BAB 4 PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

4.1.1 Data dan Sumber Data

Data dan sumber merupakan bagian dari suatu analisis data spasial. Dalam proses pengumpulan data, data dapat berasal dari berbagai sumber data. Data yang di dapat digunakan sebagai *test tuple* maupun *training tuple* dalam perbandingan algoritme yang diusulkan. Seluruh sumber data dan data yang digunakan dalam perbandingan tersebut merupakan data yang sama dengan proses pengolahan yang berbeda. Berikut merupakan penjabaran mengenai data dan sumber data yang digunakan dalam mendukung penelitian ini :

4.1.1.1 Data Jaringan Jalan

Data jaringan jalan yang didapatkan berasal dari penelitian sebelumnya (Utami, 2016). Pada penelitian tersebut data jaringan jalan diambil berdasarkan hasil digitasi koordinat jaringan jalan menggunakan *google maps* API. Data jaringan jalan yang didapatkan adalah data jaringan jalan dari seluruh daerah di Jawa Timur, namun pada penelitian ini data yang digunakan adalah daerah Malang. Pada Tabel 4.1 berikut merupakan rincian dari data jaringan jalan yang digunakan:

No	Nama Field	Tipe Data	Deskripsi
1	GID	Serial (NOT NULL)	ID dari tabel road_malang
2	JOBID	Numeric	Unknown function
3	ТҮРЕ	Character Varying (10)	Unknown function
4	CATEGORY	Character Varying (50)	Kategori dari jalan
5	ISO	Character Varying (10)	Standar yang digunakan
6	NAME_0	Character Varying (50)	Nama dari negara
7	NAME_1	Character Varying (50)	Nama dari provinsi
8	ISLAND	Character Varying (50)	Nama dari pulau
9	DATASOURCE	Character Varying (30)	Sumber data yang digunakan
10	REMARKS	Character Varying (30)	Unknown function
11	NAME	Character Varying (30)	Nama dari jalan
12	LEVEL_	Integer	Unknown function
13	SHAPE_LEN	Numeric	Jarak dari <i>line</i> yang digunakan
14	GEOM	Geometry (MultiLineString)	Representasi dari kumpulan <i>line</i> yang digunakan

Tabel 4.1 Struktur Data Jaringan Jalan

Pada Tabel 4.1 telah dijabarkan mengenai *field* dari data jaringan jalan. Data jaringan jalan kemudian akan di normalisasi untuk mendapatkan data yang sesuai dengan kebutuhan penelitian (karena sebagian data dari penelitian sebelumnya memiliki nilai *null*). Data jaringan jalan akan dimasukan kedalam tabel di dalam *database* dengan nama *table* "road_malang". Tabel "road_malang" merupakan kumpulan dari informasi mengenai jaringan jalan seperti kategori jalan, nama jalan sampai dengan bentuk *logical* geometris dari kumpulan *line* yang membentuk jaringan jalan seperti dijabarkan pada Gambar 4.1. Kumpulan data jaringan jalan yang membentuk daerah Malang dan sekitarnya memiliki jumlah baris sebesar 21455 baris seperti pada Gambar 4.2.

80	🗑 🖱 🛛 Query - gisdb on postgres@localhost:5432												
	🔀 🖶 🖁 🦓 🔗 🐢 🔎 🕨 💺 🧤 🐮 🔲 🙁 🔋 🖇 🖓 🛄 📴												
SQL E	SOL Editor Graphical Ouery Builder V Scratch pad												
Previo	Previous queries												
s	SELECT gid, jobid, type, category, iso, name_0, name_1, island, datasource, remarks, name, level_, shape_len, geom FROM road_malang;												
Output p	ane												×
Data	Output Explain	Messa	iges History										
	gid jobid inte numeric	type chara	category character vary	iso cha	name_0 character v	name_1 character v	island / charae	datasource character v	remarks characte	name character varyir	level_ integer	shape_len numeric	geom geometry(MultiLineString)
1	1 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	0	0.00056360004	0105000000010000000102000
2	2 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	Θ	0.00086707324	01050000001000000102000
3	3 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	Θ	0.00043557433	010500000001000000102000
4	4 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	0	0.00061241979	010500000001000000102000
5	5 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	Θ	0.00056908473	01050000001000000102000
6	6 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	0	0.00091127209	010500000001000000102000
7	7 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Barong	0	0.00041893716	010500000001000000102000
9	9 0.0000000	0x04	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA			Nusa Kambangan	0	0.00063189800	010500000001000000102000
OK.						·			·	Unix Ln 3, Co	ol 20, Ch 1	43	21455 ro 7.9 secs

Gambar 4.1 Data Jaringan Jalan

	Uuer Quer	y - gisab	on posegre.	Schrocarin						
Ľ	3 🔲	3 B	B 🖉 🔊	n S		G3		C:		4
SQL E	ditor c	raphical	Query Builde	er			v	S	cratch pad	
Previo	us queri	es		Delete	De	lete	All			
S	ELECT C	OUNT(*) OM road_	AS "TOTAL malang;	DATA JARI	ENGAN	JALA	N"			
Dutput p	ane									
Data (Dutput	Explain	Messages	History						
	ΤΟΤΑΙ									
	bigint		ARINGAN JA							

Gambar 4.2 Jumlah Data Jaringan Jalan

4.1.1.2 Data Kecamatan

Sumber data dari data Kecamatan yang digunakan pada penelitian ini merupakan data dari Kecamatan yang ada di seluruh Indonesia, namun pada penelitian ini data dikhususkan pada kecamatan di daerah Malang dan sekitarnya. Data Kecamatan digunakan sebagai POI atau *interest point*. POI merupakan titik yang menjadi acuan dalam menentukan pencarian atau *spatial query* (dalam penelitian ini dijadikan acuan dalam membuat diagram Voronoi). Kecamatan digunakan sebagai POI karena jumlahnya yang menyebar dan merata pada seluruh wilayah sehingga mencakup ke bagian Kota maupun Desa dari suatu daerah. Data Kecamatan yang didapatkan berbentuk poligon karena representasi dari wilayah, sedangkan data yang dapat digunakan sebagai POI merupakan titik. Maka dari itu data Kecamatan yang berbentuk poligon tersebut perlu diolah sehingga memiliki *point* atau titik. Pada penelitian ini *point* atau titik tersebut adalah titik pusat (*centroid*) dari Kecamatan. Pada Tabel 4.2 berikut merupakan penjabaran dari data Kecamatan yang digunakan :

No	Nama Kolom	Tipe Data	Deskripsi
1	ID	integer	ID dari tabel Malang_Kecamatan
2	ID_KEC	integer	ID dari Kecamatan
3	Kecamatan	Character Varying (40)	Nama dari Kecamatan
4	xcoord	double	Koordinat x dari Kecamatan
5	ycoord	Double	Koordinat y dari Kecamatan
6	kode_prop	Character Varying (2)	Kode dari Provnsi
7	kode_kab	Character Varying (10)	Kode dari Kabupaten

Tabel 4.2 Struktur Data Kecamatan

Pada Tabel 4.2 telah dijabarkan *field* dari data Kecamatan. Data Kecamatan kemudian akan di normalisasi agar mendapatkan data yang sesuai dengan kebutuhan penelitian (karena sebagian data dari penelitian sebelumnya memiliki nilai *null*). Data Kecamatan yang telah dinormalisasi akan digunakan sebagai *training tuple* untuk melakukan pembagian segmentasi dari data jaringan jalan yang ada. Pembagian segmentasi dilakukan untuk membuat data jaringan jalan agar sesuai dengan penerapan algoritme VCKNN. Data Kecamatan merupakan kumpulan informasi mengenai Kecamatan seperti lokasi kecamatan, letak Kecamatan pada Kabupaten dan letak Kecamatan pada Provinsi seperti dijabarkan pada Tabel 4.2. Jumlah data kecamatan yang berada pada daerah Malang dan sekitarnya adalah sebesar 55 baris seperti pada Gambar 4.3.

8	😣 🔿 💿 🛛 Malang_Kecamatan :: Features total: 55, filtered: 55, selected: 0							
/	7 6	2	💼 🗞 블 💟	👡 🍸 🔳 🐥 💭	8 🚹 🖬 🖬	-		
	ID	ID_Kec	Kecamatan	xcoord	ycoord	kode_prop	kode_kab	F
1	2763	3507100	Poncokusumo	112.81328600000005	-8.03170800000000	35	3507	Ξ
2	2766	3507130	Bululawang	112.643317999999994	-8.091735999999999	35	3507	
3	2773	3507180	Wagir	112.5488009999999997	-8.00112800000000	35	3507	
4	2774	3507190	Pakisaji	112.59278500000006	-8.05592100000000	35	3507	
5	2775	3507200	Tajinan	112.672980999999993	-8.05234300000000	35	3507	
6	2776	3507210	Tumpang	112.749297999999996	-8.00885000000001	35	3507	
7	2777	3507220	Pakis	112.70394500000004	-7.96038300000000	35	3507	
8	2778	3507230	Jabung	112.79414400000003	-7.93179100000000	35	3507	
9	2780	3507250	Singosari	112.630703999999994	-7.85107600000000	35	3507	
10	2781	3507260	Karangploso	112.591842999999997	-7.86123600000000	35	3507	ŀ
7	Show All	Features	-					

Gambar 4.3 Data Kecamatan

4.1.1.3 Normalisasi Data Jaringan Jalan

Data jaringan jalan yang telah dijabarkan sebelumnya akan di normalisasi untuk menghilangkan data yang tidak diperlukan (*uncertainty data*) dalam penelitian ini. Hal ini dilakukan agar jumlah data yang diolah berkurang sehingga dapat meningkatkan efisiensi dari pengolahan data sesuai dengan kaidah KNN pada kajian pustaka. Pada Tabel 4.3 adalah *field* yang akan dihilangkan dari data jaringan jalan :

No	Nama <i>Field</i>	Tipe Data
1	JOBID	Numeric
2	ТҮРЕ	Character Varying (10)
3	DATASOURCE	Character Varying (30)
4	REMARKS	Character Varying (30)
5	LEVEL_	Integer

Tabel 4.3 Uncertainty Data Jaringan Jalan

Pada normalisasi data jaringan jalan, tahapan yang dilakukan adalah menghapus kolom atau *field* yang tidak digunakan seperti Gambar 4.4. Data yang dihapus tersebut tidak akan mempengaruhi kualitas analisis data. Dengan menghapus *fields* tersebut tentunya akan mengurangi jumlah *tuple* yang ada. Maka dari itu dengan adanya normalisasi terjadi perubahan dalam struktur *table* seperti pada Gambar 4.5.

😣 🖱 💿 Query - gisdb on postgres@localhost:5432 *	
- 🖻 🖶 🌡 🗈 🗳 🖉 🧑 🖉 🔎 🕨 陆 🐂 🔚 🗟 💧 ?	田ost:5432 🗘
SQL Editor Graphical Query Builder	×
Previous queries	
ALTER TABLE road_malang DROP COLUMN jobid; ALTER TABLE road_malang DROP COLUMN type; ALTER TABLE road_malang DROP COLUMN datasource; ALTER TABLE road_malang DROP COLUMN remarks; ALTER TABLE road_malang DROP COLUMN level_;	
Output pane	×
Data Output Explain Messages History	∇
Query returned successfully with no result in 12 msec.	
OK. Unix Ln 5, Col 44, Ch 222	12 ms

Gambar 4.4 Drop Column pada Data Jaringan Jalan

	Ø ⊜ @ Query - gisdb on postgres@localhost:5432 *								
	🗈 🚰 🔚 🖁 🖓 🕼 🍋 🖉 🔊 🖉 🕨 🕨 🐘 🍓 🗉 🕏 🚖 🂡 🛛 🔤 📾 📾 🖓 👘 🖓 👘 🖓 👘 🖓 👘 🖓 👘 🖓 👘 🖓								
SOL E	litor	Craphical Or	LOCK D	uildes					Scratch pad
JQLL	JICOI	Graphical Qu	јегу в	ullder					
Previo	us qu	Jeries					▼ De	elete Delete A	at]]
SI	LEC	T gid, catego	ry, i	iso, name G	, name 1, i	sland,	name, shape len, q	eom	
	FRO	M road_malang	0	-					
Output pa	ane								X
Data	Juto		10000	and Ulator					~
	Juch	Explain N	ressa	ges Histor	У			I	
	gid	category	iso	name_0	name_1	island	name	shape_len	geom
	inte	character var	char	character	character va	charac	character varying(50	numeric	geometry(MultiLineString)
1	1	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00056360004	01050000001000000102000000200000
2	2	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00086707324	01050000001000000102000000200000
3	3	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00043557433	01050000001000000102000000200000
4	4	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00061241979	01050000001000000102000000200000
5	5	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00056908473	01050000001000000102000000200000
6	6	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00091127209	01050000001000000102000000300000
7	7	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Barong	0.00041893716	01050000001000000102000000200000
8	8	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Abdulrachman Saleh	0.00069528366	01050000001000000102000000200000
9	9	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00063189800	01050000001000000102000000300000
10	10	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Barong	0.00071030275	01050000001000000102000000200000
11	11	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Barong	0.00043424647	01050000001000000102000000200000
12	12	Jalan Arteri	IDN	INDONESIA	JAWA TIMUR	JAWA	Nusa Kambangan	0.00030016625	01050000001000000102000000200000
pgScrip	t cor	npleted.					Unix Ln 1	, Col 69, Ch 69	21455 ro 12.5 s

Gambar 4.5 Hasil Normalisasi Data Jaringan Jalan

4.1.1.4 Normalisasi Data Kecamatan

Data Kecamatan yang telah dijabarkan sebelumnya akan di normalisasi untuk menghilangkan data yang tidak diperlukan (*uncertainty data*) dalam penelitian ini. Hal ini dilakukan agar jumlah data yang diolah berkurang sehingga dapat meningkatkan efisiensi dari pengolahan data. Berikut adalah *field* yang akan dihilangkan dari data Kecamatan :

No	Nama Kolom	Tipe Data
1	ID	integer
2	kode_prop	Character Varying (2)
3	kode_kab	Character Varying (10)

Tabel 4.4 Uncertainty Data Kecamatan

Pada normalisasi data Kecamatan, tahapan yang dilakukan adalah menghapus *field* tersebut. Data yang dihapus tersebut tidak akan mempengaruhi kualitas analisis data. Dengan menghapus *fields* tersebut tentunya akan mengurangi jumlah *tuple* yang ada. Maka dari itu dengan adanya normalisasi terjadi perubahan dalam struktur *table* seperti pada Gambar 4.6.

8	🔍 🔲 Malang	_Kecamatan ::	Features tota	al: 55, filtered	
/	2 🗟 🔁	🛱 🛱 🎦	= 🛯 🖥 🍸	🖺 🏘 🎦	;
	ID_Kec	Kecamatan	xcoord	ycoord	F
1	3505050	Binangun	112.33560	-8.2131770	
2	3506120	Kandangan	112.31833	-7.7504150	
3	3507010	Donomulyo	112.42528	-8.3180549	
4	3507020	Kalipare	112.43201	-8.2129359	
5	3507030	Pagak	112.52114	-8.2420680	
6	3507040	Bantur	112.54384	-8.3019320	
7	3507050	Gedangan	112.62111	-8.3285410	
8	3507060	Sumberma	112.73431	-8.3300310	
9	3507070	Dampit	112.77217	-8.2438230	
10	3507080	Tirtoyudo	112.82668	-8.2506670	
11	3507090	Ampelgading	112.90164	-8.2273790	
Î	Show All Featu	ires 🔻)(

Gambar 4.6 Hasil Normalisasi Data Kecamatan

4.2 Lingkungan Operasi

Dalam melakukan perbandingan algoritme VCKNN dan Dijkstra dibutuhkan lingkungan operasi yang memiliki spesifikasi yang sama agar didapatkan hasil perbandingan yang akurat. Lingkungan operasi yang akan digunakan pada penelitan ini dibagi menjadi 2 yaitu lingkungan perangkat keras dan lingkungan perangkat lunak yang dijabarkan sebagai berikut :

4.2.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan untuk mendukung pengujian algoritme yang diusulkan pada penelitian ini dijabarkan pada Tabel 4.5.:

No	Nama Perangkat	Spesifikasi
1	Processor	Intel® Core™ i7-4720H CPU @ 2.60GHz (8CPUs), ~2.6GHz
2	Memory	12288MB RAM
3	Harrdisk	1 TB
4	VGA	NVIDIA [®] GEFORCE [®] GTX [™] 950M
5	Monitor	15.6 inci
6.	Tambahan	- Keyboard
		- Mouse

Tabel 4.5 Spesifikasi Perangkat Keras

4.2.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk mendukung pengujian algoritme pada penelitian ini dijabarkan pada Tabel 4.6.:

No	Jenis Aplikasi	Nama Aplikasi
1	Sistem Operasi	Windows 10 Home Single Language
2	Virtual Machine	VMware [®] Workstation 12 Pro
3	Analisis Spatial	Quantum GIS 2.18.7 Las Palmas de G.C.
4	Database	PostgreSQL 9.6, MySQL Server 5.5
5	Database Tools	pgAdmin III
6	Server	LAMP Server
7	Development Tools	PHP 7.0

Tabel 4.6 Spesifikasi Perangkat Lunak

Pada penjelasan spesifikasi di atas menggunakan bantuan perangkat lunak untuk virtualisasi dengan spesifikasi pada *virtual machine* dijabarkan pada Tabel 4.7.:

Tabel 4.7 Spesifikasi Virtual Machine

No	Nama	Spesifikasi
1	Memory	8 GB
2	Processors	4
3	Harddisk (SCSI)	100 GB
4	CD/DVD (SATA)	Auto detect

5	Network Adapter	NAT
6	Sistem Operasi	Linux Ubuntu 16.04 LTS

Penggunaan virtual machine dibutuhkan agar mendapatkan hasil pengujian yang maksimal dan tidak terganggu oleh aplikasi yang berjalan di background dan dengan menggunakan virtual machine spesifikasi dari virtual machine dapat disesuaikan dengan kebutuhan.

4.3 Instalasi dan Konfigurasi Quantum GIS 2.18.7

Release	Version	Description	Repository
Latest Release (LTR candidate)	2.18.x Las Palmas	Release for Debian and Ubuntu	http://qgis.org/debian ⊿ *
Release wit ubuntugis dependenci	Release with ubuntugis dependencies	http://qgis.org/ubuntugis a	
		Nightly build of upcoming point release for Debian and Ubuntu	http://qgis.org/debian-nightly- release ⅆ
		Nightly build of upcoming point release with ubuntugis dependencies	http://qgis.org/ubuntugis-nightly- release ⅆ

Gambar 4.7 Versi QGIS beserta Repository

Pada bagian ini akan dijelaskan proses instalasi dan konfigurasi dari *Quantum GIS* (QGIS). Pada penelitan ini, QGIS berjalan di sistem operasi Linux Ubuntu. QGIS yang digunakan adalah versi 2.18.7 Las Palmas (dapat dilihat pada Gambar 4.7.) yang merupakan versi *Lastest Release* (LTR *candidate*) yang diperuntukan untuk Linux versi Debian dan Ubuntu. Pada proses instalasi menggunakan *repository* dari <u>http://qgis.org/debian</u> dengan dukungan paket *repository* dari <u>https://launchpad.net/~ubuntugis</u> yang memiliki versi terbaru. Untuk kedua paket QGIS tersebut, *repository* yang digunakan dapat ditambahkan pada *file /etc/apt/sources.list* yang diakses melalui terminal.

Pada tahapan awal melakukan instalasi QGIS adalah menambahkan repositories kedalam file /etc/apt/source.list yang dibutuhkan QGIS seperti pada Gambar 4.8. Hal ini berfungsi untuk mendapatkan paket yang ada di dalam repository tersebut. Pada Gambar 4.8 repository yang digunakan adalah xenial menyesuaikan dengan versi Ubuntu yang ada yaitu 16.04 LTS. Kemudian setelah menambahkan repository langkah berikutnya adalah melakukan update apt sehingga repository yang ditambahkan dapat digunakan.

😣 🖱 💷 oddyaulia@ubuntu: ~			
GNU nano 2.5.3	File: /etc/apt/sources.list	Modified	
#Quantum GIS 2.18 deb http://qgis.ou deb-src http://qg deb http://ppa.la	<pre>#Quantum GIS 2.18.7 deb http://qgis.org/ubuntugis xenial main deb-src http://qgis.org/ubuntugis xenial main deb http://pgia.launchpad.net/ubuntugis/ubuntugis-unstable/ubuntu xenial main</pre>		

Gambar 4.8 Penambahan Repositories QGIS

Pada tahap berikutnya dapat dilakukan instalasi QGIS seperti yang ada pada Gambar 4.9. Pada tahapan tersebut akan dilakukan instalasi paket QGIS berserta dengan plugin dukungan yang berguna dalam mengoperasikan QGIS.

😣 🖱 💷 oddyaulia@ubuntu: ~
oddyaulia@ubuntu:~\$ sudo apt-get install qgis python-qgis qgis-plugin-grass [sudo] password for oddyaulia: Paadiaa packaga lists _ Done
Reading package (1515 Done Ruilding dependency tree
Deading state information Done
The following packages were automatically installed and are no longer required:
libogis-analysis2.14.11 libogis-analysis2.18.7 libogis-app2.14.11
libagis-app2.18.7 libagis-core2.14.11 libagis-core2.18.7 libagis-aui2.14.11
libggis-gui2.18.7 libggis-networkanalysis2.14.11
libqgis-networkanalysis2.18.7 libqgis-server2.14.11 libqgis-server2.18.7
libqgisgrass7-2.14.11 libqgisgrass7-2.18.7 libqgispython2.14.11
libqgispython2.18.7 libqwt6abi1 snap-confine
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libqgis-analysis2.18.9 libqgis-app2.18.9 libqgis-core2.18.9
libqgis-customwidgets libqgis-gui2.18.9 libqgis-networkanalysis2.18.9
libqgis-server2.18.9 libqgisgrass7-2.18.9 libqgispython2.18.9
python-ggis-common ggis-common ggis-plugin-globe ggis-plugin-globe-common
ggis-plugin-grass-common ggis-provider-grass ggis-providers
dgis-providers-common
anchahal
gpsuber
librais-analysis 18 9 librais-anal 18 9 librais-core 18 9
couges analyses couges appenders couges concernors

Gambar 4.9 Proses Instalasi Paket QGIS

Pada tahapan berikutnya adalah melakukan ekspor dan menambahkan *key* 073D307A618E5811 sebagai *keyserver* untuk mengakses *public key* dari *keyring*. Proses yang dilakukan pada Gambar 4.10 berfungsi untuk menangani *error* yang diakibatkan belum terdaftarnya akses ada QGIS.



Gambar 4.10 Ekspor dan Penambahan Keyserver QGIS

Setelah *public key* terdaftar, maka dapat dilakukan eksekusi untuk mendapat akses ke <u>keyserver.ubuntu.com</u> seperti pada Gambar 4.11.



Gambar 4.11 Eksekusi Keyserver QGIS

Pada tahapan terakhir dalam instalasi QGIS adalah memastikan bahwa *key* yang telah dimasukan telah berhasil tersambung dengan *server* seperti pada Gambar 4.12 dan melakukan pengecekan terhadap *key fingerprint* seperti pada Gambar 4.13.



Gambar 4.12 Pengecekan Terhadap Key QGIS



Gambar 4.13 Pengecekan Fingerprint QGIS

Setelah seluruh tahapan instalasi dan konfigurasi berhasil, maka QGIS akan terinstal pada Ubuntu 16.04 LTS serta dapat dijalankan dengan mengeksekusi QGIS Desktop seperti pada Gambar 4.14.

QGIS 2.18.10	👣 🐠 3:06 PM 🔅
R. 이 이 및 및 한 소 등 등 등 등 등 등 🗧 🚽 🛯	a 🔒 🔢 🔁 : 🔍 🍭 - 🔣 - 🏭 - 🖳 📰 🗮 \Sigma 🛲 - 🎾 🕮 - : 📓
Signature Street	
Favourites	
MSSQL	
SoatiaLite	
ArcGisFeatureServer	
Arccismapserver Ows	
Tile Server (XYZ)	
WFS WFS	
Ma 🖉 WMS	
a. V	
1000	
22	
Layers Panel 🛛 🕅 🗷	
8	Coordinate 12.633185,-7.980843 🗞 Scale 1:3,773 🔹 🖨 Magnifier 100% 🛟 Rotation 0.0 🛟 🖉 Render 🗅 EPSG:4326 🔍

Gambar 4.14 Workspace QGIS

4.4 Instalasi dan Konfigurasi PostgreSQL dan pgAdmin III

4.4.1 PostgreSQL

Pada bagian ini akan menjelaskan proses instalasi serta konfigurasi dari PostgreSQL. PostgreSQL merupakan salah satu DBMS yang memiliki paket ekstensi pendukung untuk melakukan *spatial query*. Ekstensi pendukung tersebut dinamakan PostGIS. Dengan adanya PostGIS memungkinkan pemindahan *file* dengan ekstensi *.shp* masuk kedalam *database*, memungkinkan *database* menyimpan data dalam bentuk geometri dan kelebihan lain dari PostGIS adalah mendukung integrasi dengan QGIS.

Pada tahapan awal dari instalasi PostgreSQL adalah melakukan verifikasi terkait versi dari Ubuntu yang berjalan seperti pada Gambar 4.15. Pada pengecekan tersebut diketahui versi dari Ubuntu yang dimiliki adalah versi *xenial* (16.04.2 LTS).

😣 🖨 🗊 oddyauli	a@ubuntu: ~	
oddyaulia@ubunt	u:~\$ sudo lsb_release -a	1
No LSB modules	are available.	
Distributor ID:	Ubuntu	
Description:	Ubuntu 16.04.2 LTS	
Release:	16.04	
Codename:	xenial	

Gambar 4.15 Versi Linux Ubuntu

Pada langkah berikutnya adalah menambahkan *repositories* kedalam *file* /*etc/apt/source.list* yang dibutuhkan pada tahap instalasi PostgreSQL seperti Gambar 4.16. Hal ini berfungsi untuk menggunakan paket yang ada didalam *repository* tersebut.



Gambar 4.16 Penambahan Repository PostgreSQL

Pada tahapan ini menambahkan *public key* untuk mengakses *repository* dari PostgreSQL pada alamat *http://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc* seperti pada Gambar 4.17.

oddyaulia@ubuntu:~\$ wgetquiet CC4CF8.asc sudo apt-key add - OK	-O - http://apt.postgresql.org/pub/repos/apt/AC

Gambar 4.17 Penambahan Key PostgreSQL

Pada tahapan berikutnya adalah proses instalasi dari PostgreSQL beserta ekstensi PostGIS. PostGIS dilengkapi dengan *tools* seperti *shp2pgsql* dan *raster2pgsql*. Berikut merupakan langkah dalam instalasi yang dilakukan seperti pada Gambar 4.18 – Gambar 4.20.

oddyaulia@ubuntu:~\$ sudo apt-get install postgresgl-9.6			
reading package (isis, bone			
Building dependency tree			
Reading state information Done			
postgresgl-9.6 is already the newest version (9.6.3-1.pgdg16.04+1).			

Gambar 4.18 Proses Instalasi PostgreSQL 9.6



Gambar 4.19 Proses Instalasi Paket PostGIS 2.3



Gambar 4.20 Proses Instalasi PostGIS

4.4.2 pgAdmin III

Pada bagian ini dilakukan instalasi pgAdmin III yang merupakan tools pendukung yang menyediakan GUI untuk mengelola database menggunakan DBMS PostgreSQL. Berikut merupakan proses instalasi dari pgAdmin III seperti pada Gambar 4.21.

😣 🖻 💷 oddyaulia@ubuntu: ~
oddyaulia@ubuntu:~\$ sudo apt-get install pgadmin3
[sudo] password for oddyaulia:
Reading package lists Done
Building dependency tree
Reading state information Done
pgadmin3 is already the newest version (1.22.2-1.pgdg16.04+1).
The following packages were automatically installed and are no longer required:
libqgis-analysis2.14.11 libqgis-app2.14.11 libqgis-core2.14.11
libqgis-gui2.14.11 libqgis-networkanalysis2.14.11 libqgis-server2.14.11
libqgisgrass7-2.14.11 libqgispython2.14.11 libqwt6abi1 snap-confine
Use 'sudo apt autoremove' to remove them.
0 upgraded. 0 newly installed. 0 to remove and 26 not upgraded.

Gambar 4.21 Proses Instalasi pgAdmin III

4.5 Pengolahan Data

4.5.1 Pengolahan Data pada QGIS

Pada bagian ini akan membahas mengenai proses pengolahan data untuk digunakan pada algoritme yang diusulkan. Pengolahan data dilakukan pada aplikasi QGIS.

- Pada tahap pertama pengolahan data adalah memasukkan data yang akan digunakan yaitu data jaringan jalan dan data Kecamatan atau data lainnya yang digunakan dengan cara menekan Add Vector Layer (berfungsi untuk menggunakan layer data bertipe vektor).
- 2. Kemudian pilih *source type* dari *dataset* yang akan dimasukan ke dalam layer, *dataset* yang digunakan pada penelitian ini berbentuk *file*.
- Setelah itu, masukan source yang merupakan dataset yang akan digunakan. Masukan direktori dari *file* tersebut atau dapat dicari melalui fitur browse untuk mendapatkan direktori *file* yang dituju.
- 4. Setelah selesai file dapat di open.

Pada Gambar 4.22 telah dijabarkan proses memasukkan *dataset* ke dalam *layer*. Pada Gambar 4.23 merupakan hasil dari *dataset* yang telah dimasukan ke dalam *layer*. *Dataset* tersebut merupakan hasil dari *clipping* yang sebelumnya telah dilakukan dengan menggunakan data jaringan jalan Jawa Timur yang di *clip* dengan data wilayah Malang kemudian diambil bagian Malang dan sekitarnya. Pada Gambar 4.23 *line* atau garis yang berwarna merah merupakan kumpulan dari data jaringan jalan sedangkan poligon yang terbentuk dengan warna hijau merupakan wilayah Kecamatan di Malang.



Gambar 4.22 Proses Memasukkan Dataset ke dalam Layer QGIS



Gambar 4.23 Penampakan dari *Dataset* Jaringan Jalan dan *Dataset* Kecamatan Malang di QGIS

- 5. Tahap berikutnya adalah membuka menu *Vector* (pada menu *vector* terdapat fitur untuk melakukan pengolahan data *vector*).
- Kemudian memilih menu *Geometry Tools* yang merupakan *tools* untuk membantu membuat bentuk geometri berdasarkan *dataset*. (Gambar 4.24)
- 7. Pilih menu *Polygon Centroids,* menu ini dapat membuat data Kecamatan yang berbentuk poligon akan ditemukan titik pusat dari setiap wilayah. (Gambar 4.24)



Gambar 4.24 Penentuan Centroid dari Data Kecamatan

- 8. Memilih *Input Layer* yang akan dijadikan data acuan dalam menentukan *centroids*. (Gambar 4.25)
- Setelah memilih *layer* Kecamatan, kemudian dapat dilakukan eksekusi algoritme pencarian titik pusat dari *dataset* Kecamatan sehingga akan didapatkan *centroids* berdasarkan *datasets*. (Gambar 4.25 - Gambar 4.26)



Gambar 4.25 Menjalankan Algoritme Polygon Centroids



Gambar 4.26 Penampakan Centroids yang Terbentuk di Wilayah Kecamatan

Pada Gambar 4.27. merupakan hasil dari penerapan algoritme *polygon centroids*. *Centroids* inilah yang disebut dengan POI atau *interest point* yang berfungsi untuk menjadi acuan dalam membuat diagram Voronoi.

- 10. Tahap berikutnya adalah membuka menu *Vector* kembali (pada tahapan ini dilakukan untuk membuat diagram Voronoi).
- 11. Memilih menu *Geometry Tools* untuk membantu membuat bentuk geometri.
- 12. Memilih menu *Voronoi Polygons,* yang berfungsi untuk membuat diagram Voronoi berdasarkan POI yang telah ditentukan.
- Pada tahap ini, memilih *Input Layer* yang akan dijadikan POI atau acuan terhadap pembentuka poligon Voronoi, dimana yang menjadi POI adalah *centroids* dari Kecamatan yang sebelumnya dilakukan pengolahan data. (Gambar 4.28)
- 14. Kemudian akan mengeksekusi algoritme pembuatan *Voronoi Polygon* berdasarkan POI yang ada dan akan menghasilkan diagram Voronoi. (Gambar 4.29)

Pada Gambar 4.29 merupakan hasil dari algoritme Voronoi Polygons yaitu membentuk diagram Voronoi, namun ada beberapa wilayah yang tidak tertutup oleh diagram Voronoi dikarenakan wilayah tersebut berada diluar dari poligon. Hal tersebut dapat diatasi dengan beberapa cara yaitu dengna membuat titik bantuan pada bagian terluar dari diagram Voronoi sehingga wilayah yang tidak tercakup dapat masuk ke dalam diagram Voronoi atau melakukan eliminasi data terhadap data yang berada diluar diagram Voronoi. Pada Gambar 4.30 digunakan titik bantuan yang berada diluar *centorids* yang ada dan menjadikan titik bantuan

tersebut sebagai *centroids* dari poligon Voronoi yang menutupi bagian yang sebelumnya tidak termasuk diagram Voronoi.



Gambar 4.27 Penentuan Voronoi Polygon Berdasarkan POI

😣 🗈 Voronoi polygons	
Parameters Log 13	Voronoi polygons
Input layer	This algorithm takes a points layer
POI [EPSG:4326]	and generates a polygon layer containing the voronoi polygons
Buffer region	corresponding to those input
0.000000	points.
Voronoi polygons	
[Create temporary layer]	
✓ Open output file after running algorithm	14
0%	
	Close Run

Gambar 4.28 Menjalankan Algoritme Voronoi Polygons



Gambar 4.29 Penampakan dari Diagram Voronoi Berdasarkan Centroids



Gambar 4.30 Penampakan dari Diagram Voronoi dengan Titik Bantuan

- 15. Tahap selanjutnya adalah membuka menu *Vector* kembali (pada tahapan ini dilakukan untuk memotong data jaringan jalan dengan diagram Voronoi). (Gambar 4.31)
- 16. Memilih menu *Geometry Tools* untuk membantu membuat bentuk geometri. (Gambar 4.31)

17. Pilih menu *Intersection,* menu ini berfungsi untuk memotong *layer* jaringan jalan dengan *layer* diagram Voronoi, sehingga dapat membentuk data jaringan jalan hasil perpotongan berdasarkan diagram Voronoi. (Gambar 4.31)



Gambar 4.31 Proses Intersection Data Jaringan Jalan

- 18. Memilih *Input Layer* yang akan dijadikan acuan dalam melakukan *intersection* yaitu *layer* jaringan jalan. (Gambar 4.32)
- 19. Memilih *layer* yang menjadi pemotong dari *input layer* yaitu diagram Voronoi. (Gambar 4.32)
- 20. Kemudian eksekusi intersection. (Gambar 4.32)

Pada Gambar 4.33 merupakan hasil dari *Intersection, layer* jaringan jalan dengan *layer* diagram Voronoi yang berfungsi mendapatkan data jaringan jalan yang memiliki struktur perpotongan sesuai dengan diagram Voronoi. *Layer* yang telah mengalami perpotongan adalah *layer* berwarna merah muda. Hasil perpotongan ini dapat dilihat dari jumlah data yang akan bertambah karena ada penambahan data dari *split nodes* hasil perpotongan dengan diagram Voronoi.

🔋 🗉 Intersection		
Parameters Log	Run as batch process	Internetion
Input layer		This 18 xtracts the
road_Malang [EPSG:4326]	:)	over apping portions of features in the input and intersect layers.
Intersect layer		Features in the Intersection layer
Voronoi polygons [EPSG:4326]	:	overlapping features from both the
Ignore NULL geometries [optional]		hout a share t layers.
Intersection		Attrib 19 modified
[Create temporary layer]		
Open output file after running algorithm		
		20
Î	0%	
		<u>Close</u> Run

Gambar 4.32 Menjalankan Intersection Data Jaringan Jalan dan Diagram Voronoi



Gambar 4.33 Penampakan Hasil dari Intersection Data Jaringan Jalan

Berikut adalah data yang dihasilkan dari *Geo-processing* yang telah dilakukan seperti pada Gambar 4.34. Data yang dihasilkan sebanyak 22161 baris bertambah dibandingkan dengan data diawal.

800	Intersectio	on :: Features	total: 22161, f	iltered: 22161	I, selected: 0			
/ #	6 2 6	ê 📒	🔊 🔩 🍸 🗷	🏘 🎾 🖻				
LEIJE	JobID	Туре	Category	ISO	Name_0	Name_1	Island	Data
22153	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22154	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22155	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22156	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22157	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22158	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22159	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22160	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	
22161	0.00000	0x04	Jalan Lokal	IDN	INDONESIA	JAWA TIMUR	JAWA	(III) ►
G T Sho	w All Features)					
4 5110		<u>.</u>						

Gambar 4.34 Data Hasil Geo-processing dengan Diagram Voronoi

Proses pengolahan data yang telah dilakukan sebelumnya, dapat diterapkan kembali apabila memiliki POI yang berbeda. Seluruh langkah-langkah yang diperlukan akan sama hanya terjadi perbedaan pada *input layer* yang dimasukkan.

4.5.2 Pengolahan Data pada PostgreSQL

Pada bagian ini membahas mengenai proses pengolahan data dari sisi *database. Database* yang digunakan adalah PostgreSQL dengan ekstensi PostGIS.

- Pada tahap awal, menggunakan aplikasi shp2psql-gui untuk melakukan impor file shapefile kedalam database. Pada aplikasi shp2psql-gui terlebih dahulu disambungkan ke PostGIS. (Gambar 4.35)
- Masukkan username, password, server host dan database yang digunakan. Server yang digunakan adalah localhost dengan port 5432. Database yang digunakan adalah gisdb. (Gambar 4.35)
- Setelah konfigurasi koneksi telah diisi sesuai dengan database yang digunakan akan dilakukan test connection untuk memastikan sambungan telah terjadi, apabila informasi sesuai maka koneksi akan berhasil. (Gambar 4.35)
- 4. Setelah konfigurasi berhasil, dapat dilakukan impor *file shapefile* ke *database* yang digunakan dengan cara masuk ke dalam halaman impor.
- Kemudian pilih add file, pilih file shapefile yang akan di import pada penelitian ini file shapefile yang dimasukan ke dalam database adalah hasil pengolahan data yang sebelumnya diolah pada QGIS yaitu data road_Malang.shp dan Voronoi_Malang.shp.
- File yang telah dipilih akan masuk kedalam import list dengan keterangan schema yang digunakan, table yang dibuat, jenis kolom, SRID dan mode pembuatan.
- Setelah *file* sudah dipilih, maka data dapat di import ke *database* dan akan di *load* data yang ada. Perhatikan hasil *load* yang ada di *log window* untuk memastikan tidak ada *error* atau *corrupt* pada data yang di *load*. (Gambar 4.36)

	View connection details.	
Import	Export	
Impor	t List	
Shar	8 🕒 🗊 PostGIS connection	
	PostGIS Connection	
	Username: postgres	
	Password: ••••••	
	Server Host: localhost 54	432
	Database:	
Ор		21
Log Wi	ОК	
3		

Gambar 4.35 Membuat Koneksi ke PostGIS

stGIS Connection							
	View conr	nection d	etails				
4							
port Export							
mport List							
Shapefile			Table	Geo Column	SRID	Mode	Rm
/home/oddyaulia/Desktop/Voronoi/Voronoi_Malang.shp			voronoi_malang	geom	0	Create	
/home/oddyaulia/Desktop/ja	atim/road_Malang.shp	public	road_malang	geom	0	Create	
	6				*	/	5
Options	6 A	Add File	About		*	Cancel	5
Options	6	Add File	About		*	Cancel	5
Options g Window 7	6	Add File	About			Cancel	5
Options g Window 7	6 Import	Add File	About			Cancel	5
Options g Window 7	6	add File	About			Cancel	5
Options g Window 7	6	add File	About			Cancel	5
Options g Window 7	6	dd File	About			Cancel	5

Gambar 4.36 Impor Shapefile ke Database

Setelah melakukan impor data ke dalam *database*, maka proses selanjutnya adalah melakukan integrasi *database* dengan *tools* pendukung lainnya seperti pgAdminIII untuk memberikan tampilan dalam mengelola *database* dan QGIS

untuk mengelola data spasial. Pada tahap ini proses integrasi dilakukan pada pgAdmin III terlebih dahulu agar diketahui struktur tabel.

- 8. Setelah membuka pgAdmin III, pilih *add a connection to a server* untuk membuat sambungan antara *database* dan pgAdmin III. (Gambar 4.37)
- Kemudian masukan new server registration, masukkan nama alias dari database, masukan host dari database dalam penelitian ini dijalankan di localhost, port yang digunakan 5432, pilih user yang dapat melakukan maintain dari database kemudian masukan username dan password. (Gambar 4.37)
- 10. Setelah melakukan pengisian registrasi *database*, maka akan keluar notifikasi mengenai koneksi antara *database* dan pgAdmin III.



Gambar 4.37 Membuat Koneksi ke pgAdmin III

Setelah koneksi dibuat maka dapat dilakukan pengecekan terhadap database tersebut, apabila sudah berhasil melakukan impor maka dapat dilihat pada schema yang dimasukan sebelumnya dan masuk ke dalam sub folder table. Pada sub folder table terdapat tabel road_malang dan voronoi_malang seperti pada Gambar 4.38.

Pada Gambar 4.39 terdapat informasi mengenai tabel *road_malang* yang telah dibuat mulai dari nama dari *table* tersebut, struktur *table* tersebut, *owner* yang dapat mengelola *table* tersebut, *index* yang digunakan pada *table* tersebut, *primary key* pada *table* tersebut, jumlah *rows* yang ada dan lain sebagainya.



Gambar 4.38 Tabel di dalam Database GISDB



Gambar 4.39 Struktur Data dari road_malang

Pada Gambar 4.40 terdapat informasi mengenai tabel voronoi_malang yang telah dibuat mulai dari nama dari table tersebut, struktur table tersebut, owner yang dapat mengelola table tersebut, index yang digunakan pada table tersebut, primary key pada table tersebut, jumlah rows yang ada dan lain sebagainya.



Gambar 4.40 Struktur Data dari voronoi_malang

Setelah proses integrasi dengan pgAdmin III selesai maka pengelolaan database dapat dilakukan melalui GUI pgAdmin III dan dengan adanya pgAdmin III memudahkan dalam mengetahui informasi dan mengelola perubahan pada database.

Setelah selesai konfigurasi integrasi dengan pgAdmin III dapat dilakukan integrasi dengan QGIS sehingga memudahkan dalam mengelola data spasial secara visual. Dengan integrasi ini dapat memungkinkan *maintain* perubahan apabila terjadi pengolahan pada QGIS. Berikut merupakan tahapan dalam membuat koneksi dengan QGIS seperti pada Gambar 4.41.

- 11. Pilih *add PostGIS layer*, dengan memlilih menu ini akan dibuatkan koneksi dengan database PostGIS
- 12. Mengisi registrasi mengenai koneksi dengan PostGIS dengan mengisikan nama alias untuk *database*, host yang digunakan, nama dari *database* tersebut dan *username* dan *password*
- 13. Untuk mengetahui informasi registrasi telah sukses dapat dilakukan *test connection* untuk mengetahui informasi registrasi telah benar.



Gambar 4.41 Membuat Koneksi ke QGIS

Setelah koneksi telah dibuat, maka akan diminta *username* dan *password* untuk membuat sambungan dengan *enter credentials* seperti pada Gambar 4.42.



Gambar 4.42 Menyambungkan PostGIS ke QGIS

Maka akan ditampilkan mengenai *database* yang ada di dalam PostGIS dan tersambunga dengan QGIS seperti pada Gambar 4.43.



Gambar 4.43 Penampakan Database di QGIS