ANALISIS *MICRO-SPATIAL* POPULASI KOMODITAS BERBASIS *INDOOR WEBGIS* (STUDI KASUS: PASAR BESAR KOTA MALANG)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Frida Ayundha Putry
NIM: 125150402111003

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

ANALISIS *MICRO-SPATIAL* POPULASI KOMODITAS BERBASIS *INDOOR WEBGIS*
(STUDI KASUS: PASAR BESAR KOTA MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Frida Ayundha Putry
NIM: 125150402111003

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Juni 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Aditya Rachmadi, S.ST., M.TI	Fatwa Ramdani, D.Sc., S.Si., M.Sc
NIK: 201201 860421 1 001	NIK: -

Mengetahui
Ketua Program Studi Nama Program Studi

Suprapto, S.T.,M.T.
NIP: 19710727 199603 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 24 Juni 2016

Frida Ayundha Putry

NIM: 125150402111003

KATA PENGANTAR

Puji syukur dipanjatkan atas kehadirat Allah SWT, karena hanya dengan rahmat, hidayah dan bimbingannya penulis dapat menyelesaikan skripsi dengan judul "Analisis *Micro-spatial* Populasi Komoditas Berbasis *Indoor webGIS* (Studi Kasus: Pasar Besar Kota Malang)" dengan baik. Selama pembuatan skripsi ini, penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada :

1. Keluarga, Alm. Bapak Mochamad Yunus dan Ibu Nafik Tafrikah serta Bapak Sungging Suharyono dan Ibu Nisrul Asmah. Saudara saya Dinda Yuana Alfisa, M. Taufan Rezzafri, dan M. Bayu Argiansyah yang telah memberikan do'a, motivasi, kasih sayang, serta dukungan baik secara moril dan materiil dalam penyelesaian skripsi ini.
2. Bapak Aditya Rachmadi, S.ST., M.TI. selaku dosen pembimbing I yang telah banyak memberikan ilmu, saran, motivasi dan do'a untuk penyelesaian skripsi ini.
3. Bapak Fatwa Ramdani, D.Sc., S.Si., M.Sc. selaku dosen pembimbing II yang telah banyak memberikan ilmu, saran, motivasi dan do'a untuk penyelesaian skripsi ini.
4. Bapak Suprapto, S.T., M.T. selaku Ketua Program Studi Sistem Informasi yang banyak memberikan ilmu, saran dan motivasi untuk penyelesaian skripsi ini.
5. Bapak Ismiarta Aknuranda, S.T., M.Sc., Ph.D. selaku Sekretaris Program Studi Sistem Informasi yang banyak memberikan ilmu, saran dan motivasi untuk penyelesaian skripsi ini.
6. Bapak Yusi Tyroni Mursityo, S.Kom., M.S. Selaku ketua Laboratorium Sistem Informasi.
7. Bapak dan Ibu dosen Fakultas Ilmu Komputer/Program Teknologi Informasi dan Ilmu Komputer yang telah memberikan ilmu baik di dalam maupun di luar perkuliahan.
8. Pihak Dinas Pasar Kota Malang dan Pasar Besar Kota Malang yang telah memperbolehkan dan membantu dalam penelitian sehingga penelitian dapat tercapai dan sukses.
9. Para sahabatku Isnaini Azizatin Lutfa, Brilliant Chindy Maharani, Mochamad Arizal, Anom Sulton Iskandar, Ika Qutsiati Utami, Putri Mutiara Tungga, Krisdamara WP, Lia Hasanah, Onny Iman KI, Rayyan Ramadhan, Dianisa Hakim, Randi Dwi Nandra, Atikah Hayyu, MG Hanifa, SBS, teman-teman Sistem Informasi 2012 dan 2011 yang telah banyak menemani, membantu, memberi masukan, dukungan, motivasi selama perkuliahan.

10. Para anggota Laboratorium Sistem Informasi Fakultas Ilmu Komputer/Program Teknologi Informasi dan Ilmu Komputer yang tidak bisa saya sebutkan satu per satu.
11. Para anggota *Geoinformatics Research Center* Fakultas Ilmu Komputer/Program Teknologi Informasi dan Ilmu Komputer yang tidak bisa saya sebutkan satu per satu.
12. Semua pihak yang tidak bisa penulis sebutkan satu per satu yang telah membantu penulis dalam penyelesaian skripsi ini.

Semoga segala kebaikan dan amal baik mendapatkan balasan dari Allah SWT. Penulis yakin masih banyak kekurangan pada skripsi ini, oleh karena itu saran dan kritik yang membangun diharapkan oleh penulis. Semoga skripsi ini dapat memberikan manfaat kepada setiap pembaca.

Malang, 24 Juni 2016

Penulis

fridayundha@gmail.com

ABSTRAK

Analisis *micro-spatial* adalah teknik yang disediakan oleh Sistem Informasi Geografis (SIG) modern pada skala peta yang lebih kecil atau resolusi spasial yang lebih halus. Dengan analisis *micro-spatial* pemetaan tempat-tempat dengan skala besar dan memiliki intensitas yang tinggi untuk dikunjungi dapat dilakukan, sehingga dalam melakukan aktivitas di dalamnya pengunjung tidak merasa kebingungan. Hal tersebut tercermin pada pasar tradisional, contohnya Pasar Besar Kota Malang. Terdapat beberapa metode yang dikembangkan untuk menghasilkan unit geografis pada skala mikro, umumnya dikenal sebagai *"Dasymetric Mapping"*. *Dasymetric mapping* merupakan solusi potensial untuk menggambarkan data populasi yang telah dikumpulkan menjadi unit area. Karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas dapat diketahui menggunakan *dasymetric algorithm*. *Dasymetric algorithm* membutuhkan data populasi komoditas, kios, posisi lantai, klasifikasi komoditas di Pasar Besar. Untuk mendukung analisis *error* terhadap *dasymetric algorithm* akan digunakan metode *Root Mean Square Error (RMSE)*. Dalam penelitian ini, dihasilkan *webGIS* menggunakan perangkat lunak Quantum GIS dan teknik analisis *micro-spatial* dengan tujuan menampilkan informasi berupa koordinat serta detail tiap-tiap kios yang terdapat pada Pasar Besar. Selain itu, sistem dapat memvisualisasikan dan melakukan pencarian terhadap komoditas tertentu berdasarkan radius.

Kata Kunci: Analisis *Micro-spatial*, *Dasymetric Mapping*, *Dasymetric Algorithm*, Pasar Besar, Komoditas, *RMSE*, Sistem Informasi Geografis.

repository.ub.ac.id

ABSTRACT

Micro-spatial analysis is a Geographic Information System (GIS) modern technique on a smaller scale or smoother spatial resolution. Mapping areas with a large scale and has a high intensity for a visit can be handled with micro-spatial analysis, so that visitors don't get confused when do activity. This is reflected in the traditional markets, such as Pasar Besar Malang City. There are several methods to produce a geographical unit on a micro scale, commonly known as "Dasymetric Mapping". Dasymetric mapping is a potential solution to describe data of population which collected into unit area. Characteristics of the commodities population density in every commodities classification can be determined using the dasymetric algorithm. Dasymetric algorithm requires population data of commodities, store, floor position, classification of commodities in the Pasar Besar Malang City. Dasymetric algorithm requires population data of commodities, stands, floor position, commodities classification in a Pasar Besar Malang City. Root Mean Square Error (RMSE) will be used to test the accuracy of the dasymetric algorithm. This research produced a webGIS that develop using Quantum GIS software and micro-spatial analysis techniques with the aim of displaying information such as the coordinates and details of each stand in Pasar Besar Malang City. In addition, the system can visualize and perform a search on a particular commodity based on radius.

Keywords: Micro-spatial Analysis, Dasymetric Mapping, Dasymetric Algorithm, Market Centre, Commodity, RMSE, Geographic Information Systems.

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xi
DAFTAR GAMBAR xiii
DAFTAR LAMPIRAN xv
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan Masalah 3
1.6 Sistematika Pembahasan 3
BAB 2 LANDASAN KEPUSTAKAAN 5
2.1 Kajian Pustaka 5
2.1.1 Metode *Dasymetric* 5
2.1.2 Pengujian *RMSE* 6
2.2 Dasar Teori 6
2.2.1 Sistem Informasi Geografis 6
2.2.2 Tahapan *SDLC* 7
2.2.3 *Unified Modelling Language* 7
2.2.4 Pengujian *Black-box* 15
2.2.5 Pengujian *Portability* 15
BAB 3 METODOLOGI 16
3.1 Studi Literatur 17
3.2 Pengumpulan Data 17
3.3 Analisis Kebutuhan Sistem 17

3.4 Perancangan Sistem 18
3.5 Implementasi Sistem 18
3.6 Pengujian dan Analisis 18
BAB 4 REKAYASA PERSYARATAN/KEBUTUHAN 19
4.1 Analisis 19
4.1.1 Gambaran Umum 20
4.1.2 Identifikasi Pengguna 21
4.1.3 Identifikasi Kebutuhan Fungsional 22
4.1.4 Identifikasi Kebutuhan Non-fungsional 23
4.1.5 *Use Case Diagram* 24
4.2 Perancangan 35
4.2.1 *Activity Diagram* 35
4.2.2 *Sequence Diagram* 37
4.2.3 *Class Diagram* 45
4.2.4 Perancangan Basis Data 47
4.2.5 Perancangan Antarmuka Pengguna 51
BAB 5 IMPLEMENTASI 55
5.1 Spesifikasi Lingkungan Implementasi 55
5.1.1 Spesifikasi Perangkat Keras 55
5.1.2 Spesifikasi perangkat lunak 56
5.2 Batasan Implementasi 56
5.3 Implementasi Kelas 56
5.4 Implementasi *Micro-spatial* 57
5.4.1 Digitalisasi Peta Pasar Besar 57
5.4.2 Implementasi Perangkat Lunak SIG 59
5.4.3 Implementasi *Dasymetric* 64
5.4.4 *Map Tools and Domain Analysis* 69
5.5 Implementasi Antarmuka Pengguna 70
5.5.1 Antarmuka Mengelola Peta 70
5.5.2 Antarmuka Mengelola *Dasymetric* 70
BAB 6 PENGUJIAN 74
6.1 Pengujian *Black-box* (Uji Validasi) 74

6.2 Pengujian *Portability* 78
6.3 Pengujian *RMSE* 78
BAB 7 PENUTUP 79
7.1 Kesimpulan 79
7.2 Saran 79
DAFTAR PUSTAKA 80
LAMPIRAN A DATA PEDAGANG PASAR BESAR ..... A
LAMPIRAN B HASIL PERHITUNGAN *DASYMETRIC ALGORITHM* ..... D
LAMPIRAN C HASIL PENGUJIAN *RMSE* ..... F
LAMPIRAN D HASIL PENGUJIAN *PORTABILITY* ..... H
LAMPIRAN E HASIL WAWANCARA ..... K

DAFTAR TABEL

Tabel 2.1 Tabel representasi *use case diagram* 8
Tabel 2.2 Representasi *activity diagram* 10
Tabel 2.3 Tabel representasi *sequence diagram* 11
Tabel 2.4 Tabel representasi *class diagram* 14
Tabel 4.1 Identifikasi pengguna 22
Tabel 4.2 Daftar kebutuhan fungsional 22
Tabel 4.3 Daftar kebutuhan non-fungsional 23
Tabel 4.4 Skenario *use case login* 25
Tabel 4.5 Skenario *use case* mengelola dasbor 25
Tabel 4.6 Skenario *use case* mengelola peta 26
Tabel 4.7 skenario *use case* mengelola *dasymetric* 27
Tabel 4.8 Skenario *use case* mengelola data pedagang 28
Tabel 4.9 Skenario *use case* mengelola data komoditas 33
Tabel 4.10 Skenario *use case* logout 34
Tabel 4.11 Tabel blok 47
Tabel 4.12 Tabel kategori komoditas 48
Tabel 4.13 Tabel kios 48
Tabel 4.14 Tabel kios point 48
Tabel 4.15 Tabel klasifikasi komoditas 49
Tabel 4.16 Tabel komoditas 49
Tabel 4.17 Tabel kota distributor 49
Tabel 4.18 Tabel lantai 50
Tabel 4.19 Tabel pedagang 50
Tabel 4.20 Tabel pedagang has komoditas 50
Tabel 4.21 Tabel pedagang has kota distributor 50
Tabel 4.22 Tabel pengguna 51
Tabel 4.23 Tabel ukuran 51
Tabel 5.1 Spesifikasi perangkat keras 55
Tabel 5.2 Spesifikasi perangkat lunak 56
Tabel 5.3 Implementasi kelas 56

Tabel 5.4 Nilai variabel perhitungan *dasymetric algorithm* 65
Tabel 5.5 Hasil perhitungan klasifikasi pakaian jadi lantai satu 65
Tabel 5.6 *Source code* implementasi grafik *dasymetric* 66
Tabel 5.7 *Source code* implementasi *dasymetric mapping* lantai dasar 67
Tabel 5.8 *Source code* implementasi *dasymetric mapping* lantai satu 68
Tabel 5.9 *Map tools and domain analysis* 69
Tabel 6.1 Hasil pengujian *black-box* 74
Tabel 6.2 Kalkulasi hasil pengujian *portability* 78
Tabel 6.3 Hasil perhitungan *dasymetric algorithm* 78

DAFTAR GAMBAR

Gambar 2.1 Diagram tahapan metode *waterfall* 7
Gambar 2.2 Contoh penggunaan *use case diagram* mengelola peta 8
Gambar 2.3 Contoh penggunaan *activity diagram login* 11
Gambar 2.4 Contoh penggunaan *sequence diagram login* 13
Gambar 2.5 Contoh penggunaan *class diagram login* 14
Gambar 3.1 Diagram alir metode penelitian 16
Gambar 4.1 Diagram blok rekayasa persyaratan/kebutuhan 19
Gambar 4.2 Logo ApsOrcim 20
Gambar 4.3 Proses bisnis Pasar Besar saat ini 20
Gambar 4.4 Proses bisnis Pasar Besar pada Sistem 21
Gambar 4.5 *Use case diagram* 24
Gambar 4.6 *Activity diagram* mengelola peta 36
Gambar 4.7 *Activity diagram* mengelola *dasymetric* 37
Gambar 4.8 *Sequence diagram* mengelola peta lantai dasar 38
Gambar 4.9 *Sequence diagram* mengelola peta lantai satu 40
Gambar 4.10 *Sequence diagram* grafik *dasymetric* 41
Gambar 4.11 *Sequence diagram dasymetric mapping* lantai dasar 42
Gambar 4.12 *Sequence diagram dasymetric mapping* lantai satu 44
Gambar 4.13 *Class diagram* 46
Gambar 4.14 Relasi antar basis data 47
Gambar 4.15 Perancangan antarmuka mengelola peta 52
Gambar 4.16 Perancangan antarmuka grafik *dasymetric* 53
Gambar 4.17 Perancangan antarmuka *dasymetric* mapping 54
Gambar 5.1 Diagram alir implementasi 55
Gambar 5.2 Diagram alir digitasi peta Pasar Besar 57
Gambar 5.3 Digitalisasi denah Pasar Besar 58
Gambar 5.4 Proses blok *polygon* pada perangkat lunak AutoCad 58
Gambar 5.5 Perintah ALIGN pada kolom *command* 58
Gambar 5.6 Proses menentukan poin pertama 58
Gambar 5.7 Proses menentukan tujuan *point* pertama 58

Gambar 5.8 Proses menentukan tujuan *point* kedua 59
Gambar 5.9 Proses menentukan *point* ketiga atau melanjutkan 59
Gambar 5.10 Proses menentukan skala pada objek 59
Gambar 5.11 Diagram alir implementasi perangkat lunak SIG 59
Gambar 5.12 Tampilan *file* *.dxf pada Quantum GIS 60
Gambar 5.13 Menambahkan *layer* dari *delimited text file* 60
Gambar 5.14 Tampilan *add vector join* 61
Gambar 5.15 Tampilan peta dengan klasifikasi warna berdasarkan blok 61
Gambar 5.16 Tampilan simpan *layer* vektor 62
Gambar 5.17 Tampilan *export to web map* 62
Gambar 5.18 Tampilan *add vector layer* 63
Gambar 5.19 Tampilan simpan *file* GeoJson dengan ekstensi *.shp 63
Gambar 5.20 Tampilan membuat koneksi dengan basis data 64
Gambar 5.21 Tampilan *import/export* manager pada PostGIS 64
Gambar 5.22 Hasil perhitungan *dasymetric algorithm* lantai dasar 66
Gambar 5.23 Hasil perhitungan *dasymetric* algorithm lantai satu 66
Gambar 5.24 Implementasi antarmuka mengelola peta 70
Gambar 5.25 Implementasi antarmuka grafik *dasymetric* 71
Gambar 5.26 Implementasi antarmuka *dasymetric* mapping 72
Gambar 5.27 Implementasi antarmuka *map tools* detail pencarian 72
Gambar 5.28 Implementasi antarmuka *map tools* grafik pencarian 73

DAFTAR LAMPIRAN

LAMPIRAN A DATA PEDAGANG PASAR BESAR ..... A
LAMPIRAN B HASIL PERHITUNGAN *DASYMETRIC ALGORITHM* ..... D
LAMPIRAN C HASIL PENGUJIAN *RMSE* ..... F
LAMPIRAN D HASIL PENGUJIAN *PORTABILITY* ..... H
D.1 Halaman *Login* ..... H
D.2 Halaman Dasbor ..... H
D.3 Halaman Peta Lantai Dasar ..... H
D.4 Halaman Peta Lantai Satu ..... H
D.5 Halaman Grafik *Dasymetric* I
D.6 Halaman *Dasymetric Mapping* Lantai Dasar I
D.7 Halaman *Dasymetric Mapping* Lantai Satu I
D.8 Halaman Tambah Data Pedagang I
D.9 Halaman Data Pedagang ..... J
D.10 Halaman Detail Data Pedagang ..... J
D.11 Halaman Tambah Data Komoditas ..... J
D.12 Halaman Data Komoditas ..... J
LAMPIRAN E HASIL WAWANCARA ..... K

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pasar sebagai bentuk pelayanan umum dan tempat terjadinya transaksi jual beli merupakan salah satu cerminan perekonomian dan sosial budaya setiap komunitas di dunia. Pasar memiliki peran paling penting dalam kehidupan sehari-hari, bagi masyarakat pasar bukan hanya tempat bertemunya antara penjual dan pembeli tetapi juga sebagai wadah untuk berinteraksi sosial (Satuhu, et al., 2010). Fenomena pertumbuhan pedagang, keterbatasan lahan, dan pengelolaan pasar yang buruk mengakibatkan kurang tertatanya kegiatan operasional. Kondisi keterbatasan lahan dapat dilihat dengan adanya eksploitasi ruang pasar untuk berdagang. Lorong dan koridor pasar dijadikan tempat jual-beli yang mengakibatkan pasar terlihat semrawut. Kasus ini dapat dilihat pada Pasar Besar Kota Malang yang selanjutnya akan ditulis dengan Pasar Besar.

Pasar Besar merupakan pusat sarana pelayanan dan jasa serta pusat kegiatan perekonomian Kota Malang dan daerah sekitarnya. Pasar Besar yang dibangun pada jaman penjajahan Hindia Belanda pada tahun 1914 saat ini memiliki 4362 kios. Informasi pedagang pada kios Pasar Besar masih menggunakan data yang sesuai pada buku induk yang lama, sedangkan setiap waktu data pada lapangan bisa saja mengalami perubahan. Dinas Pasar Kota Malang pun hanya memiliki data pedagang yang terdapat di Pasar Besar tanpa mengetahui komoditas dan kota distributor komoditas yang dijual. Hal ini mengakibatkan informasi yang diperoleh tidak sesuai dengan keadaan lapang, sehingga pendataan yang dilakukan sulit untuk diperbarui dan informasi yang dimiliki juga kurang lengkap. Dengan pendekatan teknologi diharapkan informasi terkait data pada Pasar Besar dapat divisualisasikan dengan lebih baik, sehingga pendataan informasi oleh Dinas Pasar Kota Malang maupun pihak manajemen Pasar Besar dapat dilakukan dengan lebih mudah. Kondisi tersebut dapat dipenuhi dengan adanya Sistem informasi Geografis (SIG) yang memungkinkan penggunanya untuk mengelola, menganalisis, dan memetakan informasi spasial dan data atributnya (Prahasta, 2009).

Sistem Informasi Geografis (SIG) merupakan sebuah sistem yang dapat mendukung proses pengambilan keputusan mengenai aspek spasial dan mampu menghubungkan suatu lokasi dengan karakteristik spasial yang terdapat pada lokasi tersebut. Data spasial, perangkat keras, perangkat lunak, dan struktur organisasi merupakan metodologi dan teknologi yang diperlukan untuk mengembangkan SIG (Prahasta, 2009).

Implementasi SIG sebagian besar dimanfaatkan untuk merepresentasikan keadaan suatu wilayah yang luas dan mencerminkan karakteristik tempat terbuka *(outdoor)*. Pada kenyataannya SIG berpotensi untuk menganalisis data spasial dalam tingkat kedetailan yang lebih tinggi, atau dapat disebut dengan skala mikro. Penelitian Ko Ko Lwin dan Yuji Murayama memvisualisasikan populasi penduduk yang berfokus pada populasi setiap bangunan di Kota Tsukuba, Jepang dengan

memanfaatkan teknik analisis *micro-spatial*. Analisis *micro-spatial* adalah teknik yang disediakan oleh SIG modern dengan memanfaatkan resolusi spasial yang lebih halus atau skala peta yang lebih kecil (Lwin & Murayama, 2010). Penelitian ini mengembangkan teknik analisis *micro-spatial* dengan melakukan pemetaan terhadap populasi komoditas pada tempat tertutup (*indoor*) dengan intensitas yang tinggi untuk dikunjungi seperti pasar tradisional, sehingga dalam melakukan aktivitas di dalamnya pengunjung tidak merasa kebingungan.

Pemetaan terhadap populasi komoditas pada Pasar Besar membutuhkan tingkat kedetailan yang lebih tinggi. Dari penjelasan tersebut dapat disimpulkan bahwa dengan menggunakan analisis *micro-spatial* pada Pasar Besar, maka pemetaan populasi komoditas Pasar Besar dapat dilakukan dengan lebih detail. Terdapat metode yang dikembangkan untuk menghasilkan unit geografis pada skala mikro, umumnya dikenal sebagai "*dasymetric mapping*" (Esri, 2016).

Dasymetric mapping merupakan solusi potensial untuk menggambarkan data populasi yang telah dikumpulkan menjadi unit area (Sleeter, 2001). Manajemen Pasar Besar telah mengelompokkan kios berdasarkan blok yang menggambarkan komoditas yang dijual oleh kios terkait. Populasi komoditas pada setiap blok merupakan komoditas yang dikelompokkan sesuai dengan klasifikasinya. Karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas dapat diketahui menggunakan *dasymetric algorithm*. Penelitian Ming-Dawa Su menjelaskan bahwa *dasymetric algorithm* dapat mengasumsikan setiap klasifikasi komoditas memiliki karakteristik kepadatan populasi. *Dasymetric algorithm* membutuhkan data populasi komoditas, kios, posisi lantai, klasifikasi komoditas di Pasar Besar. Untuk mendukung analisis *error* terhadap *dasymetric algorithm* akan digunakan metode *Root Mean Square Error (RMSE)* (Su, et al., 2010).

Hasil akhir dari penelitian berjudul "Analisis *Micro-spatial* Populasi Komoditas Berbasis *Indoor webGIS* (Studi Kasus: Pasar Besar Kota Malang)" ini adalah SIG berbasis web (*webGIS*) yang dibangun menggunakan bantuan perangkat lunak Quantum GIS dan teknik analisis *micro-spatial* dengan tujuan menampilkan informasi berupa koordinat serta detail tiap-tiap kios yang terdapat pada Pasar Besar. Selain itu, sistem dapat memvisualisasikan dan melakukan pencarian terhadap komoditas tertentu berdasarkan radius.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan maka rumusan masalah yang ingin dibahas pada penelitian ini adalah:

1. Bagaimana cara mengetahui karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas di Pasar Besar?
2. Bagaimana mengembangkan *webGIS* untuk visualisasi populasi komoditas di Pasar Besar?
3. Apa hasil pengujian terhadap tingkat akurasi *dasymetric algorithm* menggunakan metode *RMSE*?

1.3 Tujuan

Berdasarkan latar belakang masalah yang telah dijelaskan maka tujuan yang ingin dicapai pada penelitian ini adalah:

1. Mengetahui karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas di Pasar Besar.
2. Mengembangkan *webGIS* untuk visualisasi populasi komoditas di Pasar Besar.
3. Mengetahui hasil pengujian terhadap tingkat akurasi *dasymetric algorithm* menggunakan metode *RMSE*.

1.4 Manfaat

Manfaat yang bisa didapat dari penelitian ini adalah sebagai berikut:

1. Membantu pelanggan pasar tradisional dalam melakukan pencarian barang terkait pada pasar tradisional.
2. Membantu pelanggan pasar tradisional mengetahui secara detail informasi mengenai pasar terkait.
3. Membantu manajemen Pasar Besar dan Dinas Pasar dalam mengelola data pedagang dan data komoditas.
4. Mengetahui komoditas dengan penjualan paling tinggi untuk lebih dimanfaatkan sebagai komoditas unggulan di Kota Malang.

1.5 Batasan Masalah

Adapun batasan masalah dari permasalahan tersebut adalah sebagai berikut:

1. Batas wilayah pada penelitian ini berfokus pada Pasar Besar Kota Malang.
2. Data yang digunakan adalah data yang diperoleh dari Dinas Pasar Kota Malang, Pasar Besar Kota Malang, dan hasil survei penulis.
3. Karakteristik kepadatan populasi komoditas dibedakan berdasarkan klasifikasi komoditas yang ditentukan oleh penulis.
4. Jumlah total komoditas untuk perhitungan *dasymetric algorithm* diambil dari banyaknya kios yang dimiliki oleh pedagang terkait.
5. Pembahasan masalah ditekankan pada implementasi *webGIS* untuk visualisasi populasi komoditas di Pasar Besar Kota Malang.
6. Solusi yang dihasilkan adalah lokasi kios serta data yang ada pada kios terkait.
7. Luas kios diambil dari data yang diperoleh dari Dinas Pasar Kota Malang.

1.6 Sistematika Pembahasan

Bagian ini berisi struktur penulisan penelitian dimulai dengan bab pendahuluan sampai bab penutup dan deskripsi singkat dari masing-masing bab. Sistematika pembahasan pada penulisan penelitian ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Menguraikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB II KAJIAN DAN DASAR TEORI

Menguraikan teori-teori yang menjadi penunjang dalam pelaksanaan penelitian.

BAB III METODE PENELITIAN

Menguraikan metode penelitian atau langkah kerja yang dilakukan pada penelitian.

BAB IV REKAYASA PERSYARATAN/KEBUTUHAN

Menguraikan proses analisis dan perancangan sistem.

BAB V IMPLEMENTASI

Menguraikan proses implementasi dari sesuai dengan analisis dan perancangan sistem.

BAB VI PENGUJIAN

Menguraikan proses pengujian, hasil pengujian, dan analisis terhadap sistem yang telah direalisasikan.

BAB VII PENUTUP

Menguraikan kesimpulan yang diperoleh dari pembuatan dan pengujian sistem, serta saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bagian ini berisi kajian pustaka dan dasar teori yang digunakan untuk menunjang penelitian. Kajian pustaka berisi tentang pembahasan penelitian sebelumnya yang berkaitan dengan penelitian ini. Sedangkan dasar teori berisi tentang pembahasan teori yang digunakan untuk menyusun penelitian.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini mengacu pada penelitian berjudul "*Interactive Online Micro-spatial Population Analysis Based on GIS Estimated Building Population*" oleh Ko Ko Lwin dan Yuji Murayama pada tahun 2010. Penelitian tersebut membahas tentang metode estimasi populasi bangunan pada Kota Tsukuba, Jepang. Penelitian tersebut memanfaatkan pengindraan jauh *(remote sensing)* yaitu data *Light Detection and Ranging (LIDAR)* yang berasal dari *Digital Volume Model (DVM)*. Metode yang digunakan pada penelitian tersebut adalah "*Dasymetric Mapping*". Terdapat dua metode estimasi pada penelitian tersebut yaitu: (1) *Areametric* (untuk bangunan yang tidak memerlukan informasi banyak lantai); dan (2) *Volumetric* (untuk bangunan yang memerlukan informasi banyak lantai). Hasil dari penelitian ini yaitu *webGIS* yang merupakan visualisasi data populasi bangunan dan integrasi data LIDAR dengan data demografi yang memberikan manfaat untuk meningkatkan keputusan perencanaan kota.

Implementasi SIG sebagian besar dimanfaatkan untuk merepresentasikan keadaan suatu wilayah yang luas dan mencerminkan karakteristik tempat terbuka (*outdoor*). Ming-Dawa Su, et al. merepresentasikan pola spasial distribusi populasi untuk perencanaan ekonomi sosial dan manajemen keputusan seperti alokasi sumber pangan dan medis, transportasi, penggunaan lahan dan pembangunan wilayah Kota Taipei, Taiwan (Su, et al., 2010). Sedangkan Rachel Sleeter menerapkan penggunaan *dasymetric mapping* untuk merepresentasikan SIG data penduduk Kota Alameda, California menggunakan data survei National Land Cover Data Set dan sumber data *land-cover* lainnya (Sleeter, 2001). Kedua penelitian tersebut mengimplementasikan SIG pada tempat terbuka.

Pada penelitian ini akan diambil analisis *micro-spatial* menggunakan metode *dasymetric mapping* dari penelitian yang dilakukan oleh Ko Ko Lwin dan Yuji Murayama pada tahun 2010 dan Rachel Sleeter pada tahun 2001, pengujian *Root Mean Square Error (RMSE)* untuk melakukan pengujian terhadap *dasymetric algorithm* dari penelitian yang dilakukan oleh Ming Dawa Su, dkk pada tahun 2010.

2.1.1 Metode *Dasymetric*

Penelitian berjudul *"Dasymetric Mapping Techniques for Then San Francisco Bay Region, California"* oleh Rachel Sleeter digunakan sebagai kajian pustaka kedua. Penelitian Rachel Sleeter (Sleeter, 2001) menerapkan metode *Dasymetric Mapping* untuk merepresentasikan SIG data penduduk Kota Alameda, California menggunakan data survei *National Land Cover Data Set* dan sumber data *land-*

cover lainnya. *Dasymetric mapping* merupakan solusi potensial untuk menggambarkan data populasi yang telah dikumpulkan menjadi unit area (Sleeter, 2001). Persamaan (2.1) merupakan *dasymetric* algorithm yang digunakan untuk mendapatkan karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas pada Pasar Besar (Su, et al., 2010).

$$D_{if} = \frac{K_i}{\sum_{f=1}^{n}(A_{if})} A_{if} \quad \textbf{(2.1)}$$

D_{if} = kepadatan klasifikasi i di lantai f
K_i = jumlah total kios klasifikasi i
A_{if} = luas area kios klasifikasi i di lantai f
i = Komoditas
f = Lantai

Penggunaan metode *dasymetric* umumnya digunakan untuk mengolah detail informasi pada wilayah yang luas seperti informasi data populasi penduduk, sedangkan pada penelitian ini metode *dasymetric* digunakan untuk mengolah detail informasi data populasi komoditas.

2.1.2 Pengujian *RMSE*

Penelitian berjudul *"Multi-layer Multi-class Dasymetric Mapping To Estimate Population Distribution"* oleh Ming Dawa Su, dkk digunakan sebagai kajian pustaka ketiga. Penelitian Ming Dawa Su, dkk (Su, et al., 2010) menerapkan metode *Root Mean Square Error (RMSE)* untuk melakukan pengujian terhadap *dasymetric algorithm*. Persamaan (2.2) merupakan persamaan metode pengujian *RMSE*.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2}{n}} \quad \textbf{(2.2)}$$

x_i = Data hasil perhitungan sistem ke-i
$\bar{x}_i$ = Rata-rata data ke-i
n = Jumlah data
$\bar{x}_i$ diperoleh dari: $\frac{|D_{if}manual + D_{if}sistem|}{n}$

2.2 Dasar Teori

Pada sub bab ini berisi tentang pembahasan teori yang digunakan untuk menyusun penelitian.

2.2.1 Sistem Informasi Geografis

SIG dapat diartikan sebagai sebuah sistem yang mampu mendukung pengambilan keputusan, mengolah data spasial, dan mengintegrasikan data spasial dengan karakteristik non spasial atau atribut data. Pengembangan SIG terdiri dari metodologi dan teknologi yang diperlukan, yaitu data spasial, perangkat keras, perangkat lunak, dan struktur organisasi. SIG memberikan kemudahan untuk mengelola, menganalisis dan memetakan informasi spasial dan

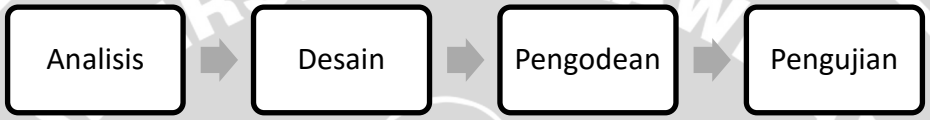
data atributnya dengan akurasi kartografi yang merupakan perpaduan antara perangkat keras dan perangkat lunak (Prahasta, 2009).

2.2.2 Tahapan *SDLC*

Software Development Life Cycle (SDLC) merupakan tahapan yang digunakan dalam melakukan pengembangan perangkat lunak menggunakan model-model dan metodologi tertentu (A.S. & Shalahuddin, 2014).

2.2.2.1 Model Waterfall

Salah satu model *SDLC* ini menyediakan alur pengembangan perangkat lunak secara sekuensial dan terurut mulai dari tahapan analisis, desain, pengembangan, pengujian dan tahapan pendukung. Gambar 2.1 merupakan tahapan *SDLC* menurut model *waterfall* (A.S. & Shalahuddin, 2014).



Gambar 2.1 Diagram tahapan metode *waterfall*

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

a. Analisis kebutuhan perangkat lunak
   Proses pengumpulan kebutuhan perangkat lunak. Pada tahapan ini dilakukan pendokumentasian spesifikasi perangkat lunak.
b. Desain
   Proses dalam mendesain perangkat lunak yang akan dibangun, termasuk struktur data, arsitektur perangkat lunak, antarmuka, dan prosedur pengodean.
c. Pembuatan kode program
   Proses dalam mengubah perancangan menjadi program perangkat lunak. Hasil dari tahapan ini adalah program komputer atau sistem yang telah sesuai dengan tahapan perancangan yang dilakukan pada tahapan sebelumnya.
d. Pengujian
   Tahapan dalam melakukan pengujian sistem. Pengujian dilakukan dari segi fungsional dan non-fungsional. Pada tahapan ini memastikan semua bagian sudah dilakukan pengujian.

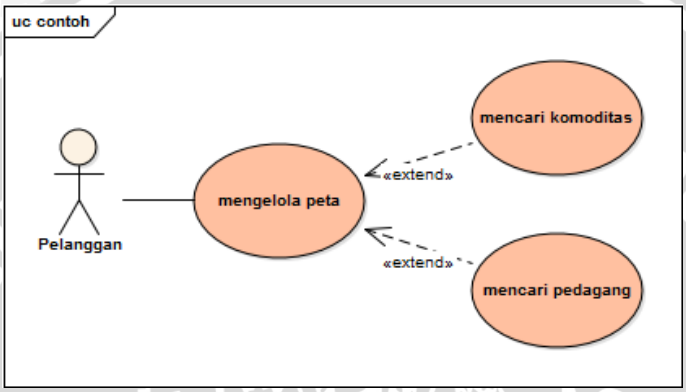
2.2.3 *Unified Modelling Language*

Unified Modelling Language (UML) adalah bahasa standar yang digunakan dan memvisualisasikan proses analisis dan desain berorientasi objek. UML merupakan bahasa pemodelan dan tidak menggambarkan bagaimana melakukan analisis dan desain berorientasi objek. Pemodelan UML tidak didominasi oleh narasi, pemodelan dapat dilakukan secara visual. Pemodelan secara visual dapat membatu menangkap struktur dan perilaku dari objek, mempermudah penggambaran interaksi antar elemen dalam sistem, dan mempertahankan

konsistensi antara desain dan implementasi dalam pemrograman. Dalam memodelkan suatu sistem UML menyediakan standar pada notasi dan diagram (Hermawan, 2015).

2.2.3.1 Use Case Diagram

Use case diagram merupakan diagram dalam UML yang mendefinisikan kebutuhan yang harus dipenuhi oleh sistem. Diagram *use case* menggambarkan hubungan antara *user*/pengguna dan fungsi dari sistem tetapi tidak membahas bagaimana rincian spesifik pelaksanaan fungsi sistem (Seidl, et al., 2012). Gambar 2.2 merupakan contoh penggunaan *use case diagram*.



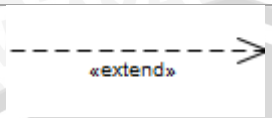
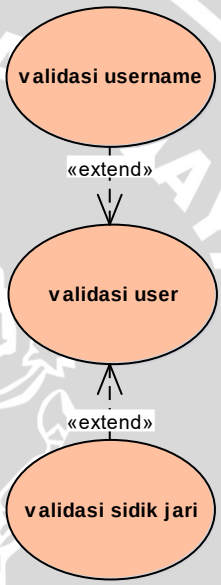
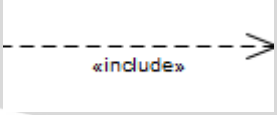
Gambar 2.2 Contoh penggunaan *use case diagram* mengelola peta

Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu dalam *Use Case Specification*. *Use case* hanya menjelaskan tentang kegiatan yang dilakukan oleh aktor dan sistem dan tidak menjelaskan bagaimana aktor dan sistem melakukan kegiatan tersebut. *Use case specification* terdiri dari nama *use case*, deskripsi singkat *(brief description)*, aliran normal *(basic flow)*, aliran alternatif *(alternate flow)*, *special requirement*, *pre-condition,* dan *post-condition* (Hermawan, 2015). Tabel 2.1 merupakan representasi *use case diagram*.

Tabel 2.1 Tabel representasi *use case diagram*

Nama	Notasi	Deskripsi
Use case	Nama Use Case	Notasi untuk menggambarkan fungsionalitas yang dimiliki oleh sistem. Penamaan *use case* dinyatakan dengan kata kerja.
Aktor	Nama Aktor	Notasi untuk menggambarkan manusia, proses atau sistem lain yang berinteraksi dengan sistem. Aktor berada di luar sistem.
Asosiasi/*association*		Notasi untuk menggambarkan komunikasi antara aktor dan *use case*. Notasi digunakan ketika

Tabel 2.1 Tabel representasi *use case diagram*

Nama	Notasi	Deskripsi
		suatu aktor ikut serta dalam pelaksanaan *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi/*extend*		Notasi yang menggambarkan relasi *use case* tambahan ke sebuah *use case*. *Use case* yang ditambahkan dapat berdiri sendiri meskipun tanpa *use case* tambahan tersebut, misalnya:  Arah panah mengarah pada *use case* yang ditambahkan.
Generalisasi/*generalization*		Hubungan generalisasi antara dan spesialisasi (umum – khusus) antara dua buah *use case* di mana fungsi yang satu merupakan fungsi yang lebih umum dari fungsi yang lainnya.
Include		Relasi tambahan ke sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat agar *use case* dapat dijalankan. Terdapat dua sudut pandang mengenai *include* pada *use case*, yaitu: 1) *include* berarti *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan dan 2) *include* berarti

Tabel 2.1 Tabel representasi *use case diagram*

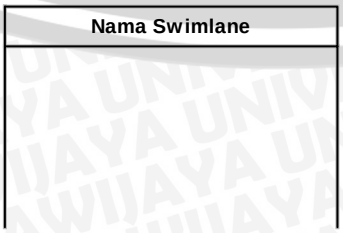
Nama	Notasi	Deskripsi
		use case tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan.

Sumber: diadaptasi dari A.S. & Shalahuddin, 2014

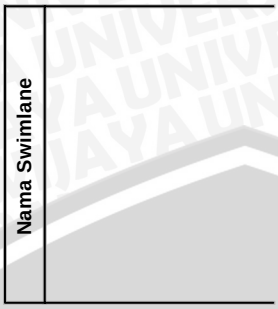
2.2.3.2 Activity Diagram

Activity diagram merupakan diagram yang menggambarkan aliran kerja atau aktivitas yang terdapat dalam sistem. Diagram ini juga dapat menggambarkan proses bisnis dan menu-menu yang terdapat dalam perangkat lunak (A.S. & Shalahuddin, 2014). *Activity diagram* menggambarkan aktivitas yang dapat dilakukan oleh sistem. Tabel 2.2 merupakan simbol-simbol yang digunakan dalam membuat *activity diagram.*

Tabel 2.2 Representasi *activity diagram*

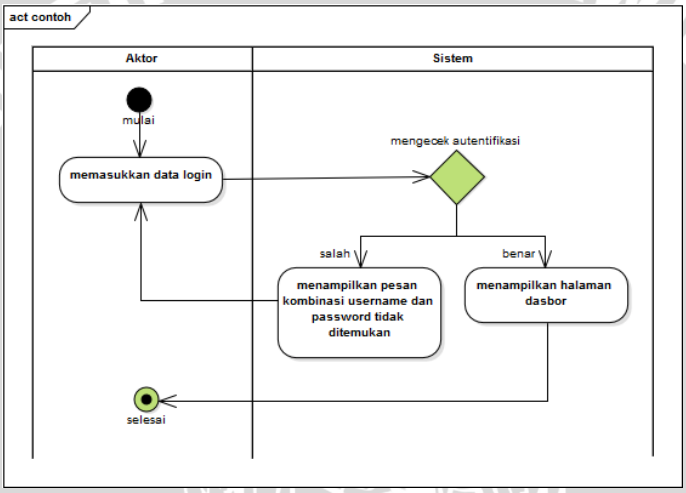
	Notasi	Deskripsi
Status Awal		Notasi yang menggambarkan status awal dari aktivitas sistem.
Aktivitas		Notasi yang menggambarkan aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision*		Notasi yang menggambarkan asosiasi percabangan dimana jika ada lebih dari satu pilihan aktivitas.
Penggabungan/*join*		Notasi yang menggambarkan asosiasi penggabungan, digunakan ketika ada lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir		Notasi yang menggambarkan status akhir yang dilakukan sistem.
Swimlane		Notasi yang digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Tabel 2.2 Representasi *activity diagram*

	Notasi	Deskripsi
		

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

Gambar 2.3 adalah contoh penggunaan *activity diagram*.



Gambar 2.3 Contoh penggunaan *activity diagram login*

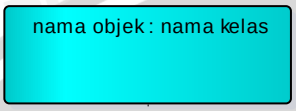
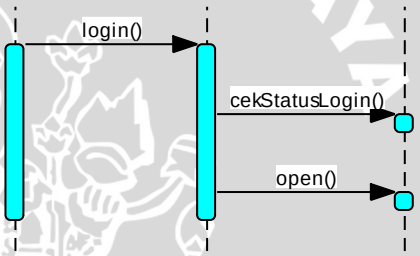
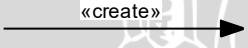
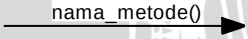
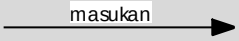
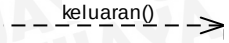
2.2.3.3 Sequence Diagram

Sequence diagram adalah diagram yang menjelaskan secara detail mengenai urutan proses yang dilakukan di dalam sistem dalam mencapai tujuan *use case*. Diagram ini mendeskripsikan waktu hidup dari objek dan pesan yang dikirim maupun diterima antar objek (A.S. & Shalahuddin, 2014). Tabel 2.3 merupakan simbol-simbol yang digunakan dalam membuat *sequence diagram.*

Tabel 2.3 Tabel representasi *sequence diagram*

Nama	Notasi	Deskripsi
Aktor	Nama Aktor	Notasi yang menggambarkan orang, proses, atau sistem lain yang berinteraksi dengan sistem. Notasi ini berada di luar sistem.

Tabel 2.3 Tabel representasi *sequence diagram*

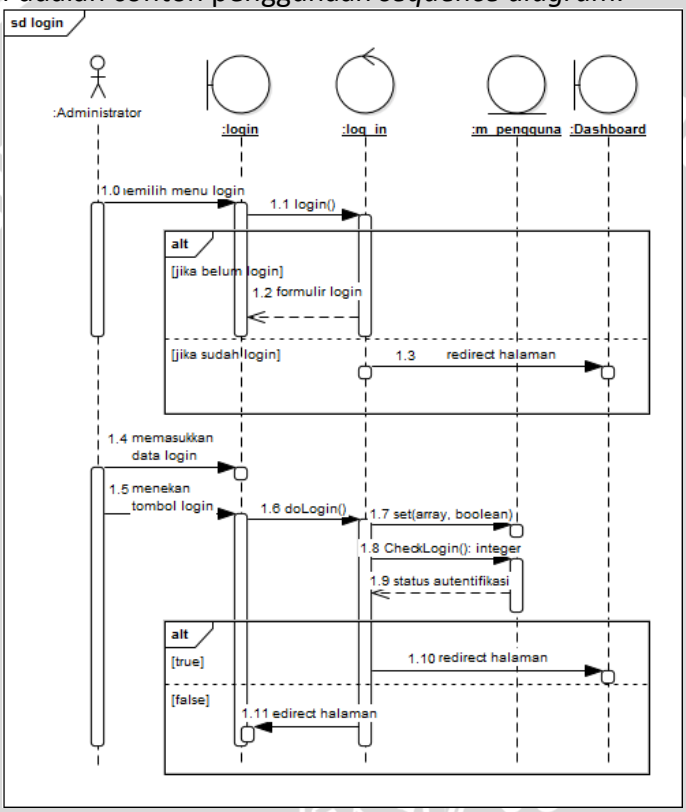
Nama	Notasi	Deskripsi
Lifeline		Notasi yang menyatakan kehidupan suatu objek.
Objek		Notasi yang menyatakan objek.
Activation bar		Notasi yang menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misal :  Maka cekStatusLogin() dan open() dilakukan di dalam *method* login().
Pesan tipe *create*		Notasi yang menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *call*		Notasi yang menyatakan suatu objek memanggil operasi / *method* yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi / *method.* Operasi / *method* harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe *send*		Notasi yang menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang menerima pesan.
Pesan tipe *return*		Notasi yang menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian objek tertentu, arah panah

Tabel 2.3 Tabel representasi *sequence diagram*

Nama	Notasi	Deskripsi
		mengarah pada objek yang menerima kembalian.
Pesan tipe *destroy*	«destroy»	Notasi yang menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

Sumber: diadaptasi dari A.S. & Shalahuddin, 2014

Gambar 2.4 adalah contoh penggunaan *sequence diagram.*



Gambar 2.4 Contoh penggunaan *sequence diagram login*

2.2.3.4 Class Diagram

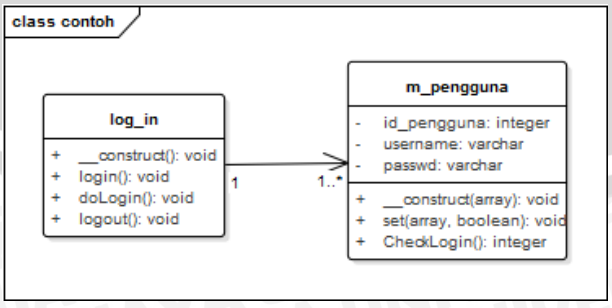
Class diagram merupakan diagram UML yang menggambarkan struktur sistem ke dalam pendefinisian kelas-kelas yang dibuat dalam perancangan. *Class* terdiri dari atribut dan metode. Atribut merupakan variabel yang dimiliki oleh sebuah *class*. Sedangkan metode merupakan fungsi-fungsi yang dimiliki oleh suatu *class*. Tujuan dibuatnya *class diagram* adalah memudahkan pembuat program dalam membangun sebuah sistem yang sesuai dengan pendefinisian *class* pada tahap perancangan. *Class* yang terdapat dalam tahapan perancangan diharuskan dapat melakukan fungsi-fungsi yang sesuai dengan kebutuhan sistem (A.S. & Shalahuddin, 2014). Tabel 2.4 merupakan simbol-simbol yang ada dalam pembuatan *class diagram*.

Tabel 2.4 Tabel representasi *class diagram*

Nama	Notasi	Deskripsi
Class / kelas		Notasi yang menggambarkan kelas pada struktur sistem.
Interface / antarmuka		Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
Asociation / asosiasi		Notasi yang menggambarkan relasi antar kelas dengan makna umum. Asosiasi biasanya disertai dengan *multiplicity*.
Directed association / asosiasi berarah		Notasi yang menggambarkan relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain.
Generalisasi		Notasi yang menggambarkan relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
Dependency / kebergantungan		Notasi yang menggambarkan relasi antar kelas dengan makna kebergantungan antar kelas.
Aggregation / agregasi		Notasi yang menggambarkan relasi antarkelas degan makna semua-bagian (*whole-part*)

Sumber: diadaptasi dari A.S. & Shalahuddin, 2014

Contoh penggunaan *class diagram* dapat dilihat dalam Gambar 2.5.



Gambar 2.5 Contoh penggunaan *class diagram login*

2.2.4 Pengujian *Black-box*

Pengujian *Black-box*, juga disebut pengujian perilaku. Pengujian *Black-box* berfokus pada persyaratan fungsional perangkat lunak. Uji kasus dibangun di sekitar spesifikasi dan persyaratan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat digunakan untuk menguji kebutuhan fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih *input* yang valid dan tidak valid dan menentukan *output* yang benar. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Metode uji coba *Black-box* memfokuskan pada keperluan fungsional dari *software*. Karna itu uji coba *Black-box* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program (Pressman, 2010). Pengujian *Black-box* berusaha menemukan kesalahan dalam kategori berikut:

a. salah atau fungsi yang hilang,
b. kesalahan antarmuka,
c. kesalahan dalam struktur data atau akses eksternal basis data, dan
d. kesalahan *performance*

2.2.5 Pengujian *Portability*

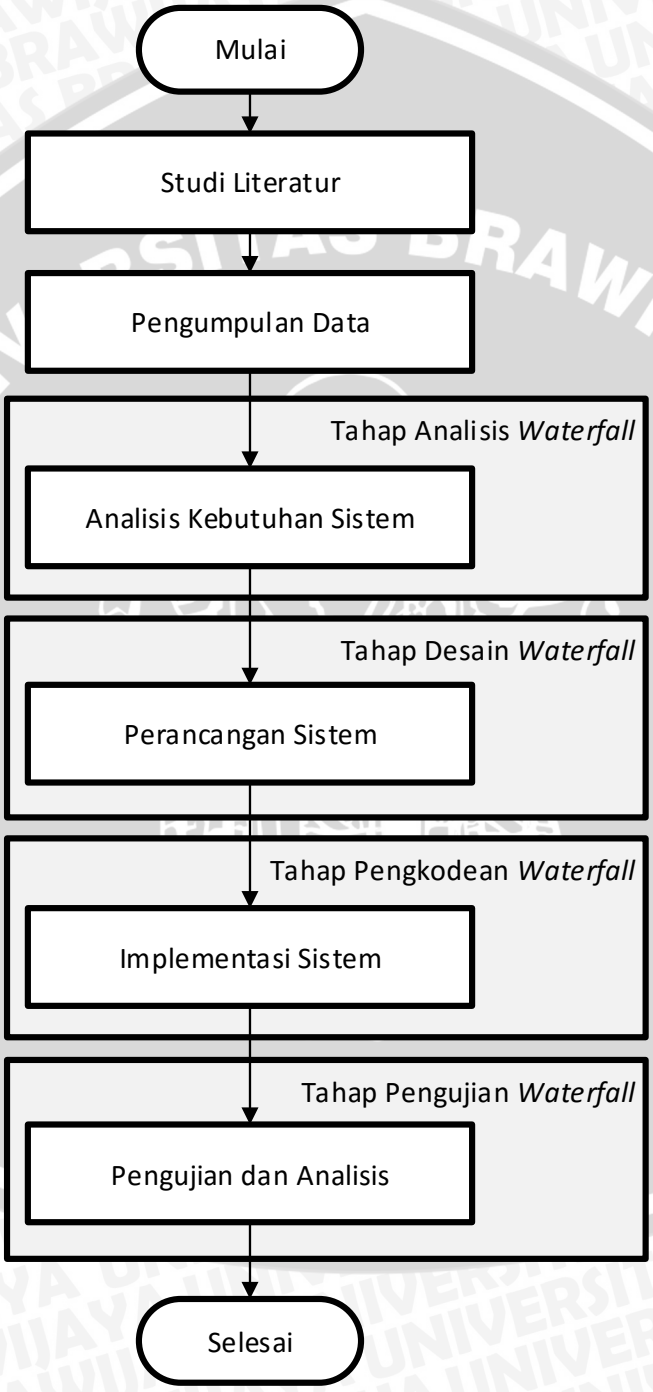
Pengujian *Portability* adalah usaha yang diperlukan untuk memindahkan program dari satu perangkat keras dan atau lingkungan sistem perangkat lunak ke yang lainya (Pressman, 2010). Karakteristik *Portability* sebuah perangkat lunak berbasis web dapat dinilai melalui kompabilitas terhadap 80% desktop browser dan juga 80% *mobile browser* yang digunakan sebagian besar masyarakat didunia. Maka dari itu diperlukan uji *Portability* perangkat lunak berbasis web tersebut pada tujuh *desktop browser* dan lima *mobile browser* (Salonen, 2012).

Pengujian kompabilitas terhadap berbagai jenis browser ini dapat menggunakan sebuah perangkat lunak SortSite. SortSite dapat menguji kompabilitas dari sebuah perangkat lunak berbasis web dengan menjalankan *virtual browser* dengan melakukan pengujian antarmuka sistem baik gambar, CSS, html, dan *javaScript* pada sistem yang diuji. SortSite dapat memperlihatkan *error* (*missing file*) dan *warning* (*major* dan *minor performance*) dari antarmuka yang diuji. Maka dari itu, pemenuhan karakteristik *portability* dengan menguji kompabilitas browser dapat dilihat dengan tidak adanya *error* maupun *warning* dari sistem (PowerMapper, 2016).

Critical Issues mengindikasikan sebuah fitur yang tidak didukung sama sekali pada satu atau lebih *browser*, sehingga pengguna tidak dapat mengakses fitur tersebut. *Major Issues* mengindikasikan sebuah fitur yang sebagian besar fungsionalitasnya tidak berjalan dengan baik pada satu atau lebih *browser* dan *Minor Issues* mengindikasikan sebuah permasalahan yang hanya mempengaruhi pengalaman pengguna seperti elemen dalam halaman *web* yang memiliki tampilan berbeda pada beberapa *browser*.

BAB 3 METODOLOGI

Pada bab ini berisi tentang metode atau langkah – langkah yang digunakan dalam penelitian untuk membuat sistem informasi geografis yang sesuai dengan kebutuhan. Adapun tahapan penelitian dapat digambarkan dalam bentuk diagram alir yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir metode penelitian

3.1 Studi Literatur

Tahap ini bertujuan untuk memenuhi kebutuhan teori dan informasi yang mendukung pengolahan data sebagai sumber acuan dalam penelitian ini. Pustaka dan teori yang terkait dengan penelitian ini meliputi:

1. Kajian Pustaka
2. Metode *Dasymetric*
3. Pengujian *RMSE*
4. Sistem Informasi Geografis
5. Tahapan *SDLC*
6. *Unified Modelling Language*
7. Pengujian *Black-box*
8. Pengujian *Portability*

3.2 Pengumpulan Data

Pengumpulan data pada penelitian ini menggunakan data primer dan data sekunder. Data primer merupakan data yang didapatkan secara langsung dari objek yang diteliti. Sedangkan data sekunder adalah data yang telah diolah oleh pihak lain, dalam hal ini adalah pemerintah, data dari instansi-instansi terkait, berupa data statistik dan informasi tertulis lainnya. Adapun data yang dibutuhkan adalah sebagai berikut:

Data Primer

a. Data lokasi Pasar Besar. Data tersebut di dapatkan melalui pengambilan koordinat menggunakan GPS.
b. Data Informasi setiap kios Pasar Besar (data penjual, data komoditas yang dijual dan lain sebagainya). Data tersebut diperoleh dengan cara wawancara dan observasi yang dilakukan di setiap kios.

Data Sekunder

a. Data denah Pasar Besar. Data tersebut diperoleh dari pengelola Pasar Besar.
b. Data Pedagang Pasar Besar. Data tersebut diperoleh dari Dinas Pasar Kota Malang.
c. Data pendukung objek penelitian. Data tersebut diperoleh dari Dinas Pasar Kota Malang. Data yang dimaksud adalah informasi yang mendukung untuk memilih objek penelitian. Objek penelitian sendiri dipilih berdasarkan indikator pasar yang memiliki keanekaragaman komoditas yang paling banyak dan pusat sarana pelayanan dan jasa serta pusat kegiatan perekonomian Kota Malang dan daerah sekitarnya.

3.3 Analisis Kebutuhan Sistem

Dalam tahap ini dilakukan analisis kebutuhan sistem yang bertujuan untuk mendapatkan semua kebutuhan yang diperlukan untuk pengembangan sistem.

Tahap ini menggunakan Bahasa pemodelan UML. Tahap analisis kebutuhan terdiri dari tiga langkah, yaitu:

1. Melakukan penjabaran gambaran umum proses yang sedang berlangsung di Pasar Besar dan proses dari sistem terkait pencarian komoditas di Pasar Besar.
2. Melakukan proses identifikasi pengguna yang terlibat dalam sistem, identifikasi kebutuhan fungsional, identifikasi kebutuhan non-fungsional.
3. Membuat daftar kebutuhan pengguna menggunakan pemodelan *use case diagram.*

3.4 Perancangan Sistem

Perancangan sistem perangkat lunak dilakukan setelah semua kebutuhan perangkat lunak didapatkan melalui tahap analisis kebutuhan. Perancangan perangkat lunak berdasarkan *object-oriented design* yaitu menggunakan pemodelan UML. Perancangan dimulai dari perancangan arsitektural. Perilaku sistem dimodelkan dengan diagram aktivitas, hubungan interaksi antar objek dimodelkan dengan diagram *sequence*, hubungan antar kelas dimodelkan dengan diagram kelas dan perancangan struktur basis data berdasarkan kelas model pada perancangan diagram kelas yang telah dilakukan pada tahap perancangan serta yang terakhir adalah perancangan antarmuka.

3.5 Implementasi Sistem

Pada tahapan ini, dilakukan pengembangan *webGIS*. Peneliti menggunakan *library leafletjs* dengan bahasa pemrograman javascript, HTML, CSS, dan PHP untuk dapat membangun *webGIS*. SIG yang dikembangkan menggunakan framework CodeIgniter. Sedangkan untuk media penyimpanan menggunakan PostGIS dan PostgreSQL. Pengembangan sistem analisis *micro-spatial* dilakukan dalam beberapa tahap, antara lain:

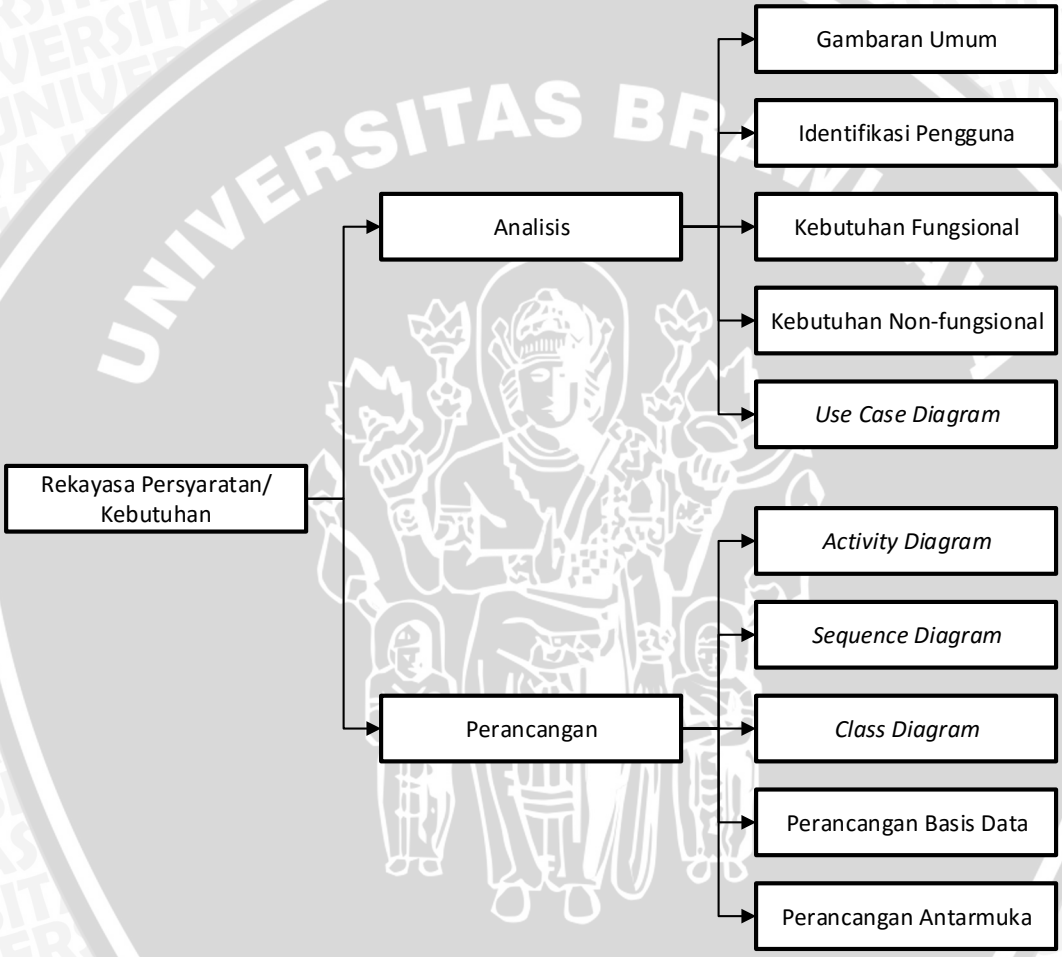
a. Digitalisasi peta pasar besar.
b. Implementasi pada perangkat SIG.
c. Implementasi *dasymetric*.

3.6 Pengujian dan Analisis

Pada tahapan ini, dilakukan pengujian terhadap hasil pengembangan sistem yang telah dilakukan. Pengujian sistem pada penelitian ini dilakukan agar dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang telah melandasinya. Pengujian ini dilakukan dengan pengujian *Black-Box* dan pengujian *Portability.* Pada tahapan ini juga dilakukan pengujian terhadap *dasymetric algorithm* menggunakan pengujian *RMSE*.

BAB 4 REKAYASA PERSYARATAN/KEBUTUHAN

Bab ini memaparkan analisis dan perancangan dari sistem yang akan dibuat. Rancangan ini dibuat untuk analisis *micro-spatial*. Analisis yang dilakukan meliputi gambaran umum, idetifikasi pengguna, kebutuhan fungsional, kebutuhan non fungsional, dan *use case diagram*. Sedangkan pada tahap perancangan meliputi *activity diagram, sequence diagram, class diagram,* perancangan basis data, dan perancangan antarmuka. Diagram blok rekayasa persyaratan/kebutuhan dapat dilihat dalam Gambar 4.1.



Gambar 4.1 Diagram blok rekayasa persyaratan/kebutuhan

4.1 Analisis

Pada tahap ini dijelaskan analisis awal yang dibutuhkan sebelum melakukan tahap perancangan. Proses analisis meliputi gambaran umum, identifikasi pengguna yang terlibat dalam sistem, identifikasi kebutuhan fungsional, identifikasi kebutuhan non-fungsional, dan *use case diagram*.

4.1.1 Gambaran Umum

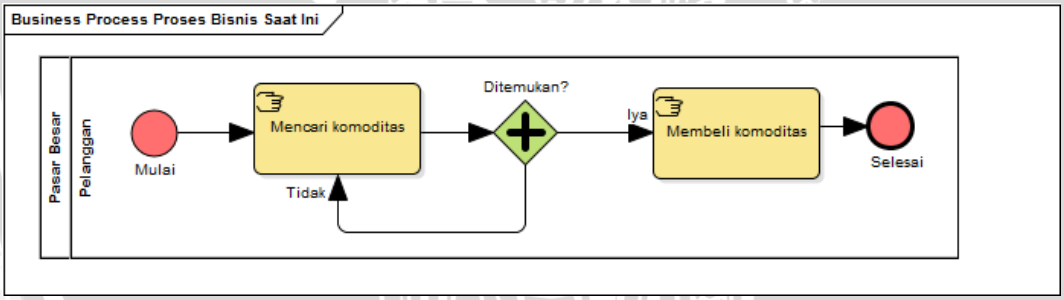
Sistem analisis *micro-spatial* merupakan sistem yang dapat membantu kegiatan operasional di Pasar Besar dan Dinas Pasar. Sistem ini mencakup dua fungsi utama yaitu pengelolaan peta Pasar Besar dan pengelolaan hasil perhitungan *dasymetric algorithm* yang nantinya dapat digunakan sebagai bahan analisis oleh pihak pengelola Pasar Besar maupun pihak pengelola Dinas Pasar.

Gambar 4.2 Logo ApsOrcim

Nama sistem analisis *micro-spatial* ini adalah ApsOrcim. Gambar 4.2 merepresentasikan logo dari sistem yang dikembangkan. Kata ApsOrcim sendiri diambil dari kebalikan kata "microspa" yang terinspirasi dari bahasa khas Kota Malang yang populer di daerah Malang.

4.1.1.1 Proses Bisnis Pasar Besar Saat Ini

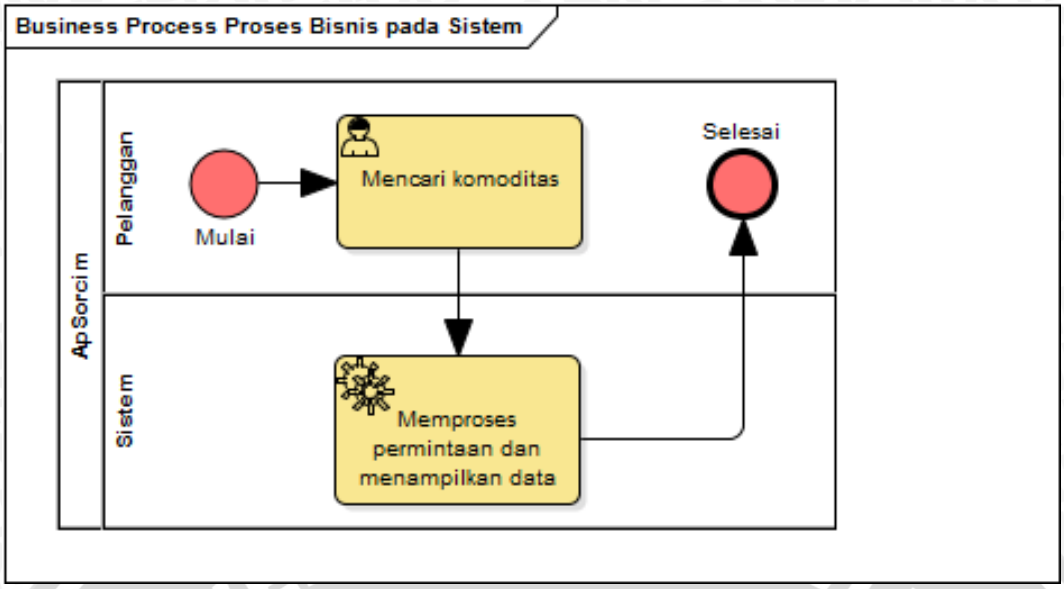
Dalam melakukan pencarian komoditas pada Pasar Besar, saat ini pelanggan cenderung melakukannya secara manual yang mana hal tersebut menimbulkan perulangan secara tidak langsung. Apabila pelanggan belum mendapatkan komoditas yang dicari di salah satu kios, maka pelanggan tersebut akan mencari lagi di kios lainnya. Gambar 4.3 menunjukkan proses bisnis saat ini.



Gambar 4.3 Proses bisnis Pasar Besar saat ini

4.1.1.2 Proses Bisnis Pasar Besar pada Sistem

Gambar 4.4 menggambarkan proses bisnis pada sistem yang mana dapat mempercepat pencarian komoditas oleh pelanggan dengan hasil yang lebih pasti. Pelanggan memulai proses bisnis dengan memilih komoditas yang dicari pada sistem, kemudian sistem akan memproses permintaan dan menampilkan lokasi pencarian pada peta.



Gambar 4.4 Proses bisnis Pasar Besar pada Sistem

Sistem ApsOrcim berfokus pada kepadatan populasi komoditas berdasarkan karakteristik klasifikasi komoditas yang akan dibantu oleh *dasymetric algorithm* dan persebaran komoditas yang ditampilkan melalui peta, sehingga didapatkan 2 fitur utama yaitu:

1. mengelola peta

   Fitur ini merupakan layanan sistem yang menyajikan data Pasar Besar melalui visualisasi berupa peta. Layanan ini menyediakan pengelolaan terhadap peta tersebut, sehingga pengguna dapat melakukan pencarian kios berdasarkan pedagang dengan parameter masukan berupa nama pedagang dan memfilter kios berdasarkan komoditas yang dijual dengan parameter masukan komoditas.

2. mengelola *dasymetric*

   Fitur ini merupakan fitur yang menyediakan layanan untuk mengelola data *dasymetric* yang meliputi data hasil perhitungan yang disajikan melalui data tabular, grafik, dan peta untuk visualisasi data hasil perhitungan *dasymetric algorithm* berdasarkan area pilihan pengguna menggunakan bantuan *map tools* dan parameter tertentu.

4.1.2 Identifikasi Pengguna

Pada tahap ini dilakukan proses identifikasi pengguna yang akan berinteraksi dengan sistem. Tabel 4.1 menunjukkan pengguna yang terlibat beserta penjelasannya.

Tabel 4.1 Identifikasi pengguna

	Identifikasi Pengguna	Karakteristik	Hak Akses ke Sistem
1.	Administrator	Admin merupakan pengguna dengan hak akses paling tinggi dibandingkan dengan pengguna yang lain. Kebutuhan pokok administrator adalah sebagai berikut: 1. *username* dan *password* untuk dapat masuk ke dalam sistem dan mendapatkan otoritas sebagai administrator. 2. dapat melihat dan memanipulasi semua data pedagang Pasar Besar yang digunakan sebagai data penunjang *dasymetric algorithm*.	MSpa-01 MSpa-02 MSpa-04 MSpa-05 MSpa-06 MSpa-07
2.	Pengguna Umum	Pengguna umum merupakan pengguna yang tidak terdaftar dalam sistem dan tidak memiliki hak akses *login* seperti yang dijelaskan pada Tabel 4.2 dengan kode fungsi Mspa-01. Berikut kebutuhan pokok pengguna umum, yaitu: 1. dapat menggunakan layanan untuk mengelola peta yaitu layanan yang digunakan untuk mencari kios berdasarkan pedagang dan memfilter kios berdasarkan komoditas yang dijual.	MSpa-03

4.1.3 Identifikasi Kebutuhan Fungsional

Daftar kebutuhan fungsional merupakan daftar layanan yang harus disediakan oleh sistem untuk memenuhi kebutuhan pengguna. Spesifikasi kebutuhan fungsional ditunjukkan pada Tabel 4.2 yang terdiri dari kode fungsi, nama fungsi, deskripsi, dan prioritas sebagai penjelasan layanan utama sistem.

Tabel 4.2 Daftar kebutuhan fungsional

No	Kode Fungsi	Nama Fungsi	Deskripsi	Prioritas
1.	MSpa-01	*Login*	Sistem menyediakan layanan *login* dengan parameter masukan berupa *username* dan *password*, sehingga hanya yang memiliki otoritas sebagai administrator yang dapat mengolah data dalam sistem.	RENDAH
2.	MSpa-02	Mengelola dasbor	Sistem menyediakan layanan untuk mengelola *dasbor. Dasbor* merupakan fasilitas yang menunjukkan ringkasan analisis sistem berupa kumpulan grafik.	RENDAH
3.	MSpa-03	Mengelola peta	Sistem menyediakan layanan untuk mengelola peta, sehingga dapat melakukan pencarian kios berdasarkan pedagang dengan parameter masukan	TINGGI

Tabel 4.2 Daftar kebutuhan fungsional

No	Kode Fungsi	Nama Fungsi	Deskripsi	Prioritas
			berupa nama pedagang, dan memfilter kios berdasarkan komoditas yang dijual dengan parameter masukan komoditas.	
4.	MSpa-04	Mengelola *dasymetric*	Sistem menyediakan layanan untuk mengelola data *dasymetric* yang meliputi data tabular, grafik dan peta dengan parameter masukan *longitude, latitude*, radius dan komoditas sehingga dapat ditampilkan visualisasi hasil perhitungan *dasymetric algorithm* berdasarkan area terpilih.	TINGGI
5.	MSpa-05	Mengelola data pedagang	Sistem menyediakan layanan untuk mengelola data pertanian yang meliputi penambahan (parameter masukan nama, alamat, nomor kios, komoditas, dan kota distributor), pembaruan, dan penghapusan data pedagang.	RENDAH
6.	MSpa-06	Mengelola data komoditas	Sistem menyediakan layanan untuk mengelola data komoditas yang meliputi penambahan (parameter masukan nama komoditas dan klasifikasi komoditas), pembaruan, dan penghapusan data komoditas.	RENDAH
7.	MSpa-07	*Logout*	Sistem menyediakan layanan *logout* agar dapat keluar dari sistem.	RENDAH

4.1.4 Identifikasi Kebutuhan Non-fungsional

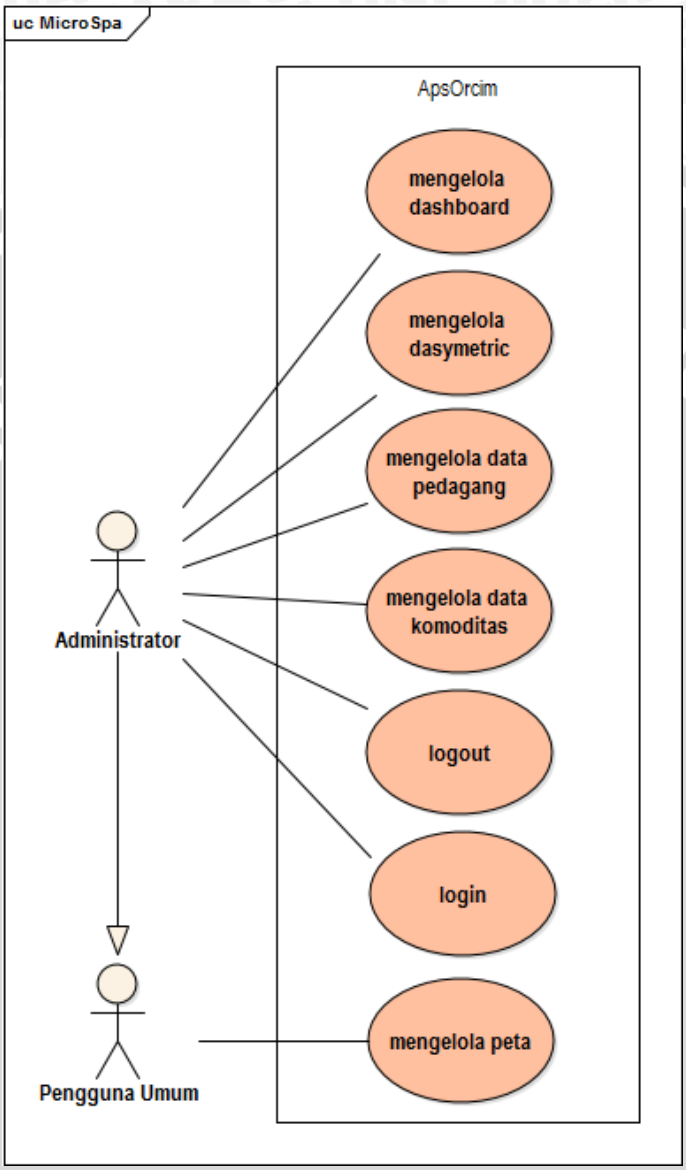
Tabel 4.3 merupakan deskripsi kebutuhan non-fungsional yang dibutuhkan oleh sistem dengan perspektif sistem secara umum.

Tabel 4.3 Daftar kebutuhan non-fungsional

No	Kode Fungsi	Parameter	Deskripsi
1.	MSpa-01	*Portability*	Sistem dapat berjalan dengan baik dalam lingkungan *desktop browser* (Edge v13, Firefox ≤ v45 dan v46, Safari v9.0, Opera v36, Google Chrome ≤ v45 dan v49) dan *mobile browser* (iOS v6.0, v8.0, dan v9.0, Android v4.0, dan BlackBerry v10.0) serta memiliki tingkat kompabilitas ≥ 80%.

4.1.5 *Use Case Diagram*

Use case diagram merupakan salah satu diagram UML yang digunakan untuk mengetahui perilaku sistem. *Use case diagram* menunjukkan hubungan antara sekumpulan *use case* dan aktor pada sistem yang dibangun. Gambar 4.5 merupakan *use case diagram* dari sistem.



Gambar 4.5 *Use case diagram*

4.1.5.1 Skenario Use Case

Skenario *use case* menjabarkan *use case diagram* pada Gambar 4.5 secara lebih detail. Tabel skenario *use case* terdiri dari 1) *use case name,* 2) kode fungsi, 3) *brief description*, 4) *actor,* 5) *Pre-conditions*, 6) *flow of events* yang terdiri dari *basic flows, alternative flows* dan *sub flows,* 7) *post-conditions*.

1. Skenario Login

Kebutuhan fungsional yang disediakan oleh sistem agar pengguna dapat masuk *(login)* ke dalam sistem. Kebutuhan tersebut dijabarkan oleh *use case* **login**, yang dipaparkan oleh Tabel 4.4.

Tabel 4.4 Skenario *use case login*

***Use Case Name* : Login**		
Kode Fungsi	MSpa-01	
Brief Description	Pengguna dapat melakukan *login* untuk mendapatkan otoritas sebagai administrator yang dapat mengolah data dalam sistem.	
Actor	Administrator	
Pre-conditions	Administrator mengakses sistem.	
Flows of Events		
Basic Flows	Aksi Aktor	Respon Sistem
	1. Administrator mengakses modul untuk melakukan *login*.	2. Sistem menampilkan halaman *login*.
	3. Administrator mengisi data *username* dan *password*.	4. Sistem menerima data *login* kemudian melakukan cek data masukan administrator.
		5. Sistem menampilkan halaman *dasbor*.
Sub Flows	-	
Alternative Flows	Terdapat data yang belum diisi pada *basic flow* nomor 3.	
		4. Sistem menampilkan pesan bahwa kombinasi *username* dan *password* tidak ditemukan. Kembali ke *basic flow* nomor 3.
Post-conditions	Administrator berhasil melakukan *login*.	

2. Skenario Mengelola Dasbor

Kebutuhan fungsional yang disediakan oleh sistem agar administrator dapat mengelola *dasbor* yang disediakan oleh sistem. Kebutuhan tersebut dijabarkan oleh *use case* **mengelola dasbor**, yang dipaparkan oleh Tabel 4.5

Tabel 4.5 Skenario *use case* mengelola dasbor

***Use Case Name* : Mengelola Dasbor**	
Kode Fungsi	MSpa-02
Brief Description	Administrator dapat mengelola *dasbor* yang menunjukkan ringkasan analisis sistem berupa kumpulan grafik.
Actors	Administrator

Tabel 4.5 Skenario *use case* mengelola dasbor

***Use Case Name* : Mengelola Dasbor**		
Pre-conditions	Administrator sudah masuk ke dalam sistem dengan melakukan skenario *use case* pada Tabel 4.4	
Flows of Events		
Basic Flow	Aksi Aktor	Respon Sistem
	1. Administrator mengakses menu *dasbor*.	2. Sistem menampilkan data berupa kumpulan grafik.
Sub Flows	-	
Alternative Flows	-	
Post-conditions	Administrator berhasil melakukan pengelolaan data dasbor.	

3. Skenario Mengelola Peta

Kebutuhan fungsional yang disediakan oleh sistem agar pengguna umum dapat mengelola peta yang disediakan oleh sistem. Kebutuhan tersebut dijabarkan oleh *use case* **mengelola peta**, yang dipaparkan oleh Tabel 4.6. *Use case* ini merujuk pada kebutuhan fungsional yang memiliki tingkat prioritas tinggi.

Tabel 4.6 Skenario *use case* mengelola peta

***Use Case Name* : Mengelola Peta**		
Kode Fungsi	MSpa-03	
Brief Description	Pengguna umum dapat mengelola peta yang meliputi pencarian kios berdasarkan pedagang dan memfilter kios berdasarkan komoditas yang dijual.	
Actors	Pengguna Umum	
Pre-conditions	Pengguna umum mengakses sistem.	
Flows of Events		
	Aksi Aktor	Respon Sistem
Basic Flow	1. Pengguna umum mengakses modul untuk mengelola peta dan memilih menu lantai.	2. Sistem menampilkan data berupa peta dan data terkait peta pada lantai terpilih.
	3. Jika pengguna umum memilih operasi untuk mencari data berdasarkan nama pedagang, maka *sub flow* **Mencari Kios Pedagang** akan dijalankan.	
	4. Jika pengguna umum memilih operasi untuk memfilter data berdasarkan komoditas yang dijual,	

Tabel 4.6 Skenario *use case* mengelola peta

Use Case Name **: Mengelola Peta**		
	maka *sub flow* **Filter Kios Pedagang** akan dijalankan.	
Sub Flows	**Mencari Kios Pedagang**	
	1. pengguna umum mengisi nama pedagang pada kolom *search*.	2. Sistem menampilkan hasil pencarian dengan memperbesar kios yang sesuai dengan data pencarian.
	Filter Kios Pedagang	
	1. pengguna umum memilih komoditas yang akan di filter.	2. Sistem menampilkan hasil pencarian dengan memberi warna pada kios yang sesuai dengan data pencarian.
Alternative Flow	-	
Post-conditions	pengguna umum berhasil melakukan pengelolaan peta yang meliputi pencarian dan filter data peta.	

4. Skenario Mengelola *Dasymetric*

Kebutuhan fungsional yang disediakan oleh sistem agar administrator dapat mengelola hasil perhitungan *dasymetric algorithm* yang disediakan oleh sistem. Kebutuhan tersebut dijabarkan oleh *use case* **mengelola *dasymetric***, yang dipaparkan oleh Tabel 4.7. *Use case* ini merujuk pada kebutuhan fungsional yang memiliki tingkat prioritas tinggi.

Tabel 4.7 skenario *use case* mengelola *dasymetric*

Use Case Name **: Mengelola *Dasymetric***		
Kode Fungsi	MSpa-04	
Brief Description	Administrator dapat mengelola *dasymetric* yang meliputi data tabular, grafik dan peta hasil perhitungan *dasymetric algorithm.*	
Actors	Administrator	
Pre-conditions	Administrator sudah masuk ke dalam sistem dengan melakukan skenario *use case* pada Tabel 4.4	
	Flows of Events	
	Aksi Aktor	Respon Sistem
Basic Flow	1. Administrator mengakses menu untuk mengelola *dasymetric*.	2. Sistem menampilkan data berupa data tabular dan grafik hasil perhitungan *dasymetric algorithm*.
	3. Jika administrator memilih operasi untuk melihat peta hasil perhitungan *dasymetric algorithm*,	

Tabel 4.7 skenario *use case* mengelola *dasymetric*

Use Case Name **: Mengelola *Dasymetric***		
	maka *sub flow* ***Dasymetric* Mapping** akan dijalankan.	
Sub Flows	***Dasymetric* Mapping**	
	1. Administrator memilih operasi untuk melihat peta hasil perhitungan pada lantai terpilih.	2. Sistem menampilkan peta lantai terpilih.
	3. Administrator memilih komoditas dan *map tools*.	
Alternative Flows	Administrator memilih ***circle map tool*** pada *sub flow* nomor 3	
	4. Administrator mengisi data berupa *longitude, latitude*, dan radius.	5. Sistem menampilkan hasil pencarian dengan memberi warna pada kios yang sesuai dengan data pencarian dan menampilkan modal berisi detail data kios, data hasil perhitungan *dasymetric algorithm*, dan data grafik yang sesuai dengan data pencarian.
	Administrator memilih ***rectangle map tool*** pada *sub flow* nomor 3	
	4. Administrator menggambar area *rectangle*.	5. Sistem menampilkan hasil pencarian dengan memberi warna pada kios yang sesuai dengan data pencarian dan menampilkan modal berisi detail data kios, data hasil perhitungan *dasymetric algorithm*, dan data grafik yang sesuai dengan data pencarian.
Post-conditions	Administrator berhasil melakukan pengelolaan *dasymetric* yang meliputi data tabular, grafik dan peta hasil perhitungan *dasymetric algorithm.*	

5. Skenario Mengelola Data Pedagang

Kebutuhan fungsional yang disediakan oleh sistem agar administrator dapat mengelola data pedagang yang disediakan oleh sistem. Kebutuhan tersebut dijabarkan oleh *use case* **mengelola data pedagang**, yang dipaparkan oleh Tabel 4.8.

Tabel 4.8 Skenario *use case* mengelola data pedagang

Use Case Name **: Mengelola Data Pedagang**	
Kode Fungsi	MSpa-05
Brief Description	Administrator dapat mengelola data pedagang yang meliputi pembuatan, pembaruan dan penghapusan data pedagang.
Actors	Administrator

Tabel 4.8 Skenario *use case* mengelola data pedagang

***Use Case Name* : Mengelola Data Pedagang**		
Pre-conditions	Administrator sudah masuk ke dalam sistem dengan melakukan skenario *use case* pada Tabel 4.4	
	Flows of Events	
	Aksi Aktor	Respon Sistem
Basic Flow	1. Administrator mengakses menu untuk mengelola data pedagang.	2. Sistem menampilkan data berupa nama pedagang, alamat pedagang dan klasifikasi komoditas yang dijual oleh pedagang terkait.
	3. Jika administrator memilih operasi untuk menambah data pedagang, maka *sub flow* **Menambah Data Pedagang** akan dijalankan.	
	4. Jika administrator memilih operasi untuk menghapus data pedagang, maka *sub flow* **Menghapus Data Pedagang** akan dijalankan.	
	5. Jika administrator memilih operasi untuk melihat detail data pedagang, maka *sub flow* **Melihat Detail Pedagang** akan dijalankan.	
	6. Jika administrator memilih operasi untuk memperbarui data pedagang, maka *sub flow* **Memperbarui Data Pedagang** akan dijalankan.	
	7. Jika administrator memilih operasi untuk menambah data komoditas pedagang terkait, maka *sub flow* **Menambah Data Komoditas** akan dijalankan.	
	8. Jika administrator memilih operasi untuk menghapus data komoditas pedagang terkait, maka *sub flow* **Menghapus Data Komoditas** akan dijalankan.	
	9. Jika administrator memilih operasi untuk menambah data kios pedagang terkait, maka *sub flow* **Menambah Data Kios** akan dijalankan.	

Tabel 4.8 Skenario *use case* mengelola data pedagang

Use Case Name **: Mengelola Data Pedagang**		
	10. Jika administrator memilih operasi untuk menghapus data kios pedagang terkait, maka *sub flow* **Menghapus Data Kios** akan dijalankan.	
	11. Jika administrator memilih operasi untuk menambah data kota distributor pedagang terkait, maka *sub flow* **Menambah Data Kota Distributor** akan dijalankan	
	12. Jika administrator memilih operasi untuk menghapus data kota distributor pedagang terkait, maka *sub flow* **Menghapus Data Kota Distributor** akan dijalankan.	
Sub Flows	**Menambah Data Pedagang**	
	1. Administrator memilih menu untuk menambahkan data pedagang.	2. Sistem menampilkan *form* untuk menambah data pedagang.
	3. Administrator mengisi data pedagang (nama pedagang, alamat pedagang, nomor kios, komoditas, dan kota distributor).	4. Sistem menambah data ke basis data dan menampilkan pesan bahwa data pedagang berhasil ditambahkan.
	Menghapus Data Pedagang	
	1. Administrator memilih operasi untuk menghapus data pedagang.	2. Sistem menampilkan *form* untuk konfirmasi penghapusan data pedagang.
	3. Administrator melakukan konfirmasi penghapusan.	4. Sistem menghapus data dari basis data dan menampilkan pesan bahwa data pedagang berhasil dihapus.
	Melihat Detail Data Pedagang	
	1. Administrator menekan tombol untuk melihat detail data pedagang.	2. Sistem menampilkan data berupa data pedagang (nama pedagang, alamat pedagang), data komoditas (komoditas, klasifikasi komoditas), data kios (nomor kios, blok, lantai) dan data distributor (kota distributor) pedagang terkait.

Tabel 4.8 Skenario *use case* mengelola data pedagang

Use Case Name **: Mengelola Data Pedagang**	
Memperbarui Data Pedagang	
1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
2. Administrator memilih operasi untuk memperbarui data pedagang.	3. Sistem menampilkan *form* untuk memperbarui data pedagang.
5. Administrator mengisi data pedagang (nama pedagang, alamat pedagang).	6. Sistem menampilkan pesan bahwa data pedagang berhasil diperbarui.
Menambah Data Komoditas	
1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
2. Administrator memilih operasi untuk menambahkan data komoditas.	3. Sistem menampilkan *form* untuk menambah data komoditas.
4. Administrator mengisi data komoditas (komoditas, klasifikasi komoditas).	5. Sistem menambah data ke basis data dan menampilkan pesan data komoditas berhasil ditambahkan.
Menghapus Data Komoditas	
1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
2. Administrator memilih operasi untuk menghapus data komoditas.	3. Sistem menampilkan *form* untuk konfirmasi penghapusan data komoditas.
4. Administrator melakukan konfirmasi penghapusan.	5. Sistem menghapus data dari basis data dan menampilkan pesan bahwa data komoditas berhasil dihapus.
Menambah Data Kios	
1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
2. Administrator memilih operasi untuk menambahkan data kios.	3. Sistem menampilkan *form* untuk menambah data kios.
4. Administrator mengisi data kios (nomor kios, blok, lantai).	5. Sistem menambah data ke basis data dan menampilkan pesan bahwa data kios berhasil ditambahkan.

Tabel 4.8 Skenario *use case* mengelola data pedagang

Use Case Name **: Mengelola Data Pedagang**		
	Menghapus Data Kios	
	1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
	2. Administrator memilih operasi untuk menghapus data kios.	3. Sistem menampilkan *form* untuk konfirmasi penghapusan data kios.
	4. Administrator melakukan konfirmasi penghapusan.	5. Sistem menghapus data dari basis data dan menampilkan pesan bahwa data kios berhasil dihapus.
	Menambah Data Kota Distributor	
	1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
	2. Administrator memilih operasi untuk menambahkan data kota distributor.	3. Sistem menampilkan *form* untuk menambah data kota distributor.
	4. Administrator mengisi data kota distributor (kota distributor).	5. Sistem menambah data ke basis data dan menampilkan pesan bahwa data kota distributor berhasil ditambahkan.
	Menghapus Data Kota Distributor	
	1. Administrator menjalankan *sub flow* **Melihat Detail Data Pedagang**	
	2. Administrator memilih operasi untuk menghapus data kota distributor.	3. Sistem menampilkan *form* untuk konfirmasi penghapusan data kota distributor.
	4. Administrator melakukan konfirmasi penghapusan.	5. Sistem menghapus data dari basis data dan menampilkan pesan bahwa data kota distributor berhasil dihapus.
Alternative Flows	Terdapat data yang belum diisi pada *basic flow* nomor 3 dan *sub flow* **Menambah Data Pedagang** nomor 3	
		4. Sistem menampilkan pesan bahwa terdapat data yang belum diisi. Kembali ke *sub flow* **Menambah Data Pedagang** nomor 3.
Post-conditions	Administrator berhasil melakukan pengelolaan data pedagang yang meliputi pembuatan, pembaruan, dan penghapusan data pedagang.	

6. Skenario Mengelola Data Komoditas

Kebutuhan fungsional yang disediakan oleh sistem agar administrator dapat mengelola data komoditas yang disediakan oleh sistem. Kebutuhan tersebut dijabarkan oleh *use case* **mengelola data komoditas**, yang dipaparkan oleh Tabel 4.9.

Tabel 4.9 Skenario *use case* mengelola data komoditas

***Use Case Name* : Mengelola Data Komoditas**		
Kode Fungsi	MSpa-06	
Brief Description	Administrator dapat mengelola data komoditas yang meliputi pembuatan, pembaruan dan penghapusan data komoditas.	
Actors	Administrator	
Pre-conditions	Administrator sudah masuk ke dalam sistem dengan melakukan skenario *use case* pada Tabel 4.4	
	Flows of Events	
Basic Flow	Aksi Aktor	Respon Sistem
	1. Administrator mengakses menu untuk mengelola data komoditas.	2. Sistem menampilkan data berupa data tabular komoditas, data tabular klasifikasi komoditas dan data tabular kategori komoditas.
	3. Jika administrator memilih operasi untuk menambah data komoditas, maka *sub flow* **Menambah Data Komoditas** akan dijalankan.	
	4. Jika administrator memilih operasi untuk memperbarui data komoditas, maka *sub flow* **Memperbarui Data Komoditas** akan dijalankan.	
	5. Jika administrator memilih operasi untuk memperbarui data klasifikasi komoditas, maka *sub flow* **Memperbarui Data Klasifikasi Komoditas** akan dijalankan.	
	6. Jika administrator memilih operasi untuk memperbarui data kategori komoditas, maka *sub flow* **Memperbarui Data Kategori Komoditas** akan dijalankan.	
Sub Flows	**Menambah Data Komoditas**	
	1. Administrator memilih menu untuk menambahkan data komoditas.	2. Sistem menampilkan halaman tambah data komoditas.

Tabel 4.9 Skenario *use case* mengelola data komoditas

Use Case Name **: Mengelola Data Komoditas**		
	3. Administrator mengisi data komoditas (nama komoditas, klasifikasi komoditas).	4. Sistem menambah data ke basis data dan menampilkan pesan bahwa data komoditas berhasil ditambahkan.
	Memperbarui Data Komoditas	
	1. Administrator memilih operasi untuk memperbarui data komoditas.	2. Sistem menampilkan *form* untuk memperbarui data komoditas.
	3. Administrator mengisi data (nama komoditas, klasifikasi komoditas).	4. Sistem menampilkan pesan bahwa data komoditas berhasil diperbarui.
	Memperbarui Data Klasifikasi Komoditas	
	1. Administrator memilih operasi untuk memperbarui data klasifikasi komoditas.	2. Sistem menampilkan *form* untuk memperbarui data klasifikasi komoditas.
	3. Administrator mengisi data (klasifikasi komoditas, kategori komoditas).	4. Sistem menampilkan pesan bahwa data klasifikasi komoditas berhasil diperbarui.
	Memperbarui Data Kategori Komoditas	
	1. Administrator memilih operasi untuk memperbarui data kategori komoditas.	2. Sistem menampilkan *form* untuk memperbarui data kategori komoditas.
	3. Administrator mengisi data kategori komoditas.	4. Sistem menampilkan pesan bahwa data kategori komoditas berhasil diperbarui.
Alternative Flows	Terdapat data yang belum diisi pada *basic flow* nomor 3 dan *sub flow* **Menambah Data Komoditas** nomor 3	
		4. Sistem menampilkan pesan bahwa terdapat data yang belum diisi. Kembali ke *sub flow* **Menambah Data Komoditas** nomor 3.
Post-conditions	Administrator berhasil melakukan pengelolaan data komoditas yang meliputi pembuatan, pembaruan, dan penghapusan data komoditas.	

7. Skenario Logout

Kebutuhan fungsional yang disediakan oleh sistem agar administrator dapat *logout* dari sistem. Kebutuhan tersebut dijabarkan oleh *use case* **logout**, yang dipaparkan oleh Tabel 4.10.

Tabel 4.10 Skenario *use case* logout

Use Case Name **: Logout**	
Kode Fungsi	MSpa-07

Tabel 4.10 Skenario *use case* logout

***Use Case Name* : Logout**		
Brief Description	Administrator dapat melakukan *logout* agar dapat keluar dari sistem.	
Actors	Administrator	
Pre-conditions	Administrator sudah masuk ke dalam sistem dengan melakukan skenario *use case* pada Tabel 4.4	
	Flows of Events	
Basic Flow	Aksi Aktor	Respon Sistem
	1. Administrator memilih operasi untuk melakukan *logout*.	2. Sistem menampilkan halaman *login*.
Sub Flows	-	
Alternative Flows	-	
Post-conditions	Administrator berhasil melakukan *logout* dan sistem menampilkan halaman *login*.	

4.2 Perancangan

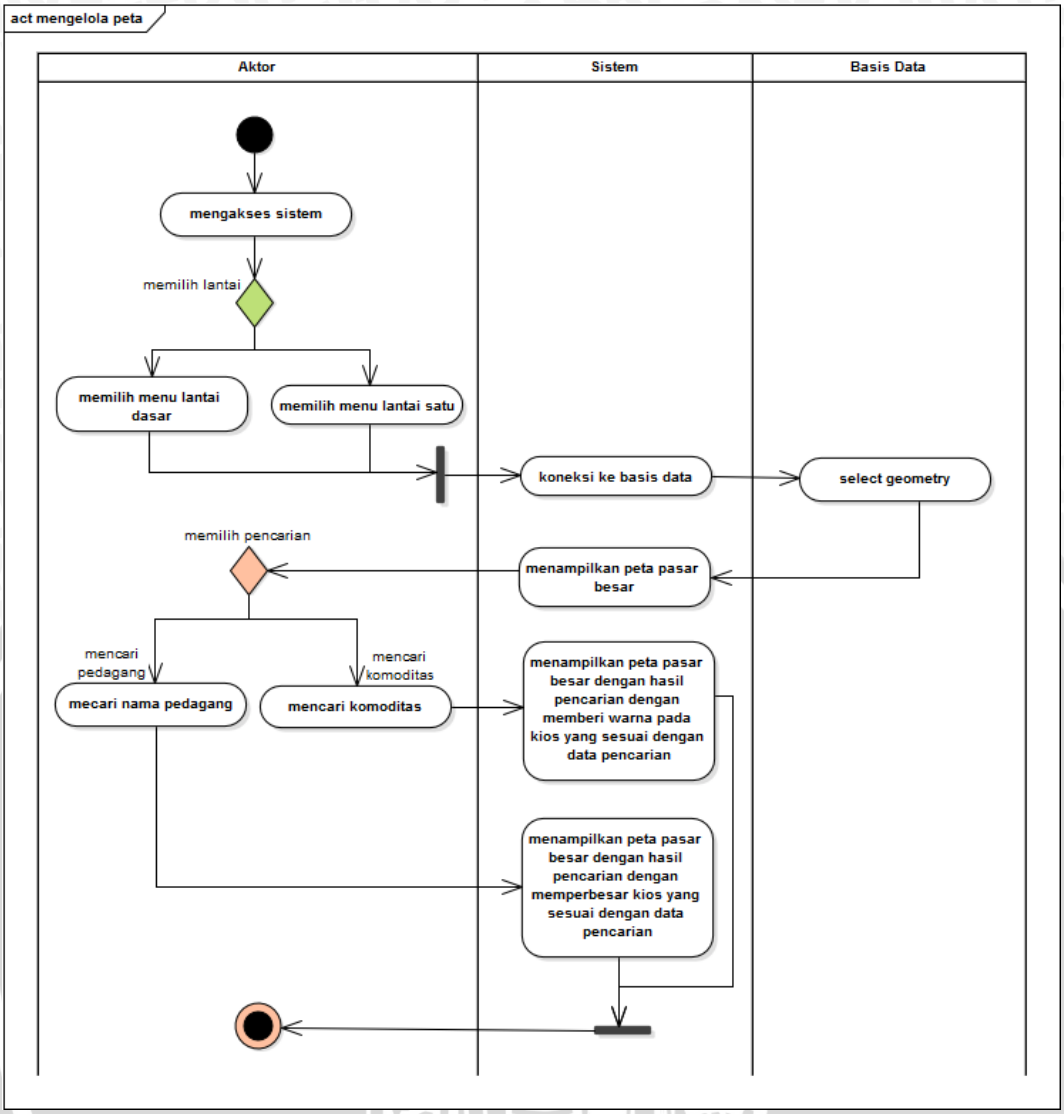
Terdapat lima tahap dalam perancangan sistem pada penelitian ini yaitu 1) Perancangan *Activity Diagram,* 2) Perancangan *Sequence Diagram*, 3) Perancangan *Class Diagram*, 4) Perancangan Basis Data, dan 5) Perancangan Antarmuka. Perancangan sistem yang dilakukan menggunakan pendekatan berorientasi objek dengan diagram UML. Perancangan yang dipaparkan merupakan perancangan terhadap kebutuhan fungsional yang memiliki prioritas tinggi.

4.2.1 *Activity Diagram*

Activity diagram menggambarkan aliran aktivitas yang terjadi dalam sistem yang sedang dirancang berupa awal aliran, pilihan yang mungkin terjadi, dan bagaimana aliran tersebut berakhir. Diagram ini lebih menggambarkan proses dan jalur aktivitas dari level atas secara umum.

1. *Activity Diagram* Mengelola Peta

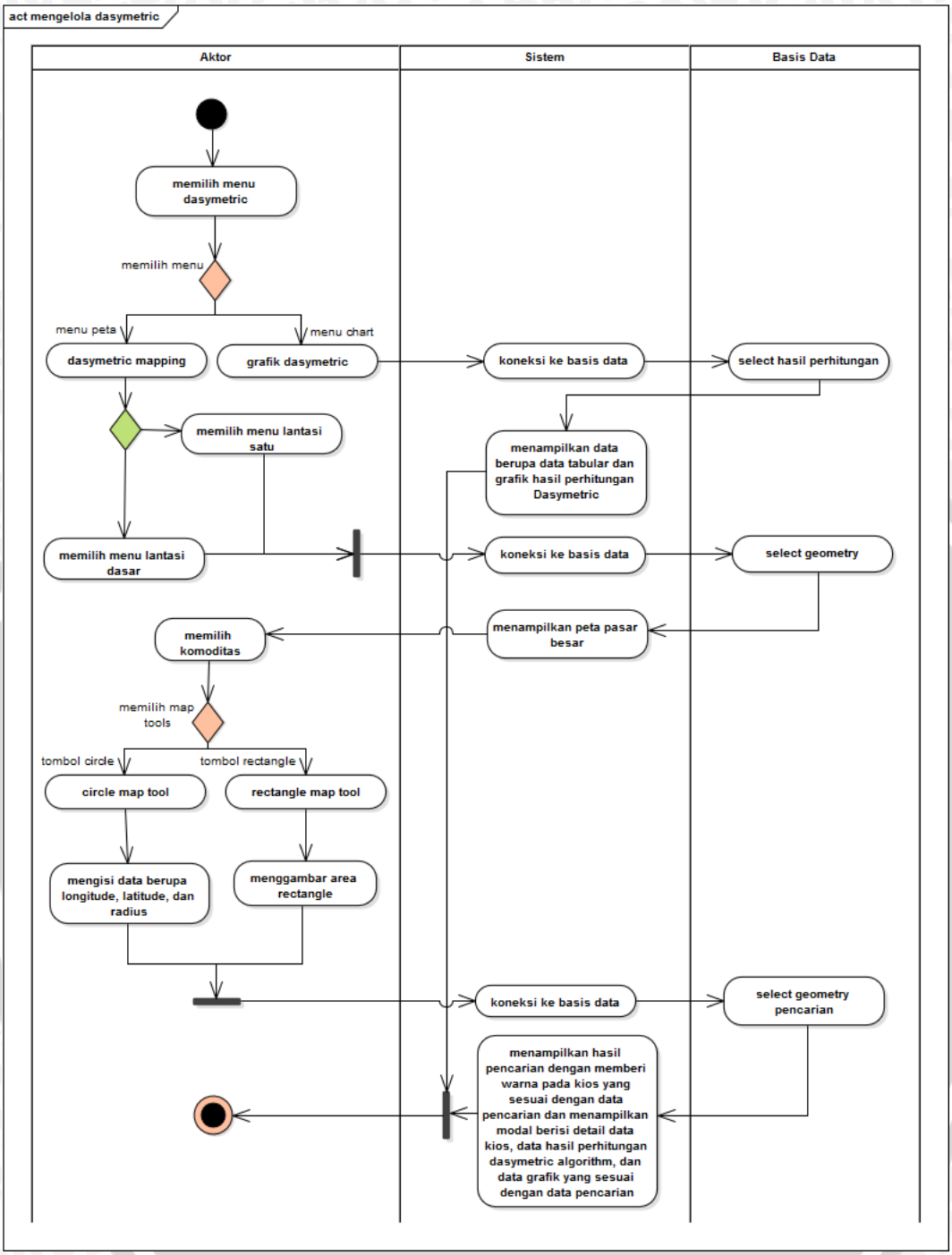
Activity diagram mengelola peta yang digambarkan oleh Gambar 4.6 menjelaskan aktivitas sistem untuk melakukan pencarian dan filter data peta yang dilakukan oleh pengguna umum. Aktivitas dimulai dengan dengan pengguna umum mengakses sistem dan diakhiri dengan sistem menampilkan peta Pasar Besar dengan hasil pencarian terpilih.



Gambar 4.6 *Activity diagram* mengelola peta

2. *Activity Diagram* Mengelola *Dasymetric*

Activity diagram mengelola *dasymetric* yang digambarkan oleh Gambar 4.7 menjelaskan aktivitas sistem untuk melakukan pengelolaan terhadap data tabular, grafik dan peta hasil perhitungan *dasymetric algorithm* yang dilakukan oleh administrator. Aktivitas dimulai dengan administrator memilih menu *dasymetric* dan diakhiri dengan menampilkan data hasil perhitungan *dasymetric algorithm*.



Gambar 4.7 *Activity diagram* mengelola *dasymetric*

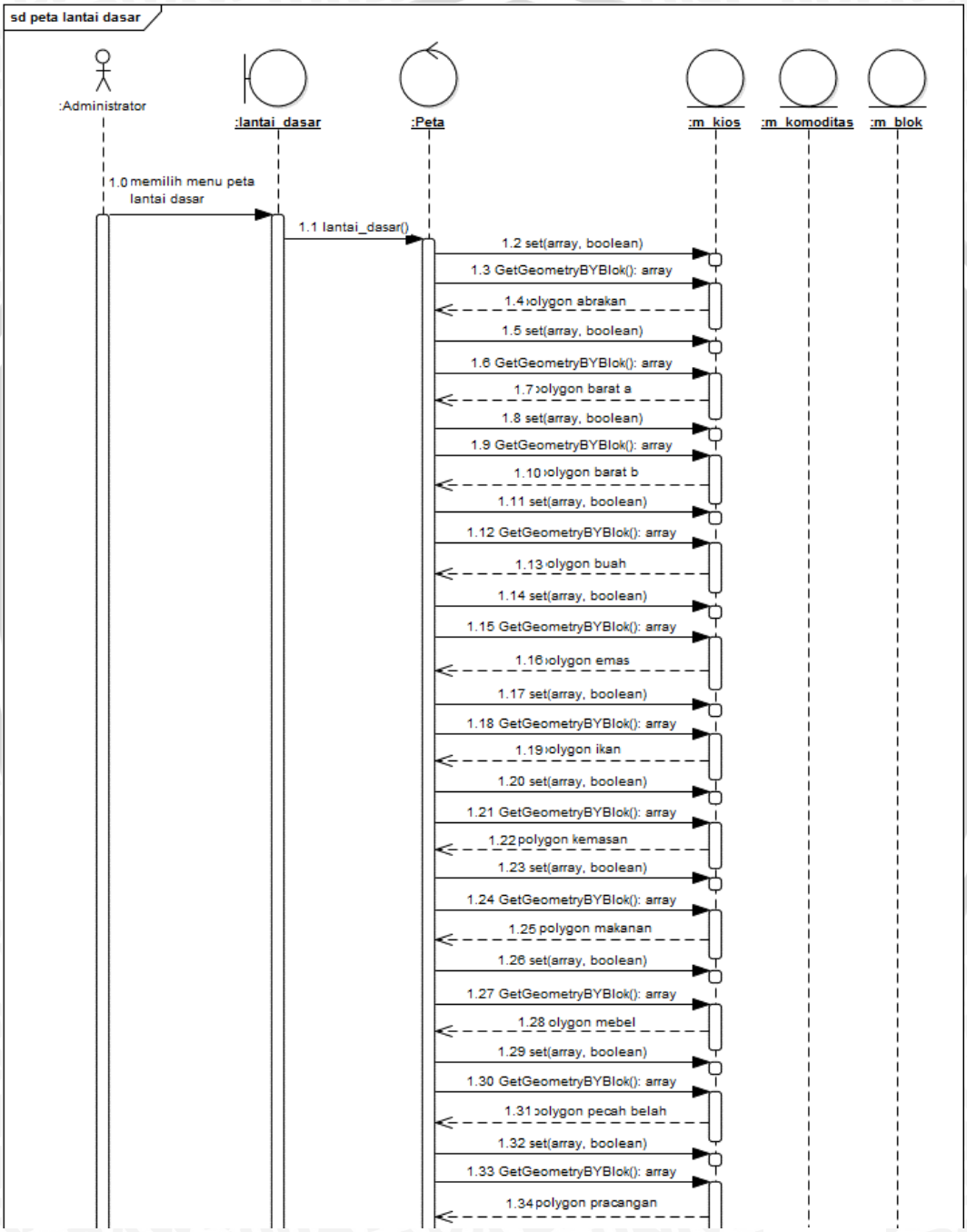
4.2.2 *Sequence Diagram*

Sequence diagram digunakan untuk menjabarkan interaksi antar komponen kode program yang akan dibuat. Diagram ini mendeskripsikan *lifeline* dan pesan yang dikirim ataupun diterima antar objek untuk menggambarkan perilaku objek pada *use case*.

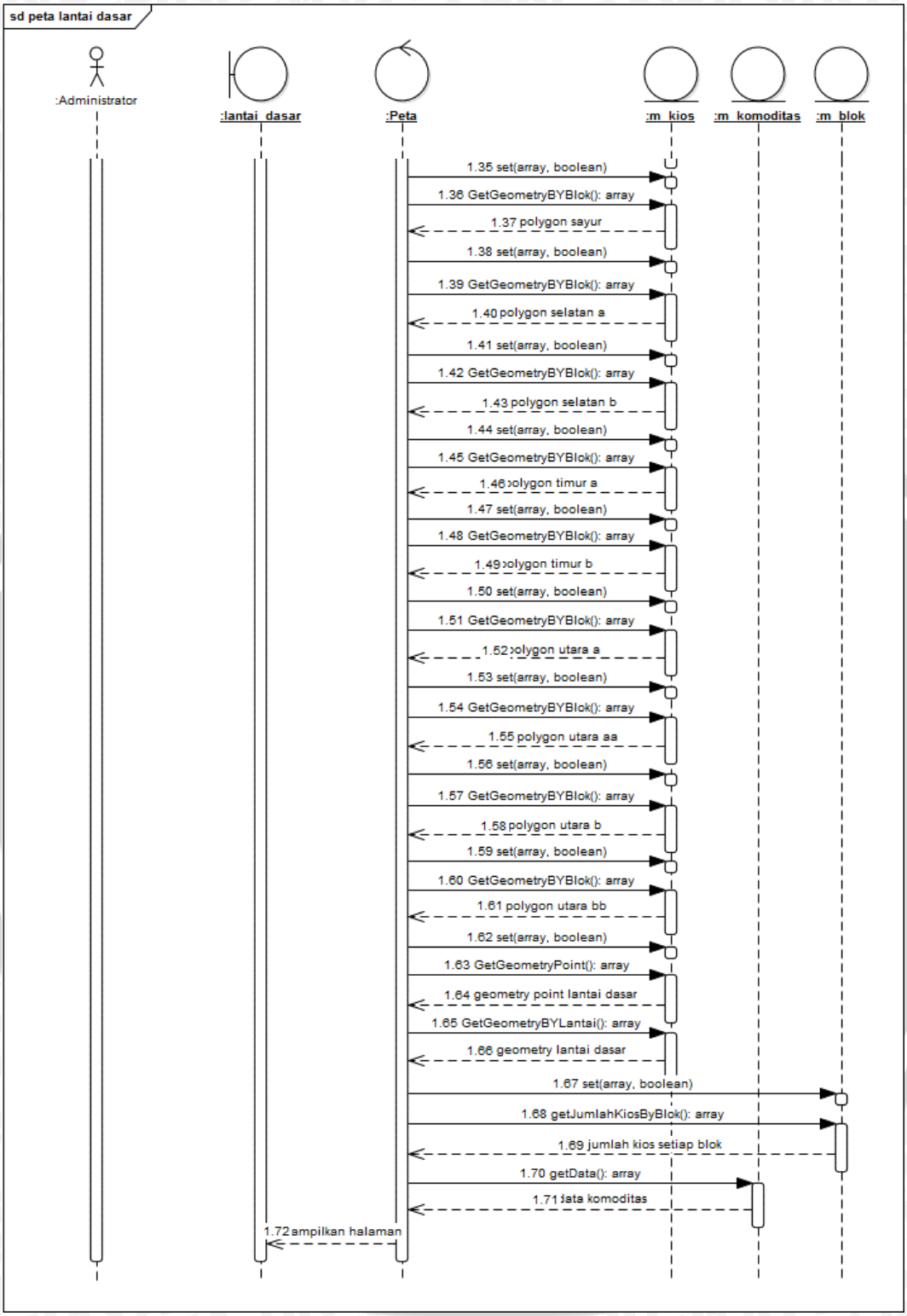
4.2.2.1 Sequence Diagram Mengelola Peta

Gambar 4.8 dan Gambar 4.9 menunjukkan *sequence diagram* dari proses pengelolaan peta yang meliputi pencarian kios berdasarkan pedagang dan memfilter kios berdasarkan komoditas yang dijual. Diagram ini merepresentasikan kode fungsi MSpa-03.

1. Lantai Dasar

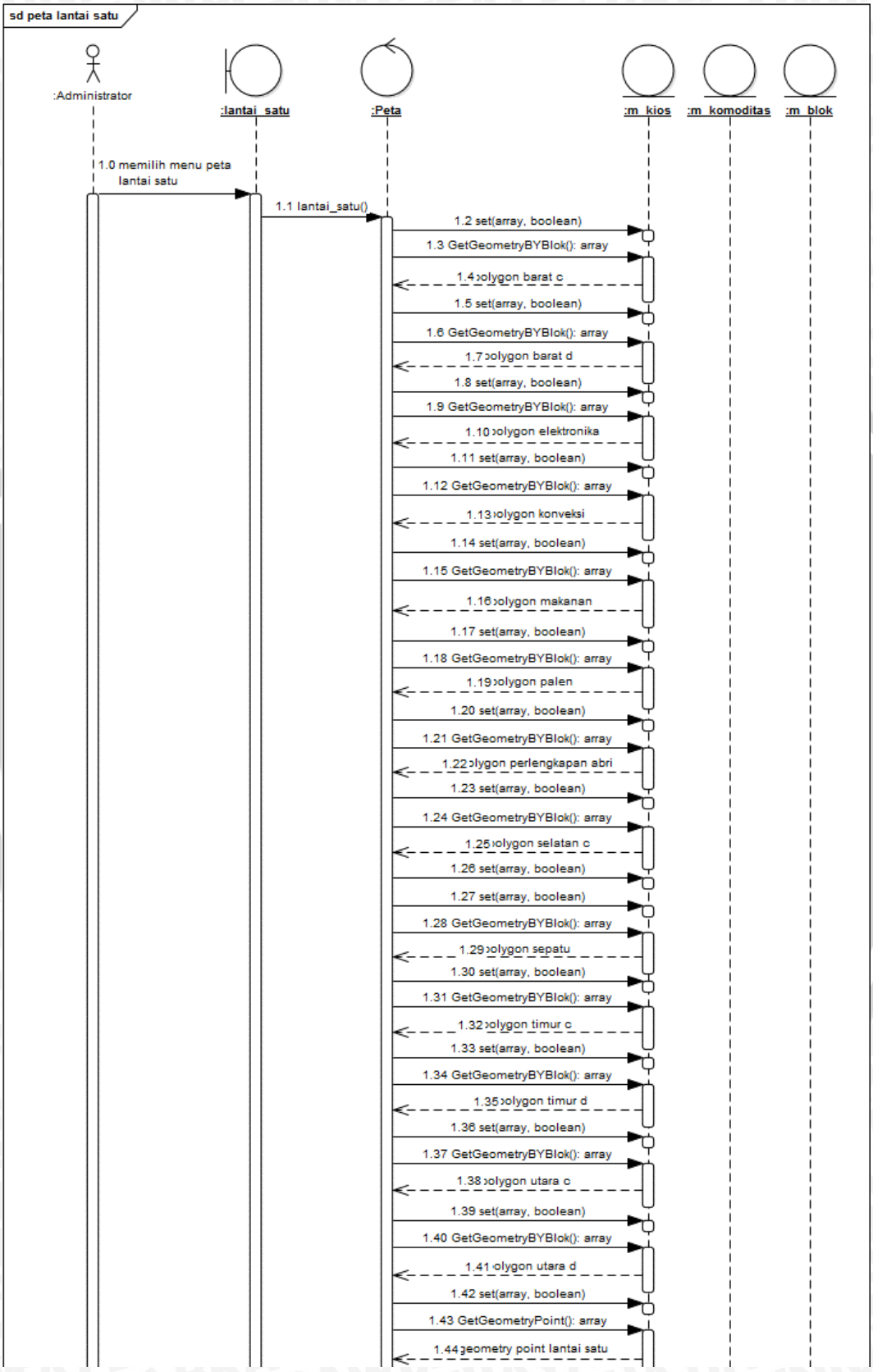


Gambar 4.8 *Sequence diagram* mengelola peta lantai dasar [bagian 1]

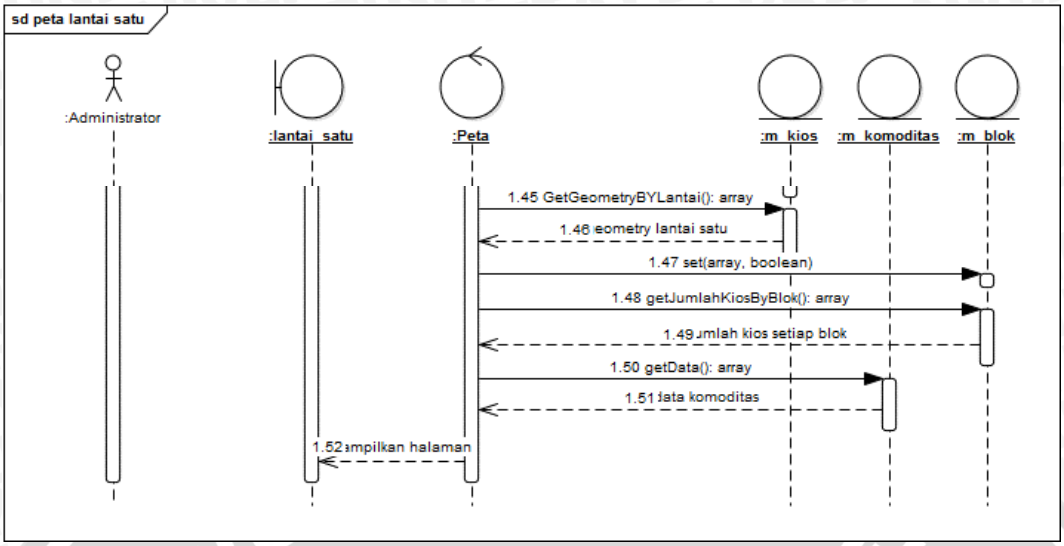


Gambar 4.8 *Sequence diagram* mengelola peta lantai dasar [bagian 2]

2. Lantai Satu



Gambar 4.9 *Sequence diagram* mengelola peta lantai satu [bagian 1]



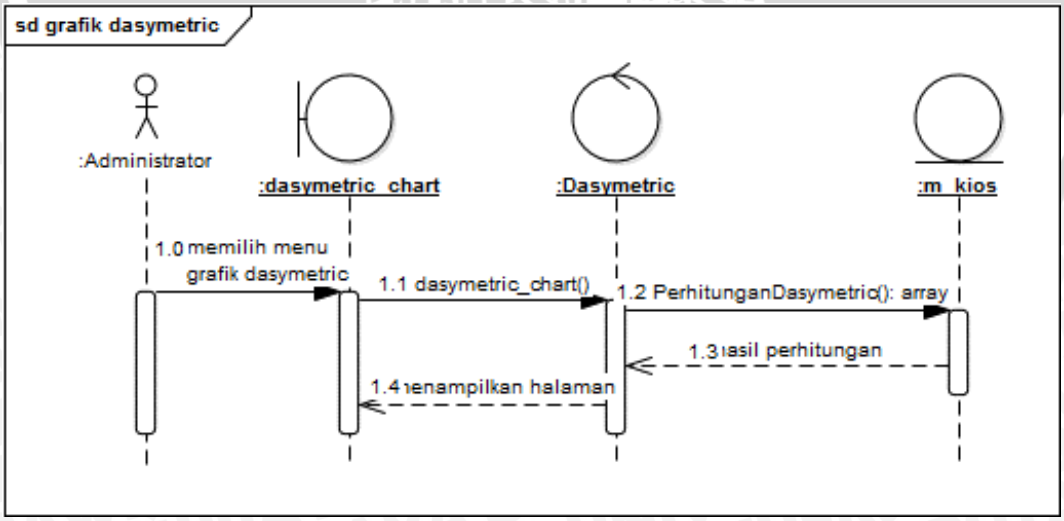
Gambar 4.9 *Sequence diagram* mengelola peta lantai satu [bagian 2]

4.2.2.2 Sequence Diagram Mengelola Dasymetric

Gambar 4.10, Gambar 4.11, dan Gambar 4.12 menunjukkan sequence diagram dari proses pengelolaan *dasymetric* yang meliputi data tabular, grafik, dan peta hasil perhitungan *dasymetric algorithm*. Diagram ini merepresentasikan kode fungsi MSpa-04.

1. Grafik Dasymetric

Gambar 4.10 merupakan *sequence diagram* dari grafik *dasymetric* yang berfungsi untuk melihat data hasil perhitungan *dasymetric algorithm* berupa data tabular dan data grafik.

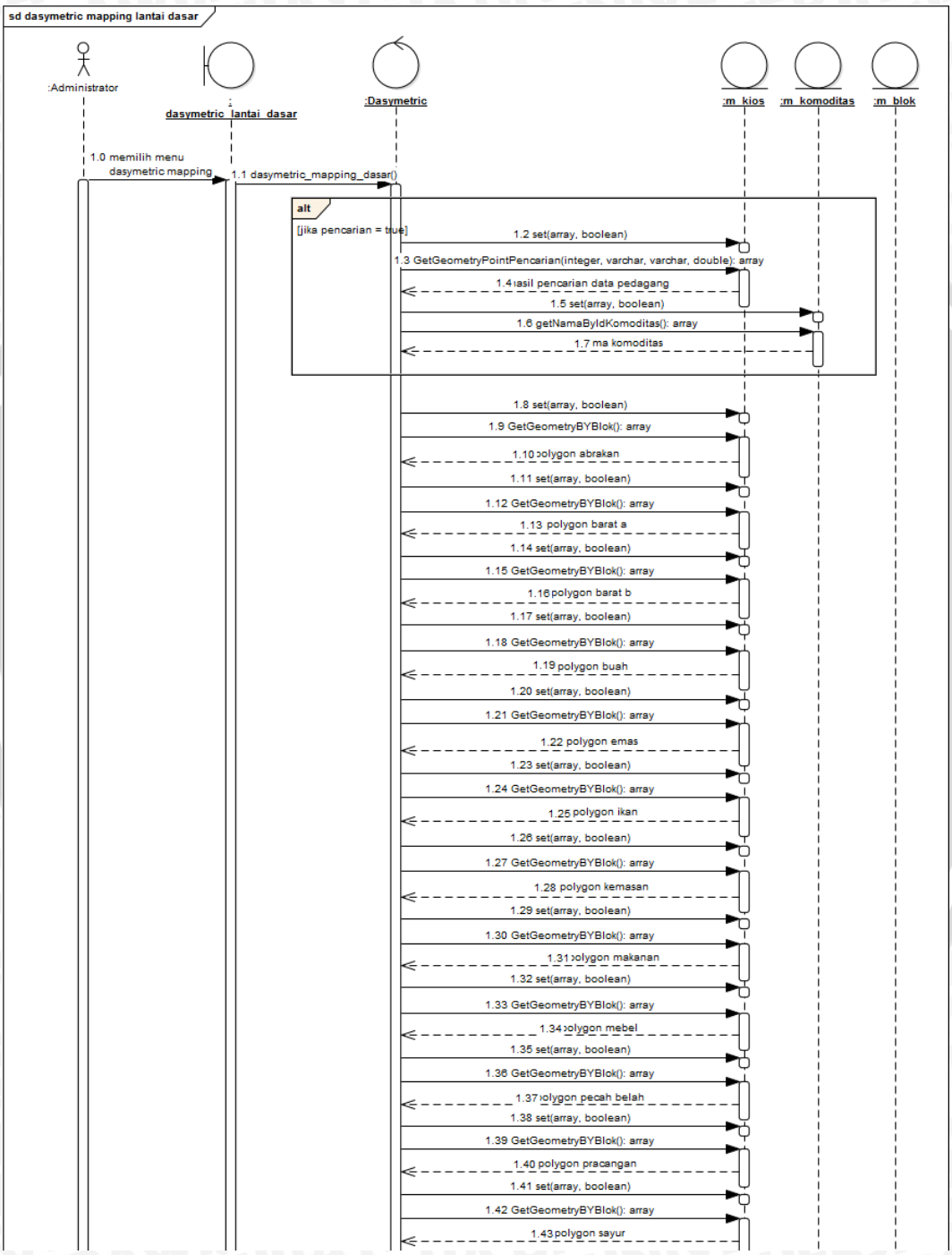


Gambar 4.10 *Sequence diagram* grafik *dasymetric*

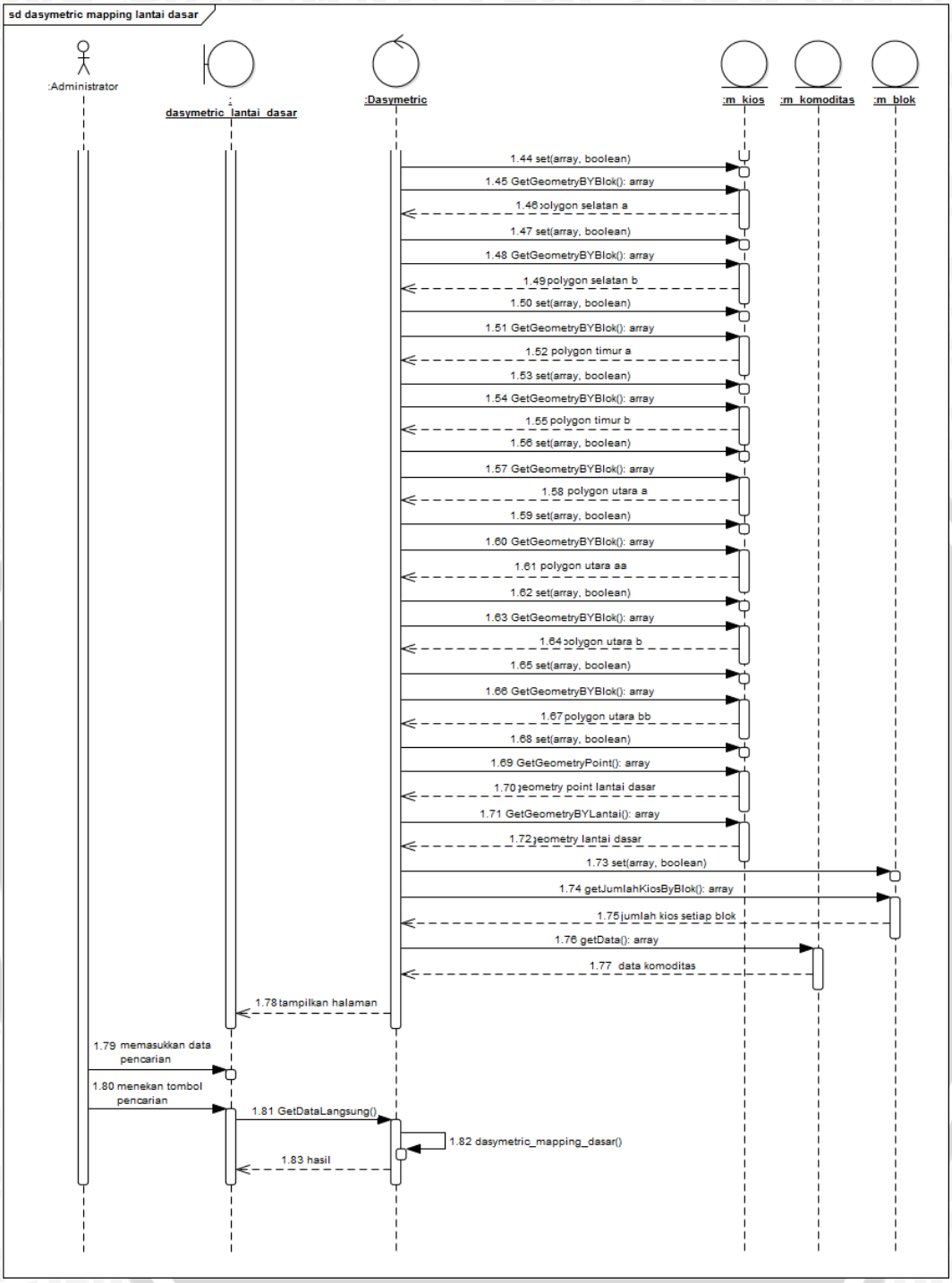
2. *Dasymetric Mapping*

Gambar 4.11 dan Gambar 4.12 merupakan *sequence diagram* dari *dasymetric mapping* yang berfungsi untuk melihat data hasil perhitungan *dasymetric algorithm* berdasarkan area pilihan pengguna menggunakan bantuan *map tools*.

a. Lantai Dasar

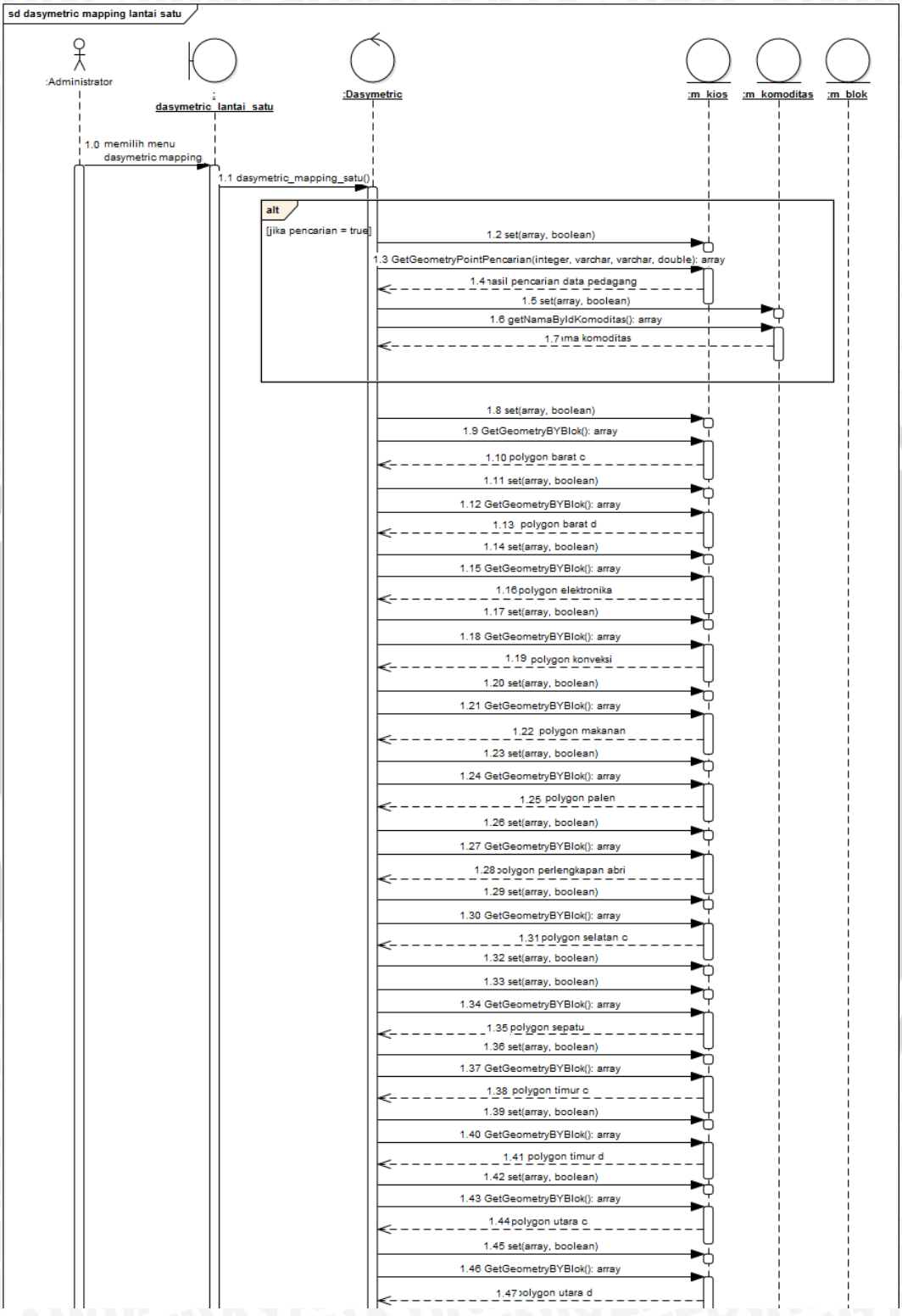


Gambar 4.11 *Sequence diagram dasymetric mapping* lantai dasar [bagian 1]

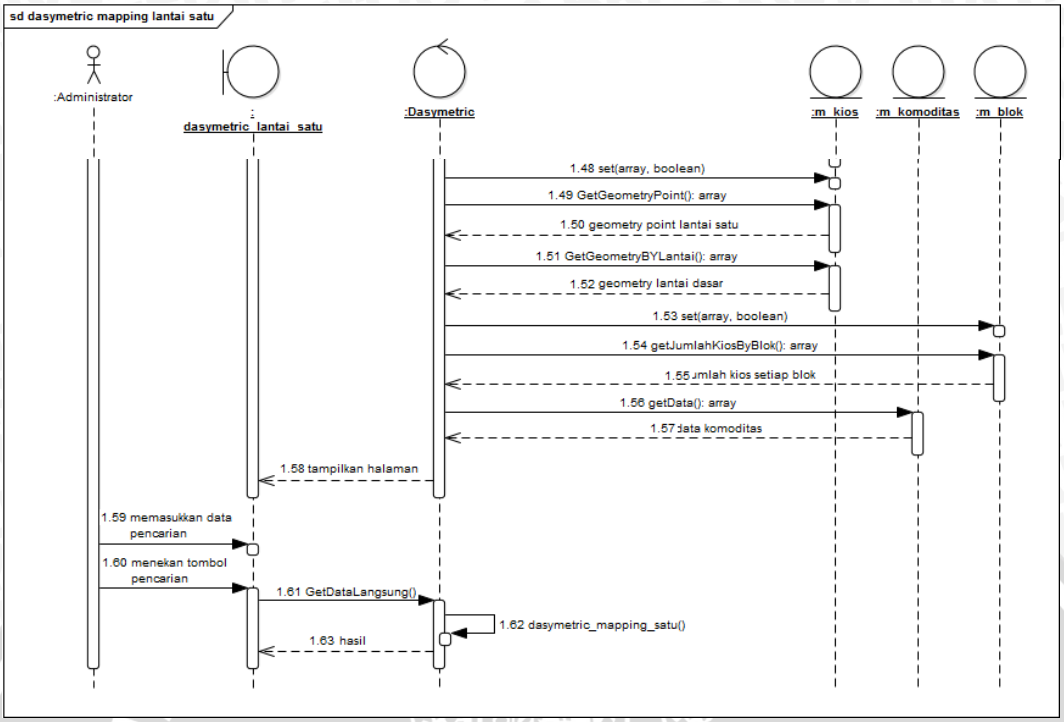


Gambar 4.11 *Sequence diagram dasymetric mapping* lantai dasar [bagian 2]

b. Lantai Satu



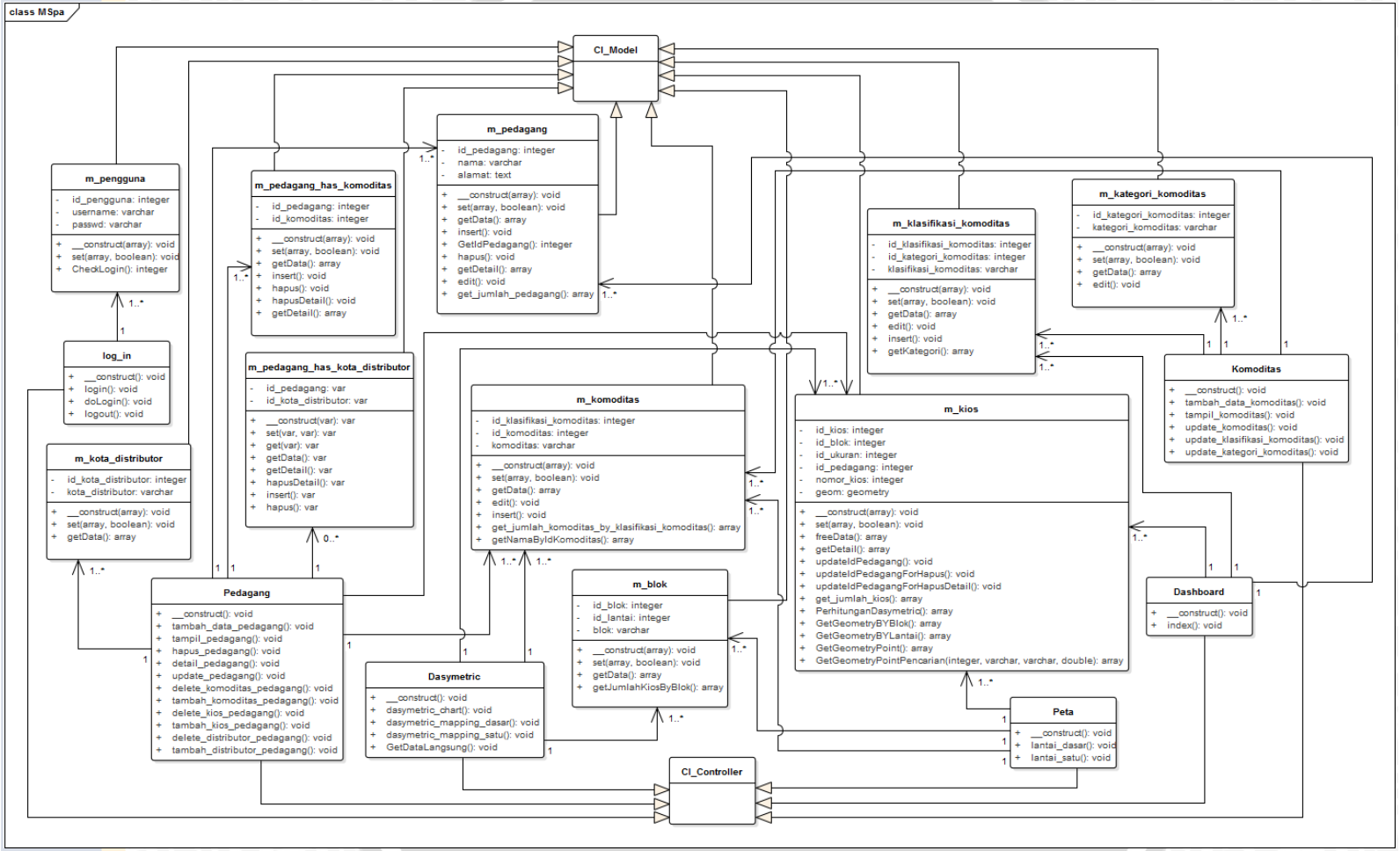
Gambar 4.12 *Sequence diagram dasymetric mapping* lantai satu [bagian 1]



Gambar 4.12 *Sequence diagram dasymetric mapping* lantai satu [bagian 2]

4.2.3 *Class Diagram*

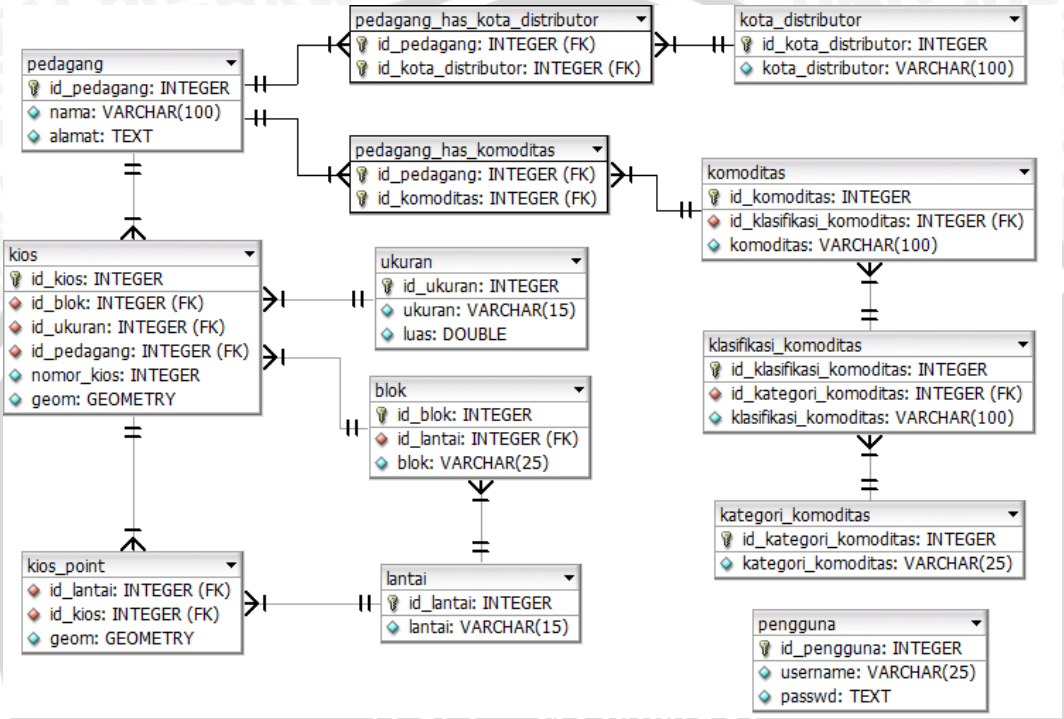
Class diagram digunakan untuk menjalankan implementasi kode program untuk membangun sistem yang telah dirancang. Melalui diagram ini dapat diketahui penyusun sistem dan interaksi antar komponen. Gambar 4.13 menunjukkan perancangan *class diagram* yang memaparkan struktur sistem dalam pendefinisian kelas-kelas yang akan dibuat untuk mengembangkan sistem.



Gambar 4.13 *Class diagram*

4.2.4 Perancangan Basis Data

Basis data digunakan untuk menyimpan seluruh data hasil proses dari sistem. Perancangan basis data bertujuan untuk merancang basis data yang akan dibuat agar keluaran dari program sesuai dengan yang diharapkan. Perancangan basis data disesuaikan dengan analisis kebutuhan dan perancangan *class diagram* (kelas model) pada tahap sebelumnya. Gambar 4.14 menunjukkan relasi antar basis data.



Gambar 4.14 Relasi antar basis data

4.2.4.1 Kamus Data

Kamus data berisi metadata untuk menjelaskan secara detail catatan atribut dari keseluruhan tabel di dalam basis data. Berikut adalah rancangan dari masing-masing tabel:

1. Tabel Blok

Nama Tabel : blok

Jumlah kolom : 3

Fungsi : menyimpan data blok

Tabel 4.11 Tabel blok

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_blok	serial	-	id blok
2.	id_lantai	integer	-	id lantai
3.	blok	character varying	25	nama blok

2. Tabel Kategori Komoditas

Nama Tabel : kategori_komoditas

Jumlah kolom : 2

Fungsi : menyimpan data kategori komoditas

Tabel 4.12 Tabel kategori komoditas

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_kategori_komoditas	serial	-	id kategori komoditas
2.	kategori_komoditas	character varying	25	nama kategori

3. Tabel Kios

Nama Tabel : kios

Jumlah kolom : 7

Fungsi : menyimpan data kios

Tabel 4.13 Tabel kios

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_kios	integer	-	id kios
2.	id_blok	integer	-	id blok
3.	id_lantai	integer	-	id lantai
4.	id_ukuran	integer	-	id ukuran
5.	id_pedagang	integer	-	id_pedagang
6.	nomor_kios	integer	-	nomor kios
7.	geom	geometry(Polygon)	-	data *longitude* dan *latitude* wilayah kios Pasar Besar

4. Tabel Kios Point

Nama Tabel : kios_point

Jumlah kolom : 3

Fungsi : menyimpan data kios

Tabel 4.14 Tabel kios point

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_kios	Integer	-	id kios
2.	id_lantai	Integer	-	id lantai
3.	geom	geometry(Point)	-	data *longitude* dan *latitude* titik tengah kios Pasar Besar

5. Tabel Klasifikasi Komoditas

Nama Tabel : klasifikasi_komoditas

Jumlah kolom : 3

Fungsi : menyimpan data klasifikasi komoditas

Tabel 4.15 Tabel klasifikasi komoditas

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_klasifikasi_komoditas	serial	-	id klasifikasi komoditas
2.	id_kategori_komoditas	integer	-	id kategori komoditas
3.	klasifikasi_komoditas	character varying	100	nama klasifikasi

6. Tabel Komoditas

Nama Tabel : komoditas

Jumlah kolom : 3

Fungsi : menyimpan data klasifikasi komoditas

Tabel 4.16 Tabel komoditas

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_komoditas	serial	-	id komoditas
2.	id_klasifikasi_komoditas	integer	-	id klasifikasi komoditas
3.	komoditas	character varying	100	nama komoditas

7. Tabel Kota Distributor

Nama Tabel : kota_distributor

Jumlah kolom : 2

Fungsi : menyimpan data kota distributor

Tabel 4.17 Tabel kota distributor

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_kota_distributor	serial	-	id kota distributor
2.	kota_distributor	character varying	100	nama kota distributor

8. Tabel Lantai

Nama Tabel : lantai

Jumlah kolom : 2

Fungsi : menyimpan data lantai

Tabel 4.18 Tabel lantai

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_lantai	serial	-	id lantai
2.	lantai	character varying	15	nama lantai

9. Tabel Pedagang

Nama Tabel : pedagang

Jumlah kolom : 3

Fungsi : menyimpan data pedagang

Tabel 4.19 Tabel pedagang

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_pedagang	serial	-	id pedagang
2.	nama	character varying	100	nama pedagang
3.	alamat	text	-	alamat rumah pedagang

10. Tabel Pedagang Has Komoditas

Nama Tabel : pedagang_has_komoditas

Jumlah kolom : 2

Fungsi : menyimpan data pedagang

Tabel 4.20 Tabel pedagang has komoditas

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_pedagang	integer	-	id pedagang
2.	id_komoditas	integer	-	id komoditas

11. Tabel Pedagang Has Kota Distributor

Nama Tabel : pedagang_has_kota_distributor

Jumlah kolom : 3

Fungsi : menyimpan data pedagang

Tabel 4.21 Tabel pedagang has kota distributor

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_pedagang	integer	-	id pedagang
2.	id_kota_distributor	integer	-	id kota distributor

12. Tabel Pengguna

Nama Tabel : pengguna

Jumlah kolom : 3

Fungsi : menyimpan data administrator

Tabel 4.22 Tabel pengguna

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_pengguna	serial	-	id pengguna
2.	username	character varying	25	*username* administrator
3.	passwd	text	-	*password* administrator

13. Tabel Ukuran

Nama Tabel : ukuran

Jumlah kolom : 3

Fungsi : menyimpan data ukuran kios

Tabel 4.23 Tabel ukuran

No.	Nama Field	Tipe	Lebar	Keterangan
1.	id_ukuran	serial	-	id ukuran
2.	ukuran	character varying	15	keterangan ukuran kios
3.	luas	double precision	-	luas kios

4.2.5 Perancangan Antarmuka Pengguna

Bagian ini menjelaskan tentang perancangan antarmuka sistem. Antarmuka merupakan sarana aktor untuk berinteraksi dengan sistem. Perancangan antarmuka yang dijelaskan pada bagian ini berfokus pada penjabaran kebutuhan fungsional yang memiliki prioritas tinggi pada Tabel 4.2.

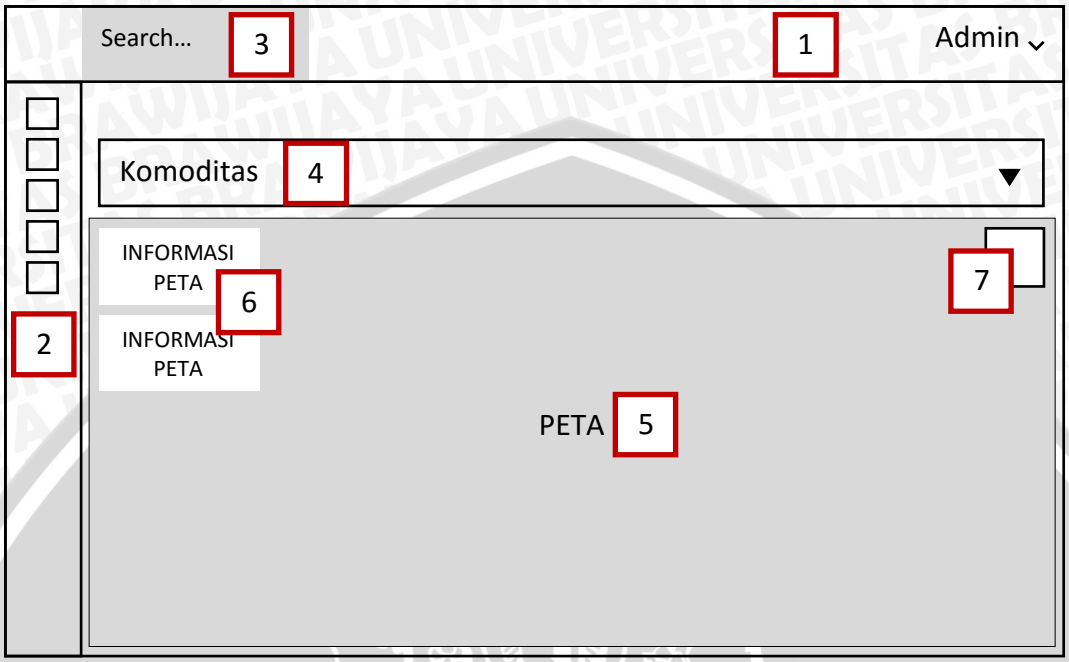
4.2.5.1 Perancangan Antarmuka Mengelola Peta

Perancangan antarmuka mengelola peta ditunjukkan oleh Gambar 4.15. Halaman ini menyediakan layanan untuk mengelola peta sehingga pengguna dapat melakukan pencarian kios berdasarkan pedagang dengan parameter masukan berupa nama pedagang, dan memfilter kios berdasarkan komoditas yang dijual dengan parameter masukan komoditas.

Berikut keterangan fungsi tampilan Gambar 4.15:

1. *header* sistem ApsOrcim
2. menu sistem ApsOrcim
3. kolom pencarian berdasarkan nama pedagang Pasar Besar
4. pilihan komoditas

5. peta Pasar Besar
6. informasi peta
7. tombol legenda



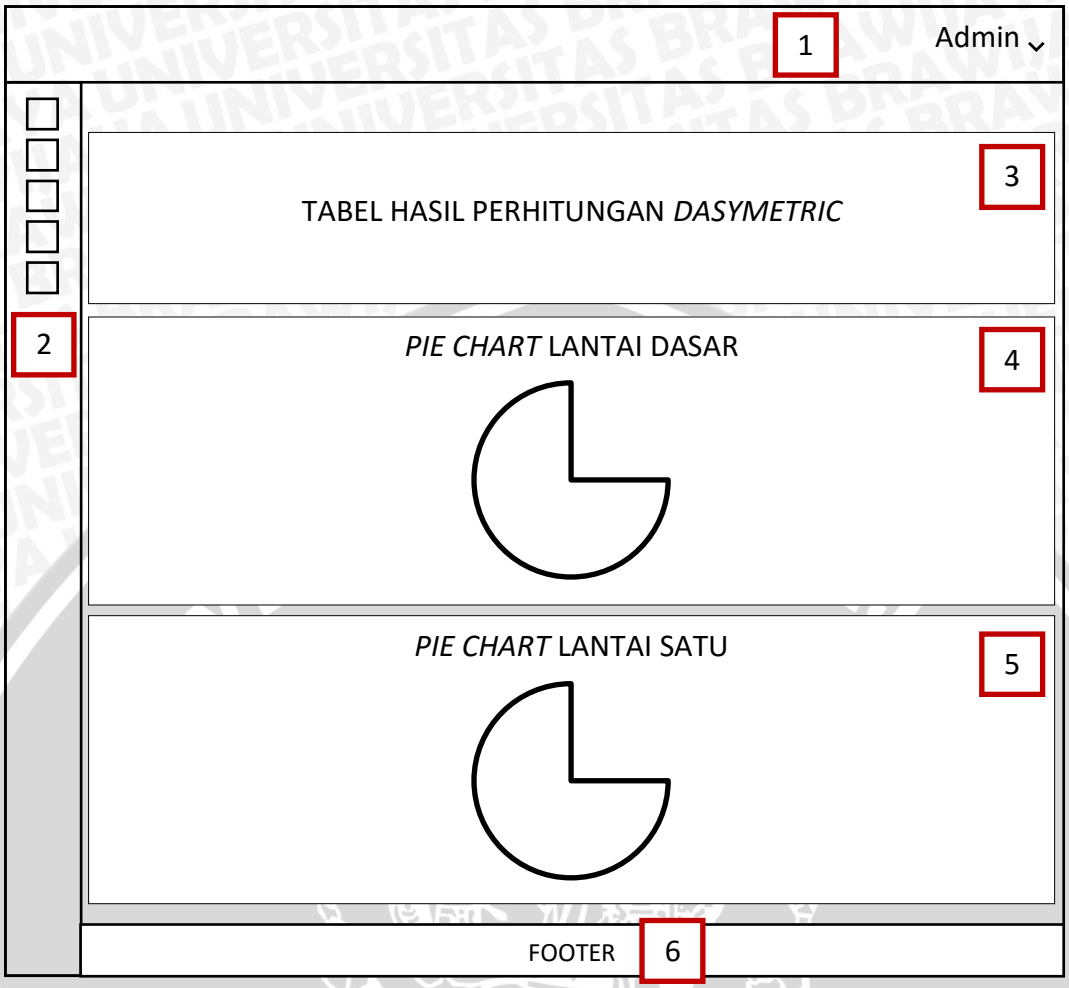
Gambar 4.15 Perancangan antarmuka mengelola peta

4.2.5.2 Perancangan Antarmuka Mengelola Dasymetric

Perancangan antarmuka mengelola *dasymetric* ditunjukkan oleh Gambar 4.16 dan Gambar 4.17. Halaman ini menyediakan layanan untuk mengelola data *dasymetric* yang meliputi data tabular, grafik dan peta dengan parameter masukan longitude, latitude, radius dan komoditas sehingga dapat ditampilkan visualisasi hasil perhitungan *dasymetric algorithm*.

Gambar 4.16 merupakan perancangan antarmuka yang menyediakan data hasil perhitungan *dasymetric algorithm* berupa data tabular dan grafik *pie*. Berikut keterangan fungsi tampilan Gambar 4.16:

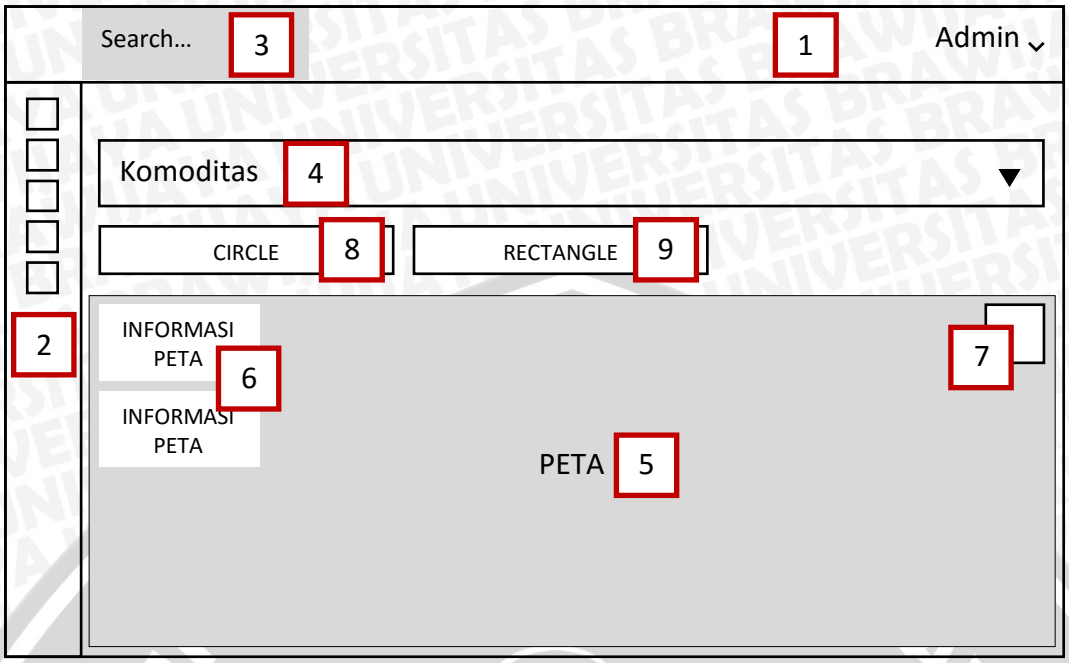
1. *header* sistem ApsOrcim
2. menu sistem ApsOrcim
3. tabel hasil perhitungan *dasymetric algorithm*
4. grafik *pie* hasil perhitungan *dasymetric algorithm* lantai dasar
5. grafik *pie* hasil perhitungan *dasymetric algorithm* lantai satu
6. *footer* sistem ApsOrcim



Gambar 4.16 Perancangan antarmuka grafik *dasymetric*

Gambar 4.17 merupakan perancangan antarmuka yang menyediakan informasi dalam bentuk visualisasi berupa peta. Terdapat *map tools* yang berfungsi untuk melihat informasi pencarian dan hasil perhitungan *dasymetric algorithm* dengan cara menggambar *circle tool* atau *rectangle tool* pada peta. Berikut keterangan fungsi tampilan Gambar 4.17:

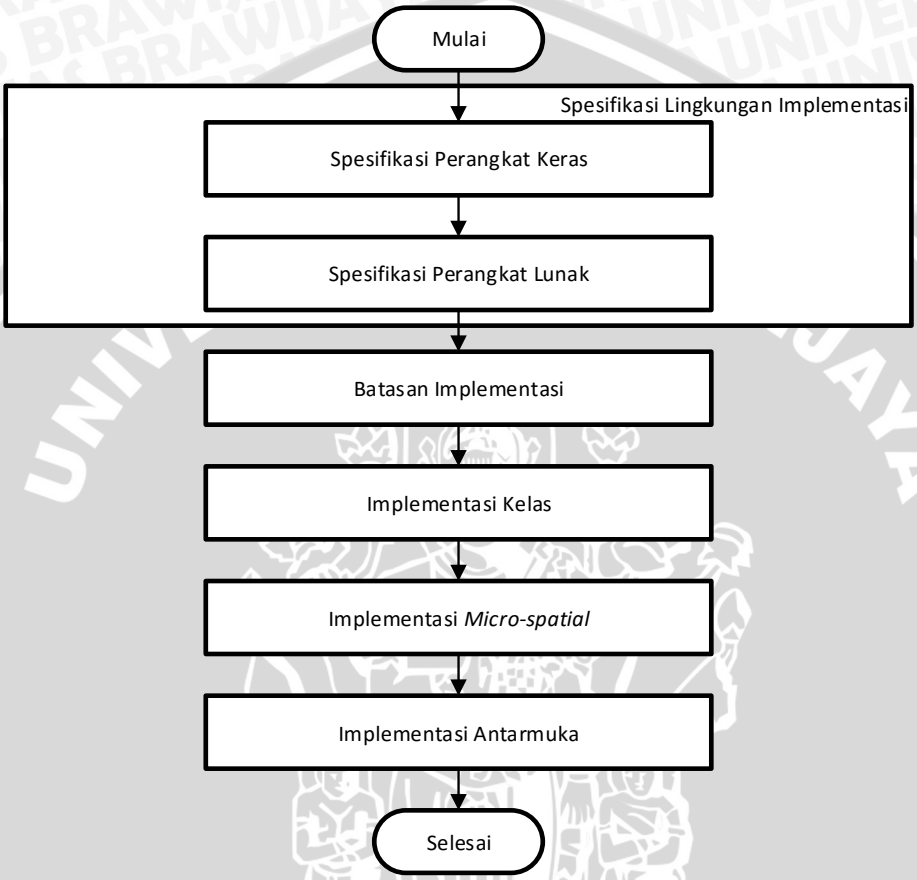
1. *header* sistem ApsOrcim
2. menu sistem ApsOrcim
3. kolom pencarian berdasarkan nama pedagang Pasar Besar
4. pilihan komoditas
5. peta Pasar Besar
6. informasi peta
7. tombol legenda
8. tombol pilihan *circle map tool*
9. tombol pilihan *rectangle map tool*



Gambar 4.17 Perancangan antarmuka *dasymetric* mapping

BAB 5 IMPLEMENTASI

Bagian ini memaparkan tahapan implementasi sistem ApsOrcim yang mengacu kepada tahap Perancangan. Bahasan pada bab ini terdiri dari penjelasan terkait spesifikasi lingkungan implementasi, batasan implementasi, implementasi kelas, implementasi *micro-spatial*, dan implementasi antarmuka.



Gambar 5.1 Diagram alir implementasi

5.1 Spesifikasi Lingkungan Implementasi

Bagian ini menjelaskan spesifikasi lingkungan implementasi pada sistem yang terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam implementasi sistem dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

HP 430	
Processor	Intel® Core™ i3-2350M CPU @ 2.30GHz
Memory (RAM)	4.00 GB

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat keras yang digunakan dalam implementasi sistem dapat dilihat pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

HP 430	
Sistem Operasi	Windows 8.1 © 2013 x64
Bahasa Pemrograman	HTML, PHP, CSS, JavaScript
Piranti Lunak	Apache MapServer 3.1.1 Sublime Text 3 Quantum GIS 2.8.1 pgAdmin PostgreSQL Tools 1.20.0 PostGIS 2.0 Shapefile and DBF Loader Exporter DBDesigner 4.0.5.6 Enterprise Architect 12.0.1215 Microsoft Visio 2013 Microsoft Word 2016 Microsoft Excel 2016 Google Chrome Versi 50.0.2661.94 m SortSite

5.2 Batasan Implementasi

Bagian ini menjelaskan batasan implementasi sistem, yaitu:

a. *Database Management System* yang digunakan adalah PostgreSQL versi 1.20.0.
b. *Framework CodeIgniter* yang digunakan adalah versi 2.2.6.
c. MapServer yang digunakan adalah versi 3.1.1.

5.3 Implementasi Kelas

Bagian ini merupakan realisasi file dengan ekstensi *.php dari setiap kelas yang telah dirancang pada tahap perancangan. Implementasi kelas dapat dilihat pada Tabel 5.3.

Tabel 5.3 Implementasi kelas

No	*Package*	Nama Kelas	Nama *File*
1	*controllers*	*dashboard*	dashboard.php
2	*controllers*	*dasymetric*	*dasymetric*.php
3	*controllers*	komoditas	komoditas.php

Tabel 5.3 Implementasi kelas

No	*Package*	Nama Kelas	Nama *File*
4	*controllers*	*log_in*	log_in.php
5	*controllers*	pedagang	pedagang.php
6	*controllers*	peta	peta.php
7	*models*	m_blok	m_blok.php
8	*models*	m_kategori_komoditas	m_kategori_komoditas.php
9	*models*	m_kios	m_kios.php
10	*models*	m_klasifikasi_komoditas	m_klasifikasi_komoditas.php
11	*models*	m_komoditas	m_komoditas.php
12	*models*	m_kota_distributor	m_kota_distributor.php
13	*models*	m_pedagang	m_pedagang.php
14	*models*	m_pedagang_has_komoditas	m_pedagang_has_komoditas.php
15	*models*	m_pedagang_has_kota_distributor	m_pedagang_has_kota_distributor.php
16	*models*	m_pengguna	m_pengguna.php

5.4 Implementasi *Micro-spatial*

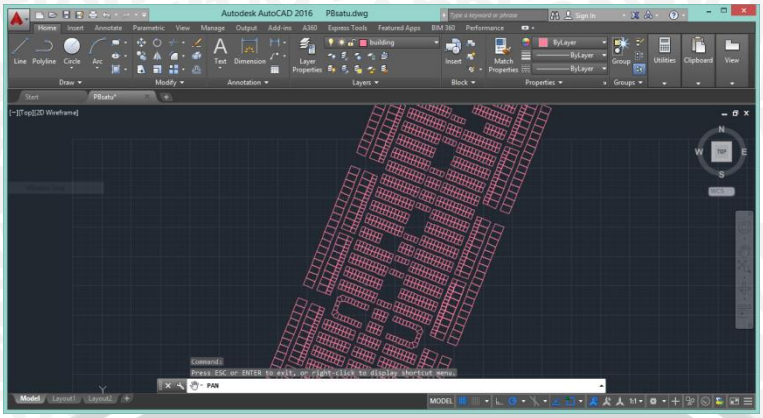
Implementasi analisis *micro-spatial* ini merupakan implementasi kebutuhan fungsional dengan kode fungsi MSpa-03 dan MSpa-04. Implementasi ini terdiri dari beberapa bagian, diantaranya: 1) digitalisasi peta Pasar Besar, 2) implementasi perangkat lunak SIG, 3) implementasi *dasymetric*, dan 4) *map tools and domain analysis*.

5.4.1 Digitalisasi Peta Pasar Besar

Bagian ini memaparkan bagaimana cara mengonversikan denah yang didapat dari manajemen Pasar Besar menjadi peta digital. Proses ini memerlukan perangkat lunak AutoCad. Pada perangkat lunak AutoCad juga akan dilakukan pemberian proyeksi pada peta terkait. Diagram alir digitalisasi peta Pasar Besar dapat dilihat pada Gambar 5.2.

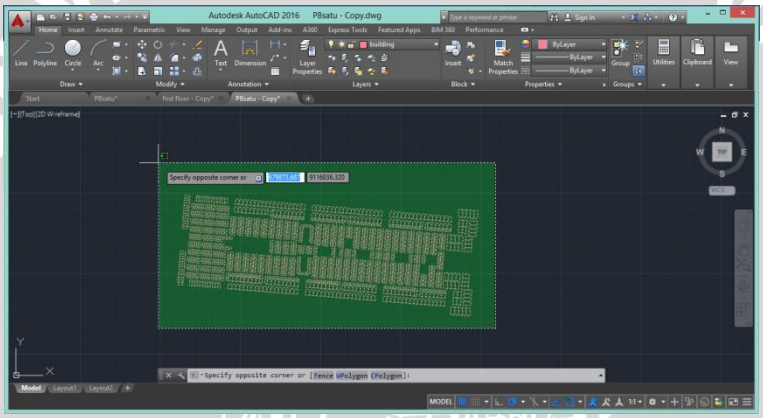
Gambar 5.2 Diagram alir digitasi peta Pasar Besar

Denah Pasar Besar yang didapat dari pengelola Pasar Besar diolah menjadi peta digital menggunakan bantuan perangkat lunak AutoCad.



Gambar 5.3 Digitalisasi denah Pasar Besar

Beri proyeksi pada denah tersebut dengan menggunakan perintah `ALIGN`. Blok seluruh *polygon* yang terdapat pada denah Pasar Besar kemudian ketikkan perintah `ALIGN` pada kolom *command*.



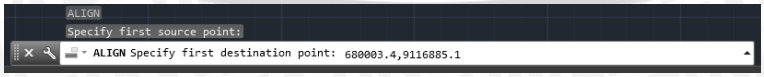
Gambar 5.4 Proses blok *polygon* pada perangkat lunak AutoCad

Gambar 5.5 Perintah ALIGN pada kolom *command*

Pilih ujung point kiri pada denah untuk diberi proyeksi pada perintah `Specify first source point`.

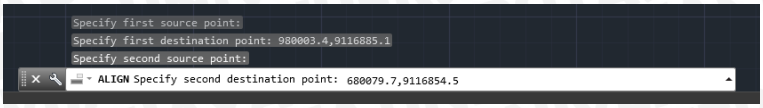
Gambar 5.6 Proses menentukan poin pertama

Isi koordinat yang diperlukan pada perintah `Specify first destination point`, koordinat yang dimasukkan merupakan koordinat standar UTM.



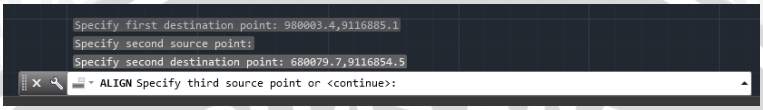
Gambar 5.7 Proses menentukan tujuan *point* pertama

Pilih ujung point kanan pada denah untuk diberi proyeksi pada perintah `Specify second source point`. Kemudian isi koordinat yang diperlukan pada perintah `Specify second destination point`.



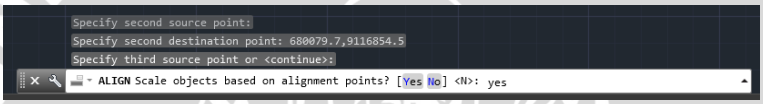
Gambar 5.8 Proses menentukan tujuan *point* kedua

Pada perintah `Specify second source point or <continue>` tekan enter.



Gambar 5.9 Proses menentukan *point* ketiga atau melanjutkan

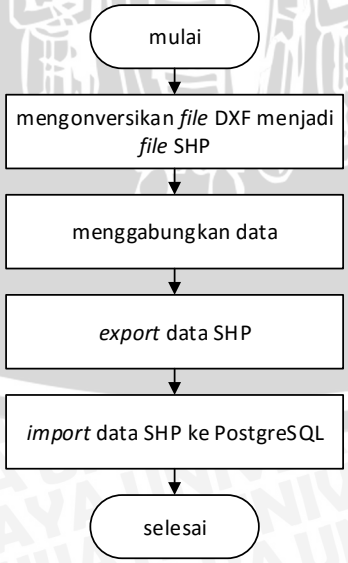
Isi yes pada perintah `Scale objects based on alignment points?` sehingga denah akan terproyeksi.



Gambar 5.10 Proses menentukan skala pada objek

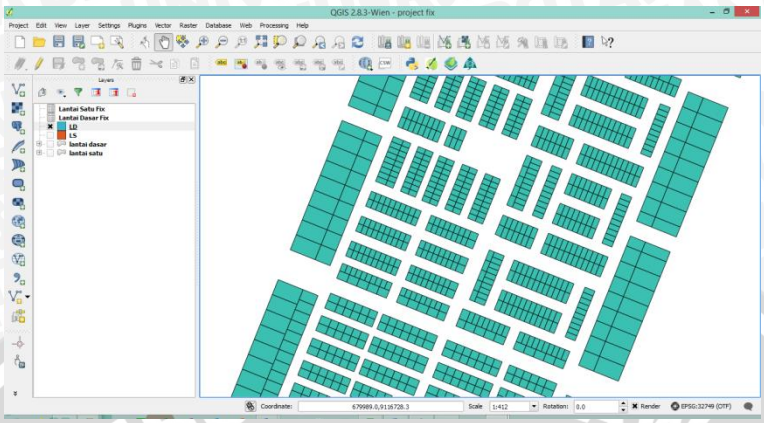
5.4.2 Implementasi Perangkat Lunak SIG

Implementasi perangkat lunak SIG ini menggunakan perangkat lunak Quantum GIS (QGIS). Data yang diolah pada perangkat lunak QGIS adalah data pedagang yang didapat dari Dinas Pasar yang kemudian di-*export* menjadi *file* dengan ekstensi *.shp. Gambar 5.11 merupakan diagram alir implementasi perangkat lunak SIG.



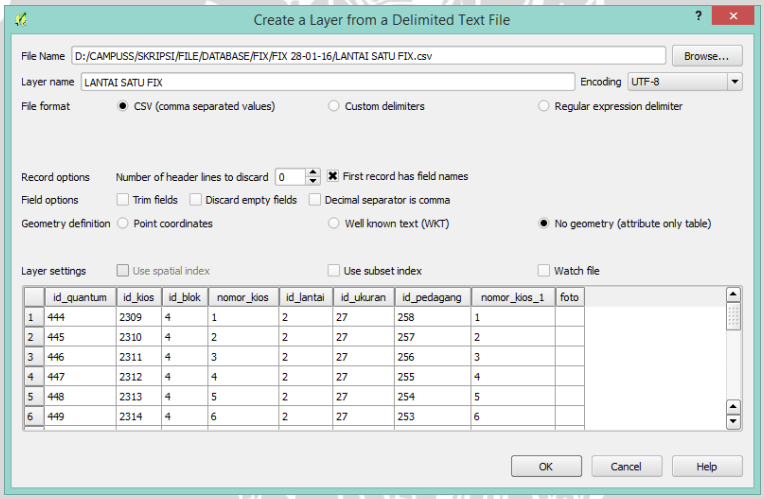
Gambar 5.11 Diagram alir implementasi perangkat lunak SIG

Untuk mengonversikan *file* dengan ekstensi *.dxf ke *file* dengan ekstensi *.shp diperlukan bantuan perangkat lunak QGIS. Buka *file* *.dxf pada QGIS dengan cara menambahkan data vektor pada *layer*.



Gambar 5.12 Tampilan *file* *.dxf pada Quantum GIS

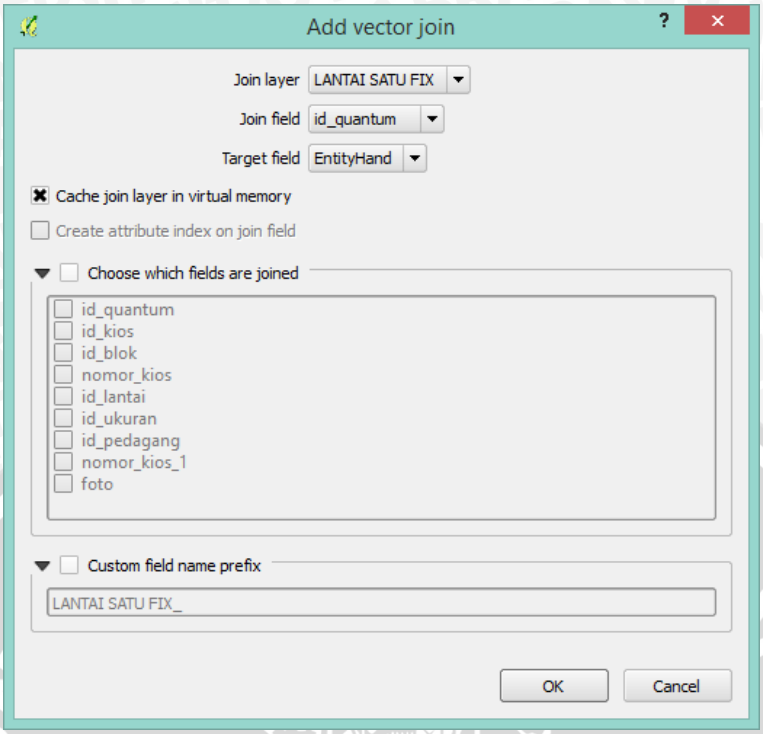
Olah data kios, beri tambahan kolom sesuai dengan kolom `EntityHand` pada data yang terdapat di QGIS. Masukkan *file* (*.csv) pada QGIS dengan aturan seperti pada Gambar 5.13.



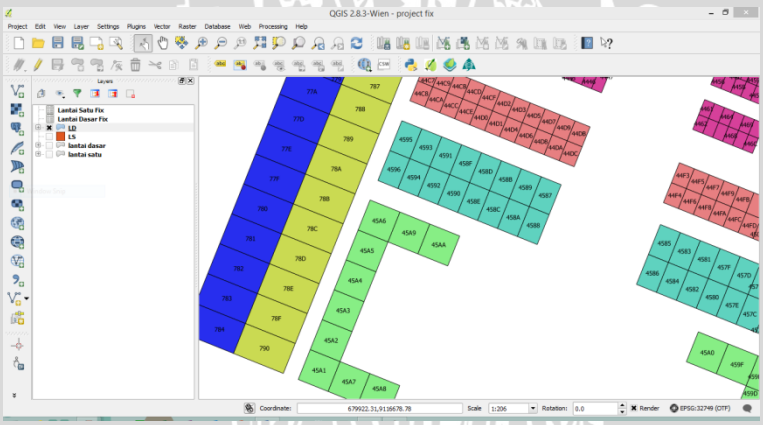
	id_quantum	id_kios	id_blok	nomor_kios	id_lantai	id_ukuran	id_pedagang	nomor_kios_1	foto
1	444	2309	4	1	2	27	258	1	
2	445	2310	4	2	2	27	257	2	
3	446	2311	4	3	2	27	256	3	
4	447	2312	4	4	2	27	255	4	
5	448	2313	4	5	2	27	254	5	
6	449	2314	4	6	2	27	253	6	

Gambar 5.13 Menambahkan *layer* dari *delimited text file*

Untuk menggabungkan data csv dan peta lakukan penambahan *join* pada data vektor dengan cara klik kanan pada peta kemudian pilih menu *properties*, kemudian pilih menu *join* dan tambahkan data *vector join* seperti pada Gambar 5.14. Hasil penambahan data dapat dilihat pada Gambar 5.15 yang menampilkan peta dengan klasifikasi warna berdasarkan blok pada kios.

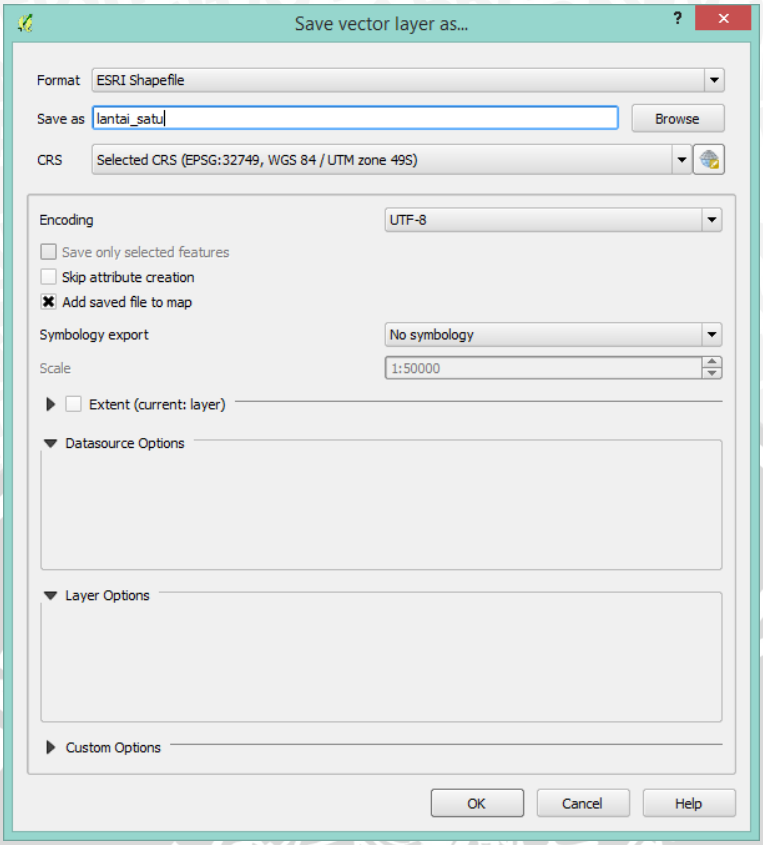


Gambar 5.14 Tampilan *add vector join*



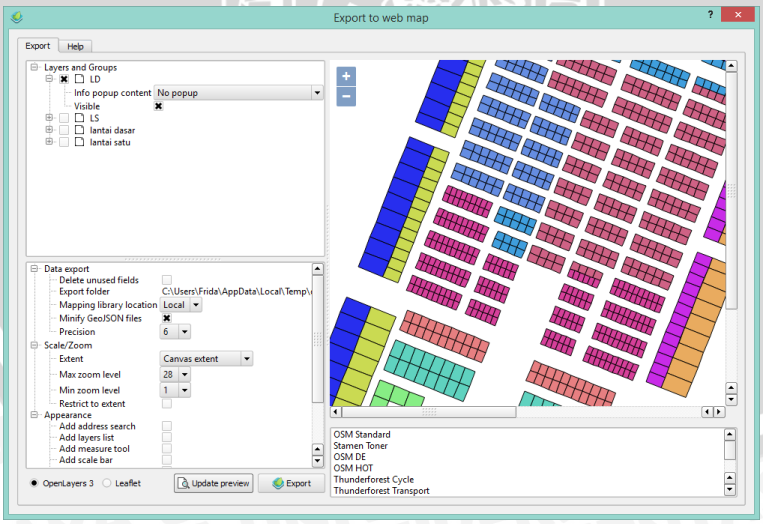
Gambar 5.15 Tampilan peta dengan klasifikasi warna berdasarkan blok

Simpan *layer* vektor peta dengan data yang telah digabung dengan cara klik kanan dan pilih menu *properties*, kemudian pilih *save as*. Pada Gambar 5.16 klik *browse* dan pilih lokasi penyimpanan.



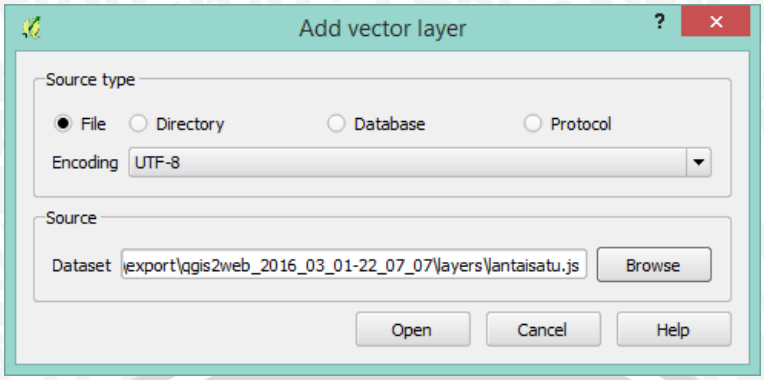
Gambar 5.16 Tampilan simpan *layer* vektor

Export file dalam bentuk *web* menggunakan *plugin Leaflet* yang terdapat di QGIS seperti pada Gambar 5.17. Centang pada data yang diperlukan untuk di-*export*.



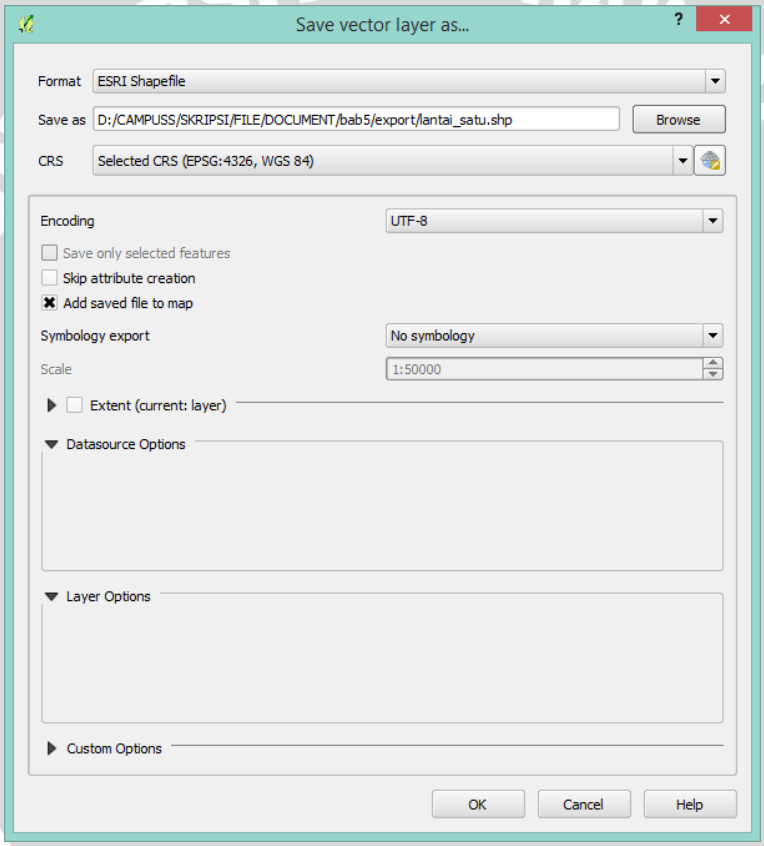
Gambar 5.17 Tampilan *export to web map*

Tambahkan *file* GeoJson yang telah di-*export* ke QGIS dengan cara menambahkan *file* vektor untuk mendapatkan proyeksi WGS 84.



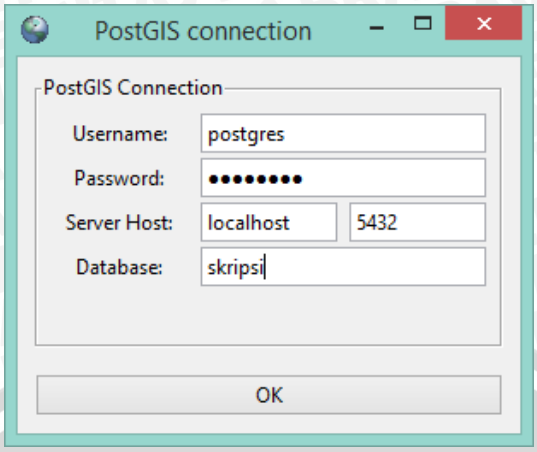
Gambar 5.18 Tampilan *add vector layer*

Simpan *layer* vektor GeoJson agar menjadi *file* dengan ekstensi *.shp yang memiliki proyeksi WGS 84.



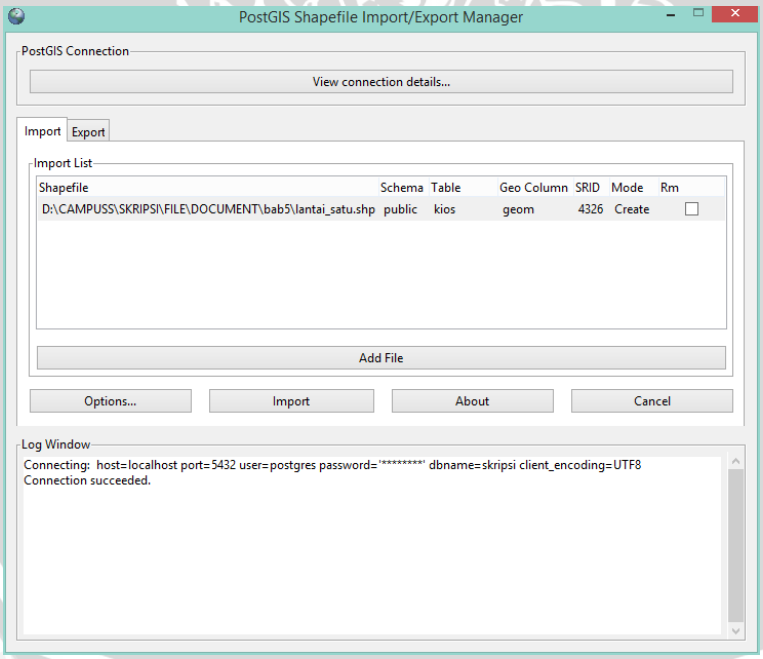
Gambar 5.19 Tampilan simpan *file* GeoJson dengan ekstensi *.shp

Berikut ini adalah cara menggabungkan data shp dengan data yang ada di basis data. Penggabungan data dibantu oleh perangkat lunak PostGIS. Gambar 5.20 merupakan tampilan gambar dalam membuat koneksi dengan basis data.



Gambar 5.20 Tampilan membuat koneksi dengan basis data

Tekan tombol *add file* pada "*PostGIS Shapefile Import/Export Manager*" kemudian pilih *file* shp denah Pasar Besar yang telah diolah sebelumnya. Lengkapi kolom Table dengan isian kios yang merupakan tabel pada basis data, Geo Column merupakan nama kolom yang akan di-*generate* secara otomatis oleh aplikasi tersebut yang akan menghasilkan data *geometry* dari setiap batas *polygon* kios Pasar Besar. Ubah SRID menjadi 4326 yang merupakan proyeksi dari WGS 84. Tekan tombol *import*. Akan muncul notifikasi `Shapefile import completed`.



Gambar 5.21 Tampilan *import/export* manager pada PostGIS

5.4.3 Implementasi *Dasymetric*

Bagian ini menjelaskan bagaimana mengimplementasikan metode *dasymetric* untuk mendapatkan karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas yang terdapat pada Pasar Besar.

5.4.3.1 Perhitungan Manual Dasymetric Algorithm

Data yang digunakan untuk perhitungan *dasymetric algorithm* merupakan data hasil survei yang telah diolah. Tabel 5.4 merupakan data nilai variabel perhitungan. Data variabel dan hasil perhitungan *dasymetric algorithm* keseluruhan dapat dilihat pada LAMPIRAN B.

Tabel 5.4 Nilai variabel perhitungan *dasymetric algorithm*

Klasifikasi Komoditas	Jumlah Kios Lantai Dasar dan Satu	Luas Kios Klasifikasi Lantai Dasar dan Satu	Luas Kios Klasfikasi Lantai Terpilih
	Lantai Satu		
Pakaian jadi	280	1286.5	1148

Dengan menggunakan persamaan 2.1 akan diperoleh nilai persentase klasifikasi komoditas Pasar Besar. Klasifikasi pakaian di lantai satu akan menjadi contoh perhitungan. Data nilai variabel dan hasil perhitungan

Variabel $K_{klasifikasi\ pakaian}$ = 280 (jumlah total kios klasifikasi pakaian)

Variabel $A_{klasifikasi\ pakaian\ lantai\ satu}$ = 1148 (luas area kios klasifikasi pakaian di lantai satu)

Variabel $\sum_f^n A_{total\ klasifikasi\ pakaian}$ = 1286.5 (luas total area kios klasifikasi pakaian)

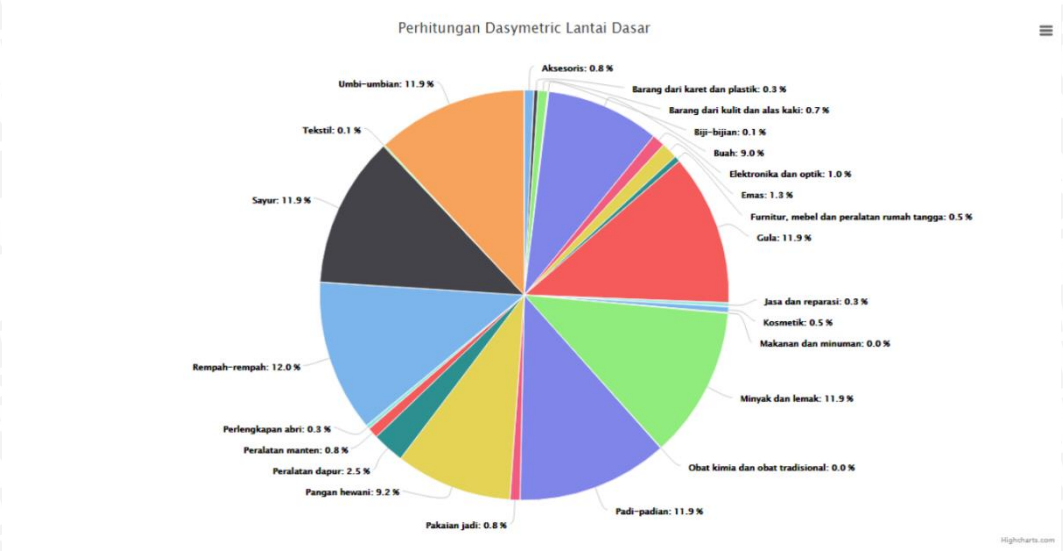
Nilai $D_{klasifikasi\ pakaian\ lantai\ satu}$ $= \frac{280}{1286.5} 1148$

$= 249.86$

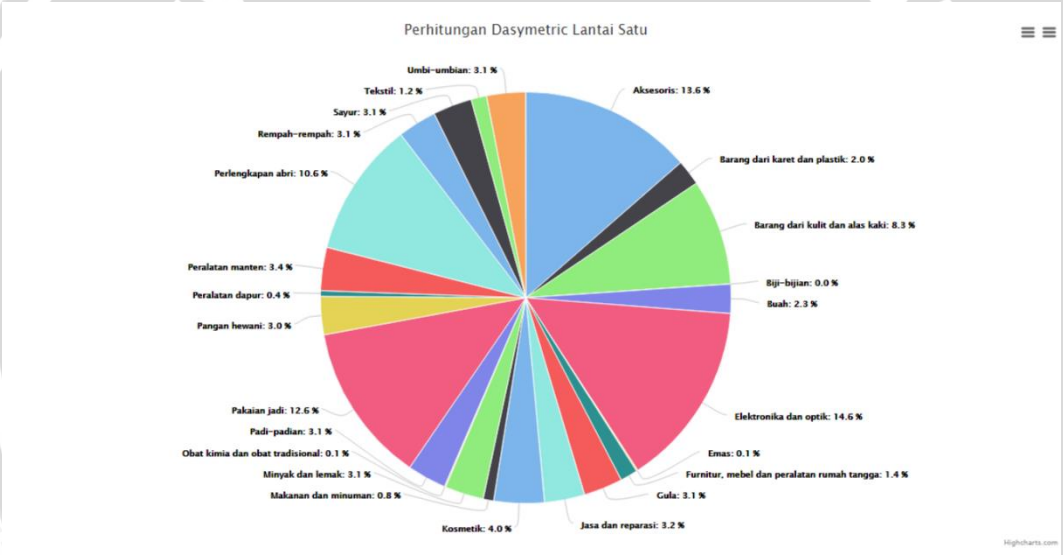
Berdasarkan nilai perhitungan tersebut dapat diketahui persentase hasil perhitungan *dasymetric algorithm* klasifikasi pakaian lantai satu dapat dilihat pada Tabel 5.5. Hasil lengkap persentase perhitungan dapat dilihat pada Gambar 5.22 dan Gambar 5.23.

Tabel 5.5 Hasil perhitungan klasifikasi pakaian jadi lantai satu

Hasil	Persentase
249.86	$\frac{249.86}{1987.57} x100 = \mathbf{12.6\%}$



Gambar 5.22 Hasil perhitungan *dasymetric algorithm* lantai dasar



Gambar 5.23 Hasil perhitungan *dasymetric* algorithm lantai satu

5.4.3.2 Source Code Implementasi Metode Dasymetric

Tabel 5.6 merupakan *source code* yang digunakan untuk menampilkan data hasil perhitungan *dasymetric algorithm*.

Tabel 5.6 *Source code* implementasi grafik *dasymetric*

No	*Source Code*
1	`$data['data_perhitungan'] = $this->m_kios->PerhitunganDasymetric();`
2	`$this->load->view('admin/layout/header');`
3	`$this->load->view('admin/dasymetric_chart', $data);`
4	`$this->load->view('admin/layout/footer');`

5.4.3.3 Source Code Implementasi Dasymetric Mapping

Tabel 5.7 merupakan *source code* yang digunakan untuk menampilkan data peta *dasymetric mapping* lantai dasar.

Tabel 5.7 *Source code* implementasi *dasymetric mapping* lantai dasar

No	*Source Code*
1	`$this->m_kios->set(array('id_blok' => 1));`
2	`$data['abrakan'] = $this->m_kios->GetGeometryBYBlok();`
3	`$this->m_kios->set(array('id_blok' => 2));`
4	`$data['barat_a'] = $this->m_kios->GetGeometryBYBlok();`
5	`$this->m_kios->set(array('id_blok' => 3));`
6	`$data['barat_b'] = $this->m_kios->GetGeometryBYBlok();`
7	`$this->m_kios->set(array('id_blok' => 6));`
8	`$data['buah'] = $this->m_kios->GetGeometryBYBlok();`
9	`$this->m_kios->set(array('id_blok' => 8));`
10	`$data['emas'] = $this->m_kios->GetGeometryBYBlok();`
11	`$this->m_kios->set(array('id_blok' => 9));`
12	`$data['ikan'] = $this->m_kios->GetGeometryBYBlok();`
13	`$this->m_kios->set(array('id_blok' => 10));`
14	`$data['kemasan'] = $this->m_kios->GetGeometryBYBlok();`
15	`$this->m_kios->set(array('id_blok' => 13));`
16	`$data['makanan'] = $this->m_kios->GetGeometryBYBlok();`
17	`$this->m_kios->set(array('id_blok' => 14));`
18	`$data['mebel'] = $this->m_kios->GetGeometryBYBlok();`
19	`$this->m_kios->set(array('id_blok' => 16));`
20	`$data['pecah_belah'] = $this->m_kios->GetGeometryBYBlok();`
21	`$this->m_kios->set(array('id_blok' => 18));`
22	`$data['pracangan'] = $this->m_kios->GetGeometryBYBlok();`
23	`$this->m_kios->set(array('id_blok' => 19));`
24	`$data['sayur'] = $this->m_kios->GetGeometryBYBlok();`
25	`$this->m_kios->set(array('id_blok' => 20));`
26	`$data['selatan_a'] = $this->m_kios->GetGeometryBYBlok();`
27	`$this->m_kios->set(array('id_blok' => 21));`
28	`$data['selatan_b'] = $this->m_kios->GetGeometryBYBlok();`
29	`$this->m_kios->set(array('id_blok' => 24));`
30	`$data['timur_a'] = $this->m_kios->GetGeometryBYBlok();`
31	`$this->m_kios->set(array('id_blok' => 25));`
32	`$data['timur_b'] = $this->m_kios->GetGeometryBYBlok();`
33	`$this->m_kios->set(array('id_blok' => 28));`
34	`$data['utara_a'] = $this->m_kios->GetGeometryBYBlok();`

Tabel 5.7 *Source code* implementasi *dasymetric mapping* lantai dasar

No	*Source Code*
35	`$this->m_kios->set(array('id_blok' => 29));`
36	`$data['utara_aa'] = $this->m_kios->GetGeometryBYBlok();`
37	`$this->m_kios->set(array('id_blok' => 30));`
38	`$data['utara_b'] = $this->m_kios->GetGeometryBYBlok();`
39	`$this->m_kios->set(array('id_blok' => 31));`
40	`$data['utara_bb'] = $this->m_kios->GetGeometryBYBlok();`
41	`$this->m_kios->set(array('id_lantai' => 1));`
42	`$data['peta_titik'] = $this->m_kios->GetGeometryPoint();`
43	`$data['lantai'] = $this->m_kios->GetGeometryBYLantai();`
44	`$this->m_blok->set(array('id_lantai' => 1));`
45	`$data['jumlah_kios'] = $this->m_blok->getJumlahKiosByBlok();`
46	`$data['list_komoditas'] = $this->m_komoditas->getData();`
47	`$this->load->view('admin/dasymetric_lantai_dasar', $data);`

Tabel 5.8 merupakan *source code* yang digunakan untuk menampilkan data peta *dasymetric mapping* lantai dasar.

Tabel 5.8 *Source code* implementasi *dasymetric mapping* lantai satu

No	*Source Code*
1	`$this->m_kios->set(array('id_blok' => 4));`
2	`$data['barat_c'] = $this->m_kios->GetGeometryBYBlok();`
3	`$this->m_kios->set(array('id_blok' => 5));`
4	`$data['barat_d'] = $this->m_kios->GetGeometryBYBlok();`
5	`$this->m_kios->set(array('id_blok' => 7));`
6	`$data['elektronika'] = $this->m_kios->GetGeometryBYBlok();`
7	`$this->m_kios->set(array('id_blok' => 11));`
8	`$data['konveksi'] = $this->m_kios->GetGeometryBYBlok();`
9	`$this->m_kios->set(array('id_blok' => 12));`
10	`$data['makanan_1'] = $this->m_kios->GetGeometryBYBlok();`
11	`$this->m_kios->set(array('id_blok' => 15));`
12	`$data['palen'] = $this->m_kios->GetGeometryBYBlok();`
13	`$this->m_kios->set(array('id_blok' => 17));`
14	`$data['perlengkapan_abri'] = $this->m_kios->GetGeometryBYBlok();`
15	`$this->m_kios->set(array('id_blok' => 22));`
16	`$data['selatan_c'] = $this->m_kios->GetGeometryBYBlok();`
17	`$this->m_kios->set(array('id_blok' => 23));`
18	`$data['sepatu'] = $this->m_kios->GetGeometryBYBlok();`
19	`$this->m_kios->set(array('id_blok' => 26));`
20	`$data['timur_c'] = $this->m_kios->GetGeometryBYBlok();`

Tabel 5.8 *Source code* implementasi *dasymetric mapping* lantai satu

No	*Source Code*
21	`$this->m_kios->set(array('id_blok' => 27));`
22	`$data['timur_d'] = $this->m_kios->GetGeometryBYBlok();`
23	`$this->m_kios->set(array('id_blok' => 32));`
24	`$data['utara_c'] = $this->m_kios->GetGeometryBYBlok();`
25	`$this->m_kios->set(array('id_blok' => 33));`
26	`$data['utara_d'] = $this->m_kios->GetGeometryBYBlok();`
27	`$this->m_kios->set(array('id_lantai' => 2));`
28	`$data['peta_titik'] = $this->m_kios->GetGeometryPoint();`
29	`$data['lantai'] = $this->m_kios->GetGeometryBYLantai();`
30	`$this->m_blok->set(array('id_lantai' => 2));`
31	`$data['jumlah_kios'] = $this->m_blok->getJumlahKiosByBlok();`
32	`$data['list_komoditas'] = $this->m_komoditas->getData();`
33	`$this->load->view('admin/dasymetric_lantai_satu', $data);`

5.4.4 *Map Tools and Domain Analysis*

Tabel 5.9 menunjukkan ringkasan dari *map tool and domain analysis* pada analisis *Micro-spatial*. Cara Analisis dibagi menjadi dua kelompok, menemukan domain dan menemukan lokasi. *Point tool* digunakan untuk menemukan lokasi dari parameter tertentu, sedangkan *circle tool* dan *rectangle tool* merupakan analisis geometri yang digunakan untuk menentukan domain analisis total populasi komoditas. *Circle tool* menentukan wilayah analisis dengan membuat lingkaran berdasarkan radius yang ditentukan oleh pengguna. Pengguna dapat menentukan posisi pusat dari lingkaran tersebut menggunakan *point tool*. *Rectangle tool* menentukan wilayah analisis dengan membuat persegi panjang. Pengguna dapat menemukan lokasi kios serta hasil perhitungan *dasymetric algorithm* berdasarkan komoditas yang akan dianalisis.

Tabel 5.9 *Map tools and domain analysis*

Map Tool	Parameter				Mode Analisis
	TPK	KP	KK	LL	
Point	☒	☑	☑	☑	Menemukan lokasi
Circle	☑	☒	☒	☒	Menemukan domain
Rectangle	☑	☒	☒	☒	

Keterangan :
TPK = Total populasi komoditas
KP = Kios Pedagang
KK = Kios Komoditas
LL = Nilai *longitude* dan *latitude*

5.5 Implementasi Antarmuka Pengguna

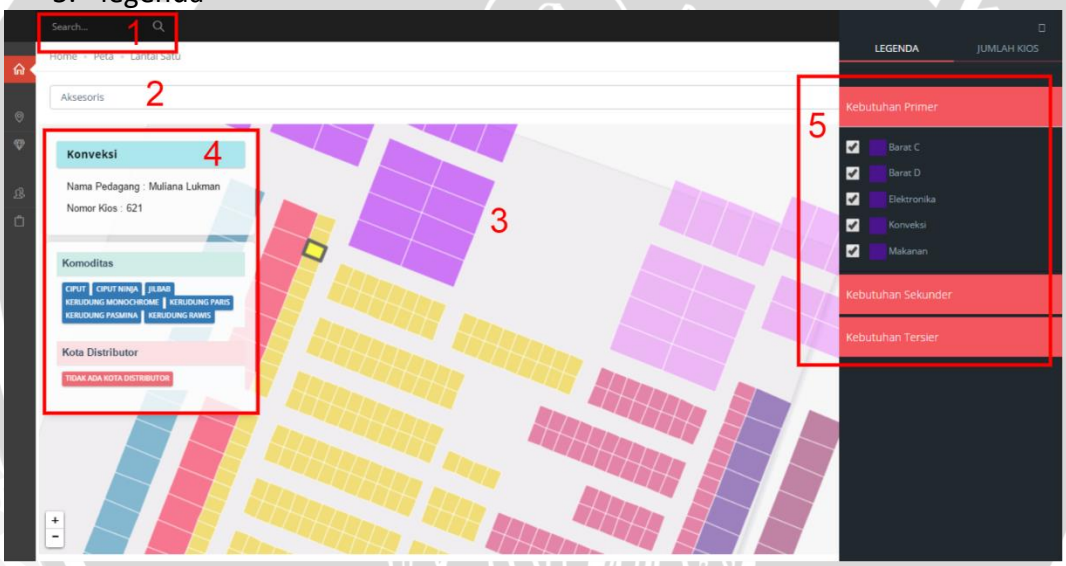
Antarmuka pada bagian ini merupakan hasil implementasi antarmuka sesuai dengan rancangan pada tahap Perancangan Antarmuka.

5.5.1 Antarmuka Mengelola Peta

Antarmuka mengelola peta ditunjukkan oleh Gambar 5.24. Halaman ini menyediakan layanan untuk mengelola peta, sehingga dapat melakukan pencarian kios berdasarkan pedagang dengan parameter masukan berupa nama pedagang, dan memfilter kios berdasarkan komoditas yang dijual dengan parameter masukan komoditas.

Berikut keterangan fungsi tampilan Gambar 5.24:

1. kolom pencarian berdasarkan nama pedagang Pasar Besar
2. pilihan komoditas
3. peta Pasar Besar
4. informasi peta
5. legenda



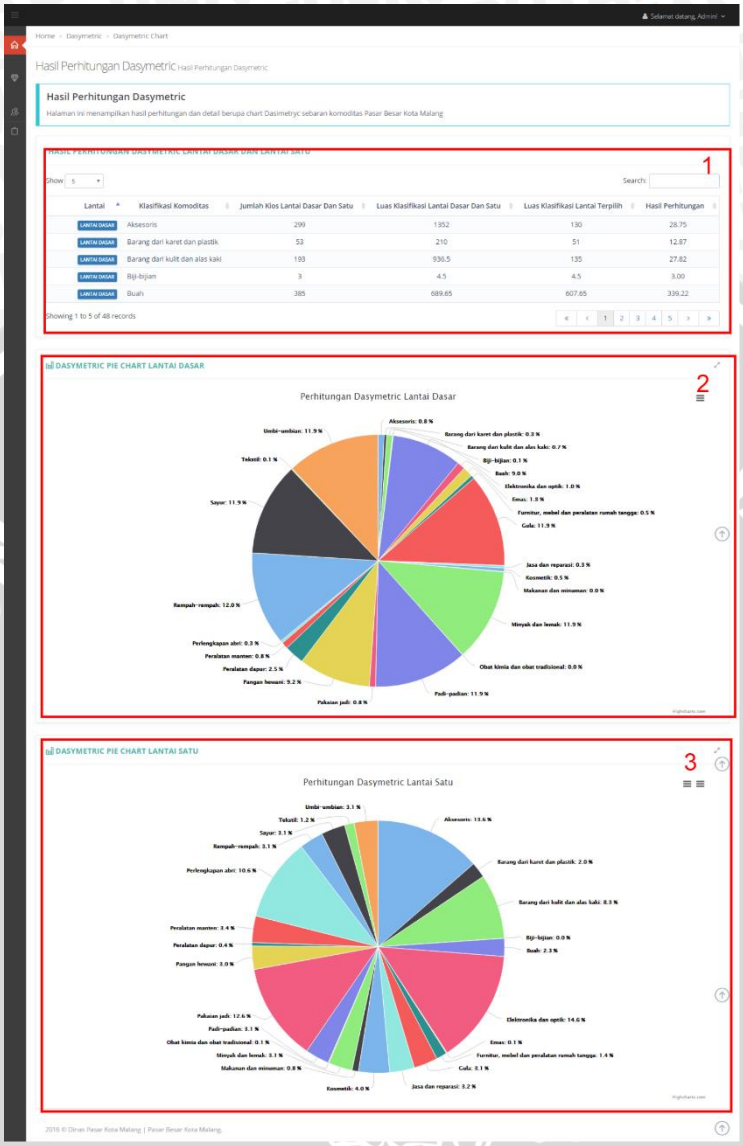
Gambar 5.24 Implementasi antarmuka mengelola peta

5.5.2 Antarmuka Mengelola *Dasymetric*

Antarmuka mengelola *dasymetric* ditunjukkan oleh Gambar 5.24, Gambar 5.25, Gambar 5.26, Gambar 5.27, dan Gambar 5.28. Halaman ini menyediakan layanan untuk mengelola data *dasymetric* yang meliputi data tabular, grafik dan peta dengan parameter masukan longitude, latitude, radius dan komoditas sehingga dapat ditampilkan visualisasi hasil perhitungan *dasymetric algorithm*.

Gambar 5.25 merupakan halaman yang menyediakan data hasil perhitungan *dasymetric algorithm* berupa data tabular dan grafik *pie*. Berikut keterangan fungsi tampilan Gambar 5.25:

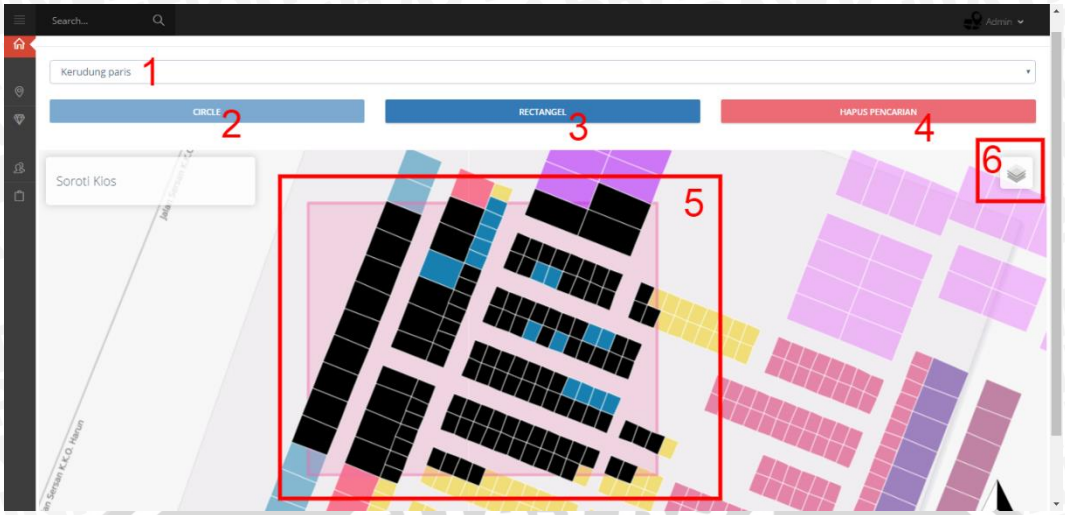
1. tabel hasil perhitungan *dasymetric algorithm*
2. grafik *pie* hasil perhitungan *dasymetric algorithm* lantai dasar
3. grafik *pie* hasil perhitungan *dasymetric algorithm* lantai satu



Gambar 5.25 Implementasi antarmuka grafik *dasymetric*

Gambar 5.26 merupakan antarmuka yang menyediakan informasi dalam bentuk visualisasi berupa peta. Terdapat *map tools* yang berfungsi untuk melihat informasi pencarian dan hasil perhitungan *dasymetric algorithm* dengan cara menggambar *circle tool* atau *rectangle tool* pada peta. Berikut keterangan fungsi tampilan Gambar 5.26:

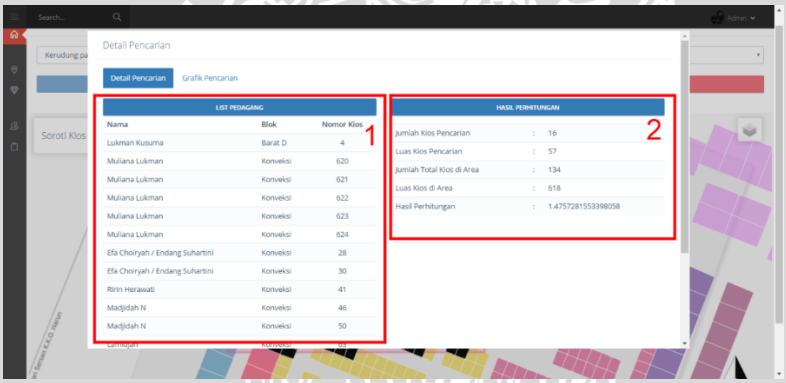
1. pilihan komoditas
2. tombol *circle map tool*
3. tombol *rectangle map tool*
4. tombol hapus pencarian
5. area pencarian
6. tombol legenda



Gambar 5.26 Implementasi antarmuka *dasymetric* mapping

Gambar 5.27 merupakan antarmuka detail hasil pencarian. Hasil pencarian yang ditampilkan adalah data tabular berupa daftar pedagang dan hasil perhitungan *dasymetric algorithm* sesuai area pencarian. Berikut keterangan fungsi tampilan Gambar 5.27:

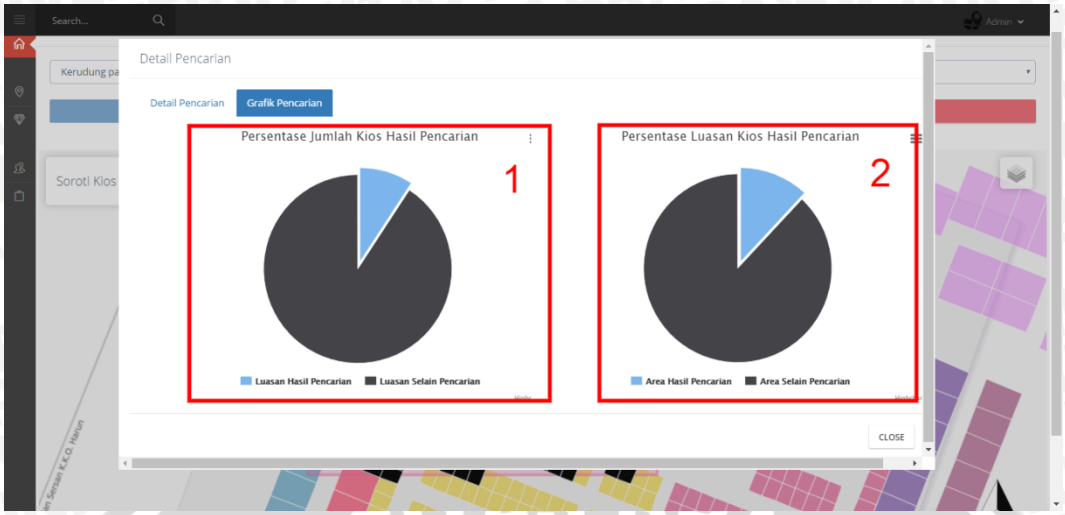
1. tabel daftar pedagang hasil pencarian
2. tabel hasil perhitungan *dasymetric algorithm*



Gambar 5.27 Implementasi antarmuka *map tools* detail pencarian

Gambar 5.28 merupakan antarmuka detail hasil pencarian. Hasil pencarian yang ditampilkan adalah data grafik berupa persentase jumlah kios dan luasan kios sesuai area pencarian. Berikut keterangan fungsi tampilan Gambar 5.28:

1. grafik persentase jumlah kios hasil pencarian
2. grafik persentase luasan kios hasil pencarian



Gambar 5.28 Implementasi antarmuka *map tools* grafik pencarian

BAB 6 PENGUJIAN

Pada bab ini memaparkan tahap pengujian dan analisis dari sistem ApsOrcim. Pengujian sistem ini bertujuan untuk menguji komponen sistem yang telah dirancang sebelumnya dan memastikan bahwa setiap fungsional dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengujian dilakukan dengan metode *black-box*, pengujian *portability*, dan *RMSE*.

6.1 Pengujian *Black-box* (Uji Validasi)

Tujuan dilakukannya pengujian *black-box* adalah untuk memastikan setiap kebutuhan fungsional perangkat lunak yang didefinisikan sesuai dengan yang diharapkan. Pengujian setiap kebutuhan fungsional pada Tabel 4.2 diuji dengan mendefinisikan kasus uji terhadap setiap kebutuhan tersebut kemudian membandingkannya dengan hasil yang diperoleh. Tabel 6.1 memaparkan seluruh pengujian menghasilkan status valid, sehingga dapat disimpulkan bahwa kebutuhan fungsional sistem dapat berjalan sesuai dengan spesifikasi kebutuhan yang sudah dijabarkan pada analisis kebutuhan.

Tabel 6.1 Hasil pengujian *black-box*

No.	Nama Kasus Uji	Kode Fungsi	Kasus Uji	Hasil yang Diharapkan	Hasil	Status
1	Pengujian *login*	MSpa-01	Pengujian *login* dengan *username* dan *password* yang sesuai (terdaftar dalam sistem).	Sistem menampilkan halaman dasbor.	Sistem menampilkan halaman dasbor.	Valid
2			Pengujian *login* dengan *username* atau *password* salah atau tidak terisi (tidak terdaftar dalam sistem).	Sistem menampilkan halaman *login* dengan pesan "Kombinasi *username* dan *password* tidak ditemukan!".	Sistem menampilkan halaman *login* dengan pesan "Kombinasi *username* dan *password* tidak ditemukan!".	Valid
3	Pengujian mengelola dasbor	MSpa-02	Pengujian dengan menekan menu dasbor.	Sistem menampilkan halaman dasbor.	Sistem menampilkan halaman dasbor.	Valid
4	Pengujian mengelola peta	MSpa-03	Pengujian dengan menekan menu peta.	Sistem menampilkan halaman peta.	Sistem menampilkan halaman peta.	Valid
5			Pengujian dengan mengisi nama pedagang pada kolom pencarian.	Sistem menampilkan hasil pencarian dengan memperbesar kios yang sesuai dengan data pencarian.	Sistem menampilkan hasil pencarian dengan memperbesar kios yang sesuai dengan data pencarian.	Valid
6			Pengujian dengan memilih komoditas pada	Sistem menampilkan hasil pencarian dengan memberi warna pada	Sistem menampilkan hasil pencarian dengan memberi warna pada	Valid

Tabel 6.1 Hasil pengujian *black-box*

No.	Nama Kasus Uji	Kode Fungsi	Kasus Uji	Hasil yang Diharapkan	Hasil	Status
			drop down pilihan komoditas.	kios yang sesuai dengan data pencarian.	kios yang sesuai dengan data pencarian.	
7	Pengujian mengelola *dasymetric*	MSpa-04	Pengujian dengan menekan menu grafik *dasymetric*.	Sistem menampilkan halaman grafik *dasymetric*.	Sistem menampilkan halaman grafik *dasymetric*.	Valid
8			Pengujian dengan menekan menu *dasymetric mapping*.	Sistem menampilkan halaman peta.	Sistem menampilkan halaman peta.	Valid
9			Pengujian dengan menekan tombol *circle* dan mengisi data pencarian.	Sistem menampilkan *pop-up* berisi detail data kios, data hasil perhitungan, dan data grafik yang sesuai dengan data pencarian.	Sistem menampilkan *pop-up* berisi detail data kios, data hasil perhitungan, dan data grafik yang sesuai dengan data pencarian.	Valid
10			Pengujian dengan menekan tombol *rectangle*.	Sistem menampilkan *pop-up* berisi detail data kios, data hasil perhitungan, dan data grafik yang sesuai dengan data pencarian.	Sistem menampilkan *pop-up* berisi detail data kios, data hasil perhitungan, dan data grafik yang sesuai dengan data pencarian.	Valid
11	Pengujian mengelola data pedagang	MSpa-05	Pengujian dengan menekan menu pedagang.	Sistem menampilkan halaman data pedagang.	Sistem menampilkan halaman data pedagang.	Valid
12			Pengujian dengan mengisi semua data dan menekan tombol tambahkan! pada halaman tambah data pedagang.	Data pedagang berhasil ditambahkan dan menampilkan pesan “Data pedagang berhasil ditambahkan.”.	Data pedagang berhasil ditambahkan dan menampilkan pesan “Data pedagang berhasil ditambahkan.”.	Valid
13			Pengujian dengan mengosongkan beberapa/semua data dan menekan tombol tambahkan! pada halaman tambah data pedagang.	Data pedagang gagal ditambahkan dan menampilkan pesan yang menandakan bahwa kolom harus di isi.	Data pedagang gagal ditambahkan dan menampilkan pesan yang menandakan bahwa kolom harus di isi.	Valid
14			Pengujian dengan	Data pedagang berhasil dihapus dan	Data pedagang berhasil dihapus dan	Valid

Tabel 6.1 Hasil pengujian *black-box*

No.	Nama Kasus Uji	Kode Fungsi	Kasus Uji	Hasil yang Diharapkan	Hasil	Status
			menekan tombol hapus pada halaman pedagang.	sistem menampilkan pesan "Data pedagang berhasil dihapus.".	sistem menampilkan pesan "Data pedagang berhasil dihapus.".	
15			Pengujian dengan menekan tombol detail pada halaman pedagang.	Sistem menampilkan halaman detail pedagang.	Sistem menampilkan halaman detail pedagang.	Valid
16			Pengujian dengan mengisi semua data dan menekan tombol simpan pada *pop-up* perbarui data pedagang halaman detail pedagang.	Sistem menampilkan pesan "Data pedagang berhasil diperbarui.".	Sistem menampilkan pesan "Data pedagang berhasil diperbarui.".	Valid
17			Pengujian dengan memilih komoditas dan menekan tombol simpan pada *pop-up* tambah data komoditas halaman detail pedagang.	Data komoditas berhasil ditambahkan dan sistem menampilkan pesan "Data komoditas berhasil ditambahkan.".	Data komoditas berhasil ditambahkan dan sistem menampilkan pesan "Data komoditas berhasil ditambahkan.".	Valid
18			Pengujian dengan menekan tombol hapus pada *pop-up* hapus data komoditas halaman detail pedagang.	Data komoditas berhasil dihapus dan sistem menampilkan pesan "Data komoditas berhasil dihapus.".	Data komoditas berhasil dihapus dan sistem menampilkan pesan "Data komoditas berhasil dihapus.".	Valid
19			Pengujian dengan memilih kios dan menekan tombol simpan pada *pop-up* tambah data kios halaman detail pedagang.	Data kios berhasil ditambahkan dan sistem menampilkan pesan "Data kios berhasil ditambahkan.".	Data kios berhasil ditambahkan dan sistem menampilkan pesan "Data kios berhasil ditambahkan.".	Valid
20			Pengujian dengan menekan tombol hapus pada *pop-up* hapus data kios halaman detail pedagang.	Data kios berhasil dihapus dan sistem menampilkan pesan "Data kios berhasil dihapus.".	Data kios berhasil dihapus dan sistem menampilkan pesan "Data kios berhasil dihapus.".	Valid

Tabel 6.1 Hasil pengujian *black-box*

No.	Nama Kasus Uji	Kode Fungsi	Kasus Uji	Hasil yang Diharapkan	Hasil	Status
21	Pengujian mengelola data komoditas	MSpa-06	Pengujian dengan menekan menu komoditas.	Sistem menampilkan halaman data komoditas.	Sistem menampilkan halaman data komoditas.	Valid
22			Pengujian dengan mengisi semua data dan menekan tombol simpan pada halaman tambah data komoditas.	Data komoditas berhasil ditambahkan dan menampilkan pesan "Data komoditas berhasil ditambahkan.".	Data komoditas berhasil ditambahkan dan menampilkan pesan "Data komoditas berhasil ditambahkan.".	Valid
23			Pengujian dengan mengosongkan beberapa/semua data dan menekan tombol simpan pada halaman tambah data komoditas.	Data komoditas gagal ditambahkan dan menampilkan pesan yang menandakan bahwa kolom harus di isi.	Data komoditas gagal ditambahkan dan menampilkan pesan yang menandakan bahwa kolom harus di isi.	Valid
24			Pengujian dengan mengisi semua data dan menekan tombol simpan pada *pop-up* perbarui data komoditas halaman komoditas.	Data komoditas berhasil diperbarui dan sistem menampilkan pesan "Data komoditas berhasil diperbarui.".	Data komoditas berhasil diperbarui dan sistem menampilkan pesan "Data komoditas berhasil diperbarui.".	Valid
25			Pengujian dengan mengisi semua data dan menekan tombol simpan pada *pop-up* perbarui data klasifikasi komoditas halaman komoditas.	Data klasifikasi komoditas berhasil diperbarui dan sistem menampilkan pesan "Data klasifikasi komoditas berhasil diperbarui.".	Data klasifikasi komoditas berhasil diperbarui dan sistem menampilkan pesan "Data klasifikasi komoditas berhasil diperbarui.".	
26			Pengujian dengan mengisi data dan menekan tombol simpan pada *pop-up* perbarui data kategori komoditas halaman komoditas.	Data kategori komoditas berhasil diperbarui dan sistem menampilkan pesan "Data kategori komoditas berhasil diperbarui.".	Data kategori komoditas berhasil diperbarui dan sistem menampilkan pesan "Data kategori komoditas berhasil diperbarui.".	Valid
27	Pengujian *logout*	MSpa-07	Pengujian dengan	Sistem menampilkan halaman *login*.	Sistem menampilkan halaman *login*.	Valid

Tabel 6.1 Hasil pengujian *black-box*

No.	Nama Kasus Uji	Kode Fungsi	Kasus Uji	Hasil yang Diharapkan	Hasil	Status
			menekan tombol *logout.*			

6.2 Pengujian *Portability*

Pengujian *portability* dilakukan dengan menguji setiap halaman sistem pada desktop browser (Edge v13, Firefox v46, Firefox v44, Safari v8.0 dan v9.0, Opera v36, Google Chrome v49) dan mobile browser (iOS v6.0, v8.0, dan v9.0, Android v4.0, dan BlackBerry v10.0) sehingga memiliki tingkat kompabilitas ≥ 80%. Pengujian dilakukan dengan bantuan perangkat lunak SortSite. Tabel 6.2 memaparkan hasil pengujian dimana terdapat 100% nilai valid pada kategori *critical issues, major issues, dan minor issues* yang menandakan bahwa sistem telah memenuhi kebutuhan non-fungsional pada parameter *portability* pada Tabel 4.3. Hasil pengujian *portability* dapat dilihat secara lengkap pada LAMPIRAN D.

Tabel 6.2 Kalkulasi hasil pengujian *portability*

Kategori	Tingkat Kompabilitas Seluruh Halaman Sistem
Critical Issues	12 versi *browser* dari 12 versi *browser* = 100%
Major Issues	12 versi *browser* dari 12 versi *browser* = 100%
Minor Issues	12 versi *browser* dari 12 versi *browser* = 100%

6.3 Pengujian *RMSE*

Pengujian *RMSE* dilakukan dengan membandingkan data perhitungan *dasymetric algorithm* secara manual. Semakin besar nilai RMSE, maka hasil estimasi model semakin tidak tepat dibandingkan dengan pengamatan. Pengujian terhadap 48 data hasil perhitungan *dasymetric algorithm* menghasilkan valid. Hasil perhitungan pengujian *RMSE* dapat dilihat pada LAMPIRAN C.

Tabel 6.3 Hasil perhitungan *dasymetric algorithm*

Data ke-	Klasifikasi Komoditas	Hasil perhitungan sistem	Hasil perhitungan manual
Lantai Satu			
1	Pakaian	249.86	249.86

Dengan menggunakan persamaan 2.2 akan diperoleh nilai eror dari hasil perhitungan pada Tabel 6.3. Dengan menggunakan persamaan (2.2) diperoleh hasil pengujian *RMSE* data ke-1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_1 - \bar{x}_1)^2}{n}}$$

$$= \sqrt{\frac{\sum_{i=1}^{n}(249.86 - 249.86)^2}{48}}$$

$$= 0$$

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut:

1. Karakteristik kepadatan populasi komoditas pada setiap klasifikasi komoditas di Pasar Besar Kota Malang didapatkan dengan menghitung *dasymetric algorithm*. Data populasi komoditas, kios, posisi lantai, klasifikasi komoditas di Pasar Besar digunakan untuk perhitungan algoritme tersebut. Perhitungan tersebut menghasilkan persentase klasifikasi komoditas yang terdapat di Pasar Besar. Hasil *dasymetric algorithm* dapat digunakan sebagai data pendukung analisis manajemen Kota Malang.
2. Pengembangan *webGIS* untuk visualisasi klasifikasi komoditas di Pasar Besar dilakukan dalam beberapa tahap, antara lain:
   a. Digitalisasi peta pasar besar.
      Denah Pasar Besar dikonversi menjadi peta digital dan diberi proyeksi peta menggunakan perangkat lunak AutoCad.
   b. Implementasi pada perangkat SIG.
      Data peta digital digabungkan dengan data pedagang Pasar Besar menggunakan perangkat lunak QuantumGIS. Data tersebut di-*export* menjadi data SHP, kemudian di-*import* ke PostgreSQL.
   c. Implementasi *dasymetric*.
      Implementasi *dasymetric* dilakukan dengan menghitung *dasymetric algorithm* secara manual, kemudian implementasi perhitungan pada sistem dan implementasi *dasymetric mapping* untuk memberikan hasil visualisasi perhitungan *dasymetric algorithm*.
3. Tingkat akurasi *dasymetric algorithm* di ukur menggunakan metode *root mean square error (RMSE)* menghasilkan valid pada setiap klasifikasi komoditas. Pengujian dilakukan dengan membandingkan data perhitungan *dasymetric algorithm* secara manual dengan perhitungan sistem.

7.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya dijabarkan sebagai berikut:

1. Dapat dikembangkan menjadi *mobile platform* seperti android dan iOS.
2. Penambahan *overlay map* pada peta Pasar Besar.
3. Pengembangan bentuk 3D dan penambahan ruas jalan pada peta Pasar Besar.

DAFTAR PUSTAKA

A.S., R. & Shalahuddin, M., 2014. *Rekayasa Perangkat Lunak Terstruktur dan Beriorientasi Objek.* Bandung: Informatika Bandung.

Esri, 2016. *Support.* [Online] Available at: http://support.esri.com/en/knowledgebase/Gisdictionary/browse [Diakses 1 Maret 2016].

Hermawan, J., 2015. *Analisa & Desain Pemrograman Berorientasi Obyek dengan UML dan VISUAL BASIC. NET.* s.l.:Penerbit Andi.

Lwin, K. K. & Murayama, Y., 2010. Online Micro-spatial Analysis Based on GIS Estimated Building Population: A Case of Tsukuba City. Januari.

PowerMapper, 2016. *SortSite - Browser Compatibility Tests.* [Online] Available at: http://www.powermapper.com/products/sortsite/checks/browser-compatibility/ [Diakses 21 Maret 2016].

Prahasta, E., 2009. *Sistem Informasi Geografis: Konsep-konsep Dasar (Perpektif Geodesi & Geomatika).* Bandung: Informatika.

Pressman, R. S., 2010. *Software Engineering: A Practitioner'S Approach.* SEVENTH penyunt. NewYork: McGraw-Hill.

Salonen, V., 2012. *Automatic Portability Testing.* Jyväskylä: University of Jyväskylä.

Satuhu, Y. M., Nugroho, A. M. & Wilandari, L. D., 2010. Redesain Pasar Bareng Kota Malang (Perancangan Pasar Tradisional Bercitra Modern).

Seidl, M., Scholz, M., Huemer, C. & Kappel, G., 2012. *UML @ Classroom An Introduction to Object-Oriented Modeling.* Heidelberg: s.n.

Sleeter, R., 2001. Dasymetric Mapping Techniques For The San Francisco Bay Region, California.

Su, M.-D.et al., 2010. Multi-layer Multi-class Dasymetric Mapping to Estimate Population Distribution. *Elsevier,* pp. 4807-4816.

LAMPIRAN A DATA PEDAGANG PASAR BESAR

DATA PEDAGANG
PASAR BESAR MALANG
KOMODITAS : E M A S

NO	NAMA	ALAMAT	NOMOR			UKURAN	KETERANGAN
			AKAD	LANTAI	TEMPAT		
a	b	c	d	e	f	g	h
1	Agus Karim	Kol. Sugiono IV/604	688	D	1	2,0 x 2,0	
2	Hari Mulyono	Ngaglik II/339	343	D	2	2,0 x 2,0	
3	Suyuthi Basri	Emb. Brantas II/1620	573	D	3	2,0 x 1,5	
4	Moch. Harry	Kotalama III A /921	348	D	4	2,0 x 1,5	
5	Suyuthi Basri	Emb. Brantas II/1620	573	D	5	2,0 x 1,5	
6	Zainin As'ari	KH. Asyari IV/10	1310	D	6	2,0 x 1,5	
7	Mislikah	G. Subroto II/575	482	D	7	2,0 x 1,5	
8	Ali Kusnadi	G. Subroto II/551	1111	D	8	2,0 x 1,5	
9	Umi Latifah	Kepuh VII/51	1768	D	9	2,0 x 1,5	
10	H. Achmad Rifa'i	Kol. Sugiono III B / 40 Rt/Rw. 007/008	369	D	10	2,0 x 1,5	
11	Badrus Salam	Kepuh VII/51	1259	D	11	2,0 x 1,5	
12	H. Achmad Rifa'i	Kol. Sugiono III B / 40 Rt/Rw. 007/008	349	D	12	2,0 x 1,5	
13	Badrus Salam	Kepuh VII/51	1259	D	13	2,0 x 1,5	
14	M. Syaiful Islam	KA Gribig 332	354	D	14	2,0 x 1,5	
15	Abd. Halim Seta	PM. Yamin V/206	1172	D	15	2,0 x 1,5	
16	Abd. Hamid A	G. Subroto II/550	2036	D	16	2,0 x 1,5	
17	Abd. Halim Seta	PM. Yamin V/206	1172	D	17	2,0 x 1,5	
18	Abd. Hamid A	G. Subroto II/550	2036	D	18	2,0 x 1,5	
19	Sutardjo	Kol. Sugiono III A / 233	Kontan	D	19	2,0 x 1,5	
20	H. Bachtiar A	P. Galang 10	453	D	20	2,0 x 1,5	

a	b	c	d	e	f	g	h
21	Sutardjo	Kol. Sugiono III A / 233	Kontan	D	21	2,0 x 1,5	
22	H. Bachtiar A	P. Galang 10	453	D	22	2,0 x 1,5	
23	Zaenal Arifin	A. Munandar II/1072	583	D	23	2,0 x 1,5	
24	Amirudin	M. Sarkoro IIIG/2408	362	D	24	2,0 x 1,5	
25	Zaenal Arifin	A. Munandar II/1072	583	D	25	2,0 x 1,5	
26	M. Toyib	Raya Tlogomas III/36	476	D	26	2,0 x 1,5	
27	Mutomimah	K.A. Gribig IV/104	1849	D	27	2,0 x 1,5	
28	Fitri Umatiyah	K.A. Gribig IV/104	1852	D	28	2,0 x 1,5	
29	Mutomimah	K.A. Gribig IV/104	1849	D	29	2,0 x 1,5	
30	Fitri Umatiyah	K.A. Gribig IV/104	1852	D	30	2,0 x 1,5	
31	Mutomimah	K.A. Gribig IV/104	1849	D	31	2,0 x 1,5	
32	Fitri Umatiyah	K.A. Gribig IV/104	1852	D	32	2,0 x 1,5	
33	H. Suraji SSH	Kol. Sugiono V/24	2197	D	33	2,0 x 1,5	
34	H. Achmad N / Saiful Islam	Jl. Muharto VII / 14	2174	D	34	2,0 x 1,5	
35	H. Suraji SSH	Kol. Sugiono V/24	2197	D	35	2,0 x 1,5	
36	H. Achmad N / Saiful Islam	Jl. Muharto VII / 14	2174	D	36	2,0 x 1,5	
37	Ach. Marjuki	KHZ Arifin Gg Buntu 72	1836	D	37	2,0 x 1,5	
38	Achmadi		Kontan	D	38	2,0 x 1,5	
39	Machfud	Jodipan Wetan I/2149	472	D	39	2,0 x 2,0	
40	Moch. Yusi Lukito	G. Subroto III/25	350	D	40	2,0 x 2,0	
41	H. Imam/Najib	L. Martadinata III/92	2010	D	41	2,0 x 1,5	
42	Moch. Kusaeri	Kotalama - 2 - 850A Rt/Rw. 005/003	352	D	42	2,0 x 1,5	
43	H. Siti Rochmah	L. Martadinata III/921	344	D	43	2,0 x 1,5	
44	Djakfar	G. Subroto II/25	361	D	44	2,0 x 1,5	
45	Nurhakim / Anik Wulandari	Jl. Jayasrani 1 / 71-39	2123	D	45	2,0 x 1,5	

a	b	c	d	e	f	g	h
46	Susanto	G. Subroto II/548	351	D	46	2,0 x 1,5	
47	Abd. Halim / Anik Wulandari	Jl. Jayasrani 1 / 71-39	2123	D	47	2,0 x 1,5	
48	M. Saleh	Muharto VB/43	432	D	48	2,0 x 1,5	
49	Dewi Masitah MZ	Juanda V/12 A	1552	D	49	2,0 x 1,5	
50	Fadelan	Ngaglik IV B/1946	1652	D	50	2,0 x 1,5	
51	Dewi Masitah MZ	Juanda V/12 A	1552	D	51	2,0 x 1,5	
52	Fadelan	Ngaglik IV B/1946	1652	D	52	2,0 x 1,5	
53	Rosiati	Simp. Dieng 23	1638	D	53	2,0 x 1,5	
54	Hj. Mutia Farida	Jend. Basuki Rachmad VI/987	Kontan	D	54	2,0 x 1,5	
55	Rosiati	Simp. Dieng 23	1638	D	55	2,0 x 1,5	
56	Hj. Mutia Farida	Jend. Basuki Rachmad VI/987	Kontan	D	56	2,0 x 1,5	
57	Hari Kriswanto	Gadang XV/36	2109	D	57	2,0 x 1,5	
58	Buadi Ali	Muharto IV/45	Kontan	D	58	2,0 x 1,5	
59	Budiono	Kol. Sugiono III/17	Kontan	D	59	2,0 x 1,5	
60	Buadi Ali	Muharto IV/45	Kontan	D	60	2,0 x 1,5	
61	Kusnianto	Ciliwung II/26	Kontan	D	61	2,0 x 1,5	
62	Saturi	Kotalama III/17	Kontan	D	62	2,0 x 1,5	
63	Kusnianto	Ciliwung II/26	Kontan	D	63	2,0 x 1,5	
64	Saturi	Kotalama III/17	Kontan	D	64	2,0 x 1,5	
65	Andi Susanto	Rupat 24	1869	D	65	2,0 x 1,5	
66	Andi Susanto	Rupat 24	1869	D	66	2,0 x 1,5	
67	Moch. Fadil AR	PM Yamin III/152	377	D	67	2,0 x 1,5	
68	Abu Bakar	Z. Arifin IA/44	447	D	68	2,0 x 1,5	
69	Abdullah	Gadang VI/43	1629	D	69	2,0 x 1,5	
70	Abdullah	Gadang VI/43	1629	D	70	2,0 x 1,5	

Malang, Agustus 2005

C

LAMPIRAN B HASIL PERHITUNGAN *DASYMETRIC ALGORITHM*

Klasifikasi Komoditas	Jumlah Kios Lantai Dasar dan Satu	Luas Kios Klasifikasi Lantai Dasar dan Satu	Luas Kios Klasfikasi Lantai Terpilih	Hasil Perhitungan
Lantai Dasar				
Aksesoris	299	1352	130	**28.75**
Barang dari karet dan plastik	53	210	51	**12.87**
Barang dari kulit dan alas kaki	193	936.5	135	**27.82**
Biji-bijian	3	4.5	4.5	**3**
Buah	385	689.65	607.65	**339.22**
Elektronika dan optik	329	1395.75	164.75	**38.83**
Emas	51	388	371	**48.77**
Furnitur, mebel dan peralatan rumah tangga	45	167.45	65.45	**17.59**
Gula	512	974.4	858.4	**451.05**
Jasa dan reparasi	75	400.75	64.75	**12.12**
Kosmetik	96	616.5	110.5	**17.21**
Makanan dan minuman	18	66	6	**1.64**
Minyak dan lemak	512	974.4	858.4	**451.05**
Obat kimia dan obat tradisional	3	10	4	**1.2**
Padi-padian	512	974.4	858.4	**451.05**
Pakaian jadi	280	1286.5	138.5	**30.14**
Pangan hewani	407	905.6	773.6	**347.68**
Peralatan dapur	105	342.2	313.2	**96.1**
Peralatan manten	99	408.5	130.5	**31.63**
Perlengkapan abri	223	1101.1	58.5	**11.85**
Rempah-rempah	514	978.4	862.4	**453.06**
Sayur	512	974.4	858.4	**451.05**
Tekstil	29	315	51	**4.7**
Umbi-umbian	512	974.4	858.4	**451.05**
Total				**3779.43**

Klasifikasi Komoditas	Jumlah Kios Lantai Dasar dan Satu	Luas Kios Klasifikasi Lantai Dasar dan Satu	Luas Kios Klasfikasi Lantai Terpilih	Hasil Perhitungan
Lantai Satu				
Aksesoris	299	1352	1222	**270.25**
Barang dari karet dan plastik	53	210	159	**40.13**
Barang dari kulit dan alas kaki	193	936.5	801.5	**165.18**
Biji-bijian	0	0	0	**0**
Buah	385	689.65	82	**45.78**
Elektronika dan optik	329	1395.75	1231	**290.17**
Emas	51	388	17	**2.23**
Furnitur, mebel dan peralatan rumah tangga	45	167.45	102	**27.41**
Gula	512	974.4	116	**60.95**
Jasa dan reparasi	75	400.75	336	**62.88**
Kosmetik	96	616.5	506	**78.79**
Makanan dan minuman	18	66	60	**16.36**
Minyak dan lemak	512	974.4	116	**60.95**
Obat kimia dan obat tradisional	3	10	6	**1.8**
Padi-padian	512	974.4	116	**60.95**
Pakaian jadi	280	1286.5	1148	**249.86**
Pangan hewani	407	905.6	132	**59.32**
Peralatan dapur	105	342.2	29	**8.9**
Peralatan manten	99	408.5	278	**67.37**
Perlengkapan abri	223	1101.1	1042.6	**211.15**
Rempah-rempah	514	978.4	116	**60.94**
Sayur	512	974.4	116	**60.95**
Tekstil	29	315	264	**24.3**
Umbi-umbian	512	974.4	116	**60.95**
Total				**1987.57**

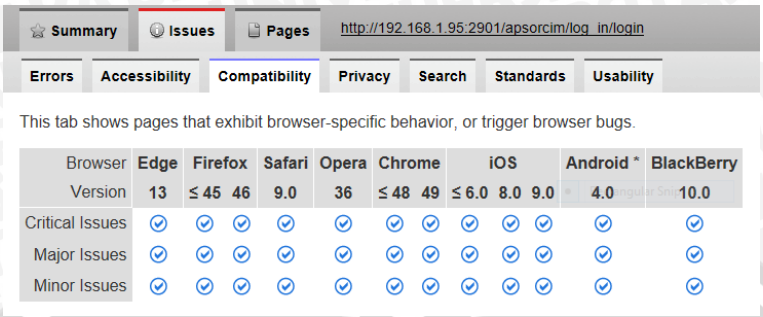
LAMPIRAN C HASIL PENGUJIAN *RMSE*

Data ke-	Klasifikasi Komoditas	Hasil perhitungan sistem	Hasil perhitungan manual	Hasil pengujian *RMSE*
Lantai Dasar				
1	Aksesoris	28.75	28.75	**0**
2	Barang dari karet dan plastik	12.87	12.87	**0**
3	Barang dari kulit dan alas kaki	27.82	27.82	**0**
4	Biji-bijian	3	3	**0**
5	Buah	339.22	339.22	**0**
6	Elektronika dan optik	38.83	38.83	**0**
7	Emas	48.77	48.77	**0**
8	Furnitur, mebel dan peralatan rumah tangga	17.59	17.59	**0**
9	Gula	451.05	451.05	**0**
10	Jasa dan reparasi	12.12	12.12	**0**
11	Kosmetik	17.21	17.21	**0**
12	Makanan dan minuman	1.64	1.64	**0**
13	Minyak dan lemak	451.05	451.05	**0**
14	Obat kimia dan obat tradisional	1.2	1.2	**0**
15	Padi-padian	451.05	451.05	**0**
16	Pakaian jadi	30.14	30.14	**0**
17	Pangan hewani	347.68	347.68	**0**
18	Peralatan dapur	96.1	96.1	**0**
19	Peralatan manten	31.63	31.63	**0**
20	Perlengkapan abri	11.85	11.85	**0**
21	Rempah-rempah	453.06	453.06	**0**
22	Sayur	451.05	451.05	**0**
23	Tekstil	4.7	4.7	**0**
24	Umbi-umbian	451.05	451.05	**0**

Data ke-	Klasifikasi Komoditas	Hasil perhitungan sistem	Hasil perhitungan manual	Hasil pengujian *RMSE*
Lantai Satu				
25	Aksesoris	270.25	270.25	**0**
26	Barang dari karet dan plastik	40.13	40.13	**0**
27	Barang dari kulit dan alas kaki	165.18	165.18	**0**
28	Biji-bijian	0	0	**0**
29	Buah	45.78	45.78	**0**
30	Elektronika dan optik	290.17	290.17	**0**
31	Emas	2.23	2.23	**0**
32	Furnitur, mebel dan peralatan rumah tangga	27.41	27.41	**0**
33	Gula	60.95	60.95	**0**
34	Jasa dan reparasi	62.88	62.88	**0**
35	Kosmetik	78.79	78.79	**0**
36	Makanan dan minuman	16.36	16.36	**0**
37	Minyak dan lemak	60.95	60.95	**0**
38	Obat kimia dan obat tradisional	1.8	1.8	**0**
39	Padi-padian	60.95	60.95	**0**
40	Pakaian jadi	249.86	249.86	**0**
41	Pangan hewani	59.32	59.32	**0**
42	Peralatan dapur	8.9	8.9	**0**
43	Peralatan manten	67.37	67.37	**0**
44	Perlengkapan abri	211.15	211.15	**0**
45	Rempah-rempah	60.94	60.94	**0**
46	Sayur	60.95	60.95	**0**
47	Tekstil	24.3	24.3	**0**
48	Umbi-umbian	60.95	60.95	**0**

LAMPIRAN D HASIL PENGUJIAN *PORTABILITY*

D.1 Halaman *Login*



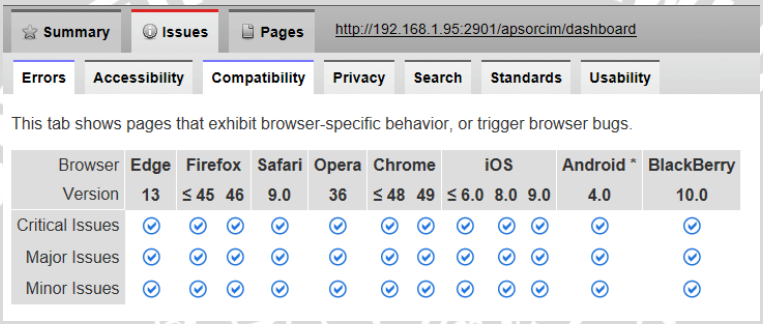
Summary | Issues | Pages | http://192.168.1.95:2901/apsorcim/log_in/login

Errors | Accessibility | Compatibility | Privacy | Search | Standards | Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Edge	Firefox		Safari	Opera	Chrome		iOS			Android *	BlackBerry
Version	13	≤ 45	46	9.0	36	≤ 48	49	≤ 6.0	8.0	9.0	4.0	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

D.2 Halaman Dasbor



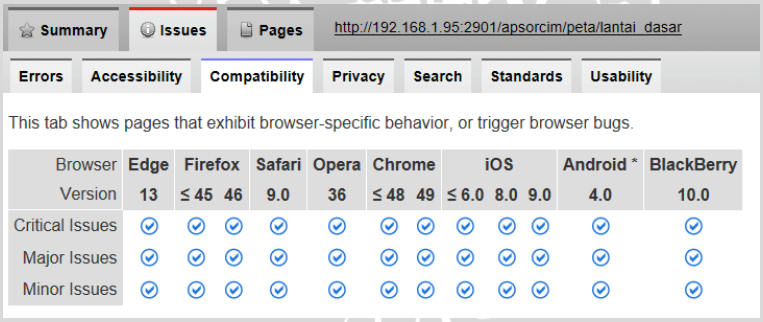
Summary | Issues | Pages | http://192.168.1.95:2901/apsorcim/dashboard

Errors | Accessibility | Compatibility | Privacy | Search | Standards | Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Edge	Firefox		Safari	Opera	Chrome		iOS			Android *	BlackBerry
Version	13	≤ 45	46	9.0	36	≤ 48	49	≤ 6.0	8.0	9.0	4.0	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

D.3 Halaman Peta Lantai Dasar



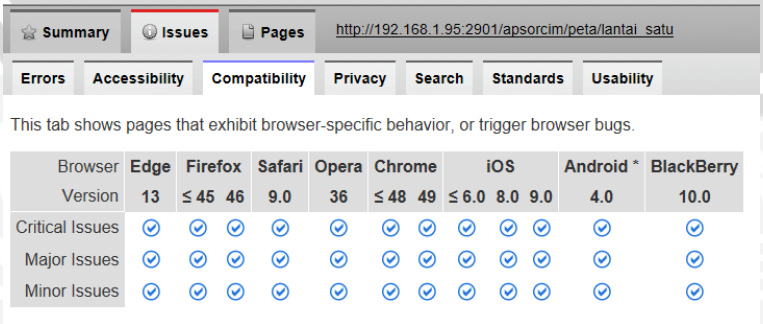
Summary | Issues | Pages | http://192.168.1.95:2901/apsorcim/peta/lantai_dasar

Errors | Accessibility | Compatibility | Privacy | Search | Standards | Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Edge	Firefox		Safari	Opera	Chrome		iOS			Android *	BlackBerry
Version	13	≤ 45	46	9.0	36	≤ 48	49	≤ 6.0	8.0	9.0	4.0	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

D.4 Halaman Peta Lantai Satu



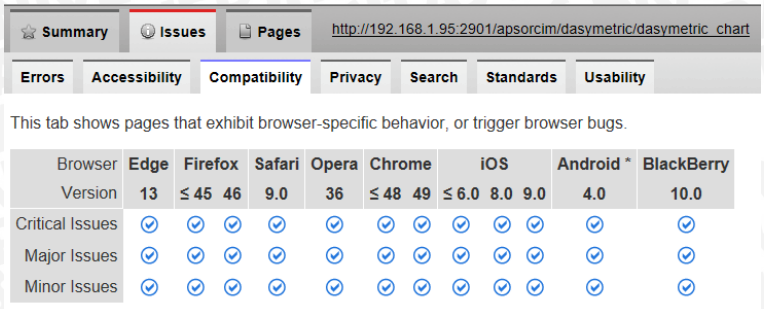
Summary | Issues | Pages | http://192.168.1.95:2901/apsorcim/peta/lantai_satu

Errors | Accessibility | Compatibility | Privacy | Search | Standards | Usability

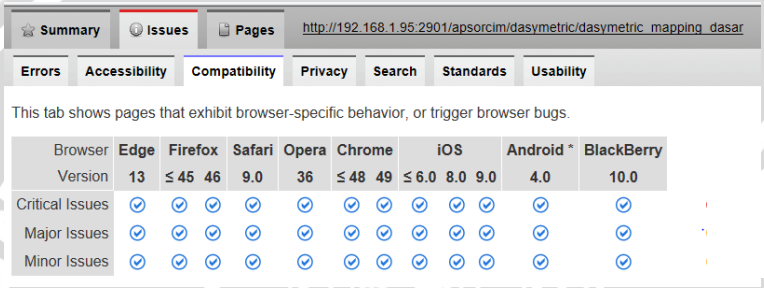
This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Edge	Firefox		Safari	Opera	Chrome		iOS			Android *	BlackBerry
Version	13	≤ 45	46	9.0	36	≤ 48	49	≤ 6.0	8.0	9.0	4.0	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

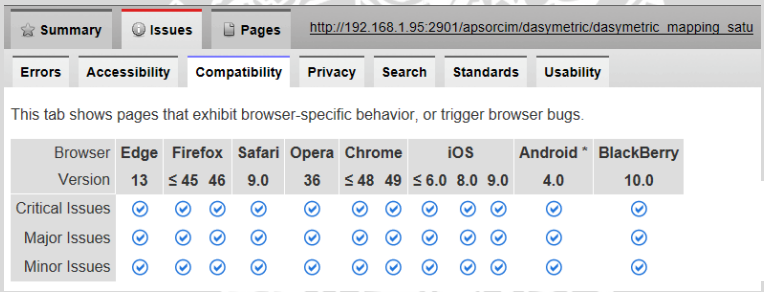
D.5 Halaman Grafik *Dasymetric*



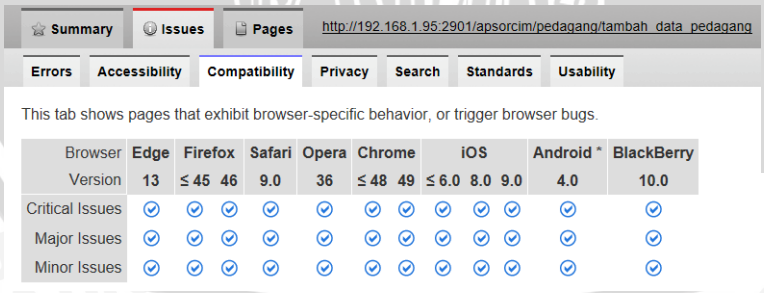
D.6 Halaman *Dasymetric Mapping* Lantai Dasar



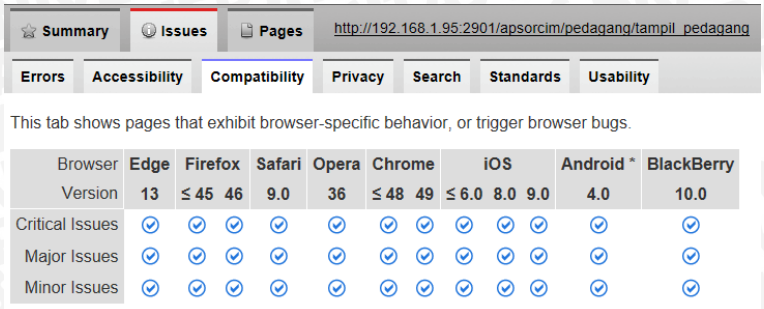
D.7 Halaman *Dasymetric Mapping* Lantai Satu



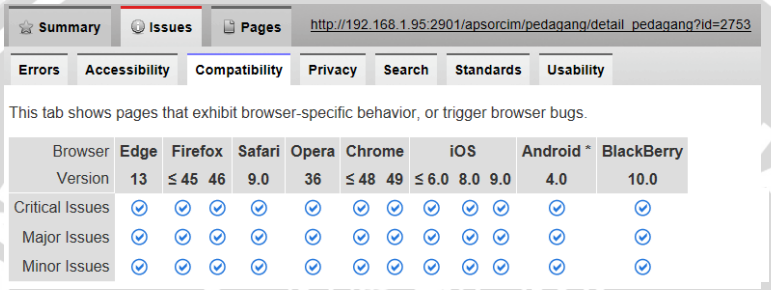
D.8 Halaman Tambah Data Pedagang



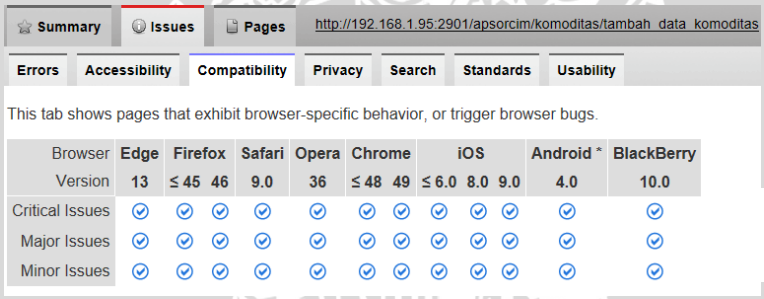
D.9 Halaman Data Pedagang



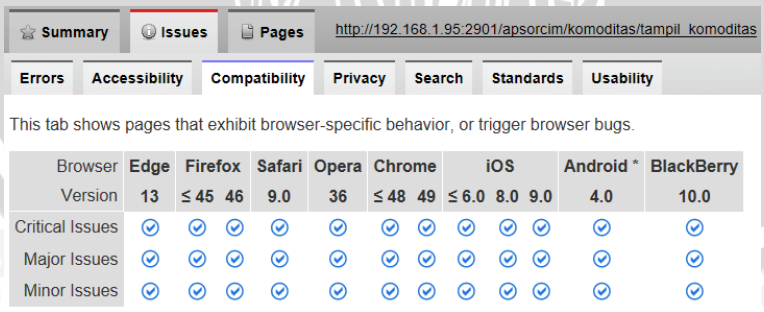
D.10 Halaman Detail Data Pedagang



D.11 Halaman Tambah Data Komoditas



D.12 Halaman Data Komoditas



LAMPIRAN E HASIL WAWANCARA

Hasil Wawancara Terkait Pasar Besar Kota Malang

Hasil wawancara dengan narasumber:

hari/tanggal : Rabu, 04 November 2015

waktu : 13.30 WIB

lokasi : Pasar Besar Kota Malang

1. Bagaimana kondisi operasional di Pasar Besar Kota Malang saat ini?

   Jawab: Pedagang yang menyewa kios akan didata melalui buku pegangan Pasar Besar dan dibuktikan dengan akta sewa milik pedagang sendiri.

2. Apa saja data yang ada di Pasar Besar Kota Malang?

   Jawab: Denah Pasar Besar dan juga buku pedoman milik Pasar Besar Kota Malang.

3. Bagaimana bila Pasar Besar memiliki aplikasi SIG yang dapat memetakan kios di Pasar Besar?

   Jawab: Tentu saja akan sangat memudahkan kami dalam melakukan pendataan serta cek tenggat sewa kios di Pasar Besar.

Rabu, 29 Juni 2016

Narasumber

Ttd

M. MUNIF

NIP. 19740507 200701 1 020

Jabatan: KEPALA PASAR BESAR