebuah molekul, maka reseptor akan mengirimkan



sinyal kepada otak untuk mengidentifikasi bau yang berkaitan dengan molekul yang diterima. Sama seperti indera penciuman manusia, ketika sensor-sensor yang berfungsi sebagai reseptor menerima molekul, sensor tersebut akan bereaksi dengan cara mengalami perubahan di karakteristik listrik sensor[6].

### 3) Sistem komputasi

Secara umum, sensor-sensor yang ada pada *electronic nose* sensitif kepada semua molekul secara spesifik. Ketika sensor menerima adanya bau, maka respons spesifik akan terekam dan perangkat *e-nose* mengubah sinyal tersebut menjadi sebuah nilai digital[6].

Diagram kerja mesin *electronic nose* dapat dilihat pada Gambar 2.1[6].

## 2.3. Boraks

Boraks adalah campuran garam mineral konsentrasi tinggi berwarna putih yang mengandung atom Boron(B) dan Oksigen(O) yang memiliki nama kimia *tetraborat decahydrate* dengan rumus molekul  $\text{Na}_2\text{B}_4\text{O}_7 \cdot 10\text{H}_2\text{O}$  atau  $\text{Na}_2[\text{B}_4\text{O}_5(\text{OH})_4] \cdot 8\text{H}_2\text{O}$ . Boraks umumnya digunakan pada industri seperti industri kertas, kayu, plastic, keramik, dan gelas.

Namun perlu diingat bahwa boraks sangat berbahaya bagi kesehatan manusia terkonsumsi dalam jumlah yang banyak. Makanan mengandung boraks yang dikonsumsi sedikit demi. Boraks yang dikonsumsi cukup tinggi dapat menyebabkan gejala pusing, muntah, mencret, kejang perut, kerusakan ginjal, hilang nafsu makan. Bahkan hingga kejadian meninggal dunia dapat terjadi pada bayi dan anak kecil hanya dengan dosis toksin boraks dalam tubuh yang telah melampaui lebih dari 5 gram. Sementara kematian pada orang dewasa dapat terjadi jika dosis toksin boraks sudah mencapai 10-20 gram. Batas aman penggunaan boraks pada makanan secara legal adalah 1 gram per 1 kilogram pangan (1/1000). Karena dampak bahaya karsinogennya terhadap kesehatan manusia maka Pemerintah secara resmi telah melarang

penggunaan boraks sebagai bahan tambahan dalam proses produksi makanan[7].

## 2.4. Klasifikasi

Klasifikasi terhadap data pada dasarnya adalah proses untuk mengklasifikasikan sebuah data ke dalam kategori atau kelas yang telah ada sebelumnya. Tujuan utama dari klasifikasi adalah untuk memprediksi kelas target dari data yang dianalisis[8]. Proses klasifikasi disusun dari dua tahap yaitu:

1. Training/Tahap Latihan

Tahap ini merupakan tahap untuk membentuk model dari klasifikasi dari data yang sudah tersedia. Model harus melalui proses *training* untuk menghasilkan prediksi dari hasil yang akurat[9].

2. Tahap Klasifikasi

Tahap ini merupakan tahap dimana data baru dites dengan *training data* untuk mendapatkan hasil kelas klasifikasi dan mengestimasi akurasi dari aturan klasifikasi[9].

## 2.5. K-Nearest Neighbor

*K-Nearest Neighbor* (KNN) merupakan sebuah algoritma klasifikasi berdasarkan jarak antara data pembelajaran dengan objek yang akan diklasifikasikan. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Nilai  $k$  adalah jumlah tetangga terdekat dari sebuah titik yang sedang dibandingkan. Nilai  $k$  yang terbaik untuk algoritma ini tergantung pada data, secara umumnya, nilai  $k$  yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur[10]. Cara kerja algoritma KNN adalah sebagai berikut:

1. Hitung jarak setiap titik pada tetangga dengan titik saat ini

2. Urutkan jarak dari yang paling dekat dan pilih  $k$  titik pertama
3. Hitung dan pilih kelas yang paling sering muncul diantara  $k$  tetangga.

Keunggulan dari algoritma ini antara lain adalah algoritma ini mudah untuk diimplementasikan, kuat terkait ruang pencarian; misalnya kelas tidak harus dipisahkan secara linear, classifier dapat diperbarui dengan biaya yang kecil apabila *instance* baru dengan kelas yang diperbarui disajikan, dan hanya sedikit parameter untuk disetel, yaitu metric jarak dan  $k$ . Sedangkan, kelemahan dari algoritma ini adalah pengujian tiap *instance* yang mahal karena dibutuhkan perhitungan jarak pada setiap *instance* yang diketahui, kepekaan terhadap atribut *noise* dan atribut yang tidak relevan, sehingga menghasilkan angka jarak yang kurang berarti, serta kesensitifan terhadap dataset yang tidak seimbang[11].

Pada Tugas Akhir ini, algoritma *k-Nearest Neighbor* akan digunakan sebagai salah satu algoritma klasifikasi yang akan dibandingkan akurasi sehingga metode dengan akurasi paling tinggi akan digunakan sebagai metode dalam membuat model klasifikasi pada Tugas Akhir ini. Metode ini dipilih karena metode ini kuat dalam melakukan klasifikasi pada *dataset* yang memiliki fitur cukup banyak.

## 2.6. Support Vector Machine

*Support Vector Machine* (SVM) adalah metode learning machine yang bekerja atas prinsip Structural Risk Minimization (SRM) dengan tujuan menemukan hyperplane terbaik yang memisahkan dua buah class pada input space[12].

Sebuah model SVM pada dasarnya merupakan presentasi dari kelas-kelas berbeda pada sebuah *hyperplane* di ruang multidimensional. *Hyperplane* tersebut akan dibentuk secara berulang oleh SVM untuk memperkecil kesalahan. Tujuan dari SVM adalah untuk membagi *dataset* menjadi kelas-kelas untuk

menemukan *maximum marginal hyperplane*(MMH)[13]. Beberapa konsep penting dari SVM antara lain:

1. *Support vectors* adalah *datapoints* terdekat dari sebuah *hyperplane*.
2. *Hyperplane* merupakan ruang yang terbagi diantara kumpulan objek yang mempunyai kelas berbeda.
3. Margin merupakan jarak antara dua garis pada *datapoints* terdekat pada setiap kelas.

Cara kerja model dari algoritma SVM secara umum adalah sebagai berikut:

1. Menemukan garis yang memisahkan dan mengklasifikasikan *dataset training* secara benar
2. Memilih garis yang mempunyai jarak maksimum dari *datapoint* terdekat diantara garis atau *boundaries* tersebut

Ada dua macam SVM, yaitu:

#### 1. Linear SVM

Linear SVM merupakan SVM yang digunakan pada data yang terpisah secara linear, yang artinya apabila sebuah *dataset* dapat diklasifikasikan menjadi dua kelas dengan sebuah garis lurus, maka data tersebut dapat dikatakan terpisah secara linear dan *classifier* yang digunakan adalah Linear SVM.

#### 2. Non-Linear SVM

Non-Linear SVM digunakan pada sekumpulan data yang tidak bisa diklasifikasikan dengan garis lurus[14].

Keuntungan dari menggunakan SVM antara lain efektif pada ruang dengan dimensi tinggi, masih efektif apabila digunakan pada kasus dimana jumlah dimensi lebih tinggi daripada jumlah sampel, menggunakan subset dari *training points* pada fungsi penentuan sehingga efisien pada memori, dan sebagai yang artinya fungsi kernel berbeda dapat digunakan pada fungsi keputusan yang berbeda. Kelemahan dari penggunaan SVM antara lain adalah apabila jumlah fitur jauh lebih banyak daripada jumlah sampel, menghindari *over-fitting* pada pemilihan fungsi Kernel merupakan hal yang penting dan SVM tidak secara langsung menyediakan estimasi kemungkinan, sehingga hal ini dilakukan dengan

menggunakan *five-fold cross-validation* yang cenderung memakan banyak biaya[15].

Pada Tugas Akhir ini, algoritma SVM linear akan digunakan sebagai salah satu algoritma klasifikasi yang akan dibandingkan akurasi sehingga metode dengan akurasi paling tinggi akan digunakan sebagai metode dalam membuat model klasifikasi pada Tugas Akhir ini. Pemilihan metode SVM pada tugas akhir ini dikarenakan metode ini cocok digunakan untuk *dataset* yang hanya memiliki 2 kelas untuk diklasifikasikan.

## 2.7. Multilayer Perceptron

Multilayer Perceptron (MLP) adalah sebuah algoritma klasifikasi yang merupakan bagian dari jaringan saraf tiruan. Ciri dari MLP adalah adanya beberapa lapisan node input yang terhubung menjadi graph antara lapisan *input* dan *output*[16].

Perceptron pada MLP terdiri dari input dan output layer yang terhubung secara penuh. Cara kerja algoritma MLP merupakan sebagai berikut:

1. Input akan didorong ke depan melalui MLP dengan cara mengambil produk titik dari input dengan bobot yang ada antara *input layer* dan *hidden layer*. Produk titik ini menghasilkan nilai pada *hidden layer*.
2. MLP memanfaatkan fungsi aktivasi pada setiap *layer* yang dihitung. *Output* yang telah dihitung pada *layer* saat ini akan didorong melalui fungsi aktivasi ini.
3. Ketika *output* yang telah dihitung pada *hidden layer* telah didorong melalui fungsi aktivasi, dorong *output* tersebut ke *layer* selanjutnya pada MLP dengan cara mengambil produk titik dengan bobot yang berkaitan
4. Ulangi langkah 2 dan 3 sampai *output layer* diraih.
5. Pada output layer, perhitungan akan digunakan untuk algoritma *backpropagation* yang berkaitan dengan fungsi aktivasi yang telah terpilih untuk MLP atau keputusan akan dibuat sesuai dengan *output* tersebut[17].

Keunggulan dari algoritma ini adalah kapabilitas dalam menangani model yang non-linear, dan kapabilitas dalam mempelajari model secara *real-time*. Sedangkan, kelemahan dari algoritma ini adalah MLP dengan *hidden layers* mempunyai fungsi *non-convex loss* yang mana mempunyai lebih dari satu minimum lokal, MLP memerlukan pengaturan pada *hyperparameter* seperti jumlah *hidden neurons, layers*, dan iterasi serta MLP sensitif terhadap *scaling* fitur[18].

Pada Tugas Akhir ini, algoritma MLP akan digunakan sebagai salah satu algoritma klasifikasi yang akan dibandingkan akurasi sehingga metode dengan akurasi paling tinggi akan digunakan sebagai metode dalam membuat model klasifikasi pada Tugas Akhir ini. Metode ini dipilih karena metode ini kuat dalam melakukan klasifikasi pada *dataset* yang memiliki fitur cukup banyak.

## 2.8. PHP

PHP atau *Hypertext Preprocessor* merupakan sebuah bahasa *server-side scripting* yang didesain khusus untuk pengembangan Web. Di dalam sebuah halaman HTML, baris kode PHP dapat disertakan yang nantinya akan dieksekusi setiap kali halaman tersebut diakses. Kode PHP akan diinterpretasikan oleh web server dan menghasilkan HTML atau *output* lain yang nantinya akan dilihat oleh pengguna. PHP merupakan *open source product*, yang artinya siapapun dapat menggunakan, mengubah, dan mendistribusikan kembali secara gratis[19].

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Pada awalnya PHP merupakan singkatan dari Personal Home Page. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana,

tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Saat ini PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: Hypertext Preprocessor.

PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. PHP dirilis dalam lisensi PHP License, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang biasa digunakan untuk proyek Open Source[20].

## 2.9. Laravel

Laravel merupakan sebuah *framework* PHP yang mengimplementasikan arsitektur model, view, controller atau MVC. Laravel didesain untuk mengurangi biaya dalam memulai pengembangan sebuah web dan biaya dalam *maintenance*, dan meningkatkan pengalaman dalam bekerja dengan cara menyediakan sintaks yang ekspresif dan jelas serta sekumpulan fungsionalitas-fungsionalitas inti yang dapat mempersingkat waktu implementasi[21]. MVC(Model View Controller) adalah sebuah arsitektur yang memisahkan antara *business logic* dan antarmuka pengguna. Secara umum, arsitektur ini memisahkan sebuah aplikasi menjadi 3 bagian yaitu; Model, View and Controller[22].

1. Model merupakan bagian tingkat terendah yang bertanggung jawab untuk memelihara data dan bagian yang menangani hubungan antara controller dengan *database*
2. View merupakan bagian yang menangani presentation logic. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
3. Controller merupakan bagian yang menangani kode perangkat lunak yang mengendalikan interaksi antara *model* dan *view*.

## 2.10. Android

Android merupakan sebuah sistem operasi berbasis Linux yang berjalan pada ponsel pintar, tablet, dan perangkat sejenisnya. Android merupakan sistem operasi yang mendukung banyak aplikasi di ponsel pintar. Perangkat keras yang mendukung sistem operasi Android adalah perangkat berbasis platform arsitektur ARM. Hal tersebut membantu pengguna untuk menggunakan fitur-fitur lanjut dari sebuah perangkat dengan tingkat kenyamanan yang lebih.

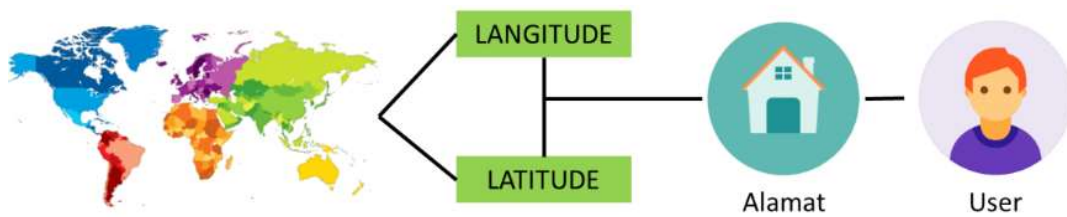
Android merupakan sistem operasi *open source* yang artinya Android merupakan hal yang gratis dan semua orang bisa menggunakannya. Pengembangan Android didukung oleh bahasa pemrograman Java. Versi pertama Android Development Kit (ADK) yaitu versi 1.0 dirilis pada 2008 dan versi terakhir yang diperbarui adalah versi Pie[23].

## 2.11. Geotagging

*Geotagging* adalah proses penambahan identifikasi ke berbagai media seperti foto, video, website, pesan SMS, atau RSS feed berupa metadata geospasial. Data ini biasanya terdiri dari koordinat lintang dan bujur (*latitude* dan *longitude*). Data lainnya juga dapat mencakup ketinggian, jarak, akurasi data, dan nama tempat. Hal ini biasanya digunakan untuk memberikan geotag pada foto. (Suhardi, 2011)

Dasar untuk geotagging adalah posisi. Posisi ini akan, dalam hampir setiap kasus, berasal dari Global Positioning System (GPS). GPS secara formal diketahui sebagai Navigation Satellite Timing and Ranging (NAVSTAR) GPS yang sebenarnya dikembangkan untuk keperluan militer. Karena kemampuan navigasi populernya dan karena teknologi GPS bisa diakses dengan peralatan kecil yang tidak terlalu mahal maka pemerintah membuat sistem ini bisa dinikmati oleh rakyat sipil. USA memiliki teknologi GPS dan Departemen Pertahanan yang memeliharanya[24].





**Gambar 2.2 Proses Reverse *GeoCoding***

Proses satu arah, misalnya dari Nama Jalan menjadi koordinat LonLat disebut dengan GeoCoding Service. Sedangkan untuk translasi (mapping) dari LonLat menjadi Nama Jalan atau Daerah, yang disebut dengan Reverse GeoCoding Service. Google Maps API merupakan aplikasi antarmuka yang dapat diakses lewat Javascript agar Google Map dapat ditampilkan pada halaman web yang sedang dibangun. Untuk dapat mengakses Google Map, harus dilakukan pendaftaran API Key terlebih dahulu dengan data pendaftaran berupa nama domain web yang dibangun. Banyak sekali kegunaan Google Map untuk website yang dibuat, diantaranya dapat digunakan untuk menampilkan lokasi pemilik website ( pada About Us ), lokasi event/kegiatan, atau dapat juga digunakan untuk aplikasi GIS berbasis web[25][26].

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis dan perancangan dari perangkat lunak yang dibangun pada Tugas Akhir ini. Perancangan sistem akan mencantumkan analisis kebutuhan yang diperlukan di perangkat lunak setelah mendapatkan proses bisnis. Perancangan sistem ini akan direpresentasikan dengan diagram *Unified Modelling Language* (disebut UML).

#### **3.1. Analisis**

Subbab ini membahas analisis kebutuhan perangkat lunak, yang meliputi analisis perancangan perangkat lunak secara garis besar, deskripsi umum, dan kebutuhan perangkat lunak.

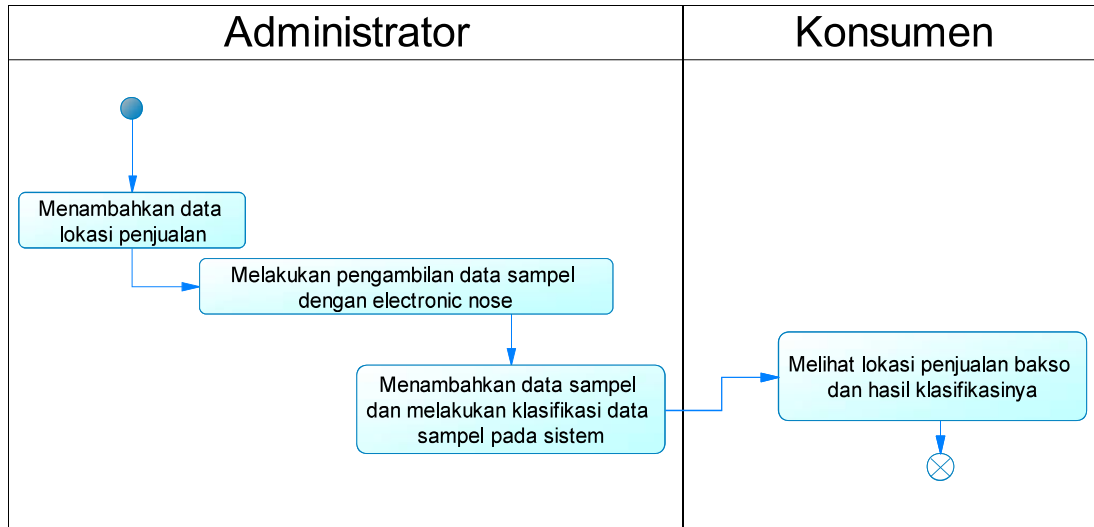
##### **3.1.1. Analisis Permasalahan**

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah bagaimana membangun sebuah aplikasi yang dapat memberi informasi tempat penjualan bakso baik yang bebas dari kandungan boraks maupun terindikasi mengandung boraks.

Era sekarang, teknologi bergerak sedang berkembang pesat dan terlibat pada hampir semua aspek pada kehidupan sehari-hari[27]. Teknologi bergerak digandrungi masyarakat karena kemudahannya untuk diakses kapan saja dan dimana saja. Selain itu, *smartphone* saat ini sudah dilengkapi dengan fitur kamera dan jaringan internet sehingga memungkinkan pengguna untuk mengunggah foto tersebut ke sebuah layanan. Fitur ini juga dilengkapi dengan keterangan mengenai lokasi dimana foto ini diambil, sehingga dapat digunakan sebagai media untuk menyimpan data lokasi-lokasi penjualan bakso yang akan dijadikan objek penelitian.

Untuk mengatasi permasalahan tersebut, penulis akan mengimplementasikan pembuatan aplikasi untuk mendeteksi lokasi-lokasi yang menjual bakso murni maupun terindikasi berkandungan boraks dengan basis Android dan menggunakan

fitur *geotagging*. Proses bisnis dari aplikasi ini dapat dilihat pada Gambar 3.1.



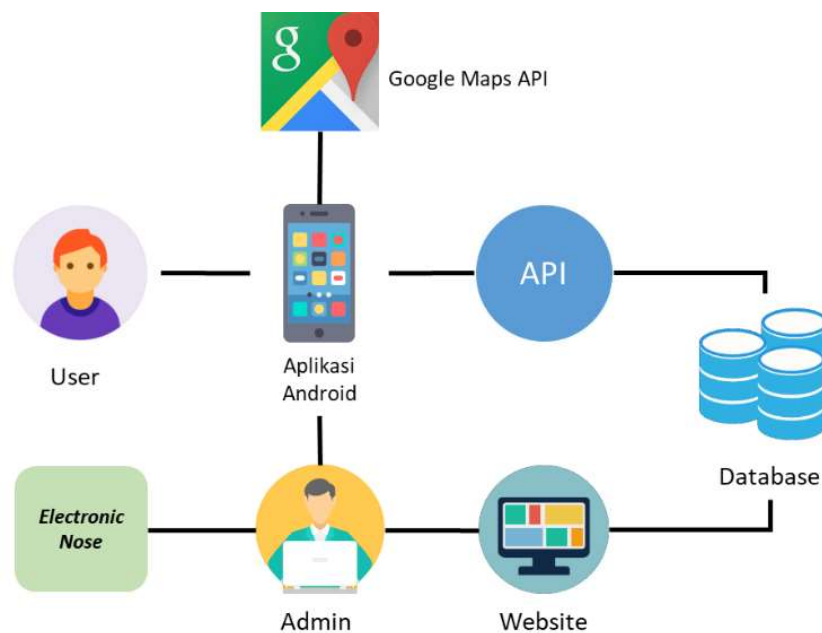
**Gambar 3.1 Proses Bisnis Sistem**

### 3.1.2. Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini akan dibuat sebuah perangkat lunak berupa aplikasi berbasis Android untuk mendeteksi kandungan boraks pada bakso dan lokasi penjualannya. Aplikasi ini terdiri dari dua *platform*, yaitu *platform* Android dan *web*. Deskripsi umum perangkat lunak dapat dilihat di Gambar 3.2.

*Platform* Android bagi konsumen berfungsi untuk menampilkan informasi mengenai lokasi-lokasi penjualan bakso yang sampelnya sudah dilakukan pengujian dan klasifikasi kandungan dan kadar boraks di dalamnya dan berfungsi untuk menambahkan laporan mengenai lokasi penjualan bakso untuk diuji sampelnya atau laporan yang lainnya. Untuk administrator, selain untuk melihat lokasi penjualan bakso, *platform* Android digunakan sebagai media untuk menambahkan lokasi penjualan yang sampelnya akan diuji. Pada *platform* Android akan memanfaatkan Google Maps API untuk mewujudkan fitur *geotagging* dan *geocoder* pada proses penambahan data lokasi penjualan. Sedangkan, *platform web* yang akan digunakan hanya

oleh administrator berfungsi sebagai aplikasi untuk menyediakan layanan-layanan yang akan digunakan oleh aplikasi Android dan sebagai aplikasi untuk mengelola lokasi-lokasi penjualan bakso, data sampel, dan melakukan klasifikasi terhadap kandungan boraks yang terdapat pada sampel-sampel bakso dari masing-masing lokasi penjualan.



**Gambar 3.2 Deskripsi Umum Sistem**

### 3.1.3. Spesifikasi Kebutuhan Perangkat Lunak

Subbab ini membahas spesifikasi kebutuhan perangkat lunak dari hasil analisis yang telah dilakukan. Bagian ini berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk kebutuhan fungsional, kebutuhan non-fungsional, diagram kasus penggunaan, dan diagram aktivitas.

#### 3.1.3.1. Aktor

Aktor adalah pihak-pihak yang terlibat dan berinteraksi langsung dengan sistem. Aktor yang berada pada sistem ini dibahas lebih rinci pada Tabel 3.1.

**Tabel 3.1 Spesifikasi Karakteristik Aktor pada Sistem**

No	Aktor	Tugas	Hak Akses
1	Administrator	<ul style="list-style-type: none"> <li>• Mengelola data</li> </ul>	<ul style="list-style-type: none"> <li>• Mengelola (melihat, menambah, mengubah, menghapus) data lokasi penjualan dan sampel</li> <li>• Melihat laporan mengenai lokasi penjualan bakso</li> </ul>
2	Konsumen	<ul style="list-style-type: none"> <li>• Melihat lokasi penjualan</li> <li>• Memberi informasi mengenai lokasi penjualan</li> </ul>	<ul style="list-style-type: none"> <li>• Melihat lokasi penjualan bakso beserta detailnya</li> <li>• Menambahkan laporan mengenai lokasi penjualan bakso</li> </ul>

### 3.1.3.2. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan yang harus dimiliki oleh perangkat lunak, reaksi terhadap suatu masukan, hasil yang dilakukan pada situasi khusus. Kebutuhan fungsional dari perangkat lunak untuk aplikasi ini dijelaskan pada Tabel 3.2.

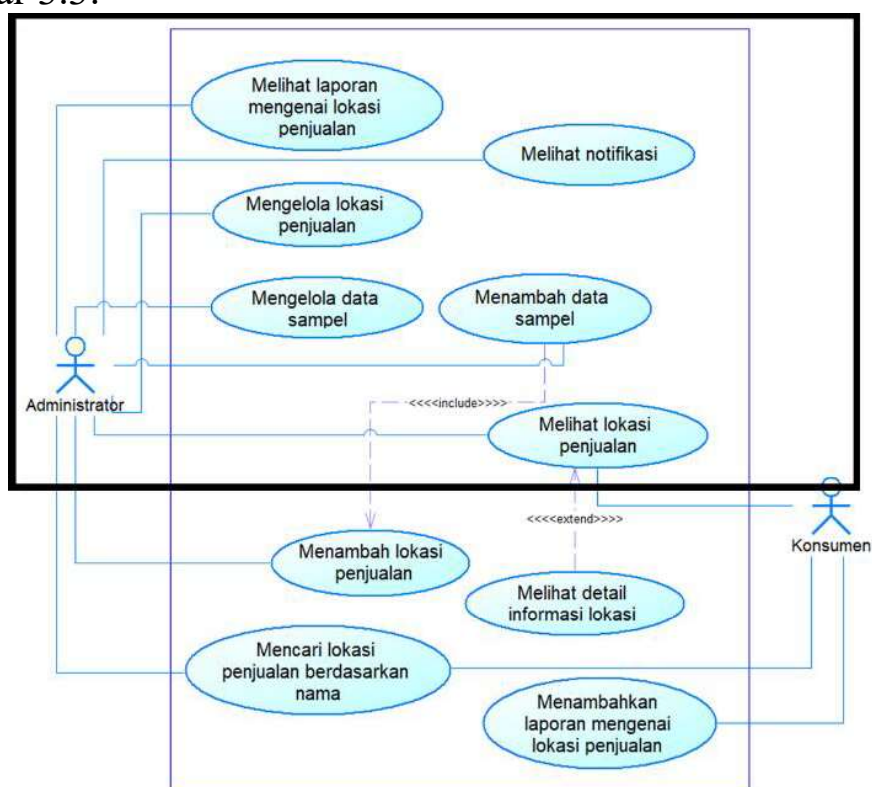
**Tabel 3.2 Kebutuhan Fungsional Sistem**

Kode	Kebutuhan Fungsional	Deskripsi
F01	Menangani data penjualan kelola lokasi	Menampilkan lokasi penjualan yang ada, melakukan penambahan data, pengubahan, serta penghapusan data lokasi penjualan.
F02	Menangani data sampel kelola	Menampilkan data sampel, melakukan penambahan data, pengubahan, serta penghapusan data sampel.
F03	Menangani data laporan kelola laporan	Menampilkan laporan mengenai lokasi penjualan dan melakukan penambahan data.

Kode	Kebutuhan Fungsional	Deskripsi
	mengenai lokasi penjualan	
F04	Melakukan klasifikasi kadar boraks pada data sampel	Melakukan proses klasifikasi kadar boraks pada data sampel yang ada.

### 3.1.3.3. Diagram Kasus Penggunaan

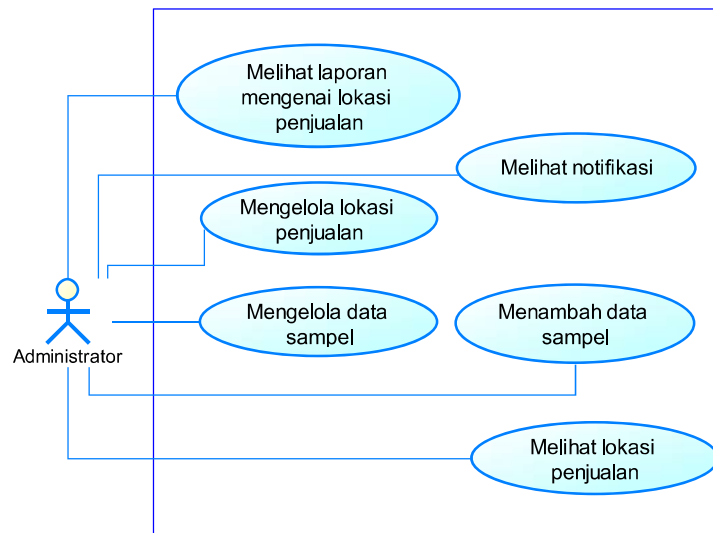
Kasus penggunaan perangkat lunak dijelaskan secara rinci dalam subbab ini. Kasus penggunaan didasarkan pada hasil analisis kebutuhan fungsional dari perangkat lunak. Kasus penggunaan aplikasi ini pada kedua *platform* dapat dilihat pada Gambar 3.3.



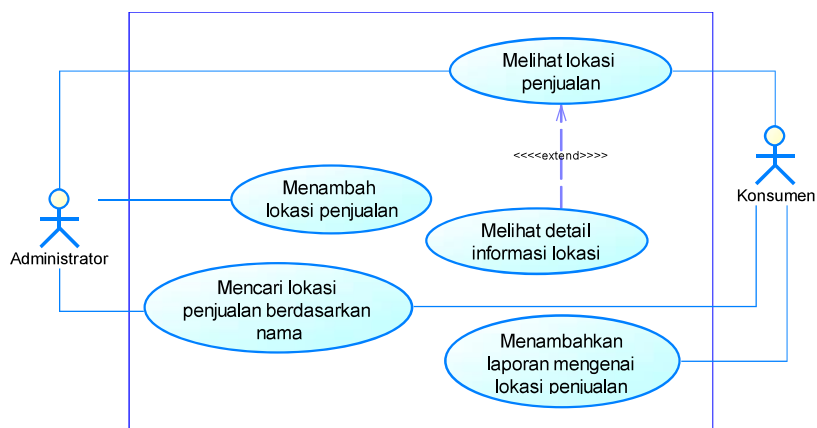
**Gambar 3.3 Diagram Kasus Penggunaan Sistem Keseluruhan**

Pada Tugas Akhir ini, pengembangan aplikasi difokuskan pada bagian *platform web* yang akan digunakan oleh aktor

administrator saja. Sehingga, pengembangan kasus penggunaan pada Tugas Akhir ini terdapat pada Gambar 3.3 bagian yang ada di dalam kotak berwarna hitam. Untuk kasus penggunaan aplikasi website dapat dilihat pada Gambar 3.4. Sedangkan, untuk kasus penggunaan aplikasi Android dapat dilihat pada Gambar 3.5.



**Gambar 3.4 Diagram Kasus Penggunaan Aplikasi Web**



**Gambar 3.5 Diagram Kasus Penggunaan Aplikasi Android**

### 3.1.3.3.1. Kasus Penggunaan Melihat Lokasi Penjualan

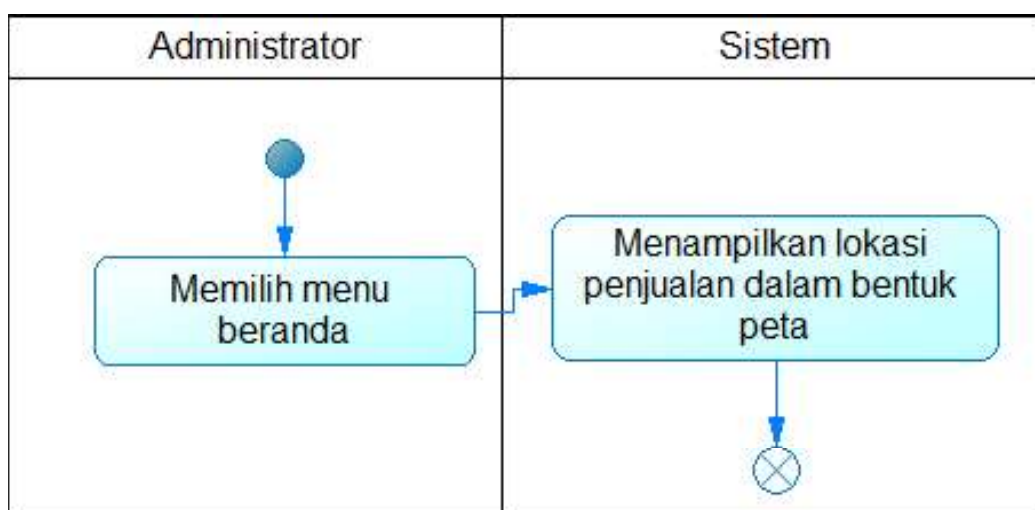
Kasus penggunaan ini diakses ketika administrator membuka halaman *web* dan memilih menu beranda untuk melihat daftar lokasi penjualan dalam bentuk peta dan detailnya pada halaman



web. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.3.

**Tabel 3.3 Spesifikasi Kasus Penggunaan Melihat Lokasi Penjualan**

<b>Nama</b>	Melihat Lokasi Penjualan
<b>Nomor</b>	UC001
<b>Deskripsi</b>	Administrator dapat melihat data lokasi penjualan yang sudah ditambahkan
<b>Aktor</b>	Administrator
<b>Kondisi Awal</b>	Data lokasi penjualan telah tersimpan di <i>database</i>
<b>Kondisi Akhir</b>	Sistem menampilkan data lokasi penjualan yang ada
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator memilih menu beranda	2. Sistem menampilkan daftar lokasi penjualan dalam bentuk peta
<b>Alur Alternatif</b>	
-	



**Gambar 3.6 Diagram Aktivitas UC001**

Skenario aktivitas dari kasus penggunaan selanjutnya digambarkan ke dalam diagram aktivitas yang dapat dilihat pada Gambar 3.6.

### 3.1.3.3.2. Kasus Penggunaan Mengelola Lokasi Penjualan

Kasus penggunaan mengubah lokasi penjualan merupakan kasus penggunaan yang memungkinkan pengguna untuk melihat, mengubah dan menghapus data lokasi penjualan yang pernah ditambahkan sebelumnya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.4 dan diagram aktivitas pada Tabel 3.4.

**Tabel 3.4 Spesifikasi Kasus Penggunaan Mengelola Lokasi Penjualan**

<b>Nama</b>	Mengelola Lokasi Penjualan
<b>Nomor</b>	UC002
<b>Deskripsi</b>	Administrator dapat melihat, mengubah, dan menghapus data lokasi penjualan yang sudah ditambahkan
<b>Aktor</b>	Administrator
<b>Kondisi Awal</b>	Data lokasi penjualan telah tersimpan di <i>database</i>
<b>Kondisi Akhir</b>	Data lokasi penjualan yang dipilih telah diperbarui atau dihapus.
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator memilih menu lokasi penjualan	2. Sistem menampilkan jenis sampel yang dapat dipilih
3. Administrator memilih jenis sampel	4. Sistem menampilkan data lokasi penjualan
5. Administrator memilih opsi ubah atau hapus pada salah lokasi penjualan	
5.1. Administrator memilih mengubah lokasi penjualan	5.1.1. Sistem menampilkan form perubahan lokasi penjualan

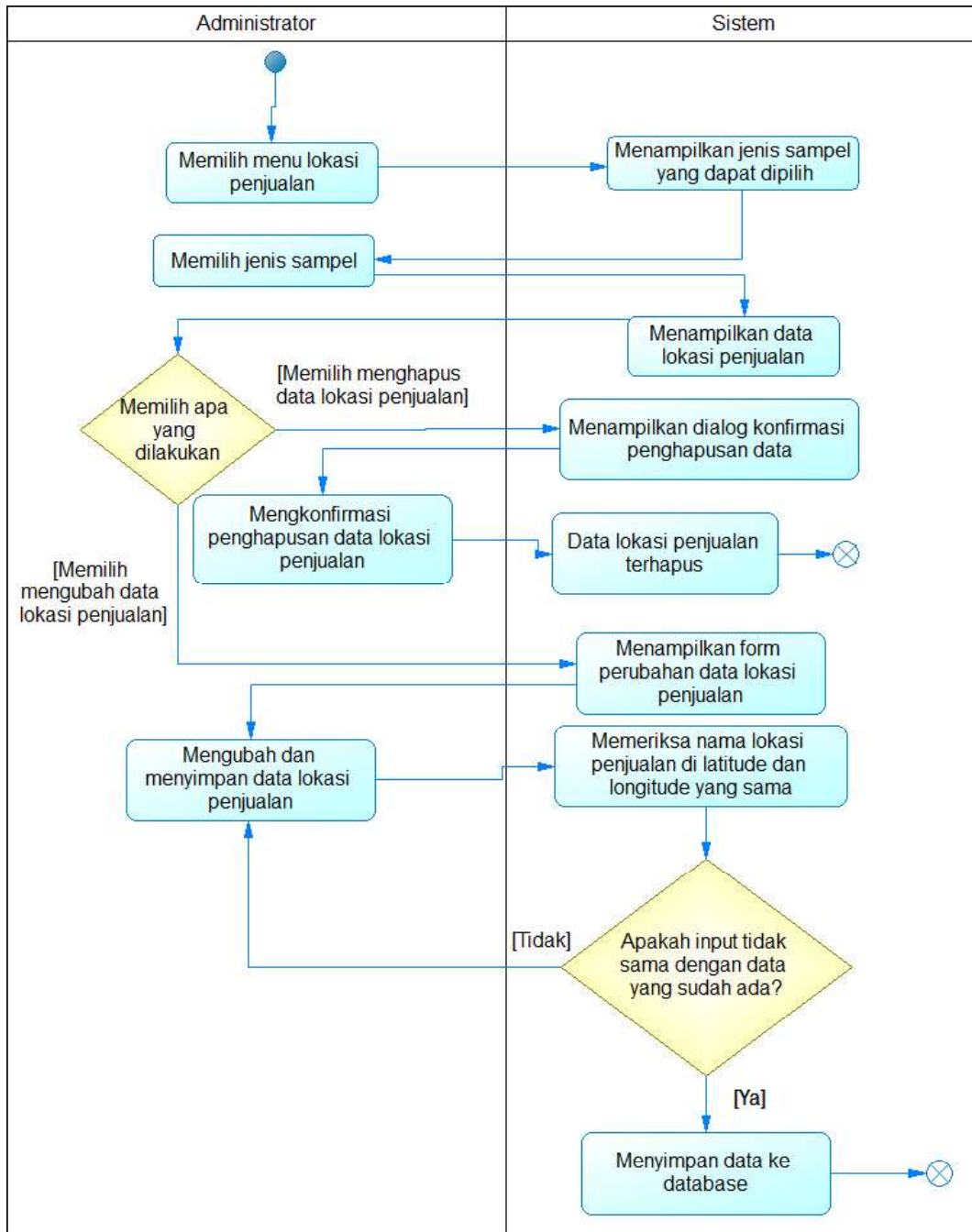
5.1.2. Administrator mengubah dan menyimpan data lokasi penjualan	5.1.3. Memeriksa nama lokasi penjualan di longitude dan latitude yang sama A1. Apabila ada nama lokasi penjualan di longitude dan latitude yang sama
5.2. Administrator memilih menghapus lokasi penjualan	5.1.4. Menyimpan data lokasi penjualan di <i>database</i> 5.2.1. Sistem menampilkan dialogi konfirmasi penghapusan
5.2.2. Administrator mengkonfirmasi penghapusan data lokasi penjualan	5.2.3. Sistem menghapus data lokasi penjualan dari <i>database</i>
<b>Alur Alternatif</b>	
A1. Apabila ada nama lokasi penjualan di longitude dan latitude yang sama	
Aktor	Sistem
	A1.1. Sistem akan menampilkan pesan error bahwa terdapat nama yang sama pada satu lokasi penjualan
A1.2. Kembali ke alur 5.1.2	

Berdasarkan skenario kasus penggunaan pada Tabel 3.4 selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan mengelola data lokasi penjualan dapat dilihat pada Gambar 3.7.

### 3.1.3.3.3. Kasus Penggunaan Menambah Data Sampel

Kasus penggunaan mengubah lokasi penjualan merupakan kasus penggunaan yang memungkinkan pengguna untuk menambah data sampel bakso yang telah didapatkan dengan *electronic nose* sesuai dengan lokasi penjualannya. Selain itu, proses klasifikasi data sampel juga terjadi di kasus penggunaan ini.

Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.5 dan diagram aktivitas pada Gambar 3.8.



**Gambar 3.7 Diagram Aktivitas UC002**

**Tabel 3.5 Spesifikasi Kasus Penggunaan Menambah Data Sampel**

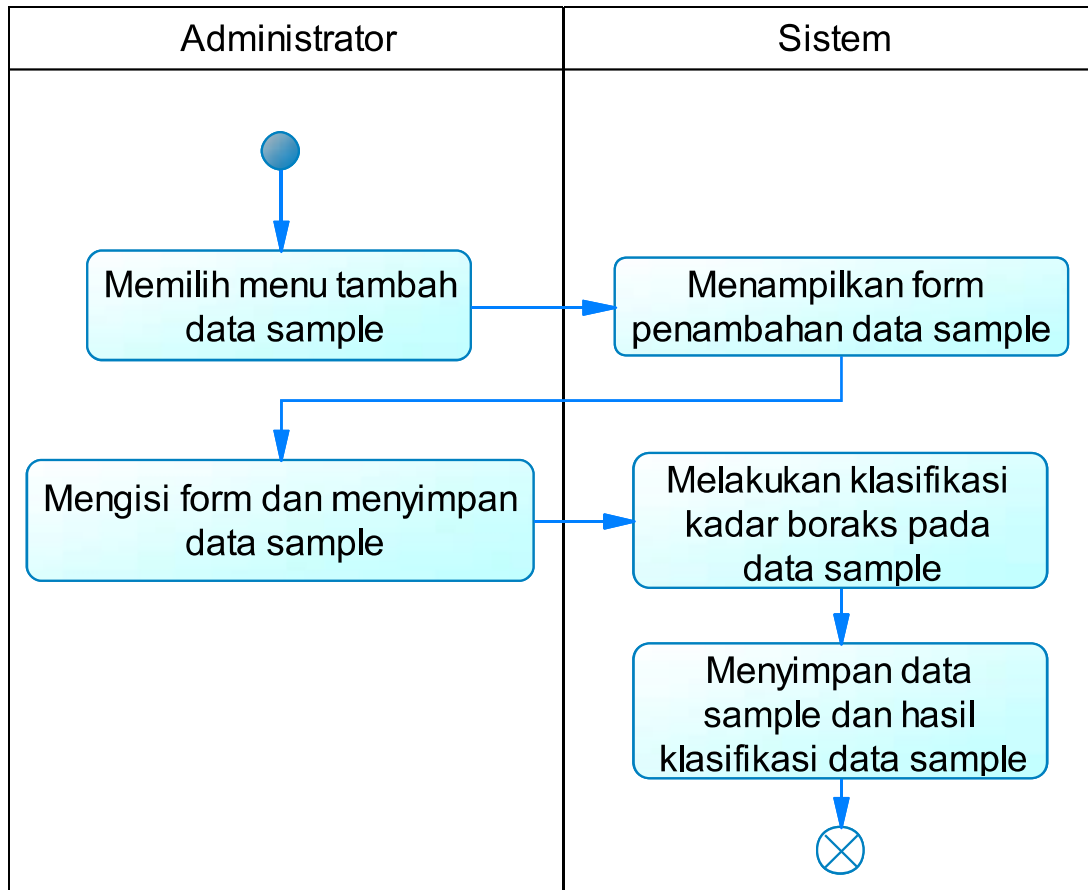
<b>Nama</b>	Melihat Data Sampel
-------------	---------------------

<b>Nomor</b>	UC003
<b>Deskripsi</b>	Administrator dapat menambahkan data sampel bakso yang telah didapatkan dengan <i>electronic nose</i> sesuai dengan lokasi penjualannya dan melakukan proses klasifikasi kadar boraks pada data sampel bakso
<b>Aktor</b>	Administrator
<b>Kondisi Awal</b>	Data lokasi penjualan bakso telah tersimpan di <i>database</i> dan data sampel bakso belum tersimpan di <i>database</i>
<b>Kondisi Akhir</b>	Data sampel tersimpan di <i>database</i> dan data sampel telah terklasifikasi kadar boraksnya
<b>Alur Normal</b>	
Aktor	Sistem
<ol style="list-style-type: none"> <li>1. Administrator memilih menu tambah data sampel</li> <li>3. Administrator mengisi dan menyimpan form data sampel</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan form penambahan data sampel</li> <li>4. Sistem menampilkan data lokasi penjualan</li> <li>5. Sistem melakukan klasifikasi kadar boraks pada data sampel</li> <li>6. Sistem menyimpan data sampel dan hasil klasifikasi data sampel di <i>database</i></li> </ol>
<b>Alur Alternatif</b>	
-	

#### 3.1.3.3.4. Kasus Penggunaan Mengelola Data Sampel

Kasus penggunaan ini memungkinkan administrator untuk melihat, mengubah, dan menghapus data sampel yang sudah ada. Jika melakukan perubahan pada data sampel, sistem akan otomatis melakukan klasifikasi pada data sampel yang baru ditambahkan. Apabila administrator memilih untuk melakukan penghapusan

data sampel, maka hasil klasifikasi juga akan terhapus. Untuk penjelasan lebih lanjut bisa dilihat pada Tabel 3.6.



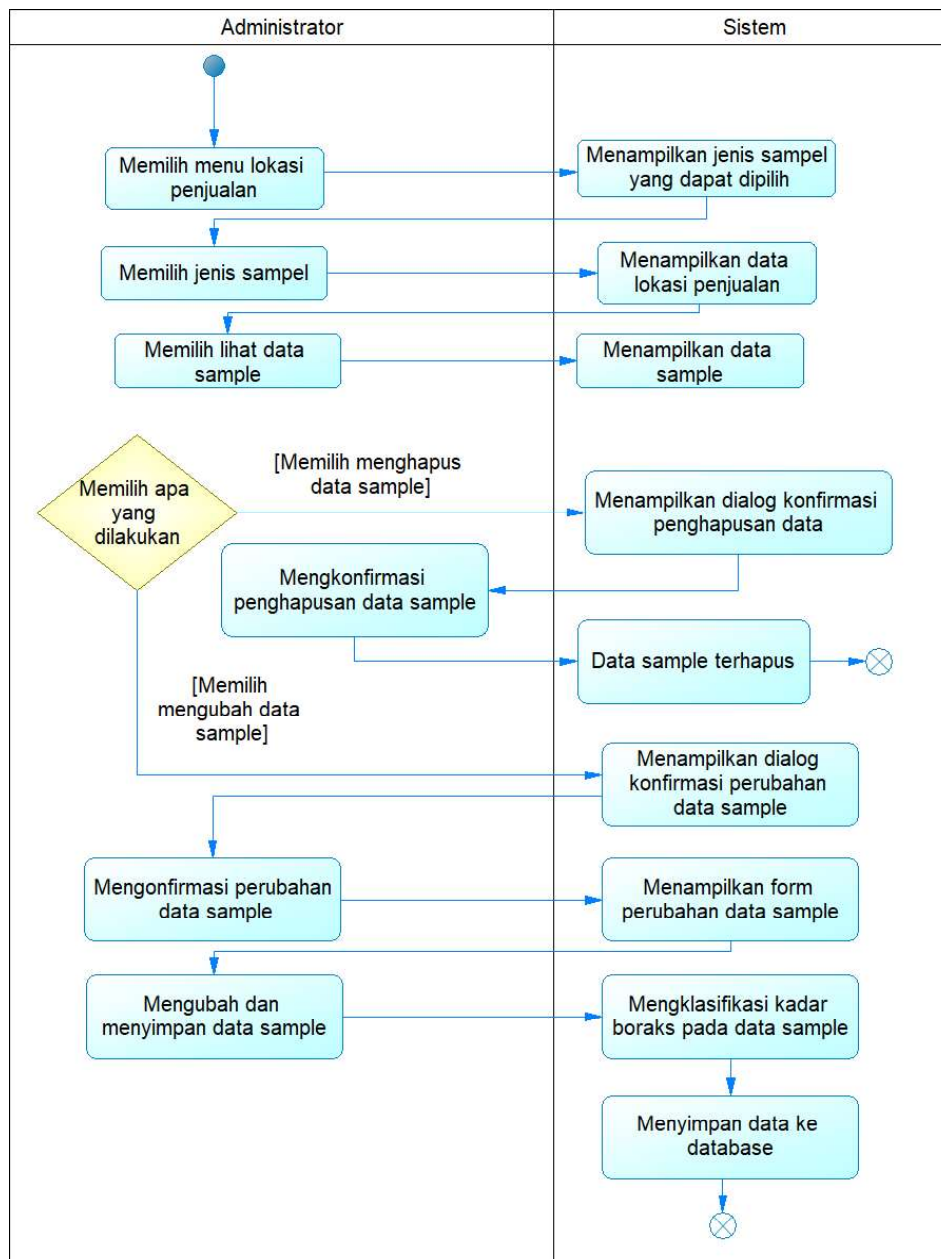
**Gambar 3.8 Diagram Aktivitas UC003**

**Tabel 3.6 Spesifikasi Kasus Penggunaan Mengolah Data Sampel**

<b>Nama</b>	Mengelola Lokasi Penjualan
<b>Nomor</b>	UC004
<b>Deskripsi</b>	Administrator dapat melihat, mengubah, dan menghapus data sampel yang sudah ditambahkan
<b>Aktor</b>	Administrator
<b>Kondisi Awal</b>	Data sampel telah tersimpan di <i>database</i>
<b>Kondisi Akhir</b>	Data sampel yang dipilih telah diperbarui atau dihapus.

<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator memilih menu lokasi penjualan	2. Sistem menampilkan jenis sampel yang dapat dipilih
3. Administrator memilih jenis sampel	4. Sistem menampilkan data lokasi penjualan
5. Administrator memilih lihat pada salah lokasi penjualan	6. Sistem menampilkan data sampel
7. Administrator memilih opsi ubah atau hapus data sampel	
7.1. Administrator memilih ubah data sampel	7.1.1. Sistem menampilkan form konfirmasi perubahan data sampel
7.1.2. Administrator mengonfirmasi perubahan data sampel	7.1.3. Sistem menampilkan form perubahan data sampel
7.1.4. Administrator mengubah dan menyimpan data sampel	7.1.5. Sistem melakukan klasifikasi data sampel
	7.1.6. Sistem menyimpan data sampel di <i>database</i>
7.2. Administrator memilih hapus data sampel	7.2.1. Sistem menampilkan dialog konfirmasi penghapusan data sampel
7.2.2. Administrator mengonfirmasi penghapusan data sampel	7.2.3. Sistem menghapus data sampel dari <i>database</i>
<b>Alur Alternatif</b>	
-	

Berdasarkan skenario kasus penggunaan pada Tabel 3.6 Spesifikasi Kasus Penggunaan Mengolah Data Sampel selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Diagram aktivitas dari kasus penggunaan melihat data sampel dapat dilihat pada Gambar 3.9.



**Gambar 3.9 Diagram Aktivitas UC004**

### 3.1.3.3.5. Kasus Penggunaan Melihat Laporan Mengenai Lokasi Penjualan

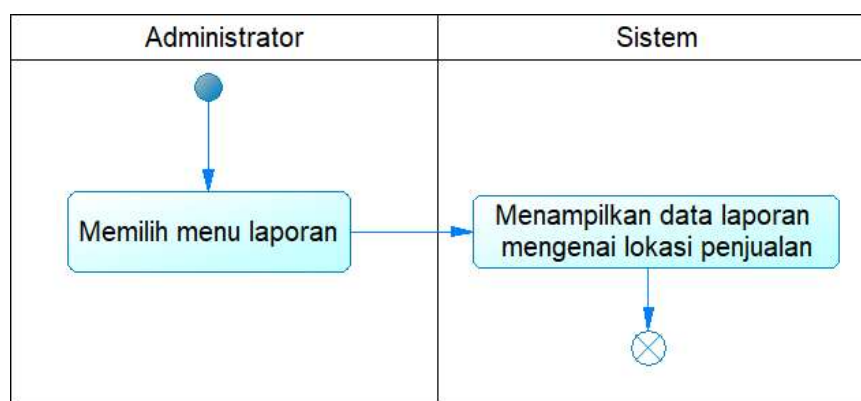
Kasus penggunaan melihat laporan mengenai lokasi penjualan memungkinkan administrator untuk melihat laporan-laporan yang didapatkan dari konsumen yang melaporkan adanya lokasi baru atau adanya perubahan pada lokasi penjualan yang sudah ada melalui *platform* Android. Spesifikasi kasus



penggunaan ini dapat dilihat pada Tabel 3.7 dan untuk diagram aktivitasnya dapat dilihat pada Gambar 3.10.

**Tabel 3.7 Spesifikasi Kasus Penggunaan Melihat Laporan Mengenai Lokasi Penjualan**

<b>Nama</b>	Melihat Laporan Mengenai Lokasi Penjualan	
<b>Nomor</b>	UC005	
<b>Deskripsi</b>	Administrator dapat melihat data laporan mengenai lokasi penjualan yang sudah ditambahkan	
<b>Aktor</b>	Administrator	
<b>Kondisi Awal</b>	Data laporan mengenai lokasi penjualan telah tersimpan di <i>database</i>	
<b>Kondisi Akhir</b>	Sistem menampilkan laporan mengenai data lokasi penjualan yang ada	
<b>Alur Normal</b>		
Aktor	Sistem	
1. Administrator memilih menu laporan	2. Sistem menampilkan data laporan mengenai lokasi penjualan	
<b>Alur Alternatif</b>		
-		



**Gambar 3.10 Diagram Aktivitas UC005**

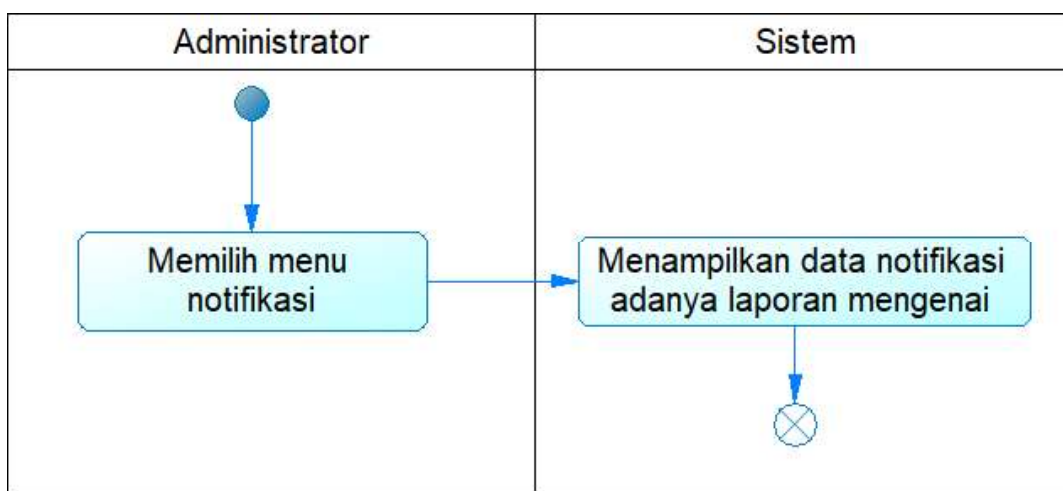
### 3.1.3.3.6. Kasus Penggunaan Melihat Notifikasi

Kasus penggunaan melihat notifikasi memungkinkan administrator untuk melihat pemberitahuan tentang adanya

laporan mengenai lokasi penjualan yang masuk. Spesifikasi kasus penggunaan dan diagram aktivitas untuk kasus penggunaan ini dapat dilihat pada Tabel 3.8 dan Gambar 3.11.

**Tabel 3.8 Spesifikasi Kasus Penggunaan Melihat Notifikasi**

<b>Nama</b>	Melihat Notifikasi
<b>Nomor</b>	UC006
<b>Deskripsi</b>	Administrator dapat melihat notifikasi apabila ada laporan mengenai lokasi penjualan yang masuk
<b>Aktor</b>	Administrator
<b>Kondisi Awal</b>	Data notifikasi adanya laporan mengenai lokasi penjualan telah tersimpan di <i>database</i>
<b>Kondisi Akhir</b>	Sistem menampilkan notifikasi adanya laporan mengenai data lokasi penjualan yang masuk
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator memilih menu notifikasi	2. Sistem menampilkan data notifikasi adanya laporan mengenai lokasi penjualan yang masuk
<b>Alur Alternatif</b>	
-	



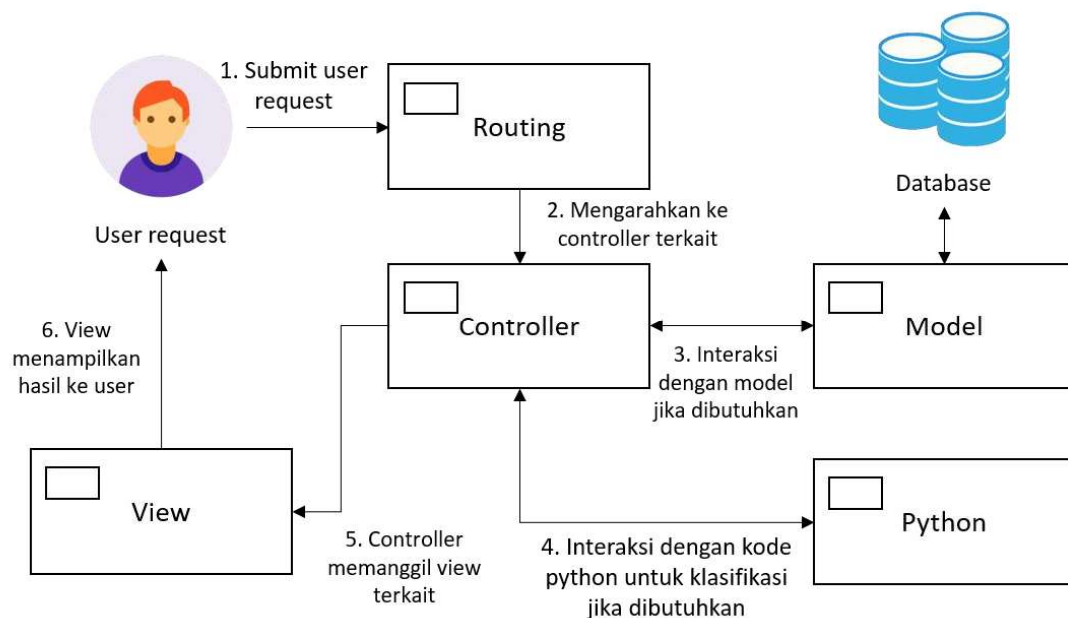
**Gambar 3.11 Diagram Aktivitas Melihat Notifikasi**

## 3.2. Perancangan

Pada subbab perancangan akan dijelaskan mengenai arsitektur sistem yang digunakan, perancangan diagram kelas, perancangan basis data, dan perancangan antarmuka.

### 3.2.1. Perancangan Arsitektur Sistem

Arsitektur sistem yang digunakan pada platform web pada perangkat lunak ini menggunakan kerangka kerja (*framework*) Laravel dengan pola perancangan *Model-View-Controller* (MVC). Ilustrasi arsitektur sistem dapat dilihat pada Gambar 3.12.



**Gambar 3.12 Ilustrasi Arsitektur Sistem Web**

Ketika client melakukan request data, Routing akan menerima request tersebut dan mengarahkan request tersebut kepada controller yang terkait dengan request yang diberikan. Controller akan berinteraksi dengan model yang dibutuhkan, dan model akan melakukan query terhadap database. Model mengembalikan data yang dibutuhkan kepada controller. Controller juga akan memanggil kode untuk proses klasifikasi data sampel yang ditulis dalam bahasa Python lalu kode proses klasifikasi tersebut akan dikembalikan ke php untuk menjadi hasil dari proses klasifikasi. Kemudian controller memanggil view


terkait, untuk menampilkan hasil yang diminta oleh user melalui browser.

### 3.2.2. Perancangan Antarmuka Pengguna

Pada bagian ini dijelaskan mengenai rancangan tampilan antarmuka pengguna dari sistem. Antarmuka pengguna yang berhubungan langsung dengan aktor, harus memiliki tampilan yang menarik bagi penggunanya. Aplikasi ini menggunakan antarmuka yang indah dilihat. Pengguna dapat mengakses menggunakan semua *web* browser dan hanya menggunakan keyboard dan mouse. Aplikasi ini memiliki satu hak akses yaitu hak akses untuk Administrator. Berikut akan dijelaskan masing rancangan antarmuka.

#### 3.2.2.1. Rancangan Antarmuka Melihat Lokasi Penjualan

Halaman melihat lokasi penjualan akan digunakan pada kasus penggunaan melihat lokasi penjualan. Halaman ini berisi peta yang didalamnya terdapat *marker-marker* penanda lokasi penjualan. Rancangan dan penjelasan terkait antarmuka menambah data sampel dapat dilihat pada Gambar 3.13 dan Tabel 3.9.

NAMA APLIKASI	ADMIN ▾ 					
Beranda	Beranda					
Tambah Data Sample	<table border="1"> <thead> <tr> <th>Warna Marker</th> <th>Keterangan</th> </tr> </thead> <tbody> <tr> <td>Merah</td> <td>Terindikasi boraks</td> </tr> </tbody> </table>		Warna Marker	Keterangan	Merah	Terindikasi boraks
Warna Marker	Keterangan					
Merah	Terindikasi boraks					
Lokasi Penjualan	<div style="border: 1px solid black; padding: 20px; text-align: center;"> <p>Google maps</p> </div>					
Bakso						
Tahu						
Laporan						

**Gambar 3.13 Rancangan Antarmuka Melihat Lokasi Penjualan**

**Tabel 3.9 Penjelasan Antarmuka Melihat Lokasi Penjualan**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>lokasiPenjualan Map</i>	<i>GoogleMap</i>	Menampilkan lokasi penjualan	-
2	<i>tabelPenjelasan Marker</i>	<i>Table</i>	Menampilkan keterangan warna marker pada peta	-

### 3.2.2.2. Rancangan Antarmuka Mengelola Lokasi Penjualan

Halaman melihat lokasi penjualan akan digunakan pada kasus penggunaan melihat lokasi penjualan. Halaman ini berisi daftar lokasi penjualan sesuai dengan jenis sampel yang dipilih sebelumnya. Pemilihan tombol lihat data sampel akan menuju ke halaman melihat data sampel, sedangkan tombol ubah akan memunculkan modal untuk mengubah lokasi penjualan dan tombol hapus akan memunculkan modal konfirmasi untuk menghapus data lokasi penjualan. Rancangan dan penjelasan terkait antarmuka melihat pengajuan petani mitra dapat dilihat pada Gambar 3.14 dan Tabel 3.10.

**Tabel 3.10 Penjelasan Antarmuka Mengelola Lokasi Penjualan**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	<i>tabelLokasiPenjualan</i>	<i>Table</i>	Menampilkan data lokasi penjualan yang telah masuk	-
2	<i>lihatDataSampelButton</i>	<i>Button</i>	Menampilkan halaman	<i>Button Clicked</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
			untuk melihat data sampel	
3	<i>ubahButton</i>	<i>Button</i>	Menampilkan modal untuk mengubah data lokasi penjualan	<i>Button Clicked</i>
4	<i>hapusButton</i>	<i>Button</i>	Menampilkan modal konfirmasi untuk manghapus data lokasi penjualan	

NAMA APLIKASI	ADMIN ▾																			
Beranda	<p>Lokasi Penjualan Bakso <span style="float: right;">Lokasi Penjualan / Bakso</span></p> <table border="1"> <thead> <tr> <th>No.</th> <th>Nama Lokasi</th> <th>Tanggal Pengambilan Lokasi</th> <th>Data Sampel</th> <th>Kelas</th> <th>Opsi</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Ps. Pucang</td> <td>18-11-2019</td> <td>Ada</td> <td>Bebas Boraks</td> <td> <span style="background-color: #28a745; color: white; padding: 2px 5px;">Lihat Data Sampel</span> <span style="background-color: #ffc107; color: white; padding: 2px 5px;">Ubah</span> <span style="background-color: #dc3545; color: white; padding: 2px 5px;">Hapus</span> </td> </tr> <tr> <td>2.</td> <td>Ps. Keputih</td> <td>18-11-2019</td> <td>Ada</td> <td>Terindikasi Boraks</td> <td> <span style="background-color: #28a745; color: white; padding: 2px 5px;">Lihat Data Sampel</span> <span style="background-color: #ffc107; color: white; padding: 2px 5px;">Ubah</span> <span style="background-color: #dc3545; color: white; padding: 2px 5px;">Hapus</span> </td> </tr> </tbody> </table>		No.	Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sampel	Kelas	Opsi	1.	Ps. Pucang	18-11-2019	Ada	Bebas Boraks	<span style="background-color: #28a745; color: white; padding: 2px 5px;">Lihat Data Sampel</span> <span style="background-color: #ffc107; color: white; padding: 2px 5px;">Ubah</span> <span style="background-color: #dc3545; color: white; padding: 2px 5px;">Hapus</span>	2.	Ps. Keputih	18-11-2019	Ada	Terindikasi Boraks	<span style="background-color: #28a745; color: white; padding: 2px 5px;">Lihat Data Sampel</span> <span style="background-color: #ffc107; color: white; padding: 2px 5px;">Ubah</span> <span style="background-color: #dc3545; color: white; padding: 2px 5px;">Hapus</span>
No.			Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sampel	Kelas	Opsi													
1.			Ps. Pucang	18-11-2019	Ada	Bebas Boraks	<span style="background-color: #28a745; color: white; padding: 2px 5px;">Lihat Data Sampel</span> <span style="background-color: #ffc107; color: white; padding: 2px 5px;">Ubah</span> <span style="background-color: #dc3545; color: white; padding: 2px 5px;">Hapus</span>													
2.			Ps. Keputih	18-11-2019	Ada	Terindikasi Boraks	<span style="background-color: #28a745; color: white; padding: 2px 5px;">Lihat Data Sampel</span> <span style="background-color: #ffc107; color: white; padding: 2px 5px;">Ubah</span> <span style="background-color: #dc3545; color: white; padding: 2px 5px;">Hapus</span>													
Tambah Data Sample																				
Lokasi Penjualan																				
Bakso																				
Tahu																				
Laporan																				

**Gambar 3.14 Rancangan Halaman Antarmuka Mengelola Data Lokasi Penjualan**

Pada Gambar 3.15 dan Tabel 3.11 merupakan rancangan dan penjelasan antarmuka untuk mengubah data lokasi penjualan

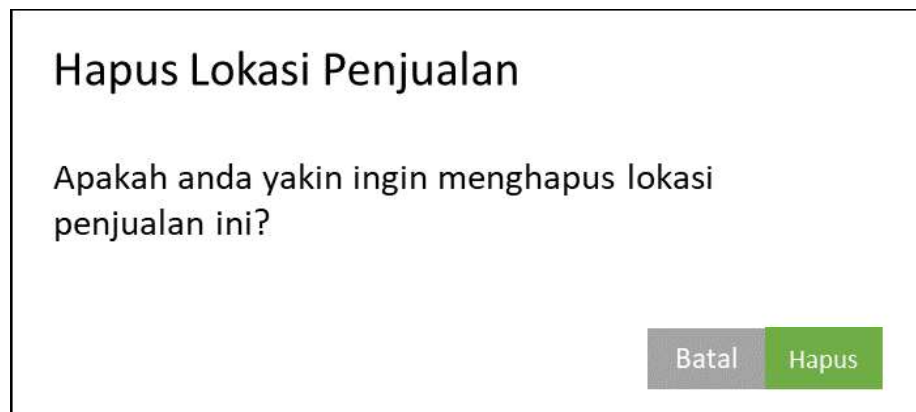
**Gambar 3.15 Rancangan Halaman Antarmuka Mengubah Data Lokasi Penjualan**

**Tabel 3.11 Penjelasan Antarmuka Mengubah Data Lokasi Penjualan**

<b>No</b>	<b>Nama Atribut Antarmuka</b>	<b>Jenis Atribut</b>	<b>Kegunaan</b>	<b>Jenis Masukan / Keluaran</b>
1	<i>namaLokasiPenjualan-Textfield</i>	<i>Textfield</i>	Field mengisi nama lokasi penjualan	<i>String</i>
2	<i>statusLokasiPenjualanDropdown</i>	<i>Dropdown</i>	Field untuk memilih status lokasi penjualan	<i>Dropdown Selected</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
3	<i>simpanButton</i>	<i>Button</i>	Menyimpan data yang dimasukkan	<i>Button Clicked</i>
4	<i>batalButton</i>	<i>Button</i>	Menutup modal untuk mengubah data lokasi penjualan	<i>Button Clicked</i>

Pada Gambar 3.16 dan Tabel 3.12 merupakan rancangan dan penjelasan antarmuka untuk mengubah data lokasi penjualan



**Gambar 3.16 Rancangan Antarmuka Menghapus Lokasi Penjualan**

**Tabel 3.12 Penjelasan Antarmuka Menghapus Lokasi Kegiatan**


No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>hapusButton</i>	<i>Button</i>	Menghapus data yang dimasukkan	<i>Button Clicked</i>



No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
2	<i>batalButton</i>	<i>Button</i>	Menutup modal untuk mengubah data lokasi penjualan	<i>Button Clicked</i>

### 3.2.2.3. Rancangan Antarmuka Menambah Data Sampel

Halaman menambah data sampel akan digunakan pada kasus penggunaan menambah data sampel. Halaman ini berisi form yang terdiri dari field-field yang harus diisi saat menambah data sampel. Rancangan dan penjelasan terkait antarmuka menambah data sampel dapat dilihat pada Gambar 3.17 dan Tabel 3.13.

NAMA APLIKASI	ADMIN ▾ 
Beranda	<p><b>Menambah Data Sample</b></p> <p>Jenis sample</p> <input type="text" value="Bakso"/> ▾
Tambah Data Sample	
Lokasi Penjualan	
Bakso	
Tahu	
Laporan	
	<p>File</p> <input type="text" value="Pilih File"/>
	<p><input type="button" value="Simpan"/> <input type="button" value="Batal"/></p>

**Gambar 3.17 Rancangan Antarmuka Menambah Data Sampel**

**Tabel 3.13 Penjelasan Antarmuka Menambah Data Sampel**


No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
----	------------------------	---------------	----------	--------------------------

1	<i>jenisSampelDropDown</i>	<i>DropDown</i>	Field untuk memilih jenis sampel	<i>DropDown Selected</i>
2	<i>namaLokasiPenjualanDropDown</i>	<i>DropDown</i>	Field untuk memilih nama lokasi penjualan	<i>DropDown Selected</i>
3	<i>tanggalPengambilanSampelTextfield</i>	<i>DropDown</i>	Field untuk mengisi tanggal pengambilan data	<i>String</i>
4	<i>fileDataSampelField</i>	<i>FileField</i>	Field untuk memilih file data sampel	<i>Field</i>
5	<i>simpanButton</i>	<i>Button</i>	Menyimpan data yang dimasukkan	<i>Button Clicked</i>
6	<i>batalButton</i>	<i>Button</i>	Menutup modal untuk membatalkan penambahan data	<i>Button Clicked</i>


#### 3.2.2.4. Rancangan Antarmuka Mengelola Data Sampel

Antarmuka mengelola data sampel terdiri dari beberapa antarmuka, yaitu antarmuka halaman melihat data sampel, melihat peta lokasi penjualan, dialog konfirmasi menghapus data sampel, dan dialog mengubah data sampel. Antarmuka ini akan digunakan pada kasus penggunaan mengelola data sampel.

Antarmuka melihat rancangan dan penjelasan terkait antarmuka melihat data sampel dapat dilihat pada Gambar 3.18, Gambar 3.19, dan Tabel 3.14.

<b>NAMA APLIKASI</b>		<b>ADMIN</b> ▾																					
Beranda	Lokasi Penjualan / Bakso																						
Tambah Data Sample	Lokasi Penjualan   Bakso																						
Lokasi Penjualan	<a href="#">Lihat Lokasi Penjualan</a>	<a href="#">Lihat Foto Lokasi Penjualan</a>	<a href="#">Ubah Data Sample</a>																				
Bakso	<a href="#">Hapus Data Sample</a>																						
Tahu	<a href="#">Lokasi Penjualan</a>																						
Laporan	Latitude:	Tanggal pengambilan lokasi:	Tanggal pengambilan sampel:																				
	Longitude:	Kelas Klasifikasi:	Rank Klasifikasi:																				
	Score Klasifikasi:																						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">No.</th> <th style="width: 10%;">MQ2_LPG</th> <th style="width: 10%;">MQ2_CO</th> <th style="width: 10%;">MQ2_SM OKE</th> <th style="width: 10%;">MQ2_ALC OHOL</th> <th style="width: 10%;">MQ2_CH4</th> <th style="width: 10%;">MQ2_H2</th> <th style="width: 10%;">MQ2_PRO PANE</th> <th style="width: 10%;">MQ3_ALC OHOL</th> <th style="width: 10%;">MQ3_BEN ZINE</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	No.	MQ2_LPG	MQ2_CO	MQ2_SM OKE	MQ2_ALC OHOL	MQ2_CH4	MQ2_H2	MQ2_PRO PANE	MQ3_ALC OHOL	MQ3_BEN ZINE											<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Grafik sensor MQ2</div>	
No.	MQ2_LPG	MQ2_CO	MQ2_SM OKE	MQ2_ALC OHOL	MQ2_CH4	MQ2_H2	MQ2_PRO PANE	MQ3_ALC OHOL	MQ3_BEN ZINE														
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Grafik sensor MQ3</div>																						

**Gambar 3.18 Rancangan Antarmuka Melihat Data Sampel - 1**

NAMA APLIKASI		ADMIN ▾	
Beranda	Grafik sensor MQ4	Grafik sensor MQ5	
Tambah Data Sample	Grafik sensor MQ6	Grafik sensor MQ7	
Lokasi Penjualan	Grafik sensor MQ8	Grafik sensor MQ9	
Bakso			
Tahu			
Laporan			
			Grafik sensor MQ135

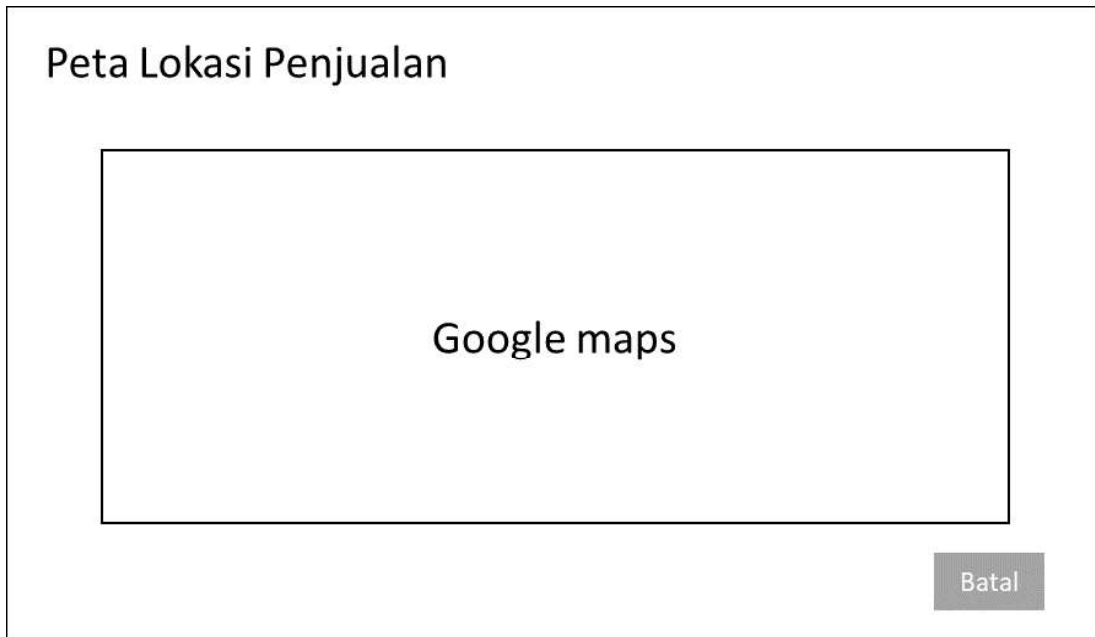
**Gambar 3.19 Rancangan Antarmuka Melihat Data Sampel - 2**

**Tabel 3.14 Penjelasan Antarmuka Melihat Data Sampel**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>buttonLihat LokasiPenjualan</i>	<i>Button</i>	Memunculkan modal untuk menampilkan peta lokasi penjualan	<i>Button Clicked</i>
2	<i>buttonLihat FotoLokasi Penjualan</i>	<i>Button</i>	Memunculkan modal untuk menampilkan foto lokasi penjualan	<i>Button Clicked</i>
3	<i>buttonHapus DataSampel</i>	<i>Button</i>	Memunculkan modal konfirmasi penghapusan data sampel	<i>Button Clicked</i>
4	<i>buttonUbah DataSampel</i>	<i>Button</i>	Memunculkan modal konfirmasi pengubahan data sampel	<i>Button Clicked</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
5	<i>tabelNilaiSensor</i>	<i>Table</i>	Menampilkan nilai sensor dari data sampel yang telah diunggah	-
6	<i>grafikSensorMQ2</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ2 enose pada grafik garis	-
7	<i>grafikSensorMQ3</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ3 enose pada grafik garis	-
8	<i>grafikSensorMQ4</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ4 enose pada grafik garis	-
9	<i>grafikSensorMQ5</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ5 enose pada grafik garis	-
10	<i>grafikSensorMQ6</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ6 enose pada grafik garis	-
11	<i>grafikSensorMQ7</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ7 enose pada grafik garis	-
12	<i>grafikSensorMQ8</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ8 enose pada grafik garis	-
13	<i>grafikSensorMQ9</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ9 enose pada grafik garis	-
14	<i>grafikSensorMQ135</i>	<i>Grafik</i>	Memvisualisasikan nilai sensor MQ135 pada grafik garis	-

Pada Gambar 3.20 dan Tabel 3.14 merupakan merupakan rancangan dan penjelasan antarmuka untuk modal melihat peta lokasi penjualan.

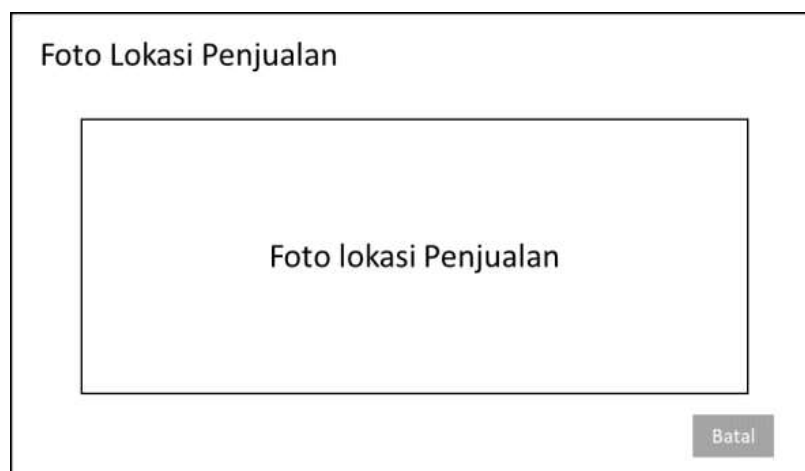


**Gambar 3.20 Perancangan Antarmuka Modal Melihat Peta Lokasi Penjualan**

**Tabel 3.15 Penjelasan Antarmuka Modal Melihat Peta Lokasi Penjualan**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>buttonTutupModal LihatPetaLokasi Penjualan</i>	<i>Button</i>	Menutup modal untuk menampilkan peta lokasi penjualan	<i>Button Clicked</i>
2	<i>mapsLokasi Penjualan</i>	<i>Google Maps</i>	Menampilkan google maps untuk lokasi penjualan	-

Pada Gambar 3.21 dan Tabel 3.16 merupakan merupakan rancangan dan penjelasan antarmuka untuk modal melihat foto lokasi penjualan.



**Gambar 3.21 Perancangan Antarmuka Melihat Modal Foto Lokasi Penjualan**

**Tabel 3.16 Penjelasan Antarmuka Modal Melihat Foto Lokasi Penjualan**

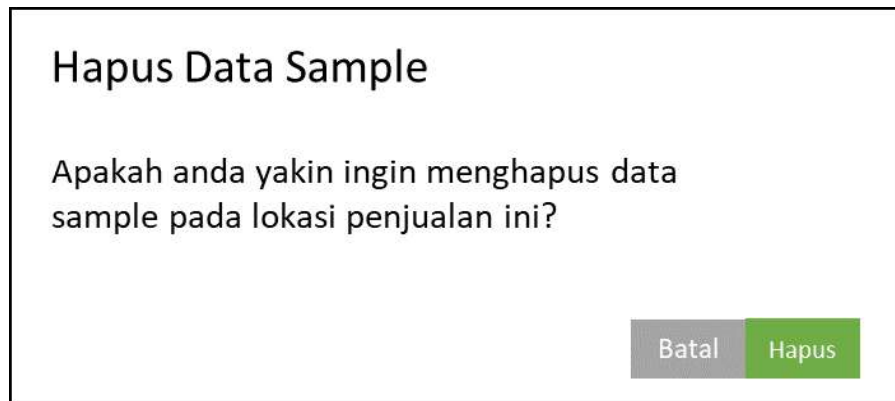
No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>buttonTutupModal LihatFotoLokasi Penjualan</i>	<i>Button</i>	Menutup modal untuk menampilkan foto lokasi penjualan	<i>Button Clicked</i>

Pada Gambar 3.22 dan Tabel 3.17 merupakan merupakan rancangan dan penjelasan antarmuka untuk dialog konfirmasi penghapusan data sampel.

**Tabel 3.17 Penjelasan Antarmuka Dialog Konfirmasi Penghapusan Data Sampel**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>buttonTutupModal HapusDataSampel</i>	<i>Button</i>	Menutup modal untuk menghapus data sampel	<i>Button Clicked</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
2	<i>buttonKonfirmasi PenhapusanData Sampel</i>	<i>Button</i>	Menghapus data sampel	<i>Button Clicked</i>



**Gambar 3.22 Perancangan Antarmuka Dialog Konfirmasi Penghapusan Data Sampel**

Pada Gambar 3.23 dan Tabel 3.18 merupakan merupakan rancangan dan penjelasan antarmuka untuk dialog konfirmasi pengubahan data sampel.



**Gambar 3.23 Perancangan Antarmuka Dialog Konfirmasi Pengubahan Data Sampel**



**Tabel 3.18 Penjelasan Antarmuka Dialog Konfirmasi Perubahan Data Sampel**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	<i>buttonTutupModal UbahDataSampel</i>	<i>Button</i>	Menutup modal untuk mengubah	<i>Button Clicked</i>
2	<i>buttonKonfirmasi PerubahanData Sampel</i>	<i>Button</i>	Menampilkan modal formulir perubahan data sampel	<i>Button Clicked</i>

Pada Gambar 3.24 dan Tabel 3.19 merupakan merupakan rancangan dan penjelasan antarmuka untuk modal perubahan data sampel.

**Gambar 3.24 Perancangan Antarmuka Modal Perubahan Data Sampel**


**Tabel 3.19 Penjelasan Antarmuka Modal Perubahan Data Sampel**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	<i>buttonTutupModal Formulir UbahDataSampel</i>	<i>Button</i>	Menutup modal untuk mengubah	<i>Button Clicked</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
2	<i>buttonUbah DataSampel</i>	<i>Button</i>	Menampilkan modal formulir perubahan data sampel	<i>Button Clicked</i>
3	<i>tanggalPengambilanSampelText field</i>	<i>Dropdown</i>	Field untuk mengisi tanggal pengambilan data	<i>String</i>
4	<i>fileDataSampel Field</i>	<i>FileField</i>	Field untuk memilih file data sampel	<i>Field</i>

### 3.2.2.5. Rancangan Antarmuka Melihat Laporan Mengenai Lokasi Penjualan

Antarmuka melihat laporan mengenai lokasi penjualan akan digunakan pada kasus penggunaan melihat lokasi penjualan. Pada Gambar 3.25 dan Tabel 3.20 merupakan rancangan dan penjelasan antarmuka untuk melihat laporan.

NAMA APLIKASI	ADMIN ▾ 													
Beranda	<p>Data Laporan</p> <table border="1"> <thead> <tr> <th>No.</th> <th>Nama Pelapor</th> <th>Tanggal laporan</th> <th>Isi Laporan</th> <th>Opsi</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Ps. Pucang</td> <td>18-11-2019</td> <td>Lokasi Ini Sudah Tutup</td> <td><a href="#">Lihat Lokasi</a></td> </tr> </tbody> </table>				No.	Nama Pelapor	Tanggal laporan	Isi Laporan	Opsi	1.	Ps. Pucang	18-11-2019	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>
No.					Nama Pelapor	Tanggal laporan	Isi Laporan	Opsi						
1.					Ps. Pucang	18-11-2019	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>						
Tambah Data Sample														
Lokasi Penjualan														
Bakso														
Tahu														
Laporan														

**Gambar 3.25 Perancangan Antarmuka Melihat Laporan Mengenai Lokasi Penjualan**

**Tabel 3.20 Penjelasan Antarmuka Melihat Laporan Mengenai Lokasi Penjualan**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>tabelLaporan</i>	<i>Table</i>	Menampilkan data laporan mengenai lokasi penjualan yang telah masuk	-
2	<i>lihatLokasi PenjualanButton</i>	<i>Button</i>	Menampilkan halaman untuk melihat modal lokasi penjualan	<i>Button Clicked</i>

### 3.2.2.6. Rancangan Antarmuka Melihat Notifikasi

Antarmuka melihat notifikasi akan digunakan pada kasus penggunaan melihat notifikasi. Pada Gambar 3.26 dan Tabel 3.21 merupakan merupakan rancangan dan penjelasan antarmuka untuk melihat notifikasi.

Notifikasi
Laporan baru masuk! Isi laporan {tanggal masuk laporan}
Laporan baru masuk! Isi laporan {tanggal masuk laporan}
Laporan baru masuk! Isi laporan {tanggal masuk laporan}
Bersihkan Notifikasi

**Gambar 3.26 Perancangan Antarmuka Melihat Notifikasi**

**Tabel 3.21 Penjelasan Antarmuka Melihat Notifikasi**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>listNotifikasi</i>	<i>List</i>	Menampilkan data notifikasi adanya laporan yang masuk	-
2	<i>buttonBersihkan Notifikasi</i>	<i>Button</i>	Membersihkan daftar notifikasi yang masuk	<i>Button Clicked</i>

### 3.2.3. Perancangan Data

Pada subbab ini menjelaskan perancangan data input yang diambil dengan *electronic nose* dan perancangan basis data.

#### 3.2.3.1. Perancangan Data Input

Data input dari tiap-tiap sampel, akan didapatkan dengan cara menggunakan sebuah *electronic nose* bernama “Enosika”[28][29]. Pengambilan sampel akan dilakukan selama 20 menit dengan rincian 5 menit pengambilan udara bebas, 10 menit pengambilan udara pada gelas kimia berisi sampel, dan 5 menit terakhir pengambilan udara bebas kembali dengan menggunakan aplikasi Arduino. Hal ini dilakukan agar perbedaan antara perbedaan nilai sensor yang diambil pada udara bebas dan udara sekitar sampel terlihat jelas. Aplikasi Arduino akan merekam nilai-nilai tegangan yang didapatkan oleh masing-masing sensor ketika reseptor dikenai molekul bau dan akan menampilkan ke dalam terminal pada aplikasi tersebut. Lalu, nilai-nilai yang ada pada terminal tersebut akan *export* dalam format csv sebagai data masukan.

Data masukan (*input*) adalah *dataset* berupa file berekstensi csv yang berisikan nilai-nilai yang didapatkan dari proses pengambilan sampel dengan mesin *electronic nose* dengan sembilan sensor di dalamnya, yaitu sensor MQ2, MQ3, MQ4, MQ5, MQ6, MQ7, MQ8, MQ9, dan MQ135. Terdapat 48 kolom parameter sensor yang akan menjadi fitur dalam proses latihan dan

klasifikasi data. Kolom-kolom tersebut dapat dilihat pada Tabel 3.22.



**Gambar 3.27 Electronic Nose Enosika**

**Tabel 3.22 Kolom Jenis Bau Sebagai Parameter Sensor Untuk Nilai Data Input**

No.	Nama Kolom	No.	Nama Kolom
1	MQ2_LPG	25	MQ6_ALCOHOL
2	MQ2_CO	26	MQ6_CH4
3	MQ2_SMOKE	27	MQ6_CO
4	MQ2_ALCOHOL	28	MQ6_H2
5	MQ2_CH4	29	MQ6_LPG
6	MQ2_H2	30	MQ7_ALCOHOL
7	MQ2_PROPANE	31	MQ7_CH4
8	MQ3_ALCOHOL	32	MQ7_CO
9	MQ3_BENZINE	33	MQ7_H2
10	MQ3_CH4	34	MQ7_LPG
11	MQ3_CO	35	MQ8_ALCOHOL
12	MQ3_HEXANE	36	MQ8_CH4
13	MQ3_LPG	37	MQ8_CO
14	MQ4_ALCOHOL	38	MQ8_H2
15	MQ4_CH4	39	MQ8_LPG
16	MQ4_CO	40	MQ9_CH4
17	MQ4_H2	41	MQ9_CO
18	MQ4_LPG	42	MQ9_LPG
19	MQ4_SMOKE	43	MQ135_ACETON
20	MQ5_ALCOHOL	44	MQ135_ALCOHOL

No.	Nama Kolom	No.	Nama Kolom
21	MQ5_CH4	45	MQ135_CO
22	MQ5_CO	46	MQ135_CO2
23	MQ5_H2	47	MQ135_NH4
24	MQ5_LPG	48	MQ135_TOLUOL

Kolom-kolom di atas merupakan parameter-parameter yang dimiliki pada sensor-sensor pada *electronic nose* yang digunakan. Ketika melakukan proses perekaman, *electronic nose* akan mengubah bau yang didapatkan dari sampel dan akan mengubahnya menjadi nilai-nilai tegangan yang diterima oleh reseptor-reseptor pada sensor. Nilai-nilai pada kolom-kolom di atas yang akan digunakan sebagai data training maupun data test pada proses klasifikasi sampel. Contoh nilai-nilai tegangan pada setiap reseptor sensor tersebut, dapat dilihat pada Gambar 3.28, Gambar 3.29, Gambar 3.30, Gambar 3.31, Gambar 3.32, dan Gambar 3.33.

Timestamp	Panah	MQ2_LPG	MQ2_CO	MQ2_SMOKE	MQ2_ALCOHOL	MQ2_CH4	MQ2_H2	MQ2_PROPANE
21:28.3	->	4.6	34.86	20.02	8.12	10.84	7.92	4.69
21:30.6	->	3.12	20.28	13.08	4.98	6.69	5.38	3.16
21:32.9	->	2.89	18.28	12.06	4.54	6.1	4.99	2.93
21:35.2	->	2.85	17.91	11.87	4.46	6	4.92	2.88
21:37.5	->	2.84	17.8	11.81	4.43	5.96	4.9	2.87
21:39.8	->	2.83	17.74	11.78	4.42	5.95	4.89	2.86
21:42.1	->	2.83	17.73	11.78	4.42	5.94	4.89	2.86
21:44.5	->	2.83	17.73	11.78	4.42	5.94	4.89	2.86
21:46.8	->	2.83	17.73	11.77	4.42	5.94	4.89	2.86
21:49.1	->	2.83	17.73	11.77	4.42	5.94	4.89	2.86
21:51.4	->	2.83	17.69	11.76	4.41	5.93	4.88	2.86
21:53.7	->	2.82	17.65	11.73	4.4	5.92	4.87	2.85
21:56.0	->	2.81	17.53	11.67	4.37	5.88	4.85	2.84
21:58.3	->	2.81	17.51	11.66	4.37	5.87	4.84	2.84

**Gambar 3.28 Contoh File CSV *Electronic Nose* Kolom 1 – 7**

MQ3_ALCOHOL	MQ3_BENZINE	MQ3_CH4	MQ3_CO	MQ3_HEXANE	MQ3_LPG
0	0	0	0.09	0.07	0.09
0	0	0	0.04	0.04	0.05
0	0	0	0.04	0.04	0.04
0	0	0	0.04	0.04	0.04
0	0	0	0.04	0.04	0.04
0	0	0	0.04	0.04	0.04
0	0	0	0.04	0.04	0.04
0	0	0	0.04	0.04	0.04
0	0	0	0.04	0.04	0.04

**Gambar 3.29 Contoh File CSV *Electronic Nose* Kolom 8 – 13**

MQ4_ALCOHOL	MQ4_CH4	MQ4_CO	MQ4_H2	MQ4_LPG	MQ4_SMOKE	MQ5_ALCOHOL	MQ5_CH4	MQ5_CO	MQ5_H2	MQ5_LPG
62.73	14.64	72.58	79.1	35.81	81.91	12.91	1.43	3.37	0.47	0.7
5.51	8.77	1.96	28.21	20.39	15.66	5.58	0.9	0.92	0.22	0.45
3.61	8.02	1.05	23.57	18.49	11.74	4.8	0.82	0.73	0.19	0.41
3.32	7.88	0.93	22.76	18.13	11.1	4.72	0.82	0.71	0.19	0.41
3.22	7.83	0.89	22.46	18	10.86	4.67	0.81	0.7	0.19	0.4
3.34	7.89	0.94	22.82	18.16	11.15	4.69	0.81	0.7	0.19	0.41
3.37	7.9	0.95	22.89	18.19	11.2	4.61	0.81	0.68	0.19	0.4
3.37	7.91	0.95	22.91	18.2	11.21	4.57	0.8	0.67	0.18	0.4
3.28	7.86	0.91	22.63	18.08	11	4.62	0.81	0.68	0.19	0.4
3.36	7.9	0.94	22.85	18.18	11.17	4.6	0.8	0.68	0.18	0.4
3.32	7.88	0.93	22.76	18.13	11.1	4.57	0.8	0.67	0.18	0.4
3.36	7.9	0.94	22.88	18.19	11.19	4.56	0.8	0.67	0.18	0.4
3.28	7.86	0.91	22.62	18.07	10.99	4.56	0.8	0.67	0.18	0.4
3.31	7.87	0.92	22.71	18.11	11.06	4.56	0.8	0.67	0.18	0.4

**Gambar 3.30 Contoh File CSV *Electronic Nose* Kolom 14 – 24**

MQ6_ALCOHOL	MQ6_CH4	MQ6_CO	MQ6_H2	MQ6_LPG	MQ7_ALCOHOL	MQ7_CH4
54.41	7.09	49.59	20.14	4.41	0.02	0.05
17.43	4.42	4.47	9.98	2.82	0	0.01
14.72	4.12	3.12	8.99	2.64	0	0.01
15.32	4.19	3.4	9.21	2.68	0	0
14.95	4.15	3.23	9.07	2.65	0	0
14.32	4.08	2.95	8.84	2.61	0	0
15.32	4.19	3.4	9.21	2.68	0	0
14.95	4.15	3.23	9.07	2.65	0	0
13.29	3.95	2.52	8.44	2.53	0	0
12.95	3.91	2.38	8.3	2.51	0	0
15.01	4.16	3.25	9.09	2.66	0	0
13.81	4.02	2.73	8.64	2.57	0	0
15.09	4.17	3.29	9.12	2.66	0	0
14.35	4.08	2.96	8.85	2.61	0	0

**Gambar 3.31 Contoh File CSV *Electronic Nose* Kolom 25 – 31**

MQ7_CO	MQ7_H2	MQ7_LPG	MQ8_ALCOHOL	MQ8_CH4	MQ8_CO	MQ8_H2	MQ8_LPG
0.7	0.81	0.01	0	0	0	0	0
0.53	0.63	0	0	0	0	0	0
0.5	0.6	0	0	0	0	0	0
0.5	0.6	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.5	0.59	0	0	0	0	0	0
0.49	0.59	0	0	0	0	0	0

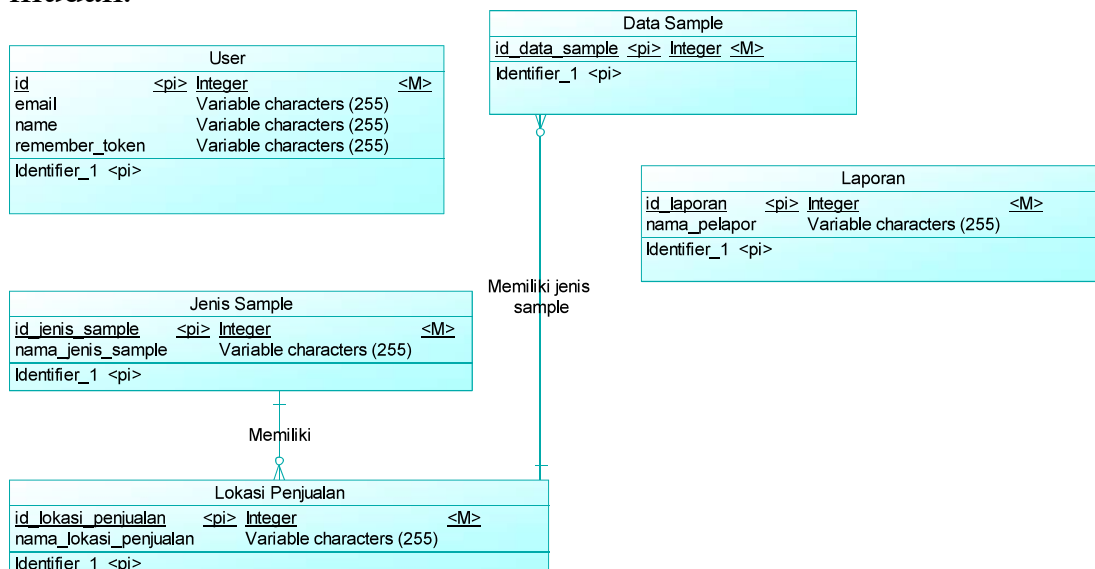
**Gambar 3.32 Contoh File CSV *Electronic Nose* Kolom 32 – 39**

MQ9_CH4	MQ9_CO	MQ9_LPG	MQ135_A CETON	MQ135_AL COHOL	MQ135_C O	MQ135_C O2	MQ135_N H4	MQ135_TOLU OL
151.53	33.22	65.18	1.3	3.54	13.2	6.91	9.24	1.63
50	12.94	26.69	0.4	1.17	3.32	2.54	3.87	0.49
38.85	10.44	21.79	0.11	0.34	0.73	0.84	1.48	0.13
24.76	7.12	15.16	0.07	0.24	0.45	0.6	1.1	0.09
27.81	7.86	16.65	0.07	0.23	0.43	0.58	1.07	0.08
4.94	1.81	4.14	0.04	0.12	0.2	0.33	0.66	0.04
11.16	3.61	7.98	0.24	0.74	1.87	1.67	2.69	0.3
115.39	26.35	52.34	0.64	1.82	5.76	3.79	5.47	0.79
69.59	17.14	34.84	0.25	0.76	1.95	1.72	2.76	0.31
24.18	6.97	14.88	0.08	0.26	0.52	0.66	1.2	0.1
88.74	21.07	42.36	0.09	0.3	0.62	0.75	1.34	0.11
28.44	8.01	16.95	0.07	0.22	0.42	0.57	1.06	0.08
117.27	26.71	53.03	0.24	0.72	1.81	1.64	2.64	0.29
73.66	17.99	36.47	1.19	3.27	11.94	6.43	8.67	1.49

**Gambar 3.33 Contoh File CSV *Electronic Nose* Kolom 40 – 48**

### 3.2.3.2. Perancangan Basis Data

Dalam membuat suatu aplikasi berupa sistem informasi, diperlukan sebuah analisis berupa perancangan basis data. MySQL dipilih menjadi aplikasi basis data karena penggunaannya yang mudah.

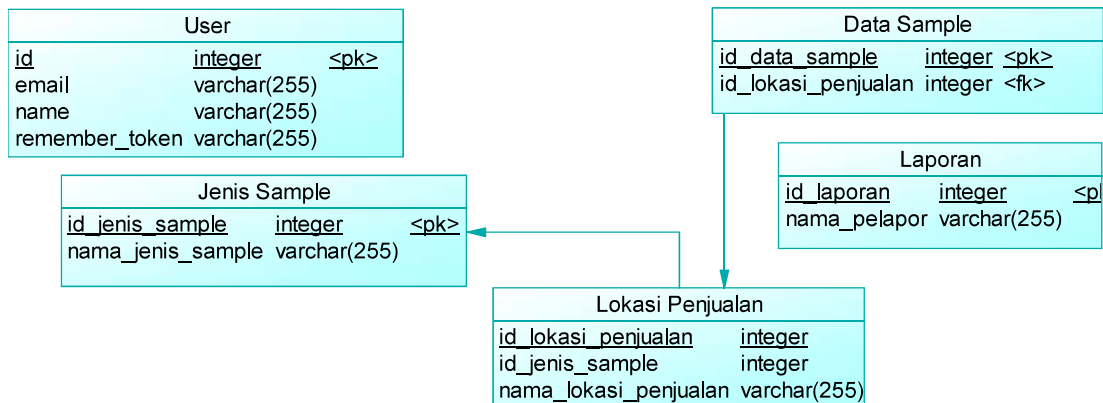


**Gambar 3.34 CDM Sistem**

Rancangan basis data ditampilkan dalam bentuk *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)*.



Secara umum, struktur CDM dan PDM dapat dilihat pada Gambar 3.34 dan Gambar 3.35. Struktur CDM dan PDM lebih detail disertakan dalam Lampiran 1 dan Lampiran 2.



**Gambar 3.35 PDM Sistem**

### 3.2.3.2.1. Tabel Users

Tabel users digunakan untuk menyimpan data pengguna berupa id, email, username, password, dan token. Tabel ini memiliki relasi dengan tabel lain yaitu password reset. Deskripsi rinci tabel users dapat dilihat pada Tabel 3.23.

**Tabel 3.23 Deskripsi Rinci Tabel Users**

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
id	Variabel untuk ID user.	Integer	Primary Key
name	Variabel untuk nama user.	Varchar(255)	
email	Variabel untuk email user.	Varchar(255)	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
username	Variabel untuk username user.	Varchar(255)	
password	Variabel untuk password user.	Varchar(255)	
remember_token	Variabel untuk menyimpan token autentikasi user.	Varchar(255)	Boleh Null

### 3.2.3.2.2. Tabel Jenis Sampel

Tabel jenis sampel digunakan untuk menyimpan jenis sampel yang akan diuji. Tabel ini menjadi tabel dasar pengisian lokasi penjualan. Deskripsi rinci tabel lokasi penjualan dapat dilihat pada Tabel 3.24.

**Tabel 3.24 Deskripsi Rinci Tabel Jenis Sampel**

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
id_jenis_sample	Variabel untuk ID jenis sampel	Integer	Primary Key
nama_jenis_sampel	Variabel untuk menyimpan nama jenis sampel	Varchar(255)	

### 3.2.3.2.3. Tabel Lokasi Penjualan

Tabel lokasi penjualan digunakan untuk menyimpan lokasi-lokasi penjualan yang sudah didapatkan. Tabel ini memiliki relasi dengan tabel data sampel dan jenis sampel. Deskripsi rinci tabel lokasi penjualan dapat dilihat pada Tabel 3.25.

**Tabel 3.25 Deskripsi Rinci Tabel Lokasi Penjualan**

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
id_lokasi_penjualan	Variabel untuk ID lokasi penjualan.	Integer	Primary Key
id_jenis_sampel	Variabel ID jenis sampel	Integer	Foreign Key
latitude_lokasi_penjualan	Variabel untuk latitude lokasi penjualan	Varchar(255)	
longitude_lokasi_penjualan	Variabel untuk longitude lokasi penjualan	Varchar(255)	
nama_lokasi_penjualan	Variabel untuk menyimpan nama lokasi penjualan	Varchar(255)	
tanggal_pengambilan_sampel	Variabel untuk menyimpan tanggal pengambilan sampel	Date	
flag_lokasi_penjualan	Variabel untuk menyimpan status apabila lokasi penjualan telah mempunyai data sampel	Integer	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
foto_lokasi_penjualan	Variabel untuk menyimpan alamat foto lokasi penjualan	Longtext	
status_klasifikasi	Variabel untuk menyimpan hasil kelas klasifikasi data sampel	Varchar(255)	
rank_klasifikasi	Variabel untuk menyimpan hasil rank klasifikasi data sampel	Varchar(255)	
score_klasifikasi	Variabel untuk menyimpan hasil score klasifikasi data sampel	Varchar(255)	
nama_file_data_sampel	Variabel untuk menyimpan nama file csv yang disimpan	Varchar(255)	
status_lokasi_penjualan	Variabel untuk menyimpan status lokasi penjualan	Varchar(255)	

#### **3.2.3.2.4. Tabel Data Sampel**

Tabel data sampel digunakan untuk menyimpan nilai dari tiap kolom-kolom data input dari sampel yang telah didapatkan dengan *electronic nose*. Tabel ini memiliki relasi dengan tabel lokasi penjualan. Deskripsi rinci tabel data sampel dapat dilihat pada Tabel 3.26.

Tabel 3.26 Deskripsi Rinci Tabel Data Sampel

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
id_data_sample	Variabel untuk ID data sampel.	Integer	Primary Key
id_lokasi_penjualan	Variabel untuk ID lokasi penjualan.	Integer	Foreign Key
mq2_lpg	Variabel untuk menyimpan nilai sensor MQ2_LPG	Double	
mq2_co	Variabel untuk menyimpan nilai sensor MQ2_CO	Double	
mq2_smoke	Variabel untuk menyimpan nilai sensor MQ2_SMOKE	Double	
mq2_alcohol	Variabel untuk menyimpan nilai sensor MQ2_ALCOHOL	Double	
mq2_ch4	Variabel untuk menyimpan nilai sensor MQ2_CH4	Double	
mq2_h2	Variabel untuk menyimpan nilai sensor MQ2_H2	Double	
mq2_propane	Variabel untuk menyimpan nilai sensor MQ2_PROPANE	Double	
mq3_alcohol	Variabel untuk menyimpan nilai	Double	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
	sensor MQ3_ALCOHOL		
mq3_benzine	Variabel untuk menyimpan nilai sensor MQ3_BENZINE	Double	
mq3_ch4	Variabel untuk menyimpan nilai sensor MQ3_CH4	Double	
mq3_co	Variabel untuk menyimpan nilai sensor MQ3_CO	Double	
mq3_hexane	Variabel untuk menyimpan nilai sensor MQ3_HEXANE	Double	
mq3_lpg	Variabel untuk menyimpan nilai sensor MQ3_LPG	Double	
mq4_alcohol	Variabel untuk menyimpan nilai sensor MQ4_ALCOHOL	Double	
mq4_ch4	Variabel untuk menyimpan nilai sensor MQ4_CH4	Double	
mq4_co	Variabel untuk menyimpan nilai sensor MQ4_CO	Double	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
mq4_h2	Variabel untuk menyimpan nilai sensor MQ4_H2	Double	
mq4_lpg	Variabel untuk menyimpan nilai sensor MQ4_LPG	Double	
mq4_smoke	Variabel untuk menyimpan nilai sensor MQ4_SMOKE	Double	
mq5_alcohol	Variabel untuk menyimpan nilai sensor MQ5_ALCOHOL	Double	
mq5_ch4	Variabel untuk menyimpan nilai sensor MQ5_CH4	Double	
mq5_co	Variabel untuk menyimpan nilai sensor MQ5_CO	Double	
mq5_h2	Variabel untuk menyimpan nilai sensor MQ5_H2	Double	
ferrormq5_lpg	Variabel untuk menyimpan nilai sensor MQ5_LPG	Double	
mq6_alcohol	Variabel untuk menyimpan nilai sensor MQ6_ALCOHOL	Double	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
mq6_ch4	Variabel untuk menyimpan nilai sensor MQ6_CH4	Double	
mq6_co	Variabel untuk menyimpan nilai sensor MQ6_CO	Double	
mq6_h2	Variabel untuk menyimpan nilai sensor MQ6_H2	Double	
mq6_lpg	Variabel untuk menyimpan nilai sensor MQ6_LPG	Double	
mq7_alcohol	Variabel untuk menyimpan nilai sensor MQ7_ALCOHOL	Double	
mq7_ch4	Variabel untuk menyimpan nilai sensor MQ7_CH4	Double	
mq7_co	Variabel untuk menyimpan nilai sensor MQ7_CO	Double	
mq7_h2	Variabel untuk menyimpan nilai sensor MQ7_H2	Double	
mq7_lpg	Variabel untuk menyimpan nilai sensor MQ7_LPG	Double	
mq8_alcohol	Variabel untuk menyimpan nilai sensor MQ8_ALCOHOL	Double	



<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
mq8_ch4	Variabel untuk menyimpan nilai sensor MQ8_CH4	Double	
mq8_co	Variabel untuk menyimpan nilai sensor MQ8_CO	Double	
mq8_h2	Variabel untuk menyimpan nilai sensor MQ8_H2	Double	
mq8_lpg	Variabel untuk menyimpan nilai sensor MQ8_LPG	Double	
mq9_ch4	Variabel untuk menyimpan nilai sensor MQ9_CH4	Double	
mq9_co	Variabel untuk menyimpan nilai sensor MQ9_CO	Double	
mq9_lpg	Variabel untuk menyimpan nilai sensor MQ9_LPG	Double	
mq135_aceton	Variabel untuk menyimpan nilai sensor MQ135_ACETON	Double	
mq135_alcohol	Variabel untuk menyimpan nilai sensor MQ135_ALCOHOL	Double	
mq135_co	Variabel untuk menyimpan nilai sensor MQ135_CO	Double	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
mq135_co2	Variabel untuk menyimpan nilai sensor MQ135_CO2	Double	
mq135_nh4	Variabel untuk menyimpan nilai sensor MQ135_NH4	Double	
mq135_toluol	Variabel untuk menyimpan nilai sensor MQ135_TOLUOL	Double	

### 3.2.3.2.5. Tabel Laporan

Tabel laporan digunakan untuk menyimpan laporan mengenai lokasi penjualan yang masuk. Tabel ini tidak memiliki relasi dengan tabel apapun. Deskripsi rinci tabel lokasi penjualan dapat dilihat pada Tabel 3.27.

**Tabel 3.27 Deskripsi Rinci Tabel Laporan**

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
id_laporan	Variabel untuk ID laporan	Integer	Primary Key
nama_pelapor	Variabel untuk menyimpan nama pelapor	Varchar(255)	
latitude_lokasi_penjualan	Variabel untuk latitude lokasi penjualan	Varchar(255)	
longitude_lokasi_penjualan	Variabel untuk longitude lokasi penjualan	Varchar(255)	

<b>Id Field</b>	<b>Deskripsi</b>	<b>Tipe &amp; length</b>	<b>Keterangan</b>
isi_laporan	Variabel untuk menyimpan isi laproan	Varchar(255)	
is_read	Variabel untuk menyimpan status apakah laporan telah terbaca atau belum	Interger	Default(0)

### 3.2.4. Perancangan Proses Klasifikasi Data

Pada subbab ini akan dijelaskan mengenai perancangan proses klasifikasi data sampel mulai dari proses latihan hingga proses klasifikasi kadar boraks yang terkandung pada data sampel.

#### 3.2.4.1. Proses Latihan Data

Proses latihan data pada tugas akhir ini akan dilakukan pada 81 sampel dari 40 buah bakso murni dari boraks dan 41 buah yang sudah diketahui adanya kandungan boraks di dalamnya. Data sampel dengan kesembilan puluh enam fitur tersebut akan dilatih dengan beberapa metode klasifikasi yang terdapat pada Tabel 3.28. Metode-metode tersebut masing-masing akan diperoleh hasil latihannya dan akan dipilih yang memiliki akurasi paling tinggi.

**Tabel 3.28 Metode Klasifikasi yang Digunakan**

<b>No</b>	<b>Metode Klasifikasi</b>
1.	Support Vector Machine
2.	K-Nearest Neighbor
3.	Multilayer Perceptron

Setelah diperoleh metode yang memiliki akurasi paling bagus dalam melatih data latih, maka proses pembuatan model klasifikasi akan dilakukan menggunakan metode tersebut.

### 3.2.4.2. Proses Klasifikasi Data

Setelah didapatkan model dari data latih dengan metode yang memiliki akurasi paling tinggi, maka data input pada sampel dari masing-masing lokasi penjualan akan diuji dan di klasifikasikan kadar boraksnya sesuai dengan model tersebut.

**Tabel 3.29 Deskripsi Atribut Keluaran Proses Klasifikasi**

No	Keluaran Pengolahan Data	Deskripsi
1.	Status Indikasi	Terdiri dari dua nilai, yaitu Terindikasi adanya formalin dan tidak terindikasi adanya formalin.
2.	Nilai probabilitas	Merupakan probabilitas dari hasil klasifikasi. Mempunyai rentang 1 – 100, semakin besar menandakan bahwa semakin positif sampel terindikasi mengandung boraks.

Di proses ini, keluaran yang dibutuhkan dari proses klasifikasi data dibentuk. Dalam Tugas Akhir ini, keluaran yang dibutuhkan adalah status sampel dan nilai probabilitas sampel. Penjelasan dari masing-masing keluaran dapat dilihat pada Tabel 3.29.

### 3.2.5. Perancangan API Endpoint

Pada Tugas Akhir ini juga akan dibuat *endpoint* yang akan digunakan pada aplikasi *platform* Android. Perancangan API *endpoint* pada Tugas Akhir ini dijelaskan pada Tabel 3.30.

**Tabel 3.30 Deskripsi Perancangan API Endpoint**

No	Endpoint	Method	Deskripsi
1	/lokasi-penjualan/get	GET	Menampilkan daftar lokasi penjualan

No	Endpoint	Method	Deskripsi
2	/lokasi-penjualan/store	POST	Menambahkan data lokasi penjualan
3	/laporan/store	POST	Menambahkan data laporan mengenai lokasi penjualan

### 3.2.5.1. Struktur JSON API Endpoint

API yang dirancang pada Tugas Akhir ini akan menggunakan JSON. Subbab ini akan menjelaskan tentang parameter-parameter pada proses *request* dan *response* yang akan digunakan oleh API pada Tugas Akhir ini.

#### 3.2.5.1.1. API Endpoint Mengambil Lokasi Penjualan

Parameter-parameter yang dibutuhkan pada response endpoint /lokasi-penjualan/get dapat dilihat pada Tabel 3.31.

#### 3.2.5.1.2. API Endpoint Menambah Lokasi Penjualan

Parameter yang dibutuhkan saat proses request pada endpoint /lokasi-penjualan/store terdapat pada Tabel 3.32. Sedangkan, parameter responses pada API /lokasi-penjualan/store dapat dilihat pada Tabel 3.33.

**Tabel 3.31 Struktur JSON Response API Endpoint Mengambil Lokasi Penjualan**

No	Object Main JSON	Key Object	Tipe Data
1	code	-	Integer
2	message	-	String
3	data	id_lokasi_penjualan	Integer
		nama_lokasi_penjualan	String
		latitude_lokasi_penjualan	Double
		longitude_lokasi_penjualan	Double

	status_klasifikasi	String
	rank_klasifikasi	String
	score_klasifikasi	String
	foto_lokasi_penjualan	String
	id_jenis_sample	String
	created_at	String
	updated_at	String

**Tabel 3.32 Struktur JSON Request API Endpoint Menambah Lokasi Penjualan**

No	Nama Parameter	Keterangan	Tipe Data
1	nama_lokasi_penjualan	Sebagai field untuk mengirim nama lokasi penjualan ketika menambahkan lokasi penjualan baru	String
2	latitude_lokasi_penjualan	Sebagai field untuk mengirim latitude lokasi penjualan ketika menambahkan lokasi penjualan baru	Double
3	longitude_lokasi_penjualan	Sebagai field untuk mengirim longitude lokasi penjualan ketika menambahkan lokasi penjualan baru	Double
4	foto_lokasi_penjualan	Sebagai field untuk mengirim foto lokasi penjualan ketika menambahkan lokasi penjualan baru	String

5	id_jenis_sample	Sebagai field untuk mengirim password pengguna ketika login	Integer
---	-----------------	---	---------

**Tabel 3.33 Struktur JSON Response API Endpoint Menambah Lokasi Penjualan**

No	Object Main JSON	Key Object	Tipe Data
1	code	-	Integer
2	message	-	String
3	data	id_lokasi_penjualan	Integer
		nama_lokasi_penjualan	String
		latitude_lokasi_penjualan	Double
		longitude_lokasi_penjualan	Double
		foto_lokasi_penjualan	String
		id_jenis_sample	Integer
		created_at	String
		updated_at	String

### 3.2.5.1.3. API Endpoint Menambah Laporan

Parameter yang dibutuhkan saat proses request pada endpoint /laporan/store terdapat pada Tabel 3.34. Sedangkan, parameter responses pada API /laporan /store dapat dilihat pada Tabel 3.35.

**Tabel 3.34 Struktur JSON Request API Endpoint Menambah Laporan**

No	Nama Parameter	Keterangan	Tipe Data
1	nama_pelapor	Sebagai field untuk mengirim nama pelapor ketika menambahkan laporan baru	String

2	latitude_laporan	Sebagai field untuk mengirim latitude lokasi penjualan ketika menambahkan laporan baru	Double
3	longitude_laporan	Sebagai field untuk mengirim longitude lokasi penjualan ketika menambahkan laporan baru	Double
4	Isi_laporan	Sebagai field untuk mengirim isi dari laporan ketika menambahkan laporan baru	String

**Tabel 3.35 Struktur JSON Response API Menambah Laporan**

No	Object Main JSON	Key Object	Tipe Data
1	code	-	Integer
2	message	-	String
3	data	nama_pelapor	Integer
		isi_laporan	String
		latitude_lokasi_penjualan	Double
		longitude_lokasi_penjualan	Double



## **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program Tugas Akhir ini dan disertai dengan kode semu masing-masing fungsi utama.

### **4.1. Lingkungan Implementasi**

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4.1

**Tabel 4.1 Lingkungan Implementasi**

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i7-7700HQ CPU @ 2.80GHz (4 CPUs) , ~2.81GHz Memori: 16000 MB
Perangkat lunak	Sistem Operasi: Microsoft Windows 10 Pro Education 64-bit Perangkat Pengembang: Laragon, Visual Studio Code, HeidiSQL, Postman Perangkat Perancang Diagram: Sybase Power Designer 16 Perangkat Database: HeidiSQL

### **4.2. Implementasi Antarmuka Sistem**

Berikut ini akan dijelaskan mengenai implementasi antarmuka sistem yang sudah direalisasikan.

### 4.2.1. Halaman Melihat Lokasi Penjualan

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC001 yaitu melihat lokasi penjualan. Di dalam halaman ini terdapat peta yang menampilkan lokasi-lokasi penjualan yang sudah diuji dan diklasifikasikan kadar boraksnya. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.1 dan kode sumber fungsi melihat lokasi penjualan dapat dilihat pada Kode Sumber 4.1.


```

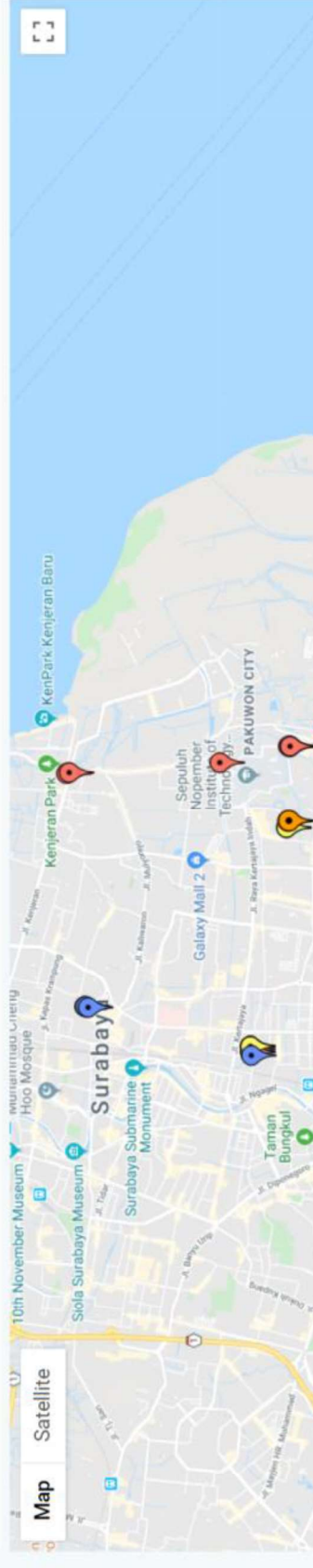
1. public function index()
2. {
3.     $laporan = Laporan::where('is_read', 0)-
       >orderBy('created_at', 'desc')->get();
4.     $daftarJenisSampel = JenisSampel::all();
5.     $lokasiPenjualan = LokasiPenjualan::where('flag_lok
       asi_penjualan', '=', 1)-
       >where('status_lokasi_penjualan' , 'Buka')
6.     ->get();
7.     $results = [
8.         'laporan' => $laporan,
9.         'pageTitle' => 'Dashboard',
10.        'daftarJenisSampel' => $daftarJenisSampel,
11.        'lokasiPenjualan' => $lokasiPenjualan,
12.    ];
13.    return view('home', $results);
14. }

```

#### Kode Sumber 4.1 Fungsi Melihat Lokasi Penjualan

Kode sumber implementasi menampilkan lokasi penjualan pada *marker-marker* Google maps dapat dilihat pada di lampiran. Pada halaman melihat lokasi penjualan, titik-titik lokasi penjualan yang telah melalui proses klasifikasi akan ditampilkan pada peta dengan warna-warna sesuai dengan status klasifikasinya. Penjelasan dari warna marker pada peta terdapat pada Tabel 4.1.

Warna lokasi:	Keterangan:
 Merah	Lokasi terindikasi menjual bakso berkandungan boraks
 Kuning	Lokasi tidak terindikasi menjual bakso berkandungan boraks
 Oranye	Lokasi terindikasi menjual tahu berkandungan formalin
 Biru	Lokasi tidak terindikasi menjual tahu berkandungan formalin
 Hijau	Data sampel tidak valid



**Gambar 4.1 Implementasi Antarmuka Melihat Lokasi Penjualan**

**Tabel 4.1 Penjelasan Warna Marker Pada Peta Melihat Lokasi Penjualan**

Warna	Keterangan
Merah	Lokasi terindikasi menjual bakso berkandungan boraks
Kuning	Lokasi tidak terindikasi menjual bakso berkandungan boraks
Oranye	Lokasi terindikasi menjual tahu berkandungan formalin
Biru	Lokasi tidak terindikasi menjual tahu berkandungan formalin
Hijau	Data tidak valid

Penjelasan Kode Sumber 4.1 dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Penjelasan Kode Sumber Melihat Lokasi Penjualan**

No. Baris	Kegunaan
3-4	Mengambil laporan belum terbaca dan jenis sampel yang akan ditampilkan pada semua tampilan aplikasi
5-6	Mengambil lokasi penjualan yang telah memiliki status klasifikasi untuk ditampilkan pada peta
7-12	Menyimpan data-data yang telah diambil ke dalam sebuah array untuk melakukan <i>passing</i> ke dalam view
13	Memanggil <i>view</i> untuk menampilkan data-data yang sudah disimpan

#### **4.2.2. Halaman Mengelola Lokasi Penjualan**

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC002 yaitu mengelola lokasi penjualan. Administrator dapat melihat, mengubah, dan menghapus data lokasi penjualan pada kasus penggunaan ini. Halaman antarmuka melihat lokasi penjualan dapat dilihat pada Gambar 4.2. dan Gambar 4.3. sedangkan, Kode Sumber 4.2 dan Tabel 4.3 merupakan fungsi dan penjelasan untuk menampilkan lokasi penjualan sesuai dengan jenis sampel yang telah dipilih pada menu sebelumnya.

No.	Nama Lokasi	Pengambilan Lokasi	Data Sample
1.	Pasar Keputih 1	05-01-2020	Ada
2.	Pasar Keputih 2	05-01-2020	Ada
3.	Superindo Arief Rahman Hakim - Bakso Urat Tanpa Merk	05-01-2020	Ada

**Gambar 4.2 Implementasi Antarmuka Melihat Daftar Lokasi Penjualan Kolom 1 – 4**

Kelas	Status	Opsi
Terindikasi ada boraks	Buka	<a href="#">Lihat Data Sample</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
Terindikasi ada boraks	Buka	<a href="#">Lihat Data Sample</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
Tidak terindikasi ada boraks	Buka	<a href="#">Lihat Data Sample</a> <a href="#">Ubah</a> <a href="#">Hapus</a>

**Gambar 4.3 Implementasi Antarmuka Melihat Daftar Lokasi Penjualan Kolom 5 – 7**

```

1. public function ambilLokasiPenjualan($sampel)
2. {
3.     $laporan = Laporan::where('is_read', 0)-
>orderBy('created_at', 'desc')->get();
4.     $daftarJenisSampel = JenisSampel::all();
5.     $jenisSampel = JenisSampel::where('nama_jenis_sampel', '=', $sampel)->first();
6.     if (is_null($jenisSampel)) {
7.         return redirect()->back()-
>with('danger', 'Jenis sampel tidak ditemukan');
8.     }
9.     $lokasiPenjualan = LokasiPenjualan::where('id_jenis_sample', '=', $jenisSampel->id_jenis_sample)->paginate(10);
10.    $results = [
11.        'laporan' => $laporan,
12.        'daftarJenisSampel' => $daftarJenisSampel,
13.        'pageTitle' => 'Lokasi Penjualan | ' . $jenisSampel->nama_jenis_sampel,
14.        'lokasiPenjualan' => $lokasiPenjualan,
15.        'jenisSampel' => $jenisSampel,
16.    ];
17.    return view('data-lokasi', $results);

```

18. }

### Kode Sumber 4.2 Fungsi Menampilkan Lokasi Penjualan Sesuai Jenis Sampel

**Tabel 4.3 Penjelasan Kode Sumber Menampilkan Lokasi Penjualan Sesuai Jenis Sampel**

No. Baris	Kegunaan
3-4	Mengambil laporan belum terbaca dan jenis sampel yang akan ditampilkan pada semua tampilan aplikasi
5-8	Melakukan pengecekan apabila jenis sampel yang diinginkan untuk ditampilkan tidak ada
9	Mengambil semua lokasi penjualan sesuai dengan jenis sampel yang telah dipilih
10-16	Menyimpan data-data yang telah diambil ke dalam sebuah array untuk melakukan <i>passing</i> ke dalam view
17	Memanggil <i>view</i> untuk menampilkan data-data yang sudah disimpan

Implementasi antarmuka untuk mengubah data lokasi penjualan dapat dilihat pada Gambar 4.4 dan fungsi untuk mengubah data lokasi penjualan dan penjelasannya dapat dilihat pada Kode Sumber 4.3. dan Tabel 4.4.



**Gambar 4.4 Implementasi Antarmuka Mengubah Data Lokasi Penjualan**

1. `public function` ubahLokasiPenjualan(Request \$request)

```

2. {
3.     try {
4.         $lokasiPenjualan = LokasiPenjualan::find($request->id_lokasi_penjualan);
5.         if(is_null($lokasiPenjualan)){
6.             return back()-
>with('danger', 'Lokasi penjualan tidak ditemukan');
7.         }
8.         if($lokasiPenjualan->nama_lokasi_penjualan == $request->nama_lokasi_penjualan && $lokasiPenjualan->status_lokasi_penjualan == $request->status_lokasi_penjualan){
9.             return back()-
>with('success', 'Tidak ada perubahan pada data');
10.        }
11.
12.        $namaSudahAda = LokasiPenjualan::where('id_lokasi_penjualan', '!=', $request->id_lokasi_penjualan)
13.            ->where('nama_lokasi_penjualan', $request->nama_lokasi_penjualan)
14.            -
>where('id_jenis_sample', $lokasiPenjualan->id_jenis_sample)->first();
15.
16.        if(!is_null($namaSudahAda)){
17.            return back()-
>with('danger', 'Sudah ada nama yang sama di latitude dan longitude yang sama');
18.        }
19.
20.        DB::beginTransaction();
21.        try {
22.            $lokasiPenjualan->nama_lokasi_penjualan = $request->nama_lokasi_penjualan;
23.            $lokasiPenjualan->status_lokasi_penjualan = $request->status_lokasi_penjualan;
24.            $lokasiPenjualan->save();
25.            DB::commit();
26.        } catch (Exception $e) {

```

```

27.         DB::rollback();
28.     }
29. } catch (QueryException | Exception $e) {
30.     return back()->with('danger', $e-
    >errorInfo[2]);
31. }
32. return back()-
    >with('success', 'Berhasil menyimpan data lokasi penjuaa
    lan');
33. }

```

### Kode Sumber 4.3 Fungsi Mengubah Data Lokasi Penjualan

Tabel 4.4 Penjelasan Fungsi Mengubah Data Lokasi Penjualan

No. Baris	Kegunaan
4-6	Melakukan pengecekan apabila lokasi yang ditemukan ada
8-9	Melakukan pengecekan apabila perubahan data form yang dimasukkan tidak melakukan perubahan pada data yang sudah ada
12-17	Melakukan pengecekan apabila nama baru yang dimasukkan sama dengan nama data lain pada latitude dan longitude yang sama
20-31	Melakukan perubahan data pada lokasi penjualan
32	Mengembalikan tampilan ke <i>view</i> semula dan memberi pemberitahuan bahwa perubahan data sukses

Implementasi antarmuka untuk menghapus data lokasi penjualan dapat dilihat pada Gambar 4.5 dan fungsi dan penjelasan fungsi untuk menghapus data lokasi penjualan dapat dilihat pada Kode Sumber 4.4. dan Tabel 4.5.

```

1. public function hapusLokasiPenjualan(Request $request)
2. {
3.     try {
4.         $lokasiPenjualan = LokasiPenjualan::find($reque
        st->id_lokasi_penjualan_hapus);

```



```

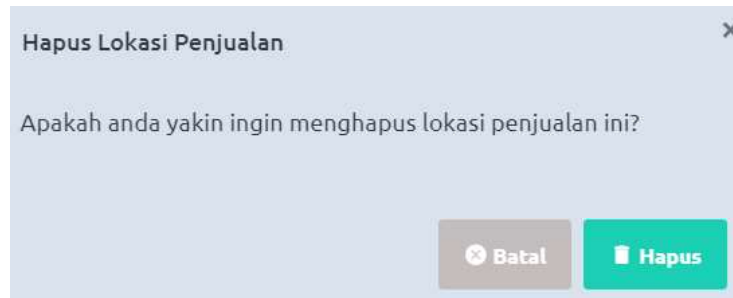
5.         $dataSampel = DataSampel::where('id_lokasi_penj
        ualan', '=', $request->id_lokasi_penjualan_hapus)-
        >get();
6.         if(is_null($lokasiPenjualan)){
7.             return back()-
            >with('danger', 'Lokasi penjualan tidak ditemukan');
8.         }
9.         if(count($dataSampel)>0){
10.            return back()-
            >with('danger', 'Tidak bisa menghapus. Terdapat data sa
            mpel di lokasi penjualan tersebut');
11.        }
12.        DB::beginTransaction();
13.        try {
14.            $lokasiPenjualan->delete();
15.            DB::commit();
16.        } catch (Exception $e) {
17.            DB::rollback();
18.        }
19.    } catch (QueryException | Exception $e) {
20.        return back()->with('danger', $e-
            >errorInfo[2]);
21.    }
22.    return back()-
        >with('success', 'Berhasil menyimpan data lokasi penju
        alan');
23. }

```

#### Kode Sumber 4.4 Fungsi Menghapus Data Lokasi Penjualan

Tabel 4.5 Penjelasan Fungsi Menghapus Data Lokasi Penjualan

No. Baris	Kegunaan
4-7	Melakukan pengecekan apabila lokasi yang ditemukan ada
8-10	Melakukan pengecekan apabila lokasi penjualan masih memiliki data sampel, jika iya maka lokasi penjualan tidak bisa dihapus
12-20	Melakukan penghapusan data lokasi penjualan
22	Mengembalikan tampilan ke <i>view</i> semula dan memberi pemberitahuan bahwa penghapusan data sukses



**Gambar 4.5 Implementasi Antarmuka Menghapus Data Lokasi Penjualan**

#### 4.2.3. Halaman Antarmuka Menambah Data Sampel

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC003 yaitu menambah data sampel. Administrator menambahkan data sampel pada lokasi penjualan yang telah tersimpan dalam sistem. Ketika administrator melakukan kasus penggunaan ini, sistem juga akan menjalankan proses klasifikasi kadar boraks pada data input yang telah dimasukkan. Halaman antarmuka menambah data sampel dapat dilihat pada Gambar 4.6. Kode Sumber 4.5 merupakan fungsi untuk menambahkan data sampel dan penjelasan mengenai kode sumber tersebut dapat dilihat pada Tabel 4.6.

Jenis Makanan \*

Silahkan pilih jenis sample

Lokasi Penjualan \*

Pilih jenis sample terlebih dulu!

Tanggal Pengambilan Data Sample \*

mm/dd/yyyy

File \*

Pilih file

Pilih File

\* Wajib diisi

Simpan Batal

**Gambar 4.6 Implementasi Antarmuka Menambah Data Sampel**

```
1. public function tambahDataSampel(Request $request)
2. {
3.     if($request->file('file_data_sample')-
>extension()!='txt'){
4.         return back()-
>with('danger', 'Format file harus csv');
5.     }
6.
7.     $idLokasiPenjualan = $request-
>id_lokasi_penjualan;
8.     $tanggalPengambilanDataSampel = $request-
>tanggal_pengambilan_data_sample;
9.     $namaFile = $request-
>id_jenis_sample.'_'.Carbon::now()-
>format('YmdHis').'.csv';
10.    $lokasiPenjualan = LokasiPenjualan::find($idLokasiP
enjualan);
11.    try {
12.        Storage::disk('store_test')-
>put($namaFile, file_get_contents($request-
>file('file_data_sample')));
13.        DB::beginTransaction();
14.        try {
15.            $lokasiPenjualan-
>tanggal_pengambilan_sample = $tanggalPengambilanDataSa
mpel;
16.            $lokasiPenjualan-
>flag_lokasi_penjualan = 1;
17.            $lokasiPenjualan-
>updated_at = Carbon::now();
18.            $lokasiPenjualan-
>nama_file_data_sample = $namaFile;
19.            $lokasiPenjualan->save();
20.            Excel::import(new DataSampelImport($idLokas
iPenjualan), request()->file('file_data_sample'));
21.            DB::commit();
22.        } catch (Exception $e) {
23.            DB::rollback();
24.        }
25.
26.
27.    } catch (QueryException | Exception $e) {
```

```

28.         return back()->with('danger', $e-
>errorInfo[2]);
29.     }
30.
31.     $namaFile = public_path().'/DataSampels/DataTest/'.
    $namaFile;
32.     $idJenisSampel = $lokasiPenjualan-
>id_jenis_sample;
33.     $command = escapeshellcmd("python3.6 /var/www/dewij
    elek/prg/predictBaksoOk.py ". $idJenisSampel. " ".$nama
    File);
34.
35.
36.     $output = shell_exec($command);
37.     $hasilKlasifikasi = explode("|", $output);
38.     try {
39.         DB::beginTransaction();
40.         try {
41.             $lokasiPenjualan-
>status_klasifikasi = $hasilKlasifikasi[0];
42.             $lokasiPenjualan-
>rank_klasifikasi = $hasilKlasifikasi[1];
43.             $lokasiPenjualan-
>score_klasifikasi = $hasilKlasifikasi[2];
44.             $lokasiPenjualan-
>updated_at = Carbon::now();
45.             $lokasiPenjualan->save();
46.             DB::commit();
47.         } catch (Exception $e) {
48.             DB::rollback();
49.         }
50.     } catch (QueryException | Exception $e) {
51.         return back()->with('danger', $e-
>errorInfo[2]);
52.     }
53.     return redirect()->back()-
>with('success', 'Sampel '.$hasilKlasifikasi[0]. ' deng
    an rank '.$hasilKlasifikasi[1]. ' dengan score '.$hasil
    Klasifikasi[2]);
54. }

```

#### Kode Sumber 4.5 Fungsi Menambah Data Sampel

**Tabel 4.6 Penjelasan Fungsi Menambah Data Sampel**

No. Baris	Kegunaan
3-4	Melakukan pengecekan apabila file yang dimasukkan tidak berekstensi csv
7-8	Mendeklarasikan variabel <i>idLokasiPenjualan</i> dan <i>tanggalPengambilanDataSampel</i> yang diambil dari <i>request parameter</i> yang telah diinputkan user
9	Mendeklarasikan nama baru file csv sehingga naman yang disimpan menjadi <i>unique</i>
10	Mengambil lokasi penjualan sesuai dengan id yang dipilih
12	Menyimpan file ke dalam <i>storage</i>
15-24	Melakukan proses penambahan data sampel pada lokasi yang dipilih
31	Mendeklarasikan alamat tempat file csv yang telah diunggah disimpan
32	Mendeklarasikan id jenis sampel yang akan diuji
33	Melakukan pemanggilan kode sumber untuk proses klasifikasi
36-51	Mengambil hasil dari proses pengujian klasifikasi dan menyimpan hasil klasifikasi tersebut ke dalam <i>database</i>
53	Mengembalikan tampilan ke <i>view</i> semula dan memberi pemberitahuan bahwa penambahan data sampel sukses

#### 4.2.4. Halaman Antarmuka Mengelola Data Sampel

Halaman ini merupakan halaman yang digunakan untuk melakukan kasus penggunaan UC004 yaitu mengelola data sampel. Administrator menambahkan data sampel pada lokasi penjualan yang telah tersimpan dalam sistem. Antarmuka mengelola data sampel terdiri dari beberapa antarmuka, yaitu antarmuka halaman melihat data sampel, melihat peta lokasi penjualan, dialog konfirmasi menghapus data sampel, dan dialog mengubah data sampel. Halaman antarmuka melihat data sampel dapat dilihat pada Gambar 4.7, Gambar 4.8, Gambar 4.9, dan Gambar 4.10.

## Detail Data Sample | Bakso

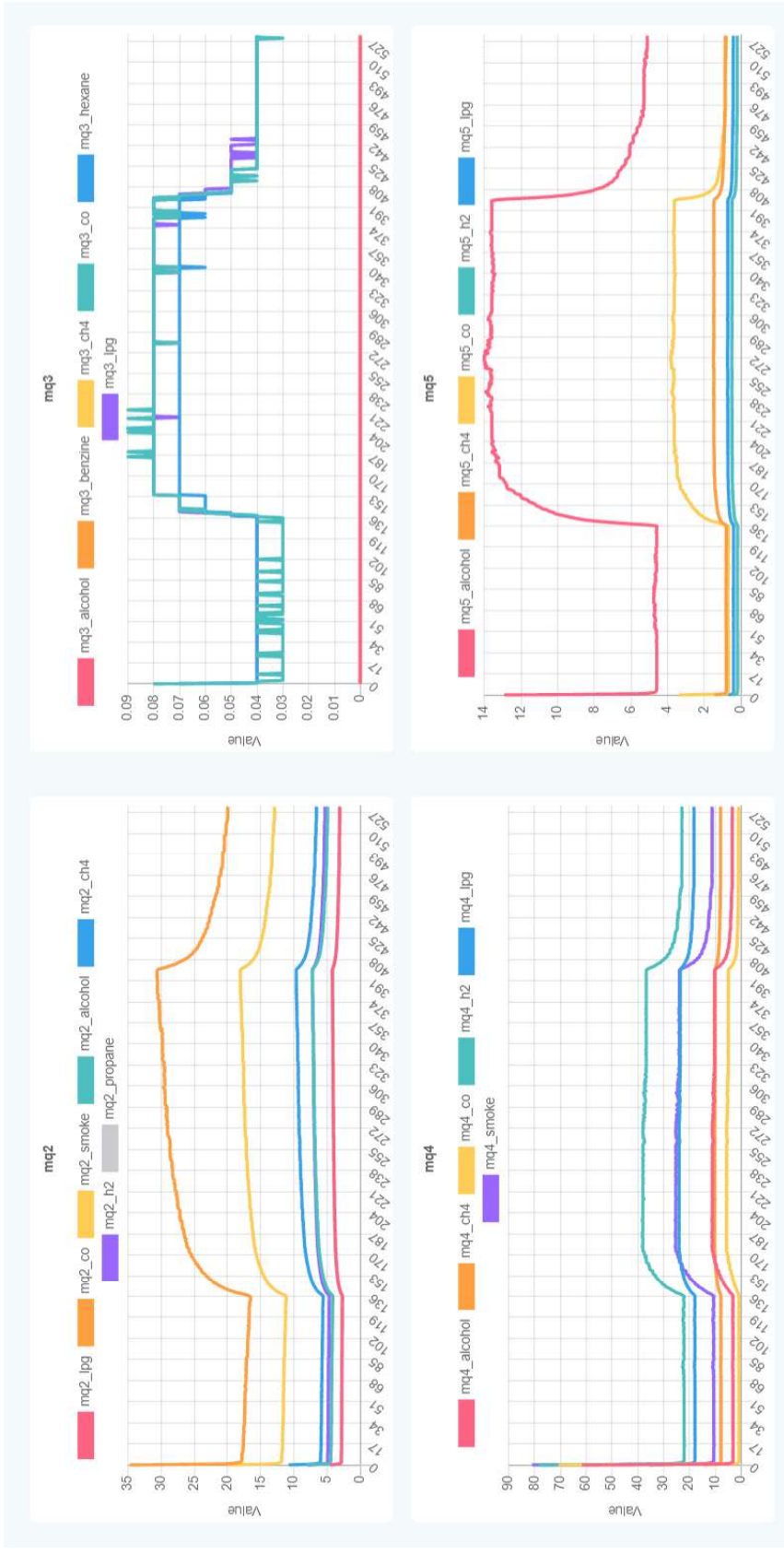
Lokasi Penjualan Bakso / Superindo Arief Rahman Hakim - Bakso Urat Tanpa Merk

[Lihat Peta Lokasi Penjualan](#)
[Lihat Foto Lokasi Penjualan](#)
[Hapus Data Sample](#)
[Ubah Data Sample](#)

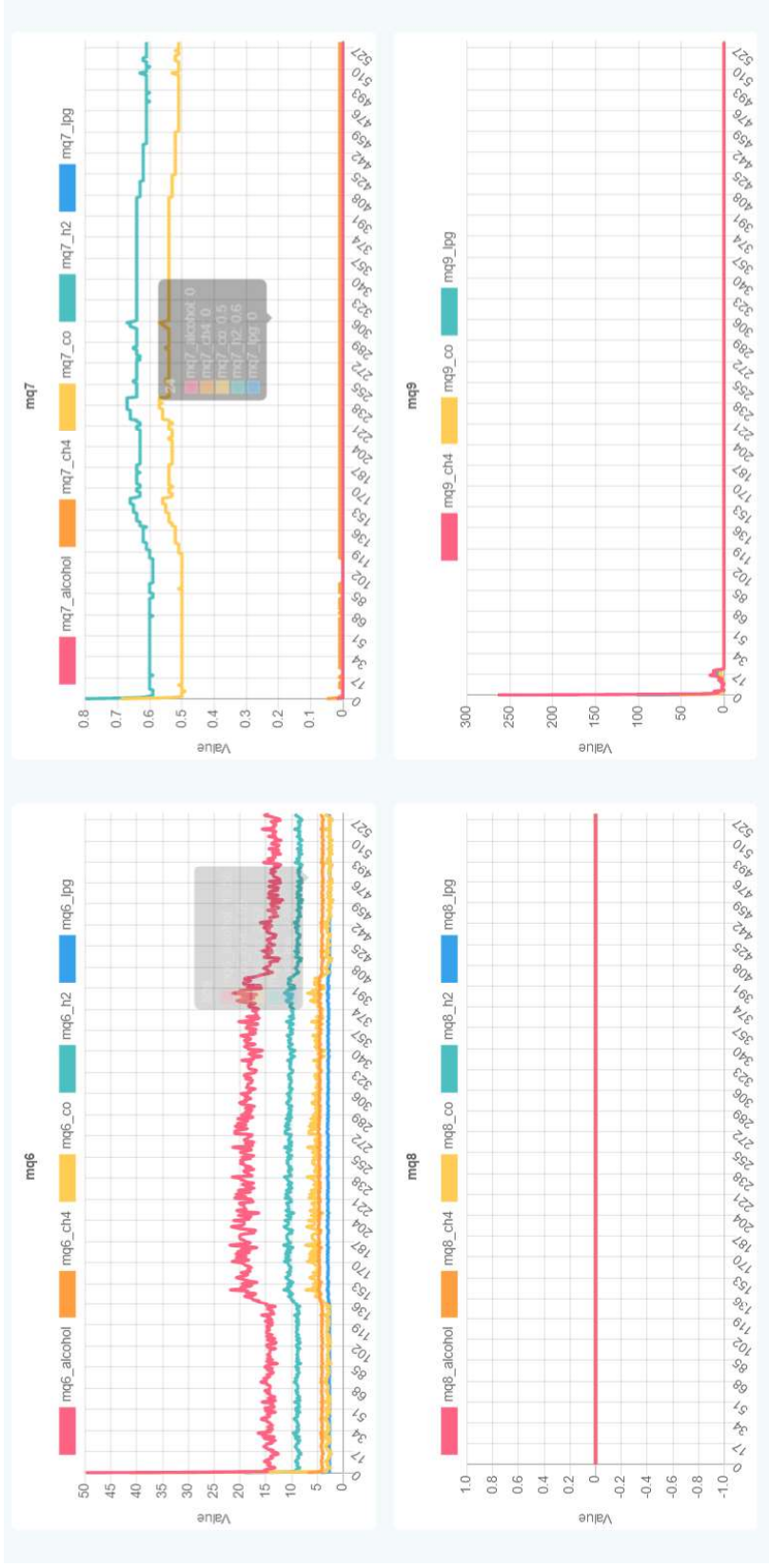
Superindo Arief Rahman Hakim - Bakso Urat Tanpa Merk			
Latitude:	-7.2896632	Tanggal Pengambilan Lokasi:	2020-01-05 03:02:32
Longitude:	112.78786705	Kelas Klasifikasi:	Tidak terindikasi ada boraks
		Tanggal Pengambilan Sample:	2020-01-06
		Tingkat probabilitas positif mengandung boraks/formalin:	11.24 %

No.	mq2_lpg	mq2_co	mq2_smoke	mq2_alcohol	mq2_ch4	mq2_h2	mq2_propane	mq3_alcohol	mq3_benzine	mq3_ch4	mq3_co	mq3_hexane	mq3_lpg	mq4_alcohol	mq4_
1.	5.17	41.01	22.75	9.4	12.53	8.89	5.28	0	0	0	0.08	0.07	0.08	98.71	16.
2.	3.68	25.54	15.69	6.14	8.22	6.34	3.74	0	0	0	0.04	0.04	0.04	10.57	10.
3.	3.57	24.46	15.16	5.9	7.91	6.15	3.62	0	0	0	0.03	0.03	0.04	8.08	9.
4.	3.6	24.78	15.32	5.97	8	6.21	3.65	0	0	0	0.03	0.03	0.03	7.77	9.
5.	3.64	25.11	15.48	6.04	8.1	6.27	3.69	0	0	0	0.03	0.03	0.03	8.02	9.
6.	3.64	25.18	15.51	6.06	8.12	6.28	3.7	0	0	0	0.03	0.03	0.03	8.08	9.
7.	3.65	25.25	15.54	6.07	8.14	6.29	3.7	0	0	0	0.03	0.03	0.03	7.87	9.
8.	3.64	25.21	15.52	6.06	8.13	6.28	3.7	0	0	0	0.03	0.03	0.03	7.83	9.
9.	3.63	25.09	15.47	6.04	8.09	6.26	3.69	0	0	0	0.03	0.03	0.03	7.61	9.
10.	3.62	25.01	15.43	6.02	8.07	6.25	3.68	0	0	0	0.03	0.03	0.03	7.27	9.
11.	3.61	24.89	15.37	6	8.03	6.23	3.67	0	0	0	0.03	0.03	0.03	7.01	9.
12.	3.58	24.6	15.23	5.93	7.95	6.17	3.63	0	0	0	0.03	0.03	0.03	6.96	9.
13.	3.58	24.54	15.2	5.92	7.93	6.16	3.63	0	0	0	0.02	0.03	0.03	6.58	9.
14.	3.55	24.32	15.09	5.87	7.87	6.12	3.6	0	0	0	0.03	0.03	0.03	6.42	9.
15.	3.54	24.22	15.05	5.85	7.84	6.11	3.59	0	0	0	0.03	0.03	0.03	6.39	9.
16.	3.51	23.95	14.91	5.79	7.76	6.06	3.56	0	0	0	0.03	0.03	0.03	6.21	8.

**Gambar 4.7 Implementasi Antarmuka Melihat Data Sampel Bagian Tabel Data Sampel**

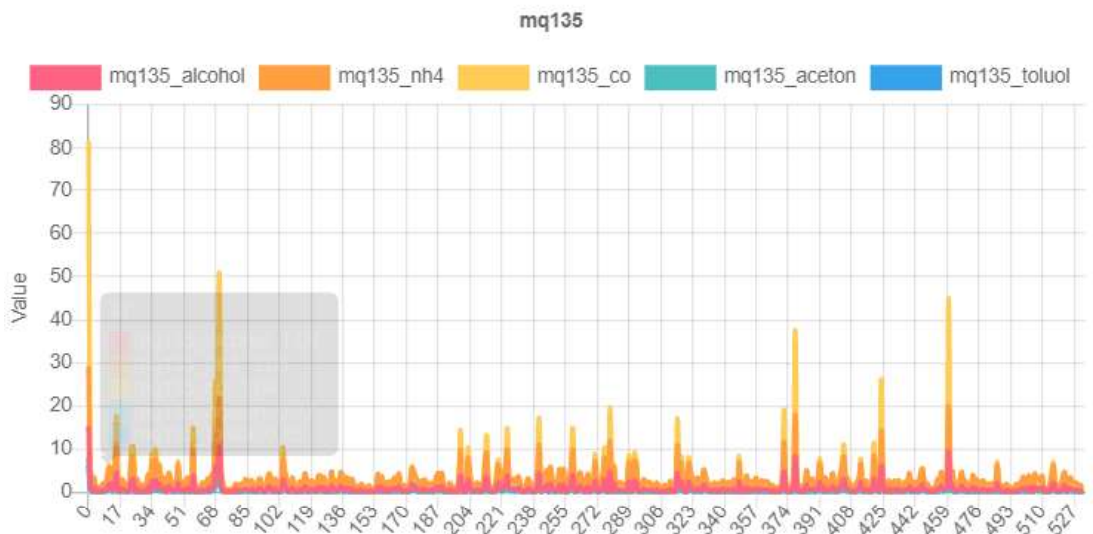


Gambar 4.8 Implementasi Antarmuka Melihat Data Sampel Bagian Grafik Sensor MQ2 – MQ5



Gambar 4.9 Implementasi Antarmuka Melihat Data Sampel Bagian Grafik Sensor MQ6 – MQ9





**Gambar 4.10 Implementasi Antarmuka Melihat Data Sampel Bagian Grafik Sensor MQ135**

Pada Gambar 4.9, nilai-nilai yang terdapat pada tabel merupakan nilai-nilai yang didapatkan oleh sensor untuk setiap parameter pada *electronic nose*. Nilai-nilai tersebut merepresentasikan bau yang didapatkan oleh *enose*. Pada gambar Gambar 4.8, Gambar 4.9, dan Gambar 4.10 merupakan grafik yang merepresentasikan nilai-nilai yang telah didapatkan dengan *enose* sebelumnya. Grafik tersebut berfungsi untuk menunjukkan parameter sensor manakah yang paling peka terhadap data sampel yang ingin diuji.

Pada Kode Sumber 4.6 merupakan fungsi untuk melihat data sampel dan penjelasan untuk fungsi tersebut dapat dilihat pada Tabel 4.7.

```

1. public function ambilDetailDataSampel($idLokasiPenjuala
   n)
2. {
3.     $laporan = Laporan::where('is_read', 0)-
   >orderBy('created_at', 'desc')->get();
4.     $daftarJenisSampel = JenisSampel::all();
5.     if (is_null(LokasiPenjualan::find($idLokasiPenjuala
   n))) {
6.         return redirect()->back()-
   >with('danger', 'Lokasi penjualan tidak ditemukan');

```

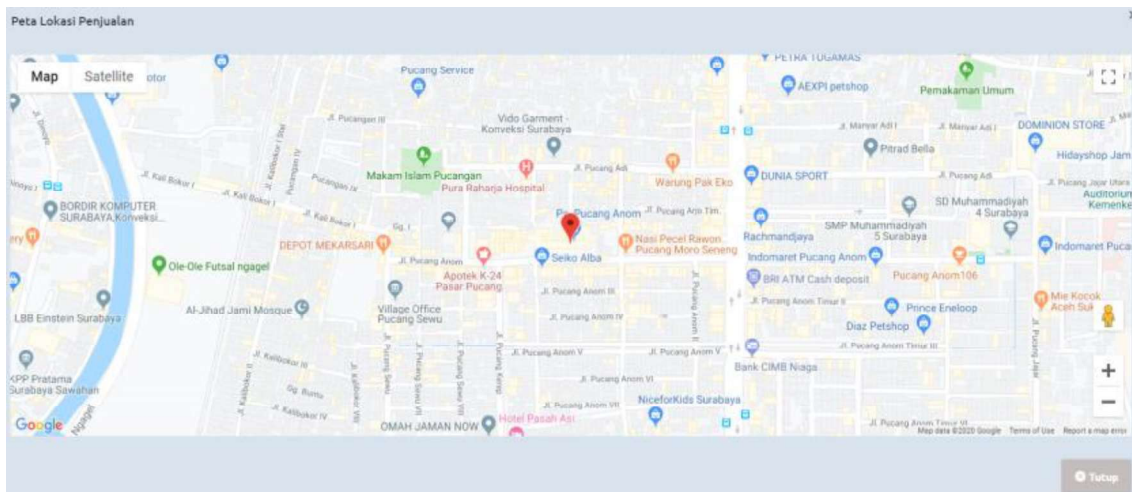
```

7.     }
8.
9.     $dataSampel = DataSampel::where('id_lokasi_penjuala
n', '=', $idLokasiPenjualan)->paginate(20);
10.    $lokasiPenjualan = LokasiPenjualan::find($idLokasiP
enjualan);
11.    $results = [
12.        'laporan' => $laporan,
13.        'daftarJenisSampel' => $daftarJenisSampel,
14.        'pageTitle' => 'Detail Data Sampel | Bakso',
15.        'dataSampel' => $dataSampel,
16.        'lokasiPenjualan' => $lokasiPenjualan,
17.    ];
18.    return view('detail-data-sampel', $results);
19. }

```

### Kode Sumber 4.6 Fungsi Menampilkan Data Sampel

Implementasi antarmuka untuk melihat peta lokasi penjualan dan foto lokasi penjualan dapat dilihat pada Gambar 4.11 dan Gambar 4.12.



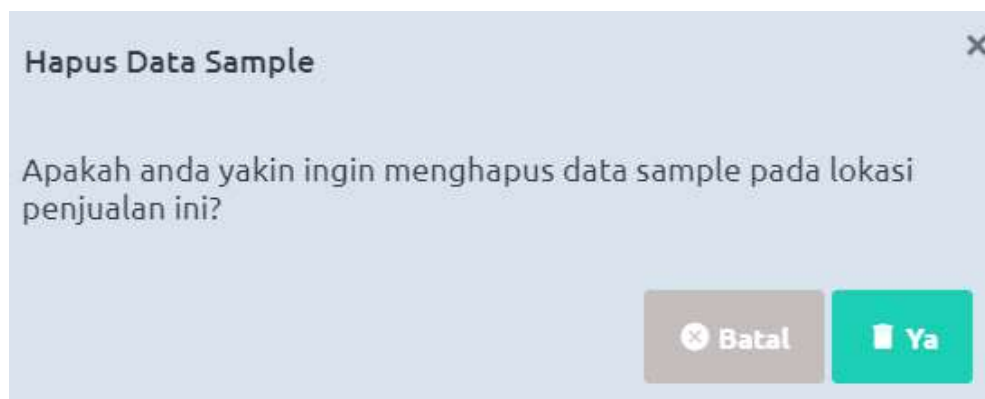
**Gambar 4.11 Implementasi Antarmuka Melihat Peta Lokasi Penjualan**

Implementasi antarmuka untuk menghapus data sampel dapat dilihat pada Gambar 4.13 dan fungsi untuk menghapus data sampel dapat dilihat pada Kode Sumber 4.7. Penjelasan dari kode

sumber untuk fungsi menghapus data sampel data dilihat pada Tabel 4.8.



**Gambar 4.12 Implementasi Antarmuka Melihat Foto Lokasi Penjualan**



**Gambar 4.13 Implementasi Antarmuka Menghapus Data Sampel**

**Tabel 4.7 Penjelasan Fungsi Menampilkan Data Sampel**

No. Baris	Kegunaan
3-4	Mengambil laporan belum terbaca dan jenis sampel yang akan ditampilkan pada semua tampilan aplikasi

No. Baris	Kegunaan
5-7	Melakukan pengecekan apabila lokasi penjualan yang dipilih ada pada <i>database</i>
9-10	Mengambil data nilai pada data sampel sesuai dengan lokasi penjualan yang telah dipilih
11-16	Menyimpan data-data yang telah diambil ke dalam sebuah array untuk melakukan <i>passing</i> ke dalam view
18	Memanggil <i>view</i> untuk menampilkan data-data yang sudah disimpan

```

1. public function hapusDataSampel(Request $request)
2. {
3.     $idLokasiPenjualan = $request-
>id_lokasi_penjualan;
4.     if (is_null(LokasiPenjualan::find($idLokasiPenjuala
n))) {
5.         return redirect()->back()-
>with('danger', 'Lokasi penjualan tidak ditemukan');
6.     }
7.
8.     $lokasiPenjualan = LokasiPenjualan::find($idLokasiP
enjualan);
9.     $dataSampel = DataSampel::where('id_lokasi_penjuala
n', '=', $idLokasiPenjualan)
10.     ->get();
11.     if (count($dataSampel) == 0) {
12.         return redirect()->back()-
>with('success', 'Tidak ada data yang dihapus');
13.     }
14.
15.     try {
16.         File::delete( public_path().'/DataSampels/DataT
est/' . $lokasiPenjualan->nama_file_data_sample);
17.         DB::table('data_sample')-
>where('id_lokasi_penjualan', '=', $idLokasiPenjualan)-
>delete();
18.         DB::beginTransaction();
19.         try {
20.             $lokasiPenjualan-
>flag_lokasi_penjualan = 0;

```

```

21.         $lokasiPenjualan-
>updated_at = Carbon::now();
22.         $lokasiPenjualan-
>tanggal_pengambilan_sample = null;
23.         $lokasiPenjualan-
>status_klasifikasi = null;
24.         $lokasiPenjualan-
>rank_klasifikasi = null;
25.         $lokasiPenjualan-
>score_klasifikasi = null;
26.         $lokasiPenjualan-
>nama_file_data_sample = null;
27.         $lokasiPenjualan->save();
28.         DB::commit();
29.     } catch (Exception $e) {
30.         DB::rollback();
31.     }
32. } catch (QueryException | Exception $e) {
33.     return back()->with('danger', $e-
>errorInfo[2]);
34. }
35. return redirect()->back()-
>with('success', 'Berhasil menghapus data sampel');
36. }

```

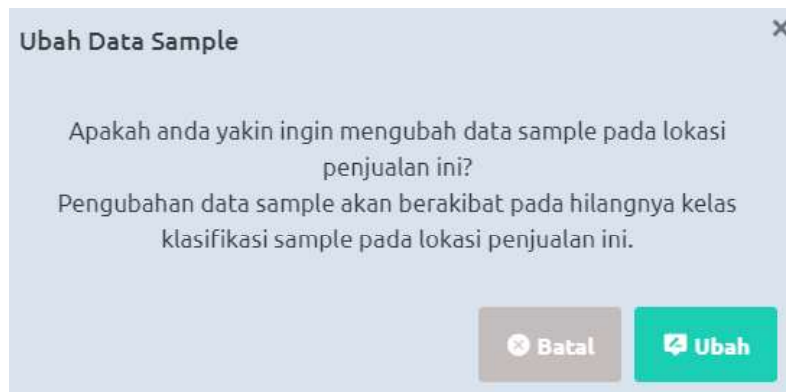
#### Kode Sumber 4.7 Fungsi Menghapus Data Sampel

Tabel 4.8 Penjelasan Fungsi Menghapus Data Sampel

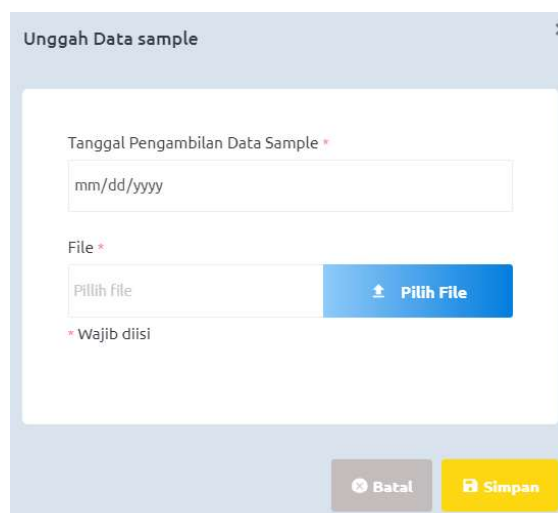
No. Baris	Kegunaan
3-6	Melakukan pengecekan apabila lokasi yang ditemukan ada
8	Mengambil lokasi penjualan sesuai yang dipilih oleh user
9-12	Melakukan pengecekan apabila lokasi penjualan yang dipilih memiliki data sampel
16-33	Melakukan proses penghapusan data sampel dari database, menghapus file csv dari <i>storage</i> dan melakukan perubahan data status klasifikasi pada lokasi penjualan

No. Baris	Kegunaan
35	Mengembalikan tampilan ke <i>view</i> semula dan memberi pemberitahuan bahwa penghapusan data sukses

Implementasi antarmuka untuk modal konfirmasi pengubahan data sampel dapat dilihat pada Gambar 4.14. Sedangkan, implementasi antarmuka modal pengubahan data sampel dan fungsi untuk mengubah data sampel dapat dilihat pada Gambar 4.15 dan Kode Sumber 4.8. Penjelasan dari fungsi mengubah data sampel dapat dilihat pada Tabel 4.9.



**Gambar 4.14 Implementasi Antarmuka Konfirmasi Mengubah Data Sampel**



**Gambar 4.15 Implementasi Antarmuka Modal Mengubah Data Sampel**

```

1. public function ubahDataSampel(Request $request)
2. {
3.     if($request->file('file_data_sample')-
>extension()!='txt'){
4.         return back()-
>with('danger', 'Format file harus csv');
5.     }
6.     $idLokasiPenjualan = $request-
>id_lokasi_penjualan;
7.     $namaFile = $request-
>id_jenis_sample.'_'.Carbon::now()-
>format('YmdHis').'.csv';
8.     if (is_null(LokasiPenjualan::find($idLokasiPenjuala
n))) {
9.         return redirect()->back()-
>with('danger', 'Lokasi penjualan tidak ditemukan');
10.    }
11.
12.    $lokasiPenjualan = LokasiPenjualan::find($idLokasiP
enjualan);
13.    $dataSampel = DataSampel::where('id_lokasi_penjuala
n', '=', $idLokasiPenjualan)
14.        ->get();
15.
16.    if (count($dataSampel) > 0) {
17.        try {
18.            DB::table('data_sample')-
>where('id_lokasi_penjualan', '=', $idLokasiPenjualan)-
>delete();
19.        } catch (QueryException | Exception $e) {
20.            return back()->with('danger', $e-
>errorInfo[2]);
21.        }
22.    }
23.    try {
24.        File::delete( public_path().'/DataSampels/DataT
est/'.$lokasiPenjualan->nama_file_data_sample);
25.        Storage::disk('store_test')-
>put($namaFile, file_get_contents($request-
>file('file_data_sample')));
26.        DB::beginTransaction();
27.        try {

```

```
28.         $lokasiPenjualan-
>flag_lokasi_penjualan = 1;
29.         $lokasiPenjualan-
>updated_at = Carbon::now();
30.         $lokasiPenjualan-
>tanggal_pengambilan_sample = $request-
>tanggal_pengambilan_data_sample;
31.         $lokasiPenjualan-
>nama_file_data_sample = $namaFile;
32.         $lokasiPenjualan->save();
33.         DB::commit();
34.         Excel::import(new DataSampelImport($idLokas
iPenjualan), $request-
>file('file_data_sample'));
35.     } catch (Exception $e) {
36.         DB::rollback();
37.     }
38.
39. } catch (QueryException | Exception $e) {
40.     dd($e);
41. }
42.
43. $namaFile = public_path().'/DataSampels/DataTest/'.
$namaFile;
44. $idJenisSampel = $lokasiPenjualan-
>id_jenis_sample;
45. $command = escapeshellcmd("python3.6 /var/www/dewij
elek/prg/predictBaksoOk.py ". $idJenisSampel. " ".$nama
File);
46.
47.
48. $output = shell_exec($command);
49. $hasilKlasifikasi = explode("|", $output);
50. try {
51.     DB::beginTransaction();
52.     try {
53.         $lokasiPenjualan-
>status_klasifikasi = $hasilKlasifikasi[0];
54.         $lokasiPenjualan-
>rank_klasifikasi = $hasilKlasifikasi[1];
55.         $lokasiPenjualan-
>score_klasifikasi = $hasilKlasifikasi[2];
```



```

56.         $lokasiPenjualan-
           >updated_at = Carbon::now();
57.         $lokasiPenjualan->save();
58.         DB::commit();
59.     } catch (Exception $e) {
60.         DB::rollback();
61.     }
62. } catch (QueryException | Exception $e) {
63.     return back()->with('danger', $e-
           >errorInfo[2]);
64. }
65. return redirect()->back()-
           >with('success', 'Berhasil mengubah data sampel. Sampel
           '.$hasilKlasifikasi[0]. ' dengan rank '.$hasilKlasifik
           asi[1]. ' dengan score '.$hasilKlasifikasi[2]);
66. }

```

#### Kode Sumber 4.8 Fungsi Mengubah Data Sampel

Tabel 4.9 Penjelasan Fungsi Mengubah Data Sampel

No. Baris	Kegunaan
3-4	Melakukan pengecekan apabila file yang dimasukkan tidak berekstensi csv
6	Mendeklarasikan id lokasi penjualan yang data sampelnya diubah
7	Mendeklarasikan nama baru file csv sehingga naman yang disimpan menjadi <i>unique</i>
8-9	Melakukan pengecekan apabila lokasi penjualan yang dipilih ada pada <i>database</i>
12	Mengambil lokasi penjualan sesuai dengan id yang dipilih
13-20	Melakukan penghapusan data sampel yang sudah ada sebelumnya pada lokasi penjualan yang dipilih
24	Menghapus file csv sebelumnya dari <i>storage</i>
25	Menyimpan file csv baru ke dalam <i>storage</i>
26-41	Melakukan proses perubahan data sampel pada lokasi yang dipilih
43	Mendeklarasikan alamat tempat file csv yang telah diunggah disimpan

No. Baris	Kegunaan
44	Mendeklarasikan id jenis sampel yang akan diuji
45	Melakukan pemanggilan kode sumber untuk proses klasifikasi
4-64	Mengambil hasil dari proses pengujian klasifikasi dan menyimpan hasil klasifikasi tersebut ke dalam <i>database</i>
65	Mengembalikan tampilan ke <i>view</i> semula dan memberi pemberitahuan bahwa perubahan data sampel sukses

#### 4.2.5. Halaman Antarmuka Melihat Laporan Mengenai Lokasi Penjualan

Antarmuka melihat laporan mengenai lokasi penjualan akan digunakan pada kasus penggunaan melihat lokasi penjualan. Implementasi antarmuka melihat laporan dan fungsi untuk melihat laporan dapat dilihat pada Gambar 4.16 dan Kode Sumber 4.9. Penjelasan mengenai fungsi melihat laporan mengenai lokasi penjualan dapat dilihat pada Tabel 4.10.

```

1. public function ambillaporan(Request $request)
2. {
3.     $laporan = Laporan::where('is_read', 0)-
>orderBy('created_at', 'desc')->get();
4.     $daftarJenisSampel = JenisSampel::all();
5.     $laporanAll = Laporan::where('is_read', 0)-
>orderBy('created_at', 'desc')->paginate(20);
6.     $results = [
7.         'laporan' => $laporan,
8.         'daftarJenisSampel' => $daftarJenisSampel,
9.         'laporanAll' => $laporanAll,
10.        'pageTitle' => 'Laporan',
11.    ];
12.    // dd($laporan);
13.    return view('data-laporan', $results);
14. }
```

#### Kode Sumber 4.9 Fungsi Melihat Laporan Mengenai Lokasi Penjualan

**Tabel 4.10 Penjelasan Fungsi Melihat Laporan Mengenai Lokasi Penjualan**

No. Baris	Kegunaan
3-4	Mengambil laporan belum terbaca dan jenis sampel yang akan ditampilkan pada semua tampilan aplikasi
5	Mengambil semua laporan yang sudah terbaca maupun belum terbaca
6-11	Menyimpan data-data yang telah diambil ke dalam sebuah array untuk melakukan <i>passing</i> ke dalam view
12	Memanggil <i>view</i> untuk menampilkan data-data yang sudah disimpan

No.	Nama Pelapor	Tanggal Laporan	Isi Laporan	Opsi
1.	diana	13-01-2020	Perbarui Data Lokasi Ini	<a href="#">Lihat Lokasi</a>
2.	Diana	07-01-2020	Perbarui Data Lokasi Ini	<a href="#">Lihat Lokasi</a>
3.	Dewi	05-01-2020	Cek Penjualan Bakso/Tahu Lokasi Ini	<a href="#">Lihat Lokasi</a>
4.	Sekarini	04-01-2020	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>

**Gambar 4.16 Implementasi Antarmuka Melihat Laporan Mengenai Lokasi Penjualan**

#### 4.2.6. Halaman Antarmuka Melihat Notifikasi

Antarmuka melihat notifikasi akan digunakan pada kasus penggunaan melihat notifikasi. Implementasi antarmuka melihat notifikasi dapat dilihat pada Gambar 4.17.



**Gambar 4.17 Implementasi Antarmuka Melihat Notifikasi**

### 4.3. Implementasi Proses Pengumpulan dan Klasifikasi Data

Pada subbab ini akan dijelaskan mengenai implementasi pengambilan data dengan *electronic nose*, proses klasifikasi data sampel mulai dari proses latihan hingga proses klasifikasi kadar boraks yang terkandung pada data sampel.

#### 4.3.1. Implementasi Proses Pengumpulan Data

Pengambilan *ground truth* dilakukan dengan cara mengambil 40 sampel bakso bebas dari boraks dan 41 sampel bakso yang dibuat dengan menggunakan boraks. Pengambilan data dilakukan selama 20 menit menggunakan mesin *enose* Enosika dan menggunakan *tools* program Arduino. Program Arduino akan menjalankan kode sumber untuk mengambil nilai-nilai tegangan yang didapatkan ketika sensor-sensor pada *electronic nose* terkena molekul bau dari data sampel. Contoh nilai-nilai tegangan yang didapatkan pada program Arduino dapat dilihat pada Tabel 4.11.

Setelah dilakukan pengambilan data dilakukan secara 20 menit, maka nilai nilai yang ada pada program Arduino akan disimpan pada file csv yang selanjutnya akan dilakukan proses *training* data sampel. Hal itu juga dilakukan untuk pengambilan sampel untuk pengujian pada sampel yang didapatkan dari 10 lokasi berbeda.

**Tabel 4.11 Contoh Hasil Pengambilan Data Sampel Dengan Electronic Nose dan Arduino**

40:52.4,-
>,4.6,34.85,20.02,8.12,10.84,7.92,4.69,0,0,0,0.08,0.07,0.08,62.67,14.64,72.48,79.07,3
5.8,81.86,12.91,1.43,3.37,0.47,0.7,58.94,7.33,58.72,21.16,4.55,0.02,0.05,0.7,0.81,0.01
,27.07,10.69,37.77,45.3,3.24,12.9,4.09,8.97,16.85,38.92,259.86,60.06,60.58,21.77
40:54.7,-
>,3.13,20.36,13.13,5,6.72,5.39,3.17,0,0,0,0.04,0.04,0.04,5.5,8.76,1.96,28.18,20.38,15.
64,5.55,0.89,0.91,0.22,0.44,18.32,4.52,4.96,10.29,2.87,0,0.01,0.53,0.63,0,26.56,9.89,3
4.48,44.95,3.1,8.27,2.8,6.27,4.8,12.03,60.37,20.83,24.12,6.12
40:57.0,-
>,2.92,18.48,12.17,4.59,6.17,5.04,2.95,0,0,0,0.04,0.04,0.04,3.5,7.97,1,23.27,18.36,11.
5,4.77,0.82,0.72,0.19,0.41,15.85,4.25,3.65,9.41,2.72,0,0.01,0.5,0.6,0,18.19,2.16,5.78,3
8.52,1.27,6.81,2.37,5.36,1.67,4.49,17.73,8.56,11.13,2.11

40:59.4,-  
 >,2.88,18.14,11.99,4.51,6.06,4.97,2.91,0,0,0,0.04,0.04,0.04,3.25,7.84,0.9,22.55,18.04,  
 10.93,4.63,0.81,0.68,0.19,0.4,15.37,4.2,3.42,9.23,2.68,0,0,0.5,0.6,0,26.83,10.31,36.22,  
 45.14,3.18,6.46,2.27,5.14,60.2,127.95,1141.45,175.77,154.14,78.85  
 41:01.6,-

### 4.3.2. Implementasi Proses Latihan Data

Setelah dilakukan pengambilan data sampel, hal yang dilakukan selanjutnya adalah *training* data. Karena pada sebuah file csv data sampel berisi banyak data pada setiap kolom, maka untuk mewakili masing-masing kolom pada setiap file sampel akan dilakukan seleksi fitur dengan menggunakan standar deviasi dan rata-rata dari nilai-nilai untuk setiap kolom. Sehingga, pada proses *training* pada data di Tugas Akhir ini menggunakan 96 fitur. Untuk setiap file sampelnya, akan dilabeli “B” untuk sampel bakso bebas boraks dan “O” untuk sampel bakso yang dibuat dengan boraks. Sehingga, proses *training* dapat mengacu ke pelabelan file tersebut.

```

Start Clasification With Parameter
-----
SVM
#####
Best score SVM accuracy=0.888889
With Parameter {'model_C': 5, 'model_gamma': 1, 'reduce_dim_n_components': 9}
#####
KNN
#####
Best score KNN accuracy=0.876543
With Parameter {'model_n_neighbors': 1, 'model_weights': 'distance',
'reduce_dim_n_components': 9}
#####
MLP
#####
Best score MLP accuracy=0.925926
With Parameter {'model_activation': 'relu', 'model_alpha': 0.1, 'model_hidden_layer_sizes':
500, 'model_learning_rate': 'constant', 'model_max_iter': 10000, 'reduce_dim_n_components':
9}
#####
Model SCORE
SVM          0.888889
KNN          0.876543
MLP          0.925926
#####
All Best Result Model = MLP
-----
scoreBest=0.925926
  
```

**Gambar 4.18 Hasil Latihan Data Latih**

Proses latihan data dilakukan dengan cara membandingkan latihan dengan ketiga metode yaitu *Support Vector Machine*, *K-*

*Nearest Neighbor*, dan *Multilayer Perceptron*. Dari proses tersebut, didapatkan bahwa metode *Multilayer Perceptron* memiliki akurasi paling tinggi dengan akurasi 0.925926 dengan menggunakan parameter mean dan standar deviasi dari setiap kolom pada data sampel. Hasil dari proses pencarian metode latihan data dapat dilihat pada Gambar 4.18.

Setelah didapatkan metode terbaik, maka model dengan metode MLP akan dibuat. Implementasi pembuatan model data latih dapat dilihat pada Kode Sumber 4.10 dan penjelasan dari implementasi pembuatan model dapat dilihat pada Tabel 4.12.

```

1. import glob
2. import pandas as pd
3. import numpy as np
4. from sklearn.preprocessing import StandardScaler
5. from sklearn.decomposition import PCA
6. from sklearn.neural_network import MLPClassifier
7. from sklearn.svm import SVC
8. import time
9. import pickle
10. import warnings
11. warnings.filterwarnings("ignore")
12. warnings.filterwarnings(action="ignore",category=DeprecationWarning)
13. warnings.filterwarnings(action="ignore",category=FutureWarning)
14. from sklearn.neural_network import MLPClassifier
15.
16.
17. if __name__=="__main__":
18.     RAW = "Datas/CSVB/*.csv"
19.     dfRAW = pd.DataFrame([])
20.     for f in glob.glob(RAW):
21.         print("Proses : %s"%f)
22.         tmpData = pd.read_csv(f,sep=',')
23.         nmFile = f[11:]
24.         tmpData['idFile']=nmFile.replace(".csv","")
25.         tmpData['kelas']=f[11:12]
26.         dfRAW = dfRAW.append(tmpData)
27.     IDFile = dfRAW.idFile.unique()

```

```
28.     KELAS = dfRAW.kelas.unique()
29.     KELAS.sort()
30. #
31.     if 'Timestamp' in dfRAW.columns:
32.         dfRAW = dfRAW.drop(columns=['Timestamp', 'Panah
33.         '])
34.     dfRAW = dfRAW.convert_objects(convert_numeric=True)
35.     dfRAW = dfRAW.fillna(0)
36.     dfAvg = dfRAW.groupby(['idFile']).agg([np.mean, np.s
37.     td])
38.     N_fitur=dfAvg.shape[1] ## jumlah kolom
39.     for i in dfAvg.index.values:
40.         kelas = list(KELAS).index(i[:1])
41.         dfAvg.loc[i, 'Kelas']=kelas
42.
43.
44.     dataValue = dfAvg.values
45.     dataX     = dataValue[:, :N_fitur]
46.     scaler   = StandardScaler()
47.     N_PCA=9
48.     dataX    = scaler.fit_transform(dataX)
49.     with open('BAKSO_scaler.pkl', 'wb') as f:
50.         pickle.dump(scaler, f)
51.     pca      = PCA(n_components=N_PCA)
52.     dataX    = pca.fit_transform(dataX)
53.     with open('BAKSO_PCA.pkl', 'wb') as f:
54.         pickle.dump(pca, f)
55.
56.
57.     yLabel   = dataValue[:, N_fitur]
58.
59.     model = MLPClassifier(activation='tanh', alpha=0.1, h
60.     idden_layer_sizes=500, learning_rate='constant', max_iter
61.     =1000)
62.     t = time.time()
63.     model.fit(dataX, yLabel)
64.     with open('BAKSO_modelMLP.pkl', 'wb') as f:
65.         pickle.dump(model, f)
```

```

65.         elapsed = time.time() - t
66.     print ("Save Model succeded Time Elapsed = %g"%elap
sed)

```

### Kode Sumber 4.10 Implementasi Pembuatan Model Data Sampel dengan Metode Multilayer Perceptron

**Tabel 4.12 Penjelasan Kode Sumber Implementasi Pembuatan Model Data Sampel dengan Metode Multilayer Perceptron**

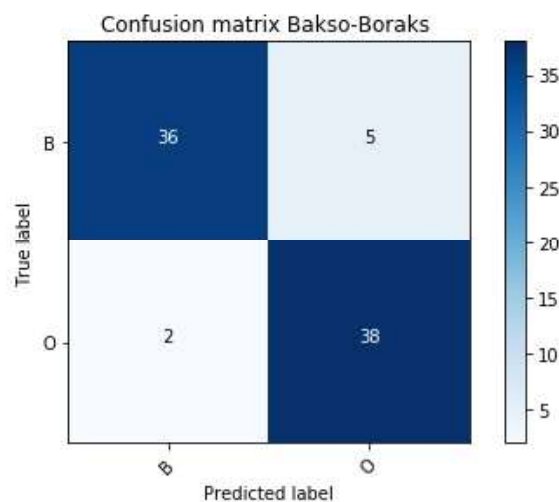
No. Baris	Kegunaan
1-14	Melakukan <i>import</i> pada dependency dan modul yang dibutuhkan pada fungsi-fungsi yang dinakan
17	Deklarasi kelas main
18	Deklarasi path untuk menyimpan data sampel untuk latihan
19	Mendeklarasikan array 2 dimensi dari modul panda
20	Melakukan pengulangan pada setiap file yang ada pada variabel RAW
21	Mencetak file apa yang sedang diproses
22	Memisahkan atribut pada file csv menjadi kolom-kolom tersendiri
23-24	Mengambil nama file
25	Mengambil nama kelas sesuai yang ditentukan
26	Menyisipkan data dari csv ke variabel dfRAW
27	Mengambil idFile secara unique
28-29	Mengambil kelas secara unique dan mengurutkan kelas
31-32	Melakukan pengecekan apakah ada kolom Timestamp dan panah, jika iya, kolom akan dihilangkan
33-34	Mengubah isi kolom menjadi double, jika bukan numerik maka akan otomatis menjadi 0
35	Mengambil standar deviasi dan rata-rata dari setiap kolom sebagai fitur
36	Mendapatkan jumlah kolom dari variabel berisi standar deviasi dan rata-rata setiap kolom
39-41	Menyimpan index kelas pada setiap data di dfAvg
44	Mengambil nilai dari data pada dfAvg
45	Menggabungkan antara dgAvg dengan jumlah fitur
46	Mendeklarasi scaler



No. Baris	Kegunaan
47	Mendeklasrasikan jumlah PCA
48	Melakukan transform dataX dengan scaler
49-50	Membuat file scaler
51	Mendeklarasikan PCA
52	Melakukan transform PCA dengan scaler
53-54	Membuat file PCA
57	Mendeklarasikan yFitur
59	Melakukan klasifikasi dengan metode MLP
60	Mendeklarasi perhitungan waktu
61	Melakukan latihan data
63-64	Membuat file model
65-66	Response bahwa proses selesai

Confusion matrix, without normalization

```
[[36  5]
 [ 2 38]]
```



```
precision: [0.95 0.88]
recall: [0.88 0.95]
fscore: [0.91 0.92]
support: [41 40]
accuracy: 0.9135802469135802
```

**Gambar 4.19 Confusion Matrix, Precision, Recall, Fscore, Support, dan Accuracy dari Proses Training Data**

Hasil dari proses ini adalah ketiga file yang akan digunakan pada proses klasifikasi yaitu BAKSO\_modelMLP.pkl,

BAKSO\_scaler.pkl, dan BAKSO\_PCA.pkl. Dari proses di atas, didapatkan akurasi sebesar 90% dan tingkat kepercayaan 91%. Selain itu, proses klasifikasi tersebut mendapatkan nilai presisi, *recall*, *fscore*, support seperti pada Gambar 4.19.

### 4.3.3. Implementasi Proses Klasifikasi Data Sampel

Ketiga file yang telah didapatkan dari proses latihan akan digunakan pada proses klasifikasi. Implementasi proses klasifikasi dan penjelasan implementasi dari proses klasifikasi dapat dilihat pada Kode Sumber 4.11 dan Tabel 4.16. Keluaran dari proses ini adalah status klasifikasi dari data sampel dan nilai probabilitas sampel tersebut positif mengandung boraks. Sampel dikatakan positif mengandung boraks apabila nilai probabilitas sampel lebih dari 60%.

```

1. import warnings
2. warnings.filterwarnings("ignore")
3. warnings.filterwarnings(action="ignore", category=DeprecationWarning)
4. warnings.filterwarnings(action="ignore", category=FutureWarning)
5. import pickle
6. import sys
7. import os
8. import operator
9. import pandas as pd
10. import numpy as np
11. #from sklearn.preprocessing import StandardScaler
12.
13. def loadModel(nmModel):
14.     f = open(nmModel, 'rb')
15.     model = pickle.load(f)
16.     return model
17.
18. def loadFile(nmFile, kelas):
19.     dfRAW = pd.DataFrame([])
20.     tmpData = pd.read_csv(nmFile, sep=',', encoding='utf
21.     8', engine='python')
21.     nmFile = nmFile[47:]

```

```

22.     tmpData['idFile']=nmFile.replace(".csv","")
23.     dfRAW = dfRAW.append(tmpData)
24.
25.
26.     if 'Timestamp' in dfRAW.columns:
27.         dfRAW = dfRAW.drop(columns=['Timestamp','Panah
28.     '])
29.     dfRAW = dfRAW.convert_objects(convert_numeric=True)
30.     dfRAW = dfRAW.fillna(0)
31.     dfAvg = dfRAW.groupby(['idFile']).agg([np.mean,np.s
32.     td])#, np.std, np.max])
33.     N_fitur=dfAvg.shape[1] ## jumlah kolom
34.     dataValue = dfAvg.values
35.     dataX     = dataValue[:, :N_fitur]
36.     return dataX
37. def doPredict(cmd,nmFile):
38.     path = "/var/www/dewijelek/prg/"
39.     borak=1
40.     if int(cmd) ==1:
41.         nmModel = path + "BAKSO_modelMLP.pkl"
42.         nmPCA    = path + "BAKSO_PCA.pkl"
43.         nmScaler= path + "BAKSO_scaler.pkl"
44.         sampel = "boraks"
45.     else:
46.         nmModel = path + "TAHU_modelSVM.pkl"
47.         nmPCA    = path + "TAHU_PCA.pkl"
48.         nmScaler= path + "TAHU_scaler.pkl"
49.         sampel ="formalin"
50.
51.
52.     if os.path.exists(nmModel):
53.         model = loadModel(nmModel)
54.     else:
55.         return nmModel + " not found, please train!"
56.
57.     if os.path.exists(nmPCA):
58.         pca    = loadModel(nmPCA)
59.     else:
60.         return nmPCA + " not found, please train!"

```

```
61.
62.     if os.path.exists(nmScaler):
63.         scaler = loadModel(nmScaler)
64.     else:
65.         return nmScaler + " not found, please train!"
66.
67.
68.     X_fitur = loadFile(nmFile, borak)
69.     X_fitur = scaler.transform(X_fitur.reshape(1, -
70.     1))
71.     X_PCA = pca.transform(X_fitur)
72.     predicted_proba = model.predict_proba(X_PCA)
73.
74.     prob = -1
75.     for i in range(len(model.classes_)):
76.         res[model.classes_[i]] = predicted_proba[0][i]
77.     res = sorted(res.items(), key=operator.itemgetter(1
78.     ))
79.     res.reverse()
80.     rank = 0
81.     for key, val in res:
82.         rank += 1
83.         if key == borak:
84.             prob = val
85.             break
86.     score = round(prob*100,2)
87.     if prob == -1:
88.         return "Data tidak valid|-|-"
89.     elif (rank <= 5 and score > 60):
90.         return "Terindikasi ada %s|%g|%g"%(sampel,rank,
91.         score)
92.     else:
93.         return "Tidak Terindikasi ada %s|%g|%g"%(sampel
94.         ,rank,score)
95.
96. if __name__=="__main__":
97.     cmd = sys.argv[1]
98.     nmFile = sys.argv[2]
```

```

97.     hasil = doPredict(cmd,nmFile)
98.     print(hasil)

```

### Kode Sumber 4.11 Implementasi Klasifikasi Data Sampel

**Tabel 4.13 Penjelasan Kode Sumber Implementasi Klasifikasi Data Sampel**

No. Baris	Kegunaan
1-10	Melakukan <i>import</i> pada dependency dan modul yang dibutuhkan pada fungsi-fungsi yang dinakan
13	Deklarasi fungsi untuk membuka model data latihan
14-16	Membuka nama model yang dipanggil
18	Deklarasi untuk membuka file yang dipanggil
19	Mendeklarasikan array 2 dimensi dari modul panda
20	Membaca nilai yang ada pada file csv dan memisahkan menjadi kolom-kolom tersendiri
21	Mengambil nama file sesuai yang ditentukan
22-23	Menyisipkan nama file ke variabel dfRAW
26-27	Melakukan pengecekan apakah ada kolom Timestamp dan panah, jika iya, kolom akan dihilangkan
28-29	Mengubah isi kolom menjadi double, jika bukan numerik maka akan otomatis menjadi 0
30	Mengambil standar deviasi dan rata-rata dari setiap kolom sebagai fitur
31	Mendapatkan jumlah kolom dari variabel berisi standar deviasi dan rata-rata setiap kolom
33	Mengambil nilai dari data pada dfAvg
34	Menggabungkan antara dgAvg dengan jumlah fitur
35	Mereturn dataX
37	Deklarasi fungsi untuk predik klasifikasi
38	Mendeklarasi path tempat menyimpan model
39	Mendeklarasikan kelas yang diuji
40-49	Mendeklarasikan file yang harus dibuka sesuai dengan jenis sampel
52-65	Memeriksa apakah file yang akan digunakan ada pada sistem dan mengambil file yang akan digunakan

No. Baris	Kegunaan
68	Mendeklarasikan variabel yang membaca isi dari file yang akan diuji
69	Melakukan transform pada scaler
71	Melakukan transform pada PCA
72	Melakukan perhitungan probabilitas pada proses prediksi
74	Mendeklarasikan variabel prob
75-77	Mengurutkan kelas dari probabilitas yang memiliki nilai terendah
78	Melakukan reverse pada variabel res
79	Mendeklarasikan variabel rank
80-84	Melakukan perhitungan rank
85	Mendeklarasikan score yang merupakan nilai dari probabilitas dikalikan dengan 100
86-91	Membandingkan apabila probabilitas $< 1$ , maka data dianggap tidak valid. Apabila hasil memiliki rank kurang lebih sama dengan 5 dan score lebih besar dari 60, maka sampel terindikasi mengandung boraks. Sebaliknya, maka sampel tidak terindikasi mengandung boraks
94	Deklarasi kelas main
95-96	Membaca argument dari console sebagai parameter jenis sampel dan file yang akan diuji
97	Memanggil fungsi untuk pengujian
98	Mengeluarkan hasil dari proses klasifikasi

#### 4.4. Implementasi API

Pada subbab ini, akan dijelaskan mengenai implementasi dari pembuatan API endpoint dan penjelasan mengenai kode sumber pada perbarisnya.

##### 4.4.1. Implementasi API Mengambil Lokasi Penjualan

Kode sumber implementasi untuk API mengambil lokasi penjualan dapat dilihat pada Kode Sumber 4.12. Sedangkan, untuk penjelasan kode sumber API mengambil lokasi penjualan dijelaskan pada Tabel 4.14.

```

1. public function ambilLokasiPenjualan(Request $request)
2. {
3.     $idJenisSampel = $request-
>input('jenis_sample', '0');
4.     try {
5.         if ($idJenisSampel == 0) {
6.             $result = LokasiPenjualan::where('flag_loka
si_penjualan', '=', 1)-
>where('status_lokasi_penjualan' , 'Buka')
7.                 ->get();
8.         } else {
9.             $result = LokasiPenjualan::where('id_jenis_
sample', '=', $idJenisSampel)
10.            -
>where('flag_lokasi_penjualan', '=', 0)
11.                ->get();
12.        }
13.        // dd(url());
14.        $output = [];
15.        foreach($result as $item){
16.            $output[] = [
17.                'id_lokasi_penjualan' => $item-
>id_lokasi_penjualan,
18.                'nama_lokasi_penjualan' => $item-
>nama_lokasi_penjualan,
19.                'latitude_lokasi_penjualan' => $item-
>latitude_lokasi_penjualan,
20.                'longitude_lokasi_penjualan'=> $item-
>longitude_lokasi_penjualan,
21.                'status_klasifikasi'=> $item-
>status_klasifikasi,
22.                'rank_klasifikasi'=> $item-
>rank_klasifikasi,
23.                'rank_klasifikasi'=> $item-
>rank_klasifikasi,
24.                'score_klasifikasi'=> $item-
>score_klasifikasi,
25.                'foto_lokasi_penjualan'=>url('/img/loka
si-penjualan/' . $item->foto_lokasi_penjualan),
26.                'id_jenis_sample' => $item-
>id_jenis_sample,

```

```

27.             'created_at' => $item->created_at,
28.             'updated_at' => $item->updated_at,
29.
30.         ];
31.     }
32. } catch (QueryException | Exception $e) {
33.     return response()->json([
34.         'code' => 500,
35.         'message' => $e->errorInfo[2],
36.         'data' => null,
37.     ]);
38. }
39.
40. return response()->json([
41.     'code' => 200,
42.     'message' => 'Berhasil mendapatkan data',
43.     'data' => $output,
44. ]);
45. }

```

#### Kode Sumber 4.12 Implementasi API Mengambil Lokasi Penjualan

**Tabel 4.14 Penjelasan Kode Sumber Implementasi API Mengambil Lokasi Penjualan**

No. Baris	Kegunaan
1	Mendeklarasikan method public, agar bisa diakses dari luar class
3	Mendeklarasikan id jenis sampel yang didapatkan dari parameter request dari user , apabila null maka id jenis sampel otomatis menjadi 0
4	Memulai try catch untuk query exception dan exception general
5	Memeriksa apakah id jenis sampel sesuai dengan kondisi
6-7	Mengambil data lokasi penjualan yang memenuhi kondisi sebelumnya
9-11	Mengambil data lokasi penjualan yang tidak memenuhi kondisi yang dideklarasasi sebelumnya



No. Baris	Kegunaan
14	Mendeklarasikan array untuk menyimpan hasil dari mendapatkan lokasi penjualan
15-31	Mendeklarasikan atribut-atribut yang akan dikirim sebagai <i>response</i>
32-37	Mendeklarasikan <i>response</i> yang dikirim apabila terjadi eksepsi
40-44	Mendeklarasikan <i>response</i> yang dikirim apabila berhasil mendapatkan data lokasi penjualan

#### 4.4.2. Implementasi API Menambah Lokasi Penjualan

Kode sumber implementasi untuk API menambah lokasi penjualan dapat dilihat pada Kode Sumber 4.13. Sedangkan, untuk penjelasan kode sumber API menambah lokasi penjualan dijelaskan pada Tabel 4.15.

```

1. public function tambahLokasiPenjualan(Request $request)
   {
2.     $validate = [
3.         'remember_token' => 'required',
4.         'nama_lokasi_penjualan' => 'required|string',
5.         'latitude_lokasi_penjualan' => 'required',
6.         'longitude_lokasi_penjualan'=> 'required',
7.         'foto_lokasi_penjualan'=> 'required',
8.         'id_jenis_sample' => 'required'
9.     ];
10.    $validator = Validator::make($request-
        >all(), $validate);
11.
12.    if ($validator->fails()) {
13.        return response()->json([
14.            'code' => 400,
15.            'message' => 'Isian tidak boleh kosong',
16.            'data' => null,
17.        ]);
18.    }
19.    $user = User::where('remember_token', $request-
        >remember_token)->first();
20.    if(is_null($user)){
21.        return response()->json([

```

```

22.         'code' => 403,
23.         'message' => 'Tidak memiliki akses / Token
    salah',
24.         'data' => null,
25.     });
26. }
27.     $namaSudahAda = LokasiPenjualan::where('id_lokasi_p
    enjualan', '!=', $request->id_lokasi_penjualan)
28.         ->where('nama_lokasi_penjualan', $request-
    >nama_lokasi_penjualan)
29.         ->where('id_jenis_sample', $request-
    >id_jenis_sample)->first();
30.
31.     if(!is_null($namaSudahAda)){
32.         return response()->json([
33.             'code' => 400,
34.             'message' => 'Sudah ada nama yang sama di l
    atitude dan longitude yang sama',
35.             'data' => null,
36.         ]);
37.     }
38.     try {
39.         DB::beginTransaction();
40.         try {
41.             $lokasiPenjualan = new LokasiPenjualan();
42.             $lokasiPenjualan-
    >nama_lokasi_penjualan = $request-
    >nama_lokasi_penjualan;
43.             $lokasiPenjualan-
    >latitude_lokasi_penjualan = $request-
    >latitude_lokasi_penjualan;
44.             $lokasiPenjualan-
    >longitude_lokasi_penjualan = $request-
    >longitude_lokasi_penjualan;
45.             $lokasiPenjualan-
    >id_jenis_sample = $request->id_jenis_sample;
46.
47.             //proses simpan file
48.             $image = $request->foto_lokasi_penjualan;
49.             $image = str_replace('data:image/jpeg;base6
    4,', ' ', $image);
50.             $image = str_replace(' ', '+', $image);

```

```

51.         $imageName = Carbon::now()-
>format('YmdHis').'_'.$request-
>id_jenis_sample.'.jpg';
52.         \File::put(public_path(). '//img//lokasi-
penjualan//' . $imageName, base64_decode($image));
53.         $lokasiPenjualan-
>foto_lokasi_penjualan = $imageName;
54.         $lokasiPenjualan->save();
55.         DB::commit();
56.
57.         return response()->json([
58.             'code' => 200,
59.             'message' => "Data berhasil disimpan",
60.             'data' => $lokasiPenjualan,
61.         ]);
62.     } catch (Exception $e) {
63.         DB::rollback();
64.     }
65. } catch (QueryException | Exception $e) {
66.     return response()->json([
67.         'code' => 500,
68.         'message' => $e->errorInfo[2],
69.         'data' => null,
70.     ]);
71. }
72. }

```

### Kode Sumber 4.13 Implementasi API Menambah Lokasi Penjualan

**Tabel 4.15** Penjelasan Kode Sumber Implementasi API Menambah Lokasi Penjualan

No. Baris	Kegunaan
1	Mendeklarasikan method public, agar bisa diakses dari luar class
3-10	Melakukan validasi dari parameter-parameter masukan pada request API
12-18	Mendeklarasikan <i>response</i> yang dikirim apabila validasi gagal
19	Memeriksa kredensial dari user

No. Baris	Kegunaan
21-25	Mendeklarasikan <i>response</i> yang dikirim apabila user tidak memiliki akses
27-29	Melakukan pengecekan apabila ada nama yang sama pada latitude dan longitude yang sama
31-36	Mendeklarasikan <i>response</i> yang dikirim apabila request memiliki nama lokasi penjualan yang sama pada latitude dan longitude yang sama
38-39	Memulai try catch untuk transaksi <i>database</i>
41-54	Proses pembuatan objek lokasi penjualan baru
55	Melakukan <i>commit</i> pada database apabila proses pembuatan berhasil
57-61	Mendeklarasikan <i>response</i> yang dikirim apabila berhasil menambah lokasi penjualan
62-64	Menangkap eksepsi apabila terjadi kesalahan pada transaksi <i>database</i> , proses transaksi pada <i>database</i> akan dianulir
65-69	Menangkap eksepsi apabila terjadi kesalahan pada query eksepsi

#### 4.4.3. Implementasi API Menambah Laporan Mengenai Lokasi Penjualan

Kode sumber implementasi untuk API menambah laporan mengenai lokasi penjualan dapat dilihat pada Kode Sumber 4.14. Sedangkan, untuk penjelasan kode sumber API menambah laporan mengenai lokasi penjualan dijelaskan pada Tabel 4.16.

```

1. public function tambahLaporan(Request $request){
2.     $validate = [
3.         'nama_pelapor' => 'required|string',
4.         'latitude_lokasi_penjualan' => 'required',
5.         'longitude_lokasi_penjualan'=> 'required',
6.         'isi_laporan'=> 'required',
7.     ];
8.     $validator = Validator::make($request-
9.         >all(), $validate);
10.     if ($validator->fails()) {

```

```
11.         return response()->json([
12.             'code' => 400,
13.             'message' => 'Isian tidak boleh kosong',
14.             'data' => null,
15.         ]);
16.     }
17.
18.     try {
19.         DB::beginTransaction();
20.         try {
21.             $laporan = new Laporan();
22.             $laporan->nama_pelapor = $request-
23. >nama_pelapor;
24.             $laporan-
25. >latitude_lokasi_penjualan = $request-
26. >latitude_lokasi_penjualan;
27.             $laporan-
28. >longitude_lokasi_penjualan = $request-
29. >longitude_lokasi_penjualan;
30.             $laporan->isi_laporan = $request-
31. >isi_laporan;
32.             $laporan->created_at = Carbon::now();
33.             $laporan->save();
34.             DB::commit();
35.             event(new LaporanPosted($request-
36. >nama_pelapor, $laporan->created_at, $request-
37. >isi_laporan));
38.             return response()->json([
39.                 'code' => 200,
40.                 'message' => "Data berhasil disimpan",
41.                 'data' => $laporan,
42.             ]);
43.         } catch (Exception $e) {
44.             DB::rollback();
45.         }
46.     } catch (QueryException | Exception $e) {
47.         return response()->json([
48.             'code' => 500,
49.             'message' => $e->errorInfo[2],
50.             'data' => null,
```

```

44.         });
45.     }
46. }

```

#### Kode Sumber 4.14 Implementasi API Menambah Laporan Mengenai Lokasi Penjualan

**Tabel 4.16 Penjelasan Kode Sumber Implementasi API Menambah  
Laporan Mengenai Lokasi Penjualan**

No. Baris	Kegunaan
1	Mendeklarasikan method public, agar bisa diakses dari luar class
2-8	Melakukan validasi dari parameter-parameter masukan pada request API
10-16	Mendeklarasikan <i>response</i> yang dikirim apabila validasi gagal
18-19	Memulai try catch untuk transaksi <i>database</i>
21-27	Proses pembuatan objek laporan baru
28	Melakukan <i>commit</i> pada database apabila proses pembuatan berhasil
29	Memanggil event untuk menambahkan notifikasi adanya laporan baru yang masuk secara <i>realtime</i>
30-34	Mendeklarasikan <i>response</i> yang dikirim apabila berhasil menambah laporan
36-37	Menangkap eksepsi apabila terjadi kesalahan pada transaksi <i>database</i> , proses transaksi pada <i>database</i> akan dianulir
39-44	Menangkap eksepsi apabila terjadi kesalahan pada query eksepsi

## BAB V

### UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi kasus penggunaan, klasifikasi sampel, dan API Endpoint.

#### 5.1. Lingkungan Uji Coba

Lingkungan uji coba adalah kombinasi antara perangkat keras dan perangkat lunak yang digunakan untuk melakukan uji coba. Pengujian dilakukan dengan menggunakan tiga buah lingkungan pengujian yaitu lingkungan pengujian aplikasi, *server* basis data, dan *server* aplikasi *web*. Adapun rincian dari masing-masing lingkungan pengujian tersebut secara berturut-turut ditunjukkan pada Tabel 5.1 dan Tabel 5.2.

**Tabel 5.1 Lingkungan Pengujian Aplikasi**

Spesifikasi	Deskripsi
CPU	Prosesor: Intel® Core™ i7-7700HQ CPU @ 2.80GHz (4 CPUs) , ~2.81GHz
RAM	16.00 GB
Sistem Operasi	Windows 10 Education 64-bit
Browser	Google Chrome versi 79

**Tabel 5.2 Lingkungan Server Aplikasi Web**

Spesifikasi	Deskripsi
CPU	Intel® Core™ i5-3337U CPU @ 1.80GHz
RAM	4.00 GB
Sistem Operasi	Windows 10 Education 64-bit
Server Web	Apache 2.4.25

#### 5.2. Skenario Pengujian Kasus Penggunaan

Pada bagian ini akan dibahas mengenai proses uji coba pada kasus penggunaan yang digunakan. Pengujian dilakukan dengan metode *black box* untuk menguji masing-masing fungsionalitas yang sudah dirancang pada sistem. Metode *black box* merupakan

metode pengujian perangkat lunak yang memeriksa fungsionalitas dari suatu perangkat lunak tanpa memandang struktur internalnya.

Pada proses uji coba, pengujian dilakukan dengan menjalankan serangkaian perintah terhadap sistem yang selanjutnya akan disebut sebagai kasus pengujian. Kasus pengujian ini berkorelasi dengan kasus-kasus penggunaan dan kebutuhan fungsional yang sebelumnya sudah dirancang dan dijelaskan pada Bab III.

### 5.2.1. Pengujian Melihat Lokasi Penjualan

Pengujian melihat lokasi penjualan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan lokasi penjualan dalam bentuk peta. Rincian kasus pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.3. Gambar contoh hasil skenario dapat dilihat pada Gambar 5.1.

**Tabel 5.3 Kasus Pengujian Melihat Lokasi Penjualan**

<b>ID</b>	UJUC1
<b>Nama</b>	Pengujian Melihat Lokasi Penjualan
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menampilkan lokasi penjualan
<b>Kondisi Awal</b>	Data lokasi penjualan belum ditampilkan namun sudah tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	1. Aktor memilih menu beranda
<b>Hasil yang diharapkan</b>	Data lokasi penjualan tampil dalam bentuk peta
<b>Hasil yang diperoleh</b>	Data lokasi penjualan tampil dalam bentuk peta
<b>Hasil Pengujian</b>	Berhasil

### 5.2.2. Pengujian Mengelola Lokasi Penjualan

Pengujian mengelola lokasi penjualan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan daftar lokasi penjualan, menghapus, dan mengubah data lokasi penjualan.



Beranda

Warna lokasi:	Keterangan:
Merah	Lokasi terindikasi menjual bakso berkontungan boraks
Kuning	Lokasi tidak terindikasi menjual bakso berkontungan boraks
Oranye	Lokasi terindikasi menjual tahu berkontungan formalin
Biru	Lokasi tidak terindikasi menjual tahu berkontungan formalin
Hijau	Data sampel tidak valid

**Gambar 5.1 Tampilan Hasil Pengujian Melihat Lokasi Penjualan**

Rincian kasus pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.4. Gambar contoh hasil skenario dapat dilihat pada Gambar 5.2, Gambar 5.3, Gambar 5.4, Gambar 5.5, Gambar 5.6, dan Gambar 5.7.

**Tabel 5.4 Kasus Pengujian Mengelola Lokasi Penjualan**

<b>ID</b>	UJUC02
<b>Nama</b>	Pengujian Mengelola Lokasi Penjualan
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu melakukan pengelolaan pada lokasi penjualan
<b>Skenario 1</b>	Administrator memilih menampilkan daftar lokasi penjualan sesuai dengan jenis sampel
<b>Kondisi Awal</b>	Data lokasi penjualan belum ditampilkan namun sudah tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lokasi penjualan</li> <li>2. Aktor memilih salah satu jenis sampel</li> </ol>
<b>Hasil yang diharapkan</b>	Data lokasi penjualan tampil dalam bentuk tabel
<b>Hasil yang diperoleh</b>	Data lokasi penjualan tampil dalam bentuk tabel
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 2</b>	Administrator memilih menghapus data lokasi penjualan yang tidak memiliki data sampel
<b>Kondisi Awal</b>	Data lokasi penjualan tersimpan di <i>database</i> dan belum ada data sampel yang mereferensikan data lokasi penjualan yang dipilih
<b>Data Uji</b>	Lokasi “Pasar Bendul Merisi”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan hapus</li> <li>2. Aktor memilih tombol hapus</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem kembali ke halaman lokasi penjualan dan menampilkan notifikasi berhasil menghapus data dan data lokasi penjualan terhapus dari <i>database</i>
<b>Hasil yang diperoleh</b>	Sistem kembali ke halaman lokasi penjualan dan menampilkan notifikasi berhasil menghapus

	data dan data lokasi penjualan terhapus dari <i>database</i>
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 3</b>	Administrator memilih menghapus data lokasi penjualan yang memiliki data sampel
<b>Kondisi Awal</b>	Data lokasi penjualan tersimpan di <i>database</i> dan sudah ada data sampel yang mereferensikan data lokasi penjualan yang dipilih
<b>Data Uji</b>	Lokasi “Pasar Pucang Anom”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan hapus</li> <li>2. Aktor memilih tombol hapus</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem kembali ke halaman lokasi penjualan dan menampilkan notifikasi gagal menghapus
<b>Hasil yang diperoleh</b>	Sistem kembali ke halaman lokasi penjualan dan menampilkan notifikasi gagal menghapus
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 4</b>	Administrator memilih mengubah data lokasi penjualan dengan isian tidak lengkap
<b>Kondisi Awal</b>	Data lokasi penjualan tersimpan di <i>database</i> dan ditampilkan di halaman lokasi penjualan
<b>Data Uji</b>	Lokasi “Pasar Pucang Anom”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan ubah</li> <li>2. Aktor mengisi form dengan tidak lengkap</li> <li>3. Aktor memilih tombol ubah</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan peringatan pada isian yang wajib diisi
<b>Hasil yang diperoleh</b>	Sistem menampilkan peringatan pada isian yang wajib diisi
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 5</b>	Administrator memilih mengubah data lokasi penjualan dengan isian lengkap dan nama baru sama dengan nama yang sudah ada pada latitude dan longitude yang sama
<b>Kondisi Awal</b>	Data lokasi penjualan tersimpan di <i>database</i> dan ditampilkan di halaman lokasi penjualan
<b>Data Uji</b>	Lokasi “Pasar Pucang Anom 2”

<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan ubah</li> <li>2. Aktor mengisi form dengan lengkap dan atribut nama sama dengan nama yang sudah ada pada latitude dan longitude yang sama</li> <li>3. Aktor memilih tombol ubah</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem kembali ke halaman lokasi penjualan dan menampilkan notifikasi gagal mengubah
<b>Hasil yang diperoleh</b>	Sistem kembali ke halaman lokasi penjualan dan menampilkan notifikasi gagal mengubah
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 6</b>	Administrator memilih mengubah data lokasi penjualan dengan isian lengkap dan nama baru tidak sama dengan nama yang sudah ada pada latitude dan longitude yang sama
<b>Kondisi Awal</b>	Data lokasi penjualan tersimpan di <i>database</i> dan ditampilkan di halaman lokasi penjualan
<b>Data Uji</b>	Lokasi “Pasar Pucang Anom 2”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan ubah</li> <li>2. Aktor mengisi form dengan lengkap dan atribut nama tidak sama dengan nama yang sudah ada pada latitude dan longitude yang sama</li> <li>3. Aktor memilih tombol ubah</li> </ol>
<b>Hasil yang diharapkan</b>	Data lokasi penjualan yang dirubah disimpan dan sistem menampilkan notifikasi data berhasil diperbarui.
<b>Hasil yang diperoleh</b>	Data lokasi penjualan yang dirubah disimpan dan sistem menampilkan notifikasi data berhasil diperbarui.
<b>Hasil Pengujian</b>	Berhasil

Lokasi Penjualan Bakso Lokasi Penjualan / Bakso

No.	Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sample	Kelas	Status	Opsi
1.	Ps. Pucang Anom	18-11-2019	Ada	Terindikasi ada formalin	Buka	Lihat Data Sample, Ubah, Hapus
2.	Pasar Bendul/Merisi	27-12-2019	Tidak Ada		Buka	Lihat Data Sample, Ubah, Hapus

**Gambar 5.2 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 1**

Sukses! Berhasil menghapus data lokasi penjualan ✕

No.	Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sample	Kelas
1.	Ps. Pucang Anom	18-11-2019	Ada	Terindikasi ada formal

**Gambar 5.3 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 2**


Error! Tidak bisa menghapus. Terdapat data sample di lokasi penjualan tersebut ✕

No.	Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sample	Kelas
1.	Ps. Pucang Anom	18-11-2019	Ada	Terindikasi ada formal

**Gambar 5.4 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 3**



**Gambar 5.5 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 4**



No.	Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sample	Kelas
1.	Ps. Pucang Anom	18-11-2019	Ada	Terindikasi ada form
2.	Ps. Pucang Anom 2	04-01-2020	Tidak Ada	

**Gambar 5.6 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 5**



No.	Nama Lokasi	Tanggal Pengambilan Lokasi	Data Sample	Kelas
1.	Ps. Pucang Anom	18-11-2019	Ada	Terindikasi ada fo
2.	Ps. Pucang Anom Edit	04-01-2020	Tidak Ada	

**Gambar 5.7 Tampilan Hasil Pengujian Pengujian Mengelola Lokasi Penjualan Skenario 6**

### 5.2.3. Pengujian Menambah Data Sampel

Pengujian menambah data sampel merupakan pengujian terhadap kemampuan aplikasi menambahkan data sampel sesuai

lokasi yang sudah ada dan melakukan proses klasifikasi pada data sampel tersebut. Rincian kasus pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.5. Gambar contoh hasil skenario dapat dilihat pada Gambar 5.8, Gambar 5.9, dan Gambar 5.10.

**Tabel 5.5 Kasus Pengujian Menambah Data Sampel**

<b>ID</b>	UJUC03
<b>Nama</b>	Pengujian Menambah Data Sampel
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menambahkan data sampel sesuai dengan lokasi penjualan yang sudah ada
<b>Skenario 1</b>	Administrator menambahkan data sampel dengan isian tidak lengkap
<b>Kondisi Awal</b>	Data lokasi penjualan telah tersimpan di <i>database</i> dan data sampel belum tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu tambah data sampel</li> <li>2. Aktor mengisi form dengan tidak lengkap</li> <li>3. Aktor memilih simpan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan peringatan pada isian yang wajib diisi
<b>Hasil yang diperoleh</b>	Sistem menampilkan peringatan pada isian yang wajib diisi
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 2</b>	Administrator menambahkan data sampel dengan isian lengkap namun format file data sampel bukan “csv”
<b>Data Uji</b>	File “001_BAKSO TIGA PUTRI_BENDUL MERISI_1.xlsx”
<b>Kondisi Awal</b>	Data lokasi penjualan telah tersimpan di <i>database</i> dan data sampel belum tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu tambah data sampel</li> <li>2. Aktor mengisi form dengan format file yang diunggah bukan “csv”</li> <li>3. Aktor memilih simpan</li> </ol>

<b>Hasil yang diharapkan</b>	Sistem menampilkan error bahwa format yang diterima hanya “csv”
<b>Hasil yang diperoleh</b>	Sistem menampilkan error bahwa format yang diterima hanya “csv”
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 3</b>	Administrator menambahkan data sampel dengan isian lengkap dan format file data sampel “csv”
<b>Data Uji</b>	Lokasi “Giant Arief Rahman – Raja Baoo” dan File “Giant Arief Rahman.csv”
<b>Kondisi Awal</b>	Data lokasi penjualan telah tersimpan di <i>database</i> dan data sampel belum tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu tambah data sampel</li> <li>2. Aktor mengisi form dengan format file yang diunggah berformat “csv”</li> <li>3. Aktor memilih simpan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan notifikasi sukses data telah ditambahkan dan status klasifikasi dari data sampel dan data sampel tersimpan di <i>database</i>
<b>Hasil yang diperoleh</b>	Sistem menampilkan notifikasi sukses data telah ditambahkan dan status klasifikasi dari data sampel dan data sampel tersimpan di <i>database</i>
<b>Hasil Pengujian</b>	Berhasil

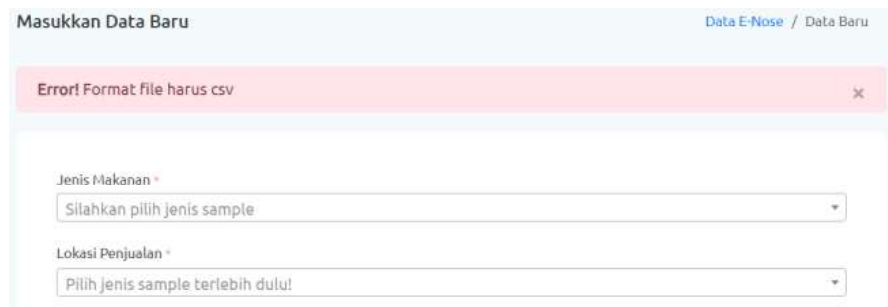
The screenshot shows a web form titled "Masukkan Data Baru" (Add New Data). The form contains several input fields:

- Jenis Makanan:** A dropdown menu with the placeholder text "Silahkan pilih jenis sample". A yellow warning box with an exclamation mark icon and the text "Please select an item in the list." is displayed over this field.
- Lokasi Penjualan:** A dropdown menu with the placeholder text "Pilih jenis sample terlebih dulu".
- Tanggal Pengambilan Data Sample:** A date input field with the placeholder "mm/dd/yyyy".
- File:** A section for file upload with a "Pilih File" button.

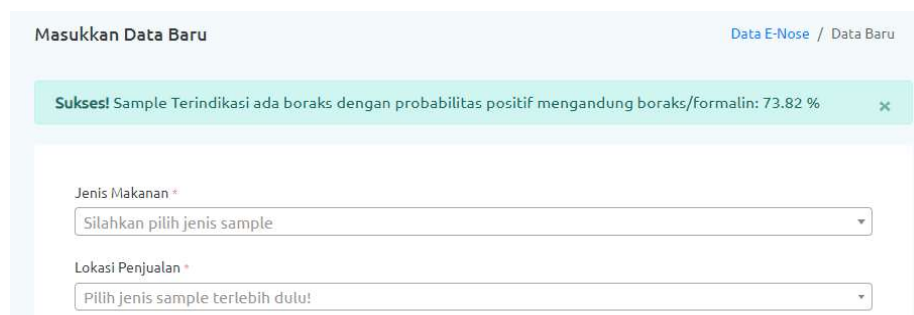
At the bottom of the form, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

**Gambar 5.8 Tampilan Hasil Pengujian Menambah Data Sampel Skenario 1**





**Gambar 5.9 Tampilan Hasil Pengujian Menambah Data Sampel Skenario 2**



**Gambar 5.10 Tampilan Hasil Pengujian Menambah Data Sampel Skenario 3**

#### 5.2.4. Pengujian Mengelola Data Sampel

Pengujian mengelola data sampel merupakan pengujian terhadap kemampuan aplikasi menampilkan, menghapus, dan mengubah data sampel sesuai lokasi yang sudah ada. Rincian kasus pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.6. Gambar contoh hasil skenario dapat dilihat pada Gambar 5.11, Gambar 5.12, Gambar 5.13, Gambar 5.14, dan Gambar 5.15.

**Tabel 5.6 Kasus Mengelola Data Sampel**

<b>ID</b>	UJUC04
<b>Nama</b>	Pengujian Mengelola Data Sampel
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menampilkan, mengubah, dan menghapus data

	sampel sesuai dengan lokasi penjualan yang sudah ada
<b>Skenario 1</b>	Administrator melihat data sampel
<b>Kondisi Awal</b>	Data sampel telah tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lokasi penjualan</li> <li>2. Aktor memilih jenis sampel</li> <li>3. Aktor memilih lihat data sampel pada salah satu data lokasi penjualan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan data sampel
<b>Hasil yang diperoleh</b>	Sistem menampilkan data sampel
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 2</b>	Administrator melakukan perubahan data sampel dengan isian tidak lengkap
<b>Data Uji</b>	Lokasi penjualan “Pasar Bendul Merisi”
<b>Kondisi Awal</b>	Data sampel telah tersimpan di <i>database</i> dan belum mengalami perubahan
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan ubah</li> <li>2. Aktor mengkonfirmasi untuk mengubah data sampel</li> <li>3. Aktor mengisi form perubahan data sampel dengan tidak lengkap</li> <li>4. Aktor menyimpan perubahan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan peringatan pada isian yang wajib diisi
<b>Hasil yang diperoleh</b>	Sistem menampilkan peringatan pada isian yang wajib diisi
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 3</b>	Administrator melakukan perubahan data dengan ekstensi file bukan “csv”
<b>Data Uji</b>	Lokasi penjualan “Pasar Bendul Merisi”
<b>Kondisi Awal</b>	Data sampel telah tersimpan di <i>database</i> dan belum mengalami perubahan
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan ubah</li> </ol>

	<ol style="list-style-type: none"> <li>2. Aktor mengkonfirmasi untuk mengubah data sampel</li> <li>3. Aktor mengisi form perubahan data sampel dengan ekstensi file bukan “csv”</li> <li>4. Aktor menyimpan perubahan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan error bahwa format yang diterima hanya “csv”
<b>Hasil yang diperoleh</b>	Sistem menampilkan error bahwa format yang diterima hanya “csv”
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 4</b>	Administrator melakukan perubahan data dengan ekstensi file “csv” dan isian lengkap
<b>Data Uji</b>	Lokasi penjualan “Giant Arief Rahman - Raja Baoo” dan file “Keputih 2.csv”
<b>Kondisi Awal</b>	Data sampel telah tersimpan di <i>database</i> dan belum mengalami perubahan
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan ubah</li> <li>2. Aktor mengkonfirmasi untuk mengubah data sampel</li> <li>3. Aktor mengisi form perubahan data sampel dengan ekstensi file “csv ” dan isian lengkap</li> <li>4. Aktor menyimpan perubahan</li> </ol>
<b>Hasil yang diharapkan</b>	Data sampel yang dirubah disimpan dan sistem menampilkan notifikasi data berhasil diperbarui dan hasil dari proses klasifikasi data sampelnya
<b>Hasil yang diperoleh</b>	Data sampel yang dirubah disimpan dan sistem menampilkan notifikasi data berhasil diperbarui dan hasil dari proses klasifikasi data sampelnya
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 5</b>	Administrator melakukan penghapusan data sampel
<b>Data Uji</b>	Lokasi penjualan “Pasar Bendul Merisi”
<b>Kondisi Awal</b>	Data sampel telah tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih pilihan hapus data sampel</li> </ol>

	2. Aktor mengkonfirmasi untuk menghapus data sampel
<b>Hasil yang diharapkan</b>	Data sampel terhapus dari <i>database</i>
<b>Hasil yang diperoleh</b>	Data sampel terhapus dari <i>database</i>
<b>Hasil Pengujian</b>	Berhasil

Detail Data Sample | Bakso Lokasi Penjualan Bakso / Superindo Arief Rahman Hakim - Bakso Urat Tanpa Merk

[Lihat Peta Lokasi Penjualan](#)
[Lihat Foto Lokasi Penjualan](#)
[Hapus Data Sample](#)
[Ubah Data Sample](#)

Superindo Arief Rahman Hakim - Bakso Urat Tanpa Merk

Latitude:	-7.2896632	Tanggal Pengambilan Lokasi:	2020-01-05 03:02:32	Tanggal Pengambilan Sample:	2020-01-06
Longitude:	112.78786705	Kelas Klasifikasi:	Tidak terindikasi ada boraks	Tingkat probabilitas positif mengandung boraks/formalin:	11.24 %

No	mq2_jng	mq2_cn	mq2_cmske	mq2_alrohnl	mq2_rhd	mq2_h2	mq2_propane	mq3_alrohnl	mq3_hoxzine	mq3_rhd	mq3_cn	mq3_hxvane	mq3_jng	mq4_alrohnl	mq4
1.	5.17	41.01	22.75	9.4	12.53	8.89	5.28	0	0	0	0.08	0.07	0.08	98.71	16.
2.	3.68	25.54	15.69	6.14	8.22	6.34	3.74	0	0	0	0.04	0.04	0.04	10.57	10.
3.	3.57	24.46	15.16	5.9	7.91	6.15	3.62	0	0	0	0.03	0.03	0.04	8.08	9.
4.	3.6	24.78	15.32	5.97	8	6.21	3.65	0	0	0	0.03	0.03	0.03	7.77	9.
5.	3.64	25.11	15.40	6.04	0.1	6.27	3.69	0	0	0	0.03	0.03	0.03	6.02	9.
6.	3.64	25.10	15.51	6.06	8.12	6.28	3.7	0	0	0	0.03	0.03	0.03	8.08	9.
7.	3.65	25.25	15.54	6.07	8.14	6.29	3.7	0	0	0	0.03	0.03	0.03	7.87	9.
8.	3.64	25.21	15.52	6.06	8.13	6.28	3.7	0	0	0	0.03	0.03	0.03	7.83	9.
9.	3.63	25.09	15.47	6.04	8.09	6.26	3.69	0	0	0	0.03	0.03	0.03	7.61	9.
10.	3.62	25.01	15.43	6.02	8.07	6.25	3.68	0	0	0	0.03	0.03	0.03	7.27	9.
11.	3.61	24.89	15.37	6	8.03	6.23	3.67	0	0	0	0.03	0.03	0.03	7.01	9.
12.	3.58	24.6	15.23	5.93	7.95	6.17	3.63	0	0	0	0.03	0.03	0.03	6.96	9.
13.	3.58	24.54	15.2	5.92	7.93	6.16	3.63	0	0	0	0.02	0.03	0.03	6.58	9.
14.	3.55	24.32	15.09	5.87	7.87	6.12	3.6	0	0	0	0.03	0.03	0.03	6.42	9.
15.	3.54	24.22	15.05	5.85	7.84	6.11	3.59	0	0	0	0.03	0.03	0.03	6.39	9.
16.	3.51	23.95	14.91	5.79	7.76	6.06	3.56	0	0	0	0.03	0.03	0.03	6.21	8.

**Gambar 5.11 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 1**

**Unggah Data sample** ✕

Tanggal Pengambilan Data Sample \*

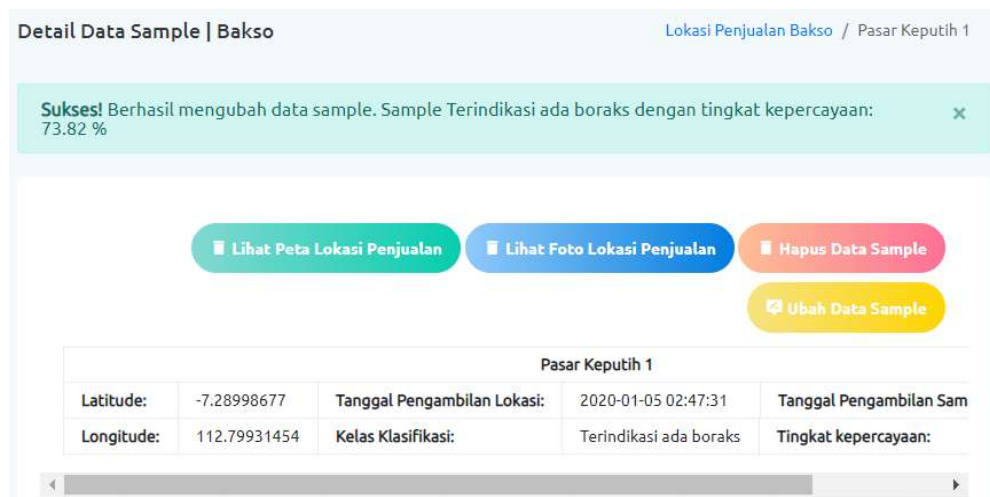
File \*

\* Wajib diisi

**Gambar 5.12 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 2**



**Gambar 5.13 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 3**



**Gambar 5.14 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 4**



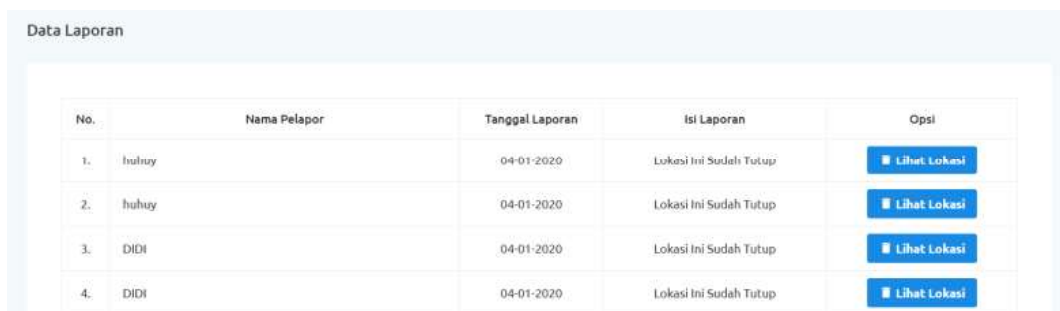
**Gambar 5.15 Tampilan Hasil Pengujian Mengelola Data Sampel Skenario 5**

### 5.2.5. Pengujian Melihat Laporan Mengenai Lokasi Penjualan

Pengujian melihat laporan mengenai lokasi penjualan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan laporan mengenai lokasi penjualan yang masuk. Rincian kasus pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.7. Gambar contoh hasil skenario dapat dilihat pada Gambar 5.16.

**Tabel 5.7 Kasus Pengujian Melihat Laporan Mengenai Lokasi Penjualan**

<b>ID</b>	UJUC05
<b>Nama</b>	Pengujian Melihat Laporan Mengenai Lokasi Penjualan
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menampilkan laporan mengenai lokasi penjualan
<b>Kondisi Awal</b>	Data laporan mengenai lokasi penjualan belum ditampilkan namun sudah tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	1. Aktor memilih menu laporan
<b>Hasil yang diharapkan</b>	Data laporan tampil
<b>Hasil yang diperoleh</b>	Data laporan tampil
<b>Hasil Pengujian</b>	Berhasil



No.	Nama Pelapor	Tanggal Laporan	Isi Laporan	Opsi
1.	huhuy	04-01-2020	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>
2.	huhuy	04-01-2020	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>
3.	DIDI	04-01-2020	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>
4.	DIDI	04-01-2020	Lokasi Ini Sudah Tutup	<a href="#">Lihat Lokasi</a>

**Gambar 5.16 Tampilan Hasil Pengujian Melihat Laporan Mengenai Lokasi Penjualan**

### 5.2.6. Pengujian Melihat Notifikasi

Pengujian melihat notifikasi merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan notifikasi apabila terdapat laporan mengenai lokasi penjualan yang masuk. Rincian kasus pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.8. Gambar contoh hasil skenario dapat dilihat pada Gambar 5.17.

**Tabel 5.8 Kasus Pengujian Melihat Notifikasi**

<b>ID</b>	UJUC06
<b>Nama</b>	Pengujian Melihat Notifikasi
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menampilkan notifikasi
<b>Kondisi Awal</b>	Data notifikasi telah tersimpan di <i>database</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor konsumen melakukan kasus penggunaan menambah laporan pada <i>platform</i> Android</li> <li>2. Aktor memilih menu notifikasi</li> </ol>
<b>Hasil yang diharapkan</b>	Data notifikasi dapat ditampilkan
<b>Hasil yang diperoleh</b>	Data notifikasi dapat ditampilkan
<b>Hasil Pengujian</b>	Berhasil



**Gambar 5.17 Tampilan Hasil Pengujian Pengujian Melihat Notifikasi**

### 5.3. Skenario Pengujian API

Pada pengujian API ini akan dilakukan pengujian terhadap semua API yang telah dibuat. Terdapat dua macam pengujian, yaitu pengujian fungsionalitas (tanpa skenario) dan pengujian dengan menggunakan skenario. Pengujian fungsionalitas dilakukan pada *endpoint* yang tidak membutuhkan pengiriman parameter, sedangkan pengujian yang menggunakan skenario dilakukan pada semua *endpoint* yang membutuhkan pengiriman parameter.

#### 5.3.1. Pengujian API Mengambil Lokasi Penjualan

Detil pengujian mengambil data lokasi penjualan dijelaskan pada Tabel 5.9. Hasil dari pengujian terdapat pada Gambar 5.18.

```

1  {
2    "code": 200,
3    "message": "Berhasil mendapatkan data",
4    "data": [
5      {
6        "id_lokasi_penjualan": 1,
7        "nama_lokasi_penjualan": "Ps. Pucang Anom",
8        "latitude_lokasi_penjualan": "-7.2841",
9        "longitude_lokasi_penjualan": "112.7535",
10       "status_klasifikasi": "Terindikasi ada formalin",
11       "rank_klasifikasi": "1",
12       "score_klasifikasi": "100\n",
13       "foto_lokasi_penjualan": "http://10.151.31.192/img/lokasi-penjualan",
14       "id_jenis_sample": 2,
15       "created_at": "2019-11-18T16:31:26.000000Z",
16       "updated_at": "2019-12-26T08:18:05.000000Z"
17     }
18   ]
19 }

```

**Gambar 5.18 Hasil Pengujian API Mengambil Lokasi Penjualan**

**Tabel 5.9 Pengujian API Mengambil Lokasi Penjualan**

<b>ID</b>	UJAPI01
<b>Nama</b>	Pengujian API Mengambil Lokasi Penjualan
<b>Endpoint</b>	/lokasi-penjualan/get



<b>{http://10.151.31.19 2/api}</b>	
<b>Method</b>	GET
<b>Kondisi Awal</b>	Data lokasi penjualan telah tersimpan di <i>database</i>
<b>Hasil yang diharapkan</b>	Data lokasi penjualan yang telah memiliki status klasifikasi dan berstatus buka tampil
<b>Hasil yang diperoleh</b>	Data lokasi penjualan yang telah memiliki status klasifikasi dan berstatus buka tampil
<b>Hasil Pengujian</b>	Berhasil

### 5.3.2. Pengujian API Menambah Lokasi Penjualan

Detil pengujian menambah data lokasi penjualan dijelaskan pada Tabel 5.10. Hasil dari pengujian terdapat pada Gambar 5.19, Gambar 5.20, dan Gambar 5.21.

**Tabel 5.10 Pengujian API Menambah Lokasi Penjualan**

<b>ID</b>	UJAPI02
<b>Nama</b>	Pengujian API Menambah Lokasi Penjualan
<b>Endpoint {http://10.151.31.19 2/api}</b>	/lokasi-penjualan/store
<b>Method</b>	POST
<b>Kondisi Awal</b>	Data lokasi penjualan belum tersimpan di <i>database</i>
<b>Skenario 1</b>	Isian Tidak Lengkap
<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>response</i> code 400 dan pesan bahwa isian tidak boleh kosong
<b>Hasil yang diperoleh</b>	Sistem menampilkan <i>response</i> code 400 dan pesan bahwa isian tidak boleh kosong
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 2</b>	Nama lokasi penjualan sama dengan nama lokasi penjualan yang sudah ada pada latitude dan longitude yang sama dan untuk jenis sampel yang sama

<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>response</i> code 400 dan pesan bahwa nama tidak boleh sama pada latitude dan longitude yang sama dan untuk jenis sampel yang sama
<b>Hasil yang diperoleh</b>	Sistem menampilkan <i>response</i> code 400 dan pesan bahwa nama tidak boleh sama pada latitude dan longitude yang sama dan untuk jenis sampel yang sama
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 3</b>	Isian lengkap dan tidak ada nama lokasi yang sama pada latitude dan longitude yang sama dan untuk jenis sampel yang sama
<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>response</i> code 200 dan pesan bahwa data berhasil disimpan
<b>Hasil yang diperoleh</b>	Sistem menampilkan <i>response</i> code 200 dan pesan bahwa data berhasil disimpan
<b>Hasil Pengujian</b>	Berhasil

```

1  {
2      "code": 400,
3      "message": "Isian tidak boleh kosong",
4      "data": null
5  }

```

**Gambar 5.19 Hasil Pengujian API Menambah Lokasi Penjualan Skenario 1**

```

1  {
2      "code": 400,
3      "message": "Sudah ada nama yang sama di latitude dan longitude yang sama",
4      "data": null
5  }

```

**Gambar 5.20 Hasil Pengujian API Menambah Lokasi Penjualan Skenario 2**

```

1  {
2    "code": 200,
3    "message": "Data berhasil disimpan",
4    "data": {
5      "nama_lokasi_penjualan": "Bakso Ku",
6      "latitude_lokasi_penjualan": "-7.27944438",
7      "longitude_lokasi_penjualan": "112.75350189209",
8      "id_jenis_sample": "1",
9      "foto_lokasi_penjualan": "20200105154448_1.jpg",
10     "updated_at": "2020-01-05 15:44:48",
11     "created_at": "2020-01-05 15:44:48",
12     "id_lokasi_penjualan": 51
13   }
14 }

```

**Gambar 5.21 Hasil Pengujian API Menambah Lokasi Penjualan Skenario 3**

### 5.3.3. Pengujian API Menambah Laporan Mengenai Lokasi Penjualan

Detil pengujian menambah laporan lokasi penjualan dijelaskan pada Tabel 5.11. Hasil dari pengujian terdapat pada Tabel 5.11.

**Tabel 5.11 Pengujian API Menambah Laporan Mengenai Lokasi Penjualan**

<b>ID</b>	UJAPI03
<b>Nama</b>	Pengujian API Menambah Laporan Mengenai Lokasi Penjualan
<b>Endpoint</b> { <a href="http://10.151.31.192/api">http://10.151.31.192/api</a> }	/laporan/store
<b>Method</b>	POST
<b>Kondisi Awal</b>	Data laporan belum tersimpan di <i>database</i>
<b>Skenario 1</b>	Isian Tidak Lengkap
<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>response</i> code 400 dan pesan bahwa isian tidak boleh kosong
<b>Hasil yang diperoleh</b>	Sistem menampilkan <i>response</i> code 400 dan pesan bahwa isian tidak boleh kosong
<b>Hasil Pengujian</b>	Berhasil
<b>Skenario 2</b>	Isian lengkap

<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>response</i> code 200 dan pesan bahwa data berhasil disimpan
<b>Hasil yang diperoleh</b>	Sistem menampilkan <i>response</i> code 200 dan pesan bahwa data berhasil disimpan
<b>Hasil Pengujian</b>	Berhasil

```

1  {
2      "code": 400,
3      "message": "Isian tidak boleh kosong",
4      "data": null
5  }

```

**Gambar 5.22 Hasil Pengujian API Menambah Laporan Mengenai Lokasi Penjualan Skenario 1**

```

1  {
2      "code": 200,
3      "message": "Data berhasil disimpan",
4      "data": {
5          "nama_pelapor": "Dewi",
6          "latitude_lokasi_penjualan": "-7.2841",
7          "longitude_lokasi_penjualan": "112.7535",
8          "isi_laporan": "Tutup",
9          "created_at": "2020-01-05 16:11:51",
10         "updated_at": "2020-01-05 16:11:51",
11         "id_laporan": 25
12     }

```

**Gambar 5.23 Hasil Pengujian API Menambah Laporan Mengenai Lokasi Penjualan Skenario 2**

#### 5.4. Pengujian Klasifikasi Sampel

Pada pengujian klasifikasi sampel ini, akan dilakukan pada satu sampel yang sudah diketahui tidak mengandung boraks, satu sampel yang sudah diketahui mengandung boraks, dan 10 sampel dari 10 lokasi berbeda yang belum diketahui ada atau tidaknya kandungan boraks di dalamnya. Keluaran dari pengujian klasifikasi sampel adalah status kandungan boraks pada bakso dan

nilai probabilitas sampel positif mengandung boraks, dimana semakin besar nilai probabilitas sebuah sampel maka sampel akan semakin positif mengandung boraks. Hasil pengujian sampel bakso dapat dilihat pada Tabel 5.12 dan Tabel 5.13.

**Tabel 5.12 Hasil Klasifikasi Data Sampel yang Sudah Diketahui Kandungan Boraksnya**

No.	Nama Sampel/Lokasi Pengambilan Sampel	Hasil Klasifikasi	Nilai probabilitas
1	Sampel bebas dari kandungan boraks	Tidak terindikasi ada boraks	0.48%
2	Sampel dengan kandungan boraks	Terindikasi ada boraks	99.37%

**Tabel 5.13 Hasil Klasifikasi Data Sampel yang Belum Diketahui Kandungan Boraksnya**

No.	Nama Sampel/Lokasi Pengambilan Sampel	Hasil Klasifikasi	Nilai probabilitas
1	Pasar Keputih - 1	Terindikasi ada boraks	73.82%
2	Pasar Keputih - 2	Terindikasi ada boraks	99.63%
3	Superindo Arief Rahman Hakim – Bakso Urat Tanpa Merk	Tidak terindikasi ada boraks	11.24%
4	Giant Arief Rahman - Raja Baoo	Tidak terindikasi ada boraks	0%
5	Pasar Pucang - Bakso Special	Terindikasi ada boraks	67.28%
6	Pasar Pucang – Bakso Kotak	Tidak terindikasi ada boraks	0.15%
7	Pasar Pucang - 1	Terindikasi ada boraks	86.16%
8	Pasar Pacarkeling - 1	Tidak terindikasi ada boraks	0.03%
9	Pasar Tempurejo - 1	Terindikasi ada boraks	96%

10	Pasar Tempurejo - 2	Terindikasi ada boraks	87.34%
----	---------------------	------------------------	--------

Tabel 5.12 menunjukkan bahwa aplikasi ini dapat mengklasifikasikan sampel yang sudah diketahui kandungan boraksnya sesuai status semestinya. Selain itu, Tabel 5.12 juga menunjukkan bahwa dari 10 sampel dari 10 lokasi penjualan berbeda yang telah diuji kandungan boraksnya dengan *electronic nose*, 4 sampel dari 4 lokasi penjualan tidak terindikasi mengandung boraks dan 6 sampel dari 6 lokasi penjualan terindikasi mengandung boraks.

### 5.5. Pengujian Usabilitas

Telah dilakukan pengujian terhadap calon pengguna pada aplikasi ini. Pengujian pada pengguna dilakukan pada aspek usabilitas untuk mengetahui seberapa mudahkan aplikasi ini digunakan sehingga calon pengguna yang akan menggunakan aplikasi ini tidak mengalami kesusahan saat penggunaan. Pengujian dilakukan untuk aktor administrator dengan partisipan 5 orang mahasiswa Departemen Teknik Informatika ITS yang memiliki akses ke mesin *electronic nose* Enosika. Setelah pengujian melakukan uji coba, akan diminta mengisi kuisisioner untuk menilai aplikasi berdasarkan kemudahan penggunaan. Kuisisioner terdiri dari 4 pertanyaan dengan jawaban berupa nilai dari rentang 1 – 4. Angka 1 menyatakan bahwa aplikasi memiliki nilai yang paling buruk pada poin yang ditanyakan pada pertanyaan tersebut sedangkan nilai 4 menyatakan bahwa aplikasi memiliki nilai yang sangat baik pada poin yang ditanyakan pada pertanyaan tersebut. Hasil survei dapat dilihat pada Tabel 5.14. Dari kuisisioner tersebut, didapatkan bahwa responden mengatakan bahwa aplikasi ini mudah digunakan dengan nilai persentase 95%.

**Tabel 5.14 Hasil Kuisisioner Usabilitas**

No	Pertanyaan	Nilai				Total	Persen tase
		1	2	3	4		

1	Apakah informasi yang ada pada aplikasi ini mudah dimengerti?	0	0	1	4	19	95%
2	Apakah penggunaan menu atau fitur aplikasi menu mudah dipahami?	0	0	1	4	19	95%
3	Apakah aplikasi dapat dengan mudah dipelajari?	0	0	2	3	18	90%
4	Secara keseluruhan apakah penggunaan aplikasi ini memuaskan?	0	0	0	5	20	100%
<b>Rata-rata</b>							95%

### 5.6. Rekapitulasi Pengujian Kasus Penggunaan dan API

Di bawah ini merupakan rekapitulasi dari kasus pengujian yang telah dilakukan pada skenario pengujian di atas. Tabel 5.15 merupakan hasil kasus pengujian kasus penggunaan pada aplikasi berbasis web dari Tugas Akhir ini. Sedangkan, Tabel 5.16 merupakan hasil kasus pengujian pada implementasi API endpoint yang dibuat pada Tugas Akhir ini.

**Tabel 5.15 Rekapitulasi Pengujian Kasus Penggunaan Aplikasi Berbasis Web**

Kasus Uji	Nama Kasus Pengujian	Skenario	Terpenuhi
UJUC01	Melihat lokasi penjualan	-	✓
UJUC02	Mengelola lokasi penjualan	Skenario 1	✓
		Skenario 2	✓
		Skenario 3	✓
		Skenario 4	✓
		Skenario 5	✓
		Skenario 6	✓

<b>Kasus Uji</b>	<b>Nama Kasus Pengujian</b>	<b>Skenario</b>	<b>Terpenuhi</b>
UJUC03	Menambah data sampel	Skenario 1	✓
		Skenario 2	✓
		Skenario 3	✓
UJUC04	Mengelola data sampel	Skenario 1	✓
		Skenario 2	✓
		Skenario 3	✓
		Skenario 4	✓
		Skenario 5	✓
UJUC05	Melihat laporan mengenai lokasi penjualan	-	✓
UJUC06	Melihat notifikasi	-	✓

**Tabel 5.16 Rekapitulasi Pengujian API Endpoint**

<b>Kasus Uji</b>	<b>Nama Kasus Pengujian</b>	<b>Skenario</b>	<b>Terpenuhi</b>
UJAPI01	Mengambil lokasi penjualan	-	✓
UJAPI02	Menambah lokasi penjualan	Skenario 1	✓
		Skenario 2	✓
		Skenario 3	✓
UJAPI03	Menambah laporan mengenai lokasi penjualan	Skenario 1	✓
		Skenario 2	✓



## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan Tugas Akhir ini berdasarkan hasil pengujian dan hal lainnya yang telah dilakukan. Selain itu, juga terdapat beberapa saran terhadap Tugas Akhir yang penulis ajukan terhadap pengembangan Tugas Akhir ini untuk pengembangan kedepannya.

#### **6.1. Kesimpulan**

Berdasarkan pengerjaan Tugas Akhir ini dan hasil yang didapatkan pada tahap uji coba aplikasi, penulis mengambil beberapa kesimpulan sebagai berikut.

1. Pengumpulan data sampel bakso dilakukan dengan menggunakan *electronic nose* Enosika dan klasifikasi kandungan boraks pada bakso dilakukan dengan metode *Multilayer Perceptron* dengan akurasi 91% dan tingkat kepercayaan 90%.
2. Pembuatan aplikasi sebagai detektor kandungan boraks pada bakso dan lokasi penjualannya berhasil diimplementasikan dengan bahasa pemrograman PHP serta kerangka kerja Laravel, dan bahasa pemrograman Python. Hal ini dibuktikan dengan rekapitulasi uji coba pada Tabel 5.15 dan Tabel 5.16.
3. Aplikasi ini berhasil menyajikan data lokasi penjualan bakso dengan kandungan boraks dalam bentuk peta dan *marker* dengan warna yang berbeda untuk menunjukkan indikasi kandungan boraks pada bakso.

#### **6.2. Saran**

Terdapat beberapa saran terkait Tugas Akhir ini yang diharapkan bisa membuat Tugas Akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

1. Menambahkan fitur untuk menyimpan foto sampel dengan jumlah lebih dari satu.

2. Menambahkan fitur untuk mengakomodasi satu lokasi yang memiliki banyak jenis bakso.
3. Menambahkan fitur riwayat pengujian data sampel.

## DAFTAR PUSTAKA

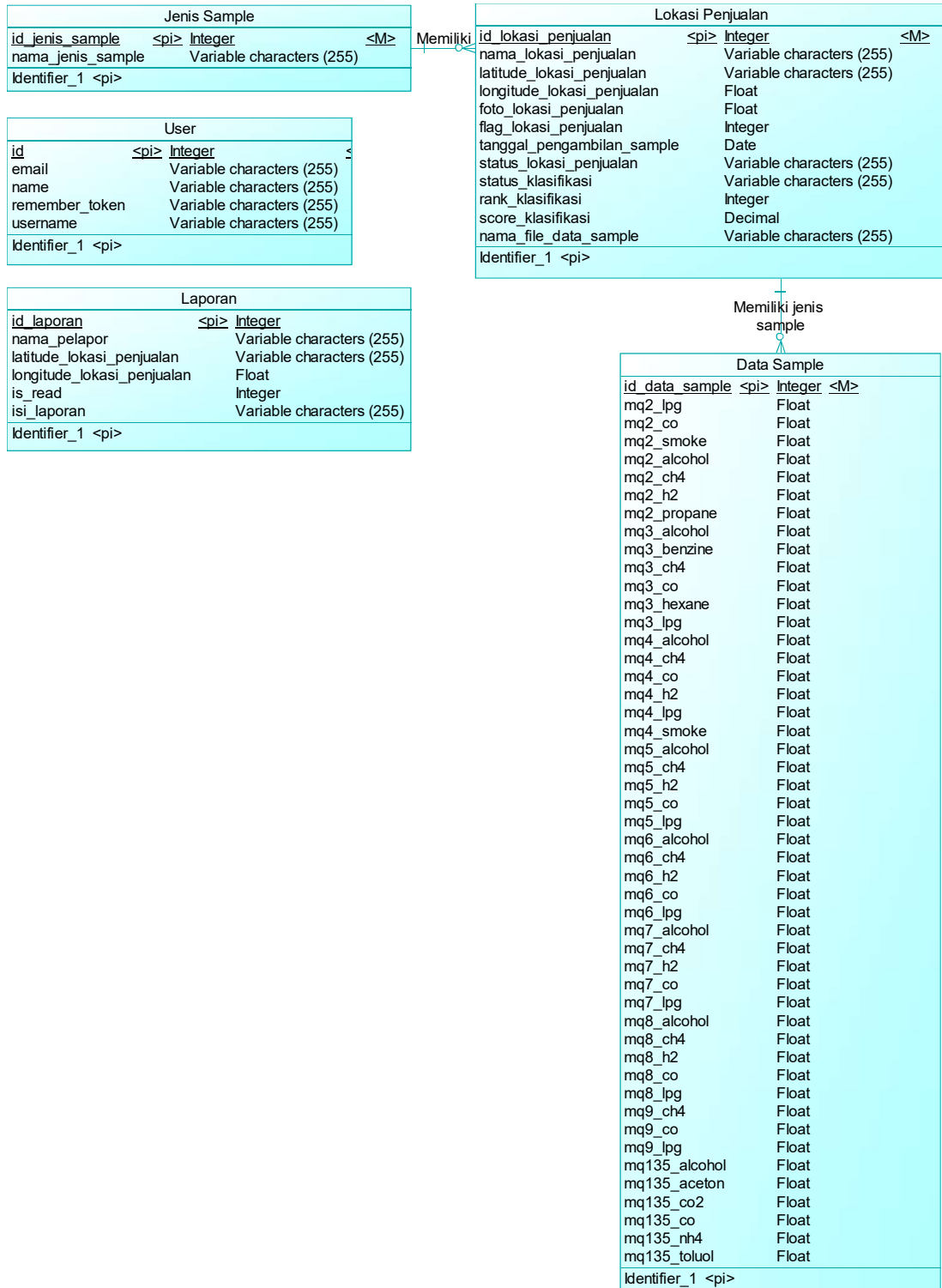
- [1] “Badan Pengawas Obat dan Makanan - Republik Indonesia.” [Online]. Available: <https://www.pom.go.id/new/view/more/berita/139/BAHAN-BERBAHAYA-YANG->. [Accessed: 13-Dec-2019].
- [2] D. Hudani, Diana; Rochimah, Siti; Sunaryono, “Rancang Bangun Aplikasi Berbasis Android Pengindikasi Formalin Pada Tahu dan Lokasi Penjualannya di Surabaya Timur Menggunakan Identifikasi Geografis,” 2020.
- [3] M. Nurjuliana, Y. B. Che Man, and D. Mat Hashim, “Analysis of lard’s aroma by an electronic nose for rapid Halal authentication,” *JAOCs, J. Am. Oil Chem. Soc.*, vol. 88, no. 1, pp. 75–82, Jan. 2011.
- [4] D. R. Wijaya, R. Sarno, and A. F. Daiva, “Electronic nose for classifying beef and pork using Naïve Bayes,” in *Proceedings - 2017 International Seminar on Sensor, Instrumentation, Measurement and Metrology: Innovation for the Advancement and Competitiveness of the Nation, ISSIMM 2017*, 2017, vol. 2017-January, pp. 104–108.
- [5] P. E. Keller, L. J. Kangas, L. H. Liden, S. Hashem, and R. T. Kouzes, “Electronic noses and their applications,” in *Northcon - Conference Record*, 1995, pp. 116–119.
- [6] “What is Electronic Nose (enose): Working Principle and Applications.” [Online]. Available: <https://www.elprocus.com/electronic-nose-work/>. [Accessed: 17-Dec-2019].
- [7] “MENGENAL BORAKS DAN DAMPAK PENGGUNAANNYA.” [Online]. Available: <http://sib3pop.menlhk.go.id/index.php/articles/view?slug=mengenal-boraks-dan-dampak-penggunaannya>. [Accessed: 26-Dec-2019].
- [8] “Classification.” [Online]. Available: [https://docs.oracle.com/cd/B28359\\_01/datamine.111/b281](https://docs.oracle.com/cd/B28359_01/datamine.111/b281)

- 29/classify.htm#DMCON004. [Accessed: 17-Dec-2019].
- [9] “Basic Concept of Classification (Data Mining) - GeeksforGeeks.” [Online]. Available: <https://www.geeksforgeeks.org/basic-concept-classification-data-mining/>. [Accessed: 17-Dec-2019].
- [10] S. Sumarlin, “Implementasi Algoritma K-Nearest Neighbor Sebagai Pendukung Keputusan Klasifikasi Penerima Beasiswa PPA dan BBM,” *J. Sist. Inf. BISNIS*, vol. 5, no. 1, Apr. 2015.
- [11] “Data Mining Algorithms In R/Classification/kNN - Wikibooks, open books for an open world.” [Online]. Available: [https://en.wikibooks.org/wiki/Data\\_Mining\\_Algorithms\\_In\\_R/Classification/kNN](https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Classification/kNN). [Accessed: 12-Jan-2020].
- [12] “Support Vector Machine — Introduction to Machine Learning Algorithms.” [Online]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>. [Accessed: 06-Jan-2020].
- [13] “ML - Support Vector Machine(SVM) - Tutorialspoint.” [Online]. Available: [https://www.tutorialspoint.com/machine\\_learning\\_with\\_python/machine\\_learning\\_with\\_python\\_classification\\_algorithms\\_support\\_vector\\_machine.htm](https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_support_vector_machine.htm). [Accessed: 12-Jan-2020].
- [14] “Support Vector Machine (SVM) Algorithm - Javatpoint.” [Online]. Available: <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>. [Accessed: 12-Jan-2020].
- [15] “1.4. Support Vector Machines — scikit-learn 0.22.1 documentation.” [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>. [Accessed: 12-Jan-2020].
- [16] “What is a Multilayer Perceptron (MLP)? - Definition from Techopedia.” [Online]. Available:

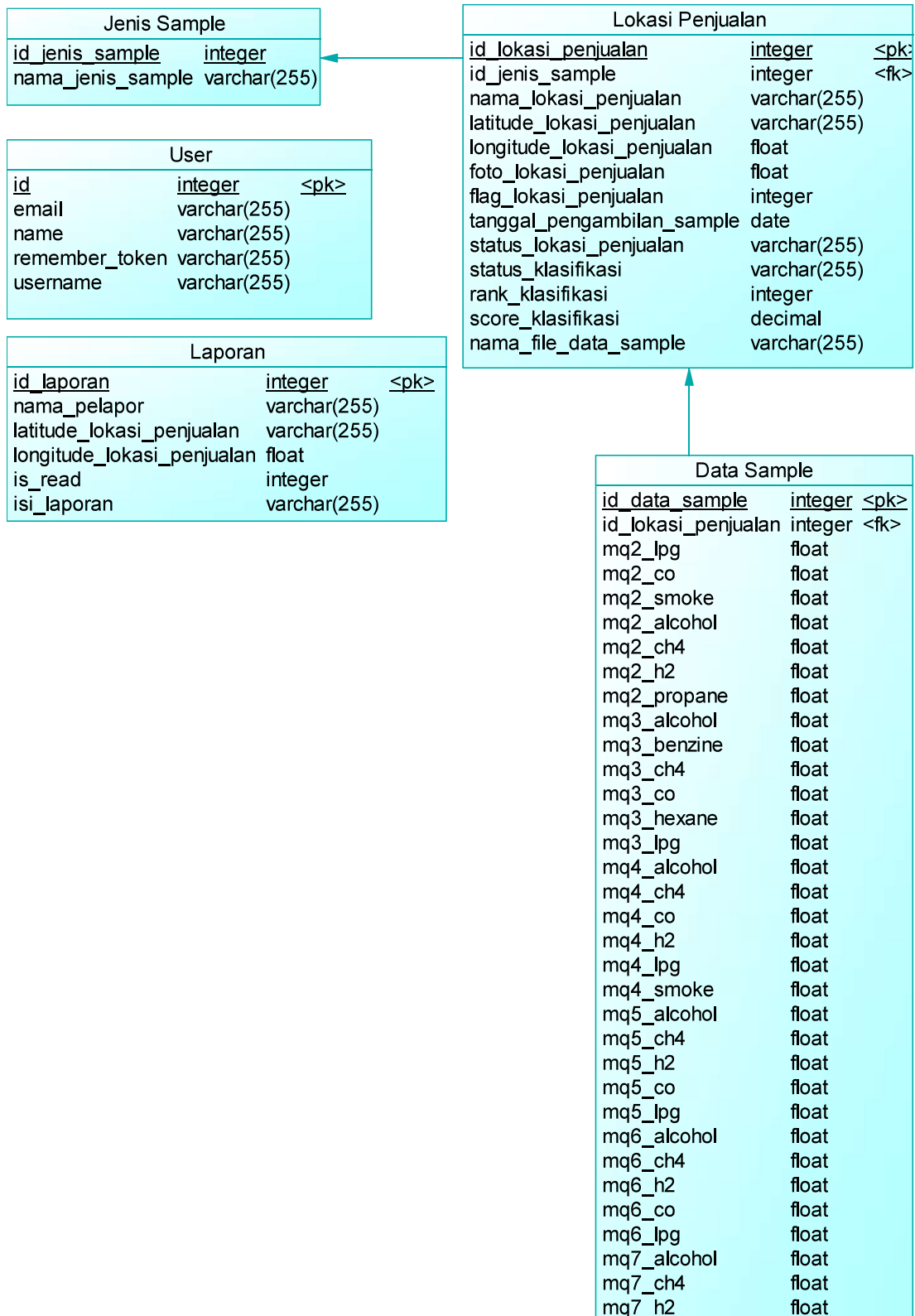
- <https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp>. [Accessed: 06-Jan-2020].
- [17] “Multilayer Perceptron Definition | DeepAI.” [Online]. Available: <https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron>. [Accessed: 12-Jan-2020].
- [18] “1.17. Neural network models (supervised) — scikit-learn 0.22.1 documentation.” [Online]. Available: [https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html](https://scikit-learn.org/stable/modules/neural_networks_supervised.html). [Accessed: 12-Jan-2020].
- [19] L. Welling and L. Thomson, *PHP and MYSQL Web Development*. 2001.
- [20] “Pengertian dan Fungsi PHP dalam Pemrograman Web | Duniaikom.” [Online]. Available: <https://www.duniaikom.com/pengertian-dan-fungsi-php-dalam-pemograman-web/>. [Accessed: 12-Jan-2020].
- [21] S. McCool, *Laravel starter : the definitive introduction to the Laravel PHP web development framework*. Packt Pub, 2012.
- [22] “How Laravel implements MVC and how to use it effectively.” [Online]. Available: <https://blog.pusher.com/laravel-mvc-use/>. [Accessed: 12-Jan-2020].
- [23] M. M. Sharma, R. Aggarwal, IT Pro - York University., and Skillsoft Books - York University., *Starting with Android*.
- [24] G. Corp, “GPS Beginner’s Guide 2 GPS Beginner’s Guide,” 2008.
- [25] R. Gibson and S. Erle, *Google maps hacks*. O’Reilly, 2006.
- [26] A. N. Sari, D. S. S. Kom, and M. Kom, “Perancangan dan Pembangunan Perangkat Fitur Geotagging,” *J. Tek. Pomits*, vol. 1, no. 1, pp. 1–6, 2012.
- [27] R. Fojtik, “The Use of Mobile Devices in Education,” vol. 3, pp. 41–47, 2017.
- [28] S. I. Sabilla, R. Sarno, and J. Siswantoro, “ScienceDirect

- Estimating Gas Concentration using Artificial Neural Network for Electronic Nose,” *Procedia Comput. Sci.*, vol. 124, p. 0, 2017.
- [29] D. Rahman Wijaya, R. Sarno, E. Zulaika, and S. I. Sabila, “ScienceDirect ScienceDirect Peer-review under responsibility of the scientific Development of mobile electronic nose for beef quality monitoring,” *Procedia Comput. Sci.*, vol. 124, 2017.

# LAMPIRAN



Lampiran 1 Gambar CDM Sistem



Lampiran 2 Gambar PDM Sistem



```
1. <script>
2.     var lotLang;
3.     var lokasiPenjualan = <?php echo $lokasiPenjualan;
4.     var markers = [];
5.
6.     function initMap() {
7.         var map = new google.maps.Map(document.getEleme
8. ntById('map'), {
9.         zoom: 13,
10.        center: new google.maps.LatLng(-
11. 7.2824, 112.7949),
12.        });
13.
14.        for (var i = 0; i < lokasiPenjualan.length; i++
15. ) {
16.         if(lokasiPenjualan[i]['status_klasifikasi']
17. =='Terindikasi ada boraks'){
18.         var marker = new google.maps.Marker({
19.         position: new google.maps.LatLng(lo
20. kasiPenjualan[i]['latitude_lokasi_penjualan'], lokasiPe
21. njualan[i]['longitude_lokasi_penjualan']),
22.         map: map,
23.         title: lokasiPenjualan[i]['nama_lok
24. asi_penjualan'],
25.         url: '{{url("lokasi-
26. penjualan/bakso")}}',
27.         icon: 'http://maps.google.com/mapfi
28. les/ms/icons/red-dot.png',
29.         });
30.         }
31.         else if(lokasiPenjualan[i]['status_klasifik
32. asi'] =='Tidak terindikasi ada boraks'){
33.         var marker = new google.maps.Marker({
34.         position: new google.maps.LatLng(lo
35. kasiPenjualan[i]['latitude_lokasi_penjualan'], lokasiPe
36. njualan[i]['longitude_lokasi_penjualan']),
37.         map: map,
38.         title: lokasiPenjualan[i]['nama_lok
39. asi_penjualan'],
40.         url: '{{url("lokasi-
41. penjualan/bakso")}}',
```

```

28.             icon: 'http://maps.google.com/mapfi
les/ms/icons/yellow-dot.png',
29.             });
30.
31.         }
32.         else if(lokalasiPenjualan[i]['status_klasifik
asi'] == 'Terindikasi ada formalin'){
33.             var marker = new google.maps.Marker({
34.                 position: new google.maps.LatLng(lo
kasiPenjualan[i]['latitude_lokasi_penjualan'], lokasiPe
njualan[i]['longitude_lokasi_penjualan']),
35.                 map: map,
36.                 title: lokasiPenjualan[i]['nama_lok
asi_penjualan'],
37.                 url: '{{url("lokasi-
penjualan/bakso")}}',
38.                 icon: 'http://maps.google.com/mapfi
les/ms/icons/orange-dot.png',
39.                 });
40.
41.             }
42.             else if(lokalasiPenjualan[i]['status_klasifik
asi'] == 'Tidak terindikasi ada formalin'){
43.                 var marker = new google.maps.Marker({
44.                     position: new google.maps.LatLng(lo
kasiPenjualan[i]['latitude_lokasi_penjualan'], lokasiPe
njualan[i]['longitude_lokasi_penjualan']),
45.                     map: map,
46.                     title: lokasiPenjualan[i]['nama_lok
asi_penjualan'],
47.                     url: '{{url("lokasi-
penjualan/bakso")}}',
48.                     icon: 'http://maps.google.com/mapfi
les/ms/icons/blue-dot.png',
49.                     });
50.
51.                 }
52.             else{
53.                 var marker = new google.maps.Marker({
54.                     position: new google.maps.LatLng(lo
kasiPenjualan[i]['latitude_lokasi_penjualan'], lokasiPe
njualan[i]['longitude_lokasi_penjualan']),

```

```

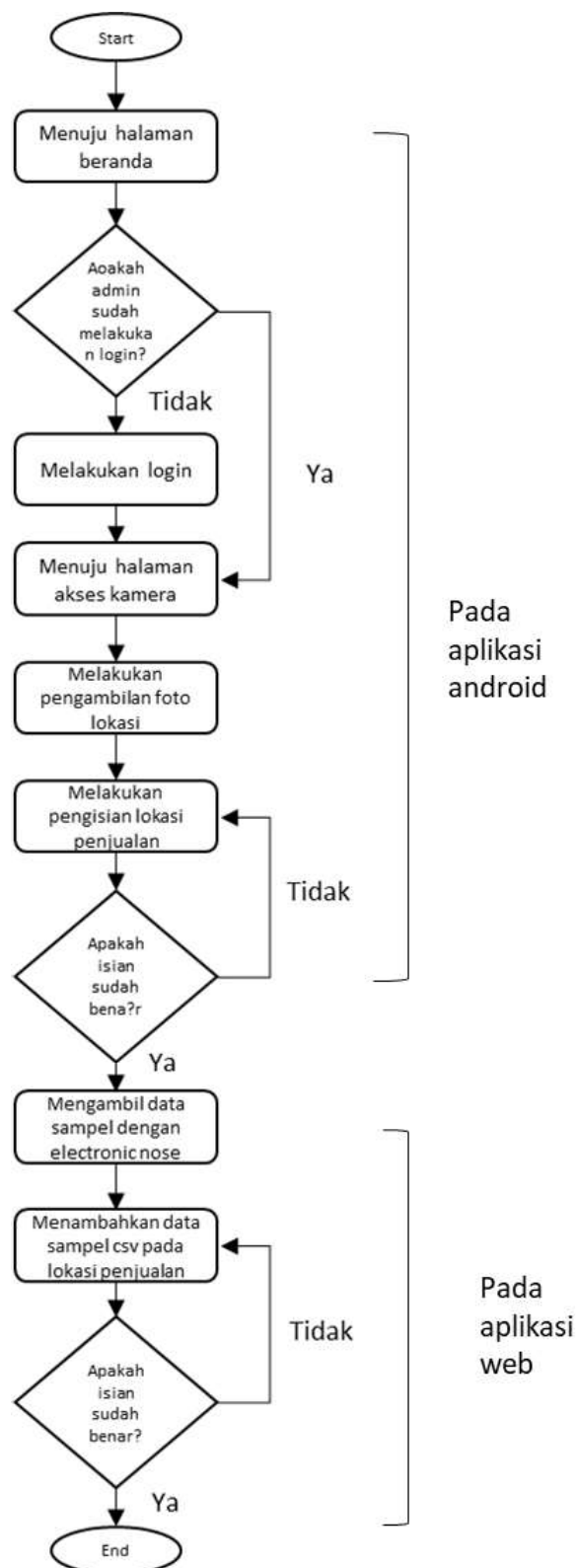
55.         map: map,
56.         title: lokasiPenjualan[i]['nama_lokasi_penjualan'],
57.         url: '{{url("lokasi-
penjualan/bakso")}}',
58.         icon: 'http://maps.google.com/mapfiles/ms/icons/green-dot.png',
59.     });
60.     }
61.
62.
63.     google.maps.event.addListener(marker, 'click', (function(marker, i) {
64.         return function() {
65.             $('#namaLokasiPenjualan').html(lokasiPenjualan[i]['nama_lokasi_penjualan']);
66.             $('#statusKlasifikasi').html('Status klasifikasi: '+lokasiPenjualan[i]['status_klasifikasi']);
67.             $('#rankKlasifikasi').html('Rank klasifikasi: '+lokasiPenjualan[i]['rank_klasifikasi']);
68.             $('#scoreKlasifikasi').html('Score klasifikasi: '+lokasiPenjualan[i]['score_klasifikasi']);
69.             $('#fotoLokasiPenjualan').attr('src', "{{url('/img/lokasi-penjualan/')}}/"+lokasiPenjualan[i]['foto_lokasi_penjualan']);
70.             $('#linkMenujuPeta').attr('href', 'https://www.google.com/maps/place/'+lokasiPenjualan[i]['latitude_lokasi_penjualan']+','+lokasiPenjualan[i]['longitude_lokasi_penjualan']);
71.             $('#modalInformasi').modal('toggle');
72.         }
73.     })(marker, i));
74.     }
75. }
76. </script>

```

### Lampiran 3 Kode Sumber Menampilkan Lokasi Penjualan Pada Marker Google Maps

No. Baris	Kegunaan
3	Deklarasi variabel yang mengambil data lokasi penjualan
4	Deklarasi variabel yang menyimpan marker lokasi
6-10	Menginisiasi lokasi tempat peta di <i>zoom</i> pertama kali saat halaman ditampilkan
12	Melakukan <i>looping</i> data untuk setiap lokasi penjualan
13-21	Membuat peta berwarna merah apabila lokasi terindikasi mengandung boraks
22-30	Membuat peta berwarna kuning apabila lokasi tidak terindikasi mengandung boraks
32-38	Membuat peta berwarna oranye apabila lokasi terindikasi mengandung formalin
42-49	Membuat peta berwarna oranye apabila lokasi tidak terindikasi mengandung formalin
53-60	Membuat peta berwarna hijau apabila lokasi tidak valid
63-73	Fungsi untuk memunculkan informasi mengenai lokasi penjualan apabila marker dipilih oleh pengguna

**Lampiran 4 Penjelasan Kode Sumber Menampilkan Lokasi Penjualan Pada Marker Google Maps**



**Lampiran 5 Flowchart Cara Penambahan Lokasi Penjualan**

*[Halaman Sengaja Dikосongkan]*

## BIODATA PENULIS



Penulis lahir di Yogyakarta, 6 April 1999. Penulis telah menempuh pendidikan dasar di SDN Margorejo I/403 Surabaya, kemudian untuk pendidikan menengah pertama di SMPN 1 Surabaya dan di jenjang menengah atas di SMAN 5 Surabaya. Lalu penulis melanjutkan pendidikan sarjana S1 di Jurusan Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Selama kuliah, penulis aktif berorganisasi menjadi staf Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS 2017/2018 sebagai Staff Departemen Kaderisasi Pemetaan, HMTC ITS 2018/2019 sebagai Sekretaris, UKM Sepakbola ITS 2018/2019 sebagai Kepala Divisi Kominfo dan aktif pada kegiatan-kegiatan lain seperti Schematics 2017, Gerigi 2018, dan IFC 2018. Selain itu, penulis juga aktif di komunitas penerima beasiswa Bank Indonesia yaitu Genbi Komisariat ITS 2019.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Rekayasa Perangkat Lunak serta memiliki ketertarikan di bidang Sistem dan Manajemen Basis Data, Pemrograman *Web*, serta pemrograman perangkat bergerak berbasis Android. Untuk komunikasi, penulis dapat dihubungi melalui surel: [sekar.6499@gmail.com](mailto:sekar.6499@gmail.com).