

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI GEOGRAFIS
KEPADATAN PENDUDUK DENGAN METODE *HEATMAP***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Lia Hasanah

NIM: 125150400111003



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

ANALISIS DAN PERANCANGAN SISTEM INFORMASI GEOGRAFIS KEPADATAN
PENDUDUK DENGAN METODE *HEATMAP*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Lia Hasanah

NIM: 125150400111003

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, S.T, M.Sc, Ph.D

NIK: 201006 740719 1 001

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

NIK: -

Mengetahui

Ketua Jurusan Sistem Informasi

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

NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 29 Juli 2016



Lia Hasanah

NIM: 125150400111003

KATA PENGANTAR

Penulis mengucapkan puji syukur atas kehadiran Allah SWT karena berkat Rahmat dan Karunia-Nya, penulis dapat menyelesaikan penyusunan skripsi dengan judul “Analisis dan Perancangan Sistem Informasi Geografis Kepadatan Penduduk dengan Metode *Heatmap*”.

Penulisan skripsi ini diajukan untuk memenuhi sebagian syarat untuk memperoleh gelar Sarjana pada program studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya. Melalui kesempatan ini, penulis ingin mengucapkan terima kasih kepada seluruh pihak yang telah berkontribusi dalam memberikan bantuan baik secara langsung maupun tidak langsung dalam pengerjaan skripsi ini. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua saya yang telah mendidik saya, membesarkan serta selalu memberikan dorongan dan nasihat yang baik dalam setiap kesempatan dengan tanpa pamrih dalam memberikan kasih sayangnya.
2. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku dosen pembimbing I yang selalu memberikan bimbingan dan mengarahkan dengan baik dalam penyelesaian skripsi ini tanpa kenal lelah serta memberikan saran-saran yang sangat berarti untuk penulis agar tetap semangat dalam pengerjaan skripsi ini.
3. Bapak Bapak Fatwa Ramdani, D.Sc., S.Si., M.Sc. selaku dosen pembimbing II yang selalu memberikan bimbingan dan mengarahkan dengan baik dalam penyelesaian skripsi ini tanpa kenal lelah serta memberikan saran-saran yang sangat berarti untuk penulis agar tetap semangat dalam pengerjaan skripsi ini.
4. Bapak Suprpto, S.T., M.T. selaku Ketua Program Studi Sistem Informasi yang telah memberikan dukungan dan bantun dalam penyelesaian skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer yang telah memimpin Fakultas Ilmu Komputer dan memberikan dukungan dalam pelaksanaan skripsi ini.
6. Seluruh dosen pengajar Fakultas Ilmu Komputer yang telah mengajarkan dan membagikan ilmu-ilmu yang bermanfaat dari awal hingga akhir masa studi.
7. Pihak Badan Kesatuan Bangsa dan Politik yang telah memberikan bantuan dan ijin rekomendasi dalam pelaksanaan penelitian untuk skripsi ini.
8. Pihak Badan Perencanaan dan Pembangunan Daerah yang telah memberikan bantuan dan dan saran dalam penyelesaian skripsi ini.

9. Pihak Dinas Pekerjaan Umum Perumahan dan Pengawasan Bangunan Kota Malang yang telah memberikan bantuan demi kelancaran proses penyelesaian skripsi ini.
10. Pihak Badan Pelayanan Perizinan Terpadu Kota Malang yang telah memberikan bantuan dalam penyelesaian skripsi ini.
11. Sahabat seperjuangan saya dalam penyelesaian skripsi, Krisdhamara Widayas Pritha, Putri Mutiara Tungga, Diannisa Hakim, Frida Ayundha Putry, Ika Qutsiati Utami, Atikah Hayyu Mulia, Randi Dwi Nandra, Anom Sulton Iskandar, Onny Iman Kandung Iriantono, Rayyan Ramadhan, Maulana Ghulam Hanifa, Mochamad Arizal yang selalu setia dalam membantu, memberika motivasi dan selalu menemani di saat susah maupun senang.
12. Teman-teman seperjuangan dalam penyelesaian skripsi yang telah memberikan saran, motivasi dan ilmu tambahan yang sangat berarti.
13. Keluarga besar Raion Community yang telah memberikan dukungan motivasi dan memberikan banyak pengalaman yang berarti.
14. Teman-teman Sistem Informasi yang telah bersama dari awal masa perkuliahan memberikan pengalaman, dukungan, saran dan motivasi dalam menyelesaikan studi.
15. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dalam proses pengerjaan skripsi ini hingga selesai.

Semoga seluruh kebaikan dan bantuan yang telah dilakukan mendapat balasan yang baik dari Allah SWT. Dokumen skripsi ini masih terdapat kekurangan sehingga kritik serta saran yang membangun penulis harapkan dari para pembaca. Semoga dokumen skripsi ini dapat bermanfaat dan membantu bagi para pembaca.

Malang, 29 Juli 2016

Penulis

liahasanah16@gmail.com

ABSTRAK

Perkotaan dengan jumlah penduduk yang sangat banyak dan kerapatan bangunan yang kecil merupakan kondisi yang sering dijumpai di kota besar. Hal ini menyebabkan pemerintah harus menetapkan aturan-aturan untuk pengendalian pembangunan dan pertumbuhan penduduk di kota sekaligus mengawasi kondisi kepadatan penduduknya agar kondisi kota tersebut tetap stabil. Kepadatan penduduk merupakan sebuah nilai atau persentase terhadap kondisi penduduk yang menempati suatu wilayah dihitung berdasarkan jumlah seluruh penduduk dibagi dengan luas wilayah tersebut. Dengan mengetahui nilai dari kepadatan penduduk ini, pemerintah melakukan perencanaan pembangunan di kota dan jumlah normal penduduknya. Namun ada salah satu masalah yang menyebabkan penduduk dan pembangunan di kota terus berkembang yaitu urbanisasi atau perpindahan penduduk dari desa ke kota. Setiap tahunnya urbanisasi selalu terjadi di kota-kota besar dengan para pendatang baru yang berharap bahwa di kota akan mendapatkan peluang kerja yang lebih baik atau dapat mendirikan suatu usaha yang sukses di masa yang akan datang. Oleh karena itu pemerintah harus sigap dan cepat dalam menanggulangi masalah-masalah yang terjadi di kota seperti urbanisasi ini. Pada penelitian ini dilakukan analisis dan perancangan untuk membangun sebuah sistem yang dapat membantu dalam memahami kondisi kepadatan penduduk di kota. Sampel kota yang digunakan adalah kota Malang. Data penduduk dikumpulkan dan diolah dengan peta bangunan di kota menjadi sebuah peta yang menampilkan variasi jumlah penduduk dengan menggunakan metode *heat map*. Sistem ini dirancang untuk menampilkan representasi visual dari data menggunakan warna dengan perbedaan dari setiap warna berdasarkan nilai pada data tersebut. Hasil dari penelitian ini terdiri dari kondisi proses bisnis saat ini dalam perencanaan tata ruang kota Malang; proses bisnis yang diusulkan dengan memanfaatkan sistem; spesifikasi persyaratan sistem, model-model perancangan sistem; prototipe; hasil evaluasi pada persyaratan sistem yaitu setiap persyaratan dapat dilacak, memiliki nama berbeda serta memiliki kode yang unik; dan hasil evaluasi pada rancangan sistem yang diusulkan, yaitu terdapat beberapa saran dari pengguna berkaitan dengan navigasi dan antarmuka pengguna dari sistem untuk Sistem Informasi Geografis Kepadatan Penduduk.

Kata kunci : *Heat Map*, Kepadatan Penduduk, Sistem Informasi Geografis, Perancangan WebGIS.

ABSTRACT

Urban with many population and a small building density is a condition that is often found in big cities. This matter cause government should set the rules for the control of development and population growth in the city as well as supervise the condition of population density that city remains stable conditions. The population density is a percentage of the value or condition of the people who occupy an area is calculated based on the total population divided by the area of the region. By recognizing the value of the density of population, the government is planning the construction of the city and the amount of the normal population. But there is one problem that cause to the population and development in the city continues to grow, namely urbanization or migration of people from villages to cities. Every year urbanization always happen in big cities with the newcomers who hoped that the city would get better job opportunities or can start a success business in the future. Therefore, the government should be alert and quick in tackling the problems that happen in a town like this urbanization. This research conducted analysis and design to build a system that can help in understanding the conditions of overcrowding in the city. Sample city is the city of Malang. Population data collected and processed with the buildings in the city map into a map showing variations in the number of people using heat map method. The system is designed to displays a visual representation of the data using a color difference of each color based on the value on the data. The results of this study consisted of current business processes in urban spatial planning at Malang city, proposed business processes by utilizing systems, system requirements specification, system design models, prototype design, evaluation results on the system requirements that every requirement can be tracked, has a different name and has a unique code and evaluation results of the proposed system design, there are some suggestions from users about navigation and interface of the system for Geographic Information System Population Density.

Keywords : Heat Map, Population Density, WebGIS, Design of WebGIS.

DAFTAR ISI

HALAMAN JUDUL	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori.....	6
2.2.1 Kepadatan Penduduk.....	6
2.2.2 Sistem Informasi Geografis	7
2.2.3 <i>Heat Map</i>	11
2.2.4 Sistem Informasi.....	13
2.2.5 Pemodelan Proses Bisnis.....	14
2.2.6 Rekayasa Perangkat Lunak.....	14
2.2.7 Pembuatan Visi Bersama	21
2.2.8 Rekayasa Persyaratan Sistem.....	26
2.2.9 Mekanisme Analisis.....	27
2.2.10 Pemodelan Sistem.....	28

2.2.11 Pemodelan Data	35
2.2.12 Analisis Spasial Pada Data Vektor	37
2.2.13 Verifikasi dan Validasi	39
2.2.14 Tinjauan Spesifikasi Persyaratan	39
2.2.15 Evaluasi Prototipe	39
2.2.16 <i>Website Usability Evaluation</i>	40
BAB 3 METODOLOGI	42
3.1 Studi Literatur	42
3.2 Pengumpulan Data	42
3.3 Pengolahan Data	43
3.4 Rekayasa Persyaratan Sistem	43
3.5 Perancangan Sistem dan Prototipe	43
3.6 Evaluasi Persyaratan dan Rancangan Sistem	44
3.7 Dokumentasi dan Kesimpulan	45
BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN	46
4.1 Proses Bisnis	46
4.1.1 Proses Bisnis Saat Ini	46
4.1.2 Proses Bisnis yang Diusulkan	47
4.2 Analisis Permasalahan	49
4.3 Analisis Pemangku Kepentingan dan Pengguna	49
4.3.1 Tipe Pemangku Kepentingan	49
4.3.2 Tipe Pengguna	50
4.4 Kebutuhan Pemangku Kepentingan dan Pengguna	51
4.5 Kedudukan Produk	53
4.6 Fitur-Fitur	53
4.7 Persyaratan secara Deklaratif	54
4.7.1 Persyaratan Fungsional	55
4.7.2 Persyaratan Non Fungsional	56
4.8 Model <i>Use Case</i>	56
4.8.1 Aktor	56
4.8.2 Diagram <i>Use Case</i>	57
4.8.3 Spesifikasi <i>Use Case</i>	58

4.8.4 Diagram Aktivitas	68
BAB 5 PERANCANGAN.....	74
5.1 Kelas-kelas Analisis	74
5.1.1 Kelas Analisis dari Analisis <i>Use case</i>	74
5.1.2 Kelas Analisis dari Mekanisme Analisis	76
5.1.3 Unifikasi Analisis Kelas	76
5.2 Elemen Desain	77
5.2.1 <i>Package</i>	77
5.2.2 Diagram Kelas.....	78
5.3 Pemodelan Interaksi	80
5.3.1 Melihat Grafik Jumlah Penduduk Kecamatan.....	80
5.3.2 Melihat Grafik Jumlah Penduduk Kelurahan	81
5.3.3 Mencari Bangunan	82
5.3.4 Melihat Detail Bangunan	83
5.3.5 Melihat Peta.....	84
5.3.6 Melihat Kepadatan Penduduk Kecamatan	85
5.3.7 Melihat Kepadatan Penduduk Kelurahan	86
5.3.8 Mengubah Data Bangunan	87
5.4 Pemodelan Data.....	89
5.4.1 <i>Entity Relationship Diagram</i>	89
5.4.2 Perancangan Tabel.....	90
5.4.3 <i>Physical Data Model</i>	93
5.5 Perancangan Antarmuka	94
5.5.1 Perancangan Antarmuka Awal.....	94
5.5.2 Perancangan antarmuka untuk menampilkan detail bangunan	96
5.5.3 Perancangan antarmuka untuk menampilkan grafik.....	97
5.5.4 Perancangan antarmuka untuk menampilkan kepadatan penduduk	98
5.5.5 Perancangan antarmuka untuk menampilkan halaman login administrator.....	99
5.5.6 Perancangan antarmuka untuk menampilkan peta ubah data	100
5.5.7 Perancangan antarmuka untuk menampilkan form ubah data	100

5.6 Prototipe Antarmuka	101
5.6.1 Prototipe antarmuka Awal	102
5.6.2 Prototipe antarmuka untuk menampilkan <i>Heat Map</i>	104
5.6.3 Prototipe antarmuka untuk menampilkan detail bangunan	105
5.6.4 Prototipe antarmuka untuk menampilkan grafik	106
5.6.5 Prototipe antarmuka untuk menampilkan kepadatan penduduk	107
5.6.6 Prototipe antarmuka untuk menampilkan halaman login administrator.....	111
5.6.7 Prototipe antarmuka untuk menampilkan peta ubah data.....	111
5.6.8 Prototipe antarmuka untuk menampilkan form ubah data	112
BAB 6 HASIL EVALUASI	113
6.1 Evaluasi <i>Traceability</i>	113
6.2 Evaluasi Prototipe	115
6.2.1 Evaluasi Berdasarkan <i>Use case</i>	115
6.2.2 Evaluasi Berdasarkan Penilaian Pengguna.....	118
BAB 7 PENUTUP	121
7.1 Kesimpulan.....	121
7.2 Saran	122
DAFTAR PUSTAKA.....	123
LAMPIRAN A HASIL EVALUASI PENILAIAN PENGGUNA	126
LAMPIRAN B PETA BANGUNAN	130

DAFTAR TABEL

Tabel 2.1 Tamplate Pernyataan Masalah.....	22
Tabel 2.2 Contoh Penulisan Fitur pada Sistem Telepon Sederhana	24
Tabel 2.3 Tamplate Pernyataan Kedudukan Produk (Bahasa Inggris)	25
Tabel 2.4 Tamplate Pernyataan Kedudukan Produk (Bahasa Indonesia)	25
Tabel 2.5 Contoh Kedudukan Produk pada sistem ATM	25
Tabel 2.6 Komponen yang Bertanggung Jawab dalam OOAD	28
Tabel 2.7 Area utama dalam UML	28
Tabel 2.8 Hubungan Antar <i>Use Case</i>	30
Tabel 2.9 Hubungan Antar Kelas.....	34
Tabel 2.10 Simbol-simbol ER diagram.....	36
Tabel 2.11 Kardinalitas	37
Tabel 2.12 Kategori Inspeksi Heuristik	41
Tabel 4.1 Kode Aktivitas Proses Bisnis Usulan	48
Tabel 4.2 Analisis Permasalahan.....	49
Tabel 4.3 Tipe Pemangku Kepentingan.....	49
Tabel 4.4 Tipe Pengguna	50
Tabel 4.5 Kebutuhan Pemangku Kepentingan dan Pengguna.....	51
Tabel 4.6 Pernyataan Kedudukan Produk.....	53
Tabel 4.7 Fitur-fitur	53
Tabel 4.8 Aturan Penomoran Persyaratan Deskriptif	54
Tabel 4.9 Persyaratan Fungsional	55
Tabel 4.10 Persyaratan Non Fungsional	56
Tabel 4.11 Penjelasan Aktor.....	57
Tabel 4.12 Kode <i>Use Case</i>	58
Tabel 4.13 Spesifikasi <i>Use Case</i> Melihat Grafik Jumlah Penduduk Kecamatan....	58
Tabel 4.14 Spesifikasi <i>Use Case</i> Melihat Grafik Jumlah Penduduk Kelurahan	59
Tabel 4.15 Spesifikasi <i>Use Case</i> Mencari Bangunan jika Masukan Kata Ditemukan	60
Tabel 4.16 Spesifikasi <i>Use case</i> Mencari Bangunan jika Masukan Kata Tidak Ditemukan	60

Tabel 4.17 Spesifikasi <i>Use case</i> Melihat Detail Bangunan	61
Tabel 4.18 Spesifikasi <i>Use case</i> Melihat Peta	62
Tabel 4.19 Spesifikasi <i>Use case</i> Melihat Kepadatan Penduduk Kecamatan	63
Tabel 4.20 Spesifikasi <i>Use case</i> Melihat Kepadatan Penduduk Kelurahan.....	63
Tabel 4.21 Spesifikasi <i>Use case</i> Mengubah Data Bangunan jika <i>password</i> benar dan masukan data sesuai format sistem	64
Tabel 4.22 Spesifikasi <i>Use case</i> Mengubah Data Bangunan jika <i>password</i> salah dan masukan data sesuai format sistem	64
Tabel 4.23 Spesifikasi <i>Use case</i> Mengubah Data Bangunan jika <i>password</i> kosong dan masukan data sesuai format sistem	65
Tabel 4.24 Spesifikasi <i>Use case</i> Mengubah Data Bangunan jika <i>password</i> benar dan Memasukkan data tidak sesuai format.....	66
Tabel 4.25 Spesifikasi <i>Use case</i> Mengubah Data Bangunan jika <i>password</i> benar dan Memasukkan Data kosong.....	67
Tabel 5.1 Kelas Analisis dari Analisis <i>Use case</i>	74
Tabel 5.2 Kelas Analisis dari Mekanisme Analisis	76
Tabel 5.3 Penjelasan Kelas Peta Controller.....	78
Tabel 5.4 Penjelasan Kelas Grafik Controller	79
Tabel 5.5 Penjelasan Kelas Kepadatan Controller	79
Tabel 5.6 Penjelasan Kelas Ubah Data Controller.....	79
Tabel 5.7 Penjelasan Kelas Bangunan.....	79
Tabel 5.8 Penjelasan Kelas Penduduk.....	79
Tabel 5.9 Perancangan Tabel Bangunan.....	90
Tabel 5.10 Perancangan Tabel Kecamatan	90
Tabel 5.11 Perancangan Tabel Kelurahan.....	91
Tabel 5.12 Perancangan Tabel PendKecam	91
Tabel 5.13 Perancangan Tabel PendKelur	92
Tabel 5.14 Perancangan Tabel LokasiBang	92
Tabel 5.15 Keterangan Perancangan Antarmuka Awal	94
Tabel 5.16 Keterangan Perancangan Antarmuka Detail Bangunan.....	96
Tabel 5.17 Keterangan Perancangan Antarmuka Grafik.....	97
Tabel 5.18 Keterangan Perancangan Antarmuka Kepadatan Penduduk.....	98
Tabel 5.19 Keterangan Perancangan Antarmuka Login Administrator	99

Tabel 5.20 Keterangan Perancangan Antarmuka Peta Ubah Data	100
Tabel 5.21 Keterangan Perancangan Antarmuka Form Ubah Data.....	101
Tabel 5.22 Potongan Kode Prototipe Antarmuka Awal	103
Tabel 5.23 Potongan Kode Prototipe Antarmuka Menampilkan <i>Heat Map</i>	104
Tabel 5.24 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Detail Bangunan	105
Tabel 5.25 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kecamatan.....	108
Tabel 5.26 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kelurahan	110
Tabel 5.27 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Peta Ubah Data	112
Tabel 6.1 Keruntutan Aktivitas Proses Bisnis, Fitur dan Persyaratan Fungsional	113
Tabel 6.2 Keruntutan Kebutuhan, Fitur, Persyaratan Fungsional dan <i>Use case</i> ..	114
Tabel 6.3 <i>Traceability</i>	115
Tabel 6.4 Evaluasi <i>Use case</i> Melihat Peta	116
Tabel 6.5 Evaluasi <i>Use case</i> Melihat Grafik Penduduk Kecamatan	116
Tabel 6.6 Evaluasi <i>Use case</i> Melihat Kepadatan Penduduk Kelurahan	116
Tabel 6.7 Evaluasi <i>Use case</i> Mengubah Data Bagian Pertama	117
Tabel 6.8 Evaluasi <i>Use case</i> Mengubah Data Bagian Kedua.....	117
Tabel 6.9 Evaluasi <i>Use case</i> Mengubah Data Bagian Ketiga.....	118
Tabel 6.10 Evaluasi Terhadap Harapan Alur Pengguna	119
Tabel 6.11 Evaluasi Terhadap Navigasi	119
Tabel 6.12 Evaluasi Terhadap Antarmuka	120

DAFTAR GAMBAR

Gambar 2.1 Rumus Kepadatan Penduduk Aritmatik	7
Gambar 2.2 Rumus kepadatan penduduk agraris	7
Gambar 2.3 Penelitian tentang Kolera oleh Dr. John Snow	8
Gambar 2.4 Ilustrasi data vektor	9
Gambar 2.5 Kontras model data raster dengan vektor	10
Gambar 2.6 Derajat Pada Garis Lintang dan Bujur	11
Gambar 2.7 Contoh <i>Heat Map</i>	12
Gambar 2.8 Contoh Penggunaan Warna Pada Heat Map	13
Gambar 2.9 Contoh Pemodelan Proses Bisnis	14
Gambar 2.10 Model <i>Waterfall</i>	15
Gambar 2.11 Pengembangan Berulang Pada RUP	17
Gambar 2.12 Dua Dimensi Pada RUP	18
Gambar 2.13 Milestone untuk Fase Siklus Hidup RUP	18
Gambar 2.14 Peran, Aktivitas dan Artefak	20
Gambar 2.15 Kerangka Kerja Proses RUP	21
Gambar 2.16 Kebutuhan dari Masalah Pemangku Kepentingan	22
Gambar 2.17 Hubungan Antara Kebutuhan, Fitur dan Sistem	24
Gambar 2.18 Masalah dan Solusinya	26
Gambar 2.19 Analisis Persyaratan	26
Gambar 2.20 Contoh <i>Use Case</i>	30
Gambar 2.21 Contoh diagram aktifitas	31
Gambar 2.22 Contoh diagram aktifitas dengan partisi	32
Gambar 2.23 Contoh Diagram Sequence	33
Gambar 2.24 Contoh Diagram Kelas	33
Gambar 2.25 Contoh Detail Kelas	34
Gambar 2.26 <i>Aggregation</i> dan <i>Composition</i>	35
Gambar 2.27 Contoh ER diagram	35
Gambar 2.28 Contoh <i>physical data model</i>	37
Gambar 2.29 Contoh Proses Digitalisasi	38
Gambar 3.1 Metodologi Penelitian	42

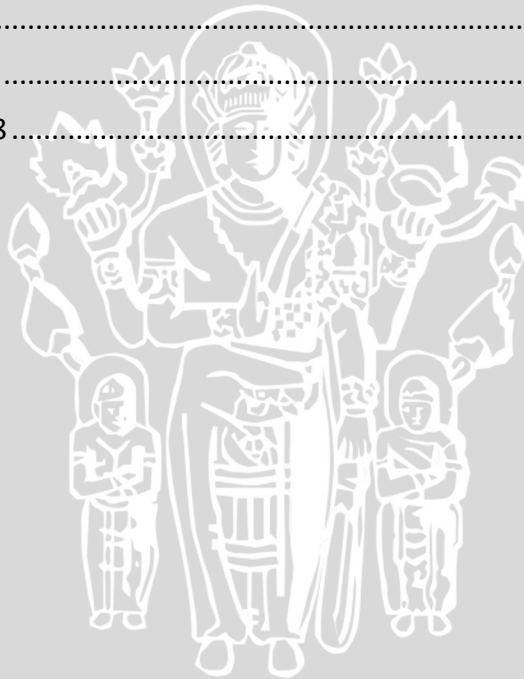
Gambar 4.1 Proses Bisnis Saat Ini	46
Gambar 4.2 Proses Bisnis Yang Diusulkan	47
Gambar 4.3 Diagram <i>Use Case</i>	57
Gambar 4.4 Diagram Aktivitas Melihat Grafik Jumlah Penduduk Kecamatan	68
Gambar 4.5 Diagram Aktivitas Melihat Grafik Jumlah Penduduk Kelurahan	69
Gambar 4.6 Diagram Aktivitas Mencari Bangunan	69
Gambar 4.7 Diagram Aktivitas Melihat Detail Bangunan	70
Gambar 4.8 Diagram Aktivitas Melihat Peta	71
Gambar 4.9 Diagram Aktivitas Melihat Kepadatan Penduduk Kecamatan	72
Gambar 4.10 Diagram Aktivitas Melihat Kepadatan Penduduk Kelurahan	72
Gambar 4.11 Diagram Aktivitas Mengubah Data Bagunan	73
Gambar 5.1 Unifikasi Analisis Kelas	76
Gambar 5.2 Package	77
Gambar 5.3 Diagram Kelas	78
Gambar 5.4 Diagram Interaksi Melihat Grafik Jumlah Penduduk Kecamatan	80
Gambar 5.5 Diagram Interaksi Melihat Grafik Jumlah Penduduk Kelurahan	81
Gambar 5.6 Diagram Interaksi Mencari Bangunan	82
Gambar 5.7 Diagram Interaksi Melihat Detail Bangunan	83
Gambar 5.8 Diagram Interaksi Melihat Peta	84
Gambar 5.9 Diagram Interaksi Melihat Kepadatan Penduduk Kecamatan	85
Gambar 5.10 Diagram Interaksi Melihat Kepadatan Penduduk Kelurahan	86
Gambar 5.11 Diagram Interaksi Mengubah Data Bangunan	87
Gambar 5.12 <i>Entity Relationship Diagram</i>	89
Gambar 5.13 <i>Physical Data Model</i>	93
Gambar 5.14 Perancangan Antarmuka Awal	94
Gambar 5.15 Perancangan Antarmuka Detail Bangunan	96
Gambar 5.16 Perancangan Antarmuka Grafik	97
Gambar 5.17 Perancangan Antarmuka Kepadatan Penduduk	98
Gambar 5.18 Perancangan Antarmuka Login Administrator	99
Gambar 5.19 Perancangan Antarmuka Peta Ubah Data	100
Gambar 5.20 Perancangan antarmuka untuk menampilkan form ubah data	100
Gambar 5.21 Prototipe Antarmuka Awal	102

Gambar 5.22 Prototipe Antarmuka Untuk Menampilkan <i>Heat Map</i>	104
Gambar 5.23 Prototipe Antarmuka Untuk Menampilkan Detail Bangunan.....	105
Gambar 5.24 Prototipe Antarmuka Untuk Menampilkan Grafik Tiap Kecamatan	106
Gambar 5.25 Prototipe Antarmuka Untuk Menampilkan Grafik Tiap Kelurahan	106
Gambar 5.26 Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kecamatan.....	107
Gambar 5.27 Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kelurahan	109
Gambar 5.28 Prototipe Antarmuka Untuk Menampilkan Halaman Login Administrator.....	111
Gambar 5.29 Prototipe Antarmuka Untuk Menampilkan Peta Ubah Data.....	111
Gambar 5.30 Prototipe Antarmuka Untuk Menampilkan Form Ubah Data.....	112



DAFTAR LAMPIRAN

LAMPIRAN A HASIL EVALUASI PENILAIAN PENGGUNA	126
A.1 Staf Bagian Tata Kota Bappeda	126
A.2 Masyarakat Umum Kota Malang.....	128
LAMPIRAN B PETA BANGUNAN	130
B.1 Aturan Indeks Lembar Peta	130
B.2 Indeks J-09	131
B.3 Indeks K-09	132
B.4 Indeks K-10	133
B.5 Indeks L-08.....	134
B.6 Indeks L-09.....	135
B.7 Indeks M07	136
B.8 Indeks M-08	137



BAB 1 PENDAHULUAN

1.1 Latar belakang

Kondisi perkotaan yang padat akan aktivitas masyarakat di dalamnya menyebabkan pembangunan untuk usaha dan pemukiman terus bertambah sehingga kerapatan antar bangunan semakin kecil. Oleh karena itu aturan untuk pembangunan harus diperhatikan sehingga gedung dan pemukiman dapat dibangun secara merata dan tidak menyebabkan kerugian di masa yang akan datang. Aturan pembangunan untuk gedung dan pemukiman ini sudah ditetapkan dalam Peraturan Daerah Nomor 4 tahun 2011 tentang Rencana Tata Ruang Wilayah Kota Malang Tahun 2010-2030. Peraturan Daerah ini menjelaskan aturan untuk pembangunan yang dilakukan oleh masyarakat seperti pembangunan pemukiman atau perumahan, perdagangan, perkantoran dan industri serta lokasi kawasan yang diperbolehkan untuk dibangun dari masing-masing pembangunan tersebut. Perumahan yang dibangun dibedakan menjadi 3 kategori yakni perumahan dengan kepadatan tinggi, sedang dan rendah. Setiap tingkat kepadatan perumahan tersebut memiliki aturan pembangunan yang berbeda seperti luas maksimal bangunannya dan tinggi bangunan atau banyaknya lantai sehingga area resapan air dapat ditampung dengan baik. Hal ini menyebabkan pihak yang bertanggung jawab yaitu pemerintah daerah harus memahami dengan baik bagaimana kondisi pembangunan di kota.

Padatnya kondisi perkotaan juga disebabkan seringnya urbanisasi atau perpindahan penduduk dari desa ke kota terjadi di suatu daerah. Urbanisasi ini dapat menyebabkan area kota dan pembangunan semakin padat karena banyaknya penduduk yang masuk ke kota untuk mencari pekerjaan atau membangun usaha. Hal ini dapat mempengaruhi tingkat kesehatan yang menimbulkan masalah kesehatan seperti penyebaran penyakit (Syarifah, 2014). Masalah selanjutnya yang terkait dengan urbanisasi adalah berkurangnya sektor pertanian yaitu menurunnya jumlah petani hingga sekitar 8 juta jiwa (Deil, 2014). Perusahaan konsultasi dan analisa global McKinsey Global Institute (MGI) memprediksi bahwa penduduk Indonesia yang tinggal di area perkotaan akan mencapai 71% di tahun 2030 dikarenakan banyaknya masyarakat yang berminat untuk mencari peluang pekerjaan yang lebih baik dan pendapatan yang lebih tinggi (Deil, 2014). Saat ini penduduk Indonesia yang tinggal di kota mencapai 54% dari total penduduk sehingga penduduk di desa lebih sedikit daripada penduduk di kota (Kiswondari, 2015). Oleh karena itu pemerintah harus cepat dalam melakukan pengendalian pada jumlah penduduk. Sebelum itu pemerintah harus memahami dengan baik bagaimana kondisi nyata saat ini yaitu dengan menganalisis kepadatan penduduk khususnya di wilayah perkotaan.

Pemerintah daerah sebagai pihak yang bertanggung jawab untuk mengatur pembangunan di perkotaan harus mampu menganalisis bagaimana merencanakan pembangunan di kota dengan baik. Pada proses analisis ini dibutuhkan data-data yang telah disimpan oleh pemerintah seperti data

pembangunan dan penduduk. Data pembangunan didapatkan dari pendaftaran yang dilakukan oleh masyarakat yang akan membangun suatu gedung usaha, perkantoran, pemukiman, atau industri dan data penduduk didapatkan dari sensus penduduk yang dilakukan oleh pemerintah yaitu Badan Pusat Statistik secara berkala maksimal 10 tahun sekali. Pelaksanaan sensus penduduk ini sudah diatur dalam Undang Undang Republik Indonesia Nomor 6 Tahun 1960 tentang Sensus dan Peraturan Pemerintah Republik Indonesia Nomor 21 Tahun 1979 Tentang Pelaksanaan Sensus Penduduk. Metodologi pelaksanaan sensus terdiri dari sensus penduduk dan sensus perumahan yang terdiri dari pendaftaran seluruh bangunan serta rumah tangga sehingga dapat mengidentifikasi kondisi kepadatan penduduk pada suatu wilayah dengan melakukan pendataan penduduk tiap bangunannya menurut Pedoman Pencacah Sensus Penduduk 2000.

Proses analisis kepadatan penduduk ini membutuhkan media atau sistem yang membantu pelaksanaan analisis tersebut supaya dapat dikerjakan dengan cepat dan hasil yang tepat akurat. Namun pemerintah khususnya pemerintah kota Malang masih belum menerapkan metode yang dapat mempercepat proses pelaksanaan analisis kepadatan penduduk karena masih mengandalkan perhitungan dari data berupa tabel yang berasal dari data sensus penduduk (BPS, 2014). Salah satu metode yang dapat membantu proses analisis tersebut adalah dengan menerapkan data tersebut ke dalam visualisasi berupa peta. Dalam visualisasi peta ini dapat memuat banyak informasi seperti informasi lokasi, data bangunan dan penduduk dalam lokasi tersebut. Visualisasi peta ini diterapkan dalam sistem informasi geografis yang memiliki kelebihan yakni cepat dan mudah untuk dipahami (Pramartha, 2012). Sistem informasi geografis sering digunakan untuk membantu proses penelitian seperti menilai kualitas ruang bawah tanah suatu perkotaan untuk perencanaan kota yang menghasilkan visualisasi kondisi tanah dan batuan di dalamnya (Hou, et al., 2015). Penelitian tersebut menggunakan salah satu metode dalam sistem informasi geografis yaitu visualisasi berupa *heat map* yang menampilkan hasil analisis kualitasnya berupa peta dengan perbedaan tingkat frekuensi warna menurut nilai parameter yang digunakan dalam penelitian tersebut menggunakan parameter kualitas tanah dan batuan. Adapun penelitian lain yang menggunakan metode *heat map* adalah penelitian tentang pemetaan kepadatan penduduk dengan memperkirakan kepadatan penduduk dari gambar polarimetrik dari SAR dan membandingkannya dengan pola kepadatan perkotaan kota-kota besar di dunia (Susaki, et al., 2014). Jadi, dari kedua penelitian tersebut bahwa penggunaan metode *heat map* merupakan metode yang dapat digunakan untuk penelitian di berbagai bidang dan bertujuan untuk memudahkan dalam proses analisis di dalamnya.

Metode *heat map* dapat diterapkan untuk analisis kepadatan penduduk suatu perkotaan dengan menggunakan data sensus penduduk, sensus perumahan dan peta daerah perkotaan khususnya di kota Malang. Oleh karena itu dibutuhkan suatu sistem yang dapat menerapkan metode tersebut yaitu sistem informasi geografis (WebGIS) untuk kepadatan penduduk. WebGIS merupakan sistem informasi geografis berbasis web yang memanfaatkan

jaringan internet (*online*) sebagai media untuk memberikan informasi spasial bereferensi keruangan untuk mengelola suatu data. Penelitian ini memodelkan proses bisnis yang berhubungan, menganalisis persyaratan dan melakukan perancangan Sistem Informasi Geografis Kepadatan Penduduk (SIGEKA).

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, terdapat beberapa rumusan masalah yang akan digunakan dalam analisis dan perancangan SIGEKA yang terdiri dari :

1. Bagaimanakah model proses bisnis yang menggambarkan kondisi saat ini dalam perencanaan tata ruang kota Malang dan proses bisnis usulan untuk keperluan tersebut yang memanfaatkan SIGEKA?
2. Bagaimanakah spesifikasi persyaratan dari SIGEKA?
3. Bagaimanakah rancangan dan prototipe yang sesuai dengan persyaratan SIGEKA?
4. Bagaimanakah hasil evaluasi dari persyaratan dan rancangan SIGEKA?

1.3 Tujuan

Tujuan dari tugas akhir ini adalah :

1. Membuat model proses bisnis yang menggambarkan kondisi saat ini dalam perencanaan tata kota Malang dan proses bisnis usulan untuk keperluan tersebut yang memanfaatkan SIGEKA.
2. Menganalisis persyaratan dan menyusun spesifikasi persyaratan dari SIGEKA.
3. Membuat rancangan dan prototipe yang sesuai dengan persyaratan dari SIGEKA.
4. Mengevaluasi persyaratan dan rancangan SIGEKA.

1.4 Manfaat

Manfaat yang diharapkan dari pembuatan laporan skripsi ini adalah sebagai berikut :

1. Membantu pemerintah khususnya Badan Perencanaan Pembangunan Daerah untuk membangun suatu sistem informasi guna mengetahui kondisi dari kepadatan penduduk di kota Malang.
2. Menghasilkan suatu analisis kebutuhan dan perancangan yang dapat digunakan untuk membangun suatu sistem informasi yang dapat menjelaskan kondisi dari kepadatan penduduk di perkotaan khususnya di kota Malang.
3. Membantu masyarakat kota Malang untuk mengetahui tentang kondisi kepadatan penduduk di beberapa area di kota Malang.

1.5 Batasan masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Data sensus yang diambil adalah sensus perumahan atau sensus bangunan.
2. Penelitian yang dilakukan berupa tahap analisis persyaratan dan perancangan untuk pembangunan WebGIS kepadatan penduduk.
3. Perancangan yang dilakukan merupakan perancangan awal yang sebaiknya dilanjutkan dengan perancangan detail di penelitian berikutnya sebelum memulai tahap implementasi.
4. Data jumlah penduduk berasal dari pemerintah Kota Malang dan sensus tahun 2014.
5. Sampel yang digunakan untuk data bangunan sebanyak 1000 bangunan di sekitar Jalan Soekarno-Hatta kota Malang.
6. Grafik penduduk yang ditampilkan pada prototipe terdiri dari data jumlah penduduk berdasarkan jenis kelamin dan usia.
7. Detail detail bangunan terdiri dari alamat, fungsi bangunan dan jumlah penduduk setiap bangunan.
8. Data detail bangunan pada prototipe merupakan data yang dibuat oleh peneliti, bukan data sebenarnya.
9. Data kepadatan penduduk kelurahan pada prototipe adalah pada Kecamatan Lowokwaru.
10. Fungsi bangunan yang ditampilkan terdiri dari fungsi untuk bangunan hunian, industri, jasa, kesehatan, pemerintahan, pendidikan, perdagangan dan wisata.

1.6 Sistematika pembahasan

Sistematika penulisan pada laporan skripsi ini terdiri dari Bab Pendahuluan, Landasan Kepustakaan, Metodologi, Analisis dan Perancanganserta Penutup.

Bab Pendahuluan berisi tentang uraian umum yang terdiri dari Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah dan Sistematika Pembahasan.

Bab Landasan Kepustakaan menjelaskan tentang uraian dari teori-teori kepustakaan yang mendukung proses analisis dan perancangan untuk membangun sistem informasi geografis kepadatan penduduk.

Bab Metodologi menguraikan tentang langkah-langkah untuk melakukan analisis, kemudian perancangan yang akan digunakan untuk membangun sistem informasi geografis kepadatan penduduk.

Bab Pemodelan Proses Bisnis dan Analisis Persyaratan berisi tentang kondisi proses bisnis saat ini dalam perencanaan tata ruang kota Malang, proses bisnis

usulan dengan memanfaatkan SIGEKA, analisis permasalahan, analisis pemangku kepentingan dan pengguna, kebutuhan pemangku kepentingan dan pengguna, kedudukan produk, fitur-fitur, persyaratan secara deklaratif, model *use case*, spesifikasi *use case* dan diagram aktivitas.

Bab Perancangan menguraikan tentang model-model untuk rancangan sistem yang terdiri dari kelas-kelas analisis, elemen desain, pemodelan interaksi, pemodelan data, perancangan antarmuka dan prototipe antarmuka.

Bab Hasil Evaluasi berisi tentang hasil dari evaluasi pada persyaratan dan rancangan sistem yang terdiri dari evaluasi *traceability* dan evaluasi prototipe. Evaluasi prototipe dibagi lagi berdasarkan *use case* dan penilaian pengguna.

Bab Penutup menguraikan tentang kesimpulan dan saran untuk penelitian yang serupa selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tinjauan pustaka yang digunakan peneliti terdiri dari penelitian-penelitian terdahulu yang menggunakan metode *heat map* untuk membantu merepresentasikan hasil dari penelitian tersebut. Peneliti mengambil tiga contoh penelitian terdahulu yang menggunakan metode *heat map* dan penggunaan metode tersebut dalam analisis kondisi penduduk pada suatu daerah.

Penelitian pertama berjudul *Assessing quality of urban underground spaces by coupling 3D geological models : The case study of Foshan city, South China* oleh Weisheng Hou dan anggota penelitiannya (Hou, et al., 2015) yang meneliti tentang kondisi tanah dan batuan di ruang bawah tanah untuk menilai kualitas ruang bawah tanahnya. Studi kasus pada penelitian ini adalah di kota Foshan, China bagian selatan. Penelitian tersebut menggunakan salah satu metode dalam sistem informasi geografis yaitu visualisasi berupa *heat map* yang menampilkan hasil analisis kualitasnya berupa peta dengan perbedaan tingkat frekuensi warna menurut nilai parameter yang digunakan dalam penelitian tersebut menggunakan parameter kualitas tanah dan batuan. Kemudian terdapat penelitian lain yang menggunakan metode *heat map* dengan mengangkat topik untuk meneliti kondisi kependudukan pada suatu kota dengan judul *Urban density mapping of global megacities from polarimetric SAR images* oleh Junichi Susaki dan anggota penelitiannya (Susaki, et al., 2014). Pada penelitian ini dilakukan pemetaan kepadatan penduduk yang memperkirakan kepadatan penduduk dari gambar polarimetrik dari SAR dan membandingkannya dengan pola kepadatan perkotaan kota-kota besar di dunia. Penelitian ini menghasilkan *heat map* yang menampilkan kondisi kepadatan kota besar seperti *New York, Tokyo, Kyoto* atau *Munich*. Jadi penggunaan metode *heat map* dapat diterapkan pada beberapa topik penelitian yang berbeda seperti meneliti tentang kondisi kepadatan penduduk suatu perkotaan.

2.2 Dasar Teori

2.2.1 Kepadatan Penduduk

Definisi dari kepadatan penduduk adalah suatu perbandingan dari jumlah penduduk dibagi dengan luas wilayah yang ditinggali oleh penduduk tersebut. Kepadatan penduduk dibedakan menjadi tiga yaitu kepadatan penduduk aritmatik, agraris dan ekonomis. Pengertian dari kepadatan penduduk aritmatik adalah rata-rata dari jumlah penduduk yang tinggal di suatu wilayah dengan luas wilayahnya 1 km^2 . Kemudian untuk agraris merupakan rata-rata jumlah penduduk yang bekerja sebagai petani tiap satuan luas lahan pertanian. Ketiga, kepadatan penduduk ekonomis merupakan rata-rata dari jumlah penduduk dengan luas lahan dalam kapasitas produksi yang dihasilkan. Berikut rumus-rumus untuk menghitung kepadatan penduduk yang disebutkan.

a. Kepadatan penduduk aritmatik

$$\text{Kepadatan Penduduk} = \frac{\text{Jumlah Penduduk}}{\text{Luas Wilayah (km}^2\text{)}}$$

Persamaan 1

Gambar 2.1 Rumus Kepadatan Penduduk Aritmatik

Sumber: Kemendikbud (2016)

b. Kepadatan penduduk agraris

$$\text{Kepadatan Penduduk} = \frac{\text{Jumlah Penduduk Agraris (Petani)}}{\text{Luas Lahan Pertanian (km}^2\text{)}}$$

Persamaan 2

Gambar 2.2 Rumus kepadatan penduduk agraris

Sumber: Kemendikbud (2016)

Faktor-faktor yang menyebabkan kepadatan penduduk yaitu faktor kelahiran, iklim dan tempat strategis, ekonomi dan sosial. Diantara keempat faktor tersebut, salah satu yang paling berpengaruh adalah faktor kelahiran karena paling berpengaruh terhadap laju pertumbuhan penduduk (Sugiharyanto, 2007). Faktor lain yang juga menyebabkan bertambahnya penduduk pada suatu wilayah seperti wilayah perkotaan adalah urbanisasi (Kiswondari, 2015). Urbanisasi atau perpindahan penduduk dari desa ke kota sudah sering terjadi pada kota-kota besar khususnya di pulau Jawa. Hal ini dikarenakan banyaknya penduduk desa yang beranggapan bahwa di kota akan mendapatkan peluang kerja yang lebih baik beserta gaji kerjanya juga.

Kemudian untuk mengatasi dari permasalahan kepadatan penduduk adalah dengan melakukan pengendalian angka kelahiran, melakukan pemindahan penduduk dari wilayah yang padat akan penduduk menuju wilayah yang kurang penduduk dan melakukan pemerataan lapangan pekerjaan. Pemerataan ini dilakukan dengan mengembangkan industri, pertanian, perkebunan, pertambangan dan perikanan pada wilayah yang lain (Sugiharyanto, 2007).

2.2.2 Sistem Informasi Geografis

Sistem informasi geografis atau *Geographic Information System* (GIS) merupakan sistem informasi yang memberikan informasi berupa informasi spasial bereferensi keruangan untuk mengelola suatu data misalnya data yang diidentifikasi menurut lokasinya dalam sebuah basis data. Perkembangan teknologi sistem informasi geografis dapat digunakan untuk membantu dalam investigasi ilmiah, perencanaan pembangunan suatu wilayah, pengelolaan sumber daya manusia dan alam serta merencanakan suatu rute yang paling efektif. Meskipun teknologi sistem informasi geografis sudah ada sejak tahun 1970-an namun sistem ini belum diterapkan berupa komputerisasi. Seiring perkembangan teknologi, penerapan komputerisasi pada sistem informasi

geografis dapat dilakukan guna menghemat proses penggambaran dengan tangan yang membutuhkan waktu, biaya, pelatihan dan energi karena dengan komputer memproses angka dan persamaan matematika jauh lebih cepat daripada dilakukan manual oleh manusia sendiri (Fang, et al., 2014).

1. Sejarah Sistem Informasi Geografis

Sebelum adanya penerapan komputerisasi, pada GIS manual peta dasar atau peta utama akan dicetak pada suatu kertas dan ditempatkan di atas meja kemudian menempatkan peta lain yang lebih transparan untuk mencari hubungan antara peta dasar dan fitur pada peta tansparan diatasnya. Hal ini sering menyebabkan peneliti pada waktu itu harus menyalin ulang peta-peta tersebut yang menghabiskan banyak waktu dan tenaga sehingga penelitian akan berlangsung lama. Namun ada suatu penelitian yang berhasil di waktu yang tepat yaitu penelitian tentang Kolera oleh Dr. John Snow.



Gambar 2.3 Penelitian tentang Kolera oleh Dr. John Snow

Sumber: Yiping Fang, Vivek Shandes, Eugenio A. Cordero (2014)

Pada tahun 1840-an wabah kolera membunuh ratusan penduduk di bagian Soho London, Inggris. Dr. John Snow adalah seorang dokter yang tinggal di daerah dengan korban kolera di sekitarnya. Kemudian Dr. Snow menggambar kejadian-kejadian kematian tersebut pada peta dan menarik lokasi tersebut dengan lokasi sumber air yaitu sumur air. Masyarakat di kota tersebut akan pergi

dengan jarak yang jauh untuk membeli suatu produk yang murah, namun tidak untuk membeli air karena mereka akan lebih memilih untuk mengambil air di sumur terdekat yang gratis dan tidak berat untuk dibawa. Oleh karena itu dokter tersebut melakukan penelitian dengan mengumpulkan titik-titik kejadian kematian yang disebabkan kolera dan lokasi sumber air (Gambar 2.3). Kemudian Snow dan murid-muridnya membuat suatu pompa air yang diletakkan pada sumber air yang bersih. Akhirnya kasus kematian terhadap penyakit kolera berangsur menurun (Fang, et al., 2014).

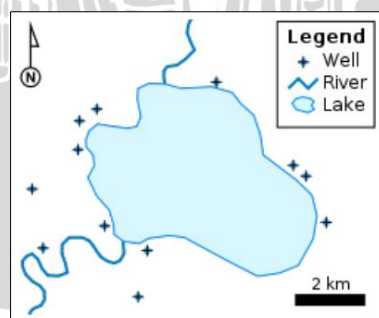
Munculnya komputer untuk mengolah data GIS sampai dapat digunakan oleh individu atau organisasi melewati beberapa tahapan yaitu yang awalnya hanya digunakan oleh para pejabat pemerintah dan beberapa akademisi. Kemudian muncul teknologi komputer dari mikrokomputer yang tidak memiliki kapasitas penyimpanan yang cukup menjadi teknologi baru yang mampu menyimpan dan memproses dataset besar yang dapat digunakan oleh banyak akademisi, instansi pemerintah, organisasi dan usaha kecil hingga besar untuk menggunakan GIS tersebut. Namun untuk mengoperasikan GIS pada suatu komputer dibutuhkan pengguna yang terampil dan sudah terlatih (Fang, et al., 2014).

2. Model Data GIS

Model data utama pada GIS ada dua yaitu model data vektor dan model data raster.

a. Model data vektor

Model data vektor menggunakan poin-poin atau titik-titik dan hubungan antara poin-poin (x dan y) tersebut yang sering disebut sebagai titik koordinat yang nantinya akan digambarkan pada sebuah peta. Informasi spasial dan informasi atribut pada model ini diidentifikasi melalui nomor-nomor sederhana yang diberikan pada setiap fitur di peta. Ada tiga jenis vektor dasar yaitu titik, garis dan poligon yang masing-masing diilustrasikan pada gambar 2.4 berikut.



Gambar 2.4 Ilustrasi data vektor

Sumber: Yiping Fang, Vivek Shandes, Eugenio A. Cordero (2014)

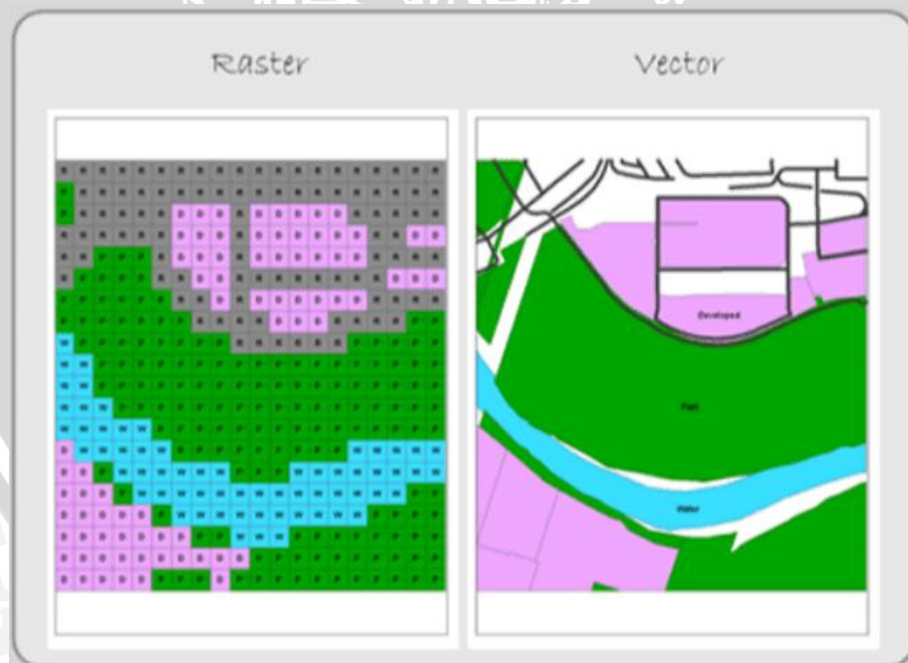
- Titik adalah objek dengan 0 dimensi yang hanya berisi sepasang koordinat. Poin sering digunakan untuk merepresentasikan bangunan, sumur, tiang listrik, lokasi sampel dan sebagainya. Jadi

titik merupakan fitur yang berdiri sendiri dari pertemuan antara titik X dan Y yang ditarik garis lurus.

- Garis adalah objek satu dimensi berupa titik-titik yang dihubungkan. Garis digunakan untuk representasi dari jalan, sungai, batas-batas dan sebagainya. Garis memiliki atribut yaitu panjang garis itu sendiri.
- Poligon adalah objek dua dimensi yang digambarkan dari beberapa garis yang menciptakan bentuk tertutup. Poligon merepresentasikan batas-batas kota, danau, komunitas vegetasi dan sebagainya.

b. Model data raster

Model data raster terdiri dari baris dan kolom piksel yang berukuran sama yang saling berhubungan untuk membentuk sebuah permukaan planar. Kumpulan piksel ini digunakan sebagai blok bangunan untuk membuat titik, garis, bidang, jaringan, dan permukaan. Dengan demikian, sebagian besar data raster yang tersedia yang dibangun pada kumpulan beberapa pixel berbentuk persegi. Kontras antara model data raster dengan vektor ini dapat digunakan untuk membuat titik, garis dan poligon ke dalam bentuk data vektor. Kontras antara data raster dan vektor dapat dilihat pada gambar 2.5.

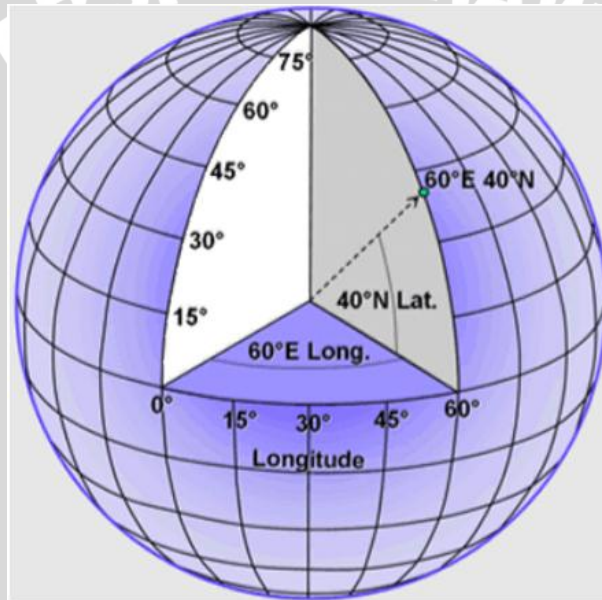


Gambar 2.5 Kontras model data raster dengan vektor

Sumber: Yiping Fang, Vivek Shandes, Eugenio A. Cordero (2014)

3. Sistem Koordinat dan Proyeksi Data GIS

Sistem koordinat digunakan untuk menemukan segala sesuatu pada permukaan bumi dalam bentuk area titik x dan y. Metode yang digunakan untuk menggambarkan bagian bumi yang berbentuk bulat ke dalam permukaan datar seperti pada peta kertas atau layar komputer disebut dengan proyeksi peta. Setiap proyeksi peta yang digunakan pada peta kertas dikaitkan dengan sistem koordinat. Untuk mempermudah penggunaan peta dan menghindari penentuan lokasi pada garis referensi yang melengkung digunakan gambar kotak persegi panjang pada peta seperti barisan grid yang dibentuk dari gabungan koordinat x dan y. Sistem koordinat ini memiliki fitur yang direferensikan oleh garis lintang dan bujur dalam derajat dari pusat bumi ke titik di permukaan bumi atau biasa disebut dengan garis ekuator (lihat gambar 2.6).



Gambar 2.6 Derajat Pada Garis Lintang dan Bujur

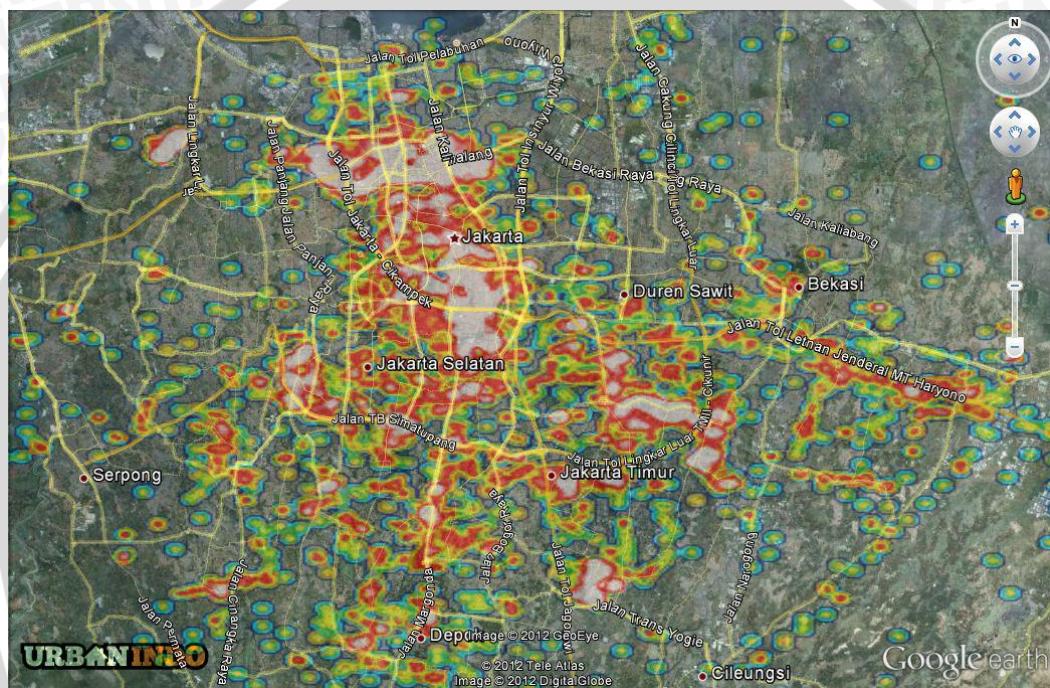
Sumber: Yiping Fang, Vivek Shandes, Eugenio A. Cordero (2014)

2.2.3 Heat Map

Istilah *heat map* awalnya merupakan sebuah merk dagang yang diciptakan oleh perusahaan desainer perangkat lunak *Cormac Kinney* pada tahun 1991 untuk menggambarkan suatu tampilan dua dimensi (2D) dengan menggambarkan kondisi *real time* tentang informasi keuangan. Ada berbagai jenis *heat map*, yaitu *heat map* untuk menampilkan area pada halaman web yang sering dibuka oleh pengguna, digunakan untuk biologi molekuler atau untuk membuat persebaran vegetasi pada suatu wilayah. Terdapat beberapa skema warna yang digunakan untuk menggambarkan data pada *heat map* yaitu warna keabu-abuan (*grayscale*), warna yang sudah terdapat pada *colormaps* suatu aplikasi, serta perubahan antar warna memberikan persepsi yang berbeda. Beberapa perangkat lunak yang telah mengimplementasikan *heat map* adalah

PermutMatrix, NeoVision Hypersystems, Inc., Sightmap, Microsoft Excel dan lain sebagainya.

Heat map merupakan sebuah peta yang menggambarkan persebaran lokasi dan frekuensi data dalam bentuk pewarnaan. Dibawah ini merupakan salah satu contoh heatmap yang menggambarkan aktivitas masyarakat selama bulan Maret (UrbanIndo, 2012). Warna biru menandakan daerah-daerah dengan frekuensi relatif tidak terlalu banyak, diikuti oleh warna hijau dan kuning dengan frekuensi yang relatif sedang, lalu warna merah dan putih yang menandai daerah yang paling banyak dikunjungi (UrbanIndo, 2012). Gambar 2.7 merupakan salah satu contoh peta yang menggambarkan *heat map*.



Gambar 2.7 Contoh Heat Map

Sumber: UrbanIndo (2012)

Definisi lain dari *heat map* adalah sebuah representasi visual dari data menggunakan warna yang memiliki tingkat perbedaan dari setiap warna berdasarkan nilai pada data tersebut. *Heat map* sering digunakan untuk representasi data jumlah penduduk (Susaki, et al., 2014), penyitaan karena tanggungan kartu kredit (Anon., 2016), kualitas ruang bawah tanah suatu kota (Hou, et al., 2015) atau pencarian jumlah pengguna suatu aplikasi terbanyak (Oksanen, et al., 2015). *Heat map* juga dapat didefinisikan sebagai grafis yang menggunakan warna untuk menunjukkan pola data dan tingkat aktivitas suatu pola tersebut (Anon., 2016).

Heat map biasa digunakan sebagai teknik untuk visualisasi pada berbagai bidang penelitian di mana sejumlah data yang sangat banyak digunakan. Misalnya dalam interaksi manusia dengan komputer "*attention heat map*" adalah alat yang populer untuk analisis pada visual data hanya dilihat dengan

mata saja (Oksanen, et al., 2015). *Heat map* juga dapat digunakan untuk mengungkapkan tempat-tempat yang populer atau yang memiliki banyak pengunjung menggunakan pengelompokan spasial berdasarkan kepadatan gambar atau untuk memvisualisasikan pola pertumbuhan sementara pada distribusi *tweet*. Warna yang digunakan untuk membuat sebuah *heat map* biasanya dipilih sedemikian rupa bahwa interpretasi dari perbedaan suatu intensitas itu sangat penting. Jadi suhu pada intensitas warna sangat diperhitungkan seperti jika data itu memiliki intensitas yang tinggi maka menggunakan warna yang hangat/panas dan untuk data dengan intensitas rendah maka diwakili dengan warna yang lebih dingin. Contoh warna yang dapat digunakan untuk membuat *heat map* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Contoh Penggunaan Warna Pada Heat Map

Sumber: Heatmap (2016)

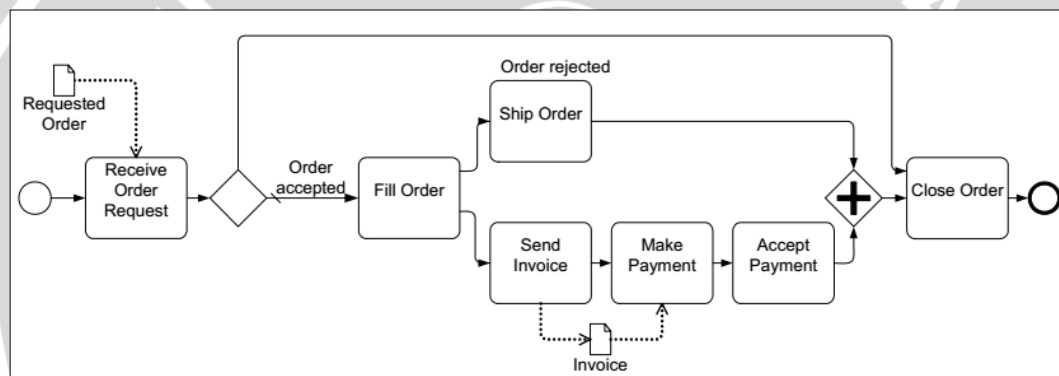
2.2.4 Sistem Informasi

Sistem informasi merupakan sistem di dalam suatu organisasi yang menggabungkan antara pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak di luar sistem dengan laporan-laporan yang diperlukan (Taufiq, 2013). Kemudian sistem informasi ini mampu untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi. Sistem informasi berbeda dengan komputer sistem TIK maupun proses bisnis. Berikut beberapa alasan yang membuat perbedaan yang jelas antara sistem informasi, dan komputer sistem TIK, dan proses bisnis. Sistem informasi yang berbeda dari teknologi informasi dalam sistem informasi biasanya terlihat seperti memiliki komponen TIK. Hal ini terutama berkaitan dengan tujuan pemanfaatan teknologi informasi. Sistem informasi juga berbeda dari proses bisnis. Sistem informasi membantu untuk mengontrol kinerja proses bisnis (Taufiq, 2013).

Definisi lain dari sistem informasi adalah sistem yang tujuan utamanya untuk mengelola dan menyediakan akses ke informasi basis data (Sommerville, 2011). Isu dalam sistem informasi meliputi keamanan, kegunaan, privasi, dan menjaga integritas data. (Sommerville, 2011).

2.2.5 Pemodelan Proses Bisnis

Salah satu media yang digunakan untuk menyederhanakan proses bisnis dan menotasikannya ke dalam sebuah diagram adalah dengan menggunakan Business Process Model and Notation atau disingkat BPMN. Tujuan utama dari BPMN adalah untuk memberikan notasi yang mudah dipahami oleh semua pengguna bisnis, dari bisnis analis yang menciptakan konsep awal dari proses, para pengembang teknis yang bertanggung jawab untuk membangun teknologi secara teknis dan untuk para pebisnis yang akan mengelola teknologi yang dibuat tersebut. Oleh karena itu, BPMN menjadi penerjemah antara desain proses bisnis dengan proses pelaksanaannya. Berikut salah satu contoh penggambaran proses bisnis menggunakan BPMN (Gambar 2.9).



Gambar 2.9 Contoh Pemodelan Proses Bisnis

Sumber : Object Management Group (2011)

2.2.6 Rekayasa Perangkat Lunak

2.2.6.1 Proses Perangkat Lunak

Sebuah proses perangkat lunak atau *software process* adalah serangkaian kegiatan terkait yang mengarah kepada proses produksi suatu produk perangkat lunak. Kegiatan ini melibatkan pengembangan perangkat lunak dari awal dalam bahasa pemrograman standar seperti Java atau C. Kemudian aplikasi bisnis baru dikembangkan dengan memperluas dan memodifikasi sistem yang ada atau dengan mengkonfigurasi dan mengintegrasikan perangkat lunak atau komponen sistem (Sommerville, 2011). Menurut Sommerville (2011) ada banyak *software process* yang berbeda namun semua harus mencakup keempat kegiatan untuk rekayasa perangkat lunak berikut ini:

1. **Spesifikasi perangkat lunak** adalah pembuatan fungsionalitas perangkat lunak dan batasan masalah pada operasi yang harus didefinisikan.
2. **Desain perangkat lunak** untuk memenuhi spesifikasi yang harus diproduksi

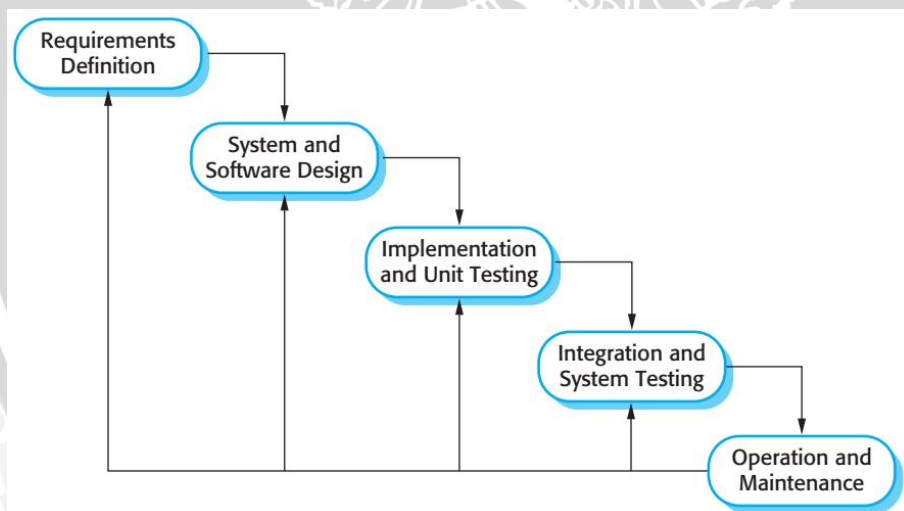
3. **Validasi perangkat lunak** dilakukan untuk proses validasi yaitu memastikan kelengkapan menurut apa yang diinginkan oleh konsumen
4. **Evolusi perangkat lunak** harus dilakukan untuk mengembangkan perangkat lunak yang sudah dibuat dan memenuhi perubahan kebutuhan pelanggan.

2.2.6.2 Model Proses Perangkat Lunak

Model proses perangkat lunak atau *software process models* merupakan penyederhanaan dari seluruh proses yang terjadi di dalam perangkat lunak (Sommerville, 2011). Setiap model proses merupakan proses dari perspektif tertentu, hanya menyediakan sebagian informasi tentang proses yang dijelaskan. Contohnya model proses suatu kegiatan menunjukkan aktivitas dan urutannya namun tidak menunjukkan peran orang yang terlibat dalam kegiatan tersebut. Berikut beberapa contoh model proses untuk membangun suatu perangkat lunak:

1. Waterfall Model

Model ini merupakan model proses yang paling dasar untuk membuat spesifikasi persyaratan, desain perangkat lunak, implementasi, pengujian dan sebagainya (Sommerville, 2011). Berikut merupakan alur untuk menggambarkan model *waterfall* (Gambar 2.10).



Gambar 2.10 Model Waterfall

Sumber: Ian Sommerville (2011)

Requirements analysis and definition adalah tahapan pada layanan sistem, batasan masalah dan tujuan yang ditetapkan telah dikonsultasikan dengan pengguna sistem. Kemudian spesifikasi sistem dibuat berdasarkan data dari konsultasi dengan pengguna tersebut.

System and software design merupakan proses untuk desain sistem yang menyesuaikan dengan persyaratan yang dibuat sebelumnya pada perangkat keras dan perangkat lunak dengan membentuk arsitektur sistem secara

keseluruhan. Desain perangkat lunak melibatkan identifikasi dan menggambarkan abstraksi sistem perangkat lunak yang mendasar dan hubungannya.

Implementation and unit testing, selama tahap ini, desain perangkat lunak akan direalisasikan ke dalam serangkaian kode program atau unit program dan unit pengujian akan melibatkan verifikasi bahwa setiap unit memenuhi spesifikasi yang telah dibuat.

Integration and system testing adalah tahapan dimana unit program diintegrasikan dan diuji secara lengkap untuk memastikan bahwa persyaratan perangkat lunak telah terpenuhi. Setelah pengujian, sistem akan disampaikan kepada pelanggan.

Operation and maintenance, fase ini biasanya merupakan fase yang paling lama. Sistem akan dipasang dan digunakan oleh pengguna, kemudian melakukan pemeliharaan yang harus mengoreksi setiap kesalahan yang tidak ditemukan di awal tahap siklus hidup pengembangan sistem. lalu meningkatkan pelaksanaan unit sistem dan meningkatkan layanan sistem sebagai persyaratan yang baru ditemukan.

2. Incremental Development

Pendekatan ini berfokus pada kegiatan membuat spesifikasi, pengembangan dan validasi. Sistem dikembangkan dengan serangkaian versi yang bertahap, masing-masing versi menambahkan fungsionalitas ke versi sebelumnya.

3. Reuse-oriented Software Engineering

Pendekatan ini didasarkan pada keberadaan sejumlah besar komponen yang dapat digunakan kembali. Proses pengembangan sistem berfokus pada mengintegrasikan komponen ke dalam sistem daripada harus mengembangkannya dari awal.

2.2.6.3 Relational Unified Process

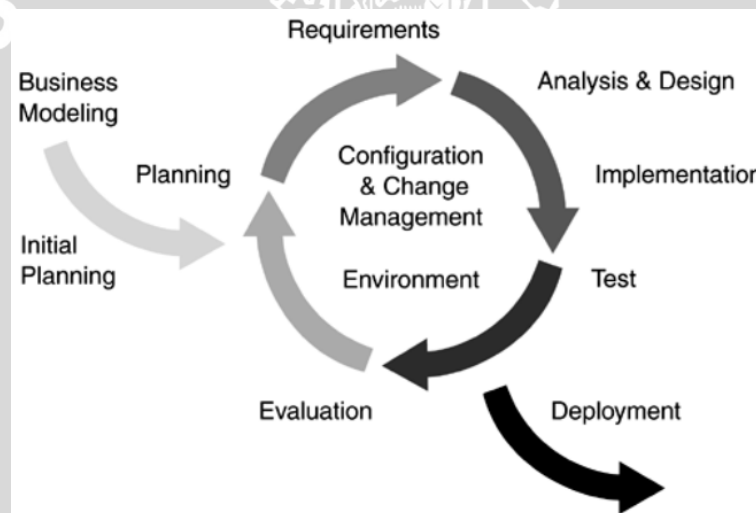
Relational Unified Process atau RUP adalah contoh modern dari model proses yang diperoleh dari pekerjaan UML dan berhubungan dengan *Unified Software Development Process* (Sommerville, 2011). Model proses ini merupakan contoh yang baik untuk model proses campuran karena dapat sekaligus membawa unsur dari semua model proses yang umum, menggambarkan praktek yang baik dalam spesifikasi dan desain serta mendukung pembuatan prototipe dan pengiriman tambahan (Sommerville, 2011).

Menurut Per Kroll et al. (2003), RUP merupakan pendekatan pengembangan perangkat lunak yang berulang, arsitektur sentris dan berbasis *use case*. Hal ini dijelaskan dalam berbagai laporan resmi maupun buku bahwa informasi yang paling lengkap dapat ditemukan dalam produk RUP itu sendiri, berisi pedoman rinci, contoh dan template yang mencakup siklus hidup perangkat lunak. RUP juga merupakan rekayasa perangkat lunak yang jelas dan

terstruktur dengan baik, karena jelas dalam mendefinisikan pencapaian proyek yakni siapa yang bertanggung jawab untuk sesuatu, bagaimana hal tersebut dilakukan dan ketika harus melakukannya (Kroll & Kruchten, 2003). Selain itu RUP dapat didefinisikan sebagai produk proses yang menyediakan kerangka proses yang disesuaikan untuk rekayasa perangkat lunak, kemudian dapat dikonfigurasi menjadi produk RUP untuk mendukung tim kecil maupun besar serta memungkinkan untuk dapat membagikan pengalaman dan artefak dengan teman sebayanya atau para ahli (Kroll & Kruchten, 2003).

1. Pendekatan RUP

RUP menggunakan pendekatan iteratif atau pengulangan yaitu urutan langkah-langkah tambahan atau disebut dengan iterasi (Kroll & Kruchten, 2003). Setiap iterasi terdiri dari sebagian disiplin pengembangan yaitu persyaratan, analisis, desain, implementasi dan sebagainya. Setiap iterasi bertujuan untuk menghasilkan implementasi kerja parsial yaitu dengan dibangun diatas iterasi sebelumnya untuk memperbaiki sistem sampai produk akhir selesai (Kroll & Kruchten, 2003). Pada gambar 2.11 merupakan ilustrasi untuk pengembangan iterasi atau berulang pada RUP.



Gambar 2.11 Pengembangan Berulang Pada RUP

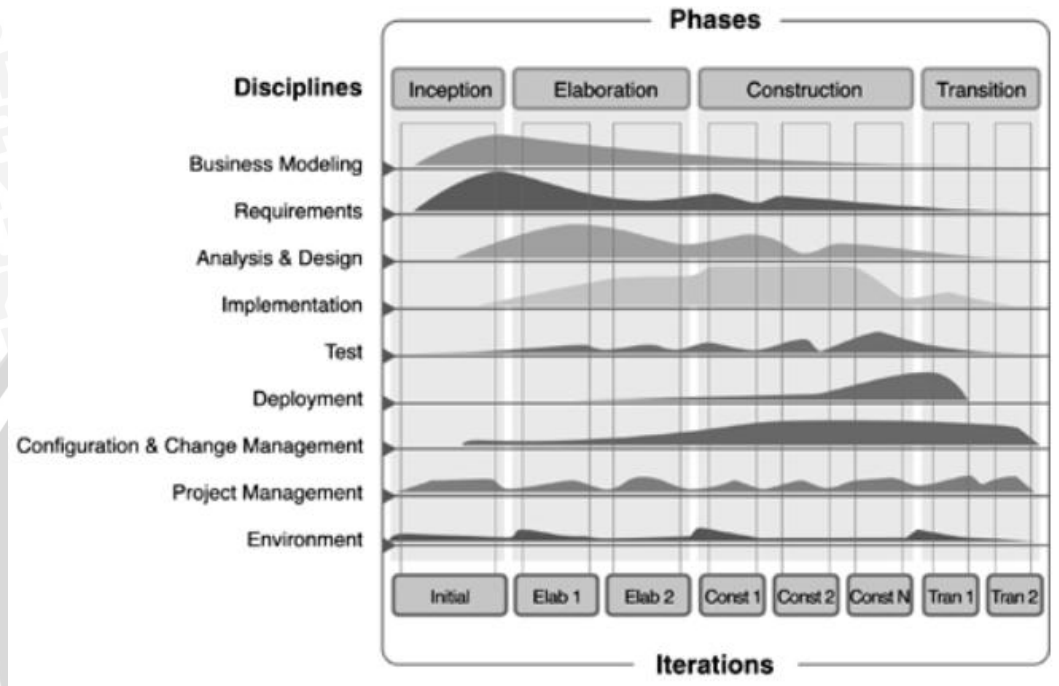
Sumber: Per Kroll, Philippe Kruchten (2003)

2. Arsitektur RUP

Menurut Per Kroll dan Philippe Kruchten (2003), Arsitektur pada RUP memiliki dua struktur atau dimensi yaitu:

- **Struktur dinamis** adalah dimensi horizontal yang menunjukkan bagaimana proses dinyatakan dalam siklus, tahapan, iterasi dan *milestone* selama siklus hidup sebuah proyek.

- **Struktur statis** merupakan dimensi vertikal yang mewakili struktur statis dari proses. Struktur ini menggambarkan bagaimana proses pada unsur-unsur kegiatan, disiplin, artefak dan peran secara logis kemudian dikelompokkan ke dalam disiplin alur kerja proyek.

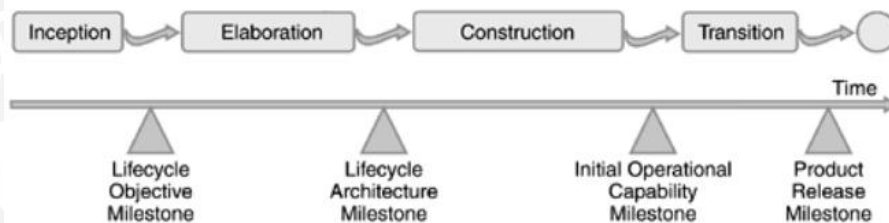


Gambar 2.12 Dua Dimensi Pada RUP

Sumber: Per Kroll, Philippe Kruchten (2003)

Gambar 2.12 menjelaskan tentang struktur dinamis dan statis pada siklus hidup RUP.

Struktur Dinamis pada RUP berhubungan dengan siklus hidup atau dimensi waktu pada proyek. RUP menyediakan pendekatan terstruktur untuk pengembangan berulang, membagi proyek menjadi empat tahap yaitu *Inception*, *Elaborasi*, *Konstruksi* dan *Transisi*. Berikut *milestone* / capaian pada struktur dinamis RUP (Gambar 2.13).



Gambar 2.13 Milestone untuk Fase Siklus Hidup RUP

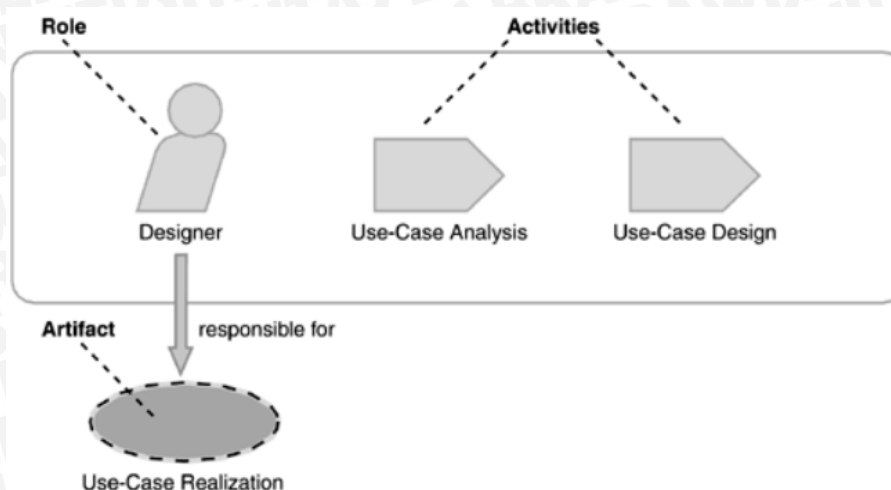
Sumber: Per Kroll, Philippe Kruchten (2003)

Setiap fase berisi satu atau lebih iterasi yang berfokus untuk memproduksi penyampaian teknis yang dibutuhkan guna mencapai tujuan bisnis pada fase tersebut (Kroll & Kruchten, 2003). Iterasi yang harus ditambahkan merupakan iterasi yang dibutuhkan saja, tidak melebihi dari yang dibutuhkan karena dapat menunda pengerjaan proyek. Untuk menghindari hal tersebut, iterasi harus berfokus pada yang dibutuhkan saja guna mencapai tujuan bisnis pada fase itu (Kroll & Kruchten, 2003).

Menurut Per Kroll dan Philippe Kruchten (2003), berikut penjelasan *milestone* untuk fase siklus hidup RUP:

- **Inception.** Memahami dengan baik tentang sistem yang akan dibangun dengan mendapatkan pemahaman dari semua persyaratan dan menetapkan ruang lingkup sistem tersebut.
- **Elaboration.** Menangani banyak tugas teknis seperti desain, implementasi, tes dan arsitektur dasar yang dieksekusi termasuk subsistem, antarmuka, komponen utama dan mekanisme arsitektur.
- **Construction.** Melakukan sebagian besar implementasi ketika arsitektur dieksekusi untuk versi pertama sistem. Menghasilkan beberapa rilis internal dan *alpha* untuk memastikan apakah sistem sudah sesuai dengan kebutuhan pengguna dan rilis *beta* yang digunakan sebagai versi akhir dengan sistem yang sudah berfungsi secara penuh termasuk pemasangan dan dokumentasi serta materi untuk pelatihan pada penggunaan sistem.
- **Transition.** Memastikan bahwa perangkat lunak ditempatkan pada pengguna yang tepat dan dilakukan pengujian produk saat persiapan rilis dan membuat penyesuaian berdasarkan masukan dari pengguna. Pada tahap ini fokus utamanya adalah pada masalah penyetelan produk, konfigurasi, pemasangan dan kegunaan.

Struktur Statis pada RUP berkaitan dengan bagaimana aktivitas proses, disiplin, artefak dan peran dikelompokkan ke dalam disiplin proses inti. Sebuah proses menjelaskan siapa (*who*) yang melakukan apa (*what*), bagaimana (*how*) dan kapan (*when*).



Gambar 2.14 Peran, Aktivitas dan Artefak

Sumber: Per Kroll, Philippe Kruchten (2003)

Gambar 2.14 merupakan ilustrasi yang berkaitan dengan struktur statis pada RUP. Berikut penjelasan keempat pemodelan pada struktur statis RUP tersebut.

- Peran atau *roles* (*who*) mendefinisikan bagaimana individu harus melakukan pekerjaannya dan menentukan kompetensi serta tanggung jawab bahwa setiap individu memainkan peran yang sesuai. Seseorang dapat melakukan satu atau lebih peran dan beberapa orang dapat melakukan peran yang sama.
- Aktivitas atau *activities* (*how*) merupakan suatu aktivitas yang dilakukan oleh peran tertentu, memiliki tujuan yang jelas seperti memperbarui beberapa artefak seperti model, komponen atau rencana. Suatu aktivitas biasanya membutuhkan waktu, melibatkan lebih dari satu orang dan harus dapat digunakan sebagai unsur perencanaan dan kemajuan.
- Artefak atau *artifacts* (*what*) adalah potongan informasi yang dihasilkan, dimodifikasi atau digunakan pada suatu proses. Artefak digunakan sebagai input oleh peran untuk melakukan suatu aktivitas dan hasil output dari kegiatan lainnya.
- Alur kerja atau *workflows* (*when*) digunakan untuk menggambarkan urutan aktivitas yang bermakna dan menghasilkan beberapa produk berharga serta untuk menunjukkan interaksi antar peran.

3. Kerangka kerja proses RUP

Setiap proyek dan setiap organisasi memiliki kebutuhan masing-masing dan membutuhkan proses disesuaikan dengan situasi yang lebih spesifik pada masing-masing proyek dan organisasi tersebut (Kroll & Kruchten, 2003). Berikut bagian-bagian dari produk RUP yang dapat membantu dalam menyelesaikan kerangka kerja proses (Gambar 2.15):



Gambar 2.15 Kerangka Kerja Proses RUP

Sumber: Per Kroll, Philippe Kruchten (2003)

Best practices. RUP dilengkapi dengan perpustakaan praktik terbaik (*best practice*) yang diproduksi oleh IBM Software dan mitra kerjanya. Praktik-praktik terbaik ini terus dikembangkan dan mencakup lingkup yang lebih luas, disajikan dalam bentuk fase, peran, kegiatan, artefak dan alur kerja.

Process delivery tools. RUP disampaikan kepada para pengemang melalui media online menggunakan teknologi web. Penyampaian ini memungkinkan proses diintegrasikan dengan banyak alat untuk mengembangkan perangkat lunak dan dengan alat-alat lain sehingga pengembang dapat mengakses panduan dengan alat mereka.

Configuration tools. Bentuk modular dan elektronik dari RUP memungkinkan dapat disesuaikan dan dikonfigurasi untuk memenuhi kebutuhan spesifik dari organisasi yang sedang membangun sistem atau perangkat lunak.

Process authoring tools. *Rational Process Workbench* (RPW) merupakan alat untuk proses penulisan yang memungkinkan untuk menangkap praktik terbaik (*best practice*) ke dalam forma RUP.

Community/Marketplace. *Rational Developer Network* (RDN) merupakan sebuah komunitas online yang memungkinkan para pengguna untuk bertukar pengalaman dengan rekan-rekan dan para ahli serta menyediakan akses informasi terbaru berupa artefak, atikel atau konten tambahan pada RUP.

2.2.7 Pembuatan Visi Bersama

2.2.7.1 Analisis Masalah

Sebuah masalah dapat didefinisikan sebagai perbedaan antara hal-hal seperti yang dirasakan dan yang diinginkan atau sebagai pertanyaan yang harus dikerjakan (Bittner & Spence, 2002). Dalam banyak kasus, mengubah persepsi pelanggan dari apa yang mereka miliki sekarang atau yang mereka inginkan

adalah cukup untuk menyelesaikan masalah. Jika perbedaan tidak ada antara apa yang dianggap dimiliki dan apa yang ingin dimiliki, maka itu berarti tidak memiliki masalah. Cara terbaik untuk menangkap masalah adalah membangun sebuah pernyataan masalah (*problem statement*). Pernyataan masalah ini merupakan ringkasan solusi yang digunakan oleh para pemangku kepentingan dari masalah yang akan dipecahkan (Bittner & Spence, 2002). Menurut Bittner dan Spence (2002), untuk membuat sebuah pernyataan masalah yang terdiri dari dampak masalah, solusi dan manfaatnya dapat menggunakan template sederhana seperti pada tabel berikut.

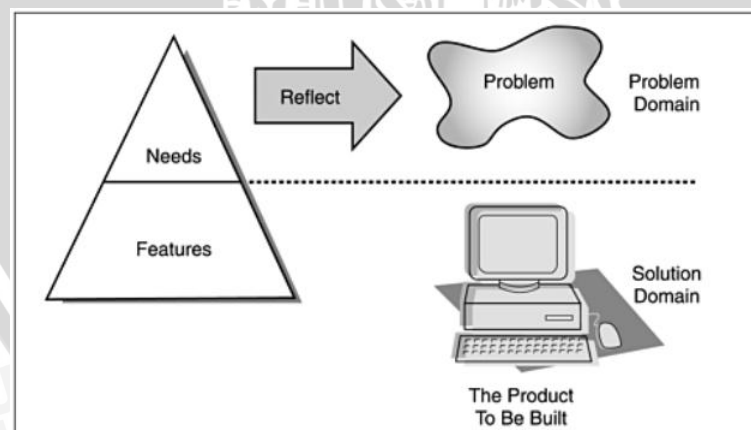
Tabel 2.1 Template Pernyataan Masalah

Masalah	[deskripsi masalah]
Mempengaruhi	[pemangku kepentingan yang terkena dampak]
Dampak Masalah	[apa dampak dari masalah?]
Solusi yang berhasil dapat	[daftar beberapa manfaat utama dari solusi yang berhasil]

Sumber: Kurt Bittner, Ian Spence (2002)

2.2.7.2 Pemangku Kepentingan Utama dan Kebutuhan Pengguna

Memecahkan masalah yang kompleks akan melibatkan beberapa pemangku kepentingan yang biasanya memiliki perspektif berbeda pada masalah dan kebutuhan kemudian akan diperbaiki pada solusinya (Bittner & Spence, 2002). Hal ini dapat dilacak dengan jelas dengan menangkap dan mendokumentasikan kebutuhan masing-masing jenis pemangku kepentingan. Definisi dari kebutuhan pemangku kepentingan adalah refleksi dari bisnis, masalah pribadi atau operasional (atau kesempatan) yang harus diatasi untuk membenarkan pertimbangan, pembelian, atau penggunaan sistem baru (Bittner & Spence, 2002).



Gambar 2.16 Kebutuhan dari Masalah Pemangku Kepentingan

Sumber: Kurt Bittner, Ian Spence (2002)

Gambar 2.16 mengilustrasikan hubungan antara kebutuhan dan pernyataan masalah dengan cara mengumpulkan kebutuhan pemangku kepentingan. Cara ini memungkinkan untuk memahami bagaimana dan sejauh

mana aspek berbeda dari masalah yang mempengaruhi berbagai jenis pemangku kepentingan (Bittner & Spence, 2002). Para pemangku kepentingan akan memberikan wawasan dari akar penyebab masalah bersama secara keseluruhan dan menentukan satu set pernyataan persyaratan solusi yang jika bertemu dapat memecahkan masalah bisnis yang mendasarinya. Deskripsi dari masing-masing kebutuhan pemangku kepentingan harus mencakup alasan yang jelas dibaliknya dan menunjukkan pentingnya kebutuhan pemangku kepentingan yang terkena dampak tersebut. Pemangku kepentingan perlu memberikan latar belakang dan alasan yang jelas mengapa dibuat persyaratan itu diperlukan. Menurut Bittner dan Spence (2002), salah satu contoh dalam pembuatan kebutuhan pemangku kepentingan yaitu pada sistem telepon sederhana yang mencakup:

Easy to use: sistem harus mudah digunakan oleh kedua pemakai teknologi yang memungkinkan semua pengguna menggunakan fitur-fitur canggih dari sistem dengan sederhana dan efektif.

Provide Up-to-Date Status Information: sistem ini akan memberikan informasi secara *real-time* untuk semua pengguna yang berhubungan dengan durasi dan biaya panggilan.

Extensible: sistem ini harus diperluas, memungkinkan pengenalan layanan dan fasilitas baru tanpa gangguan pada tingkat layanan pelanggan yang disediakan.

Bittner dan Spence (2002) merekomendasikan penggunaan aturan *MoScow* yaitu dengan memprioritaskan kebutuhan dari pemangku kepentingan. *MoScow* berasal dari huruf pertama pada kriteria prioritas sebagai berikut:

Must have (Mo) berarti harus dimiliki,

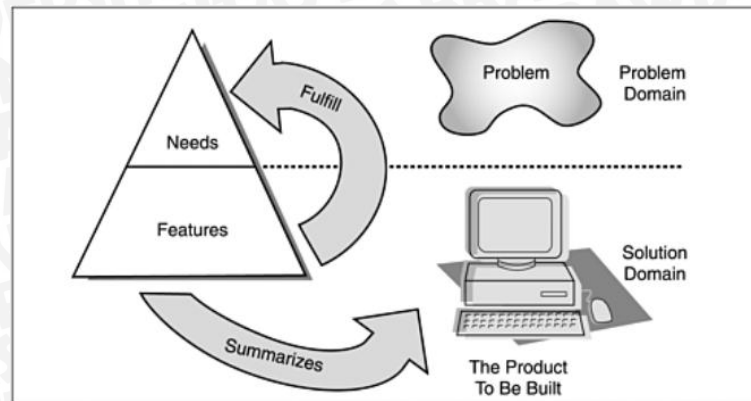
Should have (S) berarti sebaiknya dimiliki,

Could have (Co) berarti dapat dimiliki,

Want to have but will not have this time round (W) berarti dapat tidak dimiliki.

2.2.7.3 Deskripsi Fitur dan Persyaratan Produk

Menurut Bittner dan Spence (2002), fitur merupakan kemampuan tingkat tinggi (layanan dan kualitas) dari sistem yang dibutuhkan untuk memberikan manfaat kepada pengguna dan dapat membantu memenuhi kebutuhan pemangku kepentingan dan pengguna. Pada Gambar 2.17 di bawah ini menggambarkan hubungan antara kebutuhan, fitur dan sistem yang akan dibangun.



Gambar 2.17 Hubungan Antara Kebutuhan, Fitur dan Sistem

Sumber: Kurt Bittner, Ian Spence (2002)

Fitur dapat dikembangkan atau diuraikan menjadi lebih rinci lagi menjadi fungsional dan non fungsional. Banyak fitur yang dapat diakses dengan memerlukan serangkaian masukan untuk mencapai hasil yang diinginkan (Bittner & Spence, 2002). Fitur yang digunakan harus dapat meringkas kemampuan dan kualitas dari produk yang dibangun dan dapat diakses oleh semua anggot proyek dan pemangku kepentingan. Namun fitur ini harus cukup detail untuk memberikan anggota tim informasi yang dibutuhkan untuk membentuk, melakukan validasi dan mengelola model kasus uji serta spesifikasi tambahan.

Menurut Bittner dan Spence (2002), ketika mendokumentasikan fitur harus mencakup tiga bagian berikut ini, yakni menyertakan deskripsi fungsi dan masalah kegunaan yang harus diatasi, berfokus pada kemampuan yang diperlukan dan mengapa harus dilaksanakan serta menetapkan fitur pengenalan yang unik untuk referensi yang mudah dilakukan pelacakannya. Fitur dari sistem dapat dikategorikan dan disajikan berdasarkan fungsional dan jenisnya. Fitur dapat diurutkan berdasarkan prioritasnya sehingga mudah untuk membedakannya. Proses menetapkan prioritas dapat menggunakan aturan *MoSCoW*. Tabel merupakan contoh untuk penulisan fitur, deskripsi dan prioritas pada sistem telepon sederhana.

Tabel 2.2 Contoh Penulisan Fitur pada Sistem Telepon Sederhana

Identifier	Deskripsi	Prioritas
FEAT1	Sistem harus memungkinkan penelepon untuk melakukan panggilan lokal	<i>Must</i>
FEAT2	Sistem harus memungkinkan penelepon untuk melakukan panggilan jarak jauh	<i>Must</i>
FEAT3	Sistem ini akan memilih perutean termurah untuk semua panggilan jarak jauh	<i>Should</i>
FEAT4	Sistem menyediakan pembaruan riwayat panggilan secara terus menerus untuk semua akun	<i>Must</i>
FEAT5	Sistem harus tetap tersedia 24 jam sehari, tujuh hari seminggu	<i>Should</i>

Sumber: Kurt Bittner, Ian Spence (2002)

2.2.7.4 Gambaran Umum Produk

Gambaran umum produk yang terdiri dari manfaat, asumsi dan alternatif pengembangan produk perlu didokumentasikan selain pendokumentasian fitur dan persyaratan produk. Sistem yang dibangun harus memiliki alasan seperti membutuhkan pernyataan misi atau alasan sistem harus dibuat. Template untuk menjelaskan gambaran umum produk atau kedudukan produk dapat dilihat pada tabel 2.3 di bawah ini (Bittner & Spence, 2002). Tabel 2.3 ini digunakan untuk mengungkapkan pernyataan penentuan kedudukan produk, elemen kunci dan visi.

Tabel 2.3 Template Pernyataan Kedudukan Produk (Bahasa Inggris)

<i>For</i>	<i>(target customer)</i>
<i>Who</i>	<i>(statement of the need or opportunity)</i>
<i>The</i>	<i>(product name) is a (product category)</i>
<i>That</i>	<i>(statement of key benefit, that is, compelling reason to buy)</i>
<i>Unlike</i>	<i>(primary competitive alternative)</i>
<i>Our Product</i>	<i>(statement of primary differentiation)</i>

Sumber: Kurt Bittner, Ian Spence (2002)

Tabel 2.4 Template Pernyataan Kedudukan Produk (Bahasa Indonesia)

Untuk	(target pelanggan)
Siapa saja yang	(pernyataan kebutuhan dan kesempatan)
Berupa	(nama produk) adalah (kategori produk)
Bahwa	(pernyataan manfaat utama, yaitu, alasan kuat untuk membeli)
Tidak seperti	(alternatif yang kompetitif primer)
Produk kami	(pernyataan pembeda primer)

Sumber: Kurt Bittner, Ian Spence (2002)

Menurut Bittner dan Spence (2002), Ketika menuliskan deskripsi pada kedudukan produk, sebaiknya menggambarkan sistem sesuai keinginan kepada seseorang yang terbiasa dengan hal tersebut dan menyampaikan nilai lebih yang akan diberikan. Jika tidak dapat menjelaskan dengan sederhana, mungkin ide yang dimiliki masih belum jelas tentang apa yang akan dilakukan sistem. Pada tabel 2.4 merupakan contoh penulisan kedudukan produk pada sistem *Automated Teller Machine* (ATM).

Tabel 2.5 Contoh Kedudukan Produk pada sistem ATM

Untuk	Pelanggan rekening saat ini
Siapa saja yang	Membutuhkan akses cepat kepada rincian rekening mereka dan dana yang dikandungnya
Berupa	Super ATM adalah mesin teller otomatis
Bahwa	Memberikan kemampuan untuk melakukan transaksi perbankan sederhana (seperti penarikan atau penyetoran dana atau mentransfer dana antar rekening)
Tidak seperti	Mengakses dana dan rincian melalui loket kantor cabang

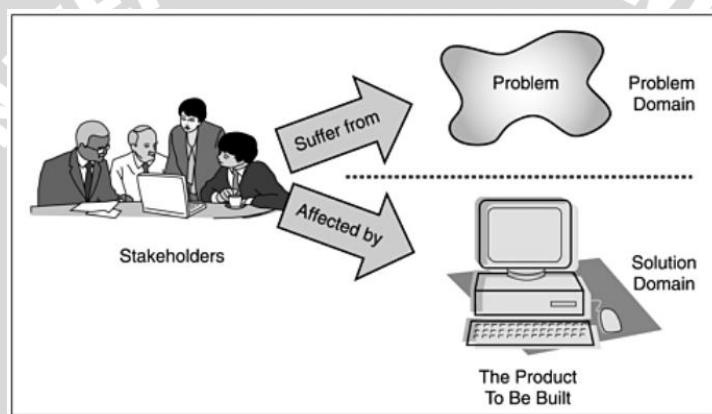
Tabel 2.5 Contoh Kedudukan Produk pada sistem ATM (lanjutan)

Produk kami	Tersedia selama 24 jam dalam sehari dan tidak memerlukan bantuan teller bank.
-------------	---

Sumber: Kurt Bittner, Ian Spence (2002)

2.2.8 Rekayasa Persyaratan Sistem

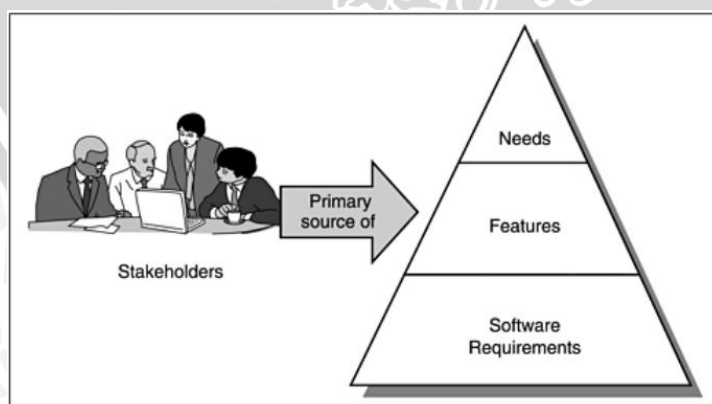
Pengembangan suatu sistem informasi harus disetujui para pemangku kepentingan (*stakeholder*) supaya sistem tersebut dapat dikembangkan dan dibuat sesuai dengan kebutuhan masing-masing pemangku kepentingan. Oleh karena itu dibutuhkan analisis persyaratan untuk membangun sistem tersebut. Pembuatan persyaratan ini diawali dengan pendaftaran siapa saja yang berperan sebagai *stakeholder* dalam pembangunan sistem. Setelah *stakeholder* sudah didefinisikan maka langkah selanjutnya yaitu menjelaskan masalah dan solusinya seperti apa. Berikut ilustrasi dalam analisis masalahnya (Gambar 2.18).



Gambar 2.18 Masalah dan Solusinya

Sumber: Kurt Bittner, Ian Spence (2002)

Kemudian dilakukan pendataan kebutuhan-kebutuhan yang selanjutnya akan dibuat fitur-fitur dalam sistem tersebut (Gambar 2.19). Sehingga persyaratan (*requirements*) sistem dapat dibuat sesuai dengan kebutuhan-kebutuhan yang telah didata sebelumnya (Bittner & Spence, 2002).



Gambar 2.19 Analisis Persyaratan

Sumber: Kurt Bittner, Ian Spence (2002)

2.2.9 Mekanisme Analisis

Mekanisme analisis merupakan pola yang digunakan sebagai solusi umum untuk masalah yang umum dengan menunjukkan pola struktur, pola perilaku atau keduanya. Pola-pola ini digunakan selama analisis untuk mengurangi kompleksitas analisis dan meningkatkan konsistensi dengan menyediakan representasi yang lebih singkat untuk perilaku yang kompleks (Siemers & Kutsick, 2004). Berikut beberapa sampel pada mekanisme analisis menurut Siemers dan Kutsick (2004):

- *Persistency*
- *Communication (IPC and RPC)*
- *Message routing*
- *Distribution*
- *Transaction management*
- *Process control and synchronization (resource contention)*
- *Information exchange, format conversion*
- *Security*
- *Error detection / handling / reporting*
- *Redundancy*
- *Legacy Interface*

Mekanisme analisis memberikan perilaku khusus pada kelas terkait komponen atau sesuai dengan implementasi kerjasama antar kelas dan atau komponen tersebut. Karakteristik pada mekanisme analisis mengambil beberapa persyaratan non fungsional dari sistem, berikut beberapa contoh untuk karakteristik mekanisme analisis menurut Siemers dan Kutsick (2004):

Persistensi digunakan untuk semua kelas yang dapat menjadi persisten, diperlukan identifikasi sebagai berikut:

- *Granularity*: jangkauan ukuran objek yang persisten
- *Volume*: jumlah objek yang tetap persisten
- *Durasi*: berapa lama untuk objek tetap persisten
- *Mekanisme akses*: bagaimana sebuah objek tertentu unik dalam identifikasi dan digunakan?
- *Frekuensi akses*: apakah objek lebih atau kurang konstan dan apakah objek tersebut diperbarui secara permanen?
- *Keandalan*: apakah objek dapat bertahan dari kecelakaan prosesor seluruh sistem?

Komunikasi antar proses digunakan untuk seluruh elemen yang membutuhkan komunikasi dengan objek, komponen dan jasa pelaksana dalam proses lain diperlukan identifikasi sebagai berikut:

- *Waktu tempuh*: seberapa cepat proses harus berkomunikasi dengan yang lain?
- *Sinkronisitas*: komunikasi *asynchronous*
- *Ukuran pesan*: sebuah spektrum mungkin lebih tepat daripada ukuran angka tunggal

- Protokol: alur kontrol, *buffering* dan sebagainya

2.2.10 Pemodelan Sistem

Dalam pengembangan suatu sistem yang melewati fase analisis kebutuhan dan desain sistem dibutuhkan suatu pemodelan untuk membuat desain sistem dapat dimengerti dan mudah untuk dipahami. Salah satu pemodelan yang digunakan untuk membangun sistem adalah pemodelan berbasis objek atau *Object Oriented Analysis and Design*. Pendekatan berbasis objek ini merepresentasikan seluruh benda atau makhluk hidup yang ada di dunia nyata sebagai objek (Booch, et al., 2005). Dalam pendekatan ini terdapat tiga komponen yang bertanggung jawab yaitu komponen yang berinteraksi dengan pengguna atau sistem di luar sistem itu sendiri, penyimpan data dan pengontrol dari aktivitas serta interaksi antara kedua komponen sebelumnya (Siemers & Kutsick, 2004). Ketiga tanggung jawab tersebut secara berurutan disebut dengan *boundary*, *entity* dan *controller* dan dinotasikan dengan simbol yang dijelaskan dalam tabel 2.6 di bawah ini.

Tabel 2.6 Komponen yang Bertanggung Jawab dalam OOAD

Tanggung jawab	Stereotype
<i>Boundary</i>	
<i>Entity</i>	
<i>Controller</i>	

Sumber: Shawn Siemers, Alex Kutsick (2004)

Untuk memodelkan dengan pendekatan berbasis objek ini menggunakan suatu aturan yang disebut dengan *Unified Modelling Language* (UML).

Unified Modelling Language (UML) bertujuan untuk memodelkan secara visual yang digunakan untuk membuat spesifikasi, memvisualkan, membangun / konstruksi dan dokumentasi objek-objek dalam sistem perangkat lunak (Booch, et al., 2005). Dalam UML terdapat beberapa area utama dalam pemodelannya yaitu struktural, dinamis, *physical* dan manajemen model. Pada tabel 2.7 merupakan penjelasan untuk area utama dalam UML.

Tabel 2.7 Area utama dalam UML

Area Utama	Gambaran	Diagram
Struktural	<i>Static View</i>	<i>Diagram Class</i>
	<i>Design View</i>	<i>Internal Structure</i>

Tabel 2.7 Area utama dalam UML (lanjutan)

Area Utama	Gambaran	Diagram
		<i>Collaboration Diagram</i>
		<i>Component Diagram</i>
		<i>Use case View</i>
Dinamis	<i>State Machine View</i>	<i>State machine diagram</i>
	<i>Activity View</i>	<i>Activity diagram</i>
	<i>Interaction View</i>	<i>Sequence diagram</i>
		<i>Communication diagram</i>
<i>Physical</i>	<i>Deployment view</i>	<i>Deployment diagram</i>
Manajemen model	<i>Model manajemen view</i>	<i>Package diagram</i>

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

2.2.10.1 Diagram Use Case

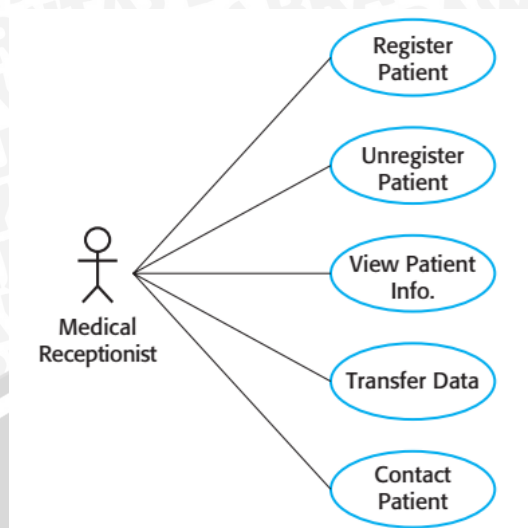
Use case mendeskripsikan interaksi aktor dengan sistem atau aktor dengan aktor lainnya.

a. Aktor

Aktor adalah penggambaran pengguna di luar sistem, sistem atau subsistem yang memiliki peran terhadap sistem. Setiap aktor berpartisipasi terhadap satu atau lebih *use case* di dalam sistem. Seorang aktor bisa berupa manusia, sistem komputer atau proses eksekusi yang lainnya.

b. Use Case

Use case adalah unit bagian dari fungsionalitas sistem yang menyediakan *classifier* dan urutan pertukaran pesan oleh subjek dan satu atau lebih aktor dalam unit sistem. Setiap *use case* merepresentasikan tugas terpisah yang melibatkan interaksi eksternal dengan sistem. Dalam kasus yang paling sederhana, *use case* ditampilkan berupa elips dan aktor yang terlibat dalam *use case* direpresentasikan berupa bentuk orang yang sederhana (Sommerville, 2011). Berikut contoh penggambaran *use case* (Gambar 2.20).



Gambar 2.20 Contoh Use Case

Sumber: Ian Sommerville (2011)

Di dalam diagram *use case*, terdapat relasi atau hubungan antara aktor dengan *use case* atau aktor dengan aktor lainnya. Berikut penjelasan perbedaan relasi-relasi tersebut (Tabel 2.8).

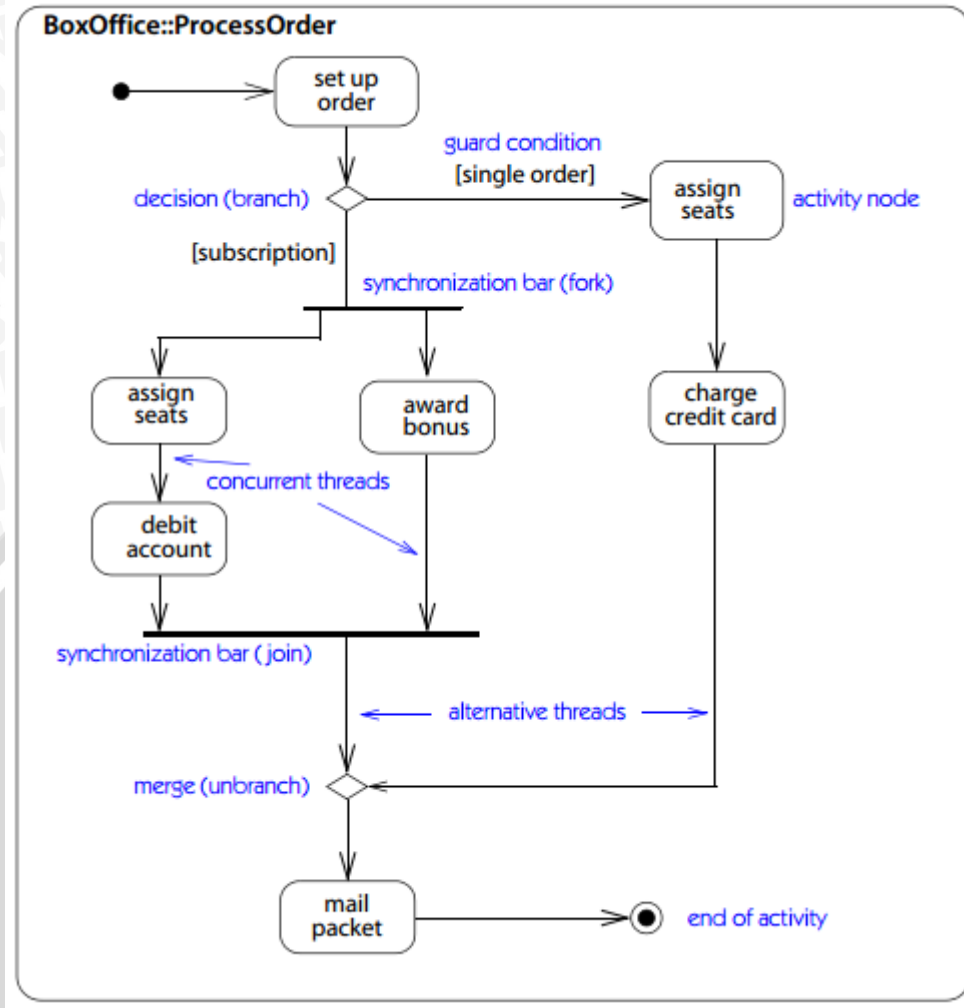
Tabel 2.8 Hubungan Antar Use Case

Hubungan	Fungsi	Notasi
<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berhubungan	_____
<i>Extend</i>	Tingkah laku yang hanya berjalan jika ada kondisi tertentu.	«extend» ----->
<i>Include</i>	Tingkah laku yang harus dipenuhi supaya <i>event</i> dapat terjadi dan <i>use case</i> tersebut merupakan bagian dari <i>use case</i> lainnya	«include» ----->
<i>Use case generalization</i>	Hubungan antar <i>use case</i> yang umum dan lebih spesifik	—————>

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005) &

2.2.10.2 Diagram Aktivitas

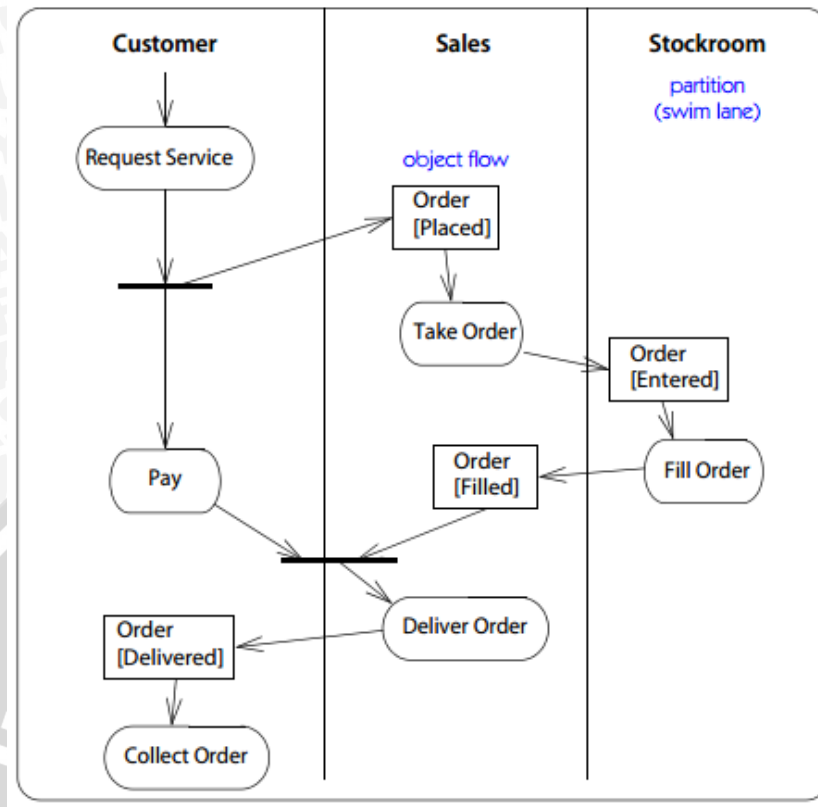
Aktivitas dalam hal ini merupakan rangkaian langkah-langkah yang terjadi di dalam suatu komputasi biasanya digambarkan pada setiap tingkah laku sistem. Aktivitas berisi node-node aktivitas yang ditampilkan dalam kotak dengan sudut yang melingkar berisi penjelasan aktivitasnya. Alur kontrolnya berupa panah dan terdapat kondisi tertentu yang digambarkan dengan bentuk belah ketupat. Berikut contoh dari aktivitas dalam bentuk diagram aktivitas (Gambar 2.21).



Gambar 2.21 Contoh diagram aktivitas

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Adapun bagian yang dibagi menurut aktor atau pengguna yang berhubungan dengan suatu aktivitas dinamakan dengan *partition*. *Partition* ini berguna untuk mengorganisasikan aktivitas yang dilakukan oleh setiap aktor atau pengguna yang melakukan. Pada Gambar 2.22 merupakan contoh dari diagram aktivitas dengan partisipasinya.



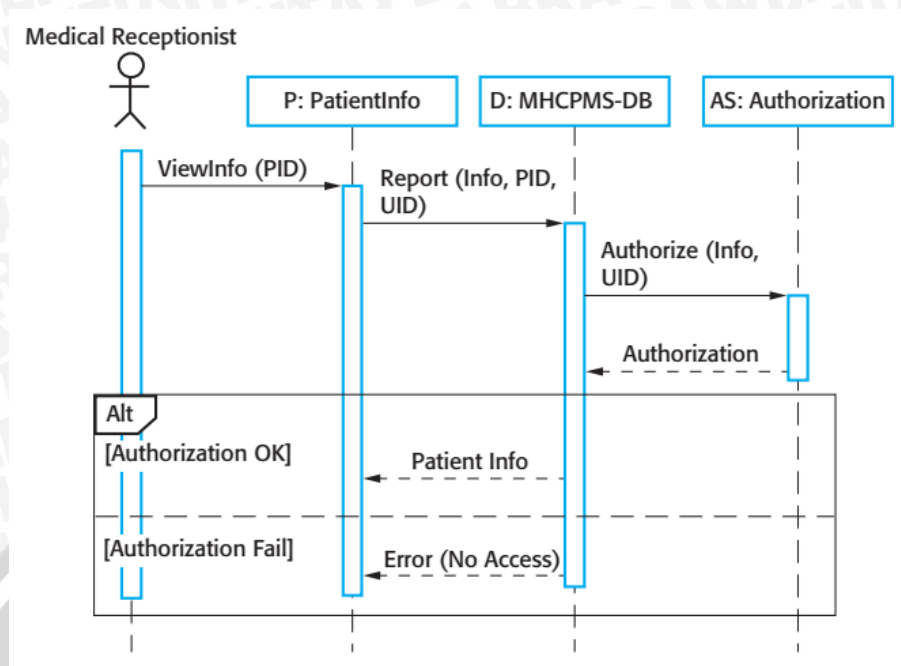
Gambar 2.22 Contoh diagram aktivitas dengan partisi

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

2.2.10.3 Diagram Sequence

Sequence diagram pada UML digunakan untuk memodelkan interaksi antara aktor dan objek dalam suatu sistem dan interaksi antar objek itu sendiri. UML memiliki sintaks yang kaya untuk sequence diagram, yang memungkinkan berbagai jenis interaksi yang dimodelkan (Sommerville, 2011).

Objek-objek dan aktor yang terlibat tercantum di bagian atas diagram, dengan garis putus-putus ditarik secara vertikal ke bawah. Interaksi antara objek ditandai dengan panah horizontal, kemudian persegi panjang pada garis putus-putus menunjukkan jalur hidup dari objek yang bersangkutan. Pembacaan urutan interaksi adalah dari atas ke bawah, penjelasan pada anak panah menunjukkan panggilan ke objek, parameter dan nilai kembalian. Kemudian pada contoh diagram sequence di bawah ini disertakan kotak alernaif yang digunakan ketika ada kondisi tertentu (Gambar 2.23).

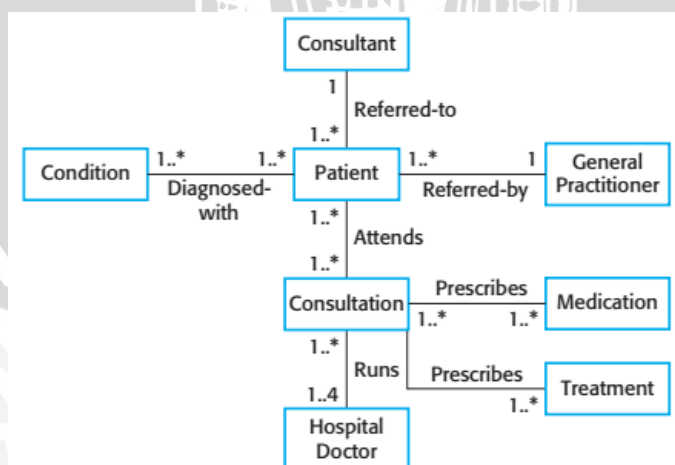


Gambar 2.23 Contoh Diagram Sequence

Sumber: Ian Sommerville (2011)

2.2.10.4 Diagram Kelas

Diagram kelas digunakan ketika mengembangkan model sistem berorientasi objek untuk menunjukkan kelas dalam sistem dan hubungan antara kelas-kelas tersebut. Objek kelas dapat dianggap sebagai definisi umum dari satu jenis objek sistem (Sommerville, 2011). Pada diagram kelas, kelas yang terlibat digambarkan dengan kotak dan jika terdapat hubungan antara kelas dihubungkan dengan sebuah garis yang menampilkan banyaknya objek yang terlibat. Berikut contoh penggambaran diagram kelas (Gambar 2.24) dan detail pada suatu kelas (Gambar 2.25).



Gambar 2.24 Contoh Diagram Kelas

Sumber: Ian Sommerville (2011)

Consultation
Doctors
Date
Time
Clinic
Reason
Medication Prescribed
Treatment Prescribed
Voice Notes
Transcript
...
New ()
Prescribe ()
RecordNotes ()
Transcribe ()
...

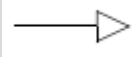
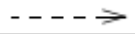
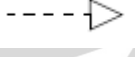
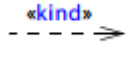
Gambar 2.25 Contoh Detail Kelas

Sumber: Ian Sommerville (2011)

Pada detail kelas (Gambar 2.25) terdapat tiga komponen utama yaitu nama kelas, atribut dan operasi. Atribut kelas menggambarkan karakteristik objek dan operasi menunjukkan hal-hal apa saja yang dapat dikerjakan pada sebuah objek.

Terdapat beberapa jenis relasi atau hubungan antar kelas yang terdiri dari *association*, *generalization*, *dependency*, *realization* dan *usage*. Berikut penjelasan dari jenis-jenis hubungan tersebut (Tabel 2.9).

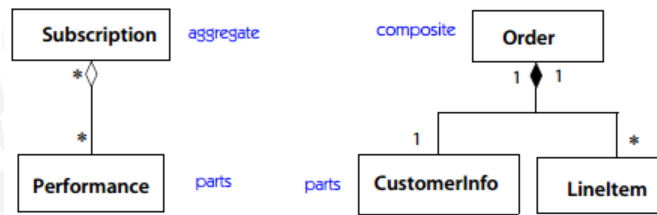
Tabel 2.9 Hubungan Antar Kelas

Hubungan	Fungsi	Notasi
<i>association</i>	Hubungan antar <i>instance</i> dalam suatu kelas	
<i>generalization</i>	Hubungan antara deskripsi umum dan yang lebih spesifik, digunakan untuk tipe deklarasi <i>inheritance</i> dan <i>polymorphic</i>	
<i>dependency</i>	Hubungan antar dua elemen model	
<i>realization</i>	Hubungan antar spesifikasi dan implementasinya	
<i>usage</i>	Situasi dimana satu elemen membutuhkan elemen lain untuk fungsi yang tepat	

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Dalam *association* terdapat dua jenis hubungan lagi yaitu *aggregation* dan *composition*. *Aggregation* adalah *association* yang merepresentasikan seluruh bagian dari hubungan tersebut. Sementara untuk *composition* adalah

association dengan bagian yang lebih kuat yakni pihak yang bertanggung jawab penuh pada bagian lainnya. Berikut contoh dari kedua jenis tersebut (Gambar 2.26).



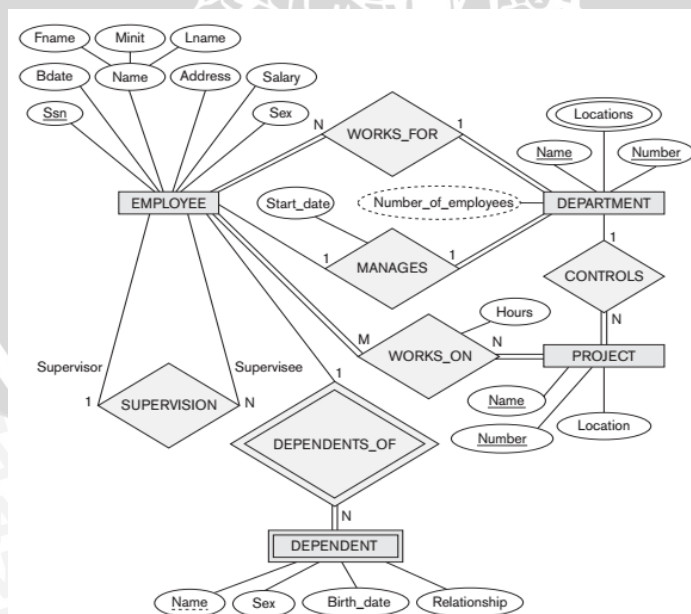
Gambar 2.26 Aggregation dan Composition

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

2.2.11 Pemodelan Data

2.2.11.1 Entity Relationship Diagram

Entity Relationship Diagram (ER diagram) atau diagram hubungan entitas adalah diagram untuk menampilkan hubungan antar entitas-entitas yang menekankan pada skema basisdata karena basisdata jarang terdapat perubahan sedangkan isi entitas sering mengalami perubahan (Elmasri & Navathe, 2011). Dalam ER diagram ini terdapat nama entitas, atribut, *key* dan hubungan antar entitas. Entitas merupakan objek nyata yang memiliki keberadaan secara fisik, kemudian setiap entitas memiliki sifat khusus yang menggambarkan entitas tersebut atau biasa disebut dengan atribut. Terdapat atribut utama sebagai kunci pembeda dari entitas lain yang sejenis dinamakan *key*. Dan yang terakhir adalah relasi antar entitas-entitas yang saling berhubungan (Elmasri & Navathe, 2011). Berikut contoh penggambaran dari ER diagram (Gambar 2.27).

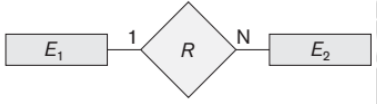


Gambar 2.27 Contoh ER diagram

Sumber: Ramez Elmasri dan Shamkant B. Navathe (2011)

Dalam ER diagram terdapat simbol-simbol yang berbeda, berikut penjelasan simbol-simbol tersebut (Tabel 2.10).

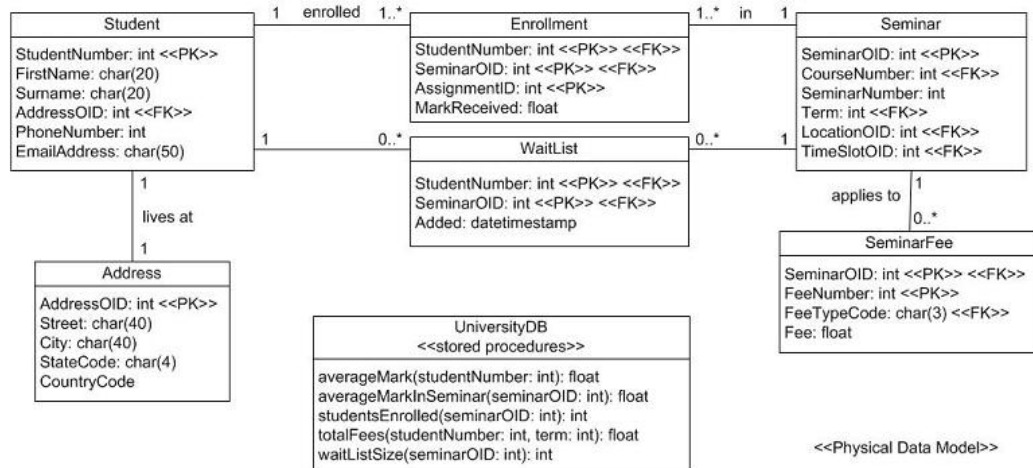
Tabel 2.10 Simbol-simbol ER diagram

No	Simbol	Nama	Definisi
1		<i>Entity</i>	Entitas
2		<i>Weak entity</i>	Entitas lemah
3		<i>Relationship</i>	Relasi
4		<i>Attribute</i>	Atribut
5		<i>Key attribute</i>	Atribut kunci
6		<i>Composite attribute</i>	Atribut yang berhubungan dengan atribut lain
7		<i>Cardinality Ratio 1:N for $E_1:E_2$ in R</i>	Kardinalitas atau banyaknya hubungan yang dibutuhkan seperti hubungan satu ke N

Sumber: Ramez Elmasri dan Shamkant B. Navathe (2011)

2.2.11.2 Physical Data Model

Physical data model adalah model data yang spesifik untuk basis data dan didasarkan pada spesifikasi data yang dibutuhkan. Model ini meliputi partisi, *table space* atau ruang tabel, indeks dan selain objek penyimpanan yang lainnya (Wilson, et al., 2011). Pemodelan *physical data* ini mirip dengan desain pemodelan kelas, yang bertujuan untuk merancang skema internal basisdata, menggambarkan data tabel, kolom tabel dan hubungan antar tabel tersebut (Ambler, 2014). Berikut contoh penggambaran *physical data model* (Gambar 2.28).



Gambar 2.28 Contoh *physical data model*

Sumber: Scott W. Ambler (2014)

Menurut gambar diatas, syarat-syarat yang harus dipenuhi dalam penggambaran *physical data model* terdiri dari, nama tabel, atribut tabel beserta tipe dan ukurannya dengan termasuk *primary key* atau *foreign key* serta relasi antar tabelnya. Adapun relasi antar tabel ini memiliki kardinalitas yang dibagi menjadi tiga (Tabel 2.11).

Tabel 2.11 Kardinalitas

No.	Kardinalitas	Definisi	Contoh
1	One-to-one	jika relasi terjadi dari satu nilai ke satu nilai atau membutuhkan satu data pada masing-masing tabel	
2	One-to-many	jika relasi terjadi dari satu nilai ke banyak nilai atau satu data dengan banyak data	
3	Many-to-many	jika relasi terjadi dari banyak nilai ke banyak nilai atau membutuhkan banyak data pada masing-masing tabel	

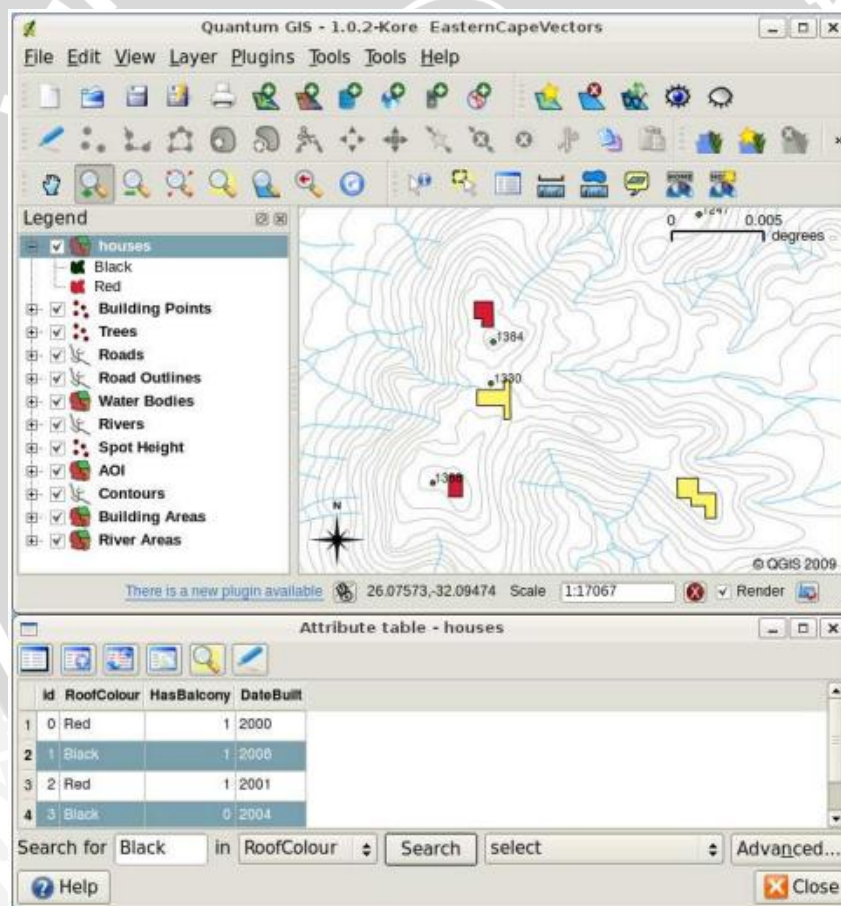
Sumber: Neeraj Sharma et al. (2010)

2.2.12 Analisis Spasial Pada Data Vektor

Analisis spasial pada data vektor menggunakan sebuah aplikasi atau perangkat lunak untuk mengolah data geografis suatu wilayah untuk dibuat menjadi data vektor. Salah satu perangkat lunak untuk mengolah data vektor ini adalah Quantum GIS. Quantum GIS merupakan sebuah aplikasi untuk mengolah data untuk membuat sistem informasi geografis dengan sumber terbuka (*opensource*) dan dapat digunakan pada beberapa sistem operasi seperti Linux,

Unix, Mac OSX, Windows dan Android, serta mendukung banyak format dan fungsionalitas data vektor, raster, dan basis data (Sutton, et al., 2016).

Aplikasi GIS ini akan memungkinkan untuk membuat dan memodifikasi data geometri dalam dalam sebuah lapisan, sebuah proses yang disebut dengan digitalisasi (Sutton, et al., 2002). Jika lapisan berisi poligon (misalnya lahan pertanian) aplikasi GIS hanya akan memungkinkan untuk membuat poligon baru pada lapisan itu. Membuat dan mengubah data vektor adalah fungsi penting yang dibutuhkan untuk membuat data vektor tersebut dapat mengungkapkan masalah yang sedang ditekuni misalnya pemantauan pencemaran di sungai. Hal ini dapat dilakukan dengan cara mendigitalkan titik pengeluaran dari saluran air hujan (berupa titik atau poin) dan sungai itu sendiri (berupa poligon), sehingga dapat diambil pembacaan tingkat PH sepanjang perjalanan sungai (Sutton, et al., 2002). Selain itu aplikasi GIS dapat digunakan untuk digitalisasi perbedaan warna atap rumah, apakah terdapat balkon dan tahun pembangunannya. Berikut contoh proses digitalisasi perbedaan warna atap rumah dan beberapa atributnya (Gambar 2.29).



Gambar 2.29 Contoh Proses Digitalisasi

Sumber: Tim Sutton, Otto Dassau, Marcelle Sutton (2002)

2.2.13 Verifikasi dan Validasi

Menurut Sommerville (2011), proses pada verifikasi dan validasi berfokus pada pemeriksaan perangkat lunak yang sedang dikembangkan apakah sudah memenuhi spesifikasi dan memberikan fungsi yang sudah diharapkan oleh pelanggan untuk pembuatan perangkat lunak tersebut (Sommerville, 2011). Proses-proses pengecekan dilakukan segera setelah persyaratan tersedia dan kemudian melalui seluruh tahapan pada proses pembangunan.

Tujuan dari verifikasi, untuk memeriksa apakah perangkat lunak dinyatakan memenuhi persyaratan fungsional dan non fungsional. Sementara tujuan pada validasi adalah memastikan bahwa perangkat lunak sudah memenuhi harapan dari pelanggan (Sommerville, 2011). Menurut Pressman (2010) proses verifikasi mengacu pada serangkaian tugas yang memastikan bahwa perangkat lunak dapat melakukan implementasi pada sebuah fungsi tertentu. Kemudian untuk validasi mengacu pada satu set tugas berbeda yang memastikan bahwa perangkat lunak telah dibangun dan dapat dilacak dengan menggunakan kebutuhan pelanggan (Pressman, 2010). Verifikasi dan validasi meliputi beragam kegiatan pada *Software Quality Assurance* (SQA) yaitu *review* teknis, audit kualitas dan konfigurasi, pemantauan kinerja, simulasi, studi kelayakan, *review* dokumentasi, *review* basisdata, analisis algoritma, pengembangan pengujian, pengujian kegunaan, pengujian kualifikasi, pengujian penerimaan dan pemasangan pengujian.

2.2.14 Tinjauan Spesifikasi Persyaratan

Tinjauan spesifikasi persyaratan (*requirements specification reviews*) dilakukan dengan menggunakan daftar periksa/*checklist* yaitu dengan memastikan daftar sudah lengkap dan konsisten dan masing-masing persyaratan tidak ambigu atau mengandung makna ganda serta dapat diuji. Daftar periksa ini dibuat pada masing-masing kriteria yang berbeda. Salah satu contoh kriteria pada daftar periksa untuk melakukan peninjauan spesifikasi persyaratan dinamakan dengan inspeksi persyaratan atau *walkthrough checklist* yang terdiri dari *Organization*, *Completeness*, *Correctness*, *Quality Attributes* (*Non functional requirements*), *Traceability* dan *Special Issues* (Wiegers, 1999). Inspeksi persyaratan ini menggunakan serangkaian langkah-langkah untuk memutuskan apakah karakteristik pada persyaratan sistem sudah layak atau belum dan membandingkannya dengan standar yang seharusnya dilakukan (UCCS, 2007).

2.2.15 Evaluasi Prototipe

Evaluasi prototipe yaitu meninjau sistem atau menggunakan sistem yang dilakukan oleh pengguna, berfungsi agar pengguna dapat terbiasa dalam menggunakan sistem kemudian pengguna dapat menemukan apakah terdapat persyaratan yang belum sesuai dengan keinginan pengguna atau sudah sesuai seluruhnya (Sommerville, 2011). Masalah umum pada prototipe adalah prototipe mungkin tidak sepenuhnya diimplementasikan ke dalam sistem yang sudah final. Jika prototipe mengalami kelambatan, penguji atau evaluator dapat mencatat

fitur yang lambat tersebut dan sistem dapat diselesaikan menggunakan cara berbeda yang lebih baik dalam sistem akhir (Sommerville, 2011). Jadi evaluasi pada prototipe dibutuhkan sebagai acuan untuk memperbaiki fungsionalitas dan desain antarmuka pada pengembangan sistem selanjutnya. Evaluasi ini dilakukan dengan melibatkan perwakilan dari pengguna sistem kedepannya.

2.2.16 Website Usability Evaluation

Menurut Nielsen (2003) definisi dari *usability* adalah atribut kualitas yang menilai seberapa mudah antarmuka pengguna untuk digunakan (Rinder, 2012). Sementara itu *website usability testing* merupakan pengujian terhadap jumlah kesuksesan pada pengalaman pengguna saat berinteraksi dengan situs web. Dengan kata lain, hal tersebut adalah sejauh mana pengguna dapat mencapai tujuan yang diinginkan saat berkunjung ke situs web tertentu (Rinder, 2012). Kemudian tujuan diadakannya pengujian pada *website usability* adalah untuk meningkatkan desain dan menyediakan kepada tim desain berupa informasi untuk membantu menghilangkan gangguan atau kesalahan yang tidak perlu, serta mengurangi kesalahan sebelum produk mencapai pengembangan dan implementasi tahap akhir (Manzari & Trinidad-Christensen, 2006). Menurut Lynch dan Horton (2008) pengujian ini dapat dilakukan oleh pengembang, ahli antarmuka manusia dengan komputer atau perwakilan pengguna (Rinder, 2012). Menurut Nielsen dan Molich (1990) terdapat empat pendekatan untuk melakukan pengujian terhadap antarmuka situs web, berikut penjelasannya:

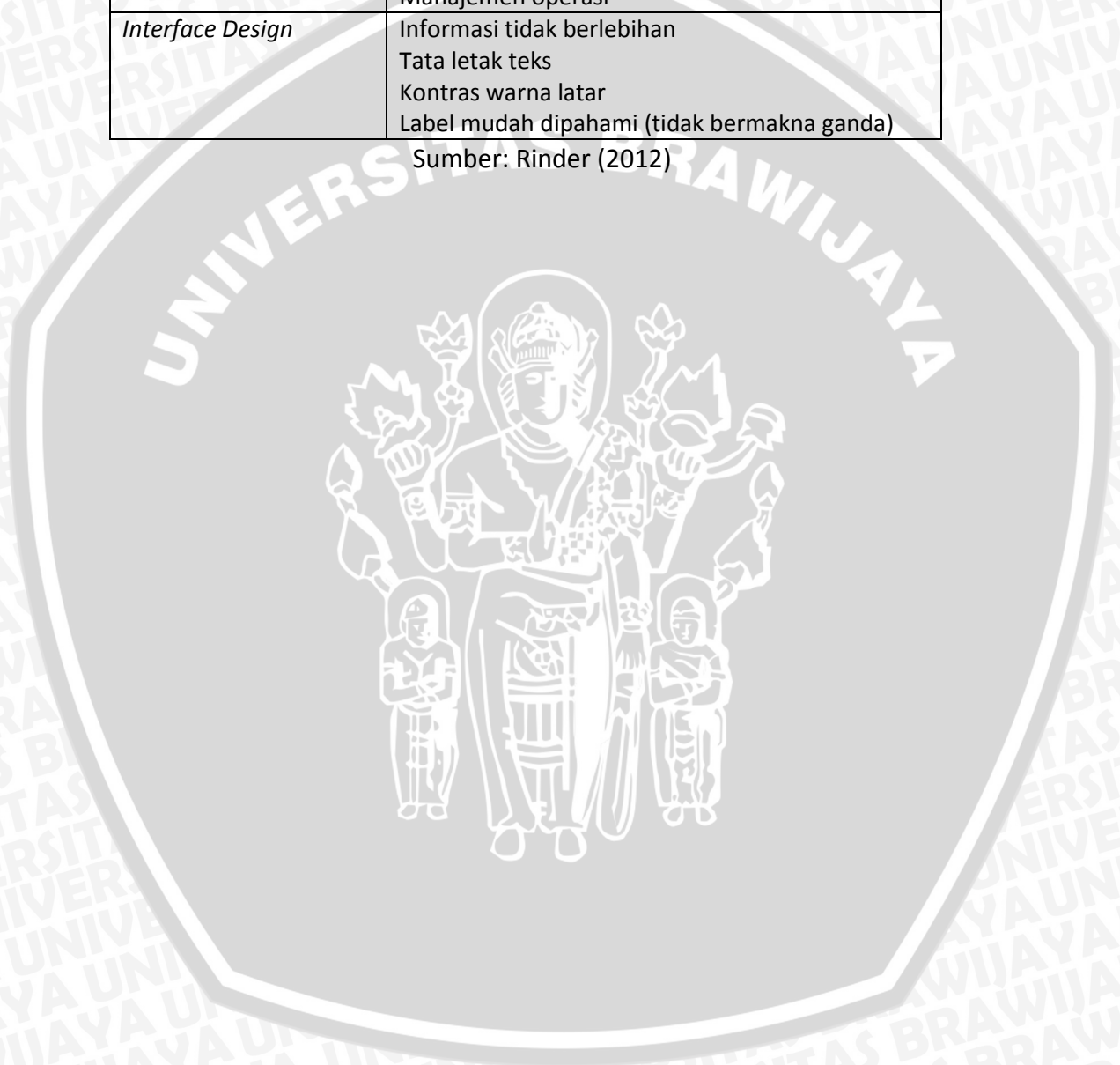
1. *Formally*, dengan beberapa teknik untuk menganalisis menggunakan model yang tepat dan rumus untuk menghitung langkah-langkah kegunaan (*usability*).
2. *Automatically*, dengan prosedur komputerisasi.
3. *Empirically*, melakukan eksperimen dengan pengguna sesungguhnya.
4. *Heuristically*, dengan melihat antarmuka dan memberikan penilaian menurut pendapat sendiri (Rinder, 2012).

Salah satu metode yang digunakan untuk evaluasi kegunaan (*usability*) adalah evaluasi inspeksi heuristik. Evaluasi ini menggunakan metode inspeksi untuk menemukan jenis masalah kegunaan tertentu dalam desain antarmuka pengguna (Rinder, 2012). Menurut Redish et al. (2002), metode ini memiliki kelebihan antara lain relatif tidak membutuhkan banyak biaya dalam proses evaluasinya, mudah untuk dipelajari, dapat dilakukan dengan segera tanpa harus merencanakan terlebih dahulu dan dapat dilaksanakan pada awal proses pembangunan (Rinder, 2012). Menurut Bolchini dan Garzotto (2007) evaluasi inspeksi heuristik terdiri dari empat kategori yaitu *content relevance*, *interface design*, *navigation/menus* dan *technical performance* (Rinder, 2012). Berikut penjelasan keempat kategori tersebut (Tabel 2.12).

Tabel 2.12 Kategori Inspeksi Heuristik

Kategori	Contoh
<i>Navigation</i>	Pola navigasi yang konsisten Kemudahan dalam menavigasi kembali tanpa menggunakan panah kembali
<i>Content</i>	Keakuratan teks Konsistensi multimedia
<i>Technical performance</i>	Reaksi sistem terhadap kesalahan pengguna Manajemen operasi
<i>Interface Design</i>	Informasi tidak berlebihan Tata letak teks Kontras warna latar Label mudah dipahami (tidak bermakna ganda)

Sumber: Rinder (2012)



BAB 3 METODOLOGI

Bab metodologi ini menguraikan tentang metode-metode yang digunakan untuk menyelesaikan penelitian ini dan mendapatkan hasil dari proses analisis yang dilakukan dalam penelitian tersebut. Metodologi penelitian yang digunakan terdiri dari Studi Literatur, Pengumpulan Data, Pengolahan Data, Rekayasa Persyaratan Sistem, Perancangan Sistem dan Prototipe, Evaluasi Persyaratan Sistem, Evaluasi Perancangan Sistem serta Dokumentasi dan Kesimpulan. Berikut alur dari metodologi penelitian (Gambar 3.1).



Gambar 3.1 Metodologi Penelitian

3.1 Studi Literatur

Studi literatur dilakukan untuk mengetahui bagaimana dasar-dasar pengertian komponen dari penelitian yang dilakukan. Memahami pengertian dari setiap komponen ini sangat penting karena dengan ini penelitian dapat dilakukan sesuai teori yang sudah ada. Studi literatur pada penelitian ini terdiri dari dua bagian yaitu dari penelitian sebelumnya yang sejenis dan dasar teori yang digunakan untuk penelitian. Dasar teori yang digunakan sebagai referensi terdiri dari kepadatan penduduk, sistem informasi geografis, teori *heat map*, sistem informasi, pemodelan untuk proses bisnis, rekayasa perangkat lunak, rekayasa persyaratan sistem, pemodelan sistem, analisis spasial pada data vektor serta metode untuk melakukan evaluasi pada persyaratan dan perancangan.

3.2 Pengumpulan Data

Pada proses pengumpulan data berasal dari data yang disimpan oleh pemerintah yaitu dari Badan Perencanaan dan Pembangunan Kota Malang, Dinas Pekerjaan Umum Perumahan dan Pengawasan Bangunan Kota Malang, Badan Pelayanan Perizinan Terpadu Kota Malang dan Badan Pusat Statistik Kota Malang. Data dari Badan Perencanaan dan Pembangunan dan Badan Pusat Statistik Kota Malang yaitu data tentang penduduk dan informasi seperti usia, jenis kelamin dan jumlah penduduk tiap kecamatan atau kelurahan. Data dari

Dinas Pekerjaan Umum Perumahan dan Pengawasan Bangunan Kota Malang adalah data berupa peta bangunan yang terdaftar di kota Malang. Lalu untuk data kepemilikan bangunan berasal dari Badan Pelayanan Perizinan Terpadu. Kemudian untuk data yang digunakan sebagai peta dasar yaitu berasal dari situs web *Open Street Map*.

3.3 Pengolahan Data

Pada proses pengolahan data akan dilakukan pendataan setiap bangunan dan mengelompokkan data penduduk berdasarkan kategorinya. Pengelompokan data penduduk ini akan dibagi lagi berdasarkan usia dan tempat tinggalnya yaitu kecamatan dan kelurahan. Kemudian data bangunan yang telah didapatkan akan diolah dan disesuaikan lokasi pada peta kota Malang serta diberikan informasi terkait pada masing-masing bangunan.

3.4 Rekayasa Persyaratan Sistem

Tahap analisis persyaratan merupakan tahapan untuk menguraikan seluruh persyaratan yang dibutuhkan untuk membangun sistem informasi kepadatan penduduk. Pada tahapan ini akan terdiri dari beberapa sub tahap yaitu:

1. Memahami alur proses bisnis sebelum dan sesudah pengembangan sistem
2. Memahami permasalahan sebelum adanya sistem dan penentuan pihak-pihak yang bertanggung jawab dan memiliki hak dalam pengembangan serta penggunaan sistem
3. Pendataan kebutuhan-kebutuhan dan bagaimana deskripsi produk yang akan dirancang
4. Pembuatan fitur-fitur dan persyaratan deskriptif sesuai daftar kebutuhan yang telah dibuat.
5. Pemodelan *use case* beserta deskripsi dari aktivitas aktor dalam menggunakan sistem dan hubungannya dan alur aktivitas pada setiap *use case* yang dibuat

3.5 Perancangan Sistem dan Prototipe

Setelah tahap analisis persyaratan dibuat, selanjutnya yaitu tahap perancangan sistem dan pembuatan prototipe. Pada tahap perancangan akan diuraikan tentang:

1. Pendataan kelas-kelas yang dibutuhkan beserta hubungan antar kelas-kelas tersebut. Pendataan kelas-kelas ini didasarkan pada analisis *use case* dan mekanisme analisis. Untuk mekanisme analisis berfokus pada kelas-kelas untuk persistensi.
2. Desain elemen-elemen yang terkandung dalam sistem seperti atribut dan perilaku tiap kelas serta pemodelan interaksi yang terjadi di dalam sistem

3. Pemodelan untuk penyimpanan data berupa *Entity Relationship Diagram*, perancangan tabel dan *Physical Data Model*
4. Perancangan antarmuka pengguna dan pembuatan prototipe antarmuka pengguna dari sistem.

3.6 Evaluasi

Evaluasi pada penelitian ini terdiri dari dua macam yaitu evaluasi untuk melakukan peninjauan keruntutan (*traceability*) dan evaluasi untuk peninjauan pada prototipe terhadap *use case* dan desain dasar antarmuka untuk pengguna. Berikut penjelasan untuk setiap evaluasi yang dilakukan.

1. Peninjauan keruntutan (*traceability*)

Peninjauan keruntutan atau *traceability* ini bertujuan untuk memeriksa keruntutan dari proses bisnis, kebutuhan pemangku kepentingan, fitur, persyaratan fungsional dan *use case*. Keruntutan ini diperiksa untuk memastikan keberadaan setiap proses pada analisis persyaratan telah disusun sesuai dengan kebutuhan pemangku kepentingan, dapat dilacak dan tidak mengandung kode yang duplikat. Terdapat dua aktivitas yang dilakukan pada peninjauan keruntutan ini yaitu:

- a. Membuat matriks keruntutan dengan mencantumkan kode-kode yang telah dibuat di dalam dokumen. Matriks keruntutan ini menyebutkan proses bisnis yang berhubungan, seluruh kebutuhan yang terdaftar, fitur, persyaratan fungsional dan *use case* yang kemudian diurutkan berdasarkan urutan kebutuhan sampai *use case*.
- b. Membuat daftar pertanyaan berisi tentang identifikasi persyaratan dan proses pelacakan setiap persyaratan yang bertujuan untuk pengecekan yang dilakukan oleh peneliti. Daftar pertanyaan ini mengacu pada beberapa pertanyaan untuk tinjauan spesifikasi persyaratan yang dibuat oleh Karl E. Wiegers (2001) dan Barbara Pfarr (2006) serta format penulisan daftarnya mengacu pada *Requirements Peer Review Checklist* oleh Barbara Pfarr (2006).

2. Peninjauan prototipe terhadap *use case* dan desain dasar antarmuka pengguna

Peninjauan pada prototipe ini bertujuan untuk memastikan kesesuaian fungsi berdasarkan kebutuhan pengguna dan jika terdapat kesalahan atau tidak sesuai dengan harapan pengguna, dapat digunakan sebagai pedoman dalam perbaikan desain sistem yang dikembangkan selanjutnya sekaligus mengacu pada saran dari pengguna pula. Terdapat dua aktivitas untuk melakukan peninjauan pada prototipe ini, yaitu:

- a. Peninjauan berdasarkan *use case* ini dilakukan dengan mengambil beberapa sampel *use case* utama kemudian membandingkan langkah-langkah yang dilakukan oleh pengguna saat menggunakan prototipe dengan alur dasar yang telah dibuat oleh peneliti. Peninjauan ini bertujuan untuk memeriksa kesesuaian alur yang

dilalui oleh pengguna dengan alur dasar yang terspesifikasi pada *use case*.

- b. Peninjauan berdasarkan penilaian dari pengguna dilakukan dengan cara menyediakan daftar pernyataan kepada pengguna terkait prototipe yang telah dibuat untuk kepuasan atau persetujuan serta kolom saran untuk diisi oleh pengguna. Daftar pernyataan ini mengacu pada pedoman pembuatan *website usability evaluation* dengan metode yang digunakan yakni evaluasi inspeksi heuristik.

3.7 Dokumentasi dan Kesimpulan

Dokumentasi akan menjelaskan seluruh aktivitas yang dilakukan dalam proses penelitian ini mulai dari rekayasa persyaratan, perancangan sistem dan hasil evaluasi dari persyaratan serta rancangan yang telah dibuat sebelumnya. Kemudian Kesimpulan yang dihasilkan dari aktivitas penelitian tersebut. Analisis serta rancangan dilakukan dengan teliti agar kesimpulan yang dihasilkan tepat akurat, sesuai dengan kebutuhan serta dapat membantu dalam hal pembangunan suatu sistem informasi kepadatan penduduk.

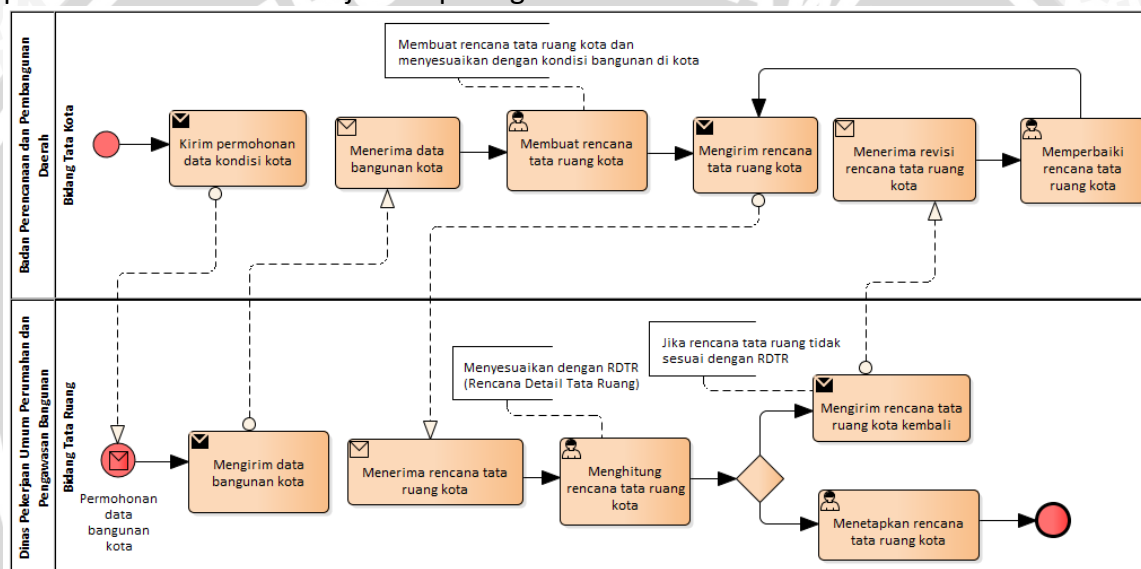


BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN

4.1 Proses Bisnis

4.1.1 Proses Bisnis Saat Ini

Proses pada pembuatan RDTR (Rencana Detail Tata Ruang) dilakukan oleh Bappeda (Badan Perencanaan dan Pembangunan Daerah) yang berhubungan dengan dinas pemerintah lain yaitu Dinas Pekerjaan Umum, Perumahan dan Pengawasan Bangunan disingkat menjadi DPUPPB. DPUPPB ini bertugas sebagai bagian teknis untuk melakukan survey bangunan dan perhitungan kesesuaian bangunan dan rencana detail tata ruang kota yang dibuat oleh Bappeda. Berikut proses bisnis saat ini ditunjukkan pada gambar 4.1 berikut ini.



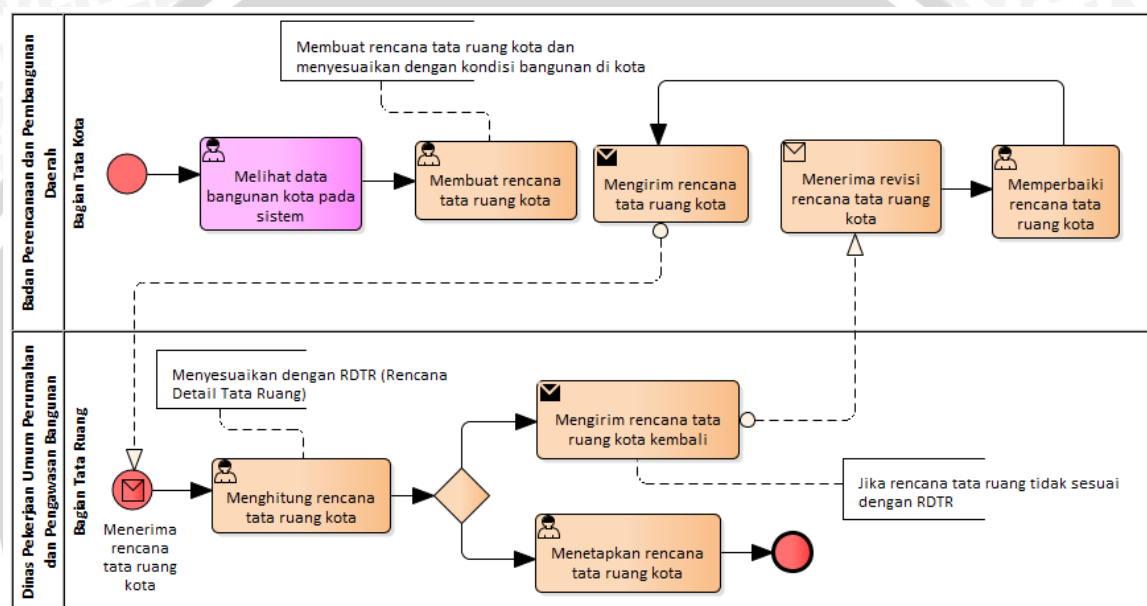
Gambar 4.1 Proses Bisnis Saat Ini

Proses dimulai dengan bidang Tata Kota pada Bappeda mengirimkan permintaan data kondisi kota yang berisi gambar-gambar bangunan kepada DPUPPB untuk digunakan sebagai landasan awal pembuatan rencana tata ruang kota. Kemudian DPUPPB bidang Tata Ruang mengirimkan data bangunan kota kepada Bappeda. Setelah menerima data bangunan, Bappeda bidang Tata Kota membuat rencana tata ruang kota yang dilakukan secara manual oleh pihak yang bertanggung jawab dalam pelaksanaan tersebut. Kemudian hasil pembuatan rencana tersebut dikirimkan kepada DPUPPB bidang Tata Ruang untuk dilakukan perhitungan yang disesuaikan dengan aturan pada RDTR (Rencana Detail Tata Ruang). Jika rencana yang dibuat sudah sesuai dan tidak melebihi dari aturan pada RDTR maka DPUPPB akan menetapkan data rencana tersebut yang kemudian akan digunakan sebagai pedoman pembangunan di kota. Namun jika masih ada rencana yang dibuat tidak sesuai atau melebihi dari RDTR maka DPUPPB bidang Tata Ruang akan mengirim revisi rencana tata ruang yang tidak sesuai dengan RDTR. Lalu Bappeda melakukan perbaikan kembali pada rencana

tata ruang kota tersebut, dikirim kembali kepada DPUPPB bidang Tata Ruang dan dilakukan perhitungan kembali sampai rencana yang dibuat sesuai dengan RDTR yang sudah ditetapkan.

4.1.2 Proses Bisnis yang Diusulkan

Kemudian untuk proses yang diusulkan setelah dibuatnya SIGEKA tiap bangunan yang menyajikan data peta bangunan dapat mengurangi tiga proses yaitu mengirimkan permohonan data, mengirim dan menerima data bangunan. Hal ini dapat mempercepat proses pembuatan rencana tata ruang kota oleh Bappeda bidang Tata Kota. Proses bisnis yang diusulkan ditampilkan pada gambar 4.2 di bawah ini.



Gambar 4.2 Proses Bisnis Yang Diusulkan

Proses bisnis yang diusulkan dimulai ketika Bappeda bidang Tata Kota melihat data bangunan kota pada sistem yang kemudian melakukan pembuatan rencana tata ruang kota. Setelah rencana tata ruang kota diselesaikan oleh pihak yang bertanggung jawab dalam pembuatan rencana tersebut, rencana tata ruang kota dikirim kepada DPUPPB bidang Tata ruang untuk dilakukan perhitungan sesuai dengan RDTR. Jika rencana yang dibuat sudah sesuai dan tidak melebihi dari aturan pada RDTR maka DPUPPB akan menetapkan data rencana tersebut yang kemudian akan digunakan sebagai pedoman pembangunan di kota. Namun jika masih ada rencana yang dibuat tidak sesuai atau melebihi dari RDTR maka DPUPPB bidang Tata Ruang akan mengirim revisi rencana tata ruang yang tidak sesuai dengan RDTR. Lalu Bappeda melakukan perbaikan kembali pada rencana tata ruang kota tersebut, setelah itu dikirim kembali kepada DPUPPB bidang Tata Ruang dan dilakukan perhitungan kembali sampai rencana yang dibuat sesuai dengan RDTR yang sudah ditetapkan.

Berikut ini kode aktivitas untuk proses bisnis yang diusulkan dengan memanfaatkan SIGEKA, dapat dilihat pada tabel 4.1. Kode aktivitas diberikan

format seperti PB-U-XXX yaitu PB (Proses Bisnis), U (Usulan) dan XXX (urutan aktivitas).

Tabel 4.1 Kode Aktivitas Proses Bisnis Usulan

Kode Aktivitas	Aktivitas	Keterangan
PB-U-001	Melihat data bangunan kota pada sistem	Dilakukan oleh staf di Bappeda bagian Tata Kota
PB-U-002	Membuat rencana tata ruang kota	Dilakukan oleh staf di Bappeda bagian Tata Kota dengan menyesuaikan rencana tata ruang kota berdasarkan kondisi bangunan di kota.
PB-U-003	Mengirim rencana tata ruang kota	Staf Tata Kota pada Bappeda mengirimkannya kepada DPUPPB bagian Tata Ruang
PB-U-004	Menghitung rencana tata ruang kota	Staf bagian Tata Ruang DPUPPB menghitung rencana tata ruang dari Bappeda dan menyesuainya dengan RDTR (Rencana Detail Tata Ruang)
PB-U-005	Mengirim rencana tata ruang kota kembali	Setelah perhitungan, jika terdapat ketidak sesuaian rencana tata ruang kota, maka akan dikirimkan kembali kepada Bappeda bagian Tata Kota untuk diperbaiki kembali
PB-U-006	Menerima revisi rencana tata ruang kota	Bappeda menerima revisi rencana tata ruang kota dari DPUPPB
PB-U-007	Memperbaiki rencana tata ruang kota	Staf Tata Kota Bappeda memperbaiki rencana tata ruang kota untuk dikirimkan kembali kepada DPUPPB
PB-U-008	Menetapkan rencana tata ruang kota	Setelah perhitungan, jika rencana tata ruang kota sudah sesuai dengan RDTR, maka DPUPPB bagian Tata Ruang akan menetapkan rencana tata ruang kota tersebut untuk dilakukan tindakan lebih lanjut.

4.2 Analisis Permasalahan

Analisis permasalahan pada penelitian ini berasal dari latar belakang masalah yang dikumpulkan sebelumnya. Analisis permasalahan dijelaskan pada tabel 4.2 di bawah ini.

Tabel 4.2 Analisis Permasalahan

Masalah	Belum adanya sistem yang menampilkan peta kota Malang dengan informasi jumlah penduduk pada setiap bangunan dan informasi tentang tingkat frekuensi kepadatan di kota
Mempengaruhi	Badan Pusat Statistik Kota Malang, Badan Perencanaan dan Pembangunan Daerah Kota Malang
Dampak Masalah	Pemahaman dimana saja lokasi di kota Malang yang padat penduduk dan kondisi kepadatan di kota Malang belum jelas sehingga pembangunan masih belum merata
Solusi yang berhasil dapat	• Menampilkan peta kota Malang beserta detail bangunan-bangunannya • Menyediakan informasi jumlah penduduk pada setiap bangunan • Menampilkan peta kota Malang dengan perbedaan warna pada setiap area berdasarkan tingkat frekuensi jumlah penduduk di setiap bangunan

4.3 Analisis Pemangku Kepentingan dan Pengguna

Pemangku kepentingan merupakan pihak yang berhubungan langsung terhadap penggunaan suatu sistem. Berikut penjelasan untuk analisis pemangku kepentingan dan pengguna pada sistem.

4.3.1 Tipe Pemangku Kepentingan

Berikut ini merupakan analisis untuk tipe pemangku kepentingan ditampilkan pada tabel 4.3.

Tabel 4.3 Tipe Pemangku Kepentingan

No.	Tipe	Deskripsi	Contoh
1	Pengguna	Orang yang langsung menggunakan sistem atau melakukan perawatan pada sistem	Masyarakat atau pengelola sistem
2	Pengembang	Beberapa orang tergabung dalam sebuah tim yang mengembangkan sistem dari mulai analisis kebutuhan hingga sistem jadi dan dapat digunakan oleh pengguna.	Analisis sistem, desainer antarmuka sistem dan programmer

Tabel 4.3 Tipe Pemangku Kepentingan (lanjutan)

No.	Tipe	Deskripsi	Contoh
3	Otoritas	Instansi pemerintahan yang bertanggung jawab sebagai pihak yang mengumpulkan serta menyimpan data terkait data bangunan dan penduduk	- DPUPPB Kota Malang Bagian Tata Ruang, - BP2T Kota Malang Bagian Izin Bangunan, - Bappeda Kota Malang bidang Tata Kota, - BPS Kota Malang Deputy Bidang Statistik Sosial

Tipe pemangku kepentingan pengguna bertanggung jawab untuk mengakses SIGEKA dan melakukan perubahan data yang dilakukan oleh pengelola sistem. Kemudian untuk tipe pengembang berperan sebagai pihak yang membuat sistem agar dapat dibangun sesuai dengan kebutuhan dan rancangannya. Dan tipe pemangku kepentingan yang terakhir adalah *authority* berperan sebagai pihak yang melakukan pendataan di lapangan dan menyimpan data-data tersebut agar dapat dimanfaatkan di masa yang akan datang.

4.3.2 Tipe Pengguna

Analisis tipe pengguna, karakteristik beserta contohnya dijelaskan pada tabel 4.4 di bawah ini.

Tabel 4.4 Tipe Pengguna

Tipe Pengguna	Karakteristik	Contoh
Masyarakat	Seluruh masyarakat kota Malang	Masyarakat kota Malang
Pengelola Sistem	Bekerja di bidang teknologi informasi	Pegawai di Badan Perencanaan dan Pembangunan Daerah

4.4 Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan para pemangku kepentingan dan pengguna dijelaskan pada tabel 4.5 berikut ini.

Tabel 4.5 Kebutuhan Pemangku Kepentingan dan Pengguna

No.	Kebutuhan	Deskripsi	Prioritas	Permasalahan	Solusi
Dinas dan Badan Pelayanan Pemerintah kota Malang					
1	Proses analisis	Mudahnya proses untuk analisis kondisi kepadatan penduduk di kota Malang	Must	Belum adanya sistem yang menyediakan informasi yang berisi jumlah penduduk tiap bangunan dan pembeda	Menyediakan sistem yang berisi informasi jumlah penduduk tiap bangunan lalu direpresentasikan dengan metode <i>heatmap</i> yaitu menampilkan tingkat frekuensi warna yang berbeda menurut jumlah penduduk supaya mudah dalam pemahaman untuk analisis kepadatan penduduk
Pengelola Sistem / Administrator					
2	Memperbarui informasi jumlah penduduk	Mendapatkan media untuk memperbarui informasi tentang jumlah penduduk dan kondisi kepadatan penduduk di kota Malang	Must	Belum ada media untuk mengelola data kepadatan penduduk	Menyediakan sistem yang memiliki informasi tentang jumlah penduduk dan informasi tersebut selalu yang paling baru (<i>up-to-date</i>)

Tabel 4.5 Kebutuhan Pemangku Kepentingan dan Pengguna (lanjutan)

No.	Kebutuhan	Deskripsi	Prioritas	Permasalahan	Solusi
Masyarakat					
3	Info jumlah penduduk	Masyarakat mendapatkan informasi lengkap tentang jumlah penduduk pada tiap bangunan di kota Malang	Must	Informasi jumlah penduduk masih dalam area luas, belum detail setiap bangunan	Informasi setiap bangunan dengan jumlah penduduk di dalamnya berupa peta
4	Daerah padat penduduk	Masyarakat dapat memahami daerah mana saja yang termasuk padat penduduk atau tidak	Must	Informasi kepadatan berupa tabel yang sulit pemahamannya	Informasi kepadatan penduduk direpresentasikan dengan tingkat frekuensi warna yang berbeda menurut jumlah penduduknya
5	Detail info	Memudahkan untuk memahami informasi detail setiap bangunan	Must	Informasi detail bangunan dan lokasi bangunan masih merupakan data terpisah pada dinas pemerintah yang berbeda	Informasi detail ditampilkan pada setiap bangunan yang ada
6	Perbandingan jumlah penduduk	Masyarakat dapat mengetahui perbandingan jumlah penduduk berdasarkan jenis kelamin	Must	Belum ada hasil data perbandingan jumlah penduduk menurut jenis kelamin	Menampilkan perbandingan jumlah penduduk menurut jenis kelamin berupa grafik piramida penduduk

4.5 Kedudukan Produk

Penjelasan kedudukan produk pada sistem yang diusulkan yaitu SIGEKA dapat dilihat pada tabel 4.6 di bawah ini.

Tabel 4.6 Pernyataan Kedudukan Produk

Untuk	Masyarakat kota Malang
Siapa saja yang	Membutuhkan informasi tentang kondisi kepadatan penduduk dan bangunan di kota Malang
Berupa	Sistem Informasi Geografis Kepadatan Penduduk merupakan sebuah sistem yang memvisualisasikan informasi dengan tampilan peta
Bahwa	Menampilkan peta dengan informasi jumlah penduduk pada setiap bangunan yang divisualisasikan berupa <i>heatmap</i> , informasi jumlah penduduk berdasarkan usia dan jenis kelamin tiap kecamatan dan kelurahan serta informasi kepadatan penduduk tiap kecamatan dan kelurahan
Tidak seperti	Data yang disimpan oleh pemerintah berupa data tabel jumlah penduduk dan bangunan kota
Produk kami	Memvisualisasikan data tabel dari pemerintah berupa peta dan dapat diakses oleh seluruh kalangan masyarakat

4.6 Fitur-Fitur

Fitur-fitur pada SIGEKA memiliki prioritas yang menjelaskan seberapa pentingnya kebutuhan stakeholder yang dibuat pada persyaratan sistem. Prioritas ini mengacu pada aturan MoSCoW yang berasal dari huruf pertama dari kriteria prioritas tersebut (Bittner & Spence, 2002). Prioritas yang digunakan adalah prioritas *Must* karena setiap kebutuhan yang dituliskan harus dapat dipenuhi oleh sistem. Berikut tabel yang menjelaskan fitur-fitur sistem, deskripsi beserta prioritasnya (Tabel 4.6).

Tabel 4.7 Fitur-fitur

Kode / Identifier	Nama Fitur	Deskripsi	Prioritas
FITUR1	Tampilan Peta	Sistem menampilkan peta bangunan kota Malang beserta <i>tools</i> umum yang harus digunakan untuk menampilkan peta	<i>Must</i>
FITUR2	<i>Heat Map</i>	Sistem menampilkan <i>heatmap</i>	<i>Must</i>

Tabel 4.7 Fitur-fitur (lanjutan)

Kode / Identifier	Nama Fitur	Deskripsi	Prioritas
FITUR3	Grafik Penduduk	Sistem menyediakan tampilan grafik jumlah penduduk	Must
FITUR4	Kepadatan Penduduk	Sistem menampilkan tingkat kepadatan penduduk	Must
FITUR5	Pencarian Bangunan	Sistem menyediakan kotak pencarian bangunan	Must
FITUR6	Informasi Bangunan	Sistem menampilkan detail dari setiap bangunan	Must
FITUR7	Ubah Data Bangunan	Sistem menyediakan form untuk mengubah data yang hanya dapat diakses oleh Administrator	Must
FITUR8	Compatibility	Sistem dapat berjalan pada beberapa browser yang berbeda seperti Google Chrome, Mozilla Firefox, Opera Mini dan Safari.	Must

4.7 Persyaratan secara Deklaratif

Persyaratan yang dispesifikasikan secara deklaratif untuk SIGEKA ini akan dibedakan menjadi dua kebutuhan yaitu kebutuhan fungsional dan non fungsional. Pada setiap kebutuhan yang dituliskan akan diberikan penomoran untuk memberikan identitas pada setiap kebutuhannya. Berikut (Tabel 4.8) merupakan aturan penomoran untuk kebutuhan fungsional dan non fungsional:

F-SIGEKA-XXX / NF-SIGEKA-XXX

Tabel 4.8 Aturan Penomoran Persyaratan Deskriptif

Bagian Kode	Deskripsi
F / NF	merepresentasikan kebutuhan Fungsional / Non Fungsional
SIGEKA	singkatan dari Sistem Informasi Geografis kEpadatan penduduk kota malAng
XXX	menunjukkan urutan persyaratan

4.7.1 Persyaratan Fungsional

Pada tabel 4.9 merupakan daftar persyaratan fungsional untuk sistem informasi kepadatan penduduk.

Tabel 4.9 Persyaratan Fungsional

Kode Fitur	Kode Dasar Fitur	Kode Lengkap Fungsional	Deskripsi
FITUR1	F-SIGEKA-001	F-SIGEKA-001-1	Pengunjung dapat melihat peta kota Malang
		F-SIGEKA-001-2	Pengunjung dapat mengatur ukuran tampilan peta yang terdiri dari: - memperbesar dan memperkecil tampilan peta (tampilan peta lebih dekat) - memperkecil tampilan peta (tampilan peta lebih jauh atau luas)
		F-SIGEKA-001-3	Pengunjung dapat melihat peta bangunan berdasarkan kategori fungsi yang terdiri dari fungsi hunian, industri, jasa, kesehatan, pemerintahan, pendidikan, perdagangan dan wisata.
FITUR2	F-SIGEKA-002	F-SIGEKA-002-1	Pengunjung dapat melihat <i>heatmap</i> menurut jumlah penduduk tiap bangunan
FITUR3	F-SIGEKA-003	F-SIGEKA-003-1	Pengunjung dapat melihat grafik jumlah penduduk tiap kecamatan di kota Malang
		F-SIGEKA-003-2	Pengunjung dapat melihat grafik jumlah penduduk tiap kelurahan di kota Malang
FITUR4	F-SIGEKA-004	F-SIGEKA-004-1	Sistem menampilkan tingkat kepadatan penduduk tiap kecamatan
		F-SIGEKA-004-2	Sistem menampilkan tingkat kepadatan penduduk tiap kelurahan
FITUR5	F-SIGEKA-005	F-SIGEKA-005-1	Pengunjung dapat mencari bangunan yang disediakan oleh sistem

Tabel 4.9 Persyaratan Fungsional (lanjutan)

Kode Fitur	Kode Dasar Fitur	Kode Lengkap Fungsi	Deskripsi
FITUR6	F-SIGEKA-006	F-SIGEKA-006-1	Pengunjung dapat melihat detail bangunan yang terdiri dari Alamat, Nama bangunan, Fungsi bangunan, Jumlah penghuni
		F-SIGEKA-006-2	Pengunjung dapat melihat skala peta yang dibandingkan dengan ukuran sebenarnya
		F-SIGEKA-006-3	Pengunjung dapat memahami arti simbol-simbol yang ditampilkan pada peta
		F-SIGEKA-006-4	Pengunjung dapat mengetahui arah utara pada peta
FITUR7	F-SIGEKA-007	F-SIGEKA-007-1	Administrator mengubah data pada tiap bangunan yang ditampilkan sistem

4.7.2 Persyaratan Non Fungsional

Persyaratan non fungsional pada penelitian ini tidak diuji namun dideskripsikan pada tabel 4.10 di bawah ini.

Tabel 4.10 Persyaratan Non Fungsional

Kode	Nama	Deskripsi
NF-SIGEKA-001	<i>Compatibility</i>	Sistem dapat berjalan pada beberapa <i>browser</i> yang berbeda seperti <i>Google Chrome, Mozilla Firefox, Opera Mini</i> dan <i>Safari</i> .

4.8 Model Use Case

Model *use case* mendeskripsikan interaksi antar sistem dengan lingkungannya (aktor atau sistem lainnya). Pada *use case* model terdapat dua komponen utama yaitu *use case* dan aktor. *Use case* merupakan urutan tindakan sistem yang memberikan nilai atau efek kepada aktor. Sementara definisi untuk aktor adalah seperangkat pengguna yang memiliki masing-masing peran terhadap sistem saat berinteraksi dengan *use case*.

4.8.1 Aktor

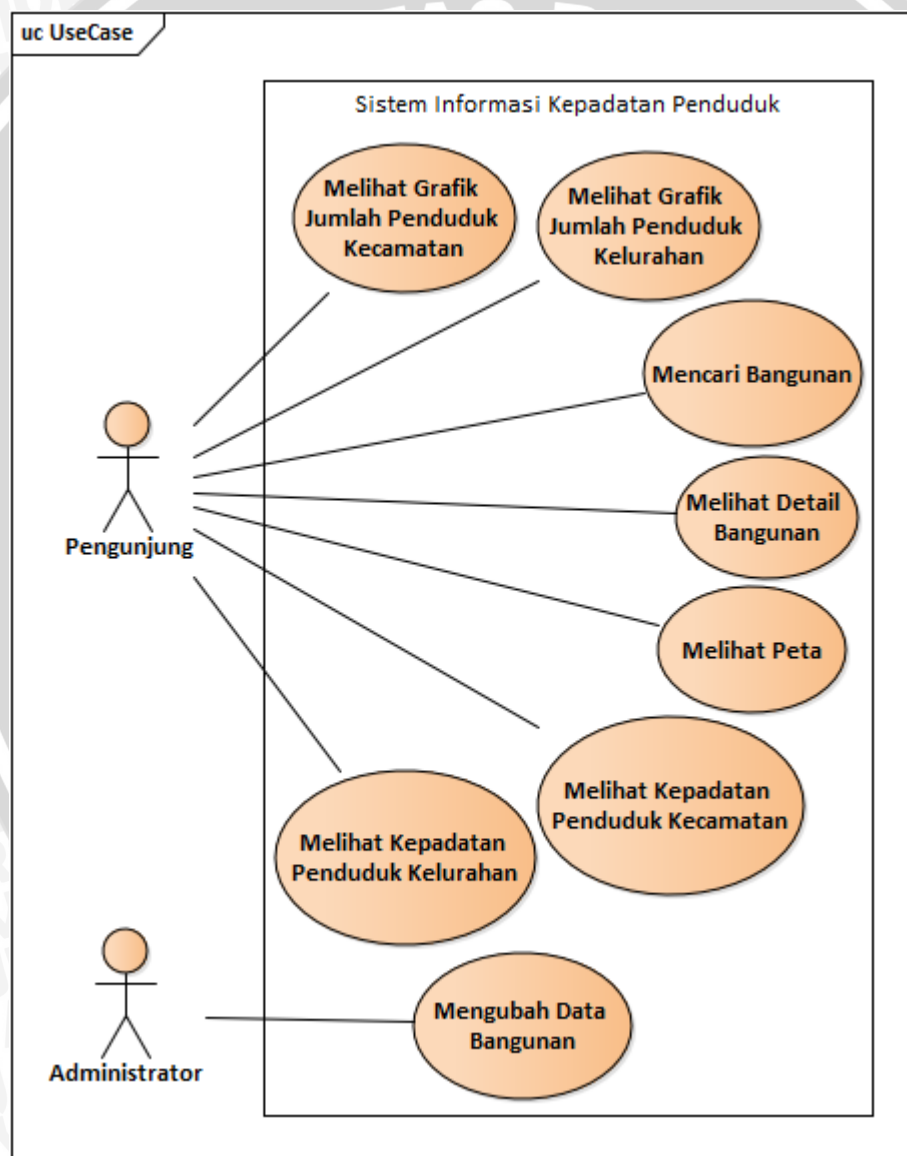
Penjelasan aktor atau pengguna yang berhubungan dengan sistem dapat dilihat pada Tabel 4.11 berikut ini.

Tabel 4.11 Penjelasan Aktor

Aktor	Penjelasan
Pengunjung	Masyarakat khususnya masyarakat kota Malang yang mengakses sistem
Administrator	Orang yang nantinya akan melakukan perawatan pada sistem

4.8.2 Diagram Use Case

Diagram *Use Case* untuk SIGEKA ini ditunjukkan pada gambar 4.3 di bawah ini.



Gambar 4.3 Diagram Use Case

Kemudian kode pada setiap *use case* yang disesuaikan dengan kode kebutuhan yang telah dituliskan sebelumnya dijelaskan pada tabel 4.12 berikut ini.

Tabel 4.12 Kode Use Case

Kode Kebutuhan	Kode Use Case	Use Case	Aktor
F-SIGEKA-003-1	UC-001	Melihat Grafik Jumlah Penduduk Kecamatan	Pengunjung
F-SIGEKA-003-2	UC-002	Melihat Grafik Jumlah Penduduk Kelurahan	Pengunjung
F-SIGEKA-005-1	UC-003	Mencari Bangunan	Pengunjung
F-SIGEKA-006-1	UC-004	Melihat Detail Bangunan	Pengunjung
F-SIGEKA-001-1	UC-005	Melihat Peta	Pengunjung
F-SIGEKA-001-2			Pengunjung
F-SIGEKA-001-3			Pengunjung
F-SIGEKA-002-1			Pengunjung
F-SIGEKA-006-2			Pengunjung
F-SIGEKA-006-3			Pengunjung
F-SIGEKA-006-4			Pengunjung
F-SIGEKA-004-1	UC-006	Melihat Kepadatan Penduduk Kecamatan	Pengunjung
F-SIGEKA-004-2	UC-007	Melihat Kepadatan Penduduk Kelurahan	Pengunjung
F-SIGEKA-007-1	UC-008	Mengubah Data Bangunan	Administrator

4.8.3 Spesifikasi Use Case

Berikut ini merupakan spesifikasi *use case* beserta spesifikasi alternatif jika tersedia.

4.8.3.1 Melihat Grafik Jumlah Penduduk Kecamatan

Tabel 4.13 Spesifikasi Use Case Melihat Grafik Jumlah Penduduk Kecamatan

<i>Brief Description</i>	Pengunjung dapat melihat grafik jumlah penduduk pada setiap kecamatan di kota Malang.
<i>Preconditions</i>	Sistem menampilkan peta kota Malang pada Sistem Informasi Kepadatan Penduduk

Tabel 4.13 Spesifikasi Use Case Melihat Grafik Jumlah Penduduk Kecamatan (lanjutan)

<i>Basic Flow</i>	1. Pengunjung memilih opsi grafik penduduk 2. Sistem menampilkan grafik jumlah penduduk tiap kecamatan di kota Malang {Use case Berakhir} 3. Use case berakhir.
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Pengunjung melihat informasi yang ditampilkan pada grafik jumlah penduduk tiap kecamatan.

Glossary

Informasi yang dimaksud adalah informasi berupa grafik jumlah penduduk berdasarkan usia dan jenis kelamin.

4.8.3.2 Melihat Grafik Jumlah Penduduk Kelurahan

Tabel 4.14 Spesifikasi Use Case Melihat Grafik Jumlah Penduduk Kelurahan

<i>Brief Description</i>	Pengunjung dapat melihat grafik jumlah penduduk pada setiap kelurahan di kota Malang.
<i>Preconditions</i>	Sistem menampilkan peta kota Malang pada Sistem Informasi Kepadatan Penduduk
<i>Basic Flow</i>	1. Pengunjung memilih opsi grafik penduduk 2. Sistem menampilkan grafik jumlah penduduk tiap kecamatan di kota Malang 3. Pengunjung memilih opsi kelurahan 4. Sistem menampilkan grafik jumlah penduduk tiap kelurahan di kota Malang {Use case berakhir} 5. Use case berakhir
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Pengunjung melihat informasi yang ditampilkan pada grafik jumlah penduduk tiap kelurahan.

Glossary

Informasi yang dimaksud adalah informasi berupa grafik jumlah penduduk berdasarkan usia dan jenis kelamin.

4.8.3.3 Mencari Bangunan [Masukan kata ditemukan]

Tabel 4.15 Spesifikasi Use Case Mencari Bangunan jika Masukan Kata Ditemukan

<i>Brief Description</i>	- Pengunjung dapat mencari bangunan yang ditampilkan oleh sistem informasi geografis kepadatan penduduk di kota Malang. - Pengunjung sudah mengakses Sistem informasi geografis kepadatan penduduk kota Malang.
<i>Preconditions</i>	Sistem menampilkan peta kota Malang pada Sistem informasi geografis kepadatan penduduk kota Malang.
<i>Basic Flow</i>	{Memasukkan kata} - Use case dimulai ketika pengunjung memasukkan kata yang berhubungan dengan bangunan yang akan dicari misalnya nama bangunan atau fungsi bangunan. {Memeriksa kata} - Sistem memeriksa pada penyimpanan untuk mencari kata yang dimasukkan. - Sistem menampilkan hasil pencarian dengan menampilkan bangunan yang dicari. {Use case berakhir} - Use case berakhir.
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Sistem menampilkan hasil dari pencarian berupa bangunan yang dicari.

Glossary

Fungsi bangunan yang dimaksud adalah fungsi suatu bangunan dalam kegunaannya yang terdiri dari hunian, industri, jasa, kesehatan, pemerintahan, pendidikan, perdagangan dan wisata.

Hasil pencarian yang dimaksud berupa tampilan bangunan yang disorot atau diberikan warna yang berbeda dari bangunan lainnya.

4.8.3.4 Mencari Bangunan [Masukan kata tidak ditemukan]

Tabel 4.16 Spesifikasi Use case Mencari Bangunan jika Masukan Kata Tidak Ditemukan

<i>Brief Description</i>	- Pengunjung dapat mencari bangunan yang ditampilkan oleh sistem informasi geografis kepadatan penduduk di kota Malang. - Pengunjung sudah mengakses Sistem informasi geografis kepadatan penduduk kota Malang.
<i>Preconditions</i>	Sistem menampilkan peta kota Malang pada Sistem informasi geografis kepadatan penduduk kota Malang.

Tabel 4.16 Spesifikasi Use case Mencari Bangunan jika Masukan Kata Tidak Ditemukan (lanjutan)

<i>Basic Flow</i>	{Memasukkan kata} 1. Use case dimulai ketika pengunjung memasukkan kata yang berhubungan dengan bangunan yang akan dicari misalnya nama bangunan atau fungsi bangunan. {Memeriksa kata} 2. Sistem memeriksa pada penyimpanan untuk mencari kata yang dimasukkan. 3. Sistem menampilkan hasil pencarian dengan menampilkan bangunan yang dicari. {Use case Berakhir} 4. Use case berakhir.
<i>Alternative Flows</i>	A1. Kata yang dimasukkan tidak ditemukan Pada {Memeriksa kata} jika kata yang dimasukkan tidak ditemukan, 1. Pengunjung harus memasukkan kata lain yang sejenis dengan mengulang kembali basic flow {Memasukkan kata} .
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Pengunjung melihat hasil pencarian bangunan.

Glossary

Fungsi bangunan yang dimaksud adalah fungsi suatu bangunan dalam kegunaannya yang terdiri dari hunian, industri, jasa, kesehatan, pemerintahan, pendidikan, perdagangan dan wisata.

Hasil pencarian yang dimaksud berupa tampilan bangunan yang disorot atau diberikan warna yang berbeda dari bangunan lainnya.

4.8.3.5 Melihat Detail Bangunan

Tabel 4.17 Spesifikasi Use case Melihat Detail Bangunan

<i>Brief Description</i>	Pengunjung melihat dan memahami informasi tentang detail pada setiap atribut bangunan.
<i>Preconditions</i>	Sistem menampilkan peta bangunan di kota Malang.
<i>Basic Flow</i>	{Memilih Bangunan} 1. Pengunjung memilih bangunan yang akan dilihat detail bangunannya 2. Sistem menampilkan detail bangunan yang dipilih {Use case Berakhir} 2. Use case Berakhir
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Pengunjung melihat dan membaca detail suatu bangunan yang dipilih.

Glossary

Detail bangunan yang dimaksud terdiri dari nama bangunan, alamat, fungsi bangunan dan jumlah penghuni dalam bangunan.

4.8.3.6 Melihat Peta

Tabel 4.18 Spesifikasi Use case Melihat Peta

<i>Brief Description</i>	Pengunjung melihat dan memahami informasi yang ditampilkan pada peta sistem informasi kepadatan penduduk
<i>Preconditions</i>	Pengunjung mengakses Sistem informasi geografis kepadatan penduduk di Kota Malang.
<i>Basic Flow</i>	1. <i>Use case</i> dimulai ketika sistem menampilkan informasi tentang peta kota Malang {Kategori} 2. Menjalankan subflow Memilih kategori fungsi 3. Sistem menampilkan peta kota Malang dengan tampilan menurut kategori yang dipilih {Heat map} 4. Pengunjung memilih opsi untuk tampilan <i>heat map</i> 5. Sistem menampilkan peta kota Malang dengan <i>heat map</i> berdasarkan jumlah penghuni tiap bangunan 6. Menjalankan subflow Tampil heat map {Zoom} 7. Menjalankan subflow Memperbesar peta atau Memperkecil peta 8. Sistem menampilkan peta sesuai ukuran yang diinginkan oleh pengunjung {Use case Berakhir} 9. <i>Use case</i> berakhir
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	S1. Memilih kategori fungsi 1. Pengunjung memilih opsi berdasarkan fungsi bangunan untuk tampilan peta 2. Melanjutkan ke basic flow {Heat Map} S2. Memperbesar peta 1. Pengunjung memperbesar tampilan peta supaya peta terlihat lebih dekat 2. Melanjutkan ke basic flow {Use case Berakhir} S3. Memperkecil peta 1. Pengunjung memperkecil tampilan peta supaya peta terlihat lebih luas 2. Melanjutkan ke basic flow {Use case Berakhir}
<i>Postconditions</i>	Sistem menampilkan peta sesuai dengan keinginan pengunjung.

Glossary

Kategori yang dimaksud adalah kategori berupa fungsi bangunan dalam kegunaannya yang terdiri dari hunian, industri, jasa, kesehatan, pemerintahan, pendidikan, perdagangan dan wisata.

4.8.3.7 Melihat Kepadatan Penduduk Kecamatan

Tabel 4.19 Spesifikasi Use case Melihat Kepadatan Penduduk Kecamatan

<i>Brief Description</i>	Pengunjung dapat melihat nilai kepadatan penduduk tiap kecamatan di kota Malang.
<i>Preconditions</i>	Sistem menampilkan peta kota Malang.
<i>Basic Flow</i>	1. <i>Use case</i> dimulai ketika pengunjung memilih opsi kepadatan penduduk 2. Sistem menampilkan nilai kepadatan penduduk tiap kecamatan di kota Malang {Use case Berakhir} 3. <i>Use case</i> berakhir.
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Pengunjung melihat informasi yang ditampilkan berupa nilai kepadatan penduduk tiap kecamatan.

Glossary

Nilai kepadatan yang dimaksud adalah berupa persentase tingkat kepadatan penduduk dalam wilayah tersebut (kecamatan yang ditunjuk).

4.8.3.8 Melihat Kepadatan Penduduk Kelurahan

Tabel 4.20 Spesifikasi Use case Melihat Kepadatan Penduduk Kelurahan

<i>Brief Description</i>	Pengunjung dapat melihat nilai kepadatan penduduk tiap kelurahan di kota Malang.
<i>Preconditions</i>	Sistem menampilkan peta kota Malang.
<i>Basic Flow</i>	1. <i>Use case</i> dimulai ketika pengunjung memilih opsi kepadatan penduduk 2. Sistem menampilkan nilai kepadatan penduduk tiap kecamatan di kota Malang 3. Pengunjung memilih opsi kelurahan 4. Sistem menampilkan nilai kepadatan penduduk tiap kelurahan di kota Malang {Use case Berakhir} 5. <i>Use case</i> berakhir.
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	Tidak ada.
<i>Postconditions</i>	Pengunjung melihat informasi yang ditampilkan berupa nilai kepadatan penduduk tiap kelurahan.

Glossary

Nilai kepadatan yang dimaksud adalah berupa persentase tingkat kepadatan penduduk dalam wilayah tersebut (kelurahan yang ditunjuk).

4.8.3.9 Mengubah Data Bangunan [Password benar dan masukan data sesuai format sistem]

Tabel 4.21 Spesifikasi Use case Mengubah Data Bangunan jika password benar dan masukan data sesuai format sistem

<i>Brief Description</i>	Administrator mengubah data informasi detail setiap bangunan
<i>Preconditions</i>	Administrator mengakses sistem untuk melakukan autentifikasi pengguna
<i>Basic Flow</i>	{Autentifikasi Pengguna} 1. Use case dimulai ketika sistem meminta pengguna untuk memasukkan password keamanan 2. Administrator memasukkan password {Menampilkan peta} 3. Sistem menampilkan peta bangunan di kota Malang {Pilih Bangunan} 4. Administrator memilih bangunan untuk diubah datanya 5. Menjalankan subflow Ubah Data {Use case berakhir} 6. Use case berakhir
<i>Alternative Flows</i>	Tidak ada.
<i>Subflows</i>	S1. Ubah Data 1. Sistem memberikan tampilan untuk mengubah data 2. Administrator mengubah data bangunan {Validasi Data} 3. Administrator menyimpan data 4. Melanjutkan ke basic flow {Menampilkan peta}
<i>Postconditions</i>	Sistem telah menyimpan data yang paling baru

Glossary

Password yang dimaksud adalah kata sandi untuk mengakses tampilan peta untuk mengubah data.

Data bangunan yang dimaksud terdiri dari nama bangunan, alamat, fungsi bangunan dan jumlah penghuni dalam bangunan.

4.8.3.10 Mengubah Data Bangunan [Password salah dan masukan data sesuai format sistem]

Tabel 4.22 Spesifikasi Use case Mengubah Data Bangunan jika password salah dan masukan data sesuai format sistem

<i>Brief Description</i>	Administrator mengubah data informasi detail setiap bangunan
<i>Preconditions</i>	Administrator mengakses sistem untuk melakukan autentifikasi pengguna
<i>Basic Flow</i>	{Autentifikasi Pengguna} 1. Use case dimulai ketika sistem meminta pengguna untuk memasukkan password keamanan 2. Administrator memasukkan password

Tabel 4.22 Spesifikasi Use case Mengubah Data Bangunan jika password salah dan masukan data sesuai format sistem (lanjutan)

	{Menampilkan peta} 3. Sistem menampilkan peta bangunan di kota Malang {Pilih Bangunan} 4. Administrator memilih bangunan untuk diubah datanya 5. Menjalankan subflow Ubah Data {Use case berakhir} Use case berakhir
Alternative Flows	A1. Password yang dimasukkan salah Pada {Autentifikasi Pengguna} jika password yang dimasukkan oleh administrator salah, 1. Administrator harus memasukkan password lagi yang sesuai dengan password yang disimpan oleh sistem 2. Melanjutkan ke basic flow {Menampilkan peta}
Subflows	S1. Ubah Data 1. Sistem memberikan tampilan untuk mengubah data 2. Administrator mengubah data bangunan {Validasi Data} 2. Administrator menyimpan data 3. Melanjutkan ke basic flow {Menampilkan peta}
Postconditions	Sistem telah menyimpan data yang paling baru

Glossary

Password yang dimaksud adalah kata sandi untuk mengakses tampilan peta untuk mengubah data.

Data bangunan yang dimaksud terdiri dari nama bangunan, alamat, fungsi bangunan dan jumlah penghuni dalam bangunan.

4.8.3.11 Mengubah Data Bangunan [Password kosong dan masukan data sesuai format sistem]

Tabel 4.23 Spesifikasi Use case Mengubah Data Bangunan jika password kosong dan masukan data sesuai format sistem

Brief Description	Administrator mengubah data informasi detail setiap bangunan
Preconditions	Administrator mengakses sistem untuk melakukan autentifikasi pengguna
Basic Flow	{Autentifikasi Pengguna} 1. Use case dimulai ketika sistem meminta pengguna untuk memasukkan password keamanan 2. Administrator memasukkan password {Menampilkan peta} 3. Sistem menampilkan peta bangunan di kota Malang {Pilih Bangunan} 4. Administrator memilih bangunan untuk diubah datanya 5. Menjalankan subflow Ubah Data {Use case berakhir} 6. Use case berakhir

Tabel 4.23 Spesifikasi Use case Mengubah Data Bangunan jika password kosong dan masukan data sesuai format sistem (lanjutan)

<i>Alternative Flows</i>	A1. Password tidak dimasukkan atau kosong Pada {Autentifikasi Pengguna} jika administrator tidak memasukkan password atau password kosong, 1. Administrator memasukkan password yang benar 2. Melanjutkan ke basic flow {Menampilkan peta}
<i>Subflows</i>	S1. Ubah Data 1. Sistem memberikan tampilan untuk mengubah data 2. Administrator mengubah data bangunan {Validasi Data} 1. Administrator menyimpan data 2. Melanjutkan ke basic flow {Menampilkan peta}
<i>Postconditions</i>	Sistem telah menyimpan data yang paling baru

Glossary

Password yang dimaksud adalah kata sandi untuk mengakses tampilan peta untuk mengubah data.

Data bangunan yang dimaksud terdiri dari nama bangunan, alamat, fungsi bangunan dan jumlah penghuni dalam bangunan.

4.8.3.12 Mengubah Data Bangunan [Password benar dan Memasukkan data tidak sesuai format]

Tabel 4.24 Spesifikasi Use case Mengubah Data Bangunan jika password benar dan Memasukkan data tidak sesuai format

<i>Brief Description</i>	Administrator mengubah data informasi detail setiap bangunan
<i>Preconditions</i>	Administrator mengakses sistem untuk melakukan autentifikasi pengguna
<i>Basic Flow</i>	{Autentifikasi Pengguna} 1. Use case dimulai ketika sistem meminta pengguna untuk memasukkan password keamanan 2. Administrator memasukkan password {Menampilkan peta} 3. Sistem menampilkan peta bangunan di kota Malang {Pilih Bangunan} 4. Administrator memilih bangunan untuk diubah datanya 5. Menjalankan subflow Ubah Data {Use case berakhir} 6. Use case berakhir
<i>Alternative Flows</i>	A3. Data yang dimasukkan tidak sesuai dengan format yang dikenali sistem Pada {Validasi Data} jika data yang dimasukkan oleh administrator tidak sesuai dengan format yang dikenali oleh sistem, 1. Administrator harus memasukkan data lagi sesuai dengan format yang dikenali oleh sistem 2. Menjalankan subflow Ubah Data

Tabel 4.24 Spesifikasi Use case Mengubah Data Bangunan jika *password* benar dan Memasukkan data tidak sesuai format (lanjutan)

<i>Subflows</i>	<i>S1. Ubah Data</i> 1. Sistem memberikan tampilan untuk mengubah data 2. Administrator mengubah data bangunan {Validasi Data} 1. Administrator menyimpan data 2. Melanjutkan ke basic flow {Menampilkan peta}
<i>Postconditions</i>	Sistem telah menyimpan data yang paling baru

Glossary

Password yang dimaksud adalah kata sandi untuk mengakses tampilan peta untuk mengubah data.

Data bangunan yang dimaksud terdiri dari nama bangunan, alamat, fungsi bangunan dan jumlah penghuni dalam bangunan.

4.8.3.13 Mengubah Data Bangunan [Password benar dan Memasukkan Data kosong]

Tabel 4.25 Spesifikasi Use case Mengubah Data Bangunan jika *password* benar dan Memasukkan Data kosong

<i>Brief Description</i>	Administrator mengubah data informasi detail setiap bangunan
<i>Preconditions</i>	Administrator mengakses sistem untuk melakukan autentifikasi pengguna
<i>Basic Flow</i>	{Autentifikasi Pengguna} 1. <i>Use case</i> dimulai ketika sistem meminta pengguna untuk memasukkan password keamanan 2. Administrator memasukkan password {Menampilkan peta} 3. Sistem menampilkan peta bangunan di kota Malang {Pilih Bangunan} 4. Administrator memilih bangunan untuk diubah datanya 5. Menjalankan subflow Ubah Data {Use case berakhir} 6. <i>Use case</i> berakhir
<i>Alternative Flows</i>	<i>A3. Data tidak dimasukkan atau kosong</i> Pada {Validasi Data} jika administrator tidak memasukkan data, 1. Administrator memasukkan data sesuai dengan format yang dikenali oleh sistem 2. Menjalankan subflow Ubah Data
<i>Subflows</i>	<i>S1. Ubah Data</i> 1. Sistem memberikan tampilan untuk mengubah data 2. Administrator mengubah data bangunan {Validasi Data} 1. Administrator menyimpan data 2. Melanjutkan ke basic flow {Menampilkan peta}
<i>Postconditions</i>	Sistem telah menyimpan data yang paling baru

Glossary

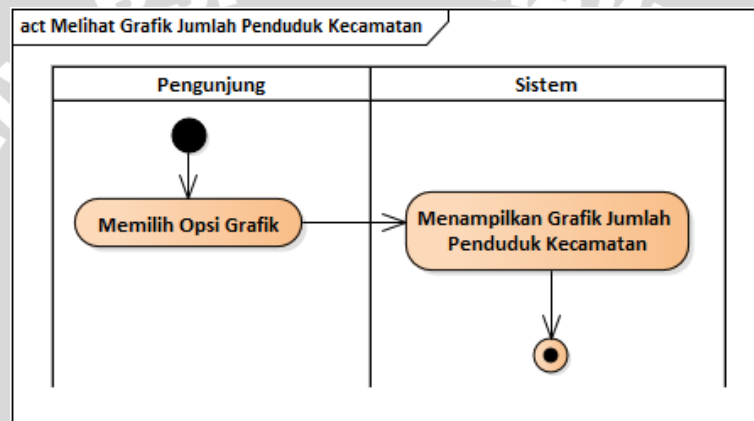
Password yang dimaksud adalah kata sandi untuk mengakses tampilan peta untuk mengubah data.

Data bangunan yang dimaksud terdiri dari nama bangunan, alamat, fungsi bangunan dan jumlah penghuni dalam bangunan.

4.8.4 Diagram Aktivitas

Setelah model *use case* dan spesifikasinya telah dibuat, kemudian dilakukan penggambaran aktivitas yang terjadi di setiap *use case* menggunakan diagram aktivitas. Penggambaran diagram aktivitas ini disesuaikan dengan alur yang telah dituliskan pada spesifikasi *use case* yang berisi alur dasar (*basic flow*) dan alur alternatif (*alternative flows*).

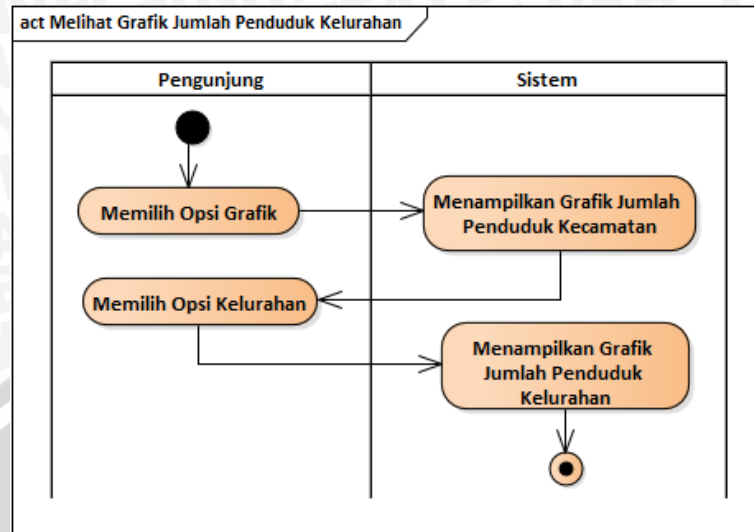
4.8.4.1 Melihat Grafik Jumlah Penduduk Kecamatan



Gambar 4.4 Diagram Aktivitas Melihat Grafik Jumlah Penduduk Kecamatan

Gambar 4.4 menjelaskan bahwa pengunjung yang ingin melihat grafik jumlah penduduk kecamatan, setelah mengakses sistem dapat memilih opsi Grafik dan kemudian akan ditampilkan grafik jumlah penduduk tiap kecamatan oleh sistem.

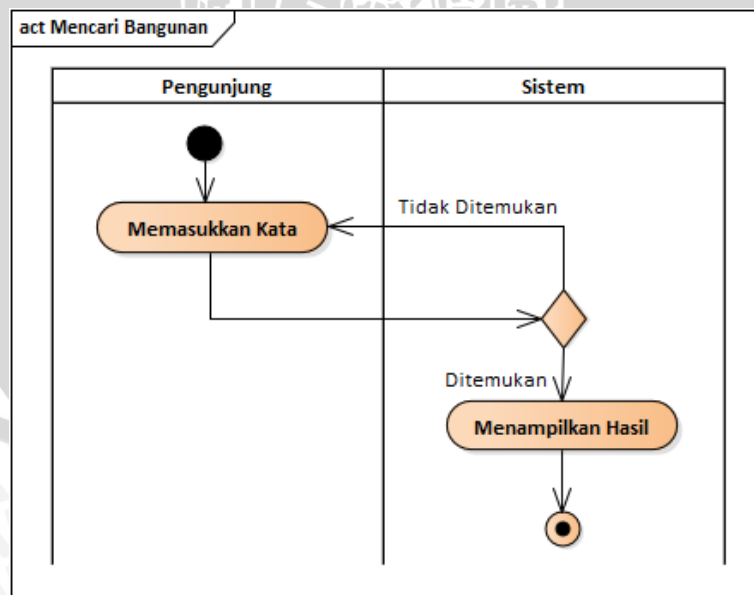
4.8.4.2 Melihat Grafik Jumlah Penduduk Kelurahan



Gambar 4.5 Diagram Aktivitas Melihat Grafik Jumlah Penduduk Kelurahan

Gambar 4.5 menjelaskan bahwa pengunjung yang ingin melihat grafik jumlah penduduk kelurahan, setelah mengakses sistem dapat memilih opsi Grafik, namun sistem akan menampilkan grafik jumlah penduduk kecamatan terlebih dulu sebagai tampilan awal, kemudian pengunjung memilih opsi kelurahan untuk melihat grafik jumlah penduduk tiap kelurahan.

4.8.4.3 Mencari Bangunan

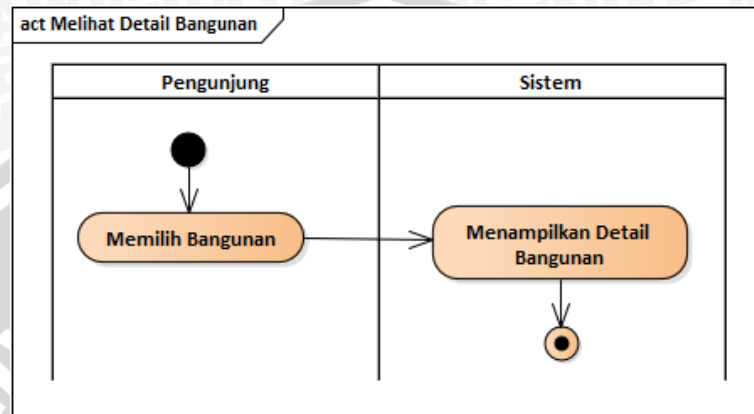


Gambar 4.6 Diagram Aktivitas Mencari Bangunan

Gambar 4.6 diatas merupakan diagram aktivitas untuk mencari bangunan yang dilakukan oleh pengunjung. Langkah pertama pengunjung memasukkan

kata yang berupa nama bangunan, fungsi bangunan atau nama jalan ke dalam tempat pencarian yang sudah disediakan sistem, kemudian sistem mencari kata yang dimasukkan tersebut. Jika kata ditemukan maka sistem menampilkan hasil sesuai dengan kata tersebut, namun jika tidak ditemukan maka pengunjung dapat memasukkan lagi kata yang lebih lengkap atau mudah dikenali oleh sistem.

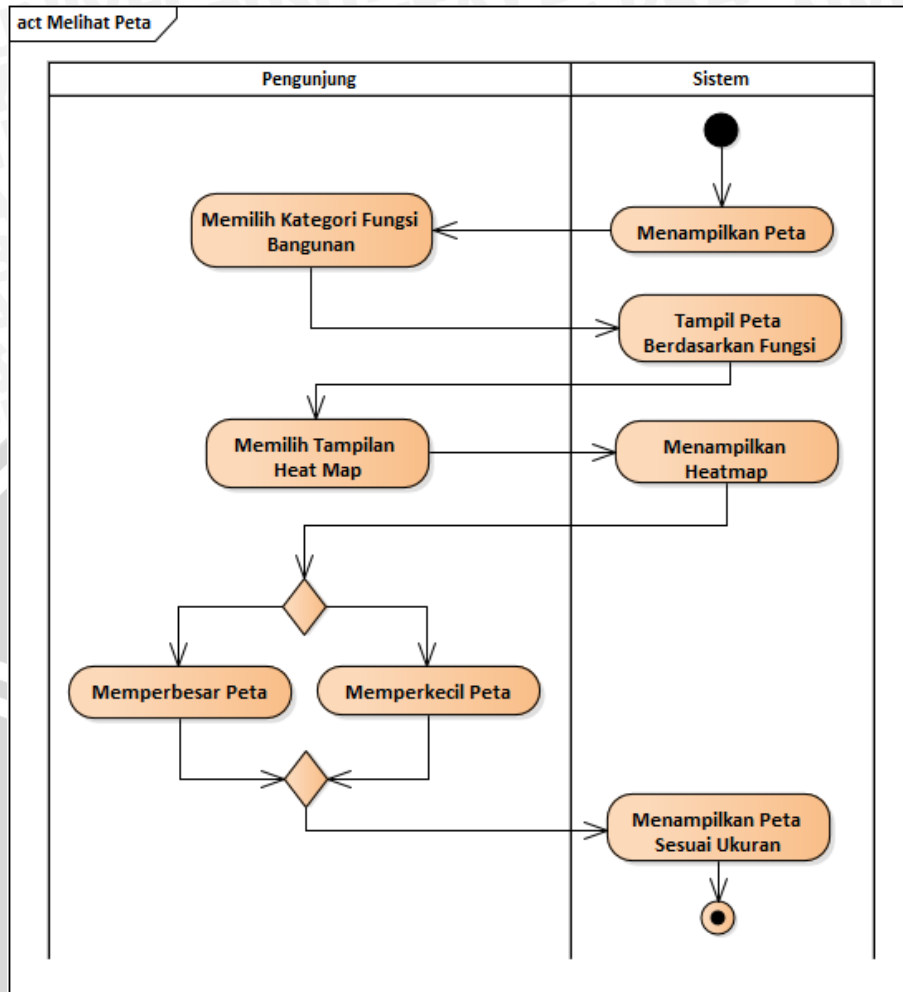
4.8.4.4 Melihat Detail Bangunan



Gambar 4.7 Diagram Aktivitas Melihat Detail Bangunan

Gambar 4.7 menjelaskan bahwa aktivitas melihat detail bangunan, pengunjung dapat memilih bangunan yang ingin dilihat detail informasi terkait bangunan seperti alamat, nama bangunan, fungsi bangunan dan jumlah orang dalam bangunan tersebut. Kemudian sistem akan menampilkan detail bangunan yang dipilih oleh pengunjung.

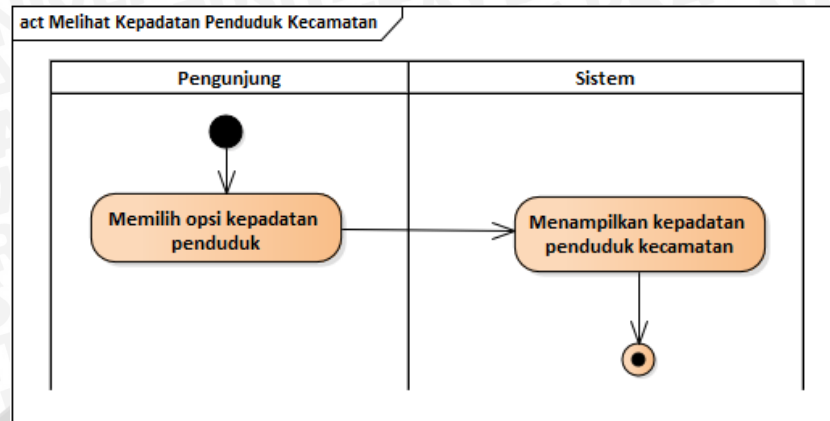
4.8.4.5 Melihat Peta



Gambar 4.8 Diagram Aktivitas Melihat Peta

Gambar 4.8 adalah diagram aktivitas untuk melihat peta, diawali dengan sistem menampilkan peta bangunan kemudian pengunjung memilih kategori fungsi bangunan dan sistem menampilkan peta bangunan berdasarkan fungsi. Langkah selanjutnya yaitu pengunjung memilih tampilan *heat map* dan sistem menampilkan *heat map* tiap bangunan sesuai jumlah orang dalam bangunan. Lalu jika pengunjung menginginkan tampilan yang lebih dekat atau bangunan tampak lebih besar, pengunjung dapat memperbesar tampilan peta, atau jika ingin melihat area peta yang lebih luas, pengunjung dapat memperkecil ukuran peta sehingga sistem dapat menampilkan peta sesuai dengan ukuran yang diinginkan oleh pengunjung.

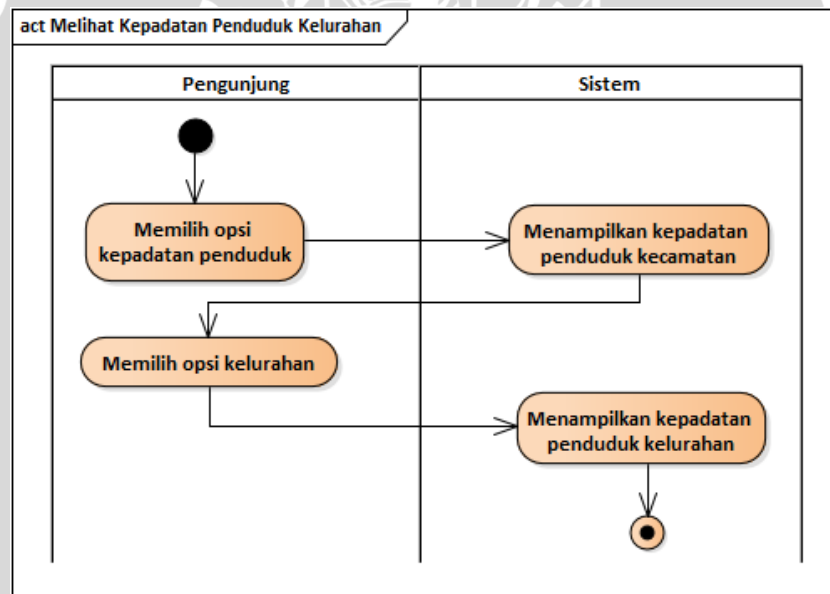
4.8.4.6 Melihat Kepadatan Penduduk Kecamatan



Gambar 4.9 Diagram Aktivitas Melihat Kepadatan Penduduk Kecamatan

Diagram aktivitas melihat kepadatan penduduk kecamatan (Gambar 4.9) diawali dengan pengunjung memilih opsi kepadatan penduduk kemudian sistem menampilkan kepadatan penduduk setiap kecamatan.

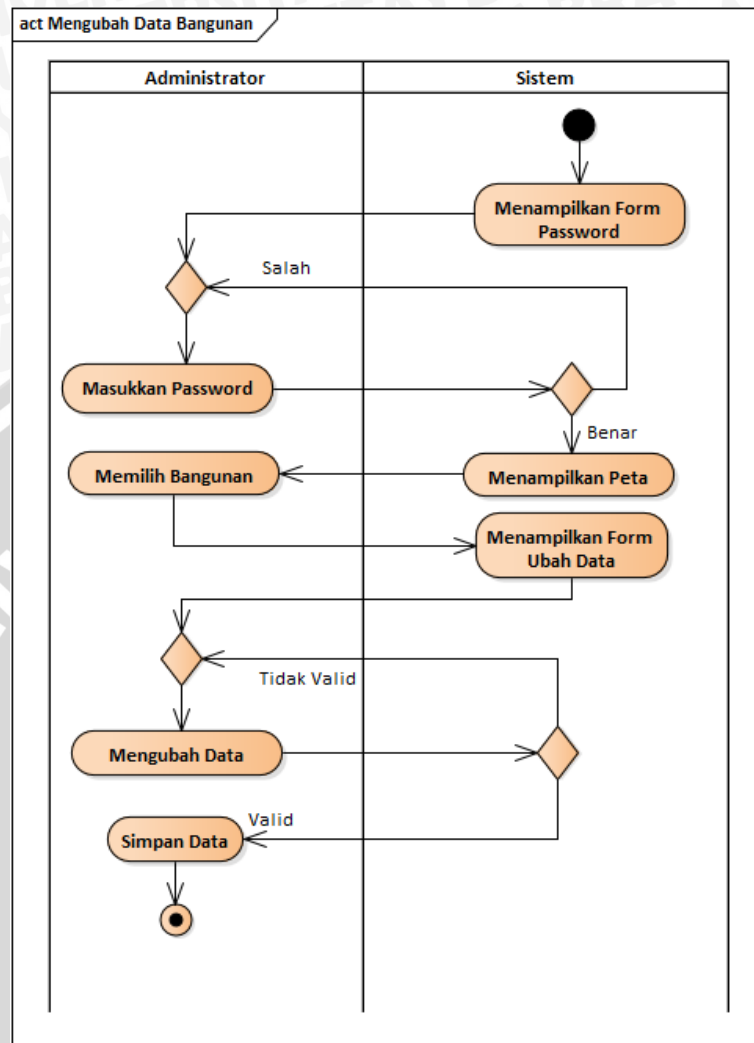
4.8.4.7 Melihat Kepadatan Penduduk Kelurahan



Gambar 4.10 Diagram Aktivitas Melihat Kepadatan Penduduk Kelurahan

Aktivitas untuk melihat kepadatan penduduk kelurahan pada gambar 4.10 diawali dengan pengunjung memilih opsi kepadatan penduduk, lalu sistem menampilkan kepadatan penduduk tiap kecamatan sebagai tampilan awal. Setelah itu pengunjung memilih opsi kelurahan dan sistem akan menampilkan kepadatan penduduk setiap kelurahan.

4.8.4.8 Mengubah Data Bangunan



Gambar 4.11 Diagram Aktivitas Mengubah Data Bagunan

Aktivitas mengubah data hanya dapat dilakukan oleh administrator ditunjukkan pada gambar 4.11 diatas. Diawali dengan sistem meminta untuk memasukkan password kemudian administrator memasukkan password. Setelah administrator memasukkan password sistem memeriksa password tersebut, jika password benar maka sistem menampilkan peta jika password salah maka administrator harus memasukkan kembali password yang lebih tepat. Kemudian jika password benar dan sistem menampilkan peta, administrator memilih bangunan untuk diubah datanya. Selanjutnya sistem menampilkan daftar konten apa saja yang dpat diubah. Saat administrator memasukkan data yang diubah jika data sesuai dengan format yang dikenali sistem maka administrator dapat menyimpan data baru tersebut, namun jika data yang dimasukkan tidak sesuai format maka administrator harus memasukkan lagi data yang lebih tepat dan sesuai dengan format yang dikenali sistem.

BAB 5 PERANCANGAN

5.1 Kelas-kelas Analisis

Pada awal perancangan penelitian ini dilakukan analisis pada kelas-kelas yang berhubungan pada masing-masing *use case* dengan memasukkannya pada subbab ini. Kelas-kelas analisis ini dilakukan untuk mendaftar seluruh kelas yang didapatkan pada masing-masing *use case* beserta hubungan setiap kelasnya yang akan digunakan untuk perancangan selanjutnya.

5.1.1 Kelas Analisis dari Analisis *Use case*

Daftar kelas-kelas analisis dari analisis *use case* yang telah dituliskan sebelumnya dijelaskan pada tabe 5.1 di bawah ini.

Tabel 5.1 Kelas Analisis dari Analisis *Use case*

<i>Use case</i>	Kelas Analisis
Melihat Grafik Jumlah Penduduk Kecamatan	<pre> classDiagram class AntarmukaPeta class GrafikController class Penduduk class Bangunan AntarmukaPeta --> GrafikController GrafikController --> Penduduk GrafikController --> Bangunan </pre>
Melihat Grafik Jumlah Penduduk Kelurahan	<pre> classDiagram class AntarmukaPeta class GrafikController class Penduduk class Bangunan AntarmukaPeta --> GrafikController GrafikController --> Penduduk GrafikController --> Bangunan </pre>
Mencari Bangunan	<pre> classDiagram class AntarmukaPeta class PetaController class Bangunan AntarmukaPeta --> PetaController PetaController --> Bangunan </pre>

Tabel 5.1 Kelas Analisis dari Analisis *Use case* (lanjutan)

<i>Use case</i>	Kelas Analisis
Melihat Detail Bangunan	<pre> graph TD A((Antarmuka Peta)) --> B((Peta Controller)) B --- C((Bangunan)) </pre>
Melihat Peta	<pre> graph TD A((Antarmuka Peta)) --> B((Peta Controller)) B --- C((Bangunan)) </pre>
Melihat Kepadatan Penduduk Kecamatan	<pre> graph TD A((Antarmuka Peta)) --> B((Kepadatan Controller)) C((Antarmuka Kep Pend)) --> B B --- D((Bangunan)) </pre>
Melihat Kepadatan Penduduk Kelurahan	<pre> graph TD A((Antarmuka Peta)) --> B((Kepadatan Controller)) C((Antarmuka Kep Pend)) --> B B --- D((Bangunan)) </pre>
Mengubah Data Bangunan	<pre> graph TD A((Antarmuka Ubah Data)) --> B((Ubah Data Controller)) C((Form Keamanan)) --> B B --- D((Bangunan)) </pre>

5.1.2 Kelas Analisis dari Mekanisme Analisis

Mekanisme analisis merupakan pola yang digunakan sebagai solusi umum untuk masalah yang umum dengan menunjukkan pola struktur, pola perilaku atau keduanya. Pola-pola ini digunakan selama analisis untuk mengurangi kompleksitas analisis dan meningkatkan konsistensi dengan menyediakan representasi yang lebih singkat untuk perilaku yang kompleks (Siemers & Kutsick, 2004). Berikut tabel untuk mekanisme analisis (Tabel 5.2).

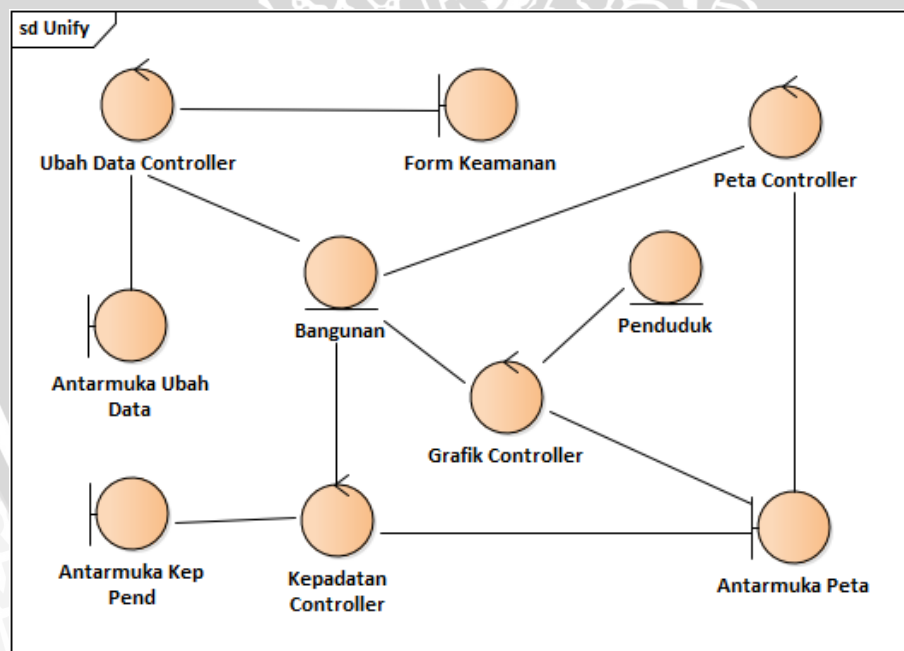
Tabel 5.2 Kelas Analisis dari Mekanisme Analisis

Analisis Kelas	Mekanisme Analisis
Bangunan	Persistensi
Penduduk	Persistensi

Menurut tabel 5.2, mekanisme analisis yang digunakan adalah persistensi. Persistensi ini merupakan daya tahan suatu kelas atau objek untuk terus-menerus digunakan dan tetap berfungsi dengan baik jika sistem dimatikan kemudian dihidupkan kembali.

5.1.3 Unifikasi Analisis Kelas

Pada gambar 5.1 merupakan gabungan atau unifikasi dari keseluruhan analisis kelas yang telah digambarkan sebelumnya.



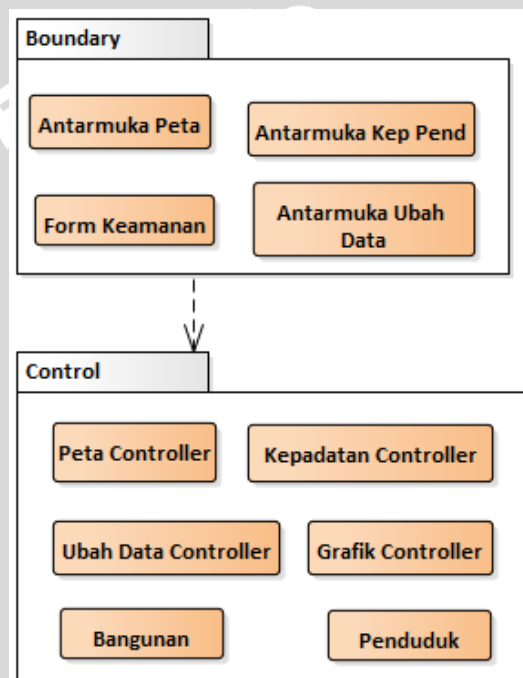
Gambar 5.1 Unifikasi Analisis Kelas

5.2 Elemen Desain

Pada elemen desain ini terdapat *package* dan diagram kelas. *Package* digunakan untuk mengelompokkan kelas-kelas berdasarkan fungsinya di dalam sistem. Sementara untuk diagram kelas berfungsi untuk menunjukkan kelas dalam sistem yang berisi atribut serta operasinya dan hubungan antara kelas-kelas tersebut.

5.2.1 *Package*

Pada elemen desain yang pertama adalah membuat *package* yang terdiri dari dua *package* pada Gambar 5.2 di bawah ini. *Package* pada rancangan awal SIGEKA terdiri dari *Boundary* dan *Control*.

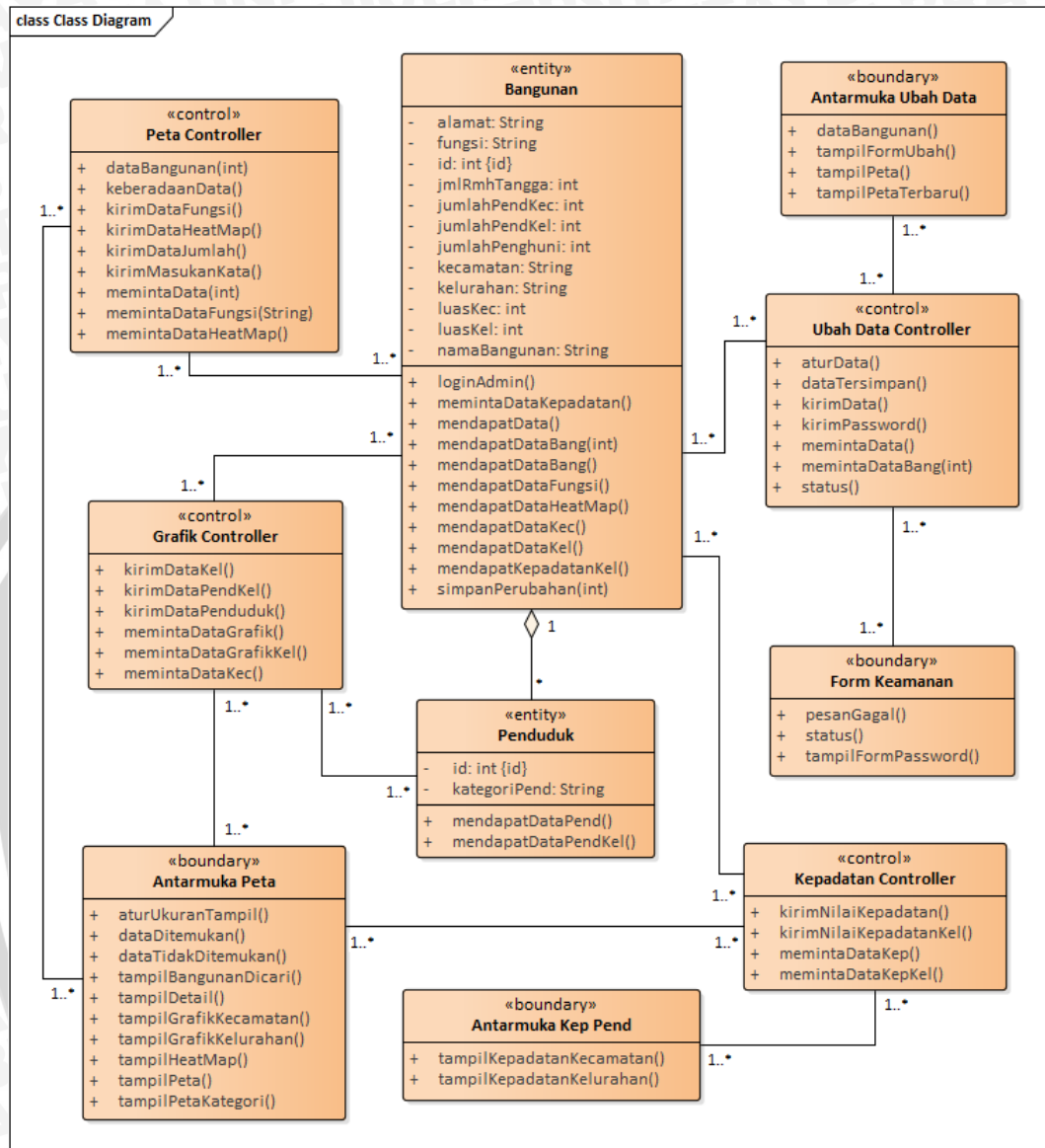


Gambar 5.2 *Package*

Package pada SIGEKA (Gambar 5.2) terdiri dari dua *package* yaitu *package Boundary* dan *Control*. *Package boundary* dipisahkan dari *control* dan *entity* karena jika terdapat perubahan pada tampilan sistem, maka perubahan dapat langsung dilakukan pada *boundary* sistem yang terdiri dari *boundary* Antarmuka Peta, Antarmuka Kep Pend, Form Keamanan dan Antarmuka Ubah Data. kemudian untuk *package control* terdiri dari kelas *control* dan *entity* yaitu Peta Controller, Kepadatan Controller, Ubah Data Controller, Grafik Controller, Bangunan dan Penduduk.

5.2.2 Diagram Kelas

Diagram kelas untuk SIGEKA dapat dilihat pada Gambar 5.3 berikut ini.



Gambar 5.3 Diagram Kelas

Penjelasan untuk setiap kelas pada Diagram Kelas diatas (Gambar 5.3) dituliskan dalam tabel-table berikut ini.

5.2.2.1 Penjelasan Kelas Peta Controller

Tabel 5.3 Penjelasan Kelas Peta Controller

Nama	Peta Controller
Deskripsi Kelas	Kelas yang menghubungkan antara <i>boundary</i> dengan <i>entity</i> serta mengatur dalam pengiriman data yang diperoleh dari <i>entity</i> ke <i>boundary</i> berkaitan dengan peta dan informasi lengkap seperti legenda, skala dan penunjuk arah.
Tipe Kelas	<i>Control</i>

5.2.2.2 Penjelasan Kelas Grafik Controller

Tabel 5.4 Penjelasan Kelas Grafik Controller

Nama	Grafik Controller
Deskripsi Kelas	Kelas yang menghubungkan antara <i>boundary</i> dengan <i>entity</i> serta mengatur dalam pengiriman data yang diperoleh dari <i>entity</i> ke <i>boundary</i> berkaitan dengan grafik jumlah penduduk tiap kecamatan dan tiap kelurahan
Tipe Kelas	<i>Control</i>

5.2.2.3 Penjelasan Kelas Kepadatan Controller

Tabel 5.5 Penjelasan Kelas Kepadatan Controller

Nama	Kepadatan Controller
Deskripsi Kelas	Kelas yang menghubungkan antara <i>boundary</i> dengan <i>entity</i> serta mengatur dalam pengiriman data yang diperoleh dari <i>entity</i> ke <i>boundary</i> berkaitan dengan kepadatan jumlah penduduk tiap kecamatan dan tiap kelurahan
Tipe Kelas	<i>Control</i>

5.2.2.4 Penjelasan Kelas Ubah Data Controller

Tabel 5.6 Penjelasan Kelas Ubah Data Controller

Nama	Ubah Data Controller
Deskripsi Kelas	Kelas yang menghubungkan antara <i>boundary</i> dengan <i>entity</i> serta mengatur dalam pengiriman data yang diperoleh dari <i>entity</i> ke <i>boundary</i> berkaitan dengan proses mengubah data setiap bangunan
Tipe Kelas	<i>Control</i>

5.2.2.5 Penjelasan Kelas Bangunan

Tabel 5.7 Penjelasan Kelas Bangunan

Nama	Bangunan
Deskripsi Kelas	Kelas yang mengakses langsung ke dalam database berkaitan dengan detail informasi bangunan dan lokasi titik koordinatnya dalam peta.
Tipe Kelas	<i>Entity</i>

5.2.2.6 Penjelasan Kelas Penduduk

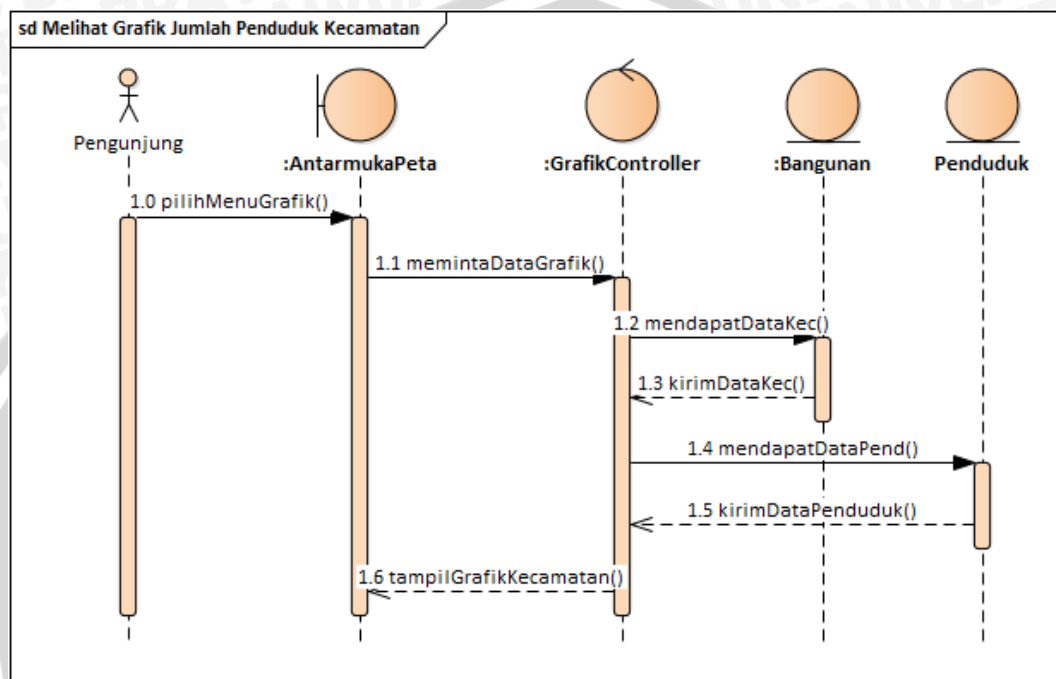
Tabel 5.8 Penjelasan Kelas Penduduk

Nama	Penduduk
Deskripsi Kelas	Kelas yang mengakses langsung ke dalam database berkaitan dengan detail informasi penduduk yang dikelompokkan berdasarkan kategori usia dan jenis kelamin.
Tipe Kelas	<i>Entity</i>

5.3 Pemodelan Interaksi

Kemudian dilakukan pemodelan interaksi yang berfungsi untuk memodelkan interaksi antara aktor dengan objek dalam suatu sistem dan interaksi antar objek itu sendiri. Pemodelan interaksi ini digambarkan pada masing-masing *use case* yang telah dimodelkan sebelumnya.

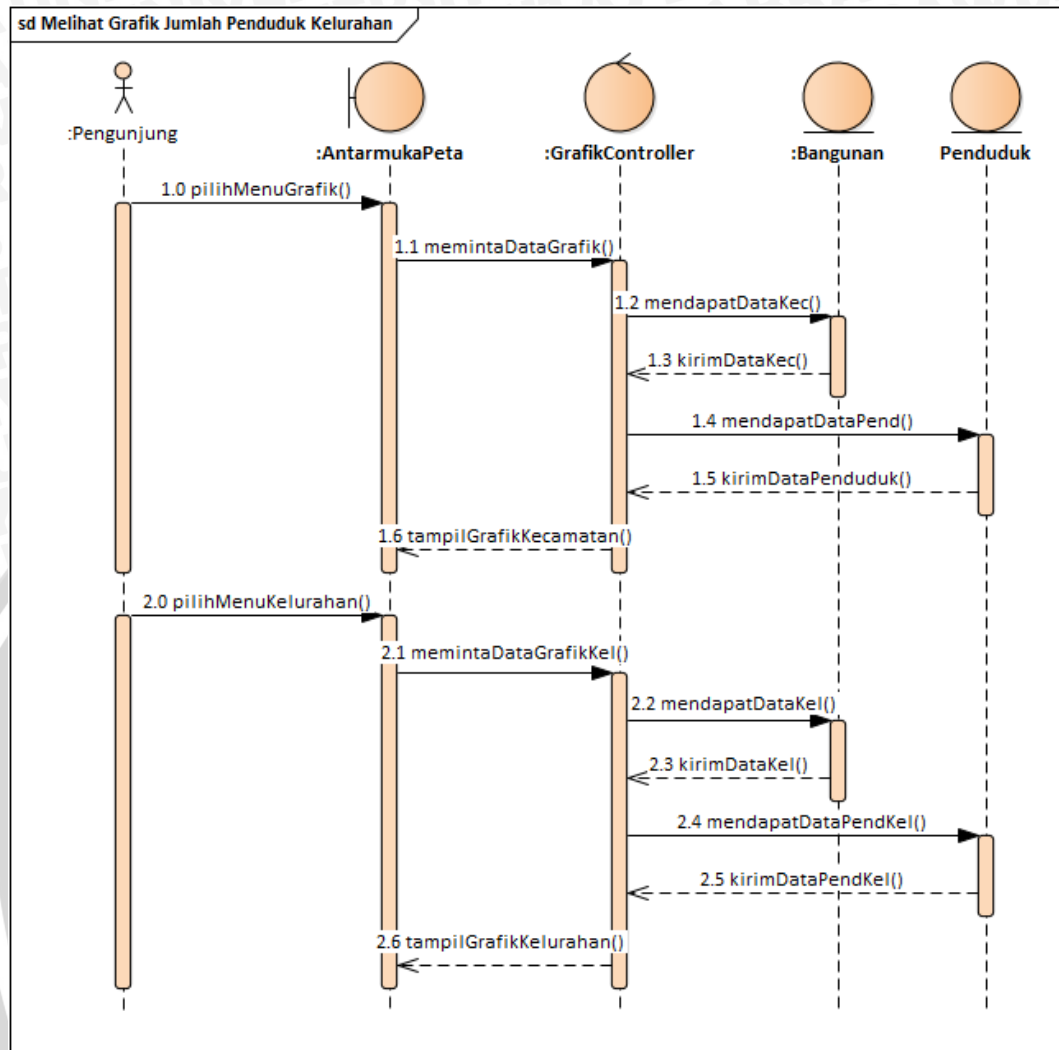
5.3.1 Melihat Grafik Jumlah Penduduk Kecamatan



Gambar 5.4 Diagram Interaksi Melihat Grafik Jumlah Penduduk Kecamatan

Gambar 5.4 menggambarkan bahwa ketika pengunjung ingin melihat grafik jumlah penduduk tiap kecamatan, pengunjung pertama kali harus memilih opsi Grafik setelah mengakses sistem. Interaksi yang terjadi yaitu *boundary* AntarmukaPeta akan meminta data grafik kepada *controller* GrafikController ditunjukkan pada method *memintaDataGrafik()* yang kemudian oleh *controller* akan mengakses dua *entity* yaitu Bangunan dan Penduduk untuk mendapatkan data terkait kecamatan dan jumlah penduduknya. Setelah itu *controller* akan mengirimkan data yang telah dikelompokkan tersebut kepada *boundary* AntarmukaPeta untuk ditampilkan, ditunjukkan pada method *tampilGrafikKecamatan()*.

5.3.2 Melihat Grafik Jumlah Penduduk Kelurahan

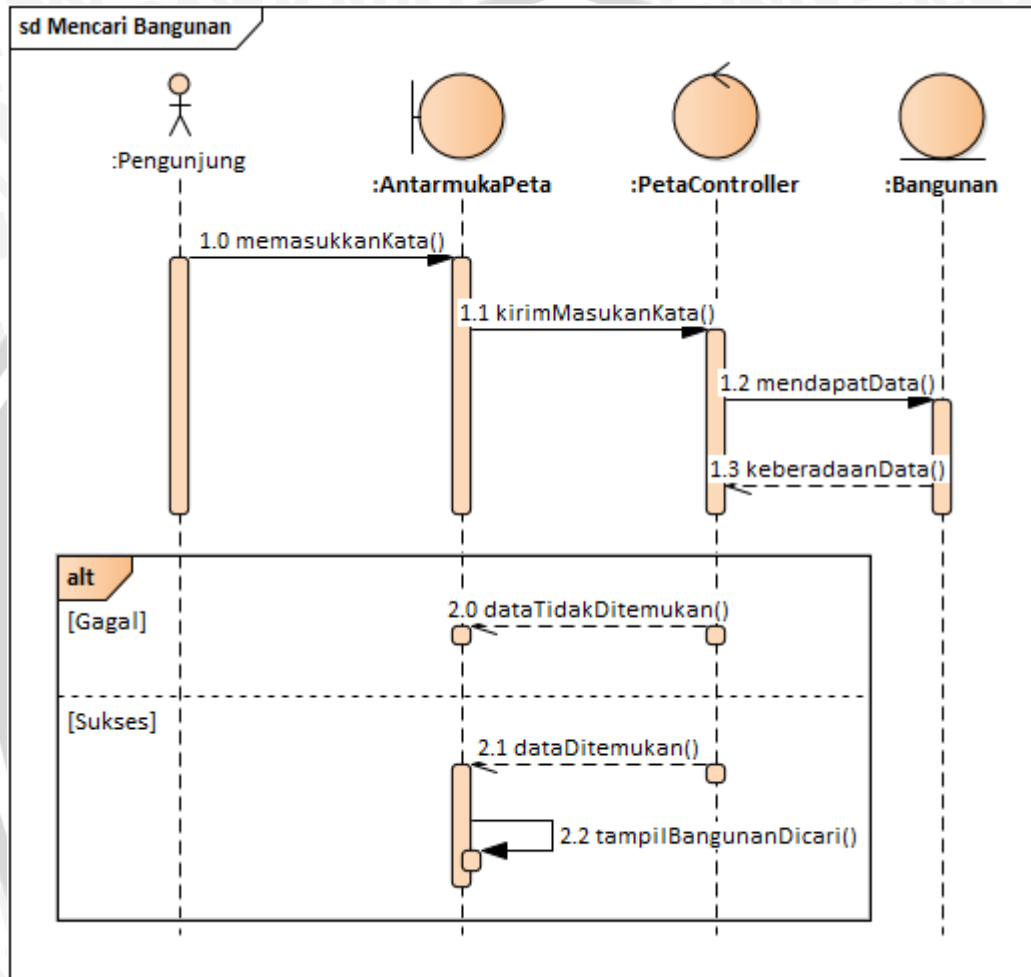


Gambar 5.5 Diagram Interaksi Melihat Grafik Jumlah Penduduk Kelurahan

Gambar 5.5 menggambarkan bahwa ketika pengunjung ingin melihat grafik jumlah penduduk tiap kelurahan, pengunjung pertama kali harus memilih opsi Grafik setelah mengakses sistem. Namun tampilan awal grafik yang ditampilkan yaitu grafik untuk tiap kecamatan. Interaksi yang terjadi pada tampilan grafik kecamatan yaitu *boundary* AntarmukaPeta akan meminta data grafik kepada *controller* GrafikController ditunjukkan pada *method* memintaDataGrafik() yang kemudian oleh *controller* akan mengakses dua *entity* yaitu Bangunan dan Penduduk untuk mendapatkan data terkait kecamatan dan jumlah penduduknya. Setelah itu *controller* akan mengirimkan data yang telah dikelompokkan tersebut kepada *boundary* AntarmukaPeta untuk ditampilkan berupa grafik tiap kecamatan, ditunjukkan pada *method* tampilGrafikKecamatan(). Lalu untuk dapat melihat grafik tiap kelurahan, pengunjung terlebih dulu memilih opsi Kelurahan. Interaksi yang terjadi yaitu *boundary* AntarmukaPeta akan meminta data grafik kelurahan kepada *controller* GrafikController (*method* memintaDataGrafikKel()) yang kemudian oleh

controller akan mengakses dua *entity* yaitu Bangunan dan Penduduk untuk mendapatkan data terkait kelurahan dan jumlah penduduknya. Setelah itu *controller* akan mengirimkan data yang telah dikelompokkan tersebut kepada *boundary* AntarmukaPeta untuk ditampilkan berupa grafik tiap kelurahan, ditunjukkan pada *method* tampilGrafikKelurahan().

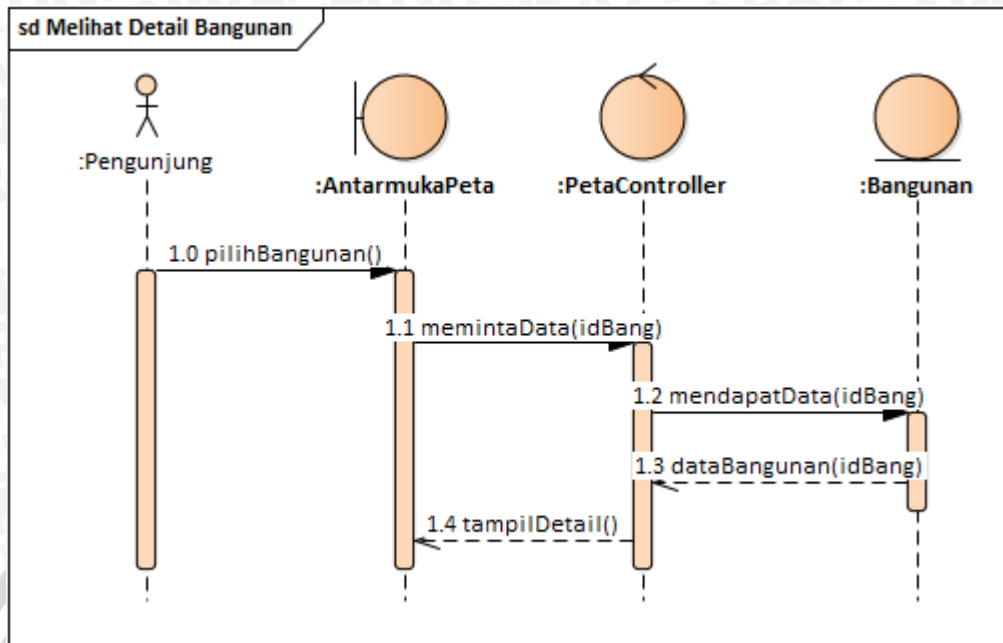
5.3.3 Mencari Bangunan



Gambar 5.6 Diagram Interaksi Mencari Bangunan

Pada saat pengunjung ingin mencari bangunan (Gambar 5.6), pengunjung harus memasukkan kata terkait nama bangunan, fungsi bangunan atau nama jalan ke dalam tempat yang sudah disediakan sistem. Kemudian interaksi yang terjadi yaitu masukan kata yang diterima di AntarmukaPeta akan dikirimkan kepada PetaController, ditunjukkan pada *method* kirimMasukanKata(). Lalu PetaController mengakses data pada *entity* Bangunan untuk mendapatkan data sesuai masukan kata tersebut. Pada PetaController akan dilakukan pemeriksaan keberadaan kata jika bangunan yang dicari tidak ditemukan (*method* dataTidakDitemukan()) maka sistem tidak menampilkan hasil, jika ditemukan (*method* dataDitemukan()) maka sistem akan menampilkan bangunan yang dicari ditunjukkan pada *method* tampilBangunanDicari().

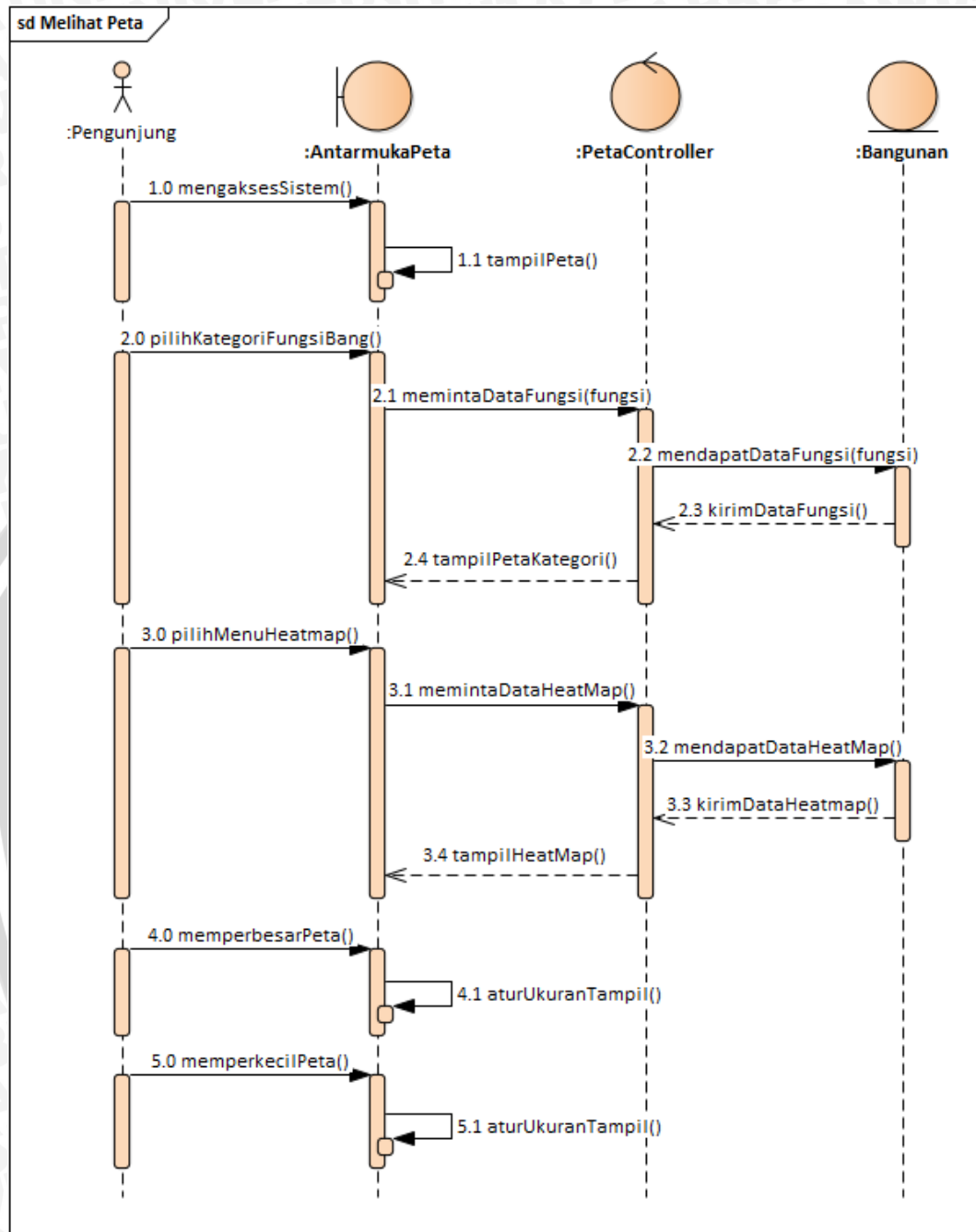
5.3.4 Melihat Detail Bangunan



Gambar 5.7 Diagram Interaksi Melihat Detail Bangunan

Gambar 5.7 menggambarkan bahwa interaksi melihat detail bangunan yaitu pengunjung memilih bangunan, kemudian *boundary* AntarmukaPeta meminta data sesuai id bangunan yang dipilih kepada PetaController, ditunjukkan pada *method* memintaData(idBang). Kemudian oleh PetaController mengakses data bangunan pada *entity* Bangunan, lalu PetaController mengirimkan detail bangunan untuk ditampilkan, ditunjukkan pada *method* tampilDetail().

5.3.5 Melihat Peta



Gambar 5.8 Diagram Interaksi Melihat Peta

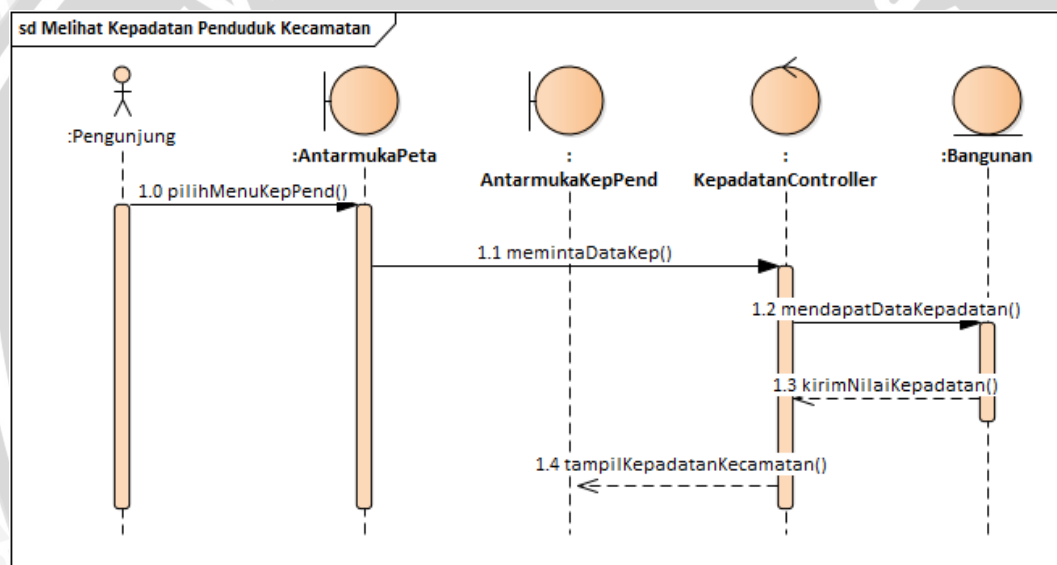
Gambar 5.8 menggambarkan bahwa saat pengunjung melihat peta, interaksi yang terjadi di dalam sistem yaitu *boundary* AntarmukaPeta menampilkan peta kemudian pengunjung memilih kateogori bangunan berdasarkan fungsi bangunan, maka AntarmukaPeta meminta data bangunan berdasarkan fungsi yang dipilih kepada *control* PetaController, ditunjukkan pada *method* memintaDataFungsi(fungsi). Kemudian PetaController mengakses *entity* bangunan untuk mendapatkan data terkait fungsi bangunan beserta koordinatnya. Kemudian setelah data diolah, PetaController akan

mengirimkannya kepada AntarmukaPeta untuk ditampilkan (*method* tampilPetaKategori()).

Setelah itu pengunjung memilih opsi *heat map*, AntarmukaPeta meminta data *heat map* kepada PetaController, ditunjukkan pada *method* memintaDataHeatMap(). Lalu PetaController mendapatkan data terkait dengan menampilkan *heat map* yang terdiri dari data jumlah penghuni tiap bangunan dan data lokasi bangunan dari *entity* Bangunan dan PetaController akan mengirimkan data tersebut kepada AntarmukaPeta untuk ditampilkan kepada pengunjung, ditunjukkan pada *method* tampilHeatMap().

Jika pengunjung ingin mengatur ukuran peta terlihat lebih dekat atau lebih jauh maka pengunjung dapat mengatur ukuran peta (*zoom in / zoom out*). AntarmukaPeta akan mengatur tampilan sesuai ukuran yang diinginkan oleh pengunjung (*method* aturUkuranTampil()).

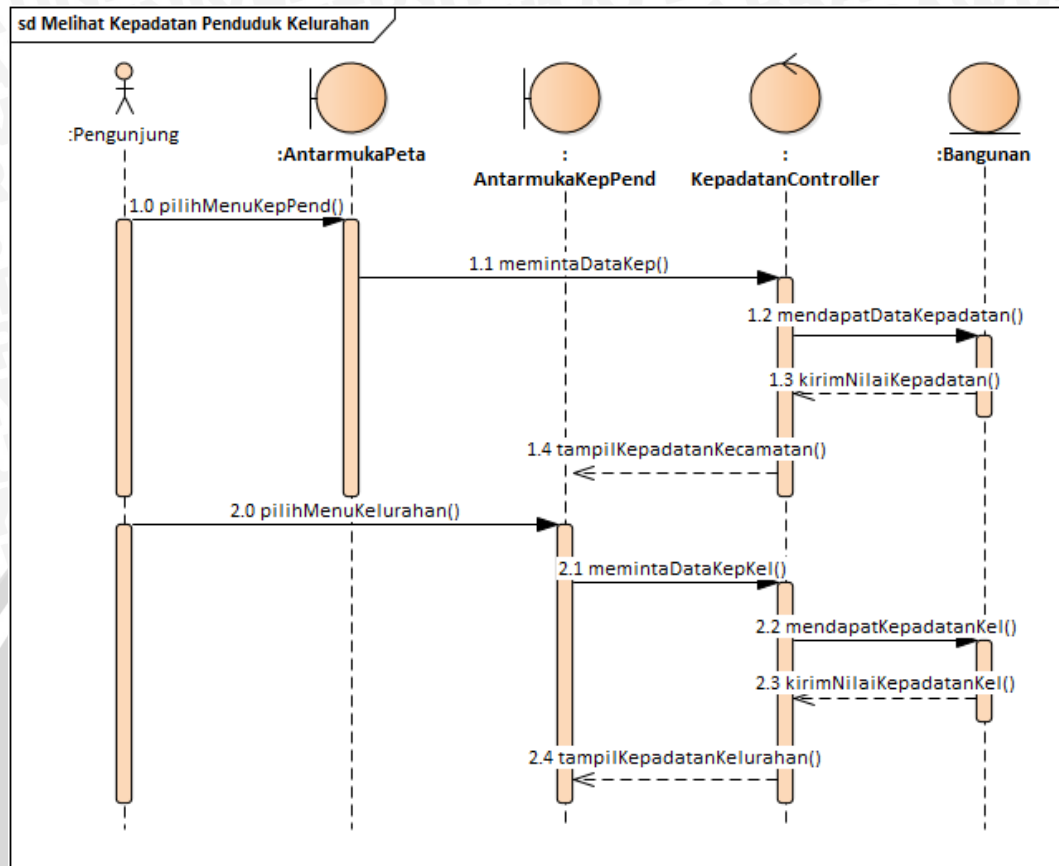
5.3.6 Melihat Kepadatan Penduduk Kecamatan



Gambar 5.9 Diagram Interaksi Melihat Kepadatan Penduduk Kecamatan

Gambar 5.9 menggambarkan bahwa ketika pengunjung ingin melihat kepadatan penduduk tiap kecamatan, pengunjung pertama kali harus memilih opsi Kepadatan setelah mengakses sistem. Interaksi yang terjadi yaitu *boundary* AntarmukaPeta akan meminta data kepadatan kepada *controller* KepadatanController menggunakan *method* memintaDataKep() yang kemudian oleh *controller* akan mengakses *entity* Bangunan untuk mendapatkan data terkait kecamatan dan kepadatan penduduknya. Setelah itu *controller* akan mengirimkan data yang telah diolah tersebut kepada *boundary* AntarmukaKepPend untuk ditampilkan kepadatan penduduk setiap kecamatan, menggunakan *method* tampilKepadatanKecamatan().

5.3.7 Melihat Kepadatan Penduduk Kelurahan

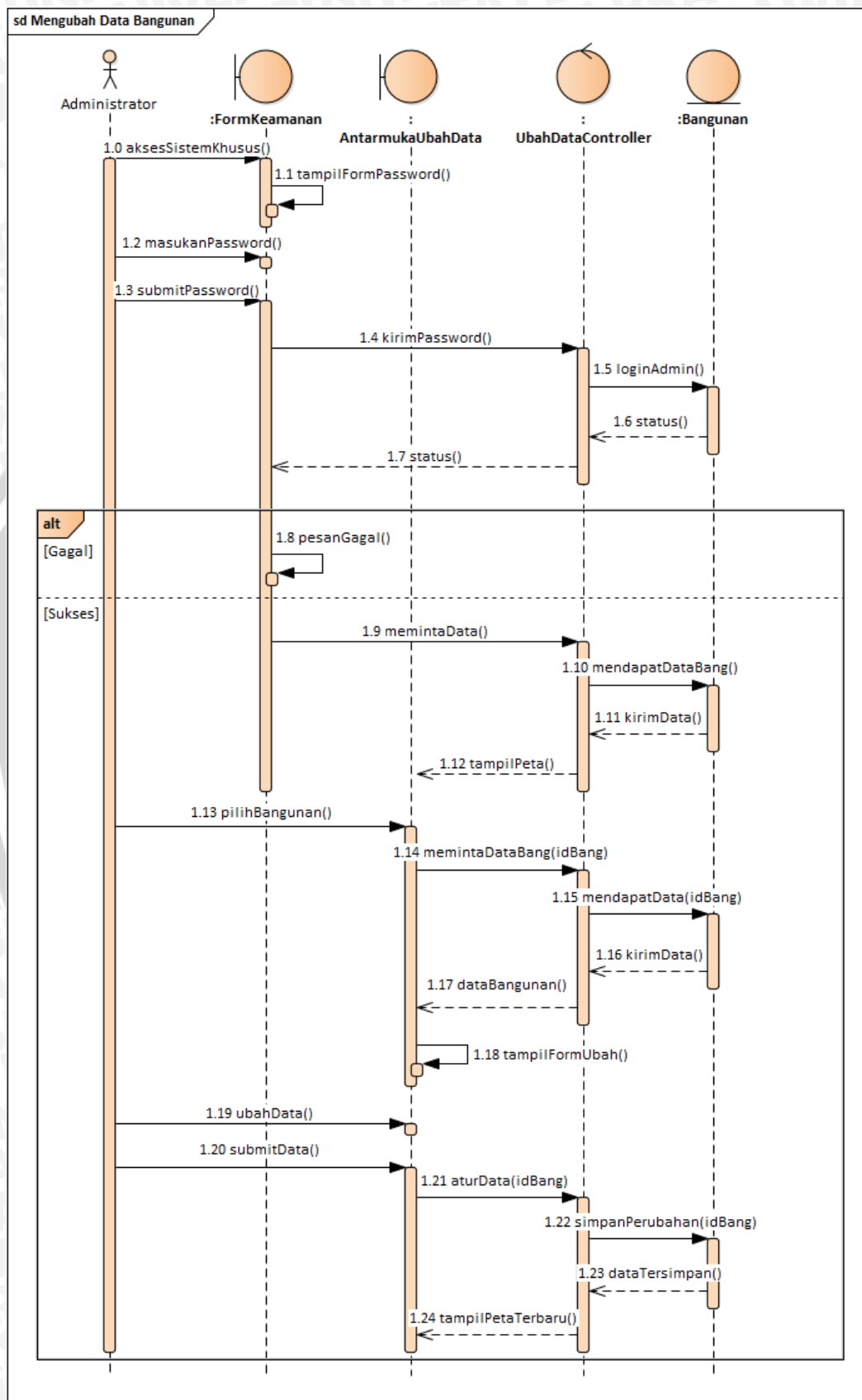


Gambar 5.10 Diagram Interaksi Melihat Kepadatan Penduduk Kelurahan

Ketika pengunjung ingin melihat kepadatan penduduk tiap kelurahan (Gambar 5.10), pengunjung pertama kali harus memilih opsi Kepadatan setelah mengakses sistem. Namun tampilan awal kepadatan penduduk yang ditampilkan yaitu kepadatan untuk tiap kecamatan. Interaksi yang terjadi yaitu *boundary* AntarmukaPeta akan meminta data kepadatan kepada *controller* KepadatanController menggunakan *method* memintaDataKep() yang kemudian oleh *controller* akan mengakses *entity* Bangunan untuk mendapatkan data terkait kecamatan dan kepadatan penduduknya. Setelah itu *controller* akan mengirimkan data yang telah diolah tersebut kepada *boundary* AntarmukaKepPend untuk ditampilkan kepadatan penduduk setiap kecamatan, menggunakan *method* tampilKepadatanKecamatan().

Lalu untuk dapat melihat kepadatan tiap kelurahan, pengunjung terlebih dulu memilih opsi Kelurahan. Interaksi yang terjadi yaitu *boundary* AntarmukaPeta akan meminta data kepadatan kelurahan kepada *controller* KepadatanController menggunakan *method* memintaDataKepKel() yang kemudian oleh *controller* akan mengakses *entity* Bangunan untuk mendapatkan data terkait kelurahan dan kepadatan penduduknya. Setelah itu *controller* akan mengirimkan data yang telah diolah tersebut kepada *boundary* AntarmukaKepPend untuk ditampilkan kepadatan penduduk setiap kelurahan, menggunakan *method* tampilKepadatanKelurahan().

5.3.8 Mengubah Data Bangunan



Gambar 5.11 Diagram Interaksi Mengubah Data Bangunan

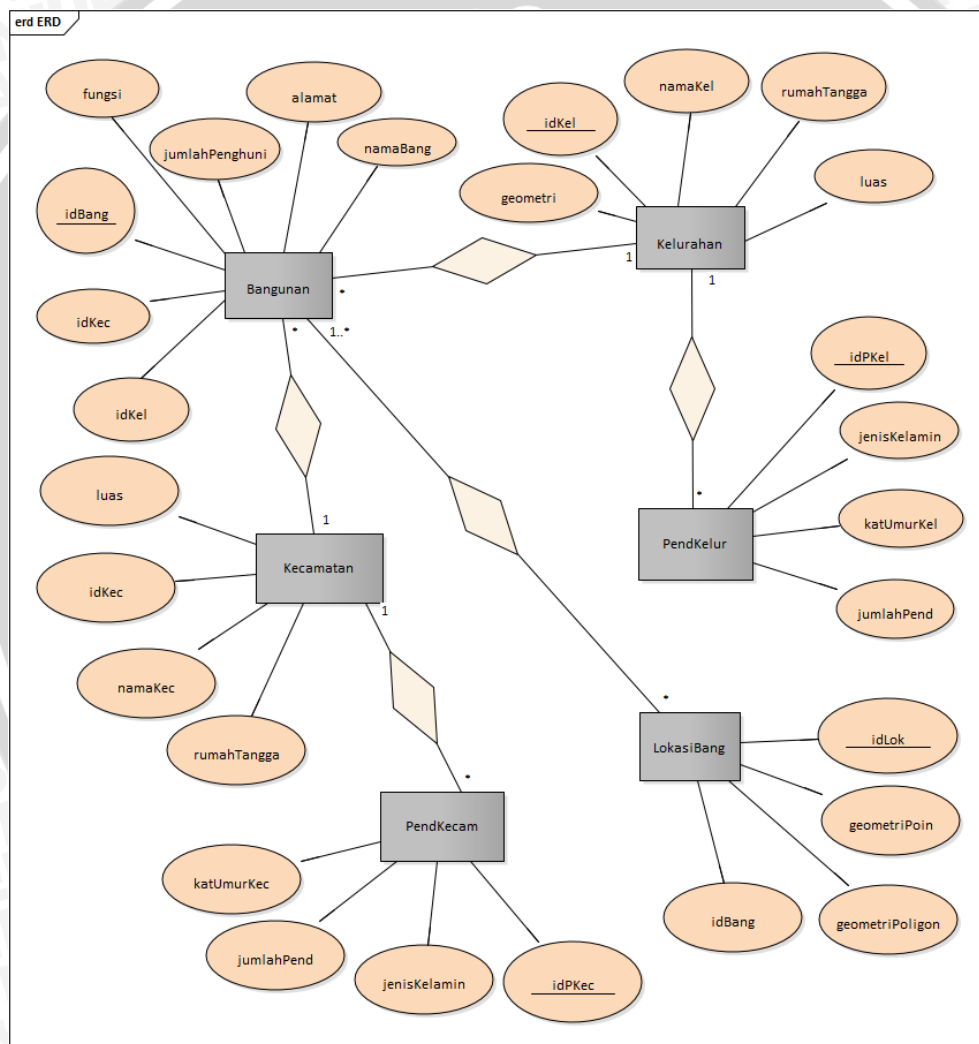
Gambar 5.11 menggambarkan bahwa interaksi mengubah data oleh administrator diawali dengan mengakses sistem khusus, kemudian sistem akan meminta masukan *password* pada *boundary* FormKeamanan. Setelah itu administrator memasukkan *password* dan *submit password*. FormKeamanan akan mengirimkan masukan *password* dari administrator kepada UbahDataController menggunakan *method* kirimPassword(), lalu UbahDataController mengirimkan *password* kepada *entity* Bangunan untuk dilakukan pemeriksaan kesesuaian *password* dan kemudian akan mengirimkan status kepada UbahDataController dan ditampilkan kepada AntarmukaUbahData. Jika *password* tidak sesuai maka akan menampilkan pesan gagal, namun jika *password* sesuai maka akan ditampilkan peta pada AntarmukaUbahData (proses dari FormKeamanan meminta data kepada UbahDataController lalu menampilkan data seluruh bangunan dari *entity* Bangunan kepada AntarmukaUbahData). Saat administrator memilih bangunan yang akan diubah datanya, AntarmukaUbahData akan mengirimkan permintaan data bangunan sesuai yang dipilih oleh administrator (*method* memintaDataBang(idBang)), lalu UbahDataController mendapatkan data tersebut dari *entity* Bangunan dan akan dikirimkan kembali kepada AntarmukaUbahData untuk ditampilkan form ubah datanya (*method* tampilFormUbah()). Setelah itu administrator melakukan ubah data dan submit data. AntarmukaUbahData akan mengirimkan data yang telah diubah oleh administrator menggunakan *method* aturData(idBang) dan UbahDataController akan menyimpannya kepada *entity* Bangunan dan mengirimkan pesan data tersimpan kepada AntarmukaUbahData menggunakan *method* tampilPetaTerbaru().

5.4 Pemodelan Data

Pemodelan data dilakukan bertujuan untuk memodelkan data yang berisi atribut tabel penyimpanan dan hubungan tiap tabel tersebut yang akan digunakan untuk menampilkan konten pada SIGEKA, supaya sistem dapat menampilkan informasi seperti informasi *heat map* yang akan menggambarkan kondisi kepadatan penduduk dalam skala detail yaitu pada setiap bangunan yang ditampilkan.

5.4.1 Entity Relationship Diagram

Penjelasan hubungan antar data dijelaskan berupa *entity relationship diagram* dapat dilihat pada Gambar 5.12 berikut ini.



Gambar 5.12 Entity Relationship Diagram

Tabel pada SIGEKA terdiri dari 6 tabel yaitu tabel Bangunan, Kelurahan, Kecamatan, PendKelur, PendKecam dan LokasiBang. Tabel Bangunan memiliki 7 atribut (kolom) yaitu idBang, namaBang, alamat, jumlahPeghuni, fungsi, idKec dan idKel. Tabel Kelurahan memiliki 5 atribut yaitu idKel, namaKel, rumahTangga, luas dan geometri. Kemudian pada Tabel Kecamatan juga memiliki 5 atribut yang

terdiri dari idKec, namaKec, rumahTangga, luas dan geometri. Tabel PendKelur memiliki 4 kolom yaitu idPKel, jenisKelamin, katUmurKel dan jumlahPend. Tabel PendKecam memiliki 4 atribut yaitu idPKec, jenisKelamin, katUmurKec dan jumlahPend. Tabel terakhir adalah LokasiBang yang memiliki 4 atribut yakni idLok, geometriPoin, geometriPoligon dan idBang.

Relasi tabel Kelurahan terhadap Bangunan memiliki relasi satu ke banyak yaitu satu kelurahan terdiri dari banyak bangunan. Kemudian relasi pada Kelurahan terhadap PendKelur memiliki relasi satu ke banyak, relasi tabel Kecamatan dengan Bangunan adalah satu ke banyak juga. Tabel Kecamatan terhadap PendKecam memiliki relasi satu ke banyak dan yang terakhir Bangunan terhadap LokasiBang memiliki relasi banyak ke banyak.

5.4.2 Perancangan Tabel

5.4.2.1 Tabel Bangunan

Nama tabel : Bangunan

Jumlah field : 7

Fungsi : untuk menyimpan detail dari setiap bangunan

Tabel 5.9 Perancangan Tabel Bangunan

No.	Nama field	Tipe	Lebar	Keterangan
1	idBang	Integer	-	Id bangunan
2	alamat	Varchar	100	Alamat bangunan
3	fungsi	Varchar	50	Fungsi bangunan
4	jmlPenghuni	Integer	-	Jumlah penghuni tiap bangunan
5	namaBang	Varchar	50	Nama bangunan
6	idKel	Varchar	6	Id kelurahan
7	idKec	Varchar	5	Id kecamatan

5.4.2.2 Tabel Kecamatan

Nama tabel : Kecamatan

Jumlah field : 5

Fungsi : untuk menyimpan data kecamatan

Tabel 5.10 Perancangan Tabel Kecamatan

No.	Nama field	Tipe	Lebar	Keterangan
1	idKec	Varchar	5	Id kecamatan
2	namaKec	Varchar	50	Nama kecamatan
3	luas	Integer	-	Luas area kecamatan
4	rumahTangga	Integer	-	Jumlah rumah tangga pada masing-masing kecamatan
5	geometri	Text	-	Titik-titik koordinat lokasi kecamatan

5.4.2.3 Tabel Kelurahan

Nama tabel : Kelurahan
 Jumlah field : 5
 Fungsi : untuk menyimpan data kelurahan

Tabel 5.11 Perancangan Tabel Kelurahan

No.	Nama field	Tipe	Lebar	Keterangan
1	idKel	Varchar	6	Id kelurahan
2	namaKel	Varchar	50	Nama kelurahan
3	luas	Integer	-	Luas area kelurahan
4	rumahTangga	Integer	-	Jumlah rumah tangga pada masing-masing kelurahan
5	geometri	Text	-	Titik-titik koordinat lokasi kelurahan

5.4.2.4 Tabel PendKecam

Nama tabel : PendKecam
 Jumlah field : 5
 Fungsi : untuk menyimpan data penduduk pada setiap kecamatan

Tabel 5.12 Perancangan Tabel PendKecam

No.	Nama field	Tipe	Lebar	Keterangan
1	idPKec	Varchar	6	Id penduduk kecamatan
2	jumlahPend	Integer	-	Jumlah penduduk kecamatan
3	katUmurKec	Varchar	5	Kategori penduduk kecamatan tiap tingkat usia (selisih 5 tahun dari usia 0 sampai 75+)
4	jenisKelamin	Char	1	Jenis kelamin penduduk
5	idKec	Varchar	5	Id kecamatan dari tabel kecamatan

5.4.2.5 Tabel PendKelur

Nama tabel : PendKelur
 Jumlah field : 5
 Fungsi : untuk menyimpan data penduduk pada setiap kelurahan

Tabel 5.13 Perancangan Tabel PendKelur

No.	Nama field	Tipe	Lebar	Keterangan
1	idPKel	Varchar	6	Id penduduk kelurahan
2	jumlahPend	Integer	-	Jumlah penduduk kelurahan
3	katUmurKel	Varchar	5	Kategori penduduk kelurahan tiap tingkat usia (selisih 5 tahun dari usia 0 sampai 75+)
4	jenisKelamin	Char	1	Jenis kelamin penduduk
5	idKel	Varchar	6	Id kelurahan dari tabel kelurahan

5.4.2.6 Tabel LokasiBang

Nama tabel : LokasiBang

Jumlah field : 7

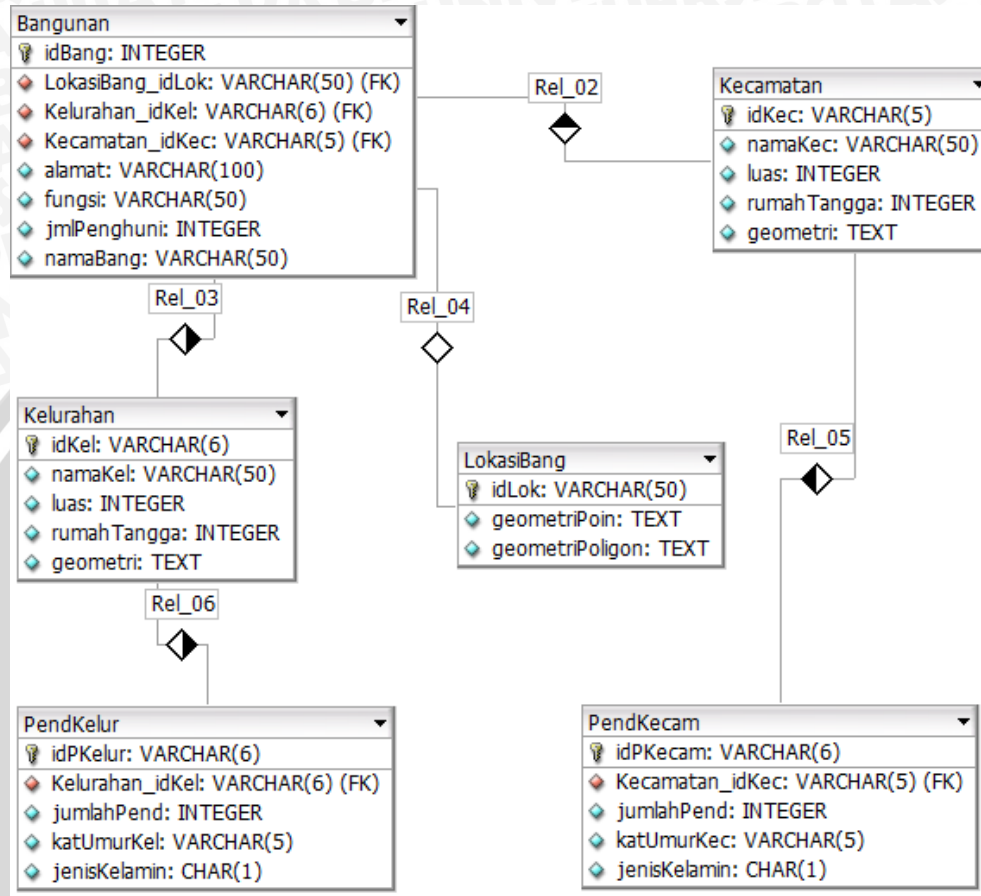
Fungsi : untuk menyimpan data lokasi bangunan

Tabel 5.14 Perancangan Tabel LokasiBang

No.	Nama field	Tipe	Lebar	Keterangan
1	idLok	Varchar	50	Id lokasi bangunan
2	geometriPoin	Text	-	Titik koordinat poin pada bangunan
3	geometriPoligon	Text	-	Titik koordinat poligon pada bangunan
4	idBang	Integer	-	Id bangunan dari tabel bangunan

5.4.3 Physical Data Model

Setelah perancangan tabel dibuat, selanjutnya penggambaran tentang relasi antar tabel beserta atribut-atributnya dengan menggunakan *Physical Data Model*, ditunjukkan pada gambar 5.13 di bawah ini.



Gambar 5.13 Physical Data Model

5.5 Perancangan Antarmuka

Perancangan antarmuka ini menampilkan rancangan antarmuka SIGEKA dan keterangan pada setiap tombol, menu serta tampilan-tampilan khusus untuk SIGEKA. Setiap tombol, menu dan tampilan khusus diberikan urutan nomor dan dijelaskan di dalam tabel setelah gambar rancangan antarmukanya.

5.5.1 Perancangan Antarmuka Awal

Gambar 5.14 merupakan rancangan antarmuka halaman awal ketika sistem pertama kali diakses oleh pengunjung.



Gambar 5.14 Perancangan Antarmuka Awal

Berikut penjelasan rancangan antarmuka awal sistem pada tabel 5.15 berikut ini.

Tabel 5.15 Keterangan Perancangan Antarmuka Awal

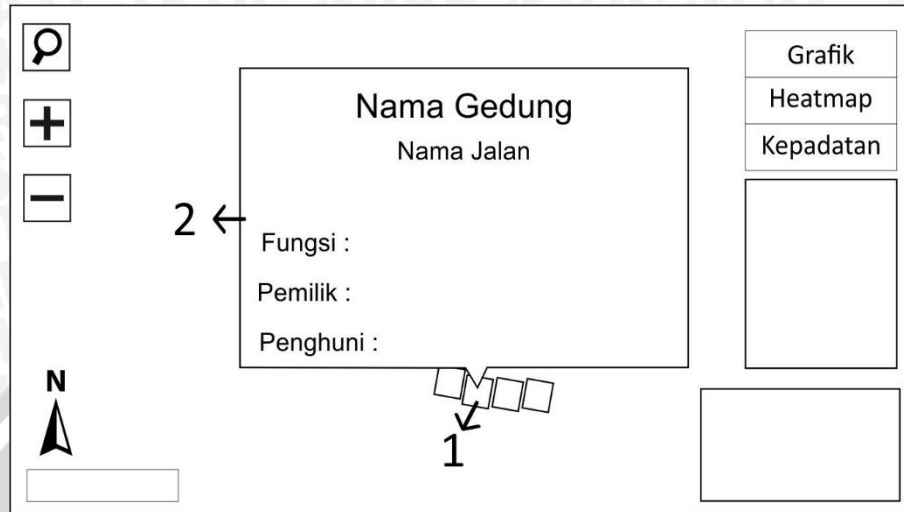
No.	Nama Objek	Tipe	Keterangan
1	Peta	Peta	Menampilkan peta bangunan dan dapat dipilih tiap bangunannya
2	Menu Grafik	Tombol	Untuk menampilkan grafik jumlah penduduk setiap kecamatan dan kelurahan
3	Menu <i>Heatmap</i>	Tombol	Untuk menampilkan <i>heatmap</i> menurut jumlah penghuni setiap bangunan
4	Menu Kepadatan Penduduk	Tombol	Untuk menampilkan kepadatan penduduk setiap kecamatan dan kelurahan

Tabel 5.15 Keterangan Perancangan Antarmuka Awal (lanjutan)

No.	Nama Objek	Tipe	Keterangan
5	Fungsi Bangunan	Tombol	Menampilkan keterangan fungsi bangunan yang terdiri dari : - Hunian - Industri - Jasa - Kesehatan - Pendidikan - Pemerintahan - Perdagangan - Wisata
6	Mini Map	Informasi	Menampilkan peta dalam ukuran yang lebih kecil dengan area yang lebih luas
7	Tombol pencarian	Tombol dan form	Untuk mencari bangunan yang tersedia di peta
8	Memperbesar	Tombol	Untuk memperbesar tampilan peta
9	Memperkecil	Tombol	Untuk memperkecil tampilan peta
10	Penunjuk arah utara	Simbol	Menampilkan penunjuk arah utara pada peta
11	Skala peta	Informasi	Memberikan informasi skala pada tampilan peta

5.5.2 Perancangan antarmuka untuk menampilkan detail bangunan

Berikut ini merupakan rancangan antarmuka untuk menampilkan detail bangunan (Gambar 5.15) beserta penjelasannya (Tabel 5.16).



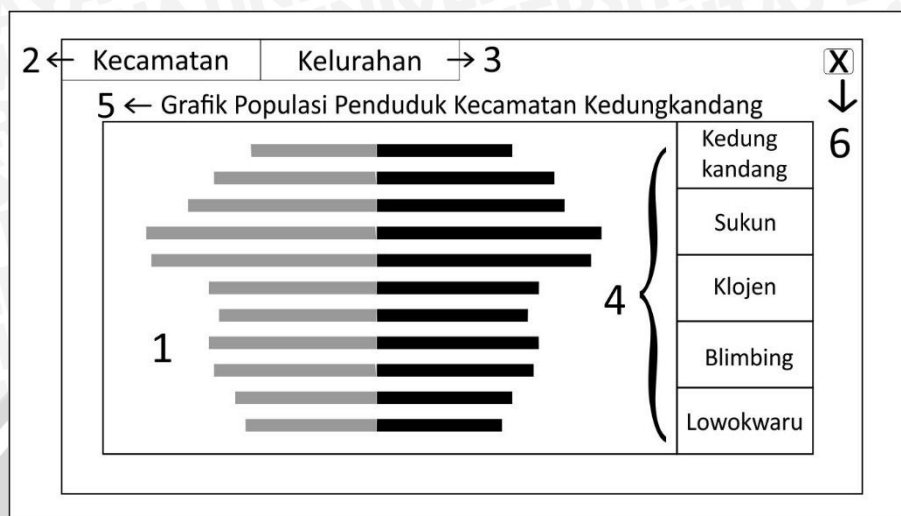
Gambar 5.15 Perancangan Antarmuka Detail Bangunan

Tabel 5.16 Keterangan Perancangan Antarmuka Detail Bangunan

No.	Nama Objek	Tipe	Keterangan
1	Bangunan	Tombol	Bangunan pada peta yang dipilih untuk melihat informasi detailnya
2	Informasi detail bangunan	Teks	Untuk menampilkan informasi detail bangunan yang terdiri dari nama gedung, nama jalan, fungsi dan jumlah penghuni.

5.5.3 Perancangan antarmuka untuk menampilkan grafik

Gambar 5.15 merupakan rancangan antarmuka untuk menampilkan grafik kecamatan atau kelurahan (Gambar 5.16)



Gambar 5.16 Perancangan Antarmuka Grafik

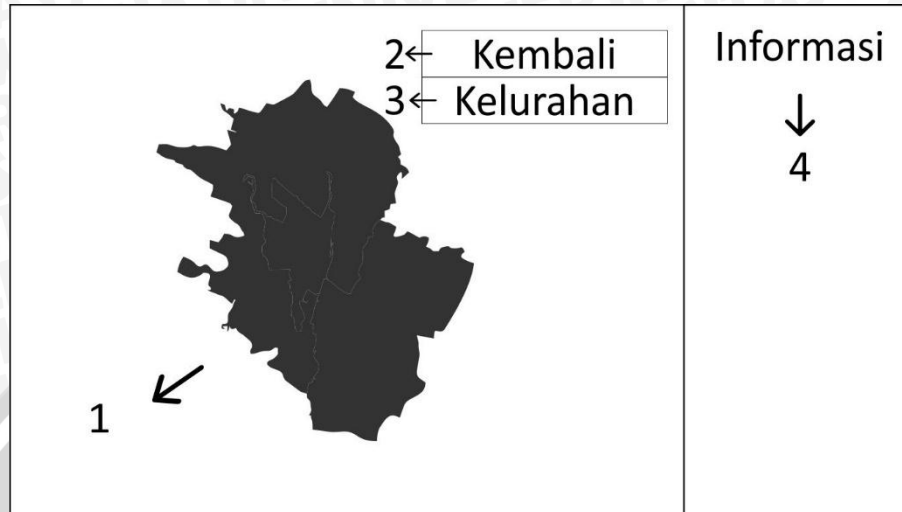
Berikut penjelasan rancangan antarmuka untuk menampilkan grafik kecamatan atau kelurahan pada tabel 5.17 berikut ini.

Tabel 5.17 Keterangan Perancangan Antarmuka Grafik

No.	Nama Objek	Tipe	Keterangan
1	Grafik	Grafik	Grafik jumlah penduduk tiap kecamatan atau kelurahan berdasarkan usia dan jenis kelamin penduduk
2	Menu Kecamatan	Tombol	Untuk menampilkan grafik jumlah penduduk tiap kecamatan
3	Menu Kelurahan	Tombol	Untuk menampilkan grafik jumlah penduduk tiap kelurahan
4	Nama Kecamatan / Nama Kelurahan	Tombol	Nama kecamatan atau kelurahan di kota Malang
5	Judul Grafik	Teks	Menampilkan judul grafik yang sedang ditampilkan
6	Tutup	Tombol	Untuk menutup tampilan grafik

5.5.4 Perancangan antarmuka untuk menampilkan kepadatan penduduk

Berikut ini merupakan rancangan antarmuka untuk menampilkan kepadatan penduduk (Gambar 5.17) beserta penjelasannya (Tabel 5.18).



Gambar 5.17 Perancangan Antarmuka Kepadatan Penduduk

Tabel 5.18 Keterangan Perancangan Antarmuka Kepadatan Penduduk

No.	Nama Objek	Tipe	Keterangan
1	Peta kota Malang	Peta dan tombol	Menampilkan peta kota Malang dengan batas wilayah kecamatan atau kelurahan
2	Kembali	Tombol	Tombol untuk kembali ke menu bangunan (halaman awal sistem)
3	Kelurahan	Tombol	Tombol untuk menuju ke halaman kepadatan wilayah lain (dari halaman kecamatan menuju halaman kelurahan atau sebaliknya)
4	Informasi Kecamatan	Teks	Menampilkan informasi kepadatan tiap kecamatan atau kelurahan yang dipilih

5.5.5 Perancangan antarmuka untuk menampilkan halaman login administrator

Berikut ini merupakan rancangan antarmuka untuk menampilkan login administrator (Gambar 5.18) beserta penjelasannya (Tabel 5.19).

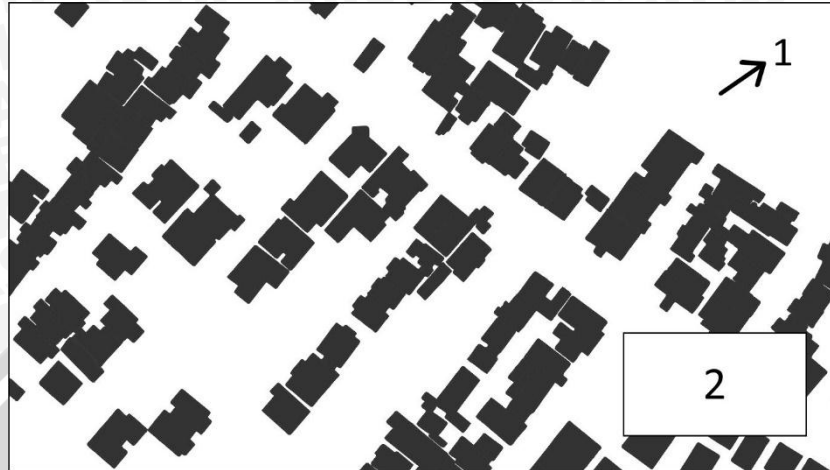
Gambar 5.18 Perancangan Antarmuka Login Administrator

Tabel 5.19 Keterangan Perancangan Antarmuka Login Administrator

No.	Nama Objek	Tipe	Keterangan
1	Status pengguna	Teks	Menampilkan informasi status pengguna sebagai administrator
2	Form Password	Form	Form memasukkan password untuk menuju halaman mengubah data
3	Login	Tombol	Tombol untuk submit password (tombol login)

5.5.6 Perancangan antarmuka untuk menampilkan peta ubah data

Gambar 5.19 merupakan rancangan antarmuka untuk menampilkan peta ubah data.



Gambar 5.19 Perancangan Antarmuka Peta Ubah Data

Berikut penjelasan rancangan antarmuka untuk menampilkan peta ubah data pada tabel 5.20 berikut ini.

Tabel 5.20 Keterangan Perancangan Antarmuka Peta Ubah Data

No.	Nama Objek	Tipe	Keterangan
1	Peta bangunan	Peta dan tombol	Menampilkan peta bangunan dan dapat dipilih tiap bangunannya
2	Mini map	Informasi	Menampilkan peta dalam ukuran yang lebih kecil dengan area yang lebih luas

5.5.7 Perancangan antarmuka untuk menampilkan form ubah data

Berikut ini merupakan rancangan antarmuka untuk menampilkan form ubah data (Gambar 5.20) beserta penjelasannya (Tabel 5.21).

Gambar 5.20 Perancangan antarmuka untuk menampilkan form ubah data

Tabel 5.21 Keterangan Perancangan Antarmuka Form Ubah Data

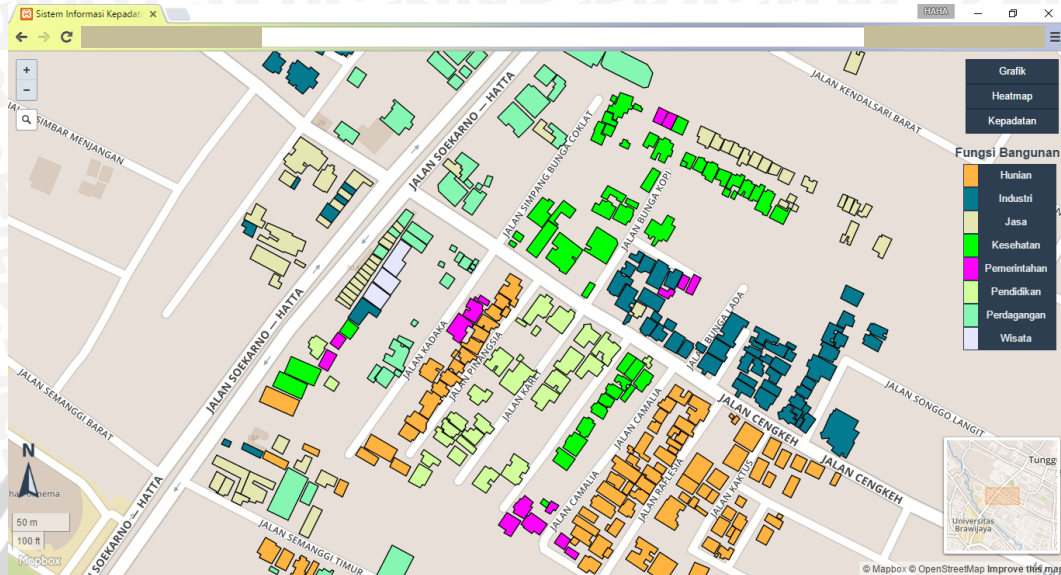
No.	Nama Objek	Tipe	Keterangan
1	ID	Teks	Menampilkan id bangunan yang dipilih untuk diubah datanya
2	Form Ubah Data	Form	Form untuk mengubah data bangunan yang terdiri dari : - Alamat - Nama pemilik - Kecamatan - Fungsi bangunan - Kelurahan - Jumlah penghuni
3	Batal	Tombol	Tombol untuk membatalkan seluruh data yang dimasukkan
4	Simpan	Tombol	Tombol untuk menyimpan data yang dimasukkan
5	Tutup	Tombol	Untuk menutup tampilan ubah data

5.6 Prototipe Antarmuka

Sebelum pembuatan prototipe berdasarkan rancangan antarmuka, pengolahan data menggunakan perangkat lunak GIS dilakukan terlebih dahulu. Perangkat lunak yang digunakan adalah Quantum GIS. Data yang telah didapatkan dari pemerintah berupa data penduduk dan data bangunan akan diolah menjadi data vektor beserta atributnya. Data yang dilakukan digitalisasi terdiri dari tiga data yaitu data bangunan, data wilayah kecamatan dan data wilayah kelurahan. Setelah proses digitalisasi selesai, data akan di ekspor menjadi format web kemudian baru dibuat prototipe sesuai rancangan antarmuka yang telah dibuat sebelumnya.

5.6.1 Prototipe antarmuka Awal

Prototipe antarmuka awal yaitu ketika pengunjung pertama kali mengakses sistem dapat dilihat pada Gambar 5.1 berikut ini.



Gambar 5.21 Prototipe Antarmuka Awal

Pada prototipe antarmuka awal menampilkan peta bangunan dengan menyesuaikan bangunan dengan peta menurut lokasinya dengan menggunakan kode *javascript* pada tabel berikut ini. Selain menyesuaikan berdasarkan lokasinya, mengatur berapa ukuran peta yang akan ditampilkan oleh sistem. Peta yang digunakan oleh sistem berasal dari website *openstreetmap* yang menyediakan peta dasar di seluruh dunia.

Kemudian untuk menampilkan bangunan menggunakan kode *javascript* pada tabel 5.22 berikut ini.

Tabel 5.22 Potongan Kode Prototipe Antarmuka Awal

No	Kode Javascript
1	<code>var map = L.mapbox.map('map', 'mapbox.streets')</code>
2	<code>.setView([-7.945400775523368 , 112.62041355672879],</code> <code>18).on('ready', function() {</code>
3	<code>new L.Control.MiniMap(L.mapbox.tileLayer('mapbox.streets'))</code>
4	<code>.addTo(map);</code>
5	<code>});</code>
6	<code>L.control.scale().addTo(map);</code>
7	<code>var output = document.getElementById('output');</code>
8	<code>var geocode = L.mapbox.geocoderControl('mapbox.places' , {</code>
9	<code>autocomplete: true</code>
10	<code>});</code>
11	<code>geocode.addTo(map);</code>
12	<code>geocode.on('found', function(res) {</code>
13	<code>output.innerHTML = JSON.stringify(res.results.features[0]); });</code>
14	<code>var feature_group = new L.featureGroup([]);</code>
15	<code>function doStylefungsiHunian(feature) {</code>
16	<code>return {</code>
17	<code>color: '#000000',</code>
18	<code>fillColor: '#FFB340',</code>
19	<code>weight: 1.3,</code>
20	<code>dashArray: '',</code>
21	<code>opacity: 1.0,</code>
22	<code>fillOpacity: 1.0</code>
23	<code>};}</code>
24	<code>var exp_fungsiHunianJSON = new L.geoJson(exp_fungsiHunian,{</code>
25	<code>style: doStylefungsiHunian</code>
26	<code>});</code>
27	<code>feature_group.addLayer(exp_fungsiHunianJSON);</code>

Kode *javascript* pada tabel diatas berfungsi untuk menampilkan bangunan-bangunan menurut fungsinya dengan mengambil contoh pada fungsi hunian. Fungsi yang disediakan sistem terdiri dari fungsi untuk hunian, industri, kesehatan, pemerintahan, pendidikan, perdagangan dan wisata. Setiap bangunan dengan fungsi yang berbeda akan diberikan warna yang berbeda pula, kemudian setiap fungsi bangunan akan dikelompokkan menjadi satu supaya setiap bangunan yang memiliki fungsi berbeda akan ditampilkan seluruhnya saat awal diakses oleh pengunjung.

5.6.2 Prototipe antarmuka untuk menampilkan *Heat Map*

Prototipe antarmuka untuk menampilkan *heat map* dapat dilihat pada Gambar 5.22 berikut ini.



Gambar 5.22 Prototipe Antarmuka Untuk Menampilkan *Heat Map*

Berikut potongan kode *javascript* untuk menampilkan *heat map* pada peta (Tabel 5.23).

Tabel 5.23 Potongan Kode Prototipe Antarmuka Menampilkan *Heat Map*

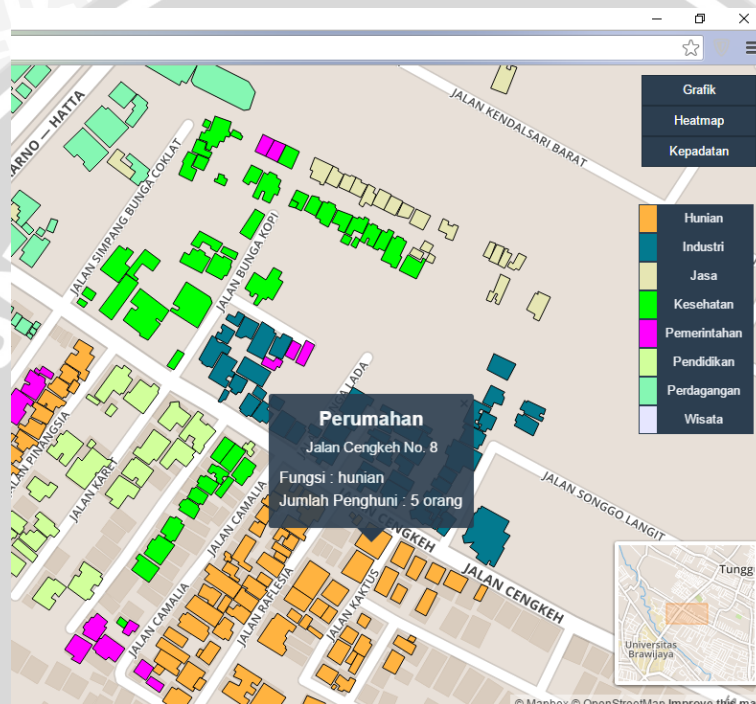
No	Kode Javascript
1	<code>var pointsFeature = exp_poinBangunan.features;</code>
2	<code>var arrPoints = [];</code>
3	<code>for (var i = 0; i < pointsFeature.length; i++) {</code>
4	<code>var temp = [pointsFeature[i].geometry.coordinates[1],</code> <code>pointsFeature[i].geometry.coordinates[0],</code> <code>pointsFeature[i].properties.jmlOrang];</code>
5	<code>arrPoints.push(temp);</code>
6	<code>};</code>
7	<code>var heat = L.heatLayer(arrPoints, {</code>
8	<code>radius: 20,</code>
9	<code>blur: 15,</code>
10	<code>maxZoom: 17,</code>
11	<code>});</code>
12	<code>var pointsFeature = exp_poinBangunan.features;</code>
13	<code>var arrPoints = [];</code>

Tabel diatas merupakan kode javascript untuk menampilkan *heat map*. Kode javascript dalam prototipe ini merupakan kode yang masih disederhanakan dan dapat digunakan untuk menampilkan *heat map* pada SIGEKA. Kode javascript ini dapat diletakkan pada kelas *boundary* AntarmukaPeta yang diakses oleh *method* tampilHeatMap() dalam kelas tersebut. Data yang digunakan untuk menampilkan *heat map* ini berasal dari basis data pada tabel Bangunan yang diakses oleh *entity* Bangunan (*method* mendapatDataHeatMap()) dan Peta Controller (*method* memintaDataHeatmap()) sesuai dengan pemodelan interaksi yang telah digambarkan sebelumnya.

Menampilkan *heat map* ini menggunakan titik-titik setiap bangunan yang telah dibuat kemudian mengambil titik koordinat dan data jumlah penghuninya untuk mengetahui jumlah penghuni tiap bangunan. Setelah itu titik-titik tersebut ditampilkan berupa *heat map* di setiap bangunannya dan disimpan dalam sebuah *array* yang dapat digunakan atau dipanggil oleh fungsi lain (fungsi untuk menampilkan *heat map* ke dalam peta).

5.6.3 Prototipe antarmuka untuk menampilkan detail bangunan

Prototipe antarmuka untuk menampilkan detail bangunan dapat dilihat pada Gambar 5.23 berikut ini.



Gambar 5.23 Prototipe Antarmuka Untuk Menampilkan Detail Bangunan

Berikut potongan kode *javascript* untuk menampilkan detail bangunan pada peta (Tabel 5.24).

Tabel 5.24 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Detail Bangunan

No	Kode Javascript
1	<code>exp_fungsiHunianJSON.eachLayer(function(layer) {</code>
2	<code>var content = '<h1 style="text-align: center;font-weight: bold;font-size:20px">' + layer.feature.properties.namaGedung + '</h1>' + '<h2 style="text-align: center">' + layer.feature.properties.alamat + '</h2>' + 'Fungsi : ' + layer.feature.properties.fungsi + '<br \/>' + 'Jumlah Penghuni : ' + layer.feature.properties.jmlPnghuni + ' orang';</code>
3	<code>layer.bindPopup(content);</code>
4	<code>});</code>

Kode *javascript* di tabel diatas berfungsi untuk menampilkan detail setiap bangunan yang terdiri dari alamat, nama, fungsi dan jumlah penghuni bangunan.

Detail bangunan ditampilkan berupa kotak informasi yang menunjuk pada bangunan yang ingin dilihat detail informasinya.

5.6.4 Prototipe antarmuka untuk menampilkan grafik

Prototipe antarmuka untuk menampilkan grafik tiap kecamatan dapat dilihat pada Gambar 5.24 berikut ini.



Gambar 5.24 Prototipe Antarmuka Untuk Menampilkan Grafik Tiap Kecamatan

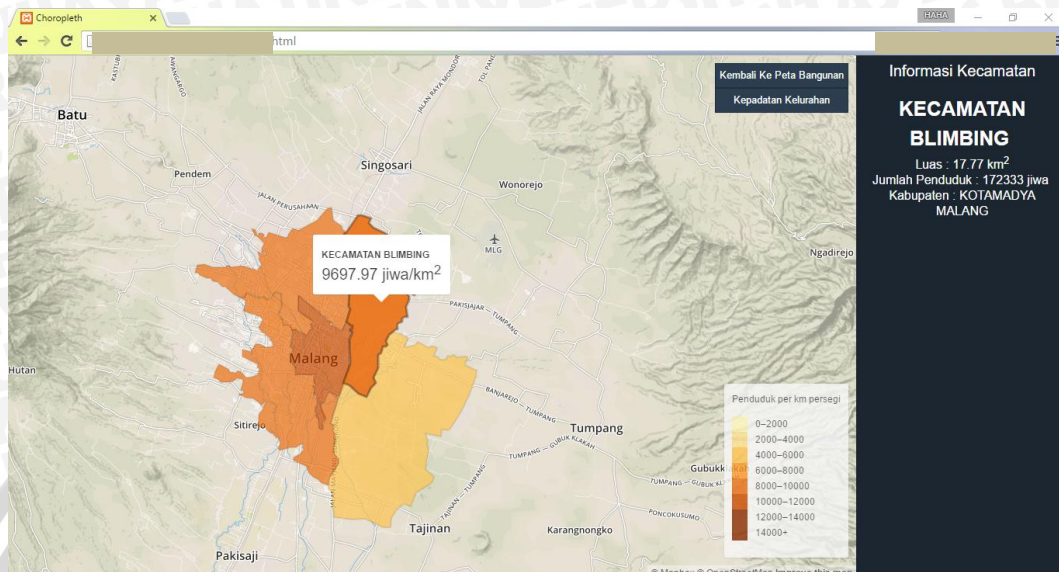
Kemudian prototipe antarmuka untuk menampilkan grafik tiap kelurahan dapat dilihat pada Gambar 5.25 berikut ini.



Gambar 5.25 Prototipe Antarmuka Untuk Menampilkan Grafik Tiap Kelurahan

5.6.5 Prototipe antarmuka untuk menampilkan kepadatan penduduk

Gambar 5.26 berikut ini merupakan prototipe antarmuka untuk menampilkan kepadatan penduduk tiap kecamatan.



Gambar 5.26 Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kecamatan

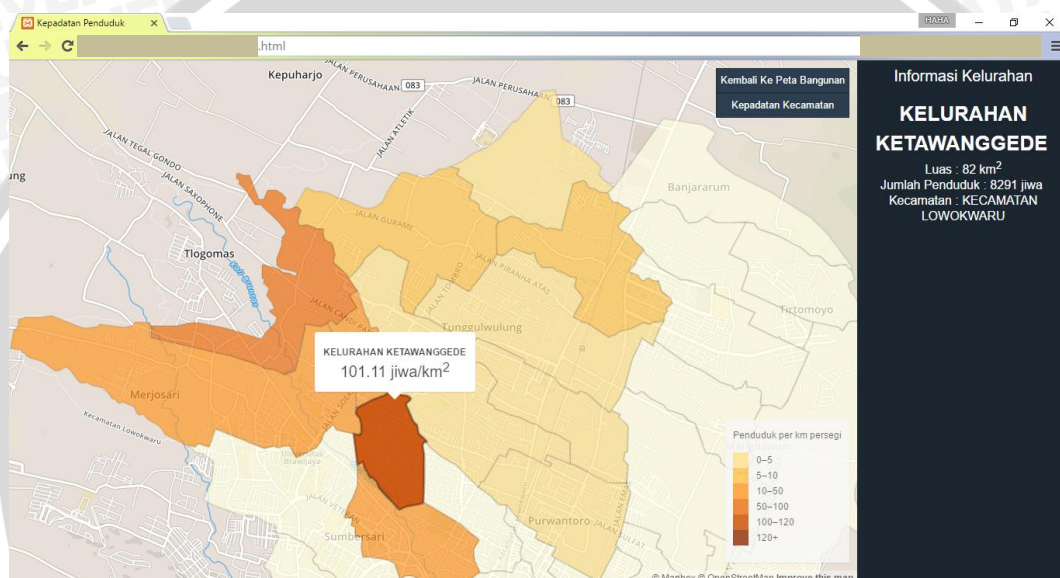
Tabel 5.25 merupakan potongan kode *javascript* untuk menampilkan kepadatan penduduk tiap kecamatan.

Tabel 5.25 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kecamatan

No	Kode Javascript
1	var statesLayer = L.geoJson(exp_DissolveKec, {
2	style: getStyle,
3	onEachFeature: onEachFeature
4	}).addTo(map);
5	function getStyle(feature) {
6	return {
7	weight: 2,
8	opacity: 0.1,
9	color: 'black',
10	fillOpacity: 0.7,
11	fillColor: getColor(feature.properties.DENSITY) }; }
12	function getColor(d) {
13	return d > 12000 ? '#8c2d04' :
14	d > 10000 ? '#cc4c02' :
15	d > 8000 ? '#ec7014' :
16	d > 6000 ? '#fe9929' :
17	d > 4000 ? '#fec44f' :
18	d > 2000 ? '#fee391' :
19	d > 0 ? '#fff7bc' :
20	'#ffffe5'; }
21	function onEachFeature(feature, layer) {
22	layer.on({
23	mousemove: mousemove,
24	mouseout: mouseout,
25	click: zoomToFeature }); }
26	var closeTooltip;
27	function mousemove(e) {
28	var layer = e.target;
29	popup.setLatLng(e.latlng);
30	popup.setContent('<div class="marker-title">' + layer.feature.properties.KECAMATAN + '</div><div class="ui-title">' + layer.feature.properties.DENSITY + ' jiwa/km²</div>');
31	info.innerHTML = '<div class="ui-label"><label>Informasi Kecamatan</label></div> <div class="ui-content">' + layer.feature.properties.KECAMATAN + ' </div> Luas : ' + layer.feature.properties.AREA + ' km²</div> Jumlah Penduduk : ' + layer.feature.properties.POPULATION + ' jiwa </div> Kabupaten : ' + layer.feature.properties.KABUPATEN + '</div>';
32	if (!popup._map) popup.openOn(map);
33	window.clearTimeout(closeTooltip);
34	layer.setStyle({
35	weight: 3,
36	opacity: 0.3,
37	fillOpacity: 0.9});
38	if (!L.Browser.ie && !L.Browser.opera) {
39	layer.bringToFront();
40	}}
41	function mouseout(e) {
42	statesLayer.resetStyle(e.target);
43	closeTooltip = window.setTimeout(function() {
44	map.closePopup();
45	}, 100); }
46	function zoomToFeature(e) {
47	map.fitBounds(e.target.getBounds());}
48	map.legendControl.addLegend(getLegendHTML());

Kode pada Tabel 5.25 berfungsi untuk menampilkan peta wilayah seluruh kecamatan dengan setiap kecamatannya memiliki tingkat kepadatan berbeda akan diberikan warna menurut tingkat kepadatannya. Penetapan warna setiap wilayah dibedakan menurut tingkatnya yang termasuk dalam rentang ukuran pada legenda yang ditampilkan. Kemudian, setiap wilayah jika disorot akan menampilkan nama kecamatan dan ukuran kepadatannya dengan satuan jiwa per kilometer persegi (jiwa/km^2) menggunakan rumus kepadatan penduduk pada persamaan 1 beserta detail informasi tambahan pada bagian kanan halaman yaitu luas wilayah dan jumlah penduduk.

Gambar 5.27 berikut ini merupakan prototipe antarmuka untuk menampilkan kepadatan penduduk tiap kelurahan.



Gambar 5.27 Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kelurahan

Tabel 5.26 merupakan potongan kode *javascript* untuk menampilkan kepadatan penduduk tiap kelurahan.

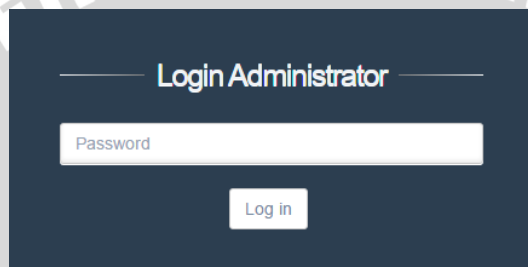
Tabel 5.26 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Kepadatan Penduduk Tiap Kelurahan

No	Kode Javascript
1	var statesLayer = L.geoJson(exp_jatim, {
2	style: getStyle,
3	onEachFeature: onEachFeature
4	}).addTo(map);
5	function getStyle(feature) {
6	return {
7	weight: 2,
8	opacity: 0.1,
9	color: 'black',
10	fillOpacity: 0.7,
11	fillColor: getColor(feature.properties.DENSITY) }; }
12	function getColor(d) {
13	return d > 120 ? '#8c2d04' :
14	d > 100 ? '#cc4c02' :
15	d > 50 ? '#ec7014' :
16	d > 10 ? '#fe9929' :
17	d > 5 ? '#fec44f' :
18	d > 0 ? '#fee391' :
19	'#ffffe5';
20	}
21	function onEachFeature(feature, layer) {
22	layer.on({
23	mousemove: mousemove,
24	mouseout: mouseout,
25	click: zoomToFeature });}
26	var closeTooltip;
27	function mousemove(e) {
28	var layer = e.target;
29	popup.setLatLng(e.latlng);
30	popup.setContent('<div class="marker-title">' + layer.feature.properties.KELURAHAN + '</div><div class="ui- title">' +
31	layer.feature.properties.DENSITY + ' jiwa/km²</div>');}
32	info.innerHTML = '<div class="ui-label"><label>Informasi Kelurahan</label></div><div class="ui-content">' + layer.feature.properties.KELURAHAN + ' Luas : ' + layer.feature.properties.LARGE + ' km²</div> Jumlah Penduduk : ' + layer.feature.properties.POPULATION + ' jiwa Kecamatan : ' + layer.feature.properties.KECAMATAN + '</div>';
33	if (!popup._map) popup.openOn(map);
34	window.clearTimeout(closeTooltip);
35	layer.setStyle({
36	weight: 3,
37	opacity: 0.3,
38	fillOpacity: 0.9 });}
39	
40	if (!L.Browser.ie && !L.Browser.opera) {
41	layer.bringToFront();}
42	function mouseout(e) {
43	statesLayer.resetStyle(e.target);
44	closeTooltip = window.setTimeout(function() {
45	map.closePopup();
46	}, 100); }
47	function zoomToFeature(e) {
48	map.fitBounds(e.target.getBounds());
49	}
50	map.legendControl.addLegend(getLegendHTML());

Tabel 5.26 merupakan kode *javascript* yang berfungsi untuk menampilkan peta wilayah seluruh kelurahan dengan setiap kelurahannya memiliki tingkat kepadatan berbeda akan diberikan warna menurut tingkat kepadatannya. Penetapan warna setiap wilayah dibedakan menurut tingkatnya yang termasuk dalam rentang ukuran pada legenda yang ditampilkan. Kemudian, setiap wilayah jika disorot akan menampilkan nama kecamatan dan ukuran kepadatannya dengan satuan jiwa per kilometer persegi (jiwa/km^2) menggunakan rumus kepadatan penduduk pada persamaan 1 beserta detail informasi tambahan pada bagian kanan halaman yaitu luas wilayah dan jumlah penduduk.

5.6.6 Prototipe antarmuka untuk menampilkan halaman login administrator

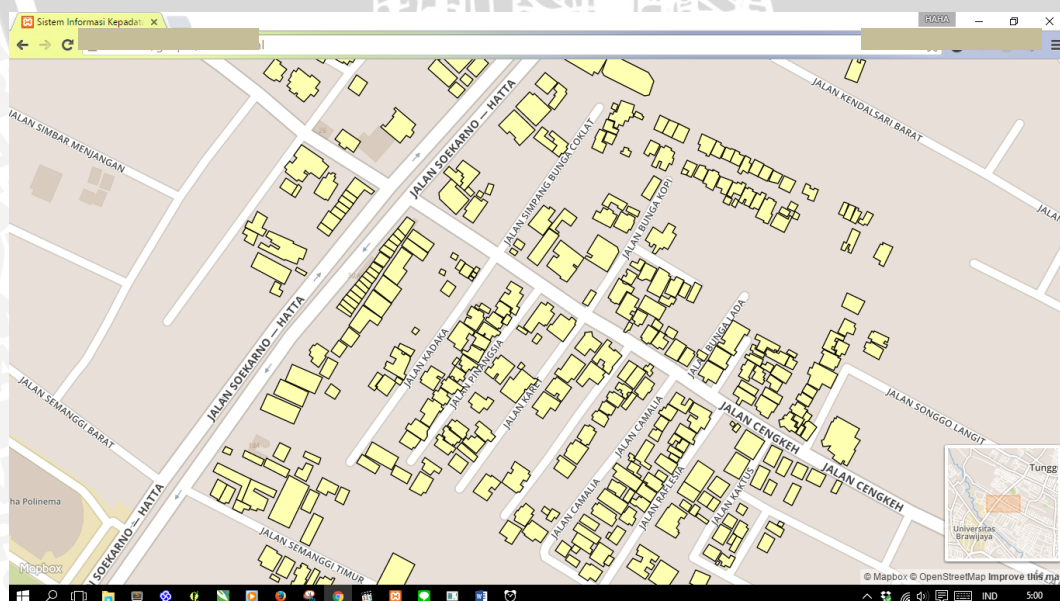
Prototipe antarmuka yang menampilkan halaman login untuk administrator dapat dilihat pada Gambar 5.28 di bawah ini.



Gambar 5.28 Prototipe Antarmuka Untuk Menampilkan Halaman Login Administrator

5.6.7 Prototipe antarmuka untuk menampilkan peta ubah data

Prototipe antarmuka yang menampilkan peta ubah data untuk administrator dapat dilihat pada Gambar 5.29 di bawah ini.



Gambar 5.29 Prototipe Antarmuka Untuk Menampilkan Peta Ubah Data

Tabel 5.27 berikut ini merupakan potongan kode *javascript* yang menampilkan peta ubah data untuk administrator.

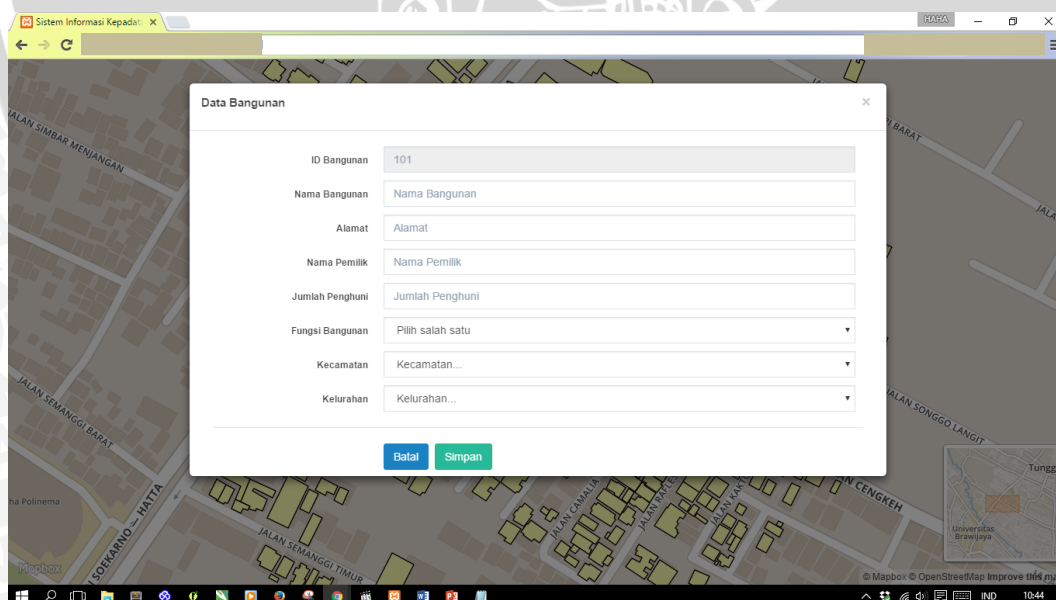
Tabel 5.27 Potongan Kode Prototipe Antarmuka Untuk Menampilkan Peta Ubah Data

No	Kode Javascript
1	<code>var feature_group = new L.featureGroup([]);</code>
2	<code>function doStyleatributGedung(feature) {</code>
3	<code>return {</code>
4	<code>color: '#000000',</code>
5	<code>fillColor: '#ffffb2',</code>
6	<code>weight: 1.3,</code>
7	<code>dashArray: '',</code>
8	<code>opacity: 1.0,</code>
9	<code>fillOpacity: 1.0</code>
10	<code>};</code>
11	<code>}</code>
12	<code>var exp_atributGedungJSON = new L.geoJson(exp_atributGedung, {</code>
13	<code>style: doStyleatributGedung</code>
14	<code>});</code>
15	<code>feature_group.addLayer(exp_atributGedungJSON);</code>
16	<code>feature_group.addTo(map);</code>

Kode *javascript* pada tabel di atas, setiap atribut bangunan akan diakses yang terdiri dari id, nama gedung, alamat, nam pemilik, fungsi dan jumlah bangunan, kemudian disimpan dalam sebuah variabel dengan nama `popupContent` dan ditampilkan ke dalam peta dengan memberikan warna pada masing-masing bangunannya.

5.6.8 Prototipe antarmuka untuk menampilkan form ubah data

Prototipe antarmuka yang menampilkan form ubah data untuk administrator dapat dilihat pada Gambar 5.30 di bawah ini.



Gambar 5.30 Prototipe Antarmuka Untuk Menampilkan Form Ubah Data

BAB 6 HASIL EVALUASI

Setelah analisis persyaratan dan rancangan sistem, dilakukan evaluasi *traceability* dan evaluasi pada prototipe. Evaluasi *traceability* yaitu evaluasi pada keruntutan dari kebutuhan pemangku kepentingan, fitur-fitur, persyaratan fungsional dan non fungsional hingga *use case* yang dibuat, kemudian dibuat pertanyaan berkaitan dengan keruntutan pada dokumen. Lalu untuk evaluasi prototipe terdiri dari evaluasi berdasarkan *use case* dan berdasarkan penilaian pengguna.

6.1 Evaluasi *Traceability*

Evaluasi *traceability* ini bertujuan untuk memeriksa keruntutan dari proses bisnis, kebutuhan pemangku kepentingan, fitur, persyaratan fungsional dan *use case*. Keruntutan ini diperiksa untuk memastikan keberadaan setiap proses pada analisis persyaratan telah disusun sesuai dengan kebutuhan pemangku kepentingan, dapat dilacak dan tidak mengandung kode yang duplikat. Sebelum menjawab pertanyaan yang berkaitan dengan daftar pertanyaan untuk *traceability*, dilakukan pemeriksaan terlebih dulu pada aktivitas proses bisnis yang menggunakan sistem, kemudian setiap kebutuhan, fitur, persyaratan fungsional dan *use case* untuk memeriksa keruntutannya.

Aktivitas proses bisnis yang menggunakan sistem adalah aktivitas Melihat data bangunan kota pada sistem yang memiliki kode PB-U-001. Aktivitas tersebut memiliki keruntutan dengan kebutuhan proses analisis dan kode fiturnya yaitu FITUR1. Tabel 6.1 merupakan tabel pemeriksaan keruntutan pada aktivitas proses bisnis yang berhubungan dengan sistem.

Tabel 6.1 Keruntutan Aktivitas Proses Bisnis, Fitur dan Persyaratan Fungsional

Kode Aktivitas	Aktivitas	Fitur	Kode Dasar Fitur
PB-U-001	Melihat data bangunan kota pada sistem	FITUR1	F-SIGEKA-001

Pada tabel 6.2 di bawah ini merupakan tabel untuk pemeriksaan kerunutan kebutuhan, fitur, persyaratan fungsional dan *use case* menggunakan kode-kode yang tersedia di dalam dokumen.

Tabel 6.2 Kerunutan Kebutuhan, Fitur, Persyaratan Fungsional dan *Use case*

Kebutuhan	Kode Fitur	Kode Dasar Fitur	Kode Lengkap Fungsi	Kode <i>Use case</i>	Nama <i>Use Case</i>	Prototipe
Proses Analisis	FITUR1	F-SIGEKA-001	F-SIGEKA-001-1	UC-005	Melihat Peta	Gambar 5.21 Gambar 5.22
			F-SIGEKA-001-2			
			F-SIGEKA-001-3			
	FITUR2	F-SIGEKA-002	F-SIGEKA-002-1			
Info jumlah penduduk	FITUR3	F-SIGEKA-003	F-SIGEKA-003-1	UC-001	Melihat Grafik Jumlah Penduduk Kecamatan	Gambar 5.24
Perbandingan jumlah penduduk			F-SIGEKA-003-2	UC-002	Melihat Grafik Jumlah Penduduk Kelurahan	Gambar 5.25
Daerah padat penduduk	FITUR4	F-SIGEKA-004	F-SIGEKA-004-1	UC-006	Melihat Kepadatan Penduduk Kecamatan	Gambar 5.26
			F-SIGEKA-004-2	UC-007	Melihat Kepadatan Penduduk Kelurahan	Gambar 5.27
Detail info	FITUR5	F-SIGEKA-005	F-SIGEKA-005-1	UC-003	Mencari Bangunan	Gambar 5.21
	FITUR6	F-SIGEKA-006	F-SIGEKA-006-1	UC-004	Melihat Detail Bangunan	Gambar 5.23
			F-SIGEKA-006-2	UC-005	Melihat Peta	Gambar 5.21
			F-SIGEKA-006-3			
			F-SIGEKA-006-4			
Memperbarui informasi jumlah penduduk	FITUR7	F-SIGEKA-007	F-SIGEKA-007-1	UC-008	Mengubah Data Bangunan	Gambar 5.28 Gambar 5.29

Kemudian pada daftar pertanyaan *traceability* sistem berisi tentang identifikasi persyaratan dan proses pelacakan setiap persyaratan. Berikut tabel daftar pertanyaan evaluasi *traceability* beserta jawaban dan penjelasannya (Tabel 6.3).

Tabel 6.3 Traceability

No.	Pertanyaan	✓	Keterangan
1	Apakah setiap persyaratan memiliki kode unik yang membedakan antar persyaratan dan diidentifikasi dengan benar sehingga dapat digunakan untuk tujuan <i>traceability</i> ?	✓	Kode setiap persyaratan sudah dituliskan dengan kombinasi huruf dan angka yang berbeda
3	Apakah setiap persyaratan tidak terjadi duplikasi?	✓	Setiap persyaratan tidak ada yang duplikat atau sudah memiliki nama yang berbeda
4	Apakah setiap kebutuhan dapat dilacak pada setiap persyaratan ke arah belakang menuju definisi yang lebih umum atau diteruskan menuju tingkat yang lebih detail?	✓	Setiap kebutuhan dapat dilacak dari kebutuhan dengan definisi yang umum sampai definisi lebih detail seperti dari kebutuhan pemangku kepentingan, fitur, persyaratan fungsional sampai <i>use case</i> . Kemudian dapat dilacak secara mundur yaitu dari persyaratan dengan definisi detail menuju kebutuhan dengan definisi yang lebih umum seperti dari <i>use case</i> , persyaratan fungsional, fitur hingga kebutuhan pemangku kepentingan.
5	Apakah setiap persyaratan fungsional dapat dilacak ke persyaratan yang lebih detail?	✓	Persyaratan fungsional dapat dilacak ke persyaratan yang lebih detail seperti pada pemodelan <i>use case</i>

6.2 Evaluasi Prototipe

Evaluasi pada prototipe terdiri dari dua tahap yaitu evaluasi berdasarkan *use case* dan evaluasi berdasarkan penilaian yang dilakukan oleh pengguna. Evaluasi *use case* dilakukan dengan mendaftar aktivitas atau melakukan observasi secara langsung dengan mengamati apa yang dilakukan oleh pengguna (sebagai partisipan) pada sistem di setiap *use case* dan membandingkannya dengan alur dasar (*basic flow*) pada setiap *use case*-nya. Kemudian untuk evaluasi berdasarkan penilaian pengguna dilakukan dengan cara menyediakan pertanyaan-pertanyaan dan pemberian saran atau kritik oleh pengguna dengan memberikan kuesioner kepada partisipan atau pengguna tersebut.

6.2.1 Evaluasi Berdasarkan *Use case*

Evaluasi *use case* menggunakan empat *use case* utama yaitu *use case* Melihat Peta, *use case* Melihat Grafik Penduduk Kecamatan, *use case* Melihat

Kepadatan Penduduk Kelurahan dan *use case* Mengubah Data. Evaluasi pada *use case* melihat peta dijelaskan pada tabel 6.4 di bawah ini.

Tabel 6.4 Evaluasi *Use case* Melihat Peta

<i>Use case</i>	Melihat Peta	
<i>Test Case</i>	Aktivitas melihat peta dengan fitur tambahannya	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Kesesuaian
Mengakses sistem	Sistem menampilkan peta	Langkah-langkah yang dilakukan pengguna sudah sesuai walaupun tidak sama urutan pada basic flow
Memperbesar atau memperkecil	Memilih opsi berdasarkan fungsi bangunan	
Sistem menampilkan ukuran peta sesuai keinginan	Sistem menampilkan sesuai opsi yang dipilih	
Memilih opsi berdasarkan fungsi bangunan	Memilih opsi heatmap	
Sistem menampilkan sesuai opsi yang dipilih	Sistem menampilkan heatmap	
Memilih opsi heatmap	Memperbesar atau memperkecil	
Sistem menampilkan heatmap	Sistem menampilkan ukuran peta sesuai keinginan	

Kemudian untuk evaluasi pada *use case* melihat grafik penduduk kecamatan dijelaskan pada tabel 6.5 di bawah ini.

Tabel 6.5 Evaluasi *Use case* Melihat Grafik Penduduk Kecamatan

<i>Use case</i>	Melihat Grafik Penduduk Kecamatan	
<i>Test Case</i>	Aktivitas melihat grafik dari jumlah penduduk tiap kecamatan	
Langkah-Langkah Pengguna	Langkah-Langkah Basic dan Alternative Flow	Kesesuaian
Memilih opsi grafik	Memilih opsi grafik penduduk	Sudah sesuai
Melihat grafik dengan memilih kecamatan	Sistem menampilkan grafik	

Setelah itu untuk evaluasi pada *use case* melihat kepadatan penduduk kelurahan dijelaskan pada tabel 6.6 di bawah ini.

Tabel 6.6 Evaluasi *Use case* Melihat Kepadatan Penduduk Kelurahan

<i>Use case</i>	Melihat Kepadatan Penduduk Kelurahan	
<i>Test Case</i>	Aktivitas melihat kepadatan penduduk tiap kelurahan	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Kesesuaian
Memilih opsi kepadatan	Memilih opsi kepadatan penduduk	Sudah sesuai
Melihat kepadatan dengan memilih kelurahan	Sistem menampilkan kepadatan	

Kemudian untuk evaluasi pada *use case* mengubah data terdiri dari tiga bagian dengan kasus uji yang berbeda. Kasus uji pertama yaitu pengguna memasukan password dan format data yang sesuai dengan ketentuan pada sistem dijelaskan pada tabel 6.7 di bawah ini.

Tabel 6.7 Evaluasi *Use case* Mengubah Data Bagian Pertama

Use case	Mengubah Data	
Test Case	Aktivitas mengubah data dengan memasukkan password dan format data yang dimasukkan sesuai dengan ketentuan pada sistem	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Kesesuaian
Memasukkan password keamanan	Sistem menampilkan form untuk memasukkan password keamanan	Sudah sesuai
Melihat bangunan	Memasukkan password keamanan	
Memilih bangunan	Sistem menampilkan peta bangunan kota Malang	
Mengubah data	Memilih bangunan	
Menyimpan data	Sistem menampilkan form ubah data	
	Mengubah data	
	Menyimpan data	

Kasus uji kedua yaitu pengguna memasukan pasword yang tidak sesuai dengan ketentuan pada sistem dijelaskan pada tabel 6.8 di bawah ini.

Tabel 6.8 Evaluasi *Use case* Mengubah Data Bagian Kedua

Use case	Mengubah Data	
Test Case	Aktivitas mengubah data dengan memasukkan password salah	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Kesesuaian
Memasukkan password	Sistem menampilkan form untuk memasukkan password keamanan	Sudah sesuai
Memasukkan kembali password	Memasukkan password keamanan	
Melihat bangunan	Sistem tidak mengijinkan masuk	
Memilih bangunan	Memasukkan kembali password	
Mengubah data	Sistem menampilkan peta bangunan kota Malang	
Menyimpan data	Memilih bangunan	
	Sistem menampilkan form ubah data	
	Mengubah data	
	Menyimpan data	

Kasus uji kedua yaitu pengguna memasukkan data yang tidak sesuai dengan format yang ditentukan pada sistem dijelaskan pada tabel 6.9 di bawah ini.

Tabel 6.9 Evaluasi *Use case* Mengubah Data Bagian Ketiga

Use case	Mengubah Data	
Test Case	Mengubah data dengan data yang dimasukkan tidak sesuai dengan format yang ditetapkan pada sistem	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Kesesuaian
Memasukkan password keamanan	Sistem menampilkan form untuk memasukkan password keamanan	Harus mengulangi 2 kali untuk menyesuaikan format data yang dimasukkan agar sesuai dengan format data yang diterima oleh sistem
Melihat bangunan	Memasukkan password keamanan	
Memilih bangunan	Sistem menampilkan peta bangunan kota Malang	
Mengubah data	Memilih bangunan	
Menyesuaikan masukan data yang lebih sesuai dengan format	Sistem menampilkan form ubah data	
Memasukkan kembali data yang lebih sesuai	Mengubah data	
Menyimpan data	Memasukkan data kembali yang sesuai dengan format	
	Menyimpan data	

6.2.2 Evaluasi Berdasarkan Penilaian Pengguna

Evaluasi berdasarkan penilaian pengguna mengacu pada pedoman pembuatan *website usability evaluation* dengan metode yang digunakan yakni evaluasi inspeksi heuristik yang terdiri dari *content relevance*, *interface design*, *navigation/menus* dan *technical performance*.

Evaluasi penilaian pengguna terdiri dari tiga bagian yaitu evaluasi terhadap harapan alur pengguna, evaluasi terhadap navigasi dan evaluasi terhadap antarmuka. Evaluasi ini dilakukan oleh pengguna di Bappeda Kota Malang bidang Tata kota. Hasil penilaian pada fitur sistem yang didapatkan dari pengguna yaitu menekankan pada tampilan kategori warna bangunan dan informasi yang disajikan pada setiap bangunan. Untuk fitur khusus pada administrator yaitu mengubah data bangunan, pengguna menerima alur, navigasi dan tampilan dari antarmuka pada fitur tersebut. Kemudian untuk penilaian antarmuka pada fitur lainnya dijelaskan pada tiga tabel di bawah ini.

Bagian pertama yaitu evaluasi terhadap harapan alur pengguna dijelaskan pada tabel 6.10 di bawah ini.

Tabel 6.10 Evaluasi Terhadap Harapan Alur Pengguna

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang diberikan oleh pengguna sudah sesuai dengan harapan	✓			Respon sistem sudah sesuai harapan
Alur penggunaan system sudah sesuai dengan harapan pengguna	✓			Alur sudah sesuai harapan
Masukan dan Saran	-			

Bagian kedua yaitu evaluasi terhadap navigasi dijelaskan pada tabel 6.11 berikut ini.

Tabel 6.11 Evaluasi Terhadap Navigasi

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui peran saya terhadap sistem	✓			Kurang penambahan keterangan status sebagai admin untuk halaman khusus admin
Sistem menyediakan label menu yang mudah dipahami untuk mendapatkan informasi tertentu	✓			Label menu mudah dipahami
Sistem mudah digunakan dengan menggunakan tombol untuk menuju ke antarmuka yang lain	✓			Tombol untuk kembali mudah digunakan
Sistem tidak membuka banyak tab atau jendela browser ketika digunakan	✓			Window sudah sesuai, tidak membuka tab terlalu banyak
Peletakan menu atau link dapat dengan mudah dikenali	✓			Peletakan menu mudah dikenali
Navigasi untuk mengatur ukuran peta mudah digunakan	✓			Navigasi untuk mengukur peta mudah digunakan
Masukan dan Saran	- Letak navigasi untuk mengatur ukuran peta sebaiknya berada di sebelah kanan bawah pada peta - Sebaiknya ganti antarmuka menu yang berupa tulisan dengan <i>icon</i> atau gambar			

Kemudian untuk bagian ketiga yaitu evaluasi terhadap antarmuka dijelaskan pada tabel 6.12 di bawah ini.

Tabel 6.12 Evaluasi Terhadap Antarmuka

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			Antarmuka mudah dipelajari
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			Fitur tidak mengandung animasi berulang
Antarmuka peta sudah memberikan informasi yang jelas	✓			Peta sudah jelas
Pemilihan warna pada peta sudah jelas dan mudah dipahami		✓		Pemilihan warna sudah jelas perbedaannya, namun sebaiknya disesuaikan dengan ketentuan yang sudah ada
Sistem memiliki antar muka yang konsisten	✓			Antarmuka sistem sudah konsisten
Masukan dan Saran	- Warna fungsi bangunan sesuaikan dengan aturan dari peraturan pemerintah - Sebaiknya ditambahkan informasi untuk luasan pada setiap bangunan - Kemudian tambahkan informasi jumlah lantai setiap bangunan seperti rumah, apartemen dan kawasan perdagangan			

BAB 7 PENUTUP

Penelitian Analisis dan Perancangan Sistem informasi Geografis Kepadatan Penduduk dengan Metode *Heat Map* menghasilkan empat butir kesimpulan dan dua saran yang peneliti dapatkan.

7.1 Kesimpulan

1. Model proses bisnis dalam SIGEKA ini terdiri dari proses bisnis saat ini dan proses bisnis yang diusulkan. Proses bisnis saat ini diawali dengan badan pemerintah yaitu Bappeda bidang Tata Kota yang akan membuat rencana tata ruang di kota Malang. Bappeda akan mengajukan permohonan untuk data bangunan yang disimpan oleh dinas pemerintah yaitu DPUPPB bidang Tata Ruang supaya dapat memulai untuk membuat rencana tata ruang kota. Setelah tata ruang kota sudah selesai dirancang, Bappeda akan mengirimkannya kepada DPUPPB untuk disesuaikan dengan RDTR (Rencana Detail Tata Ruang). Setelah diterima, rencana yang dibuat oleh Bappeda akan dihitung oleh DPUPPB sesuai RDTR yang sudah ditetapkan. Jika rencana yang dibuat masih terdapat kesalahan maka rencana tata ruang akan dikirimkembali dengan penjelasan revisi yang salah, jika sudah sesuai maka DPUPPB akan menetapkan rencana tata ruang kota dan memulai pekerjaan lapangan. Proses bisnis usulan akan menghemat dua proses yaitu mengajukan permohonan data oleh Bappeda dan mengirim data bangunan oleh DPUPPB menjadi sistem yang menampilkan kondisi bangunan kota saat ini yaitu SIGEKA.
2. Persyaratan sistem untuk membangun WebGIS kepadatan penduduk yaitu SIGEKA ini terdiri dari hasil analisis permasalahan, analisis pemangku kepentingan, kebutuhan pemangku kepentingan, 8 fitur, 14 persyaratan fungsional dan 1 persyaratan non fungsional, 8 *use case* beserta spesifikasinya dan diagram aktivitas.
3. Rancangan dan prototipe yang sesuai dengan persyaratan sistem terdiri dari kelas analisis dari *use case*, kelas analisis dari mekanisme analisis, unifikasi kelas dari *use case*, penjelasan elemen desain, pemodelan interaksi, pemodelan data, dan perancangan antarmuka.
4. Hasil evaluasi *traceability* pada penelitian ini berupa tabel keruntutan terhadap kesesuaian antara kebutuhan, fitur dan persyaratan yang memiliki kode unik agar mudah dilacak pada dokumen. Hasil dari tabel keruntutan menunjukkan bahwa setiap kode sudah runut, dituliskan dengan kombinasi huruf dan angka yang berbeda, setiap persyaratan dapat dilacak, tidak terdapat kode yang duplikat dan setiap persyaratan sudah memiliki nama yang berbeda. Kemudian untuk hasil evaluasi prototipe didapatkan hasil sebagai berikut:
 - a. Evaluasi berdasarkan *use case* menunjukkan bahwa pada *use case* Melihat Peta masih terdapat beberapa langkah yang dilakukan oleh

pengguna yang tidak sama urutannya dengan alur dasar. Setelah itu pada *use case* Mengubah Data, pengguna memasukkan data sebanyak dua kali untuk menyesuaikannya dengan format yang dikenali sistem. Kemudian untuk *use case* lain yang dievaluasi yaitu *use case* Melihat Grafik Penduduk Kecamatan dan Melihat Kepadatan Penduduk Kelurahan sudah sesuai antara langkah-langkah yang dilakukan pengguna dengan alur dasar.

- b. Evaluasi dengan masukan dari pengguna terhadap harapan alur pengguna, navigasi dan antarmuka sistem. Pada evaluasi berdasarkan penilaian pengguna ini terdapat saran pada navigasi sistem yaitu letak navigasi untuk mengatur ukuran peta sebaiknya berada di sebelah kanan bawah pada peta dan antarmuka pilihan menu yang berupa tulisan sebaiknya diganti dengan gambar (*icon*). Saran untuk antarmuka pengguna sistem yaitu pemilihan warna pada fungsi bangunan sebaiknya disesuaikan dengan aturan dari pemerintah dan penambahan informasi luas setiap bangunan dan informasi jumlah lantai bangunan.

7.2 Saran

1. Penelitian ini yang berisi analisis persyaratan dan perancangan sistem dapat digunakan untuk perancangan detail dan implementasi Sistem Informasi Kepadatan Penduduk
2. Fitur-fitur dapat ditambahkan lagi berupa fitur untuk melakukan perhitungan kepadatan bangunan dengan luas area tertentu.

DAFTAR PUSTAKA

Ambler, S. W., 2014. *Physical Data Model (PDM)s: An Agile Introduction*. [Online] Available at: <http://agilemodeling.com/artifacts/physicalDataModel.htm> [Diakses 20 June 2016].

Anon., 2016. *Heatmap*. [Online] Available at: <http://www.investopedia.com/terms/h/heatmap.asp> [Diakses 6 April 2016].

Anon., 2016. *SDLC-Waterfall Model*. [Online] Available at: http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [Diakses 7 April 2016].

Bittner, K. & Spence, I., 2002. *Use Case Modeling*. 1st penyunt. Canada: Addison Wesley.

Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide*. 2nd penyunt. Massachusetts: Addison Wesley Professional.

BPS, 2000. *Pedoman Pencacah Sensus Penduduk 2000*, s.l.: BPS.

BPS, 2014. *Kota Malang Dalam Angka 2014 Malang City in Figures 2014*, Malang: Badan Pusat Statistik Kota Malang.

Chiew, T. K. & Salim, S. S., 2003. WEBUSE: Website Usability Evaluation Tool. *Malaysian Journal of Computer Science*, 16(1), pp. 47-57.

Deil, S., 2014. *Urbanisasi Bikin RI Kehilangan 8 Juta Petani*. [Online] Available at: <http://bisnis.liputan6.com/read/2042795/urbanisasi-bikin-ri-kehilangan-8-juta-petani> [Diakses 16 Maret 2016].

Elmasri, R. & Navathe, S. B., 2011. *Fundamentals of Database Systems*. 6th penyunt. Boston: Addison-Wesley.

Fang, Y., Shandas, V. & Cordero, E. A., 2014. *Spatial Thinking in Planing Practice*. Portland: Portland State University.

Hallahan, K., 2001. Improving public relations web sites through usability research. *Public Relations Review*, 27(2), pp. 223-239.

Hou, W. et al., 2015. Assessing quality of urban underground spaces by coupling 3D geological models : The case study of Foshan city, South China. *Computers & Geosciences*, Issue Elsevier Ltd..

Huisman, O. & By, R. A. d., 2009. *Principles of Geographic Information System*. 4th penyunt. Enschede: The International Institute fo Geo-Information Science and Earth Observation (ITC).

Kemdikbud, 2016. *Rumah Belajar*. [Online] Available at: <https://belajar.kemdikbud.go.id/SumberBelajar/tampilajar.php?ver=11&idmateri=626&mnu=Materi4&kl=7> [Diakses 7 April 2016].

Kiswondari, 2015. *Pemerintah Harus Serius Hadapi Urbanisasi*. [Online]

Available at: <http://nasional.sindonews.com/read/1024841/15/pemerintah-harus-serius-hadapi-urbanisasi-1437396950>[Diakses 16 Maret 2016].

Kroll, P. & Kruchten, P., 2003. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. 1st penyunt. Boston: Addison Wesley.

Library, W. e., 2002. *Heatmap*. [Online] Available at: <http://ebooklibrary.org/articles/Heatmap>[Diakses 6 April 2016].

Manzari, L. & Trinidad-Christensen, J., 2006. *User-centered design of a web site for library and information science students: Heuristic evaluation and usability testing*. [Online] Available at: http://faculty.mercer.edu/lewis_am/pdfs/UCD%20for%20Library.pdf[Diakses 20 Juni 2016].

Oksanen, J., Bergman, C., Sainio, J. & Westerholm, J., 2015. Methods for deriving and calibrating privacy-preserving heat maps from mobile sports tracking application data. *Journal of Transport Geography*, 48(Elsevier B.V.), pp. 135-144.

OMG,, 2011. *BPMN 2.0*. [Online] Available at: <http://www.omg.org/spec/BPMN/2.0>[Diakses 1 April 2016].

Pfarr, B., 2006. *Requirements Peer Review Checklist*. [Online] Available at: <http://www.cs.rug.nl/search/uploads/Teaching/RE2009Fall/project/NASA-Requirements%20Peer%20Review%20Checklist.doc>[Diakses 1 Juni 2016].

Pramartha, I. M. A., 2012. Implementasi Aplikasi SIG Dalam Pengolahan Data Jumlah Penduduk berbasis Web. *Jurnal Elektronik Ilmu Komputer - Universitas Udayana*, 1(Universitas Udayana), pp. 87-91.

Pressman, 2010. *Software Engineering A Practitioner's Approach*. Sydney: McGraw-Hill.

Rinder, J. M., 2012. *The Importance of Website Usability Testing*. [Online]Available at: <https://scholarsbank.uoregon.edu/xmlui/bitstream/handle/1794/12257/Rinder2012.pdf?sequence=1>[Diakses 12 Mei 2016].

Sage, A. P. & Rouse, W. B., 2009. *Handbok of System Engineering and Management*. 2nd penyunt. New Jersey: John Wiley & Sons, Inc..

Sharma, N. et al., 2010. *Database Fundamentals : Ideal for application developers and administrators*. 1st penyunt. Canada: IBM Canada.

Siemers, S. & Kutsick, A., 2004. *Mastering Object-Oriented Analysis and Design with UML 2.0*. 4 penyunt. Cupertino: IBM Rational University.

Sommerville, I., 2011. *Software Engineering Ninth Edition*. 9th penyunt. Boston: Addison-Wesley.

Speelpenning, J., Daux, P. & Gallus, J., 2001. *Data Modeling and Relational Database Design*. [Online] Available at: <http://www.darkopetrovic.com/pdf/Data-Modeling-and-Relational-Database-Design.pdf>[Diakses 1 Juni 2016].

Sugiharyanto, 2007. *Geografi dan Sosiologi 2*. Yogyakarta: Yudistira.

Susaki, J., Kajimoto, M. & Kishimoto, M., 2014. Urban density mapping of global megacities from polarimetric SAR images. *Remote Sensing of Environment*, Issue Elsevier Inc. .

Sutton, T., Dassau, O. & Sutton, M., 2002. *A Gentle Introduction to GIS*. [Online] Available at: http://docs.qgis.org/2.0/ca/docs/gentle_gis_introduction/[Diakses 8 February 2016].

Sutton, T., Dassau, O. & Sutton, M., 2016. *QGIS*. [Online] Available at: <https://www.qgis.org/id/site/about/index.html>[Diakses 12 April 2016].

Syarifah, F., 2014. *Menkes: Urbanisasi Juga Bisa Tingkatkan Risiko Penyakit*. [Online] Available at: <http://health.liputan6.com/read/2086315/menkes-urbanisasi-juga-bisa-tingkatkan-risiko-penyakit>[Diakses 16 Maret 2016].

Taufiq, R., 2013. *Sistem Informasi Manajemen : Konsep Dasar, Analisis dan Metode Pengembangan*. 1st penyunt. Yogyakarta: Graha Ilmu.

UCCS, 2007. *SRS Walkthrough*. [Online] Available at: www.uccs.edu/Documents/tboults/SRS-walkthrough.doc[Diakses 20 June 2016].

UrbanIndo, 2012. *Daerah Yang Paling Dicari di Jakarta dan Sekitarnya*. [Online] Available at: <http://blog.urbanindo.com/tag/heatmap>[Diakses 4 Oktober 2015].

Wiegers, K. E., 1999. *Wiegers' Checklist for Inspecting Requirements Specifications*. [Online] Available at: http://www.cs.toronto.edu/~sme/CSC340F/2005/assignments/inspections/reqts_checklist.pdf[Diakses 16 Mei 2016].

Wilson, E. et al., 2011. *InfoSphere Data Architect : Ideal for application developers and administrators*. 1st penyunt. Canada: IBM Canada.

LAMPIRAN A HASIL EVALUASI PENILAIAN PENGGUNA

A.1 Staf Bagian Tata Kota Bappeda

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang diberikan oleh pengguna sudah sesuai dengan harapan	✓			
Alur penggunaan system sudah sesuai dengan harapan pengguna	✓			
Pernyataan / Masukan dan Saran				

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui peran saya terhadap sistem	✓			
Sistem menyediakan label menu yang mudah dipahami untuk mendapatkan informasi tertentu	✓			
Sistem mudah digunakan dengan menggunakan tombol untuk menuju ke antarmuka yang lain	✓			
Sistem tidak membuka banyak tab atau jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali	✓			
Navigasi untuk mengatur ukuran peta mudah digunakan	✓			
Masukan dan Saran				

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Antarmuka peta sudah memberikan informasi yang jelas	✓			
Pemilihan warna pada peta sudah jelas dan mudah dipahami		✓		
Sistem memiliki antar muka yang konsisten	✓			
Masukan dan Saran	- tambahan informasi lokasi pensil/ bangunan - tambahan % informasi jumlah lantai bangunan (rumah, apartemen, kawasan perdagangan berbeda)			

Reviewer,
Malang, 18 Mei 2016

(Donny Wahyu W)

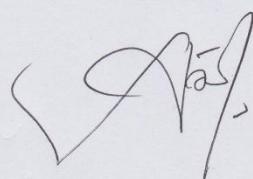
A.2 Masyarakat Umum Kota Malang

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang diberikan oleh pengguna sudah sesuai dengan harapan	✓			
Alur penggunaan system sudah sesuai dengan harapan pengguna	✓			
Pernyataan / Masukan dan Saran				

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui peran saya terhadap sistem		✓		
Sistem menyediakan label menu yang mudah dipahami untuk mendapatkan informasi tertentu		✓		
Sistem mudah digunakan dengan menggunakan tombol untuk menuju ke antarmuka yang lain	✓			
Sistem tidak membuka banyak tab atau jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali		✓		
Navigasi untuk mengatur ukuran peta mudah digunakan		✓		
Masukan dan Saran	- Peletakan menu, button (+) (-), mohon disesuaikan Google Maps (agar Atlas familiarity).			

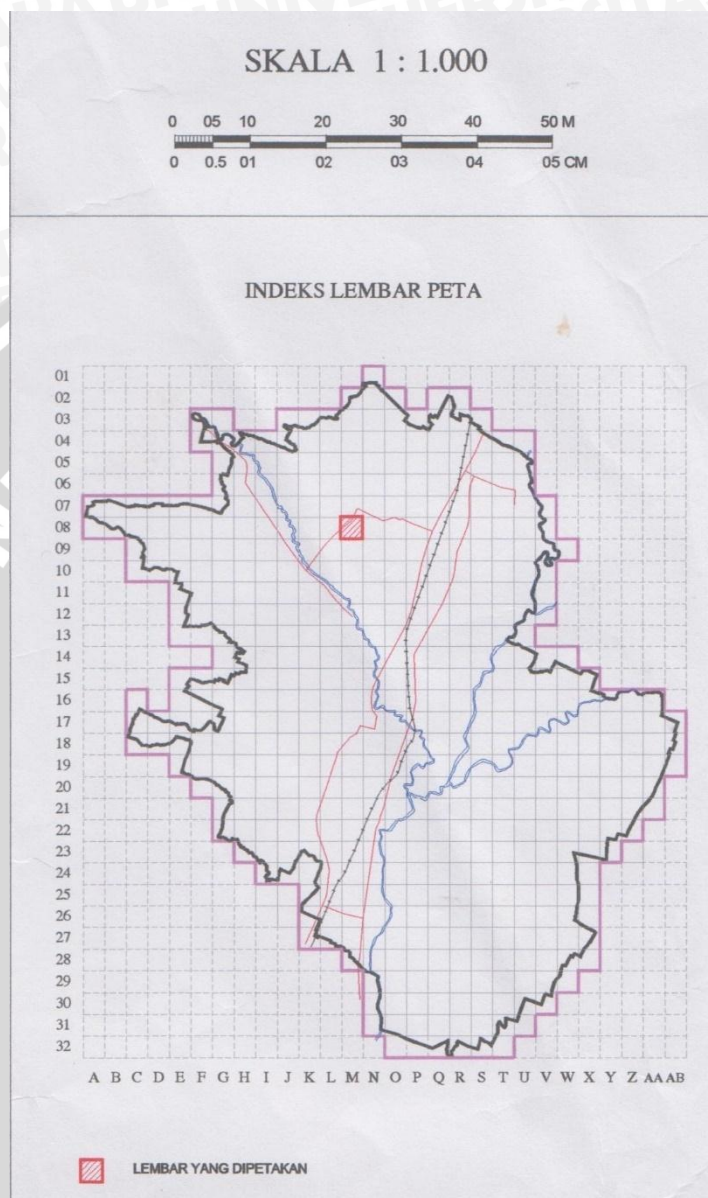
Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari		✓		
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Antarmuka peta sudah memberikan informasi yang jelas	✓			
Pemilihan warna pada peta sudah jelas dan mudah dipahami		✓		
Sistem memiliki antar muka yang konsisten	✓			
Masukan dan Saran	- pemilihan warna pada button kategori fungsi belum sesuai dengan harapan saya.			

Reviewer,
Malang, 18 Mei 2016

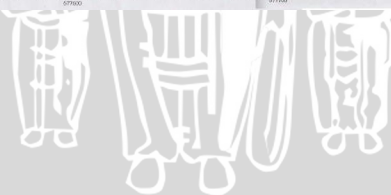
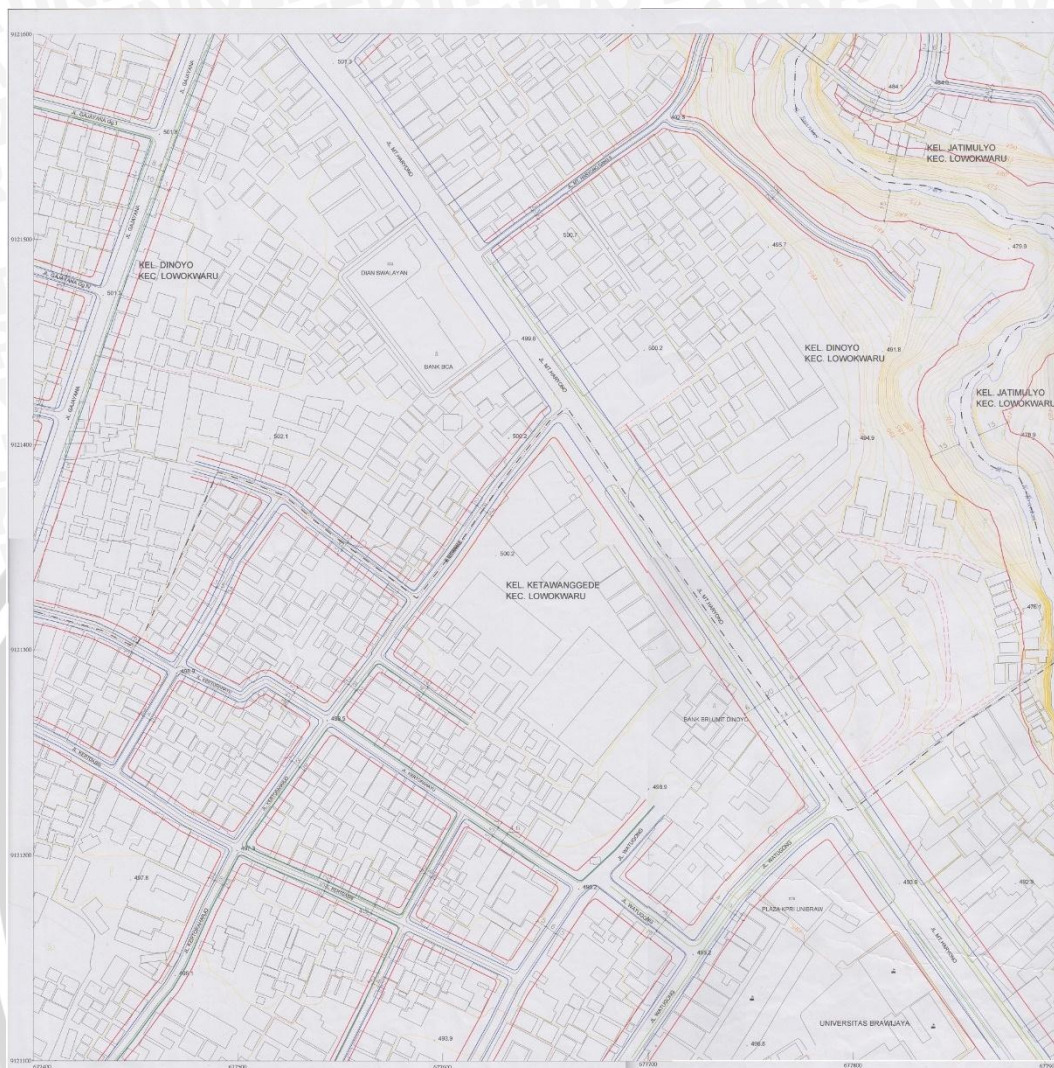

(Adh. Ais Haryansyah W.)

LAMPIRAN B PETA BANGUNAN

B.1 Aturan Indeks Lembar Peta



B.2 Indeks J-09



B.3 Indeks K-09





B.5 Indeks L-08



B.6 Indeks L-09



B.7 Indeks M07



B.8 Indeks M-08

