

Pelatihan Analis Sistem Informasi Geografis

Dr. Firman Hadi

Email: firman@indotechsa.com
WhatsApp: +62-812-2000-1994
Youtube: www.youtube.com/jalmiburung

21 Oktober - 11 November 2019
Program Studi Geomatika,
Politeknik Batam, Batam

Daftar Isi

Pengantar	5
I Basis Data	7
1 Basis Data	9
1.1 Perancangan Basis Data	9
1.2 Pembuatan Basis Data dengan PostGIS	21
2 Studi Kasus : Pembangunan Basis Data untuk Analisis Wilayah Layanan	43
2.1 Praktek Perancangan Basis Data	44
3 Basis Data Lanjut	45
3.1 Non-spatial SQL Queries	45
3.2 Spatial SQL Queries	46
3.3 Backup dan Restore	46
II Sistem Informasi Geografis	47
4 Pemodelan Sistem Informasi Geografis	49
4.1 Conceptual framework	50
4.2 Data	50
4.3 Final regression model	51
4.4 Studi Kasus : Pemodelan SIG	51
5 Analisis Sistem Informasi Geografis	55
5.1 Basic Geoprocessing	55
5.2 Terrain analyses	63
5.3 Density Analyses (Analisis Kepadatan)	67
5.4 Analisis kesesuaian	78

III WebGIS	103
6 GeoServer	105
6.1 Pendahuluan	105
6.2 Komponen yang diperlukan	107
6.3 Instalasi	107
6.4 Menjalankan GeoServer	110
6.5 Administrasi GeoServer	111
6.6 Menambahkan Data	115
6.7 Mengakses Layer	120
7 MapStore2	121
7.1 Quick Start	122
7.2 Menggunakan MapStore2	123
IV Visualisasi dan Penyajian Data	125
8 Visualisasi Data	127
8.1 QGIS	127
8.2 GeoServer	130
9 Penyajian Data	131

Pengantar

Repositori ini berisi kumpulan materi untuk Pelatihan Persiapan Uji Kompetensi Analis SIG yang dilaksanakan di Program Studi Teknik Geomatika, Politeknik Negeri Batam.

Bagian I

Basis Data

Bab 1

Basis Data

1.1 Perancangan Basis Data ¹

1.1.1 Rumusan Dasar

Dalam membangun basis data relasional dari awal, adalah penting untuk memberikan waktu lebih dalam memikirkan **business process** yang terjadi. Basis data yang tidak dirancang dengan baik akan memberikan masalah bagi pengguna, termasuk :

- hilangnya integritas data seiring dengan waktu
- ketidakmampuan dalam mendukung query yang diperlukan
- performa yang buruk, misalnya, lambat dalam menampilkan hasil query

Rumusan dasar dalam merancang basis data adalah membuat tabel yang :

- meminimalisir data berlebih (redundant)
- menggambarkan satu subyek
- memiliki satu Primary Key (kode unik untuk setiap baris rekaman (record))
- tidak mengandung kolom dengan banyak bagian (multi-part field) (Contoh: “302 Walker Bldg, University Park, PA 16802”)
- tidak mengandung kolom dengan ragam nilai (Contoh: Kolom Author hendaknya tidak berisi data seperti “Jones, Martin, Williams”)
- tidak memiliki duplikasi yang tidak perlu (Contoh: hindari penamaan kolom seperti Author1, Author2, Author3)
- tidak memiliki kolom yang nilainya tergantung dari kolom lain(Contoh: jangan membuat kolom gaji (Wage) untuk tabel yang memiliki kolom PayRate dan HrsWorked)

¹Sumber: https://www.e-education.psu.edu/spatialdb/l2_p4.html

1.1.2 Normalisasi

Proses perancangan sebuah basis data seperti yang diuraikan dalam aturan di atas, secara formal disebut dengan normalisasi. Semua perancang basis data melakukan normalisasi, baik mereka menggunakan istilah tersebut untuk menggambarkan prosesnya atau tidak.

Ada tiga tingkatan normalisasi, yaitu :

- **First Normal Form (1NF)**

menggambarkan sebuah basis data yang tabel-tabelnya merepresentasikan entitas yang unik, tidak ada duplikasi kolom (misal, tidak ada Author1, Author2, Author3), memiliki satu atau banyak kolom yang merupakan identitas unik dari setiap baris (primary key, PK). Basis data yang memenuhi persyaratan ini termasuk ke dalam First Normal Form (1NF).

- **Second Normal Form (2NF)**

menggambarkan sebuah basis data yang dalam kondisi 1NF dan juga menghindari kolom non-key yang tergantung dari subset Primary Key. Silakan lihat penjelasan di tautan berikut untuk melihat contoh sederhananya [www.1keydata.com/database-normalization/second-normal-form-2nf.php].

Dalam contoh di tautan tersebut, CustomerID dan StoreID membentuk sebuah composite key (kunci gabungan) – gabungan dari nilai kedua kolom tersebut bersifat unik, menjadi identitas setiap kolom dalam tabel. Dalam kata lain, hanya akan ada satu baris dalam tabel yang memiliki CustomerID 1 dengan StoreID 1, hanya akan ada satu baris dengan CustomerID 1 dan StoreID 3, dan seterusnya. Kolom PurchaseLocation tergantung dari kolom StoreID, yang merupakan sebagian dari Primary Key. Dengan demikian, solusi untuk menempatkan tabel ke dalam 2NF adalah dengan memindahkan hubungan StoreID-PurchaseLocation ke dalam tabel terpisah. Pendekatan ini bersifat intuitif di mana kita dapat membaca nilai PurchaseLocation satu kali dibandingkan berulang kali

- **Third Normal Form (3NF)**

menggambarkan sebuah basis data dalam tingkat 2NF dan juga menghindari kolom yang nilainya diturunkan dari kolom lain yang bukan Primary Key. Contoh kolom Wage seperti disebutkan di atas adalah sebuah pelanggaran aturan 3NF.

Dalam banyak kasus, normalisasi basis data hingga tingkatan 3NF sudah cukup. Perlu ditambahkan juga ada format normalisasi lain seperti Boyce-Codman Normal Form (BCNF, atau 3.5NF), Fourth Normal Form (4NF) dan Fifth Normal Form (5NF). Namun demikian, daripada membuang waktu lebih lama dalam menggambarkan normalisasi tingkat lanjut ini, yang paling simpel adalah mengingat karakteristik dasar dari sebuah tabel yang didesain secara baik seperti di atas. Apabila Anda mengikuti petunjuk tersebut secara seksama, khususnya

secara konsisten mencari data berlebih (redundant), Anda akan mampu melakukan normalisasi basis data.

Singkatnya, semakin tinggi tingkatan normalisasi, maka akan semakin banyak jumlah tabel yang ada dalam basis data. Semakin banyak jumlah tabel, usaha untuk menggabungkan data dengan cara *joins* juga akan semakin sulit, dalam arti semakin tinggi level keahlian yang diperlukan untuk membuat queri dan dalam meningkatkan performa basis data. Proses normalisasi kadang-kadang menghasilkan desain yang terlalu sulit untuk diimplementasikan atau kalaupun dapat dibuat, ia memiliki performa yang buruk (baca: lambat). Pada dasarnya, perancangan basis data adalah menyeimbangkan antara kebutuhan yang terkait dengan integritas data dan efisiensi penyimpanan data dengan kebutuhan yang terkait dengan penggunaannya (insert, update, query).

Contoh

Untuk memahami proses normalisasi basis data, contoh berikut memperlihatkan bagaimana aturan-aturan yang telah dijelaskan dapat diterapkan untuk menghasilkan sebuah basis data yang efisien. Pengusaha Jen dan Barry mengembangkan usaha es krim dan membutuhkan sebuah basis data untuk menjejak pesanan (order). Ketika menerima pesanan, mereka mencatat nama pelanggannya, detil pesanan seperti rasa (flavors) dan jumlah es krim yang dibutuhkan, tanggal pesanan dan alamat pengiriman pesanan. Basis data ini diharapkan dapat membantu mereka dalam menjawab dua penting sebagai berikut :

- Pesanan mana yang tenggatnya dalam 2 hari ke depan?
- Perasa apa yang harus dibuat dalam jumlah yang lebih banyak?

Tabel pesanan yang dapat dibuat pertama kali adalah seperti gambar berikut :

Customer	Flavor1	Qty1	Flavor2	Qty2	Flavor3	Qty3	DeliveryDate	DeliveryAdd
Eric Cartman	vanilla	1	chocolate	2			12/1/11	101 Main St
Bart Simpson	chocolate	10	vanilla	10	strawberry	5	12/3/11	202 School Ln
Stewie Griffin	rocky road	1					12/3/11	303 Chestnut St
Bart Simpson	mint chocolate chip	3	strawberry	2			12/5/11	202 School Ln
Hank Hill	coffee	2	vanilla	3			12/8/11	404 Canary Dr
Stewie Griffin	rocky road	5					12/10/11	303 Chestnut St

Dengan skema di atas, masalah yang muncul adalah ketika mereka akan membuat sebuah query yang menghitung jumlah perasa vanilla yang diperlukan sesuai pesanan. Jumlahnya bercampur dengan nama perasa dan perasa apapun dapat dimasukkan ke dalam daftar di bagian mana saja (tidak akan konsisten

dituliskan dalam kolom pertama atau kedua).

Oleh karenanya, desain seperti berikut sepertinya lebih baik:

Customer	Flavor1	Qty1	Flavor2	Qty2	Flavor3	Qty3	DeliveryDate	DeliveryAdd
Eric Cartman	vanilla	1	chocolate	2			12/1/11	101 Main St
Bart Simpson	chocolate	10	vanilla	10	strawberry	5	12/3/11	202 School Ln
Stewie Griffin	rocky road	1					12/3/11	303 Chestnut St
Bart Simpson	mint chocolate chip	3	strawberry	2			12/5/11	202 School Ln
Hank Hill	coffee	2	vanilla	3			12/8/11	404 Canary Dr
Stewie Griffin	rocky road	5					12/10/11	303 Chestnut St

Skema yang kedua ini lebih baik karena memungkinkan mereka untuk melakukan query perasa tertentu dan menjumlahkan kuantitasnya. Namun untuk menghitung perasa vanila yang diperlukan, mereka harus menghitung jumlahnya dari tiga kolom berbeda. Desain ini juga tidak akan dapat menjawab permasalahan apabila satu orang pelanggan memesan lebih dari tiga perasa.

Desain seperti berikut mungkin menjawab pertanyaan di atas:

Customer	Flavor	Qty	DeliveryDate	DeliveryAdd
Eric Cartman	vanilla	1	12/1/11	101 Main St
Eric Cartman	chocolate	2	12/1/11	101 Main St
Bart Simpson	chocolate	10	12/3/11	202 School Ln
Bart Simpson	vanilla	10	12/3/11	202 School Ln
Bart Simpson	strawberry	5	12/3/11	202 School Ln
Stewie Griffin	rocky road	1	12/3/11	303 Chestnut St
Hank Hill	coffee	2	12/8/11	404 Canary Dr
Hank Hill	vanilla	3	12/8/11	404 Canary Dr
Stewie Griffin	rocky road	5	12/10/11	303 Chestnut St

Desain seperti di atas memungkinkan penghitungan kuantitas perasa vanila yang dibutuhkan sesuai pesanan dengan lebih mudah. Sayangnya, desain

tersebut menghasilkan data berlebih (redundant) dan menuliskan pesanan dari satu pelanggan ke dalam banyak baris (record).

Desain yang paling baik adalah dengan memisahkan data ke dalam empat entitas (Customers, Flavors, Orders and Order Items):

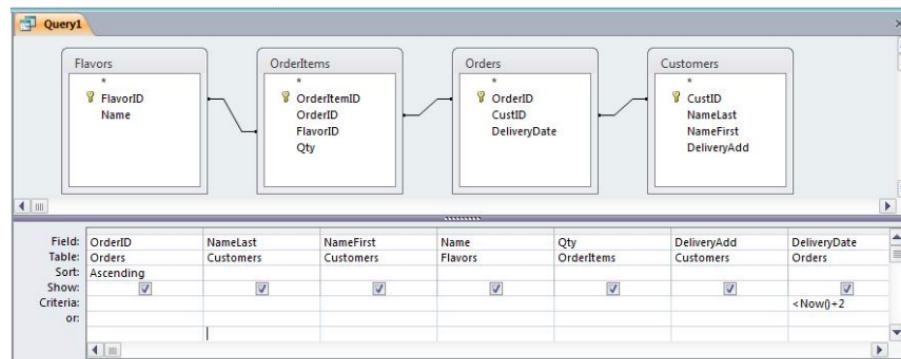
CustID	NameLast	NameFirst	DeliveryAdd
1	Cartman	Eric	101 Main St
2	Simpson	Bart	202 School Ln
3	Griffin	Stewie	303 Chestnut St
4	Hill	Hank	404 Canary Dr

FlavorID	Name
1	vanilla
2	chocolate
3	strawberry
4	rocky road
5	mint chocolate chip
6	coffee

OrderID	CustID	DeliveryDate
1	1	12/1/11
2	2	12/3/11
3	3	12/3/11
4	2	12/5/11
5	4	12/8/11
6	3	12/10/11

OrderItemID	OrderID	FlavorID	Qty
1	1	1	1
2	1	2	2
3	2	2	10
4	2	1	10
5	2	3	5
6	3	4	1
7	4	5	3
8	4	3	2
9	5	6	2
10	5	1	3
11	6	4	5

Apabila mereka hendak mengimplementasikan desain seperti di atas dalam MS-Access, query yang diperlukan untuk menampilkan pesanan yang harus dikirim dalam 2 hari ke depan adalah seperti yang terlihat pada GUI :



1.1.3 Pemodelan Data

Pemodelan data dimulai dengan analisis kebutuhan, dapat dilakukan secara formal atau informal, tergantung dari skala proyeknya. Salah satu perangkat umum yang biasa digunakan dalam proses pemodelan adalah diagram entity-relationship (ER). Diagram ER mengilustrasikan kategori data yang harus disimpan (entitas), juga hubungan (relasi) antara entitas-entitas yang didefinisikan.

Data models describe real-world entities such as customer, service, products, and the relation between these entities. Data models provide an abstraction for the relations in the database. Data models aid the developers in modeling business requirements and translating business requirements to relations. They are

also used for the exchange of information between the developers and business owners.

In the enterprise, data models play a very important role in achieving data consistency across interacting systems. For example, if an entity is not defined, or is poorly defined, then this will lead to inconsistent and misinterpreted data across the enterprise. For instance, if the semantics of the customer entity are not defined clearly, and different business departments use different names for the same entity such as customer and client, this may lead to confusion in the operational departments.

Data model perspectives are defined by ANSI as follows:

- Conceptual data model:

Describes the domain semantics, and is used to communicate the main business rules, actors, and concepts. It describes the business requirements at a high level and is often called a high-level data model.

- Logical data model:

Describes the semantics for a certain technology, for example, the UML class diagram for object-oriented languages.

- Physical data model:

Describes how data is actually stored and manipulated at the hardware level, such as storage area network, table space, CPUs, and so on.

According to ANSI, this abstraction allows changing one part of the three perspectives without amending the other parts. One could change both the logical and the physical data models without changing the conceptual model. To explain, sorting data using bubble or quick sort is not of interest for the conceptual data model. Also, changing the structure of the relations could be transparent to the conceptual model. One could split one relation into many relations after applying normalization rules, or by using enum data types in order to model the lookup tables.

The entity-relation model The entity-relation (ER) model falls into the conceptual data model category. It captures and represents the data model for both business users and developers. The ER model can be transformed into the relational model by following certain techniques.

Conceptual modeling is a part of the software development life cycle (SDLC). It is normally done after the functional and data requirement-gathering stage. At this point, the developer is able to make the first draft of the ER diagram as well as describe functional requirements using data flow diagrams, sequence diagrams, user stories, and many other techniques.

During the design phase, the database developer should give great attention to the design, run a benchmark stack to ensure performance, and validate user requirements. Developers modeling simple systems could start coding directly.

However, care should be taken when making the design, since data modeling involves not only algorithms in modeling the application but also data. The change in design might lead to a lot of complexities in the future such as data migration from one data structure to another.

While designing a database schema, avoiding design pitfalls is not enough. There are alternative designs where one could be chosen. The following pitfalls should be avoided:

Data redundancy: Bad database designs elicit redundant data. Redundant data can cause several other problems, including data inconsistency and performance degradation. When updating a tuple that contains redundant data, the changes on the redundant data should be reflected in all the tuples that contain this data.

Null saturation: By nature, some applications have sparse data, such as medical applications. Imagine a relation called diagnostics, which has hundreds of attributes for symptoms such as fever, headache, sneezing, and so on. Most of them are not valid for certain diagnostics, but they are valid in general. This could be modeled by utilizing complex data types such as JSON.

Tight coupling: In some cases, tight coupling leads to complex and difficult-to-change data structures. Since business requirements change with time, some requirements might become obsolete. Modeling generalization and specialization (for example, a part-time student is a student) in a tightly coupled way may cause problems.

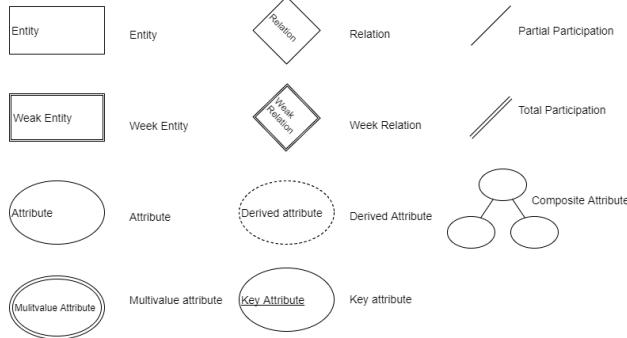
Sebuah diagram ER pada dasarnya adalah cetak biru dari satu struktur basis data. Beberapa RDBMS menyediakan perangkat untuk menggambar diagram ER (contohnya Oracle Designer, MySQL workbench, PostgreSQL pgModeler) dan juga kadang menyediakan kemampuan untuk membuat struktur tabel yang telah dikonsepkan dalam bentuk diagram ke basis data.

Dalam konteks GIS, ESRI menyediakan sebuah perangkat yang memungkinkan kita membuat geodatabase baru berdasarkan diagram dengan CASE (Computer-Aided Software Engineering) tools. Untuk lebih lanjutnya, Anda dapat membaca blog ini yang menyelaskan secara detil penggunaan CASE, Using Case tools in Arc GIS 10, [<http://blogs.esri.com/esri/arcgis/2010/08/05/using-case-tools-in-arcgis-10/>].

1.1.4 Sample application

In order to explain the basics of the ER model, an online web portal to buy and sell cars will be modeled. The requirements of this sample application are as follows, and an ER model will be developed step by step:

The portal provides the facility to register users online and provides different services for users based on their categories. Users might be sellers or normal users. The sellers can create new car advertisements; other users can explore and search for cars. All users should provide their full name and a valid email address during registration. The email address will be used for logging in. The



Gambar 1.1: Simbol Entitas

seller should also provide an address. The user can rate the advertisement and the seller's service quality. All users' search history should be maintained for later use. The sellers have ranks and this affects the advertisement search; the rank is determined by the number of posted advertisements and the user's rank. The car advertisement has a date and the car can have many attributes such as color, number of doors, number of previous owners, registration number, pictures, and so on. Entities, attributes, and keys The ER diagram represents entities, attributes, and relationships. An entity is a representation of a real-world object such as a car or a user. An attribute is a property of an object and describes it. A relationship represents an association between two or more entities.

The attributes might be composite or simple (atomic). Composite attributes can be divided into smaller subparts. A subpart of a composite attribute provides incomplete information that is semantically not useful by itself. For example, the address is composed of a street name, building number, and postal code. Any one of them isn't useful alone, without its counterparts.

Attributes could also be single-valued or multi-valued. The color of a bird is an example of a multi-valued attribute. It can be red and black, or a combination of any other colors. A multi-valued attribute can have a lower and upper bound to constrain the number of values allowed. In addition, some attributes can be derived from other attributes. Age can be derived from the birth date. In our example, the final rank of a seller is derived from the number of advertisements and the user ratings.

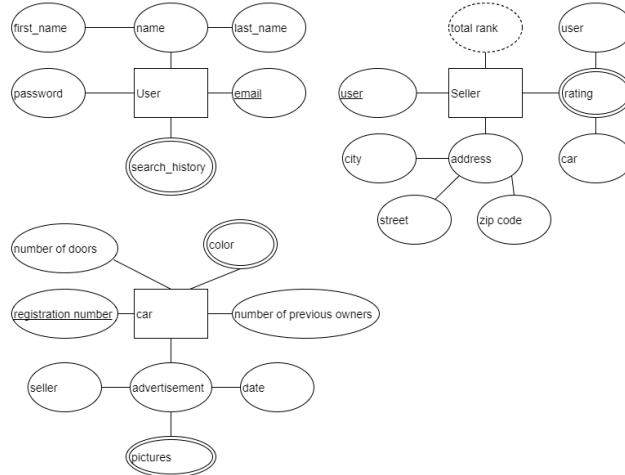
Key attributes can identify an entity in the real world. A key attribute should be marked as a unique attribute, but not necessarily as a primary key, when physically modeling the relation. Finally, several attribute types could be grouped together to form a complex attribute:

Entities should have a name and a set of attributes. They are classified into the following:

Weak entity: Does not have key attributes of its own Strong entity/regular entity: Has a key attribute A weak entity is usually related to a strong entity. This strong entity is called the identifying entity. Weak entities have a partial key, also known as a discriminator, which is an attribute that can uniquely identify the weak entity, and it is related to the identifying entity. In our example, if we assume that the search key is distinct each time the user searches for cars, then the search key is the partial key. The weak entity symbol is distinguished by surrounding the entity box with a double line.

The following diagram shows the preliminary design of a car portal application. The user entity has several attributes. The name attribute is a composite attribute, and email is a key attribute. The seller entity is a specialization of the user entity. The total rank is a derived attribute calculated by aggregating the user ratings and the number of advertisements. The color attribute of the car is multi-valued. The seller can be rated by the users for certain advertisements; this relation is a ternary relation, because the rating involves three entities, which are car, seller, and user.

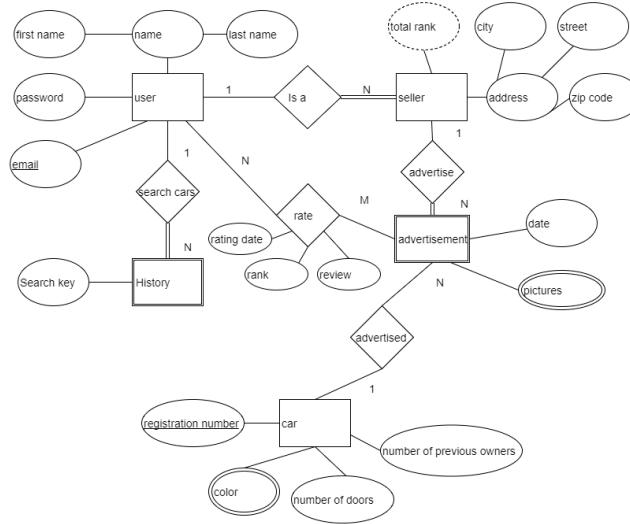
The car picture is a subpart attribute of the advertisement. The following diagram shows that the car can be advertised more than once by different sellers. In the real world, this makes sense, because one could ask more than one seller to sell his car:



When an attribute of one entity refers to another entity, some relationships exist. In the ER model, these references should not be modeled as attributes but as relationships or weak entities. Similar to entities, there are two classes of relationships: weak and strong. Weak relationships associate the weak entities with other entities. Relationships can have attributes as entities. In our example, the car is advertised by the seller; the advertisement date is a property of the relationship.

Relationships have cardinality constraints to limit the possible combinations of entities that participate in a relationship. The cardinality constraint of car and seller is 1:N; the car is advertised by one seller, and the seller can advertise many cars. The participation between seller and user is called total participation, and is denoted by a double line. This means that a seller cannot exist alone, and he must be a user.

Note The many-to-many relationship cardinality constraint is denoted by N:M to emphasize the different participation by the entities.



Up until now, only the basic concepts of ER diagrams have been covered. Some concepts, such as (min, max) cardinality notation, ternary/n-ary relationships, generalization, specialization, and enhanced entity relation(EER) diagrams, have not been discussed.

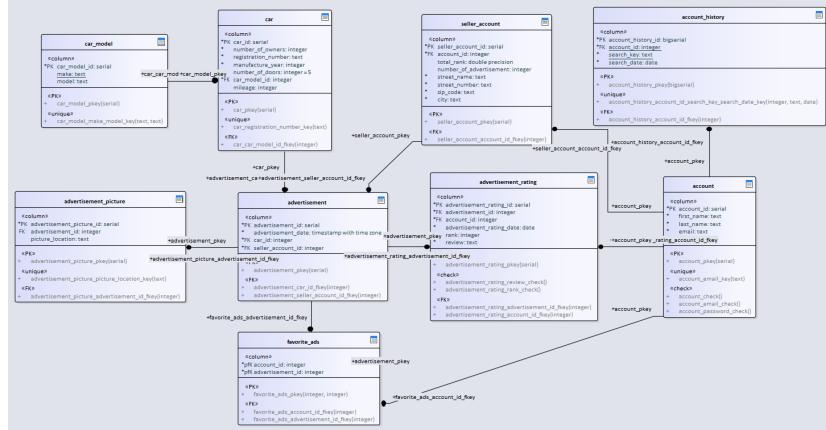
Mapping ER to relations The rules to map an ER diagram to a set of relations (that is, the database schema) are almost straightforward, but not rigid. One could model an entity as an attribute, and then refine it to a relationship. An attribute that belongs to several entities can be promoted to be an independent entity. The most common rules are as follows (note that only basic rules have been covered, and the list is not exhaustive):

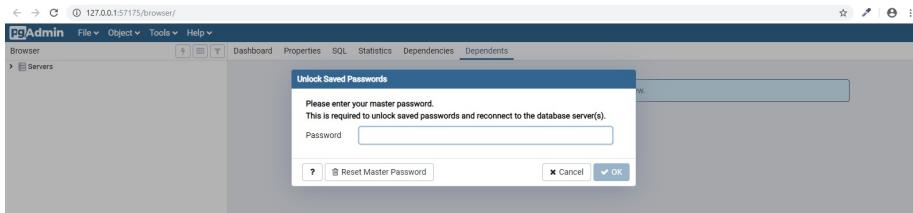
Map regular entities to relations. If entities have composite attributes, then include all the subparts of the attributes. Pick one of the key attributes as a primary key. Map weak entities to relations. Include simple attributes and the subparts of the composite attributes. Add a foreign key to reference the identifying entity. The primary key is normally the combination of the partial key and the foreign key. If a relationship has an attribute and the relation cardinality is 1:1, then the relation attribute can be assigned to one of the participating entities. If a relationship has an attribute and the relation cardinality is 1:N,

then the relation attribute can be assigned to the participating entity on the N side. Map many-to-many relationships, also known as N:M, to a new relation. Add foreign keys to reference the participating entities. The primary key is the composition of foreign keys. Map a multi-valued attribute to a relation. Add a foreign key to reference the entity that owns the multi-valued attribute. The primary key is the composition of the foreign key and the multi-valued attribute. UML class diagrams Unified Modeling Language (UML) is a standard developed by the Object Management Group (OMG). UML diagrams are widely used in modeling software solutions, and there are several types of UML diagrams for different modeling purposes including class, use case, activity, and implementation diagrams.

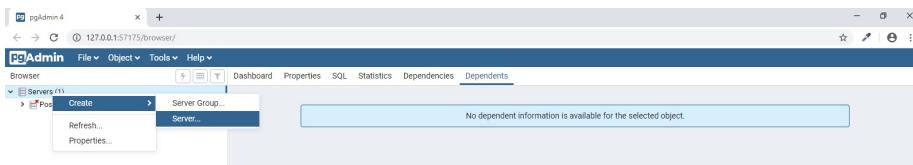
A class diagram can represent several types of associations, that is, the relationship between classes. They can depict attributes as well as methods. An ER diagram can be easily translated into a UML class diagram. UML class diagrams also have the following advantages:

Code reverse-engineering: The database schema can be easily reversed to generate a UML class diagram. Modeling extended relational database objects: Modern relational databases have several object types such as sequences, views, indexes, functions, and stored procedures. UML class diagrams have the capability to represent these object types. The following class diagram is generated from reverse-engineering the SQL code of a car_portaldatabase:





Gambar 1.2: PgAdmin 4



Gambar 1.3: PgAdmin 4

1.2 Pembuatan Basis Data dengan PostGIS²

1.2.1 Membuat Skema Baru dengan pgAdmin 4

Instalasi **PgAdmin 4** sangat mudah. Anda tinggal men-download-nya dari download page di situsnya, dan laksanakan proses instalasi di **workstation** hingga selesai. Sebagai catatan, Anda akan diminta untuk membuat **master password**, yaitu *password* yang digunakan saat pertama kali mengakses **PgAdmin 4** di **workstation** Anda.

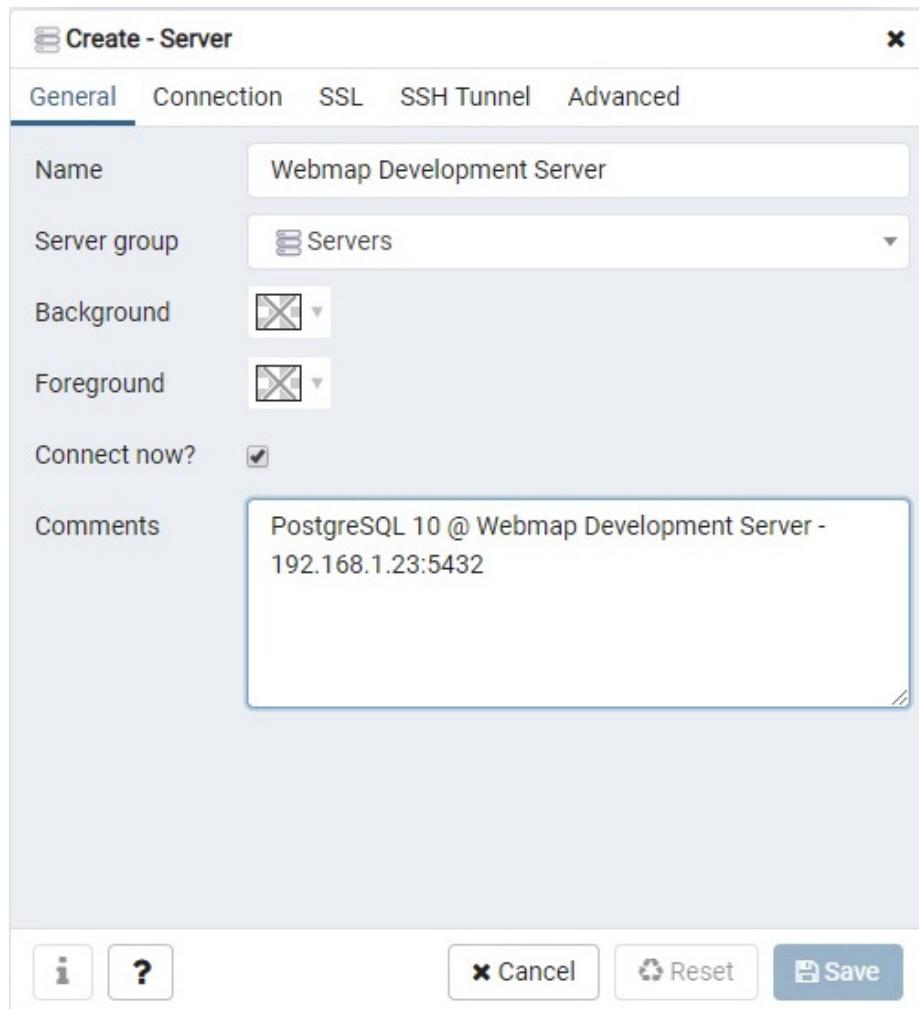
Setelah Anda berhasil masuk ke **PgAdmin 4**, maka yang pertama kali harus dilakukan adalah *create connection* ke server PostgreSQL yang akan Anda akses.

Pada *dialog* ini, di *tab General* kita isi **Name** dengan **Webmap Development Server** (atau sesuka Anda), kemudian *checkbox Connect now?*-nya kita *check*, dan **Comments**-nya kita isi dengan deskripsi koneksinya.

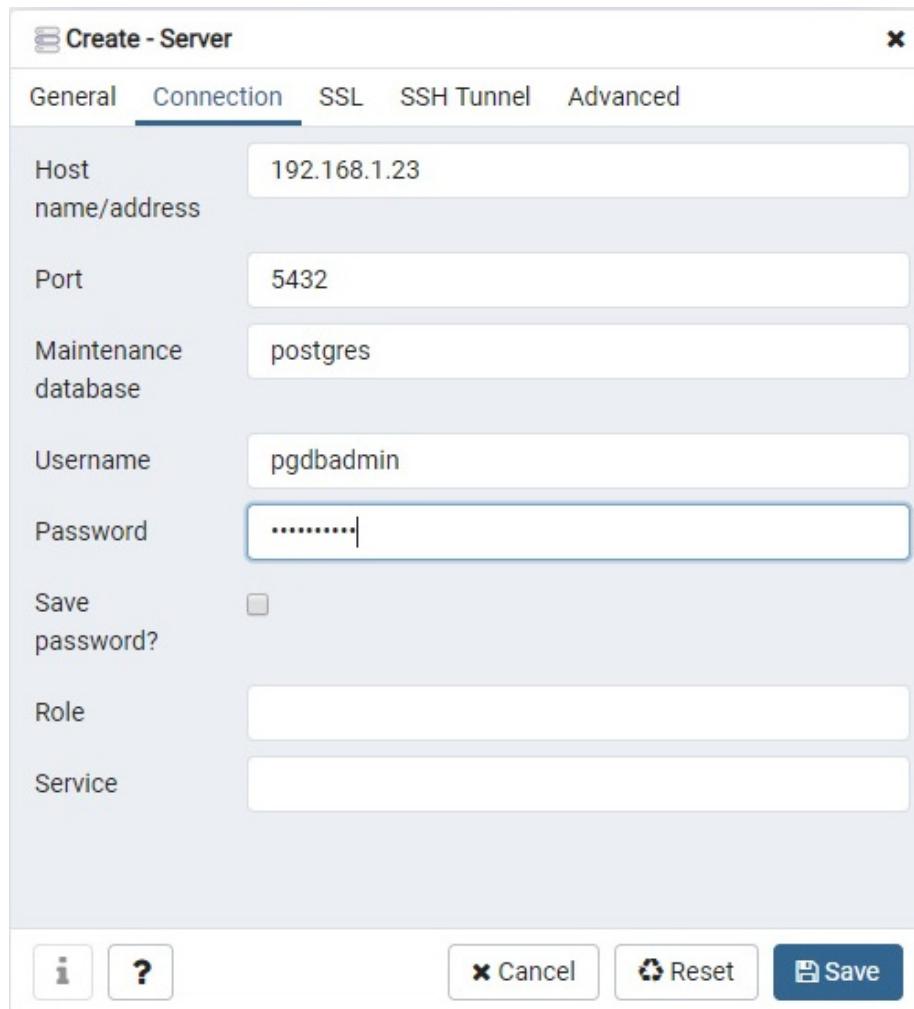
Pindah ke *tab Connection*, kita isi **Host name/address** dengan **192.168.1.23** (IP server PostgreSQL-nya), **Port: 5432**, **Username: pgdbadmin** (biar bisa mengakses seluruh database yang ada), dan **password**-nya. *Checkbox Save Password?*-nya boleh di-*check*, tapi lebih baik dibiarkan *unchecked* saja, sehingga setiap kali koneksi Anda akan diminta untuk memasukkan *password*.

Kalau seluruh isian kita sudah benar, maka begitu kita klik **Save**, maka entry **Webmap Development Server** akan muncul di pilihan server pada **PgAdmin 4**:

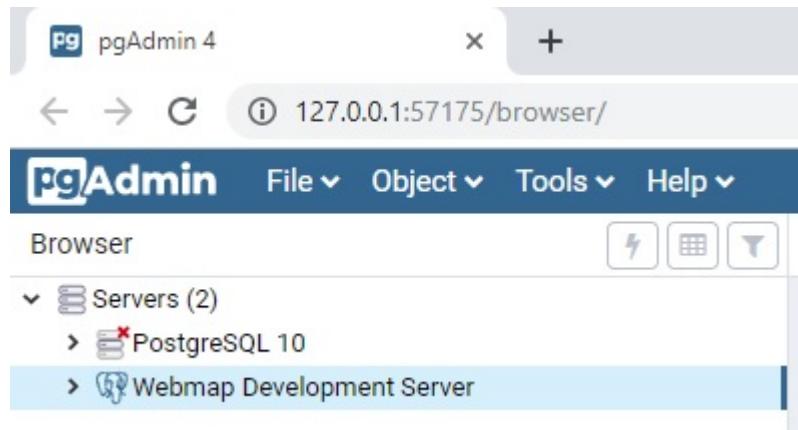
²Sumber: <https://github.com/andyprasetya/webmap-development-server>



Gambar 1.4: PgAdmin 4



Gambar 1.5: PgAdmin 4



Gambar 1.6: PgAdmin 4

Waktu kita *unfold entry* ini, maka akan muncul pilihan akses ke **Databases**, **Login/Group Roles** dan **Tablespaces**. Selanjutnya, kita akan fokus ke **Databases** dulu.

Setelah kita *unfold Databases*, maka akan terlihat **3 database**, yaitu **postgres** (*default database*, yang digunakan oleh PostgreSQL), **postgis_template** (*database* yang sudah kita *create* sebelumnya dan kita fungsikan sebagai *template database*) dan **webmap_db** (*database* yang akan kita akses selanjutnya).

Masuk ke **webmap_db -> Schemas -> public -> Tables**, maka akan terlihat *table* bernama **ne_10m_admin_0_countries**, yang mana itu adalah hasil *upload* shapefile yang sudah kita laksanakan pada bagian sebelumnya.

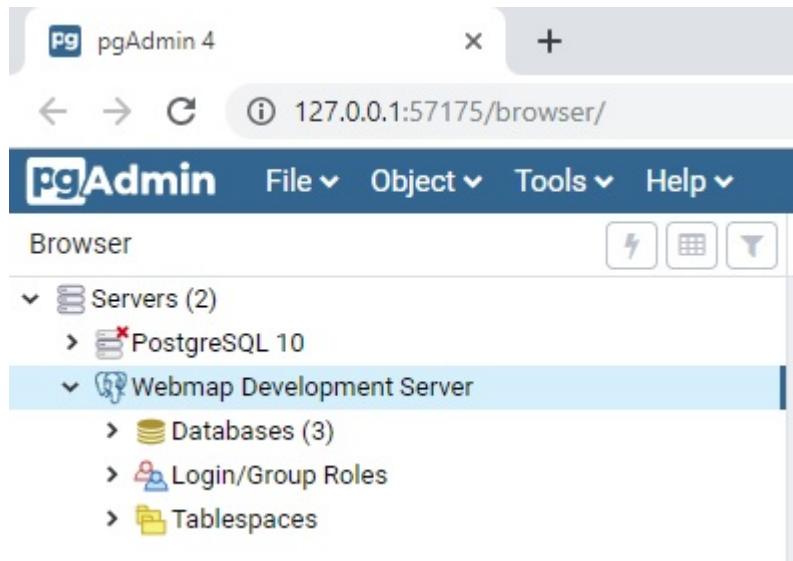
Klik-kanan pada *table* tersebut (**ne_10m_admin_0_countries**), pilih **View/Edit Data -> All Rows:**

Maka selanjutnya pada bagian kanan (tampilan utama) dari PgAdmin 4 akan muncul tampilan *query* dan seluruh *rows* yang ada dalam *table ne_10m_admin_0_countries*.

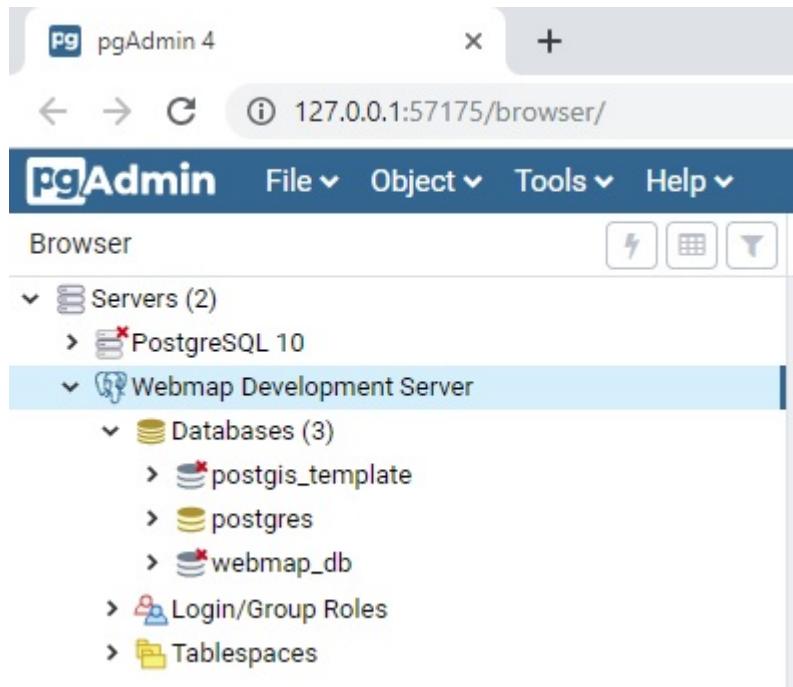
Menariknya pada PgAdmin 4 ini, jika Anda *scroll* ke kanan terus hingga akhir *table*, akan ada sebuah *button* yang berfungsi untuk menampilkan/visualisasi data *geometry*-nya.

Kalau di-klik *geometry viewer button* ini, maka selanjutnya akan muncul *wemap* berbasis **Leaflet.JS** yang menampilkan data *geometry*-nya.

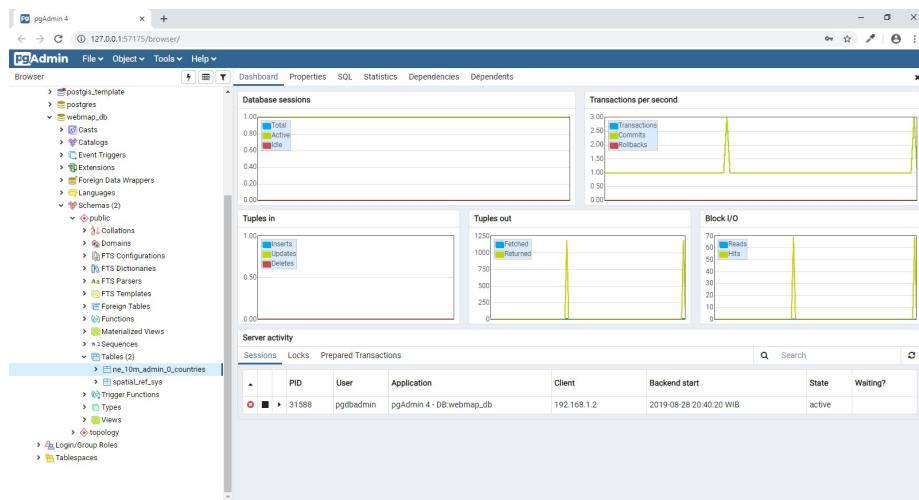
Sebagai catatan, *basemap* dari **OpenStreetMap** hanya akan muncul apabila SRID-nya **EPSG 4326**. Saya belum mencoba untuk **EPSG 3857** atau lainnya. Untuk lebih jelasnya mengenai perbedaan antara **EPSG 4326** dan **EPSG 3857**, dapat Anda baca di artikel bertajuk EPSG 4326 vs EPSG 3857 ini.



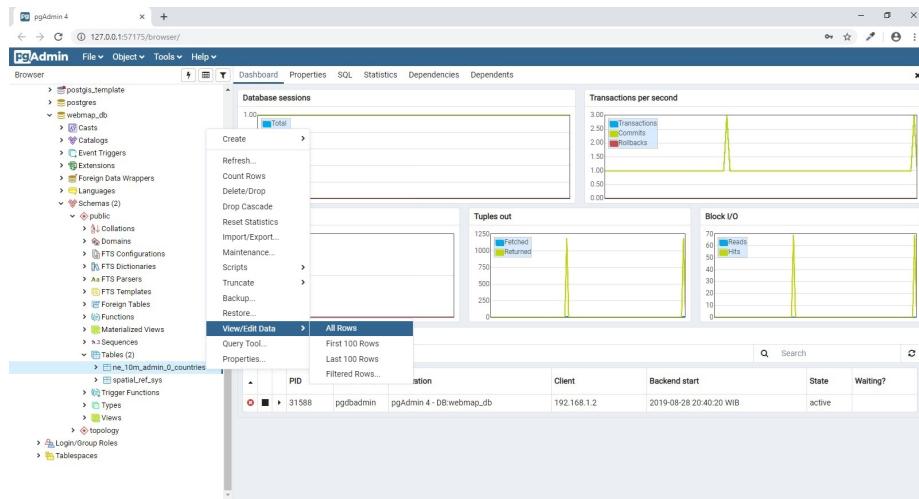
Gambar 1.7: PgAdmin 4



Gambar 1.8: PgAdmin 4



Gambar 1.9: PgAdmin 4



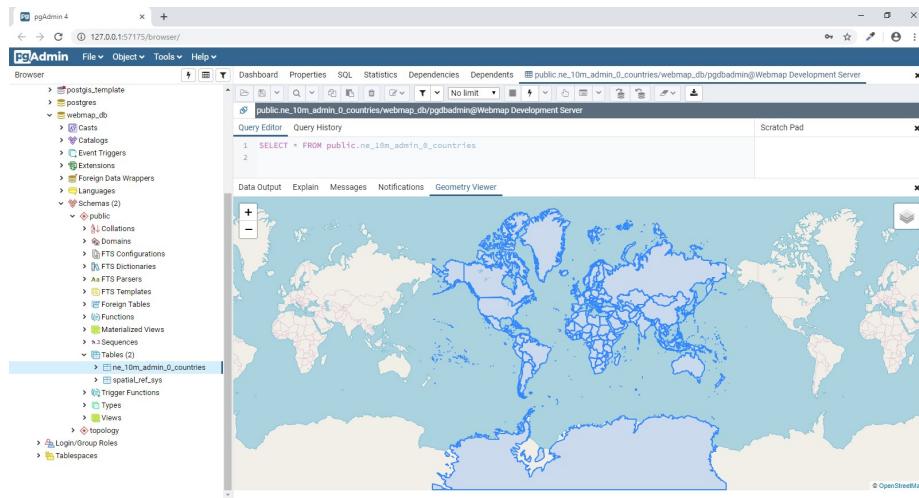
Gambar 1.10: PgAdmin 4

gid	name_pt	name_en	name_ar	name_tr	name_vi	name_zh	geom
1	Indonesia	Indonesia	Indonesia	Indonesia	Indonesia	印度尼西亚	0106000000E61...
2	Malásia	Malaysia	ماليزيا	Malezya	Malezya	马来西亚	0106000000E61...
3	Chile	Chile	Чили	Sill	智利	智利	0106000000E61...
4	Bolívia	Bolivia	بوليفيا	Bolivia	Bolivia	玻利维亚	0106000000E61...
5	Peru	Peru	پرو	Peru	Peru	秘鲁	0106000000E61...
6	Argentina	Argentina	آرژانتینا	Argentina	Argentina	阿根廷	0106000000E61...
7	Etiópia	Ethiopia	إثيوبيا	Ethiopia	Ethiopia	埃塞俄比亚	0106000000E61...
8	Ciprus	Cyprus	کیپر	Cyprus	Cyprus	塞浦路斯	0106000000E61...
9	Índia	India	ભારત	India	India	印度	0106000000E61...
10	China	China	چین	China	China	中国	0106000000E61...
11	Israel	Israel	ישראל	İsrail	Israel	以色列	0106000000E61...
12	Lebanon	Lebanon	لبنان	LBN	Lebanon	黎巴嫩	0106000000E61...
13	Bolívia	Bolivia	بوليفيا	Bolivia	Bolivia	玻利维亚	0106000000E61...
14	Peru	Peru	پرو	Peru	Peru	秘鲁	0106000000E61...
15	Somália	Somalia	صومالی	SOM	Somali	索马里	0106000000E61...
16	Kenya	Kenya	کنیا	KEN	Kenya	肯尼亚	0106000000E61...
17	Pakistan	Pakistan	پاکستان	PAK	Pakistan	巴基斯坦	0106000000E61...
18							

Gambar 1.11: PgAdmin 4

gid	name_pt	name_en	name_ar	name_tr	name_vi	name_zh	geom
1	Indonesia	Indonesia	Indonesia	Indonesia	Indonesia	印度尼西亚	0106000000E61...
2	Malaysia	Malaysia	ماليزيا	Malezya	Malezya	马来西亚	0106000000E61...
3	Chile	Chile	Чили	Sill	智利	智利	0106000000E61...
4	Bolivia	Bolivia	بوليفيا	Bolivia	Bolivia	玻利维亚	0106000000E61...
5	Peru	Peru	پرو	Peru	Peru	秘鲁	0106000000E61...
6	Argentina	Argentina	آرژانتینا	Argentina	Argentina	阿根廷	0106000000E61...
7	Ethiopia	Ethiopia	إثيوبيا	Ethiopia	Ethiopia	埃塞俄比亚	0106000000E61...
8	Cyprus	Cyprus	کیپر	Cyprus	Cyprus	塞浦路斯	0106000000E61...
9	India	India	ભારત	India	India	印度	0106000000E61...
10	China	China	چین	China	China	中国	0106000000E61...
11	Israel	Israel	ישראל	İsrail	Israel	以色列	0106000000E61...
12	Palestina	Palestina	پaleستینا	Palestina	Palestine	巴勒斯坦	0106000000E61...
13	Lebanon	Lebanon	لبنان	Lban	Lebanon	黎巴嫩	0106000000E61...
14	Etiopia	Etiopia	إثيوبيا	Etiopia	Ethiopia	埃塞俄比亚	0106000000E61...
15	Sudão do Sul	Sudan	سودان	Sudan	Sudan	苏丹	0106000000E61...
16	Somália	Somalia	صومالی	Somalia	Somali	索马里	0106000000E61...
17	Kenya	Kenya	کنیا	Kenya	Kenya	肯尼亚	0106000000E61...
18	Pakistan	Pakistan	پاکستان	Pakistan	Pakistan	巴基斯坦	0106000000E61...

Gambar 1.12: PgAdmin 4



Gambar 1.13: PgAdmin 4

1.2.2 PostGIS Shapefile and DBF Loader/Exporter

Lakukan langkah-langkah ini di workstation Anda.

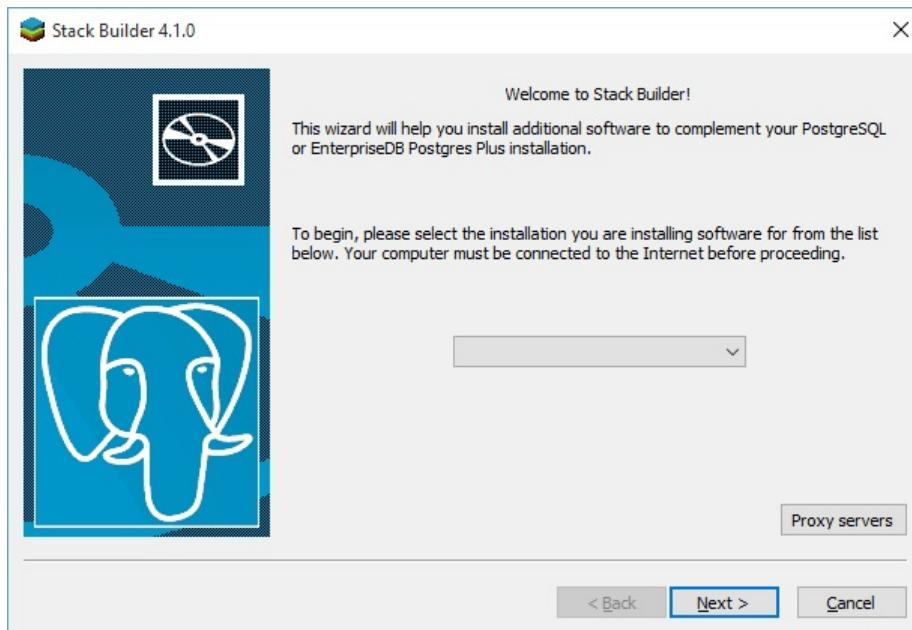
FYI, PostGIS Shapefile and DBF Loader/Exporter adalah sebuah aplikasi seberhana yang menjadi bagian dari paket instalasi PostGIS. Instalasi PostGIS membutuhkan PostgreSQL yang sudah terinstall (dan aktif) sebelumnya. Nah, menurut opini saya, instalasi PostgreSQL dan PostGIS di workstation *nggak* berguna, kecuali hanya untuk “*memancing*” instalasi **Application Stack Builder**, biar bisa install PostGIS yang mana di proses instalasinya akan mengikutkan **PostGIS Shapefile and DBF Loader Exporter**. PostgreSQL dan PostGIS *toh* sudah ada di server. Tapi ya... mau *gimana lagi?* *Let's just do it!*

- **Download** installer PostgreSQL dari EnterpriseDB/EDB, dan install sampai selesai.

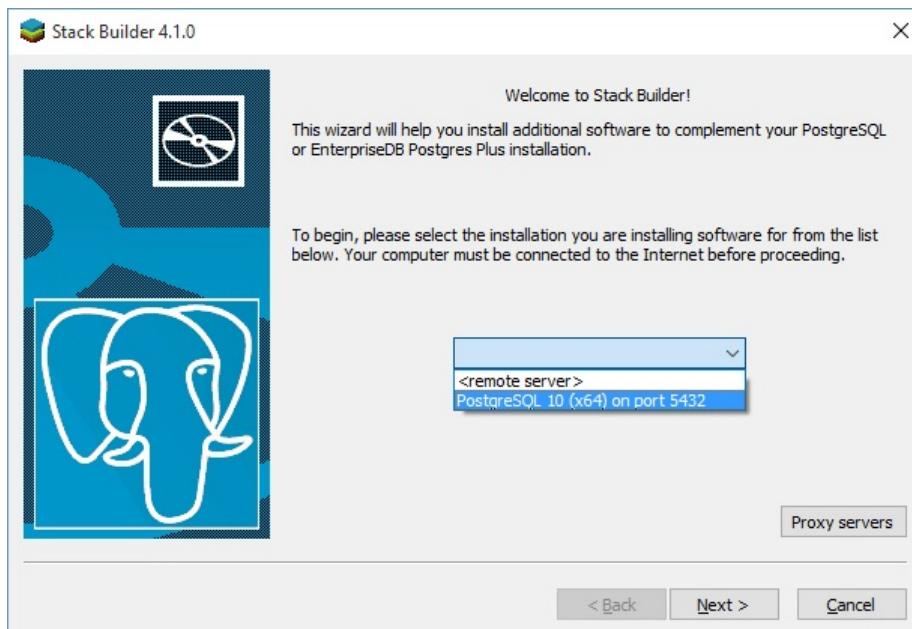
Jika Anda akan langsung melakukan shapefile *upload test*, Anda bisa men-*download* 1 file dari situs **Natural Earth**. Ambil contoh, **batas administrasi negara level 0**.

- Setelah instalasi PostgreSQL selesai, dari *Start menu* jalankan **Application Stack Builder**.
- Pilih **PostgreSQL 10 (x64) on port 5432**, dan klik **Next >**. Jangan pilih yang <**remote server**> yaa..., karena pilihan ini selanjutnya tidak menyediakan opsi instalasi Spatial Extensions (PostGIS dll.).

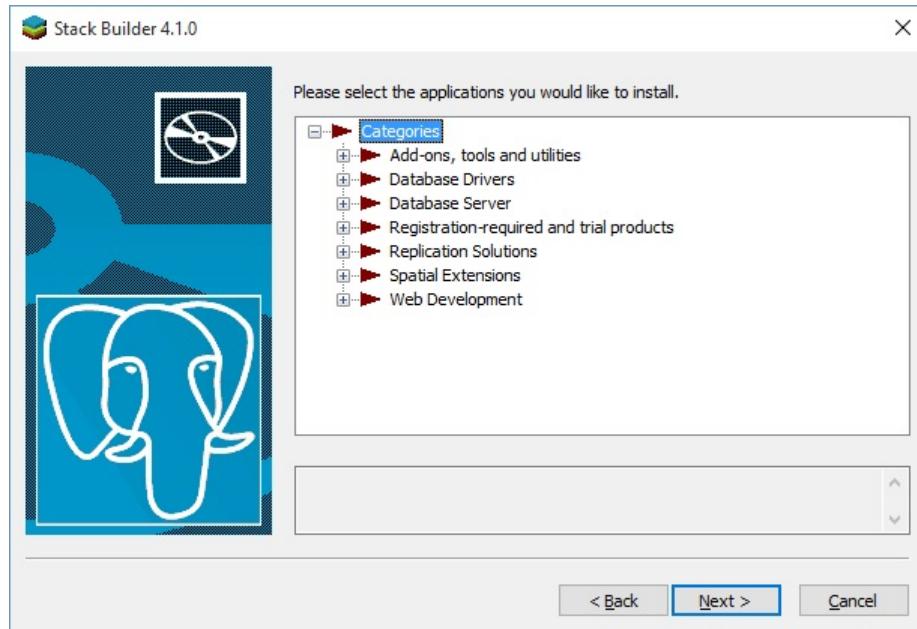
Tunggu beberapa saat, Application Stack Builder akan men-*download list* aplikasi yang bisa Anda *install* di tahap selanjutnya. Jika sudah muncul



Gambar 1.14: Application Stack Builder



Gambar 1.15: Application Stack Builder



Gambar 1.16: Application Stack Builder

tampilan:

- Pilih **Npgsql**, **pgJDBC** dan **psqlODBC** pada kelompok **Database Drivers**, dan **PostGIS 2.5 Bundle for PostgreSQL 10 (64 bit)** pada kelompok **Spatial Extensions**.

Klik **Next >**, dan tunggu beberapa saat hingga Application Stack Builder selesai men-*download* dan meng-*install* seluruh aplikasi yang sudah dipilih. Setelah selesai, keluar/matikan Application Stack Builder-nya dan buka *Start menu*.

Pilih (atau cari dulu) menu **PostGIS 2.x Shapefile and DBF Loader Exporter**. Tampilan aplikasinya:

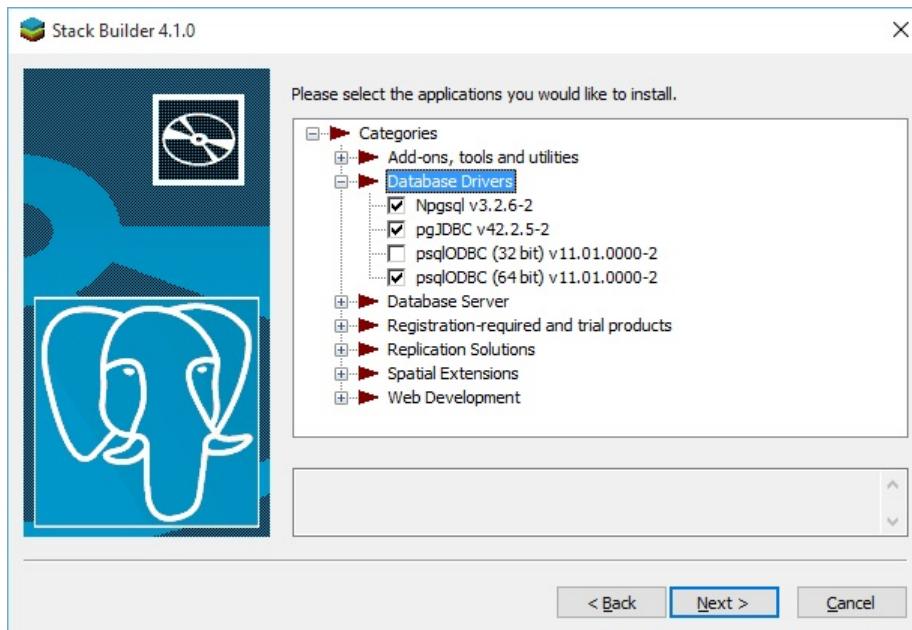
- Connection testing* ke server.

Klik **View connection details...**, dan isikan Username: **pgdbuser**, Password: **[password]**, Server Host: **192.168.1.23** dan port-nya: **5432**, Database: **webmap_db** seperti ini:

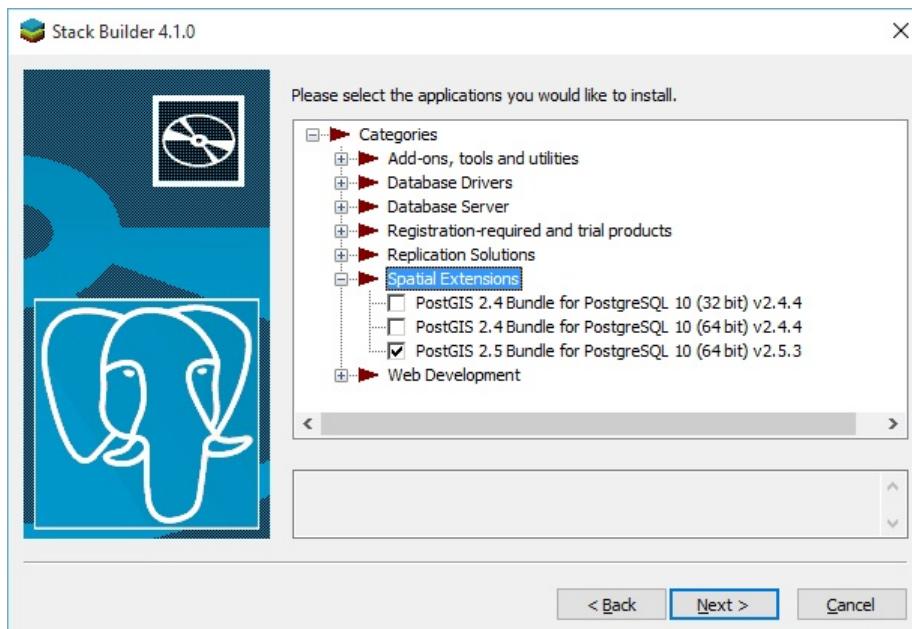
Jika koneksinya sukses, maka pada bagian **Log Window** akan muncul log yang mengkonfirmasi bahwa koneksi berhasil.

Jika koneksi gagal, periksa kembali pengaturan koneksinya.

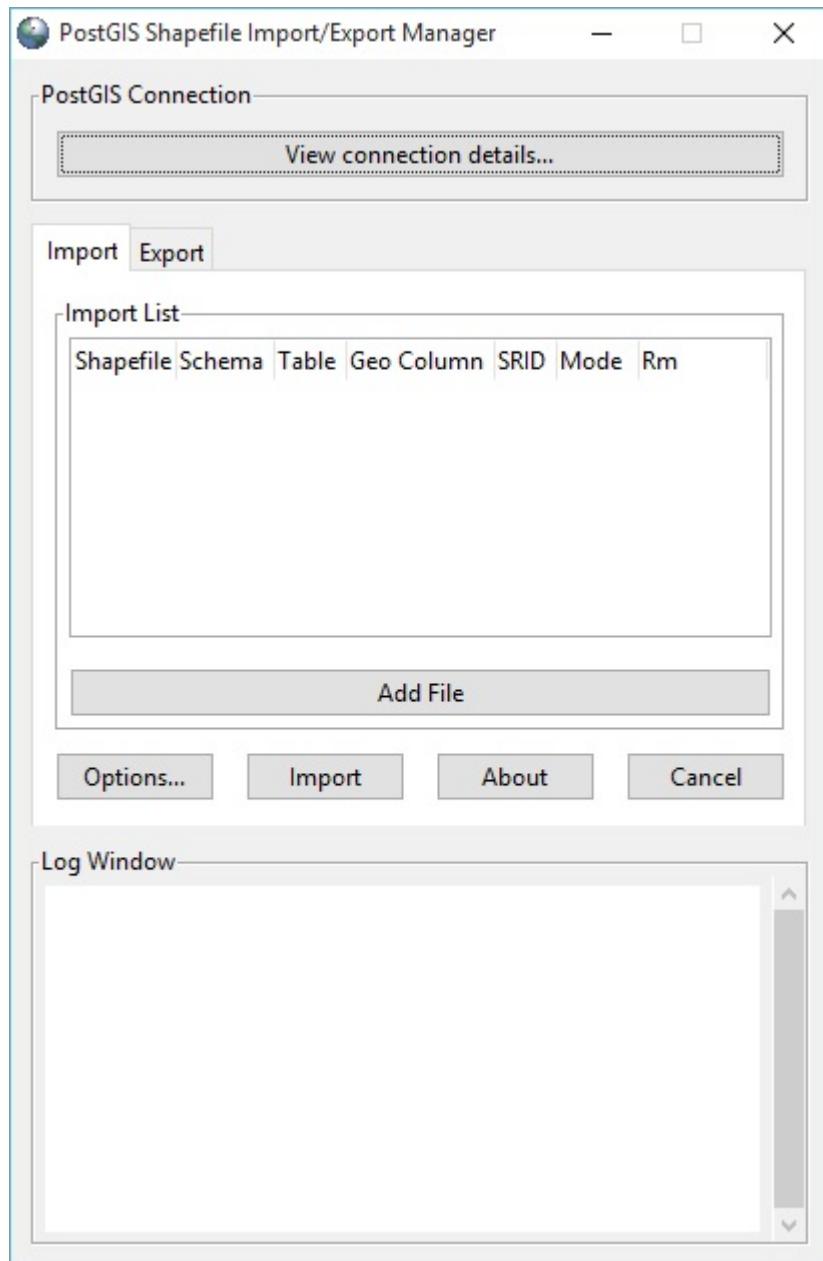
Sekarang saatnya Anda mencoba meng-*upload* sebuah shapefile ke Pos-



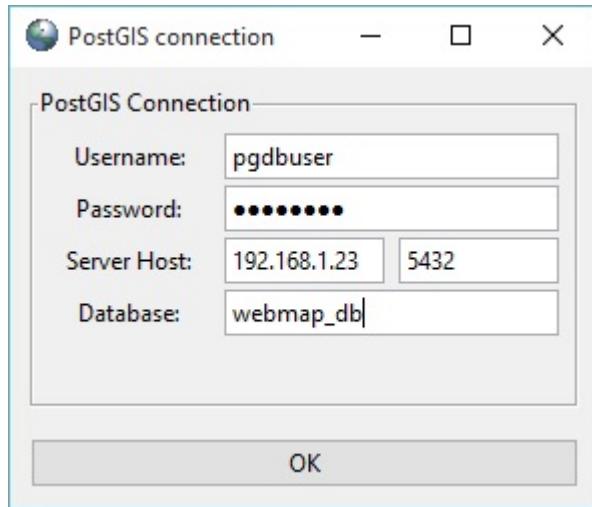
Gambar 1.17: Application Stack Builder



Gambar 1.18: Application Stack Builder



Gambar 1.19: PostGIS Shapefile DBF Loader



Gambar 1.20: PostGIS Shapefile DBF Loader

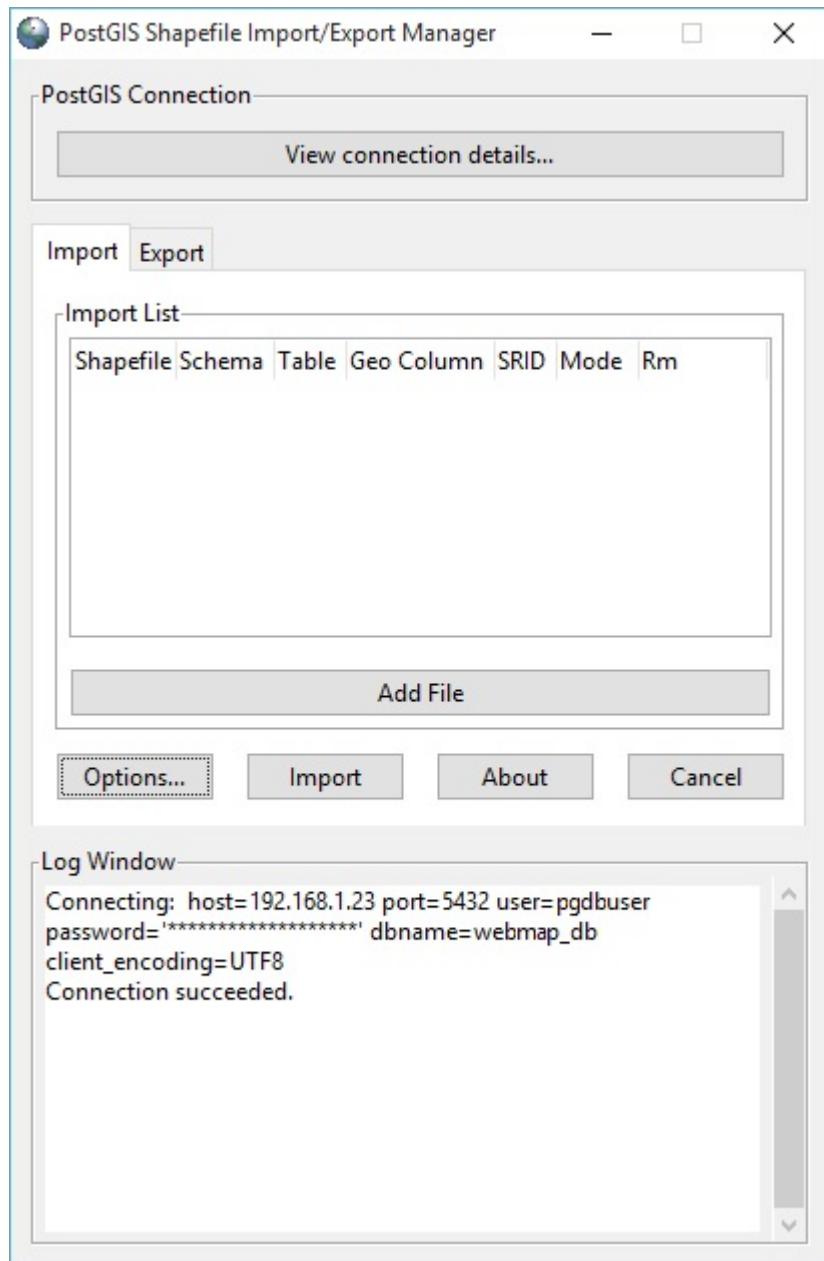
tgreSQL/PostGIS di server menggunakan **PostGIS 2.x Shapefile and DBF Loader Exporter**.

O ya, tapi lebih baik kita bahas dulu spesifikasi file yang pada bagian 8.1 di atas saya sarankan untuk di-*download*, yaitu **batas administrasi negara level 0**. Kalau kita ekstrak file ini, maka kita akan memiliki 1 *set files* yang nama file-nya identik, tapi *extension*-nya berbeda. Dalam konteks pembahasan ini, kita hanya akan fokus pada file `ne_10m_admin_0_countries.prj` saja, karena pada saat *upload* shapefile nanti, kita butuh informasi **SRID** (**Spatial Reference System Identifier**).

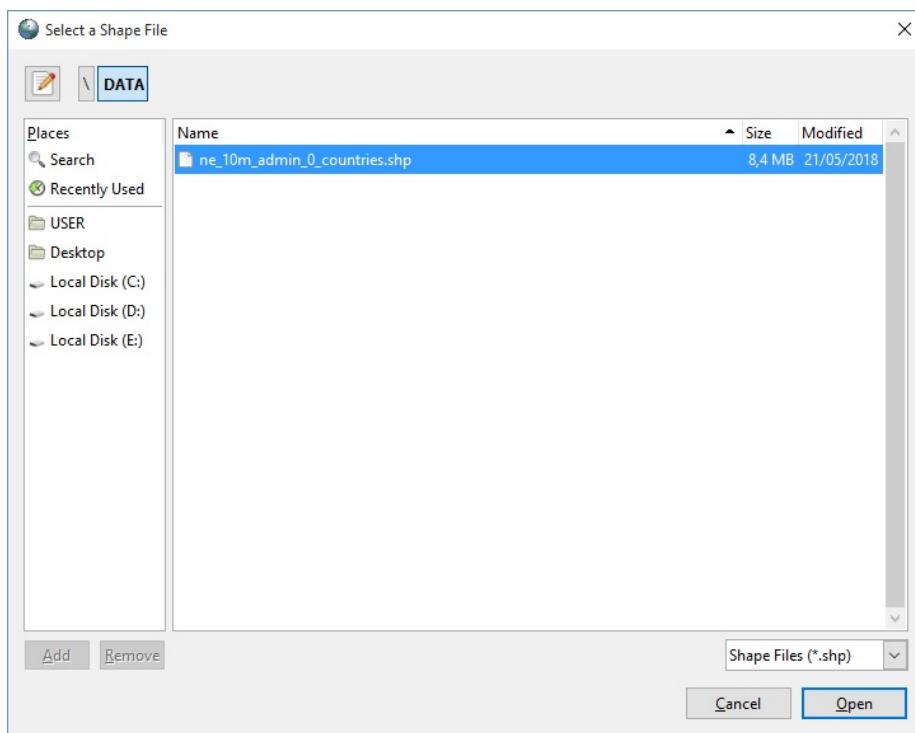
File ber-*extension* `*.prj` ini berisi informasi tentang CRS (Coordinate Reference System) yang diterapkan/digunakan oleh file ber-*extension* `*.shp` dan `*.shx` di direktori yang sama. Kalau Anda membuka file ini di ASCII text editor seperti Notepad atau Notepad++, dan Anda menemui *entry* yang bertuliskan **WGS_1984**, maka besar kemungkinan **SRID**-nya adalah **EPSG 4326**. Lebih lanjut lagi, “tebakan” SRID ini saya kira cukup masuk-akal karena shapefile ini *coverage*-nya *world*. Jika Anda ingin mengetahui lebih lanjut tentang CRS, WGS 1984, SRID, dan lain sebagainya yang terkait, silahkan gali lebih dalam, asal jangan “tersesat” saja (baca: menyerah, dan langsung ngikut paham bumi-datar. Hahaha...).

Kembali ke tampilan **PostGIS Shapefile and DBF Loader Exporter**, langsung saja klik **Add File**, maka dialog **Select a Shape file** muncul, pilih (klik) shapefile yang akan di-*upload*, dan klik **Open**.

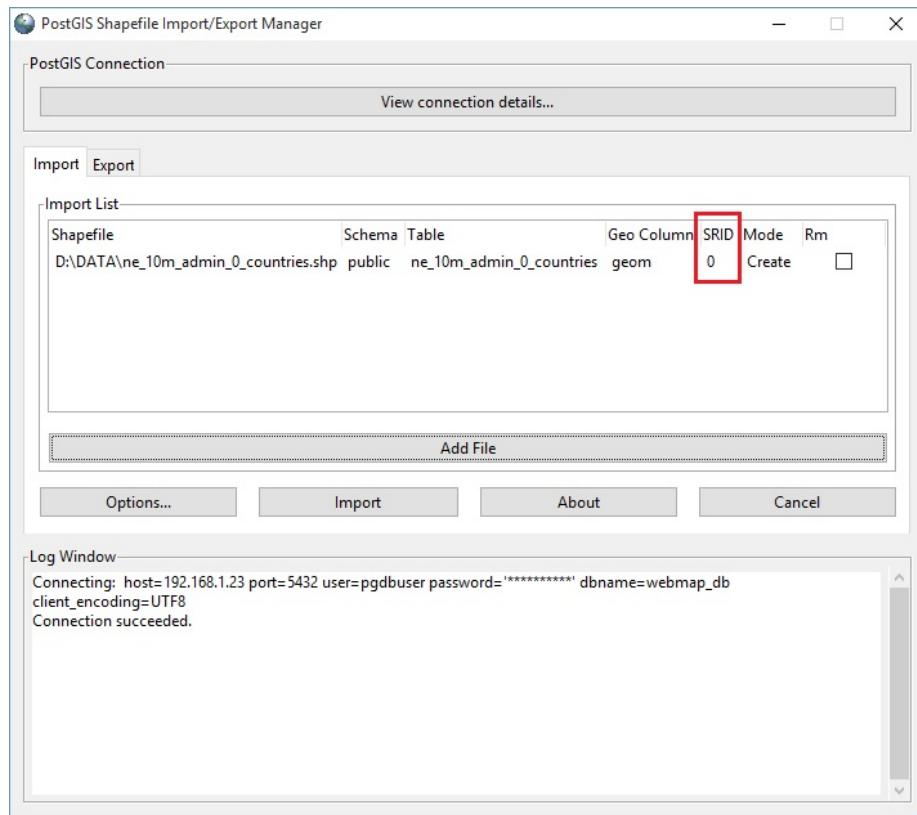
Setelah klik **Open**, maka shapefile tersebut akan masuk ke **Import List**.



Gambar 1.21: PostGIS Shapefile DBF Loader



Gambar 1.22: PostGIS Shapefile DBF Loader



Gambar 1.23: PostGIS Shapefile DBF Loader

Dalam tampilan ini mari kita fokus ke boks merah, yaitu kolom **SRID**.

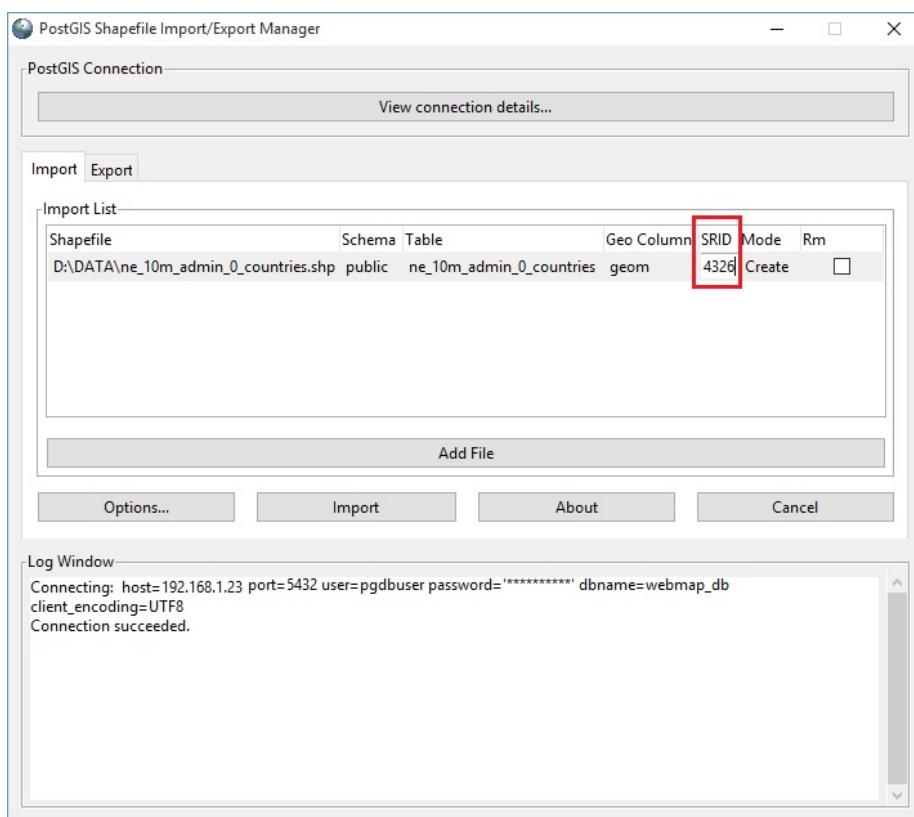
Klik angka **0** dalam kolom, dan isi dengan angka **4326**, dan klik pada ruang kosong dalam **Import List**, di bawah *entry* shapefile-nya. Untuk nama **Table** dan **Geo Column** yang akan jadi target di PostgreSQL/PostGIS biarkan saja apa-adanya.

Selanjutnya klik **Import**, dan tunggu beberapa saat sampai selesai. Jika tidak ada *error* saat proses *upload*, maka setelah selesai di **Log Window**-nya akan muncul konfirmasi bahwa *upload* shapefile-nya berhasil.

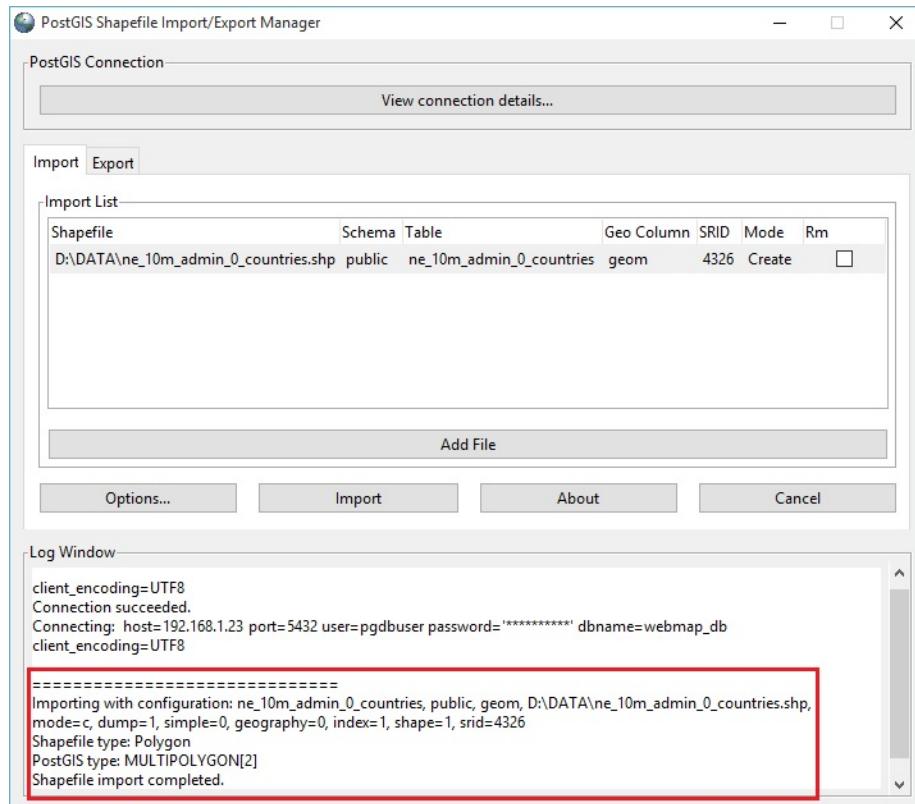
Sampai pada tahap ini, di PostgreSQL/PostGIS server sudah ada contoh geodata yang sudah siap diakses dari berbagai kanal.

- Testing PostGIS Layer di Quantum GIS.

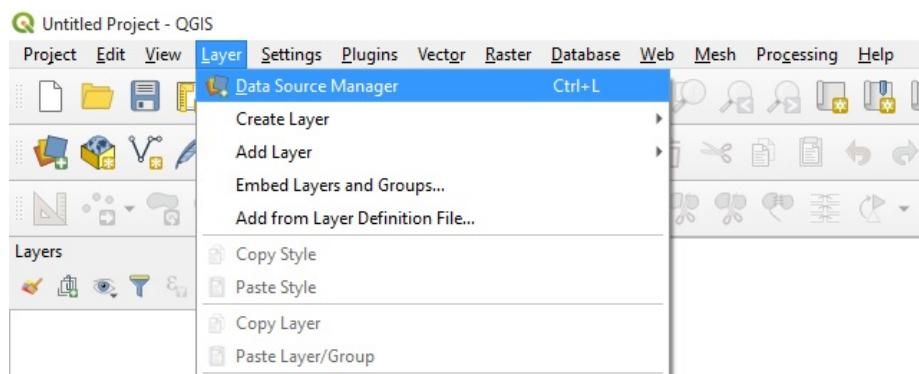
Test mengakses PostGIS *layer* yang paling sederhana adalah dengan menggunakan **Quantum GIS**. Aktifkan Quantum GIS Anda, buat *project* baru, kemudian klik menu **Layer -> Data Source Manager**:



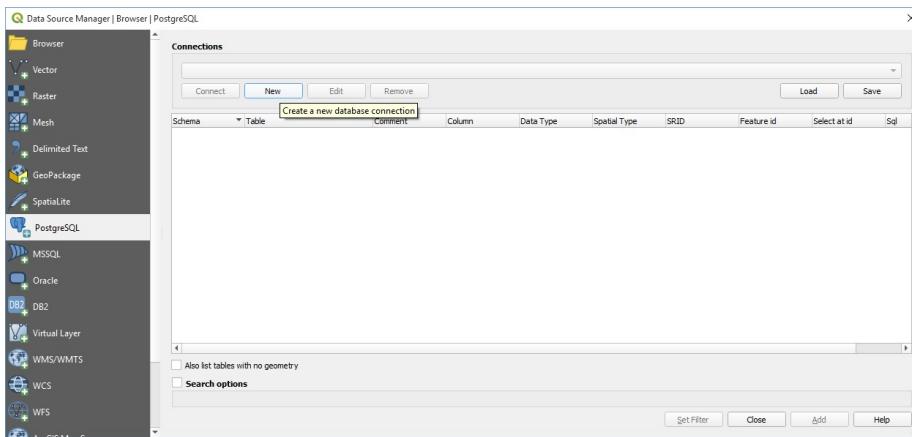
Gambar 1.24: PostGIS Shapefile DBF Loader



Gambar 1.25: PostGIS Shapefile DBF Loader



Gambar 1.26: QGIS PostGIS Layer



Gambar 1.27: QGIS PostGIS Layer

Setelah dialog **Data Source Manager** muncul, klik **PostgreSQL** pada bagian kiri, sehingga muncul tampilan koneksi ke PostgreSQL di bagian kanan, dan pada bagian **Connections**, klik **New**:

Setelah Anda klik **New**, maka akan muncul dialog **Create a New PostGIS Connection**.

Pada bagian **Connection Information**, isi Name: **webmap_db@192.168.1.23** (atau yang lain sesuka Anda), Service dibiarkan kosong saja, Host: **192.168.1.23**, Port: **5432** dan Database: **webmap_db**.

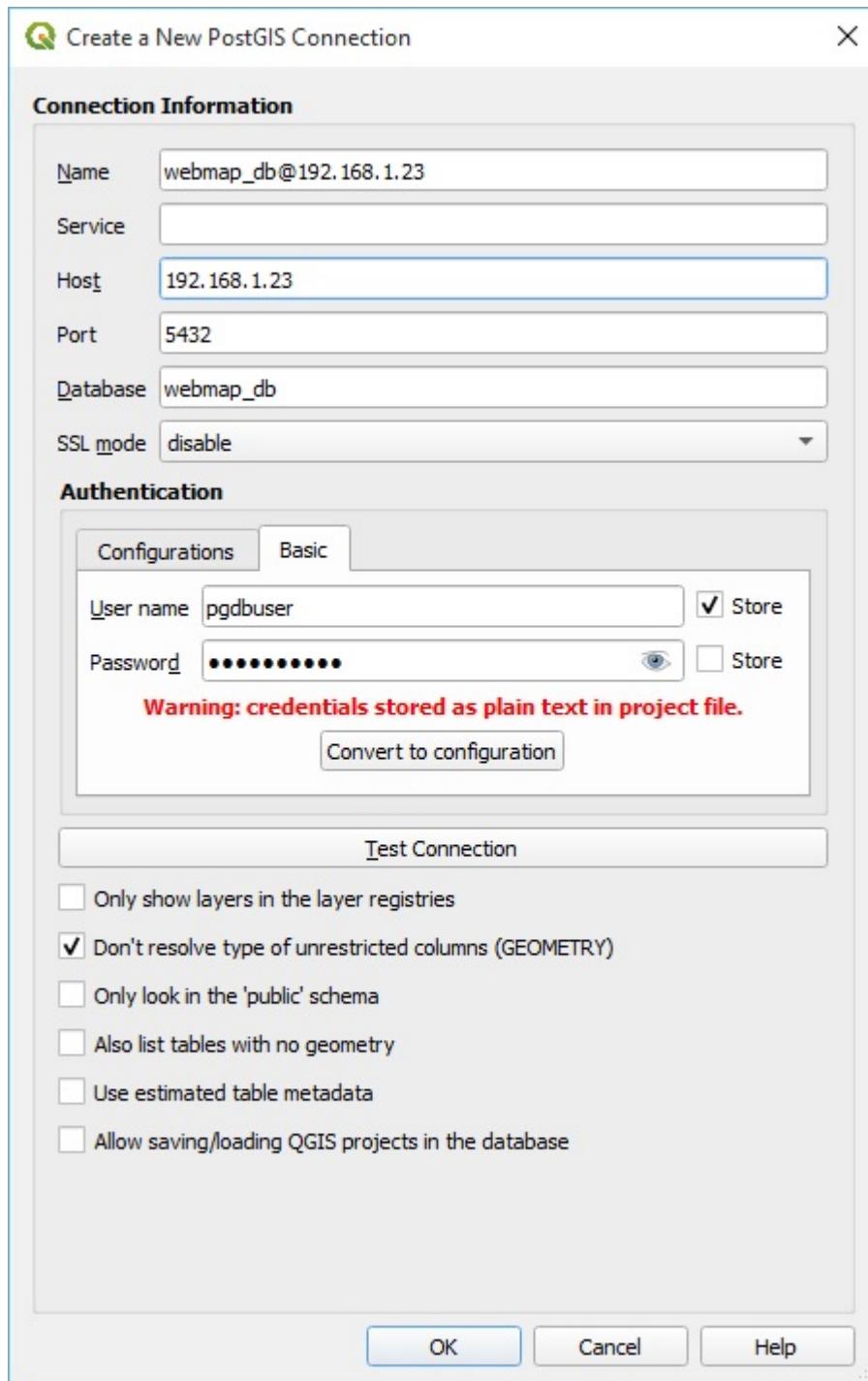
Pada bagian **Authentication**, klik tab **Basic**, dan isi **User name**: **pgdbuser**, *checkbox Store*-nya di-check, **Password** diisi dengan *password*-nya pgdbuser, dan *checkbox Store*-nya dibiarkan *unchecked* saja.

Berikutnya Anda bisa melakukan connection testing dengan meng-klik **Test connection**. Konfirmasi berhasil atau tidaknya koneksi akan muncul pada bagian atas dialog box ini.

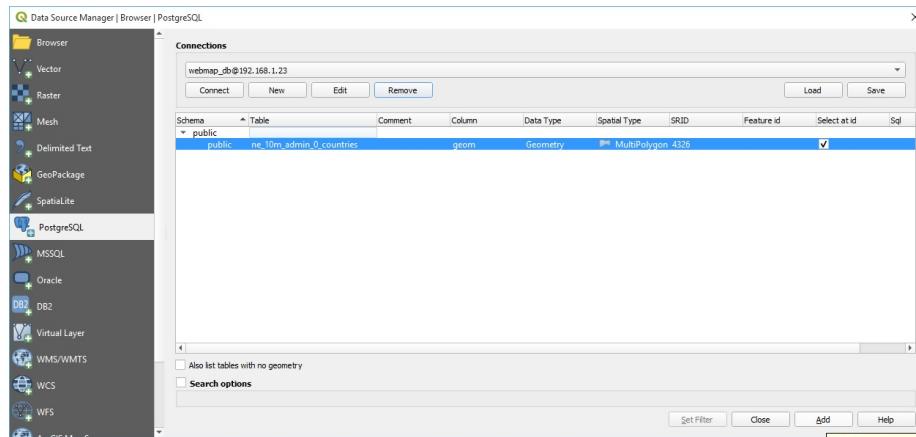
Jika Anda ingin hanya menampilkan *table* yang memiliki *geometry field* saja, *check* saja *checkbox* pada opsi **Don't resolve type of unrestricted columns (GEOMETRY)**.

Setelah Anda klik **OK**, maka *table* yang tadi sudah terbentuk saat kita meng-upload shapefile akan muncul sebagai pilihan *layer* yang akan ditampilkan.

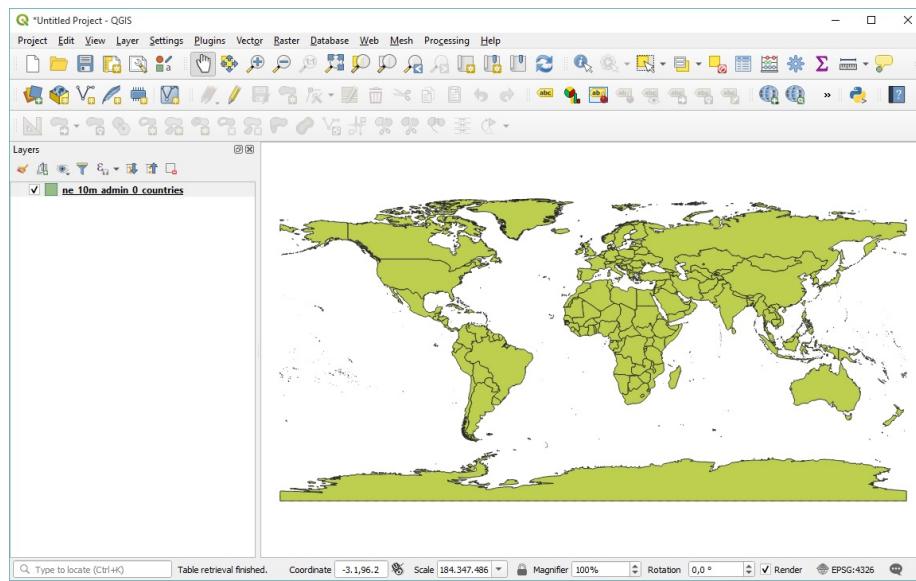
Klik (pilih) pada *table* tersebut, kemudian klik **Add** pada bagian bawah dan tunggu sejenak hingga tampilan *layer*-nya muncul di belakang dialog **Data Source Manager** ini. Selanjutnya klik **Close**.



Gambar 1.28: QGIS PostGIS Layer



Gambar 1.29: QGIS PostGIS Layer



Gambar 1.30: QGIS PostGIS Layer

Jika *layer ne_10m_admin_0_countries* sudah muncul, maka test PostGIS *layer* Anda sudah berhasil.

1.2.3 Download Data

Service Kemendagri

1.2.4 Impor Data

Bab 2

Studi Kasus : Pembangunan Basis Data untuk Analisis Wilayah Layanan

Topik

- membuat desain basis data relasional menggunakan diagram ER
- melakukan normalisasi dalam perancangan basis data
- membangun basis data spasial untuk analisis wilayah layanan (*service area analyses*) terkait fasilitas pendidikan dan kesehatan
- melakukan integrasi data spasial dan non-spasial

Perangkat

- pgModeler
 - Download (<https://pgmodeler.io/download>)
 - Source code (<https://github.com/pgmodeler/pgmodeler.git>)
- CASE tools (ArcGIS 10)

Studi Kasus

- Analisis Layanan Fasilitas Kesehatan

Data yang diperlukan :

1. **Fasilitas**
 - Rumah Sakit
 - Rumah Bersalin
 - Posyandu

44BAB 2. STUDI KASUS : PEMBANGUNAN BASIS DATA UNTUK ANALISIS WILAYAH LAYANAN

- Puskesmas

2. Batas Administrasi
3. Jalan
4. Jumlah penduduk
5. Kepadatan penduduk (WorldPop)
 - Analisis Layanan Fasilitas Pendidikan (SMA)

Data yang diperlukan:

1. SMA
2. Batas Administrasi
3. Jalan
4. Jumlah penduduk
5. Kepadatan penduduk (WorldPop)

2.1 Praktek Perancangan Basis Data

Intisari

1. Melakukan identifikasi entitas
2. Melakukan identifikasi atribut untuk setiap entitas
3. Menggambarkan hubungan antara entitas
4. Melakukan normalisasi tabel

Sumber data

Dapat dapat diperoleh dari :

- OpenStreetmap (<http://download.geofabrik.de/asia/indonesia.html>)
- Direktori Layanan ArcGIS Rest Service Kemendagri (<http://gis.dukcapil.kemendagri.go.id/arcgis/rest/services/>)

Bab 3

Basis Data Lanjut

Topik

3.1 Non-spatial SQL Queries

1. Membuat tabel baru

Tujuan : Membuat tabel baru dari tabel yang sudah ada dengan kolom terpilih

```
CREATE TABLE kelurahan as SELECT geom, giskemen_2, giskemen_4, giskemen18,giskemen19,giskemen20,g
```

2. Rename column

Tujuan : Mengganti nama kolom tertentu

```
ALTER TABLE public.kelurahan RENAME COLUMN giskemen_2 TO nama_desa;
```

3. Mengubah Primary Key

```
ALTER TABLE desa_batam DROP CONSTRAINT id;
```

```
ALTER TABLE desa_batam RENAME COLUMN giskemen_4 TO id;
```

```
ALTER TABLE desa_batam ADD PRIMARY KEY (id);
```

4. Menambah Kolom

```
ALTER TABLE public.kelurahan ADD COLUMN giskemen33 FROM public.desa WHERE giskemen_4.public.desa.
```

3.2 Spatial SQL Queries

3.2.1 Adjacent

1. Create New Layer

```
SELECT * into qlayer FROM kelurahan WHERE st_touches(kelurahan.geometry, ( SELECT geom
```

2. Create As View

```
CREATE VIEW TestView AS  SELECT * FROM kelurahan WHERE st_touches(kelurahan.geom, ( SELECT geom FROM kelurahan WHERE nama_desa = 'SADAI'));
```

3.3 Backup dan Restore

- **Presentasi :** <https://gitpitch.com/firmanhadi/lapan-okt-19/master#/>

- **Video Proof of Concept**

- Backup dengan pgBarman

https://www.dropbox.com/s/tk2g296cc0pv6dh/backup_poc_v1.mkv?dl=0

- Pooling Mechanism

https://www.dropbox.com/s/nl3xa2bnlay6ppk/pooling_mechanism_poc_v1.mkv?dl=0

https://www.dropbox.com/s/dtugpqi6k8hcsc3/pooling_mechanism_poc_v2.mkv?dl=0

- Failover Mechanism

https://www.dropbox.com/s/j5f9ks21db18ebv/replication_failover_poc_v1.mkv?dl=0

- **Software**

- <https://sqlbackupandftp.com/postgresql-backup>

Bagian II

Sistem Informasi Geografis

Bab 4

Pemodelan Sistem Informasi Geografis

Referensi

“Geographic Information System (GIS) modeling approach to determine the fastest delivery routes”: <https://www.sciencedirect.com/science/article/pii/S1319562X15001370>

4.1 Conceptual framework

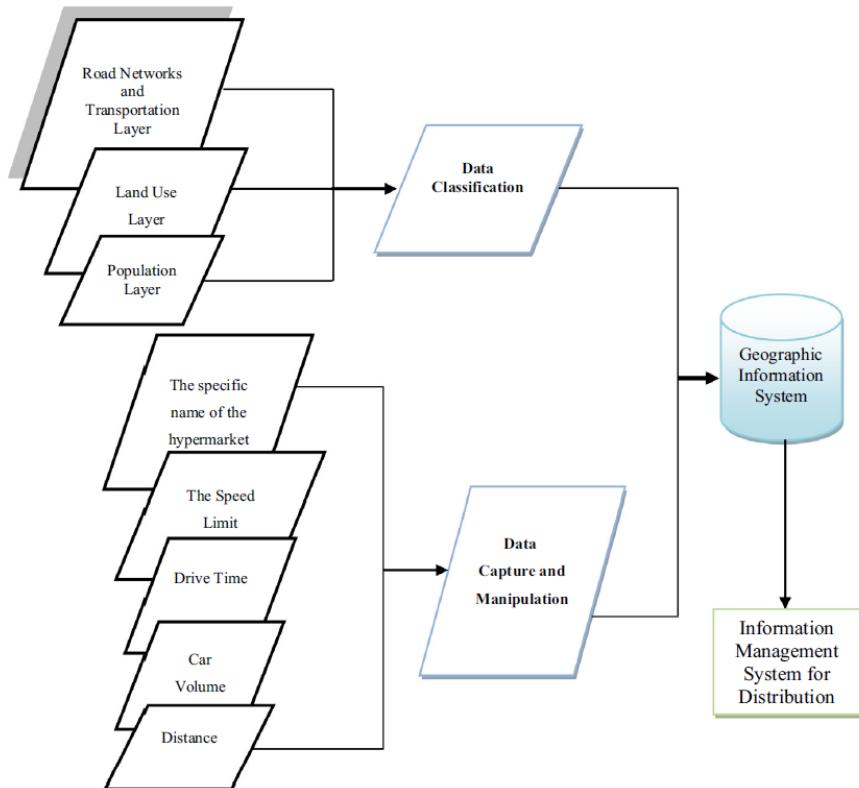


Figure 3 Conceptual framework.

4.2 Data

Table 1 Types of data for GIS analysis.

No.	Type of data	
	Spatial data (Feature)	Non-spatial data (Theme)
1.	Base Map ● Road Networks	Length of Drive Time includes: ● The specific travel distance between the certain places ● The average speed along the routes and also considering speed limit ● Drive time ● Car volume
2.	Land use Map ● Market location ● Residential Area ● Commercial Area ● Population ● School zone ● Industrial Area	The name of markets for the specific location

4.3 Final regression model

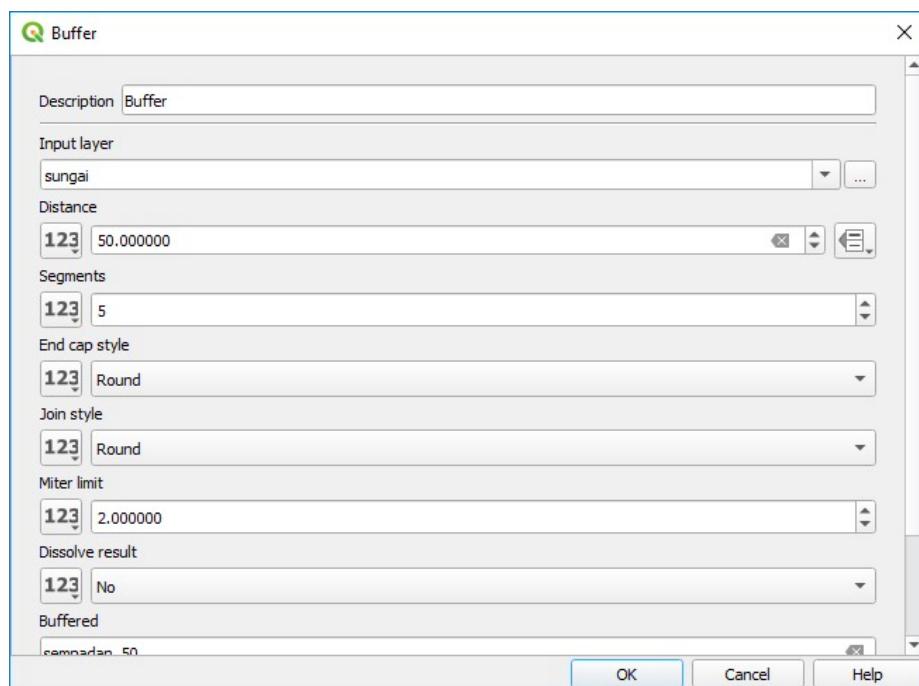
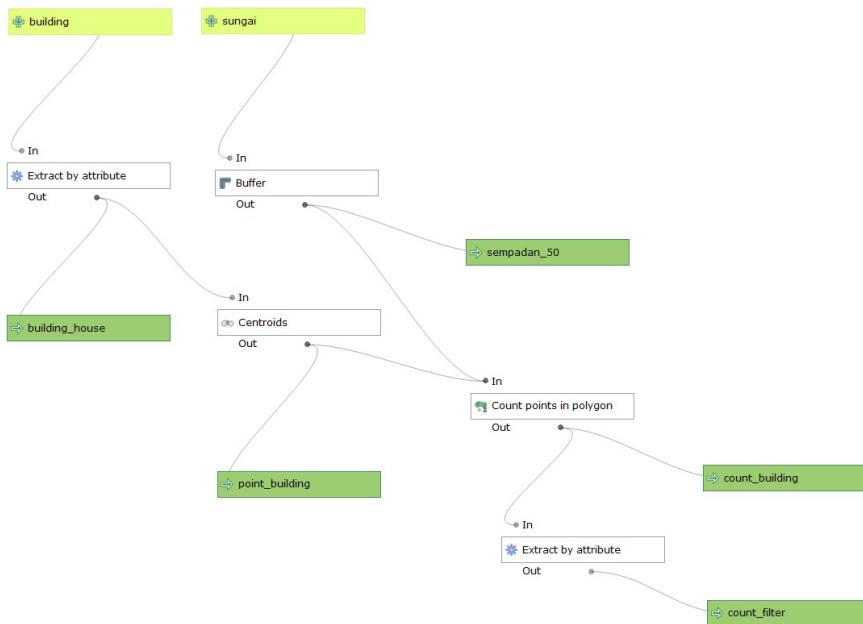
LTIME = +0.2663 * CARVOLUME + 0.6984 * LLENGTH + 0.0203 * LPOP +
0.0605 * TWOWAY + 0.1681 * SCHOOL + 0.0317 * RESIDENTIAL - 0.5497

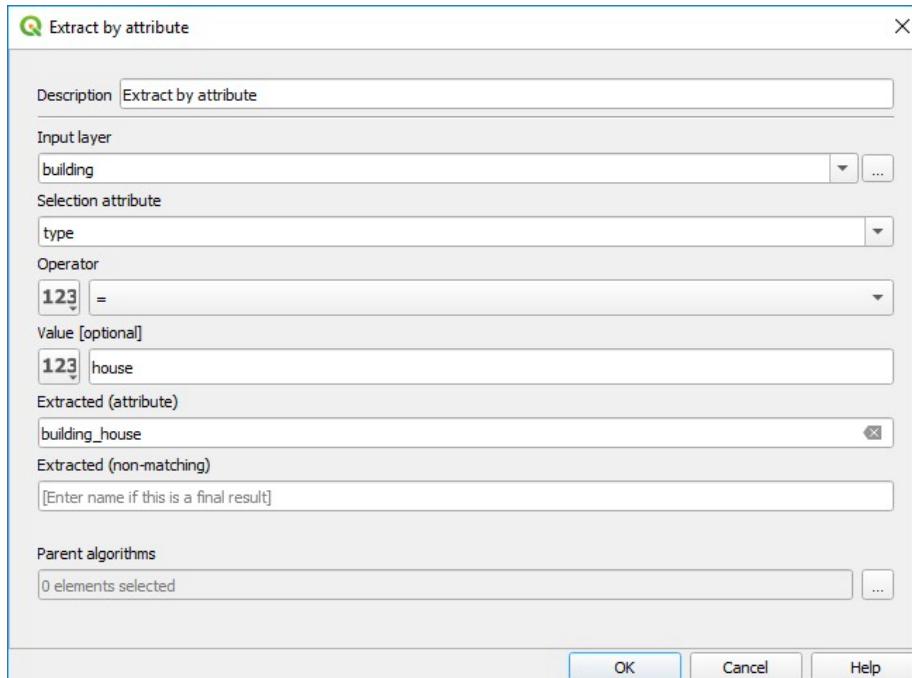
4.4 Studi Kasus : Pemodelan SIG

4.4.1 Analisis Wilayah Layanan (Service Area Analyses)

1. Load layer jalan dan fasilitas kesehatan dalam format UTM (Zona 48N).
2. Load raster kepadatan penduduk
3. Melakukan analisis *service area*
 - Processing > Network analyses > Service area
4. Create buffer from service area lines
5. Calculate raster using vector
 - Zonal statistics

4.4.2 Menghitung jumlah rumah di sempadan sungai





Bab 5

Analisis Sistem Informasi Geografis

Data

Data dapat diunduh di tautan berikut <https://firmanhadi.github.io/training-for-gis-analyses/img/Day6.zip>

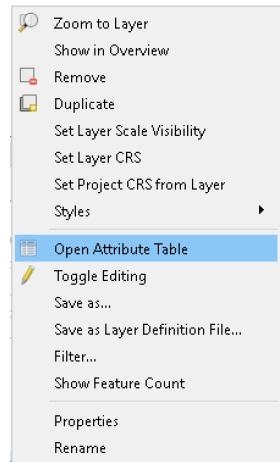
Analisis spasial adalah sebuah proses untuk mengkaji lokasi, atribut dan hubungan antara fitur dari data spasial melalui cara overlay dan teknik analisis lainnya, dalam rangka menjawab pertanyaan atau mendapatkan pemahaman yang bermanfaat. Analisis spasial mengekstrak atau membuat informasi baru dari data spasial.

5.1 Basic Geoprocessing

Geoprocessing adalah operasi SIG untuk memanipulasi data. Operasi geoprocessing membutuhkan input, melakukan operasi tertentu pada data tersebut dan memberikan hasil dari operasi dalam bentuk output dataset, seringkali disebut juga data turunan.

Operasi geoprocessing yang umum adalah overlay, feature selection dan analisis, pemrosesan topologi dan konversi data. Geoprocessing memungkinkan Anda untuk mendefinisikan, mengelola dan menganalisis informasi geografis yang digunakan untuk membuat keputusan.

Dengan kata lain, ekstraksi atau pengubahan informasi seperti yang Anda harapkan dari data selalu melibatkan geoprocessing.



Gambar 5.1: Open Attribute Table

5.1.1 Penapisan data

Kawasan lindung (taman nasional, suaga margasatwa dan hutan lindung) direncanakan dan dikelola dengan tujuan utama untuk konservasi biodiversitas. Hampir semua kawasan lindung terpapar interaksi dengan manusia, baik di dalam ataupun di luar kawasan. Hal ini akan berpengaruh terhadap hidupan liar dan habitatnya. Oleh karena itu, pertumbuhan populasi merupakan salah satu masalah utama dalam pengelolaan kawasan lindung. Hal tersebut akan memicu perubahan tutupan lahan dan penggunaan lahan, yang berdampak pada semakin tingginya tekanan terhadap kawasan lindung.

Walaupun efek kawasan lindung terhadap permukiman manusia masih menjadi perdebatan, ada sebuah kebutuhan pengelolaan interaksi tersebut, baik dalam hal positif ataupun negatif, yang menjadi vital dalam menjaga kelestarian layanan ekosistem. Dengan alasan ini, memetakan sebaran permukiman adalah salah hal yang dilakukan pertama kali dalam mengelola kawasan lindung.

Dalam pelatihan ini, permukiman direpresentasikan sebagai titik perkampungan dari OpenStreetMap yang akan diekstrak melalui query atau penapisan data. Caranya adalah sebagai berikut :

1. Buka osm_points.shp
2. Klik-kanan pada Layer dan pilih Open Attribute Table.
3. Klik tombol **Select features by expression**.
4. Pilih **place** untuk opsi **Field and Values**. Klik tombol **All unique** untuk melihat nilai yang ada di kolom Place. Ketik ekspresi “place” = ‘village’, klik tombol **Select features** di bagian bawah untuk melakukan

	id	natural	ele	name	name_en	place	tourism
1	369522431			Chè Nofi	Chè Nofi	town	
2	369522458			Ca Giun	Ca Giun	town	
3	1011648564			Ké Bang National Park			attraction
4	1586507267	cave_entrance		Bang Fai River Cave West			yes
5	1586509254	cave_entrance		Bang Fai River Cave East			
6	2159851039			Ban Nongchan		village	
7	2159868187			???????	Boualapha	village	
8	2929098411			Karia Pass			
9	3043121676			Ban Tha He		village	

Gambar 5.2: Select Features by expression button

penapisan data.

5. Baris yang terpilih akan berwarna biru (highlight).
5. Klik kanan pada Layer **osm_points** layer dan pilih “Save as”
6. Isikan pilihan seperti pada Gambar 5.6. Dan jangan lupa untuk memilih **Save only selected features**.

Untuk mengubah Coordinate Reference System (CRS), klik ikon Globe pada opsi CRS dan ketikkan 32648 dalam kotak Filter.

5.1.2 Dissolve

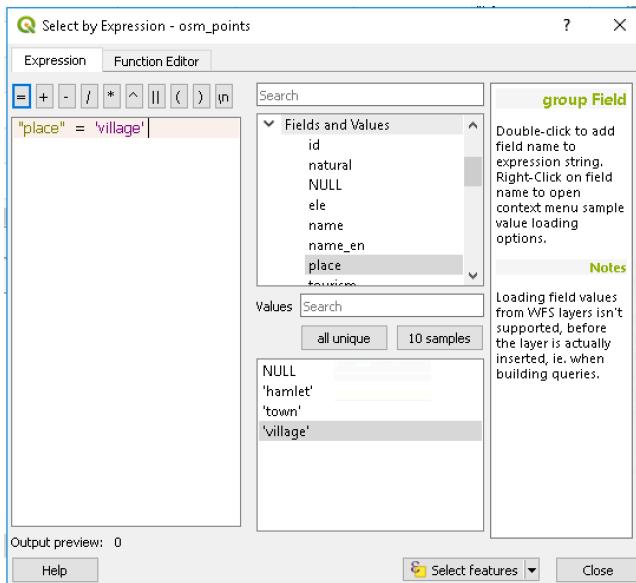
Satu atau lebih atribut dapat dipilih untuk menggabungkan (merge) geometri yang termasuk ke dalam kelas yang sama, atau semua geometri, dapat digabungkan.

Semua luaran (output) geometri akan dikonversi ke dalam bentuk multi geometri. Apabila inputnya adalah Layer poligon, common boundaries (batas bersama) dari poligon-poligon tetangga yang digabungkan, akan dihapus.

Untuk latihan ini, data yang akan digunakan adalah batas administrasi Laos. Batas ini terdiri dari tiga level (1) negara, (2) propinsi dan (3) distrik. Kita akan menggabungkan batas distrik ke tingkat propinsi.

Silakan ikuti langkah berikut untuk menggabungkan poligon berdasarkan atribut:

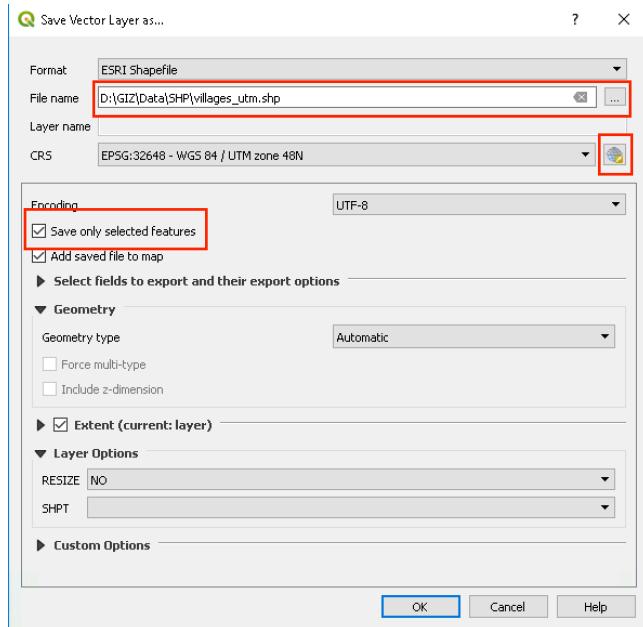
1. Buka gadm36_LA0_2.shp di Map Display.
2. Pilih **Vector -> GeoProcessing Tools -> Dissolve**
3. Klik ... pada **Unique ID Field**, pilih **NAME1** dan klik OK.



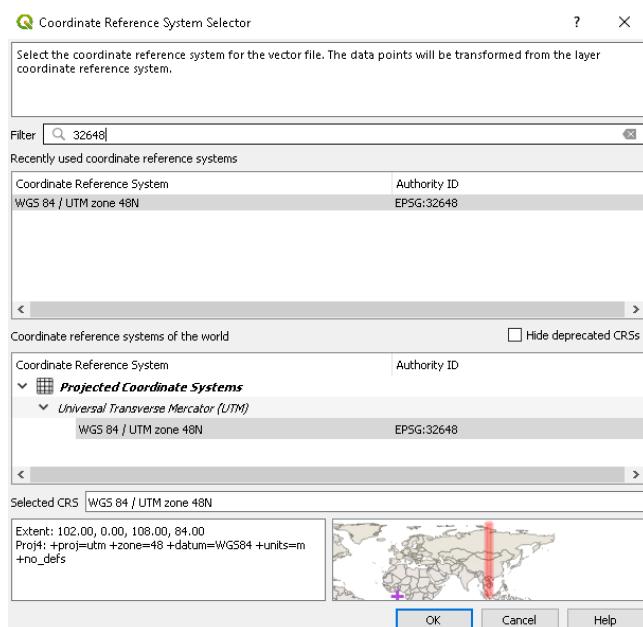
Gambar 5.3: Select by expression window

	id	natural	ele	name	name_en	place	tourism
1	369522431			Ch& N&i	Ch& N&i	town	
2	369522458			Ca Giun	Ca Giun	town	
3	1011648564				Ke Bang National Park		attraction
4	1586507267	cave_entrance		Bang Fai River Cave West			yes
5	1586509254	cave_entrance		Bang Fai River Cave East			
6	2159851039			Ban Nongchan		village	
7	21598568187			???????	Boualapha	village	
8	2929098411			Karia Pass			
9	3043121676			Ban Tha He		village	

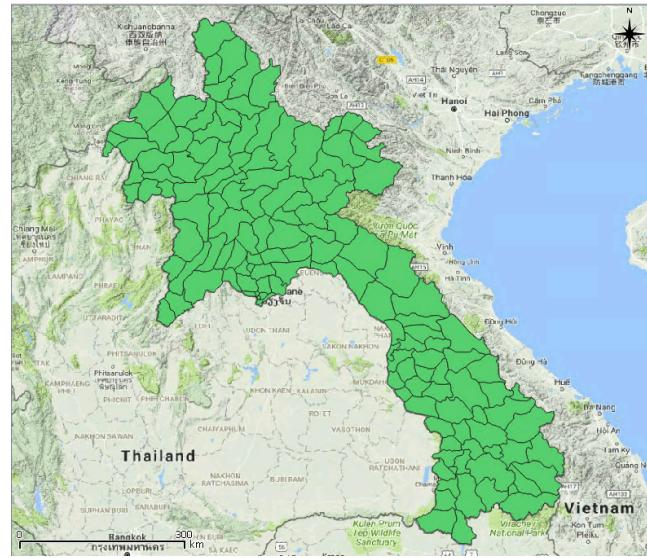
Gambar 5.4: Selected records



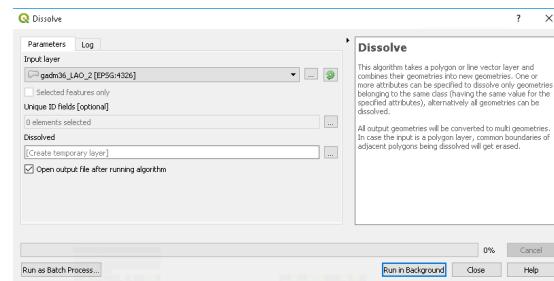
Gambar 5.5: Save Vector Layer As window



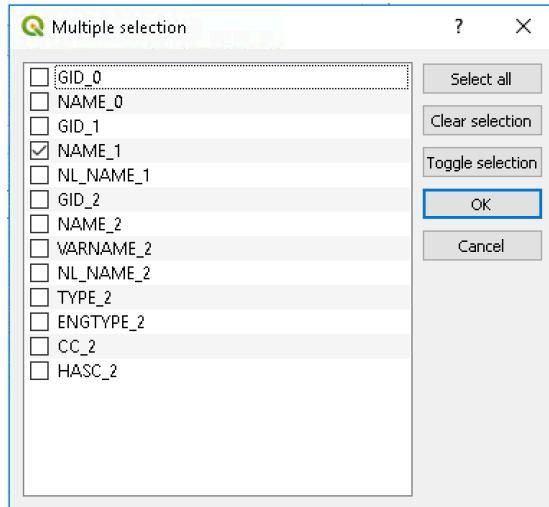
Gambar 5.6: Selecting new CRS



Gambar 5.7: Lao Admin Boundary Level 2



Gambar 5.8: Dissolve menu



Gambar 5.9: NAME1 selected as Unique ID

4. Pada menu Dissolve, klik **Run in the background** untuk menjalankan proses. Hasilnya akan ditampilkan di Map Display.

5.1.3 Polygon dari layer extent

Fungsi ini bermanfaat ketika kita ingin memotong raster atau vektor dengan menggunakan extent (bounding box) dari fitur tertentu.

Ikuti langkah berikut untuk melakukan ekstraksi batas dari poligon:

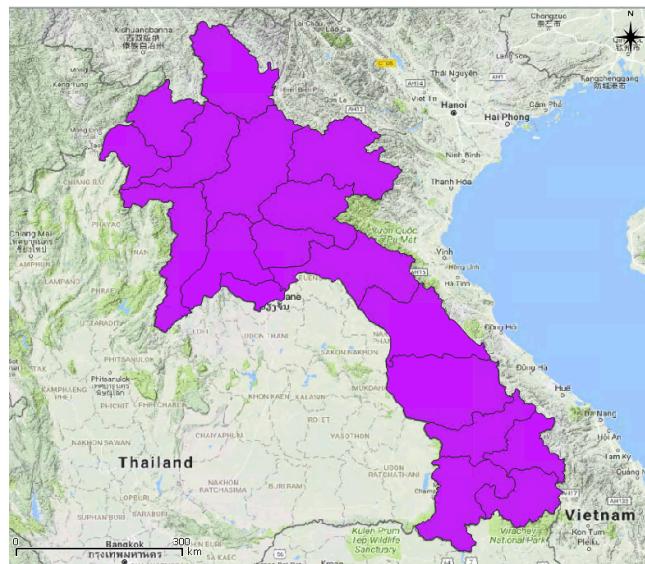
1. Pilih **Vector -> Research Tools -> Extract layer extent**
2. Anda dapat mengatur apakah luaran disimpan sebagai layer sementara atau disimpan ke dalam berkas baru.
3. Hasil dari proses ini adalah sebuah poligon kotak (persegi panjang) berdasarkan batas koordinat kiri-atas dan kanan-bawah dari fitur yang digunakan.

5.1.4 Reklasifikasi

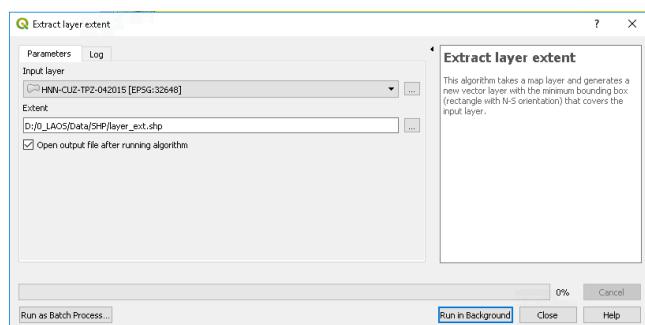
Fitur ini merupakan salah satu teknik yang bermanfaat untuk mengubah rentang nilai atau mengelompokkannya ke dalam kategori yang baru.

Kita akan melakukan klasifikasi ketinggian ke dalam tiga kelas:

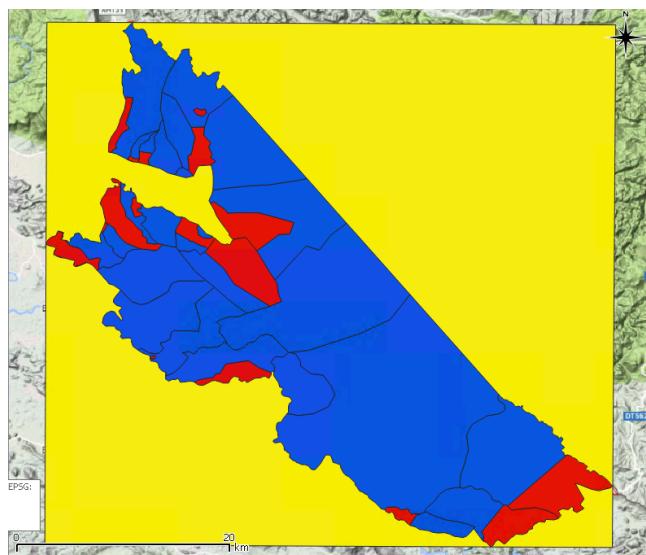
- Kurang dari 1.000 m



Gambar 5.10: Dissolved region



Gambar 5.11: Extract layer extent



Gambar 5.12: Layer extent created

- Antara 1.000 dan 2.000 m
- Lebih dari 2.000 m

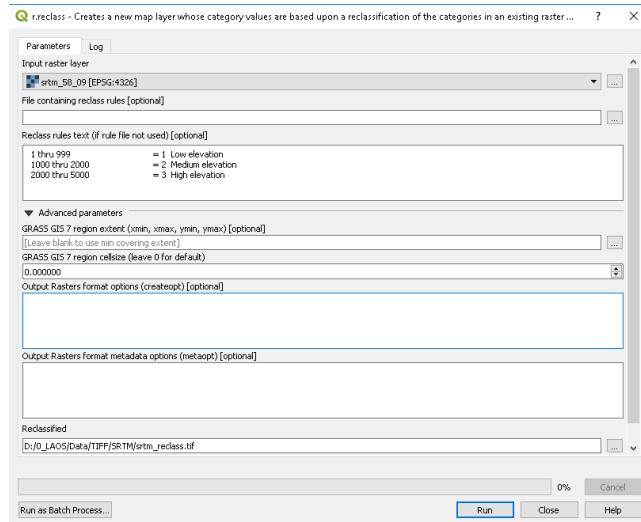
Untuk melakukan ini, silakan ikuti tahapan berikut :

1. Buka **srtm_58_09.tif**
2. Buka fungsi **r.reclass** dari menu **Processing Toolbox**. Isi pilihan seperti terlihat dalam gambar dan klik **Run**.
3. Hasilnya adalah layer dengan kelas baru

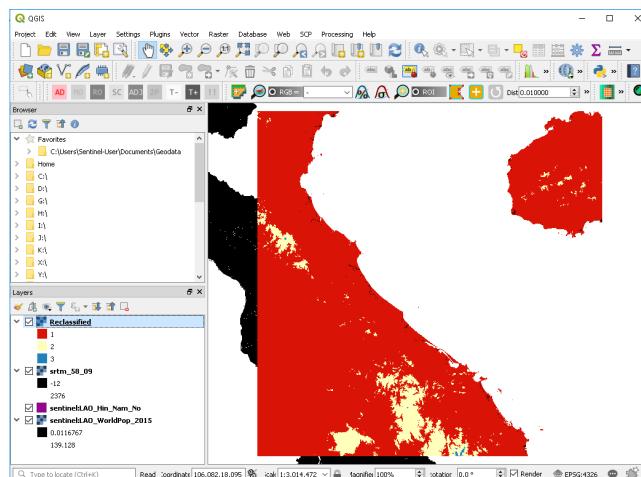
Nilai dari setiap piksel dibandingkan dengan rentang limit yang ada di lookup table. Apabila nilai piksel termasuk ke dalam kelas tertentu, nilai kelas untuk rentang ini akan digunakan di dalam layer luaran.

5.2 Terrain analyses

Tipe raster tertentu memungkinkan Anda untuk mendapatkan informasi yang lebih terkait terrain. Biasanya Digital Elevation Models (DEMs) digunakan untuk keperluan ini.



Gambar 5.13: Reclassifying the elevation data



Gambar 5.14: Reclassified elevation data

5.2.1 Persiapan

1. Buka **srtm_58_09.tif** (ada di dalam sub folder TIF). Layer ini merupakan DEM dengan EPSG:4326 CRS dan ketinggian dalam kaki (feet). Karakteristik ini tidak cocok untuk algoritma terrain analyses, harus dikonversi terlebih dahulu ke dalam proyeksi meter (Universal Transverse Mercator).
2. Reproyeksi layer ke sistem CRS EPSG:32648, menggunakan pilihan **Save as...** pada menu yang muncul ketika di-klik kanan pada nama layer.
3. Buka layer yang dihasilkan.

Ada hal yang perlu diperhatikan ketika menerapkan algoritma terrain analyses, agar hasilnya benar.

Salah satu permasalahan utamanya adalah apabila raster yang digunakan memiliki piksel dengan ukuran panjang dan lebar yang tidak sama (bukan bujur sangkar). Asumsi yang biasa digunakan adalah semua piksel pasti bujur sangkar. Namun seringkali data yang kita dapatkan tidak seperti itu.

Oleh karena itu, tahapan yang perlu dilakukan adalah mengekspor layer dan mendefinisikan ukuran piksel dengan nilai yang sama, misalnya 30 atau 90 m. Caranya adalah dengan klik kanan pada nama layer dan pilih **Save as ...**. Pada dialog **save** yang ada, pastikan untuk mengisikan nilai piksel di bagian bawah dialog :

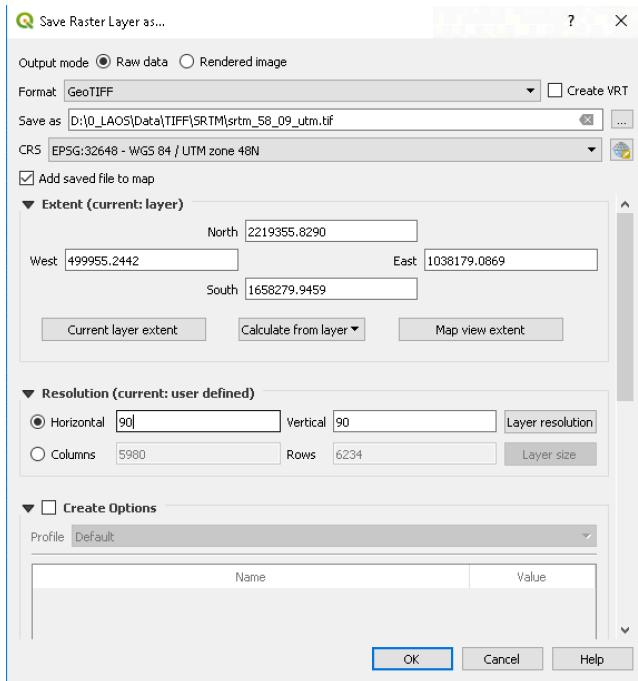
5.2.2 Kelerengan

Kelerengan (**Slope**) merupakan salah satu parameter dasar yang dapat diturunkan dari DEM. Ia adalah turunan pertama dari DEM dan menggambarkan laju perubahan ketinggian. **Slope** dihitung dengan melakukan analisis ketinggian dari setiap piksel, membandingkannya dengan ketinggian piksel di sekitarnya. Untuk menghitungnya di QGIS, silakan ikuti cara berikut :

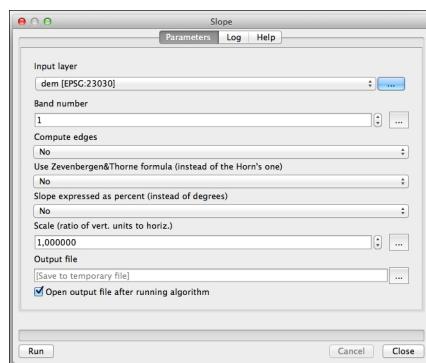
1. Pada opsi **Processing Toolbox**, pilih algoritma **Slope**, klik ganda untuk membukanya.
2. Pilih **DEM** sebagai layer masukan (input).
3. Klik **Run** untuk menjalankan prosesnya.

5.2.3 Hillshade

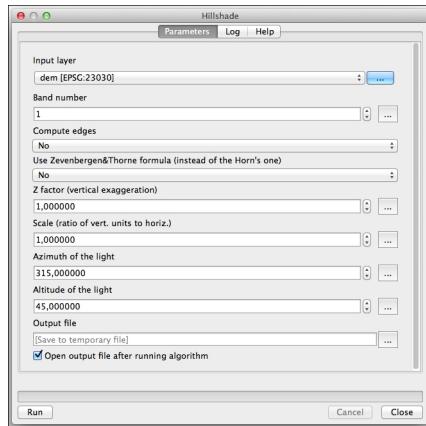
Layer **hillshade** umumnya digunakan untuk memperbaiki tampilan peta dan topografi yang intuitif, dengan mensimulasikan sumber cahaya dan bayangan yang dibentuk oleh permukaan bumi. **Hillshade** dapat dihitung dengan cara sebagai berikut :



Gambar 5.15: Save raster layer as



Gambar 5.16: Calculating slope



Gambar 5.17: Calculating hillshade

1. Pada opsi **Processing Toolbox**, cari algoritma **Hillshade** dan klik-ganda untuk membuka menunya.
2. Pilih DEM sebagai **Input layer**, gunakan parameter default.
3. Klik **Run** untuk menjalankan algoritmanya.

5.3 Density Analyses (Analisis Kepadatan)

Seringkali kita harus bekerja dengan data besar dan padat fiturnya, yang membuat penampilan datanya lambat. Jumlahnya yang ribuan atau bahkan jutaan fitur seringkali sulit diinterpretasi, terjadi overlap antar fitur yang menyebabkan pendekripsi pola cluster atau distribusinya tidak mudah untuk dilakukan.

Pada bagian ini, kita akan belajar mengenai teknik yang memungkinkan visualisasi dataset semacam itu dengan bentuk yang lebih mudah dibaca dan dengan waktu pemuatan (**loading**) yang lebih cepat. Setelah melakukan praktik di bagian ini, Anda diharapkan mampu melakukan analisis kepadatan (**density analyses**) untuk data yang Anda miliki dan mengekstrak informasi dari peta kepadatan.

5.3.1 Konsep

Peta kepadatan memungkinkan estimasi visual konsentrasi objek atau peristiwa di area studi. Peta seperti itu sangat berguna untuk penilaian pola distribusi fitur di wilayah studi. Ketika kita cukup menambahkan lokasi fitur atau peristiwa (misalnya, sebagai titik) ke peta, kita tidak dapat melihat perubahan konsentrasi mereka di area yang berbeda. Analisis kepadatan memberi kita

fungsionalitas seperti itu dengan menggunakan karakteristik area yang seragam, seperti jumlah fitur per hektar atau kilometer persegi.

Peta kepadatan memberi kita kemampuan untuk memperkirakan konsentrasi beberapa fitur dalam suatu area. Ini membantu kita menemukan area di mana reaksi mendesak diperlukan atau yang cocok dengan kriteria. **Heatmap** juga membantu mengontrol kondisi dan perubahannya.

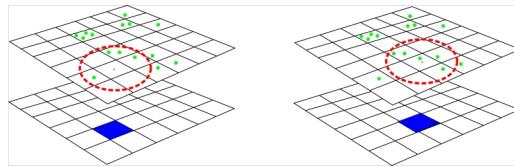
Peta kepadatan juga sangat berguna ketika wilayah yang dipetakan (misalnya, kabupaten) memiliki ukuran yang berbeda. Misalnya, jika kita ingin tahu berapa banyak orang yang tinggal di setiap kabupaten, kita hanya perlu peta ordinal dengan data populasi. Menurut peta ini, sebuah distrik besar mungkin memiliki populasi yang lebih tinggi daripada distrik yang lebih kecil. Tetapi jika kita ingin mengidentifikasi kabupaten dengan konsentrasi populasi yang lebih tinggi, maka kita membutuhkan peta kepadatan untuk melihat jumlah orang per kilometer persegi. Dan peta kepadatan akan menunjukkan kepada kita bahwa, pada kenyataannya, daerah kecil dengan kepadatan penduduk yang tinggi mungkin memiliki lebih banyak orang per kilometer persegi daripada kabupaten yang lebih besar.

Secara umum, kita dapat menunjukkan pada peta distribusi kepadatan fitur itu sendiri (misalnya, sekolah), serta distribusi beberapa karakteristik numerik fitur ini (misalnya, jumlah siswa di sekolah). Hasilnya akan sangat berbeda dalam kasus ini. Peta kepadatan sekolah dapat membantu departemen pendidikan menemukan daerah-daerah di mana lebih banyak sekolah dibutuhkan, sementara peta kepadatan dibuat dari informasi tentang jumlah siswa di setiap sekolah dapat membantu perusahaan transportasi untuk merencanakan rute bus dan untuk memutuskan di mana menempatkan halte bus. .

Kasus penggunaan yang paling umum adalah pembuatan peta kepadatan untuk menampilkan kerapatan fitur titik. Peta seperti ini sering disebut **heat map**. Apa itu **heat map**? Ini adalah layer raster. Setiap pikselnya menggambarkan kepadatan fitur di sekitarnya (misalnya, jumlah orang per kilometer persegi), yang tergantung pada jumlah fitur dalam beberapa area.

Untuk membuat **heat map**, dalam kasus paling sederhana, GIS melihat fitur di sekitar pusat piksel, menggunakan radius pencarian yang diberikan. Kemudian jumlah fitur yang termasuk dalam radius yang diberikan dihitung dan dibagi dengan luas wilayah. Nilai ini akan menjadi nilai piksel. Kemudian piksel selanjutnya akan dianalisis, dan seterusnya. Sebagai hasilnya, kita akan mendapatkan kombinasi nilai, yang menciptakan permukaan yang halus. Untuk memahami ini lebih baik, lihat diagram berikut:

Diagram ini menunjukkan prinsip umum pembuatan **heat map**. Titik-titik hijau menggambarkan fitur yang digunakan untuk pembuatan peta kepadatan, kotak biru adalah sel raster saat ini, dan lingkaran titik-titik merah menandai radius pencarian, misalnya, 1 km. Dalam hal ini, area yang dicakupi adalah sekitar 3,14 km persegi. Seperti yang dapat kita lihat pada diagram di sebelah kiri, empat fitur berada dalam radius pencarian. Jadi, piksel raster akan



Gambar 5.18: General principle of creating heatmaps

mendapatkan nilai $4 / 3,14 = 1,27$. Di sisi kanan, kita perhatikan bahwa piksel berikutnya akan mendapatkan nilai 1,59 karena sekarang ada lima fitur di dalam radius pencarian.

Ini adalah pendekatan paling sederhana. Dalam aplikasi dunia nyata, algoritma yang lebih kompleks digunakan, di mana setiap titik memiliki dampak pada nilai-nilai piksel tetangga, tergantung pada jaraknya dari piksel-piksel tersebut.

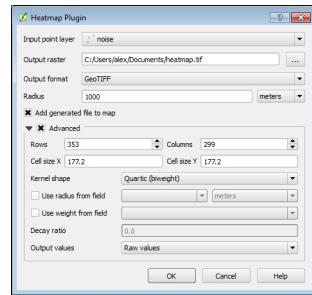
5.3.2 Membuat Heatmap dengan plugin QGIS

Dengan bantuan plugin inti QGIS yang disebut **Heatmap**, kita dapat dengan mudah membuat **heat map** dari data titik vektor dan menggunakannya untuk analisis lebih lanjut. Pertama, kita perlu mengaktifkan plugin ini, jika belum diaktifkan. Setelah aktivasi, ia membuat submenu di bawah menu Raster dan menempatkan tombol pada bilah alat Raster .

Mari kita buat peta kepadatan untuk lapisan kebisingan , yang berisi informasi tentang keluhan tentang tingkat kebisingan yang tinggi. Lapisan ini berisi 44.397 fitur, dan sulit untuk mengetahui tempat mana yang berisik.

Informasi tentang tempat-tempat seperti itu mungkin berguna bagi departemen kepolisian atau lembaga lain untuk merencanakan beberapa kegiatan untuk mengurangi kebisingan, atau bagi mereka yang mencari apartemen dan tidak ingin hidup berdampingan dengan tetangga yang hobinya mendengarkan musik cadas! .

1. Mulai plugin dengan klik tombol **Heatmap** pada panel **Raster**, atau dengan melakukan navigasi **Raster | Heatmap | Heatmap....**
2. Pilih layer **noise** dari kotak centang **Input point layer**.
3. Dengan memilih tombol ... di bagian kanan dari kolom **Output raster**, tentukan direktori di mana Anda akan menyimpan peta **heat map**. Catatan, Anda tidak perlu menuliskan ekstensi berkasnya, ia akan dipilih secara otomatis.
4. Gunakan kotak centang **Output format** untuk memilih format data yang diinginkan. Pilihan paling umum adalah **GeoTIFF**, namun untuk peta yang mencakup area luas, lebih baik gunakan format yang lain, misalnya **Erdas Imagine**.



Gambar 5.19: General principle of creating heatmaps

5. Yang terakhir harus ditentukan adalah **Radius**. Nilai ini menentukan jarak dari setiap piksel di mana QGIS akan mencari fitur tetangganya dan mengikutsertakan keberadaan piksel tetangga ke dalam perhitungan. Secara umum, radius yang lebih besar akan memberikan hasil yang lebih umum (tergeneralisasi), karena jumlah fitur akan dibagi dengan area yang lebih besar. Radius lebih kecil akan memberikan hasil yang lebih presisi, namun apabila terlalu kecil maka kita tidak akan dapat melihat pola distribusinya. Radius pencarian dapat didefinisikan dalam meter atau unit peta.

Untuk menentukan radius pencarian dari wilayah tertentu, kita dapat menggunakan formula sederhana berikut yang diperoleh dari formula untuk luas lingkaran:

$$r = \sqrt{\frac{S}{\pi}}$$

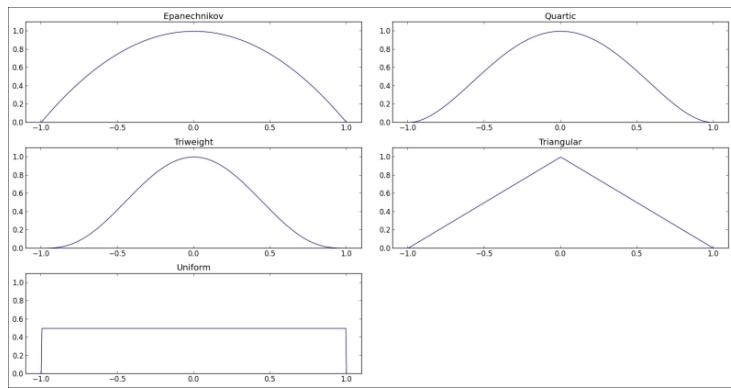
Sebagai contoh, apabila kita ingin menghitung kepadatan per kilo meter persegi, maka radius pencarinya adalah sebagai berikut:

$$r = \sqrt{\frac{1 \text{ km}^2}{\pi}} = \sqrt{\frac{1000000 \text{ m}^2}{3.1415926}} \approx 564.2 \text{ m}$$

Untuk mengatur hasil yang lebih baik, kita dapat memilih kotak **Advanced** dan menentukan beberapa parameter tambahan:

- **Rows and Columns:**

Ini memungkinkan kita untuk mendefinisikan dimensi raster luaran. Dimensi yang lebih besar akan menghasilkan ukuran berkas luaran yang lebih besar, sedangkan dimensi yang lebih kecil akan menghasilkan luaran yang kasar dan kotak-kotak (*pixelated*). Kolom input ditautkan satu sama lain, sehingga mengubah nilai dalam bidang baris (misalnya, membagi dua) juga akan menyebabkan



Gambar 5.20: Distribution of the point influence for different kernel

perubahan yang sesuai dengan nilai di bidang kolom, dan sebaliknya. Selanjutnya, nilai-nilai ini memiliki pengaruh langsung pada ukuran piksel (lihat poin berikutnya). Perlu ditekankan bahwa luas raster dipertahankan saat mengubah dimensi raster.

- **Ukuran piksel X dan Y:**

Ukuran piksel raster menentukan seberapa kasar atau terperinci tampilan pola distribusi. Ukuran piksel yang lebih kecil akan memberikan hasil yang lebih halus, tetapi waktu pemrosesan dan memori yang diperlukan untuk analisis akan meningkat. Sel besar akan diproses lebih cepat, tetapi raster yang dihasilkan akan *pixelated*. Jika piksel-pikselnya sangat besar, beberapa pola akan menjadi tidak terlihat, jadi Anda mungkin perlu menjalankan analisis beberapa kali, mencoba ukuran sel yang berbeda untuk mendapatkan hasil yang memenuhi kebutuhan Anda.

Ukuran piksel tergantung pada dan terkait dengan dimensi raster. Menambahnya akan mengurangi jumlah baris dan kolom, dan sebaliknya.

- **Kernel shape:**

Ini mengontrol bagaimana titik mempengaruhi perubahan dengan perubahan jarak dari titik ini. **Heatmap plugin** QGIS saat ini mendukung kernel seperti berikut:

- quartic (dikenal juga dengan biweight)
- triangular
- uniform
- triweight
- Epanechnikov

Bergantung pada bentuk kernel, kita akan mendapatkan **heat map** yang lebih halus, atau hotspot yang lebih jelas. Misalnya, kernel triweight akan memberikan hotspot yang lebih jelas, lebih tajam daripada kernel Epanechnikov, karena kernel Epanechnikov memiliki pengaruh lebih rendah di dekat pusat hotspot. Juga, dalam bidang ilmiah yang berbeda, kernel yang berbeda lebih disukai; misalnya, dalam analisis kejahatan, kernel kuartik biasanya digunakan.

Dimungkinkan juga untuk menggunakan jari-jari pencarian variabel untuk setiap titik dengan memilih kotak centang **Use radius from field** dan memilih bidang atribut dengan nilai jari-jari dari kotak centang. Jika Anda perlu membobot poin (dengan kata lain, menambah atau mengurangi pengaruhnya) dengan beberapa atribut numerik, aktifkan kotak centang **Use weight from field** dan pilih bobot yang sesuai. Dalam contoh kita, kita tidak akan menggunakan fungsi ini, tetapi Anda dapat mencobanya sendiri.

Seperti yang dijelaskan sebelumnya, ukuran piksel memiliki pengaruh langsung pada kualitas **heat map** yang dihasilkan, jadi penting untuk memilihnya dengan hati-hati. Dalam kebanyakan kasus, ukuran sel dipilih sedemikian rupa sehingga kita mendapatkan 10 hingga 100 sel per unit area (yang pada gilirannya ditentukan oleh radius pencarian). Untuk menghitung ukuran piksel, kita perlu menyelaraskan unit area dengan unit jarak; misalnya, jika kita menghitung kepadatan menggunakan kilometer persegi dan menentukan radius pencarian dalam meter, maka perlu untuk mengubah kilometer persegi menjadi meter persegi. Langkah selanjutnya adalah membagi area dengan jumlah sel yang diinginkan. Akhirnya, karena ukuran piksel ditentukan oleh lebar atau tingginya (karena sel raster biasanya memiliki bentuk persegi), kita perlu mengekstrak akar kuadrat dari nilai ini.

Dalam contoh kita, kita akan membuat **heat map** dengan radius pencarian 1000 m, sehingga area pencarian akan sekitar 3,14 kilometer persegi. Saat dinyatakan dalam meter, ini akan menjadi sebagai berikut:

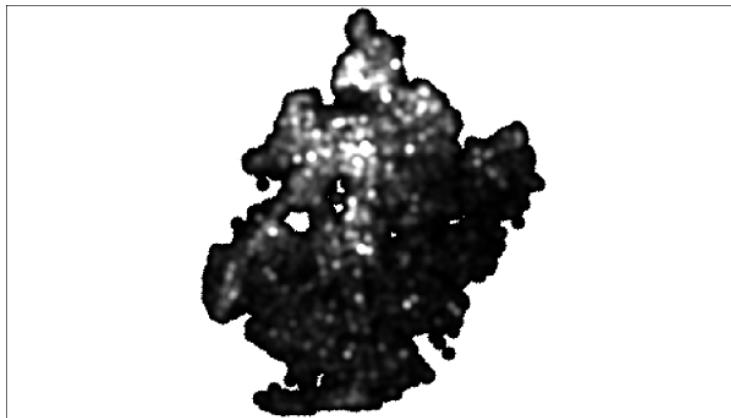
$$3.14km^2 = 3.14 \cdot 1000m \cdot 1000m = 3140000m^2$$

Karena kita ingin **heat map** yang halus, kita akan menggunakan jumlah piksel yang relatif besar per satuan luas; katakanlah 100 piksel per 3,14 kilometer persegi. Jadi, kita membagi area dalam meter persegi dengan jumlah piksel yang diinginkan:

$$\frac{3140000m^2}{100cells} = 314000m^2 per cell$$

Akhirnya, kita menghitung akar kuadrat dari nilai ini untuk mendapatkan ukuran yang memungkinkan kita memiliki 100 piksel per 3,14 kilometer persegi:

$$\sqrt{31400m^2} \approx 177.2m$$



Gambar 5.21: Heatmap result in greyscale

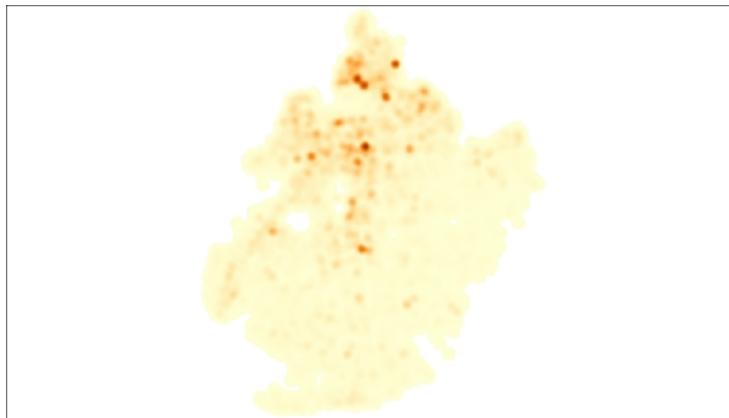
Tentu saja ini bukan aturan yang kaku tapi hanya rekomendasi. Anda dapat menggunakan ukuran piksel lain dengan aman, tergantung datadan hasil yang Anda inginkan. Hanya ingat bahwa nilai yang lebih kecil mengarah pada **heat map** yang lebih halus, tetapi pada saat yang sama meningkatkan waktu analisis dan menghasilkan ukuran raster yang lebih besar.

Ketika semua input dan parameter diatur, tekan tombol OK untuk memulai proses pembuatan *peta panas **heat map**. Proses pembentukan **heat map** akan ditampilkan dalam dialog progres kecil. Jika proses ini terlalu lama untuk diselesaikan, Anda dapat menghentikannya dengan menekan tombol **Cancel**. Perhatikan bahwa setelah membatalkan pembuatan **heat map**, Anda masih mendapatkan hasilnya, tetapi hasilnya tidak lengkap dan tidak berguna untuk analisis lebih lanjut.

Ketika proses selesai, **heat map** yang dihasilkan akan ditambahkan ke QGIS sebagai raster grayscale, di mana wilayah yang lebih terang menggambarkan dengan nilai kepadatan yang lebih tinggi dan daerah yang lebih gelap berarti memiliki nilai kepadatan yang lebih rendah, seperti ini:

Untuk meningkatkan keterbacaan dan membuatnya terlihat seperti **heat map** nyata, kita perlu mengubah gayanya. Untuk melakukan ini, ikuti langkah selanjutnya.

1. Klik kanan pada layer **heatmap**. Pada menu, pilih **Properties**.
2. Buka tab **Style** dan pilih **Singleband pseudocolor** untuk **Render type**.
3. Pada grup **Load min/max values**, aktifkan opsi **Min/max**. Pilih **Extent to Full** dan **Accuracy to Actual (slower)**. Tekan tombol **Load** untuk mendapatkan statistik layer. Hasilnya akan digunakan untuk klasifikasi nilai.



Gambar 5.22: Heatmap result pseudocolor

4. Pilih pola warna (color ramp) pada grup **Generate new color map**, sebagai contoh, **YlOrBr** (di mana warna berubah dari kuning ke jingga dan kemudian coklat), atau **Red** (yang menggunakan gradasi warna merah). Jika dirasa perlu, ubah jumlah kelas dan tekan tombol **Classify**.
5. Klik **OK** untuk menerapkan perubahan dan menutup dialog properties.

Sekarang kita dapat dengan mudah menemukan titik terpanas (ditampilkan dalam warna lebih dekat ke merah jika peta warna Merah digunakan), dan bahkan mengenali beberapa pola distribusi yang tidak terlihat ketika kita melihat lapisan titik asli. Juga, lapisan **heat map** kita muncul jauh lebih cepat daripada vektor yang digunakan untuk membuat peta panas ini.

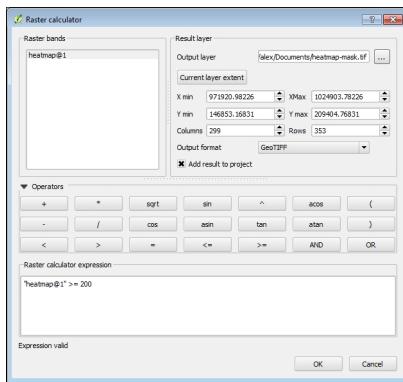
Mendeteksi wilayah terpanas (“hottest”)

Terkadang, Anda tidak perlu **heat map** itu sendiri, tetapi hanya ingin menemukan hotspot — area dengan kepadatan tertinggi — dan menggunakannya dalam analisis lebih lanjut. Sangat mudah untuk menemukan wilayah seperti itu di QGIS dan mengekstraknya dalam bentuk vektor.

Pertama, kita harus mendefinisikan nilai ambang, yang akan digunakan untuk mengenali hotspot. Sebagai nilai awal, kita dapat menggunakan nilai piksel maksimum dalam **heat map** kita dan kemudian menyesuaikannya dengan kebutuhan kita.

Cara paling sederhana untuk menemukan nilai piksel maksimum adalah dengan menggunakan alat Identify Features . Pilih satu layer di pohon layer QGIS, aktifkan alat Identify Features , klik pada daerah yang paling “terpanas” secara visual, dan lihat nilai yang dilaporkan. Dengan peta panas kita, ini akan menjadi 540,32.

Jika kita akan menggunakan nilai ini sebagaimana adanya, kita tidak dapat me-



Gambar 5.23: Heatmap result pseudocolor

nemukan semua cluster penting, jadi nilai ini harus dikurangi terlebih dahulu. Semakin kecil nilai yang dipilih (dibandingkan dengan nilai maksimum), semakin besar jumlah cluster yang ditemukan. Luas cluster yang terpisah juga akan tumbuh. Sebagai contoh kita, kita memilih nilai 200.

Sekarang, buka **Raster Calculator** dari menu **Raster**, tentukan path di mana berkas akan disimpan dalam kolom **Output layer**, dan masukkan formula “**heatmap@1”>=200** pada kolom **Raster calculator expression**, seperti berikut:

Formula ini digunakan untuk membuat apa yang disebut dengan **mask** (topeng). Jika nilai piksel dari lapisan input lebih besar atau sama dengan nilai ambang batas kita 200, maka nilai piksel output akan menjadi 1. Jika tidak, itu akan menjadi 0. Jadi, raster luaran kita akan menjadi raster biner, dengan hanya dua piksel nilai — 0 dan 1 — yang sangat mudah dikonversi menjadi vektor.

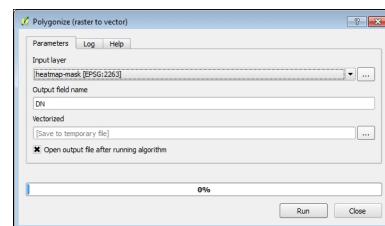
Biarkan semua nilai lainnya tetap tidak berubah, sehingga raster yang dihasilkan akan memiliki dimensi dan ukuran sel yang persis sama dengan yang dimasukkan. Tekan tombol **OK** untuk memulai perhitungan. Ketika selesai, layer raster hitam-putih baru akan ditambahkan ke kanvas QGIS, seperti yang ditunjukkan di sini:

Untuk mengonversi topeng raster ke dalam format vektor, kita perlu membuat poligon dari semua piksel yang terhubung dengan nilai yang sama. Di sinilah tool **Polygonize** amat membantu. Di kotak alat **Processing Toolbox**, Anda dapat menemukan algoritma **Polygonize** dengan mengetik namanya di bidang filter di bagian atas kotak alat. Klik dua kali pada nama algoritma untuk membuka dialognya, dan Anda akan melihat sesuatu seperti ini:

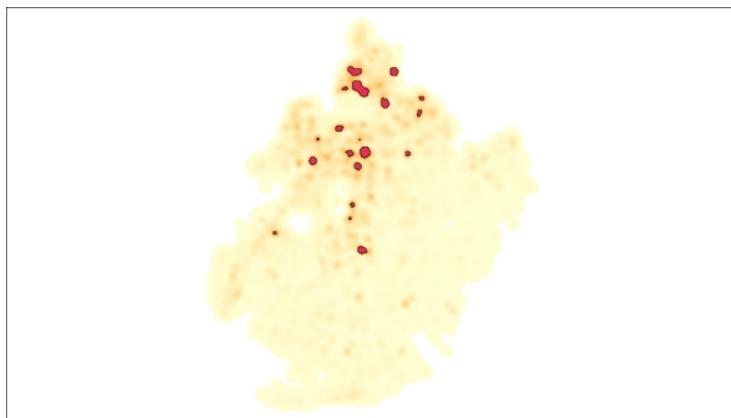
Pilih layer mask yang sebelumnya dibuat sebagai layer Input , tentukan path tempat hasilnya akan disimpan menggunakan bidang layer Output , dan klik tombol **Run** untuk memulai algoritma. Setelah selesai, layer vektor baru akan ditambahkan ke QGIS. Lapisan ini memiliki atribut yang disebut DN (jika



Gambar 5.24: Mask for heatmap



Gambar 5.25: Polygonized mask



Gambar 5.26: Cleaned hotspot polygon

Anda tidak mengubahnya) yang menunjukkan nilai piksel setiap poligon dalam lapisan. Jadi, yang perlu kita lakukan adalah menghapus semua fitur yang memiliki nilai atribut sama dengan nol. Fitur yang tersisa adalah hotspot.

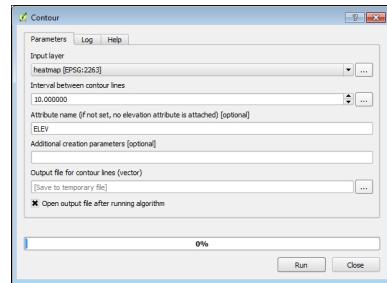
Untuk menghapus fitur yang tidak perlu dari lapisan hotspot, pilih di **Layer Tree QGIS**, klik kanan untuk membuka menu konteks, dan pilih **Open Attribute**. Klik pada tombol **Select features using an expression**. Dalam dialog **Select by expression**, masukkan “DN” = 0 (jika perlu, ganti DN dengan nama kolom tertentu), klik tombol **Select**, dan tutup dialog. Mulai mengedit dengan mengklik tombol **Toggle editing mode**, atau tekan **Ctrl + E**. Untuk menghapus fitur yang dipilih, tekan tombol **Delete** atau klik **Delete selected features**. Terakhir, matikan mode pengeditan dengan menekan **Ctrl + E** atau klik kembali **Toggle editing mode**

Sekarang, lapisan hotspot hanya berisi poligon hotspot, yang dapat digunakan untuk analisis lebih lanjut. Misalnya, kita dapat menggabungkan kluster ini dengan informasi tentang bangunan terdekat dan jenis kebisingan untuk menemukan ketergantungan dan mengembangkan beberapa saran untuk mengurangi tingkat kebisingan di sana.

Mengamati pola distribusi dengan garis kontur

Selain mendeteksi titik panas, **heatmap** juga dapat digunakan untuk mendeteksi perubahan intensitas atau memvisualisasikan arah perubahan nilai. Cara paling umum untuk melakukan kedua tugas ini adalah dengan membuat garis kontur.

Untungnya, QGIS memiliki semua alat yang diperlukan untuk ini. kita akan menggunakan **Processing Toolbox** lagi, tetapi pembuatan garis kontur juga tersedia di plugin **GDALTools** (yang dapat ditemukan di menu Raster). Di kotak **Processing Toolbox**, Anda dapat menemukan algoritma **Contour** de-



Gambar 5.27: Creating Contour

ngan mengetik kata tersebut di bidang filter di bagian atas kotak alat. Klik dua kali pada nama algoritma untuk membuka dialognya, yang terlihat seperti ini:

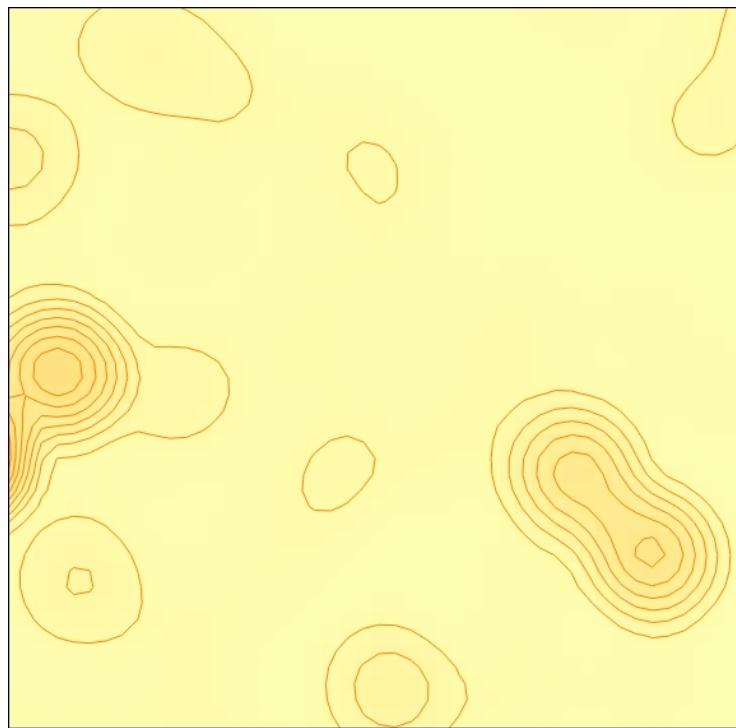
Pilih layer **heatmap** sebagai input. Di bagian **Output**, tentukan direktori di mana hasil akan disimpan. Juga, perlu untuk menentukan **Interval between contour lines**. Tidak ada prinsip yang kaku tentang penentuan interval ini. Aturan umum adalah memilih interval yang mendekati pola di area dengan perubahan kepadatan halus. Di sini kita akan memilih interval 10.

Ketika semua informasi yang diperlukan telah ditentukan, klik **Run** untuk mulai pembuatan garis kontur. Setelah beberapa waktu, layer vektor poligonal baru akan ditambahkan ke QGIS, dan kita dapat mulai menganalisisnya. Pertama, jika perlu, pindahkan layer kontur ke atas **heat map** di QGIS. Selain itu, lebih baik menyesuaikan simbolologi kontur untuk membuatnya lebih mudah dikenali dari latar belakang **peta panas**.

Kontur yang lebih padat sesuai dengan perubahan kepadatan yang lebih intens. Selain itu, kita dapat mengidentifikasi arah perubahan kebisingan. Misalnya, dalam *screenshot* sebelumnya, kita dapat melihat bahwa di beberapa tempat, distribusi kebisingan di sekitar pusat tidak sama; intensitas berkurang lebih cepat di tenggara daripada di barat laut. Jadi, kita dapat berasumsi bahwa ada beberapa hambatan untuk kebisingan di sana.

5.4 Analisis kesesuaian

Kita hidup di dunia yang penuh dengan berbagai hubungan yang dapat dianalisis dalam konteks fungsional, temporal, atau spasial. Hubungan spasial merupakan hal yang sangat menarik di bidang GIS, karena di sini objek ruang direpresentasikan dengan cara yang membantu penjelasan serta memperlihatkan hubungan geografis mereka. Analisis kesesuaian adalah bagian mendasar dari analisis GIS yang menjawab pertanyaan, “Di mana tempat terbaik untuk menempatkan fasilitas baru?” Dalam bab ini, kita akan dihadapkan pada dasar-dasar analisis kesesuaian melalui pencarian tempat terbaik untuk tempat



Gambar 5.28: Contour heatmap

tinggal. Kita akan belajar cara:

- menginterpretasikan relasi spasial antar obyek
- mengekspresikan relasi ini melalui data spasial
- menganalisis data spasial sesuai serangkaian kriteria yang ditentukan
- melakukan analisis overlay dan menjelaskan hasilnya

Dasar Analisis Kesesuaian

Analisis kesesuaian diakui sebagai pendekatan *multi-criteria decision support*. Dengan kata lain, tujuan utamanya adalah untuk membagi bidang yang diminati menjadi dua kategori berdasarkan seperangkat kriteria yang telah ditentukan: sesuai untuk beberapa jenis penggunaan (hidup, bangunan, konservasi, dan sebagainya) dan tidak sesuai. Pendekatan umum yang digunakan untuk penilaian kesesuaian adalah overlay lapisan ganda yang mendukung keputusan multi-kriteria. Bergantung pada data yang mewakili kriteria kesesuaian dan overlay, ada dua pendekatan dasar yang tersedia:

1. Analisis kesesuaian dengan data vektor.

Ini terutama menggunakan operasi seperti buffering dan kombinasi berurutan mereka menggunakan operasi overlay vektor, seperti **Clipping**, **Intersection**, dan **Union**.

2. Analisis kesesuaian dengan data raster.

Ini sangat bergantung pada aljabar raster, yang digunakan untuk mengklasifikasi ulang cakupan raster awal dan kemudian menggabungkannya untuk menghasilkan raster kesesuaian biner atau peringkat. Pendekatan ini lebih fleksibel, karena memungkinkan untuk menghasilkan beberapa kelas kesesuaian dan mengubah bobot raster sesuai dengan pentingnya faktor yang diwakilinya. Dampaknya, pengguna dapat menghasilkan kombinasi hasil, tetapi alur kerja membutuhkan lebih banyak upaya yang terhubung ke pengambilan data dan keputusan.

	Vector data	Raster data
Main operations	Buffering Clip overlay Intersection overlay Union overlay	Vector data rasterization Proximity raster creation Raster reclassification Raster algebra addition Raster algebra multiplication Raster algebra subtraction
Advantages	Workflow quickness Workflow simplicity Good representation of man-made features	Simple data reclassification Good representation of continuous features Crisp and fuzzy classes are possible Different weighting according to their importance Various assessments are possible
Limitations	Provide only crisp classes Usually provide binary assessment only	Reclassification and ranking subjectivity Workflow complexity

Apa pun pendekatan yang akan diikuti, alur kerja analisis kesesuaian umum melibatkan beberapa langkah umum. kita sekarang akan melihat lebih dekat pada mereka untuk memastikan pemahaman yang lebih baik tentang sistem analisis kesesuaian:

1. Definisikan maksud dan tujuan dari analisis Anda

Pertanyaan yang akan dipelajari dirumuskan secara umum, dan signifikansi terapannya ditentukan, yang nantinya akan menjadi seperangkat kriteria kesesuaian.

Beberapa aplikasi kesesuaian yang populer meliputi yang berikut:

Pertanian : Penilaian kesesuaian area untuk budidaya tanaman tertentu.

Ritel : Area dinilai dari sudut pandang pemasaran — apakah akan menarik pelanggan atau pembeli baru, atau tidak. Jenis analisis ini sangat diminati ketika memilih lokasi belanja yang disukai.

Energi terbarukan : Menilai kesesuaian lahan untuk lokasi tenaga angin atau stasiun tenaga surya adalah tren yang luar biasa di bidang perencanaan geospasial untuk keberlanjutan.

Konservasi alam : Kebutuhan konservasi diprioritaskan menggunakan pemodelan kesesuaian habitat, dan kawasan ini dibagi menjadi lokasi yang lebih atau kurang bernilai untuk kelangsungan hidup dan reproduksi spesies tertentu.

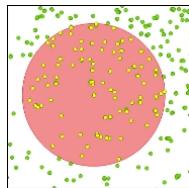
Secara umum, aplikasi utama dari analisis kesesuaian adalah di bidang perencanaan penggunaan lahan, yang bertujuan memprioritaskan berbagai jenis kegiatan manusia dalam ruang dan sumber daya alam yang terbatas.

2. Lakukan analisis ketersediaan data dan tentukan relevansinya terhadap maksud dan tujuan

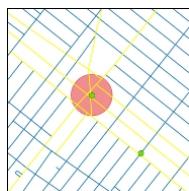
Relevansi data dengan tujuan dan sasaran hendaknya ditentukan. Ketersediaan data saat ini dan kebutuhan data masa depan sebaiknya dianalisis, terutama pengetahuan tentang apakah turunan data saat ini dapat digunakan untuk analisis atau tidak. Misalnya, jika kita harus menganalisis kesesuaian untuk kebutuhan pertanian, DEM dapat menjadi sumber yang bagus. Ini tidak hanya memberikan informasi dasar tentang **relief**, tetapi juga beberapa turunan yang bermanfaat, seperti kemiringan dan aspek. Hasil utama dari tahap ini adalah daftar sumber data primer dan turunan potensial mereka.

3. Definisikan kriteria dari analisis

Ini adalah tahap yang paling penting, di mana tujuan analisis digambarkan sebagai kriteria numerik yang jelas berdasarkan data yang relevan. Tujuan deskriptif diterjemahkan ke dalam bahasa analisis SIG. Pada tahap ini, berbagai jenis hubungan spasial antara objek dianalisis, dan beberapa hubungan paling populer termasuk yang berikut:

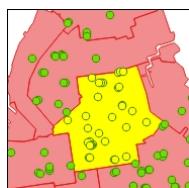


Gambar 5.29: Example of the "is within" point-to-point relationship

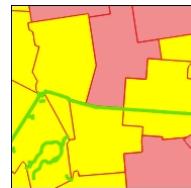


Gambar 5.30: Example of the "nearest to point-to-line" relationship

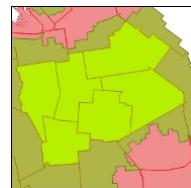
- hubungan point-to-point :
 - "is within": Semua sekolah yang berjarak 1 km dari tempat tinggal
 - "is nearest to": Sekolah dasar yang paling dekat dengan tempat tinggal
- hubungan point-to-line:
 - "is nearest to": alan yang paling dekat dengan pintu masuk stasiun kereta bawah tanah
 - point-to-polygon relationships:
 - "is contained in": Semua sekolah negeri dalam batas komunitas tertentu
 - relasi line-to-line:
 - "crosses": Apakah jalan setapak tertentu melintasi jalan
 - "is within": Temukan semua jalan setapak yang berjarak 1 km dari sungai tertentu
 - "is connected to": Temukan semua jalan yang terhubung ke jalan raya tertentu



Gambar 5.31: Example of the "is contained in" point-to-polygon relationship



Gambar 5.32: Example of the "intersects" line-to-polygon relationship



Gambar 5.33: Example of the "completely within" polygon-to-polygon relationship

- relasi line-to-polygon:
 - “intersects”: Temukan semua distrik yang dilintasi oleh jalan setapak
 - “contains”: Temukan semua jalan yang benar-benar dalam satu kabupaten tertentu
- relasi polygon-to-polygon:
 - “completely within”: Temukan semua kabupaten yang sepenuhnya berada dalam zona bahaya
 - “is nearest to”: Find the building that is nearest to a park

4. Analisis primer dan penyiapan data

Data dianalisis sesuai dengan serangkaian kriteria yang ditentukan pada tahap sebelumnya. Operasi analisis umum melibatkan pemilihan berdasarkan lokasi, buffering, rasterisasi, *proximity* (jarak raster), dan sebagainya. Setelah semua lapisan yang diperlukan siap, data tersebut harus siap untuk di-overlay, yang melibatkan proyeksi ulang ke sistem referensi koordinat umum (jika perlu), menetapkan rentang untuk berbagai peringkat, dan klasifikasi ulang dalam sistem peringkat umum. Semua layer harus berisi nilai dalam unit yang seragam, jika tidak, overlaynya akan menjadi tidak berarti dan sulit untuk ditafsirkan.

5. Overlay data dan interpretasikan hasilnya

Lapisan yang disiapkan sebelumnya digabungkan menjadi satu cakupan yang didasarkan pada seperangkat aturan yang ditentukan pengguna. Bergantung pada data yang tersedia dan aturan yang diterapkan, penilaian kesesuaian berikut dimungkinkan:

- **Binary suitability assessment**

Semua area dibagi ke dalam kategori yang sesuai dan tidak sesuai. Ini adalah tipe penilaian paling sederhana yang dapat diperoleh dari overlay data vektor.

- **Ranked suitability assessment**

Tempat diberi peringkat dari yang paling tidak sesuai hingga yang paling sesuai berdasarkan seluruh rentang kriteria yang telah ditentukan. Jenis penilaian ini dapat diturunkan dari data vektor dan raster. Ini memungkinkan Anda menghindari penilaian ya / tidak yang sederhana, yang tidak melekat pada kata aslinya. Keuntungan ini diimbangi oleh subjektivitas peringkat data dan kepentingan yang sama dari berbagai faktor. Namun demikian, di dunia nyata, kontribusi mereka terhadap penilaian keseluruhan dapat bervariasi.

- **Weighted suitability assessment**

Ini serupa dengan jenis penilaian sebelumnya dan hanya memiliki satu perbedaan signifikan: berbagai faktor dapat diberi bobot berbeda sesuai dengan kepentingannya untuk jenis kegiatan tertentu. Jenis penilaian ini bergantung pada pendekatan aljabar raster dan dianggap inklusif, tetapi bukan tanpa subjektivitas, terutama ketika menyangkut faktor pembobotan dan menafsirkan hasil akhir.

5.4.1 Tahap 1: Mendefinisikan maksud dan tujuan

Sepanjang tutorial ini, kita akan menganggap bahwa kita bekerja untuk satu pasangan muda dengan anak kecil. Mereka mencari tempat yang sempurna untuk tinggal di wilayah tertentu yang menarik perhatian mereka. Tujuan kita adalah menggunakan kekuatan metode analisis kesesuaian berbasis GIS dan memberikan jawaban yang objektif dan dapat diandalkan untuk pertanyaan mereka.

Tujuan dari analisis ini adalah untuk menemukan area yang cocok untuk keluarga muda dengan anak, dengan pertimbangan tertentu. Banyak dari persyaratan mereka mirip dengan yang dimiliki oleh perusahaan pengembang perumahan tradisional. Misalnya, kedekatan dengan stasiun kereta bawah tanah, zona hijau, dan keselamatan publik harus dipertimbangkan. Ada juga beberapa persyaratan khusus keluarga yang harus dipertimbangkan. Seperti yang telah disebutkan, keluarga memiliki anak kecil, yang berarti bahwa kita harus memperhitungkan keberadaan anak usia dini atau sekolah dasar di dekatnya. Juga, mereka tertarik pada olahraga, dan akan lebih bagus jika daerah tempat mereka akan tinggal memiliki infrastruktur istirahat yang berkembang dengan baik dan aktif. Setelah tinjauan semacam ini, kita dapat merumuskan beberapa persyaratan dan tujuan yang lebih spesifik:

- **Keselamatan:** Area tidak boleh terkena atau memiliki resiko tinggi terhadap ragam bahaya alam dan kejahatan

- **Konektivitas:** Harus terhubung dengan baik ke jaringan transportasi kota
- **Greenness and openness:** Harus dekat dengan taman atau area hijau lainnya
- **Educational potential:** Ada lokasi pendidikan anak usia dini atau sekolah dasar di lingkungan tersebut
- **Active rest opportunities:** Termasuk jaringan bersepeda dan fasilitas atletik
- **Cultural life:** Galeri seni dan museum dapat dikenali sebagai tanda umum hidupnya budaya

Sekarang setelah tujuan dan persyaratan utama telah diklarifikasi, kita dapat melanjutkan ke langkah berikutnya dan mempelajari semua data yang tersedia untuk menilai relevansinya dengan contoh kita.

5.4.2 Tahap 2 : Analisis ketersediaan data dan definisikan relevansi

Segara setelah kita menetapkan persyaratan dasar, kita perlu melakukan eksplorasi data yang mungkin untuk digunakan dan relevan untuk analisis. Dataset pelatihan berisi sejumlah besar dataset, dan yang paling relevan di antaranya tercantum dalam daftar berikut:

5.4.2.1 Keselamatan

- Layer *hurricane_evacuation_zones* :

Zona evakuasi badai adalah area kota yang mungkin perlu dievakuasi karena ancaman yang terkait dengan keselamatan dan kehidupan dari badai topan.

- Layer *hurricane_inundation_zones* :

Zona genangan badai adalah area genangan badai terburuk.

- Layer *noise_heatmap* :

Raster yang dibuat di bagian membuat *Heat map*, yang menunjukkan kepadatan spasial pengaduan kebisingan yang terdaftar mungkin berguna untuk penilaian potensial untuk keselamatan publik.

5.4.2.2 Konektivitas

- Layer *subway_entrances* :

Lokasi pintu masuk kereta bawah tanah.

5.4.2.3 Greenness dan openness

- Layer parks :

Lapisan yang berisi informasi ruang terbuka hijau, seperti lapangan, trek, taman, dan sebagainya.

- Layer *tree_density* :

Ini adalah layer raster yang dibuat dari data sensus pohon.

5.4.2.4 Educational potential

- Layer elementary_schools :

Ini adalah lokasi titik sekolah berdasarkan alamat resmi. Lapisan ini mencakup beberapa informasi dasar tentang sekolah, seperti nama, alamat, jenis, dan informasi kontak kepala sekolah.

5.4.2.5 Active rest opportunities

- Layer bike_routes :

Lokasi jalur sepeda dan rute di seluruh kota.

- Layer athletic_facilities

Lapisan ini berisi fasilitas atletik dan beberapa informasi dasar tentangnya, termasuk jenis olahraga utama, permukaan, dimensi, dan sebagainya.

5.4.2.6 Cultural life

- Layer musemart :

Lokasi museum dan galeri seni.

5.4.3 Tahap 3 : Definisikan kriteria analisis

Pada layer *hurricane_evacuation_zones*, ada enam zona, yang diperingkat dalam zona bidang atribut berdasarkan risiko dampak gelombang badai, dengan zona 1 menjadi wilayah yang kemungkinan besar akan banjir. Jika terjadi badai atau badai tropis, penghuni di zona ini harus mengungsi. Daerah dengan nilai zona X tidak berada dalam zona evakuasi. Area dengan nilai zona 0 adalah salah satu dari yang berikut: air, dermaga kecil, atau pulau-pulau tak berpenghuni. Untuk keperluan analisis, lapisan ini harus dirasterisasi dan diberi peringkat sesuai dengan risiko dampak badai, dengan nilai peringkat turun dari area yang tidak berada dalam zona evakuasi ke yang paling mungkin terkena banjir.

Lapisan poligon *hurricane_inundation_zones* berisi informasi tentang risiko geangan akibat badai, ditulis di kolom atribut, di mana nilainya adalah ketinggian lonjakan (dalam feet). Area yang kemungkinan besar akan tergenang diberi nilai 1, dan area yang dikecualikan dari pemodelan inundasi diberi nilai 5. Lapisan ini harus dirasterisasi dan diberi peringkat dengan nilai kesesuaian potensial tertinggi untuk area yang dikecualikan, dan yang terendah untuk bidang kategori 1.

Layer raster *noise_heatmap* adalah raster yang harus diberi peringkat menggunakan beberapa kategori, dengan nilai kesesuaian terendah untuk tempat paling berisik dan sebaliknya. Hal yang baik di sini adalah kita tidak perlu melakukan rasterisasi layer ini, seperti yang kita lakukan pada layer sebelumnya. Pada saat yang sama, menetapkan jumlah dan kisaran untuk peringkat membawa subjektivitas ke dalam penilaian kita. Layer *tree_density*, yang juga merupakan raster kepadatan, harus dianalisis dengan cara yang sama.

Lapisan yang dipilih lainnya harus dianalisis terlebih dahulu untuk kedekatannya. Untuk tujuan ini, pertama-tama kita akan merasterisasi mereka, kemudian membuat *continuous raster proximity*, dan akhirnya peringkat mereka di bawah beberapa kategori sesuai dengan nilai *proximity* (semakin dekat suatu objek, semakin tinggi nilai kesesuaian). Sekali lagi, dalam hal peringkat pengguna, kita tidak akan dapat menghindari beberapa subjektivitas dalam penilaian kita. Selain itu, raster kedekatan akhir dapat ditimbang menurut kepentingannya dalam penilaian kesesuaian keseluruhan.

5.4.4 Tahap 4 : Analisis dan persiapan data

Ada tiga pendekatan utama untuk analisis data primer. Ini tergantung pada tipe data awal dan atribut yang tersedia:

- **Rasterisasi dan rangking layer vektor yang dikategorikan**

Ini adalah layer yang sudah mengandung semua nilai yang diperlukan, dan pada tahap persiapan, semuanya harus dirasterisasi ke tingkat dan resolusi yang sama. Selain itu, kategorinya harus diberi peringkat dengan benar, dengan nilai tertinggi untuk area yang paling cocok dan sebaliknya. Contoh dari lapisan ini adalah *hurricane_evacuation_zones*, *hurricane_inundation_zones*, dan sebagainya.

- **Raster peringkat kepadatan**

Ini adalah raster **heat map** yang harus dikonversi dari *continuous coverage* ke nilai yang dikategorikan di mana nilai tertinggi melambangkan area yang paling tepat, dan yang terendah terkait dengan area yang paling tidak cocok. Contoh dari layer ini adalah *noise_heatmap* dan *tree_density*.

- **Menghasilkan dan menentukan peringkat proximity raster**

Ini adalah alur kerja yang paling membosankan. Lapisan vektor harus diraster terlebih dahulu, dan kemudian *proximity raster* harus dibuat dan diberi peringkat dengan benar. Kategori ini mencakup lapisan vektor berikut: subway_entrances, taman, public_schools, bike_routes, athletic_facilities, dan museumart.

Perhatikan bahwa untuk hasil akhir, kita akan selalu memiliki peringkat raster nilai unik, dengan nilai tertinggi menunjukkan area yang paling cocok. Juga, penting bahwa semua raster keluaran berbagi tingkat yang sama, yang diperlukan untuk overlay yang tepat dengan kalkulator raster dan penilaian kesesuaian keseluruhan.

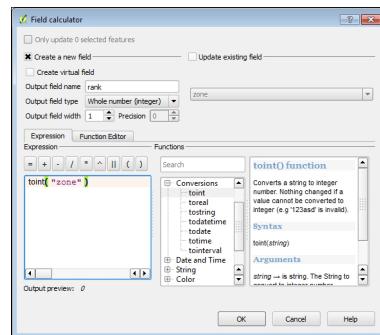
Di bagian mendatang, kita akan melalui alur kerja yang disebutkan sebelumnya untuk contoh lapisan raster. Segera setelah Anda memahami prinsipnya, Anda akan dapat menyiapkan lapisan lain secara mandiri.

5.4.4.1 Rasterisasi dan pemeringkatan layer vektor kategori

Dalam contoh ini, kita akan mengerjakan lapisan hurricane_evacuation_zones. Atribut yang sangat kita minati adalah zona, yang menggambarkan daerah diprioritaskan untuk evakuasi. Daerah dengan nilai terendah, 1, kemungkinan besar akan dievakuasi, dan sebaliknya. Dalam hal ini, kita dapat menggunakan nilai-nilai ini secara langsung untuk memberi peringkat raster.

Satu hal yang perlu diperhatikan ketika kita melakukan rasterisasi adalah kolom atribut yang digunakan untuk rasterisasi harus berupa angka. Jika Anda memeriksa kolom atribut di bagian **Field** di bawah **Properties**, Anda akan melihat bahwa zona bidang memiliki Tipe QString, dan nilainya yang berisi angka 1-6 dan huruf (X) ditafsirkan bukan sebagai angka tetapi sebagai urutan simbol atau string. Itu sebabnya kolom ini tidak tersedia untuk rasterisasi dan pertama-tama harus dikonversi menjadi angka. Ini dapat dilakukan dengan mudah dengan **Field calculator**:

1. Buka tabel atribut lapisan menggunakan klik kanan **Open Attribute Table**, atau tekan tombol relatif dari toolbar **Attribute**. Di bilah alat tabel atribut, klik tombol **Open field calculator** atau gunakan pintasan keyboard Ctrl + I.
2. Pertama-tama, kita harus membuang nilai X yang tidak dapat diartikan sebagai angka dan tidak dapat dikonversi ke dalamnya. Karena kita hanya memiliki satu baris yang berisi nilai X, kita cukup beralih mode pengeditan dengan mengklik tombol di toolbar tabel atribut. Klik dua kali pada sel dan masukkan nilai baru 7. secara manual. Jika Anda memiliki beberapa nilai untuk diubah, Anda dapat menggunakan ekspresi berikut di Kalkulator bidang untuk mengubah beberapa nilai bidang zona: CASE WHEN "zone" = 'X' THEN '7' ELSE "zone" END.

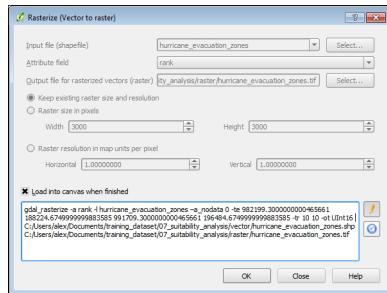


Gambar 5.34: Field Calculator

3. Klik pada tombol untuk membuka jendela dialog **Field calculator** seperti yang ditunjukkan pada gambar berikut, dan lakukan penyesuaian berikut:
 - Pastikan bahwa sakelar **Create a new field** diaktifkan.
 - Ketikkan nama kolom **Output** secara manual, misalnya, peringkat.
 - Pilih **Whole number (integer)** dari daftar tipe field Output, karena kita akan menggunakan integer pendek untuk peringkat.
 - Kurangi lebar bidang Output ke 1. Ini karena nilai peringkat tidak melebihi 10 dan kita tidak ingin membuat data berlebih dengan menghasilkan kolom panjang yang melebihi nilai data aktual.
 - Di kolom **Expression**, kita perlu mengetik fungsi yang akan digunakan untuk membuat nilai-nilai dari bidang baru. Dalam daftar **Function**, perluas item **Conversions** dan klik dua kali pada fungsi **toint**. Menurut deskripsinya, ini mengonversi string ke angka integer. Tidak ada yang berubah jika nilai tidak dapat dikonversi ke integer (misalnya, 123asd tidak valid). Setelah mengklik dua kali, fungsi akan ditambahkan ke ekspresi dengan braket terbuka, setelah itu Anda harus mengetik (atau mengklik dua kali untuk menambahkan item dari **Field and Values**) nama kolom yang akan dikonversi dalam tanda kutip ganda, dan tutup kurung. Dalam kasus kita, ekspresi yang dihasilkan adalah **toint ("zone")**.
 - Setelah mengklik tombol **OK**, kolom baru akan muncul di akhir tabel. Nonaktifkan mode pengeditan, konfirmasi penyimpanan hasil edit, dan keluar dari jendela tabel atribut.

Lapisan siap untuk rasterisasi. Buka jendela dialog dengan pergi ke **Raster | Conversion | Rasterize (Vektor to Raster)**. Di jendela dialog ini seperti yang ditunjukkan pada gambar berikut, sesuaikan pengaturan berikut:

1. Dari daftar drop-down File input (shapefile), pilih *hurricane_evacuation_zones*.
2. Dari daftar **Attribute**, pilih peringkat.



Gambar 5.35: Rasterize

3. Dalam **Output file for rasterized vectors** (raster), klik tombol **Select**. Arahkan ke direktori kerja Anda, dan ketik hurricane_evacuation_zones.tif sebagai nama layer baru. Pesan ini akan ditampilkan: **The output file doesn't exist**. Anda harus mengatur ukuran atau resolusi output untuk membuatnya. Klik OK dan lanjutkan ke tahap berikutnya.
4. Pada langkah sebelumnya, Anda menetapkan beberapa opsi utama. Untuk kenyamanan, kita akan membuat raster dalam tutorial ini dengan tingkat dan resolusi yang sama menggunakan lidar_dem.tif sebagai template. Klik pada tombol untuk membuat parameter baris perintah *gdal_rasterize* dapat diedit. Setelah modifikasi, garis harus terlihat mirip dengan contoh berikut:

```
gdal_rasterize -a rank -l hurricane_evacuation_zones -a_nodata 0 -te 982199.3000000000465661 188224.67499999983585 991709.30000000465661 196484.67499999983585 -tr 10 10 -ot UInt16
```

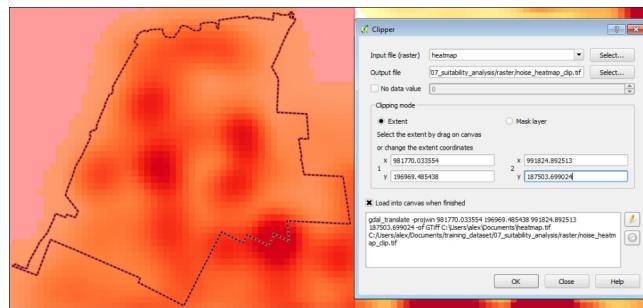
Ini berarti bahwa raster luaran akan berisi nilai raster dari bidang atribut peringkat. Area di luar layer poligon akan diberi nilai 0, yang akan ditafsirkan sebagai nodata. Tipe data output adalah bilangan bulat *unsigned 16-bit*.

5. Klik pada tombol **OK**. Hasil rasterisasi akan muncul di panel Layers.

Layer *hurricane_inundation_zones* harus diproses sebelumnya dengan cara yang sama. Lapisan berisi beberapa nomor zona. Ini dapat diartikan sebagai tingkat keparahan risiko genangan; yaitu, semakin tinggi angkanya, semakin rendah risikonya. Ini berarti bahwa kita dapat menggunakan nilai-nilai ini secara langsung untuk peringkat. Dalam hal lapisan ini, parameter baris perintah *gdal_rasterize* akan terlihat sebagai berikut:

```
gdal_rasterize -a category -l hurricane_inundation_zones -a_nodata 0 -te 982199.3000000000465661
```

Ketika dikombinasikan, lapisan-lapisan ini dapat memberikan penilaian kumulatif kesesuaian berdasarkan keparahan risiko dari bahaya alam.



Gambar 5.36: Clipper

5.4.4.2 Mengurutkan raster kepadatan

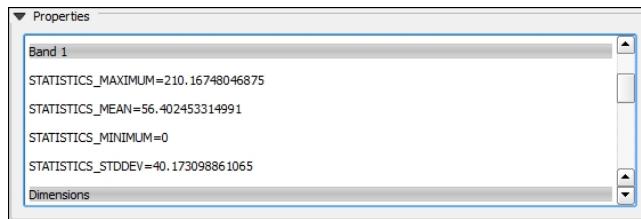
Dalam kasus raster kepadatan, kita dapat menggunakan peta noise_heatamp.tif dan membuat layer tree_density.tif sendiri. Pertama, kita akan menyiapkan noise_heatmap.tif yang awalnya jauh lebih besar daripada bidang minat yang kita kerjakan dalam bab ini. Untuk klip layer raster, gunakan menu berikut **Raster | Extraction | Clipper** :

1. Dari daftar drop-down **File input** (raster) pilih *noise-heatmap*, yang merupakan raster yang akan dipotong.
2. Dalam file **Output**, klik **Select button** untuk mengatur direktori dan nama untuk file output, misalnya, **noise_heatmap_clip.tif**.
3. Di bagian **Clipping mode**, ada dua mode yang dapat dipilih:
 - *Extent* adalah tempat Anda dapat memasukkan koordinat kotak pembatas secara manual atau dengan menyeret kanvas peta. Gunakan pendekatan ini untuk mengatur sejauh mana file output Anda. Ingat saja bahwa luasnya harus melebihi batas bidang minat yang ditetapkan oleh zipcode_bound shapefile.
 - Jika **Mask layer** aktif, Anda dapat memilih bentuk poligonal, dan itu akan digunakan sebagai batas kliping.

Setelah Anda mengklik tombol **OK**, layer akan dimuat ke kanvas peta.

Langkah-langkah berikut harus dilakukan untuk menilai rentang nilainya, membaginya ke dalam kategori, dan memberi peringkat:

1. Klik dua kali pada layer *noise_heatmap_clip* untuk membuka jendela **Layer Properties**. Di jendela ini, buka bagian **Metadata**. Jelajahi jendela Properties di bagian bawah dialog sampai Anda menemukan garis yang disorot Band 1 dengan statistik raster dasar, seperti yang ditunjukkan di sini:



Gambar 5.37: Band 1 Metadata

Kita akan menangani raster kepadatan, dan nilainya ditafsirkan sebagai jumlah keluhan gangguan per area piksel. Permukaan yang terus menerus ini harus dibagi ke dalam beberapa kategori dan harus diberi peringkat sesuai dengan jumlah pengaduan. Untuk ini, sejumlah kritis pengaduan harus ditetapkan. Tidak ada batasan dan persyaratan yang diakui secara resmi, tetapi kita dapat mengartikan nilai dataset rata-rata sebagai beberapa angka kritis (sedikit lebih besar dari 56, tetapi kita akan menggunakan 50 untuk kenyamanan). Bagilah nilai-nilai ke dalam kategori-kategori dan berikan mereka peringkat berikut (semakin sedikit jumlah keluhan, semakin baik):

Category	Value range	Suitability rank
1	Less than 50	5
2	50 to 100	4
3	100 to 150	3
4	150 to 200	2
5	Greater than 200	1

Untuk pemeringkatan, gunakan menu **Raster | Raster Calculator** dan ubah suaikan opsi berikut dalam dialog:

- Pilih direktori dan nama layer **Output**, sebagai contoh, **noise_ranked**
- Pilih band raster **noise_heatmap_clip** ? dan klik pada tombol **Current layer extent** untuk memastikan layer luaran memiliki resolusi dan extent yang sama.
- Pada window ekspresi **Raster Calculator**, masukkan ekspresi berikut:

```
("noise_heatmap_clip@1" <= 50) *5 + ("noise_heatmap_clip@1" > 50 AND "noise_heatmap_clip@1" <= 100)
```

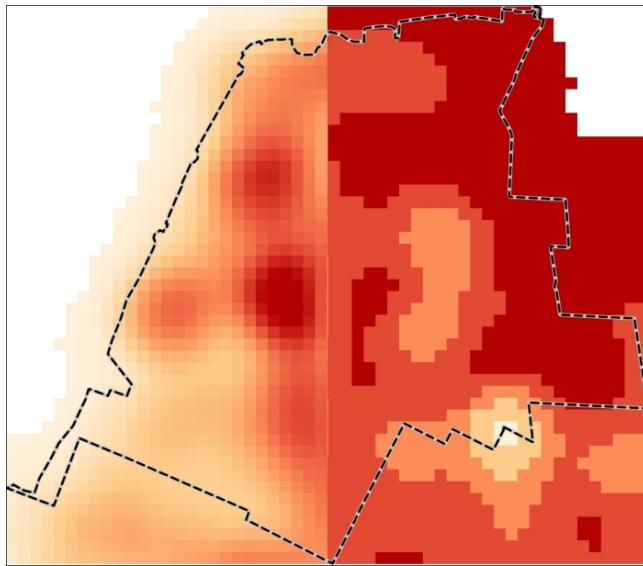
Ekspresi ini berarti bahwa setiap piksel yang berada di bawah rentang tertentu yang diberikan dalam tanda kurung pertama diberi nilai 1, dan kemudian diperingkat oleh nilai pengali tertentu, yang berada di luar tanda kurung. Setelah Anda mengklik tombol OK, raster yang direklasifikasi akan dimuat ke panel Layers.

Perhatikan hal ini: karena peringkat, peta panas terbalik; yaitu, hotspot paling berisik mendapatkan peringkat terendah dan tempat paling tenang mendapatkan peringkat tertinggi.

Dengan cara yang sama, kita dapat membuat dan memberi peringkat peta panas untuk objek titik lain yang menarik dalam analisis kesesuaian kita, yaitu pohon dan museumart .

5.4.4.3 Pembuatan dan pengurutan proximity rasters

Alur kerja akan dijelaskan dalam contoh lapisan vektor titik subway_entrances:



Gambar 5.38: Reclassified raster

- Pertama, layer harus dirasterisasi dengan cara yang umum. Baris `gdal_rasterize` akan berisi parameter berikut:t, the layer should be rasterized in a common way. The `gdal_rasterize` line will contain the following parameters:

```
gdal_rasterize -l subway_entances -burn 1 -a_nodata 0 -te 982199.3000000000465661 1882
```

Di lapisan output, lokasi titik yang ada akan ditandai dengan nilai piksel 1, sementara semua area lain akan diberi nilai nodata 0.

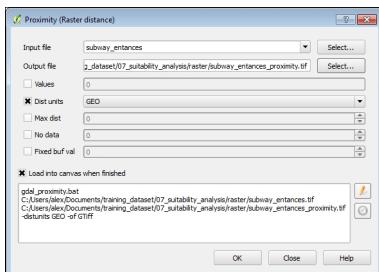
- rgi ke _ Raster | Analisis | Kedekatan (Jarak Raster) . Dialog ini menghasilkan peta kedekatan raster yang menunjukkan jarak dari pusat setiap piksel ke pusat piksel terdekat yang diidentifikasi sebagai piksel target. Pixel target adalah piksel dalam raster sumber yang nilai piksel rasternya berada di set nilai piksel target. Jika tidak ditentukan, semua piksel bukan nol akan dianggap piksel target. Di jendela dialog, sesuaikan parameter berikut: Dari daftar drop-down File input, pilih layer `subway_entances`

Dalam file Output, klik tombol Select dan tentukan path dan nama untuk layer output, misalnya, `subway_entances_proximity.tif` .

Pastikan parameter unit Dist diaktifkan dan diatur ke GEO . Dalam hal ini, jarak yang dihasilkan akan berada dalam koordinat georeferensi (kaki).

Baris parameter yang dihasilkan akan terlihat seperti ini:

```
gdal_proximity.bat fullpath/subway_entances.tif fullpath/subway_entances_proximity.tif
```



Gambar 5.39: Proximity analyses

3. Kedekatan raster harus dibagi ke dalam kategori diskrit dan mereka harus diberi peringkat. Untuk operasi ini, kita akan menggunakan Kalkulator Raster. Masalah utama di sini adalah untuk memutuskan jumlah dan memilih kategori kedekatan yang tepat untuk peringkat. Keputusan optimal tergantung pada apa yang disebut jari-jari berjalan, yaitu jarak yang membuat orang nyaman untuk berjalan. Secara umum, perencana transportasi telah mengamati bahwa jarak berjalan yang tampaknya dilalui sebagian besar orang dengan nyaman — di luar itu penumpang turun secara drastis — adalah sekitar 400 m (sekitar 1.300 kaki). kita akan menerapkan aturan 400 m ini untuk mengkategorikan nilai-nilai layer (min sebagai nol dan maks sebagai 4.888,71) menjadi empat kategori, dengan peringkat berikut:

Kategori	Proximity values range (feet)	Suitability rank
1	Less than 1,300	4
2	1,300 to 2,600	3
3	2,600 to 3,900	2
4	Greater than 3,900	1

Buka **Raster | Raster Calculator** dan sesuaikan opsi utamanya. Masukkan ekspresi berikut untuk menghasilkan peta Enter the following expression to generate a new **subway_entrances_proximity_ranks** raster:

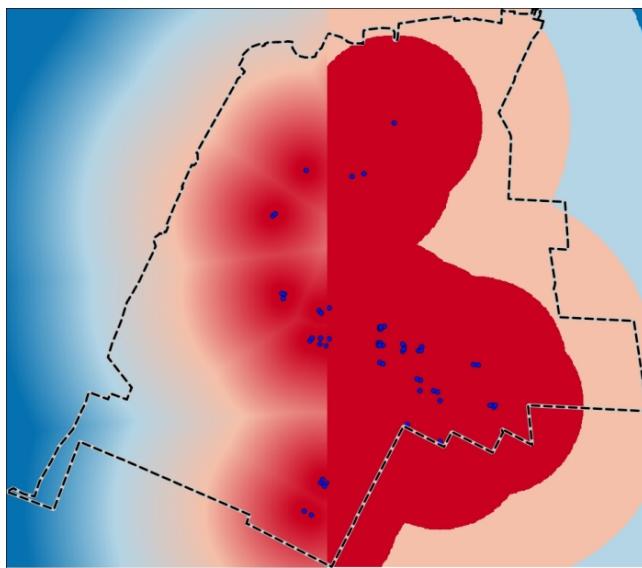
```
("subway_entances_proximity@1" <= 1300) *4 + ("subway_entances_proximity@1" > 1300 AND "subway_en
```

Di tangkapan layar berikut, Anda dapat melihat seperti apa raster berkelanjutan (di kiri) dan peringkat (di kanan):

Dengan cara yang sama, Anda dapat memproses ulang layer vektor lain yang harus dianalisis dari posisi kedekatan, yaitu taman, bike_routes, athletic_facilities, dan sekolah dasar. Sebagai hasilnya, Anda akan memiliki seperangkat lapisan raster yang diberi peringkat dengan beberapa kategori sesuai dengan kedekatan objek yang dipilih. Semakin tinggi pangkat, semakin dekat objek. Sebagai pedoman umum, terapkan nilai 400 m (atau 1300 kaki) untuk memberi peringkat pada raster dengan benar.

5.4.5 Tahap 5 : Overlay data dan interpretasi hasil

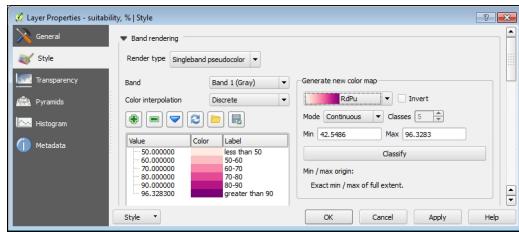
Sekarang kita memiliki segalanya siap menutupi raster dan menghasilkan penilaian kesesuaian kumulatif. Pada gambar berikut, Anda dapat melihat daftar lengkap lapisan, ditimbang oleh kepentingannya untuk kesesuaian umum. Bobot adalah koefisien sederhana dalam kisaran 0 hingga 1, dan mereka digunakan untuk memodifikasi peringkat dengan benar.



Gambar 5.40: Field Calculator

No.	Raster layer name	Raster layer type	Rank min	Rank max	Weight coefficient	Max possible value = max rank * weight
1	athletic_facilities_proximity_ranks	Proximity	1	5	0.04	0.2
2	bike_routes_proximity_ranks	Proximity	1	3	0.06	0.18
3	hurricane_evacuation_zones	Zoning	1	7	0.16	1.12
4	hurricane_inundation_zones	Zoning	1	5	0.16	0.8
5	Museumart_ranked	Density	1	4	0.05	0.2
6	Noise_ranked	Density	1	5	0.13	0.65
7	Parks_proximity_ranks	Proximity	1	3	0.12	0.36
8	Schools_proximity_ranks	Proximity	1	4	0.15	0.6
9	subway_entrances_proximity_ranks	Proximity	1	4	0.06	0.24
10	Tree_ranked	Density	1	4	0.07	0.28
Total			10	44	1.0	4.63

Gambar 5.41: list of layers weighted by their importance for general suitability



Gambar 5.42: Suitability layer

Ungkapan yang digunakan untuk penilaian kesesuaian kita dapat dikonstruktikan dengan beberapa langkah:

1. Semua faktor yang tersedia yang diwakili oleh layer raster harus dikalikan dengan koefisien bobotnya dan diringkas, seperti ini:

```
(factor_1*weight + factor_2*weight + factor_3*weight + ... factor_n*weight)
```

Jenis penilaian ini memberikan kesesuaian kotor, yang sulit untuk ditafsirkan karena nilai-nilai yang diperoleh tidak dihitung secara relatif ke minimum minimum dan maksimum.

2. Untuk kesederhanaan interpretasi, penilaian kesesuaian bruto dapat dibagi menjadi jumlah dari nilai maksimum yang mungkin. Rumus yang diperluas akan terlihat seperti ini:

```
(factor_1*weight + factor_2*weight + factor_3*weight + ... factor_n*weight) / (factor_1_max*weight)
```

Akibatnya, rentang nilai output akan dari 0 hingga 1, di mana nilai kesesuaian maksimum mendekati 1.

3. Secara opsional, hasilnya dapat dikalikan dengan 100, dan nilai output akan menjadi persentase kesesuaian.

Pergi ke Raster | Kalkulator Raster untuk melakukan penilaian:

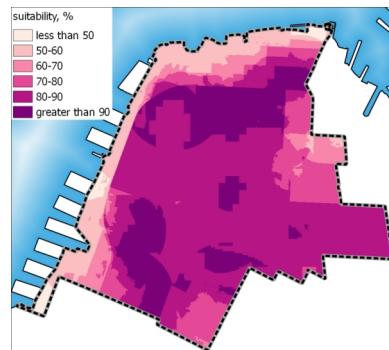
Atur path dan nama untuk layer Output

Pilih satu layer dari daftar Raster bands untuk mengatur Luas layer sekarang, misalnya, hurricane_inundataion

Di jendela Ekspresi, masukkan rumus berikut (jika Anda tidak yakin tentang nilai, lihat tabel sebelumnya):

```
(( "athletic_facilities_proximity_ranks@1" * 0.04 + "bike_routes_proximity_ranks@1" * 0.06 + "hurr
```

Setelah Anda mengklik tombol OK, lapisan yang dihasilkan akan ditambahkan ke kanvas peta. Kisaran nilai kesesuaian raster bervariasi dari 42 hingga 96 persen. Dengan demikian, dapat dengan mudah diklasifikasikan dan ditafsirkan. Arahkan ke properti Layer | Gaya dan sesuaikan properti rendering dengan yang ditunjukkan pada tangkapan layar berikut:



Gambar 5.43: Suitability map

Setelah menerapkan pengaturan ini, layer akan terlihat sebagai berikut:

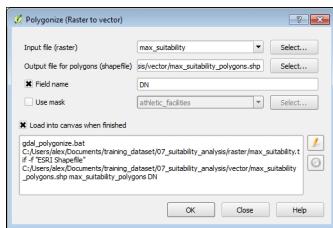
Kita dapat menggunakan raster ini sebagai dasar untuk penilaian kesesuaian visual utama dari bidang yang kita minati, atau melangkah lebih jauh dan menggabungkannya dengan lapisan lain untuk mengidentifikasi blok yang tepat yang paling cocok dengan seluruh rentang kriteria kesesuaian.

Dalam hal ini, kita perlu melakukan urutan terbalik langkah-langkah: pilih area yang paling cocok, vektorisasi, dan overlay poligon kesesuaian maksimum dengan ruang tamu perumahan untuk mengidentifikasi blok dan bangunan yang akan sangat cocok:

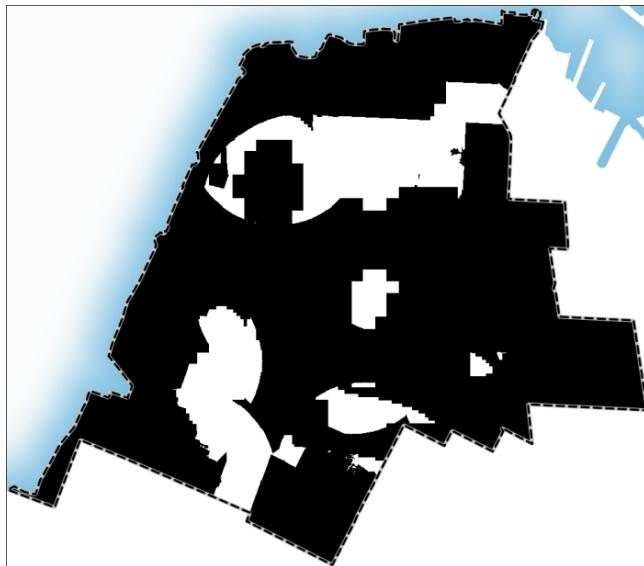
1. Raster kesesuaian awal harus dikategorikan ke dalam hanya dua kelas dengan batas kesesuaian 90 persen. Pergi ke Raster | Kalkulator Raster dan tentukan jalur dan nama untuk raster keluaran (misalnya, max_suitability). Di jendela Ekspresi, masukkan “kesesuaian,% (???)”> = 90. Pada halaman berikutnya, Anda dapat melihat bahwa lapisan yang dihasilkan hanya berisi dua kelas: cocok (nilainya 1; ditugaskan ke area dengan kesesuaian yang lebih besar dari atau sama dengan 90 persen), dan tidak cocok (nilainya 0).
2. Sekarang kita perlu melakukan vektorisasi area-area ini untuk dapat melakukan query layer vektor dengannya. Buka jendela dialog Polygonize (Raster to vector) dengan masuk ke Raster | Konversi dan sesuaikan parameter berikut:
 - Pilih max_suitability sebagai raster yang akan dipolygonisasi dari file Input (raster) .

Berikan path dan nama shapefile keluaran dalam file Output untuk poligon (shapefile) , misalnya, max_suitability_polygons.

Aktifkan sakelar Nama bidang dan terima nilai DN default. Opsi ini bertanggung jawab untuk membuat dan mengisi bidang dengan nilai kelas dari raster



Gambar 5.44: Polygonize



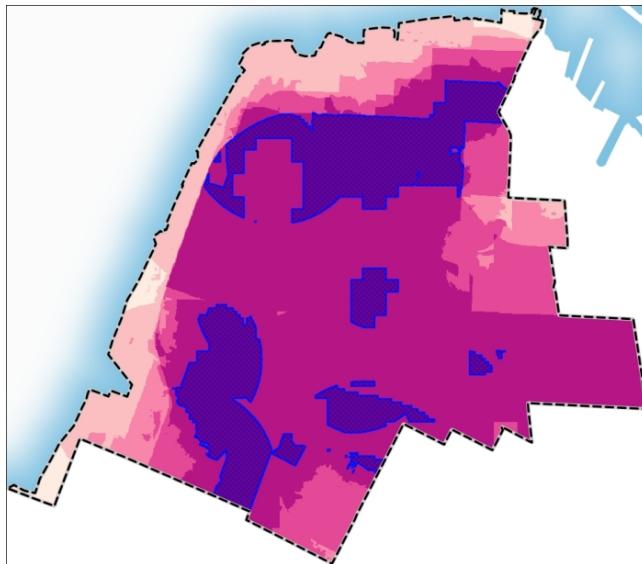
Gambar 5.45: Masking raster

awal.

3. Setelah Anda mengklik tombol OK, shapefile yang dihasilkan akan ditambahkan ke panel Layers. Awalnya, layer berisi poligon semua kelas, sedangkan kita hanya tertarik pada kelas 1. Untuk tujuan memilih dan menghapus poligon yang tidak perlu, kita akan menggunakan fitur Pilih menggunakan opsi ekspresi: Buka tabel atribut max_suitability dengan mengklik tombol di panel Attribute, atau dari klik kanan jalan pintas layer Open Attribute Table.

Di panel bilah alat tabel atribut, klik pada fitur Pilih menggunakan tombol ekspresi,, dan masukkan ekspresi berikut: “DN” = 0. Setelah mengklik OK, semua poligon yang memenuhi kondisi akan disorot dalam tabel atribut dan di kanvas peta.

Karena kita tidak memerlukan poligon ini untuk analisis lebih lanjut, kita harus



Gambar 5.46: Reclass maximum suitability

menghapusnya. Di panel bilah alat tabel atribut, klik tombol untuk mengaktifkan mode pengeditan, atau gunakan pintasan keyboard Ctrl + E. Sekarang kita dapat menjalankan Hapus fitur yang dipilih menggunakan tombol, atau cukup tekan Del dari keyboard.

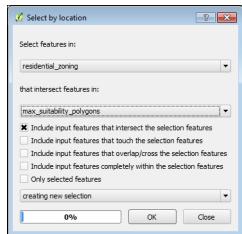
Setelah menghapus data yang tidak perlu, jangan lupa untuk menyimpan hasil edit Anda (menggunakan atau Ctrl + S) dan menonaktifkan mode pengeditan. Dalam tangkapan layar berikut, Anda dapat melihat bahwa layer hanya berisi poligon yang mencakup nilai kesesuaian maksimum, yang ditetapkan hingga 90 persen:

4. Sekarang layer ini dapat digunakan untuk overlay dengan layer vektor lainnya dan menganalisis hubungan spasial antara objek, seperti yang dijelaskan dalam bagian Dasar analisis kesesuaian. Sebagai contoh, kita dapat mengidentifikasi area zonasi area tempat tinggal primer yang berpotensi menarik bagi kita sesuai dengan kriteria kesesuaian:

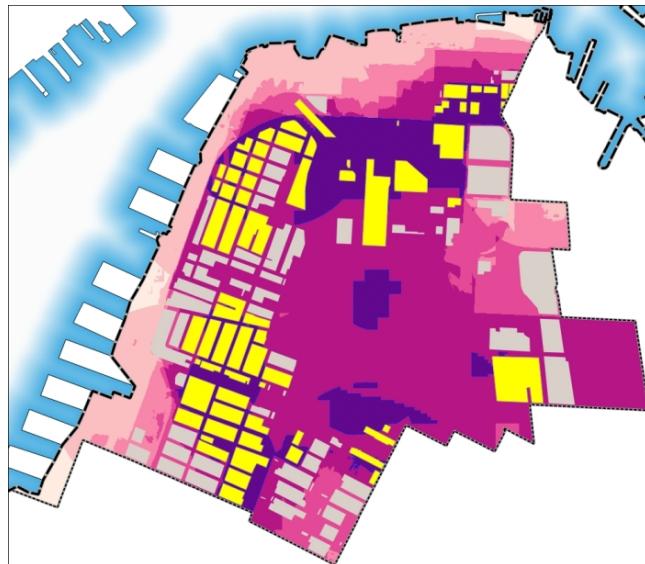
Buka jendela dialog dengan masuk ke Vector | Alat penelitian | Pilih berdasarkan lokasi . Dalam dialog ini, Anda pertama-tama harus memilih layer dari objek mana yang akan dipilih — residential_zoning dari fitur Pilih di : daftar turun bawah — seperti ini:

- fitur yang berpotongan di: daftar drop-down, pilih max_suitability_polygons, yang akan digunakan sebagai pemilihan.

Ada beberapa opsi pemilihan overlay. Aktifkan Sertakan fitur input yang memotong fitur seleksi. Hanya poligon yang berada di dalam batas atau kueri



Gambar 5.47: Select by location



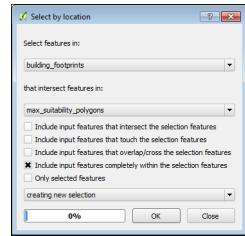
Gambar 5.48: Selected maximum suitability

mask berpotongan yang akan ditambahkan ke seleksi. Klik pada tombol OK. Anda akan melihat hasil berikut di kanvas:

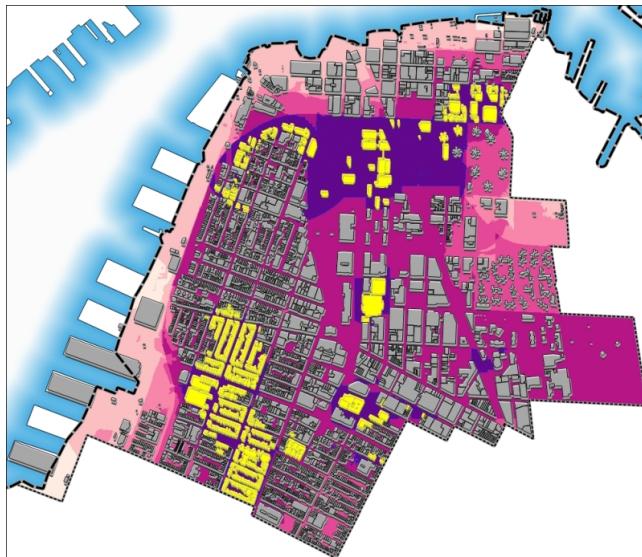
Demikian pula, Anda dapat mencoba jenis kueri lain dan mengidentifikasi bangunan yang tepat yang memenuhi kondisi kesesuaian. Dalam hal ini, jendela dialog akan terlihat sebagai berikut:

Setelah mengklik tombol **OK**, Anda akan melihat hasil berikut di kanvas peta:

Perhatikan bahwa saat ini, kita telah memilih bangunan yang sepenuhnya berada dalam area kesesuaian maksimum. Jadi, kita siap memberikan jawaban yang solid dan spesifik untuk pertanyaan, “Tempat mana yang terbaik untuk tinggal?”



Gambar 5.49: Exact buildings from selected maximum suitability



Gambar 5.50: Final result

Bagian III

WebGIS

Bab 6

GeoServer

6.1 Pendahuluan

Konsep WebGIS

GIS berbasis internet telah dikenal luas baik dalam organisasi publik maupun swasta sebagai alat mendasar untuk penyimpanan dan distribusi data ke audiens yang ditargetkan (Brovelli et al. 2016). Portal Web berbasis GIS menyediakan antarmuka terpusat dan seragam untuk mengakses sumber daya yang terdistribusi dan beragam dan layanan data (Karnatak et al. 2007). Menurut (Peng 2001), GIS Internet mengacu pada alat GIS berbasis jaringan yang menggunakan Internet sebagai sarana utama untuk menyediakan akses ke data yang didistribusikan dan informasi lainnya, menyebarkan informasi spasial, dan melakukan analisis GIS.

GIS Web Service

Layanan Web GIS menyediakan akses langsung ke data, menghilangkan kebutuhan untuk mengunduh dataset dan mengimportnya ke aplikasi desktop Anda, sebagai gantinya menggunakan antarmuka HTTP sederhana (URL) untuk mengakses data. Departemen ini menggunakan berbagai protokol layanan termasuk Web Map Service (WMS), Web Map Tile Service (WMPS), dan Web Feature Service (WFS) yang disediakan oleh Open Geospatial Consortium (OGC) dan layanan pemetaan Web ESRI.

Layanan Web untuk Data Spasial

Layanan web untuk data spasial memungkinkan menggunakan data dari server secara langsung dengan perangkat lunak GIS desktop, tanpa mengunduh data pertama sebagai file ke komputer sendiri. Sedemikian rupa, dimungkinkan untuk menggunakan data yang selalu terbaru dengan mudah. Semua produk

perangkat lunak desktop GIS komersial dan open-source yang paling umum mendukung standar Web (MapInfo, ArcGIS, QGIS, dan GRASS).

Layanan web digunakan sehingga pengguna terhubung ke layanan menggunakan menu khusus. Untuk koneksi, pengguna perlu mengetahui URL server. Setelah terhubung ke server, pengguna mendapat daftar lapisan peta yang tersedia. Layanan web juga mudah digunakan dalam aplikasi peta di Web. Data dapat diminta juga secara langsung dengan HTTP GET atau permintaan POST.

Standar layanan Web yang paling umum dari Open Geospatial Consortium's (OGC) adalah: Layanan Web Map (WMS), Layanan Web Map Tile (WMTS), Layanan Fitur Web (WFS), dan Layanan Cakupan Web (WCS). WMS dan WMTS mengembalikan gambar peta dalam format raster, data WFS dalam format vektor, dan data WCS dalam format raster. Dalam WMTS, peta hanya tersedia dalam skala dan ukuran yang ditentukan sebelumnya. Dalam WMS, skala dan ukuran peta dapat diatur tanpa batasan. WMTS, lebih cepat, karena seringkali petak peta sudah siap di server. Untuk meminta hanya sebagian data filter yang berbeda dapat digunakan, misalnya, BBOX menentukan bidang yang diminati.

Servise WebGIS

Tantangan utama data geospasial (Shengru dan Abdelguerfi, 2006) adalah: Data geospasial berukuran besar; rumitnya pemuat dan pemetaan data ke klien ; memuat ulang peta dan data membutuhkan waktu; data geospasial sangat heterogen, dan masalah geospasial yang kompleks membutuhkan sejumlah besar data geospasial dari berbagai sumber dan lokasi.

Komponen WebGIS

Komponen utama dari WebGIS dikategorikan ke dalam alat navigasi, alat legenda, alat pencarian, dan alat lainnya seperti alat menambahkan lapisan WMS, manajer lapisan. Fitur pada peta dapat diidentifikasi menggunakan alat identifikasi peta (Identifier). Alat ukur jarak dari peta juga telah dimasukkan sebagai bagian dari alat analisis peta spasial. Hasil permintaan peta atau bidang yang diinginkan dapat dicetak bersama dengan legenda detail memakai alat cetak. Ukuran, jenis font, dan output peta dapat dikustomisasi menggunakan alat ini. Peta dapat diproduksi dalam berbagai format file seperti PNG, JPEG, GIF, atau PDF. Zoom khusus kawasan dimungkinkan dengan menggunakan alat zoom cepat.

Pencarian atribut non-spasial juga dapat dilakukan oleh pengguna. Untuk melakukan ini, pengguna pertama-tama memilih layer dan kemudian, mengisi dari bidang / kolom yang terdaftar. Atribut yang akan dicari dimasukkan dalam kotak teks. Mengklik “Cari berdasarkan Nilai” akan membuat jendela sembulan yang mencantumkan tabel atribut yang cocok dengan kriteria pencarian dan selanjutnya memperbesar peta dan disorot.

Identifikasi otomatis peta pada pergerakan kursor mouse menggunakan widget “Identifikasi Otomatis”. Pengguna memilih layer dan atributnya yang sesuai,

setelah diterapkan; pengguna melihat atribut yang sesuai dengan lokasi peta ketika kursor mouse bergerak. Penambahan lapisan WMS juga dimungkinkan di portal WebGIS ini. Pertama-tama kita meng-host lapisan yang memiliki proyeksi serupa melalui GeoServer (Kamel dan Honda 2006).

6.2 Komponen yang diperlukan

Sebelum Anda dapat menggunakan GeoServer, Anda perlu menginstal beberapa perangkat lunak yang dibutuhkan. GeoServer merupakan aplikasi Java. Salah satu hal penting yang paling Anda pastikan adalah virtual Java bekerja pada mesin Anda.

Ada dua paket utama Java. Tergantung pada apa yang Anda berencana untuk melakukan dengan Java, Anda mungkin ingin menginstal sebuah JDK (Java Development Kit) atau JRE (Java Runtime Environment).

JDK memungkinkan Anda untuk mengkompilasi kode Java, sedangkan JRE cukup memenuhi syarat untuk menjalankan sebagian besar aplikasi Java. Mulai dari rilis 2.0, GeoServer tidak perlu instalasi JDK penuh, dan Anda dapat memilih JRE. JDK hanya diperlukan jika Anda berencana untuk menulis dan mengkompilasi kode Java. Hal ini dilakukan jika Anda ingin memodifikasi kode sumber GeoServer, untuk memperbaiki kode, atau menambahkan fungsionalitas.

Versi saat ini dari GeoServer membutuhkan Java 8. Perlu diingat bahwa Java 9 tidak didukung, mungkin Anda dapat menginstal Geoserver. Hanya saja Anda tidak akan mendapatkan dukungan untuk setiap masalah yang Anda temukan saat menggunakan Geoserver dengan Java 9.

Untuk melakukan pemeriksaan apakah sistem Anda memiliki Java JRE 8, silahkan ikuti langkah berikut :

1. Dari menu Start, pilih Control Panel.
2. Pilih Programs. Jika sistem anda memiliki JRE / JDK, Anda akan melihat ikon dengan logo Java, seperti yang ditunjukkan pada gambar berikut. Ini adalah cara pintas ke panel kontrol Java:

6.3 Instalasi

6.3.1 Tomcat

Paket Tomcat yang digunakan dalam pelatihan ini adalah paket yang disediakan oleh Bitnami. Paket Bitnami Tomcat versi MS Windows tersebut



Gambar 6.1: Cek Java

dapat diunduh di tautan berikut <https://bitnami.com/redirect/to/654730/bitnami-tomcatstack-7.0.96-0-windows-x64-installer.exe>

Tahapan instalasi Bitnami Tomcat dapat dilihat pada video berikut :

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is
```

Instalasi Tomcat

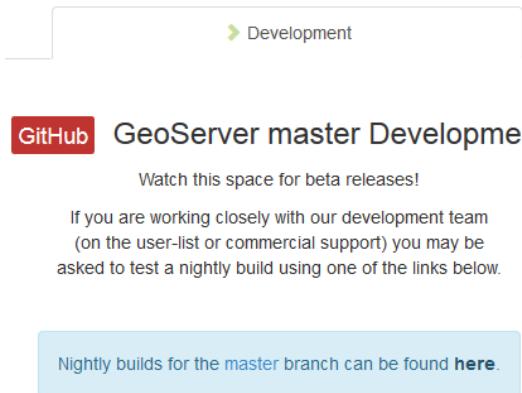
6.3.2 GeoServer

GeoServer dapat diunduh di halaman tautan berikut <http://geoserver.org/download/> dan pilihlah tipe instalasi yang tersedia. Anda akan menemukan beberapa paket berbeda untuk GeoServer. Kita akan menggunakan versi Arsip Web (.war):

Choose a version of GeoServer to download.	
Production	Development
Archived	
Stable GeoServer 2.11.0 ⓘ	Maintenance GeoServer 2.10.2 ⓘ
The recommended release or GeoServer tested and supported by the community.	
GeoServer 2.10 releases:	
2.10.2 2.10.1 2.10.0 2.10-RC1 2.10-beta 2.10-M0	
Nightly builds for the 2.11.x series can be found here .	
Nightly builds for the 2.10.x series can be found here .	

Anda dapat memilih rilis dari dua cabang yang berbeda - **Stable** dan **Maintenance**. Keduanya dibangun untuk keperluan produksi, sehingga Anda dapat memilih apa pun yang Anda inginkan. Kami sarankan Anda memilih rilis terbaru karena berisi semua fitur baru yang baru saja dirilis oleh tim pengembang.

Terlepas dari Production tab, Anda dapat memilih dua cabang lainnya - **Development** dan **Archived**. Di dalam bagian Development, Anda akan menemukan **Night Build**, rilis ini tidak cocok untuk produksi karena rentan terhadap *bug*, tetapi berisi semua perubahan yang dilakukan pengembang pada kode sumber, sehingga Anda dapat menggunakan untuk menguji perbaikan bug yang Anda temukan:



Archived berisi rilis yang lebih lama. Ini mungkin berguna jika Anda menggunakan rilis GeoServer lama dan perlu memutakhirkannya ke yang lebih baru, tetapi Anda tidak ingin melompat ke yang terakhir. Kasus penggunaan lain mungkin jika Anda menggunakan GeoServer di lingkungan di mana versi JRE lama, 7.x atau 6.x, tersedia dan Anda tidak dapat memutakhirkannya.

Tahapan pengunduhan GeoServer dapat dilihat pada video berikut :

Download GeoServer

Di bagian berikutnya, kita akan melakukan instalasi GeoServer dengan cara men-*deploy* arsip web di Apache Tomcat. Server aplikasi Java hampir dapat dijalankan pada sistem operasi apa pun dengan cara yang sama. Begitu pula dengan GeoServer yang di-*deploy* pada Tomcat.

Proses instalasi GeoServer adalah sebagai berikut:

1. Pastikan bahwa Tomcat Server berjalan dengan membuka tautan `http://localhost/index.jsp`. Apabila isi dari halaman tersebut seperti pada Gambar ??, maka kita dapat melanjutkan ke proses berikutnya.
2. Buka tautan **Manager App**. Ketikkan username dan password seperti yang dibuat ketika instalasi.
3. Unggah berkas `geoserver.war` di bagian WAR file to deploy.
4. Klik Deploy.
 - Apabila proses ini gagal dengan pesan kesalahan yang menunjukkan bahwa ukuran berkas melebihi yang ditentukan, edit berkas `web.xml` yang ada di direktori ***C:/Bitnami/tomcatstack-7.0.96-0/apache-tomcat/webapps/manager/WEB-INF***
 - Ubah ukuran max-file-size dan max-request agar lebih besar dari 100 MB, dengan menambahkan angka 1 di depan angka 8, sehingga menjadi 183886080.

<multipart-config>



Gambar 6.2: Stop Tomcat Server



Gambar 6.3: Start Tomcat Server

```
<!-- 50MB max -->
<max-file-size>83886080</max-file-size>
<max-request-size>83886080</max-request-size>
<file-size-threshold>0</file-size-threshold>
</multipart-config>
```

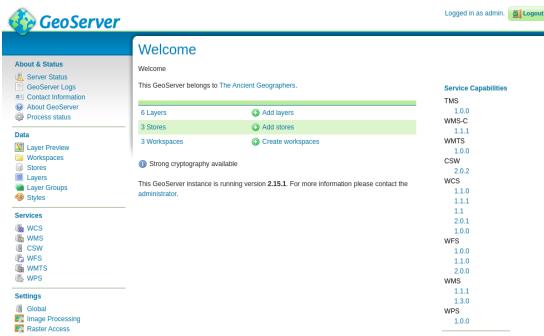
- Simpan berkas web.xml
 - Stop Tomcat Server
 - Jalankan ulang Tomcat Server.
 - Ulangi kembali proses Deploy.
5. Apabila proses Deploy Geoserver berhasil, maka menu Manager App akan terlihat /**geoserver** di dalam daftar aplikasi Tomcat.

6.4 Menjalankan GeoServer

Start GeoServer

Applications						
Path	Version	Display Name	Running	Sessions	Commands	
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Expire sessions with idle 30 minutes	
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Expire sessions with idle 30 minutes	
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Expire sessions with idle 30 minutes	
/geoserver	None specified	GeoServer	true	0	Start Stop Reload Expire sessions with idle 30 minutes	

Gambar 6.4: Folder aplikasi GeoServer



Gambar 6.5: Dashboard GeoServer

6.5 Administrasi GeoServer

Dalam bab ini, kita akan menjelajahi antarmuka administrasi GeoServer. Ini adalah konsol utama tempat Anda dapat mengontrol hampir semua pengaturan instalasi GeoServer Anda. Dalam rilis pertama, itu sedikit rumit, tetapi karena seri 2.x, nama menu dan ikon konsisten di setiap bagian. Ada juga antarmuka yang ditingkatkan untuk GeoWebCache terintegrasi, di mana Anda dapat melakukan hampir semua konfigurasi caching dari antarmuka GeoServer. Kabar baiknya adalah bahwa kita akan menggunakan mouse lebih banyak di sini daripada bab lain, sehingga keyboard akan mendapatkan istirahat.

Dalam bab ini, kita akan membahas semua bagian dari Antarmuka Administrasi Web. Secara khusus, kami akan mengeksplorasi topik berikut secara rinci:

Memeriksa status GeoServer Anda dan mencatat konten file Mempratinjau data Anda Pengaturan tuninguntuk Layanan Peta Web (WMS), Layanan Fitur Web (WFS), dan Layanan Cakupan Web (WCS) Tuning global pengaturan Melakukan permintaan melalui antarmuka demo

6.5.1 Dashboard

Mengakses antarmuka hanya membutuhkan Anda untuk membuka browser dan arahkan ke `http://localhost:8080/geoserver/web`. Jika Anda telah membaca bab sebelumnya dan menjalankan contoh-contohnya, Anda harus mengubah kata sandi default untuk pengguna admin.

Anda bisa menggunakan yang baru untuk login lagi di GeoServer; kita sekarang akan memusatkan perhatian kita pada tata letak antarmuka.

Pertimbangkan tangkapan layar berikut. Anda dapat mengenali tiga area utama di antarmuka web GeoServer:

Area pusat menunjukkan Anda beberapa informasi tentang status GeoServer - elemen di dalamnya berubah sesuai dengan operasi yang Anda lakukan. Tepat setelah Anda masuk, ini menunjukkan kepada Anda detail singkat dari data yang dikonfigurasi, dan peringatan atau kesalahan yang harus Anda koreksi. Baris terakhir menunjukkan nomor rilis dan ada tautan ke kotak surat administrator; ini default ke ahli geografi kuno yang terkenal sampai Anda memasukkan data Anda.

Di sisi kanan, ada daftar yang menunjukkan kemampuan GeoServer kepada Anda. Akronim yang tercantum mengacu pada protokol OGC standar; kita akan membicarakan beberapa dari mereka secara terperinci dalam buku ini, dan masing-masing dari mereka memiliki setidaknya satu rilis yang didukung. Angka-angka adalah tautan ke dokumen XML yang secara tepat menggambarkan data dan operasi yang didukung oleh setiap protokol. Mereka adalah sumber daya yang sangat berharga bagi klien yang mau menggunakan layanan Anda.

Cobalah untuk membuka WMS 1.1.1tautan dan buka hasilnya dalam editor teks. Anda dapat dengan mudah memahami beberapa bagian. Pertimbangkan yang berikut ini:

Yang pertama berisi deskripsi layanan yang dibuka:

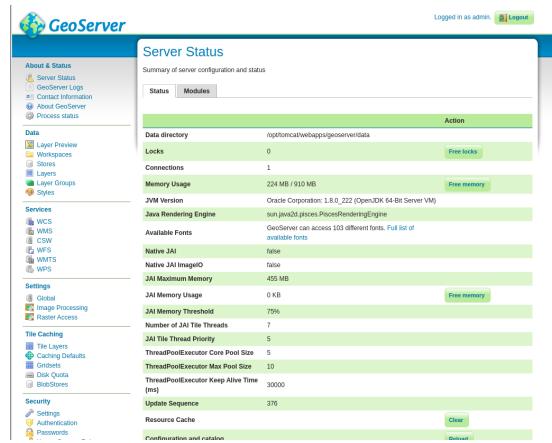
```
<Service>
<Name>OGC:WMS</Name>
<Title>GeoServer Web Map Service</Title>
```

Kemudian Anda dapat menemukan beberapa informasi tentang administrator:

```
<ContactInformation>
  <ContactPersonPrimary>
    <ContactPerson>Stefano Iacovella</ContactPerson>
    <ContactOrganization>Packt Publishing
      Ltd.</ContactOrganization>
  </ContactPersonPrimary>
<ContactPosition>Chief geographer</ContactPosition>
```

Daftar operasi yang didukung dan URL dasar untuk permintaan adalah sebagai berikut:

```
<Request>
  <GetCapabilities>
  <Format>application/vnd.ogc.wms_xml</Format>
  <DCPType>
  <HTTP>
    <Get>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple" xlink:href="http://localhost:8080
        /geoserver/wms?SERVICE=WMS&gt;"/>
    </Get>
  <Post>
```

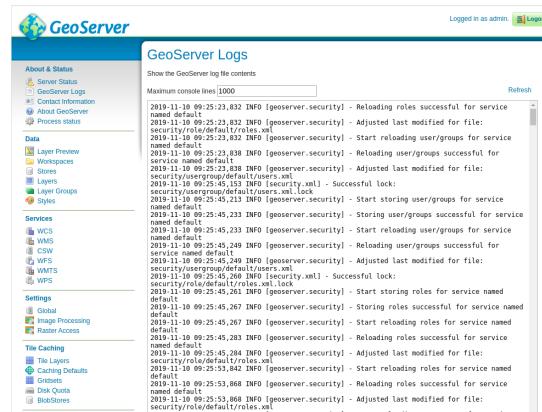


Gambar 6.6: Status server

```

<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="simple" xlink:href="http://localhost:8080
    /geoserver/wms?SERVICE=WMS&amp;;" />
</Post>
</HTTP>
...
</GetCapabilities>
<GetMap>
...
</GetMap>
<GetFeatureInfo>
...
</GetFeatureInfo>
<DescribeLayer>
...
</DescribeLayer>
<GetLegendGraphic>
...
</GetLegendGraphic>
<GetStyles>
...
</GetStyles>
</Request>

```



Gambar 6.7: Status server

6.5.2 About dan Status

6.5.2.1 Server Status

6.5.2.2 Logs

6.5.3 Data

Bagian ini berisi tautan ke mesin konfigurasi data. Seperti yang Anda duga, untuk menerbitkan peta di internet dengan GeoServer, Anda perlu menambahkan data spasial ke dalamnya. Pertimbangkan tangkapan layar berikut:

Di area ini, Anda dapat mengonfigurasi akses data dan cara aksesnya ke klien:

####Layer Preview

membuka formulir termasuk setiap lapisan yang diterbitkan di GeoServer. Selain itu, jika Anda belum menambahkan data ke instalasi GeoServer Anda, Anda akan menemukan beberapa lapisan sampel sudah terdaftar. Mengklik pada OpenLayerstautan, yang ditempatkan di sebelah kanan nama lapisan, Anda dapat membuka contoh aplikasi web untuk melihat seperti apa data Anda terlihat. Suka.

The Keyhole Markup Language (KML) link membiarkan Anda men-download data dalam format yang sesuai untuk pratinjau di Google Earth. Ada juga beberapa format lain yang tersedia, tercantum dalam kotak drop-down di ujung kanan baris. Pertimbangkan tangkapan layar berikut:



Gambar 6.8: Layer preview

6.5.4 Layanan

6.5.5 Pengaturan

6.5.6 Tile Caching

6.5.7 Security

6.6 Menambahkan Data

Anda akan mempelajari tipe data apa yang dapat Anda gunakan dengan GeoServer. Kami akan mengambil ikhtisar singkat tentang format yang didukung, baik bawaan maupun melalui ekstensi, dan cara menambahkannya ke konfigurasi Anda. Lebih khusus lagi, kami akan memuat data dari shapefile, tabel PostGIS, dan tabel Oracle menggunakan data sensus AS.

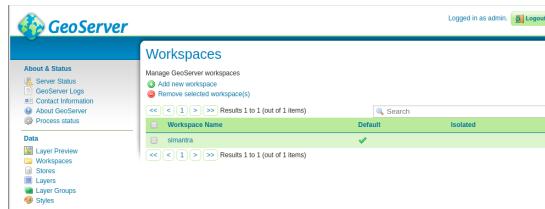
6.6.1 Pengaturan Data

6.6.1.1 Workspace

Bagian ini menjelaskan cara melihat dan mengkonfigurasi ***workspace***. Dianalogikan dengan ***namespace***, ***workspace*** adalah wadah yang mengatur item lainnya. Di GeoServer, ***workspace*** sering digunakan untuk mengelompokkan layer yang sama. Layer dapat dirujuk dengan nama ruang kerja mereka, titik dua, nama lapisan (misalnya sf:archsites). Dua layer yang berbeda dapat memiliki nama yang sama selama mereka memiliki ***workspace*** yang berbeda (misalnya sf:states dan topp:states).

Untuk melihat atau mengedit workspace, klik nama workspace. Halaman konfigurasi workspace akan ditampilkan pada Gambar 6.10

Workspace didefinisikan oleh nama dan URI Namespace (Uniform Resource Identifier). Nama workspace terbatas hingga sepuluh karakter dan tidak boleh memiliki spasi. URI mirip dengan URL, kecuali URI tidak perlu menunjuk ke



Gambar 6.9: Workspaces

Edit Workspace

Edit existing workspace

Name
batam_pg

Namespace URI
batam_pg

The namespace uri associated with this workspace

Default Workspace
 Isolated Workspace

Settings

Enabled

Services

- WCS
- WPS
- WMS
- CSW
- WFS
- WMPS

Save Cancel

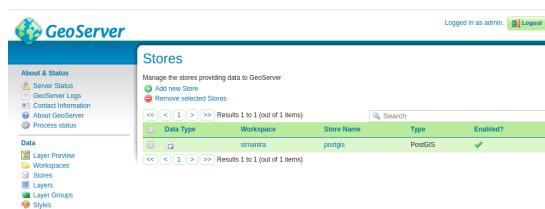
Gambar 6.10: Edit Workspace

lokasi aktual di web, dan hanya perlu menjadi pengidentifikasi unik. Untuk URI Workspace, kami sarankan menggunakan URL yang terkait dengan proyek Anda, dengan pengidentifikasi jejak yang berbeda. Misalnya, http://localhost/batam_pg adalah URI untuk ruang kerja “batam_pg”.

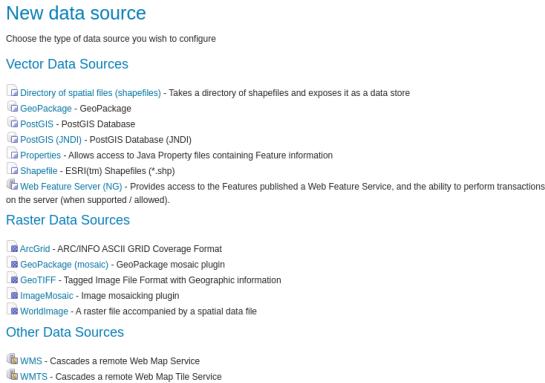
6.6.1.2 Store

6.6.2 Penambahan Data

Data yang dapat diunggah ke dalam GeoServer dapat bermacam-macam, di antaranya adalah direktori Shapefiles, GeoPackage, PostGIS, GeoTIFF, WMS



Gambar 6.11: Store



Gambar 6.12: Workspaces

The screenshot shows the 'New Workspace' configuration dialog. It includes fields for 'Name' (containing 'sf'), 'Namespace URI' (containing 'san_fransisco'), and checkboxes for 'Default Workspace' and 'Isolated Workspace'. At the bottom are 'Submit' and 'Cancel' buttons.

Gambar 6.13: Workspaces

dan lainnya seperti disajikan pada Gambar ??.

6.6.2.1 Direktori Shapefiles

Untuk menambah data ke dalam GeoServer, lakukan tahapan berikut

1. Buatlah workspace baru.

Kita akan menambahkan workspace berdasarkan data sampel GeoServer yaitu San Fransisco. Isikan ***sf*** sebagai nama workspace baru dan ***san_fransisco*** untuk Namespace_URI. Jangan lupa untuk klik **Submit** untuk membuat workspace.

2. Tambahkan Stores baru.

- Klik Stores di panel sebelah kiri, dan klik tautan **Add new Store**. Pilih **workspace sf** dan isikan ***sf_shp*** untuk **Data Source Name**.

New Vector Data Source

Add a new vector data source

Directory of spatial files (shapefiles)
Takes a directory of shapefiles and exposes it as a data store

Basic Store Info

Workspace * sf

Data Source Name * sf.shp

Description

Enabled

Connection Parameters

Directory of shapefiles * file:data/sf [Browse...](#)

DBF files charset ISO-8859-1

Create spatial index if missing/outdated

Use memory mapped buffers (Disable on Windows)

Cache and reuse memory maps (Requires 'Use Memory mapped buffers' to be enabled)

[Save](#) [Cancel](#)

Gambar 6.14: Sumber data vektor

New Layer

Add a new layer

You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#)

Here is a list of resources contained in the store 'sf_shp'. Click on the layer you wish to configure

Published	Layer name	Action
	archsites	Publish
	bugsites	Publish
	restricted	Publish
	roads	Publish
	streams	Publish

<< < | 1 | > >> Results 1 to 5 (out of 5 items) [Search](#)

Gambar 6.15: Memilih Layer baru

- Untuk **Connection Parameters** klik Browse untuk memilih direktori di mana vektor shapefiles tersimpan. Untuk kasus ini, klik direktori **data/data/sf**.
 - Klik Save.
3. Pilih Layer yang akan diunggah dengan mengklik tautan **Publish**, sebagai contoh **archsites**.
 4. Definisikan data layer dan publikasinya.
 - Isikan nama layer, misalnya **archsites**.
 - Klik **Compute from data** untuk mengisi **Bounding Boxes**, dan klik **Compute from native bounds** untuk **Lat/Lon Bounding Box**.
 - Klik **Save** untuk menyimpan data.

Edit Layer

Edit layer data and publishing

sf:archsites

Configure the resource and publishing information for the current layer

Data **Publishing** **Dimensions** **Tile Caching**

Edit Layer

Basic Resource Info

Name
archsites

Enabled

Advertised

Title
archsites

Abstract

Keywords

Current Keywords
features
archsites

New Keyword

Vocabulary

Gambar 6.16: Edit layer name

Metadata links

No metadata links so far

Note only FGDC and TC211 metadata links show up in WMS 1.1.1 capabilities

Data links

No data links so far

Coordinate Reference Systems

Native SRS
EPSG:26713

Declared SRS
EPSG:26713 EPSG:NAD27 / UTM zone 13N...
SRS handling
Force declared

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
569,860	4,914,479	608,355	4,926,490

Compute from data
Compute from SRS bounds

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
-103.87237752497	44.377319797073	-103.63775627308	44.487959052494

Compute from native bounds

Curved geometries control

Linear geometries can contain circular arcs

Linearization tolerance (useful only if your data contains curved geometries)

Gambar 6.17: Parameter Bounding Boxes

Feature Type Details

Property	Type	Nullable	Min/Max Occurrences
the_geom	Point	true	0/1
cat	Long	true	0/1
str1	String	true	0/1

Reload feature type --

Restrict the features on layer by CQL filter

Gambar 6.18: Klik Save

6.6.2.2 PostGIS

6.6.2.3 Raster

6.7 Mengakses Layer

Bab 7

MapStore2

MapStore adalah kerangka kerja webgis open source yang sangat modular, dikembangkan oleh GeoSolutions untuk membuat, mengelola, dan berbagi peta dan mashup secara aman. Kerangka kerja yang sederhana dan intuitif ini menggabungkan konten yang disajikan dari server seperti Google Maps, OpenStreetMap, Bing atau dari server lain yang mengikuti standar OGC seperti WFS, CSW, WMS, WMTS, dan TMS. MapStore2 diinstal dan dikonfigurasi pada server aplikasi Web, memungkinkan banyak pengguna mengakses situs menggunakan peramban web. MapStore2 digunakan untuk mencari, memvisualisasikan, dan melakukan kueri data geospasial yang diterbitkan dan untuk mengintegrasikan berbagai sumber ke dalam tampilan peta terintegrasi yang dapat dengan mudah dinavigasi. Selain itu, dengan merilis versi baru, perangkat lunak ini memungkinkan pengguna untuk membuat dasbor untuk hosting widget seperti peta mini, grafik statistik, tabel dan banyak lagi.

MapStore2 bertujuan untuk menjadi kerangka kerja produk dan webgis. Sebagai produk geoportal standar, MapStore2 adalah produk berbasis web untuk visualisasi dan analisis peta yang memungkinkan Anda membangun situs web interaktif geospasial atau layanan web. MapStore2 memungkinkan akses langsung dan real-time ke penyimpanan data geospasial dari semua format yang didukung. Ini juga menyediakan semua fungsi analisis spasial. Anda dapat menggunakan untuk membangun situs web yang memberikan kepada pengguna, dengan tidak lebih dari browser web, aplikasi geospasial yang kuat, dinamis, dan terbuka, yang sebelumnya hanya tersedia dalam aplikasi desktop. MapStore2 juga merupakan *framework* dalam arti dapat digunakan sebagai titik awal untuk membangun aplikasi geospasial yang canggih. Ia tidak bergantung secara eksplisit pada *Engine Map* tertentu, tetapi ia dapat bekerja dengan OpenLayers , LeafletJS dan Cesium 3D viewer, yang memastikan fleksibilitas terbesar ketika seseorang ingin menggunakan sebagai *framework*.

MapStore2 telah dirancang sejak awal untuk memberi pengguna pengalaman

yang koheren dan komprehensif di berbagai perangkat, oleh karena itu ia harus secara otomatis beradaptasi dengan ukuran layar yang berbeda seperti yang ditunjukkan nanti.

MapStore2 dikembangkan dengan basis teknologi pada OpenLayers, Leaflet dan ReactJS, dan dilisensikan di bawah lisensi Simplified BSD.

7.1 Quick Start

Anda dapat memilih untuk mengunduh paket biner mandiri atau file WAR untuk menggunakan MapStore.

7.1.1 Paket Biner

Cara termudah untuk mencoba MapStore adalah dengan mengunduh dan meng-ekstrak paket biner yang tersedia di halaman rilis MapStore . Di sini Anda dapat menemukan beberapa peta yang telah dikonfigurasikan sebelumnya serta pengguna dan grup. Tujuan dari paket ini adalah untuk memudahkan semua persyaratan yang diperlukan agar Anda mengambil MapStore untuk test-drive.

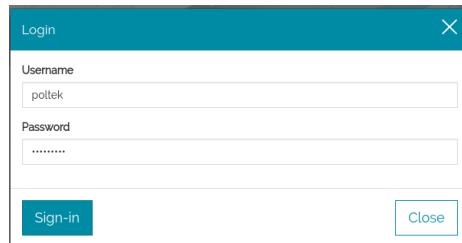
Bagaimana cara menjalankannya

1. Buka lokasi tempat Anda menyimpan file zip, unzip kontennya dan jalankan:
 - Windows: mapstore2_startup.bat
 - Linux: ./mapstore2_startup.sh
2. Arahkan browser Anda ke: <http://localhost:8082/mapstore>
3. Untuk menghentikan MapStore cukup lakukan:
 - Windows: mapstore2_shutdown.bat
 - Linux: ./mapstore2_shutdown.sh

7.1.2 Arsip Web (WAR)

Setelah mengunduh file war MapStore, instal dalam wadah web java Anda (mis. Tomcat), dengan prosedur biasa untuk wadah tersebut (biasanya Anda hanya perlu menyalin file war di subfolder webapps).

Jika Anda tidak memiliki wadah web java, Anda dapat mengunduh Apache Tomcat dari sini dan menginstalnya. Anda juga akan membutuhkan Java7 JRE .



Gambar 7.1: Open Attribute Table

Kemudian Anda dapat mengakses MapStore menggunakan URL berikut (dengan asumsi wadah web pada port 8080 standar):

`http://localhost:8080/mapstore`

Gunakan kredensial default (admin / admin) untuk masuk dan mulai membuat peta Anda!

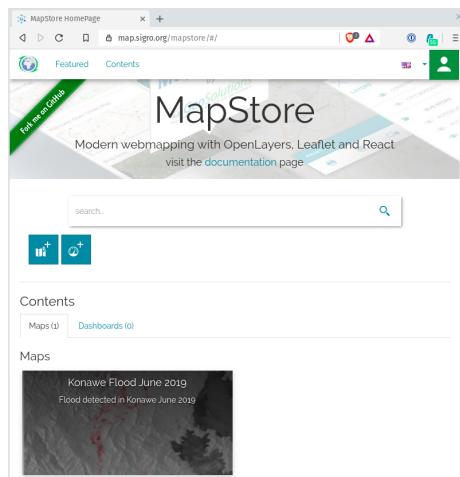
7.2 Menggunakan MapStore2

1. Login

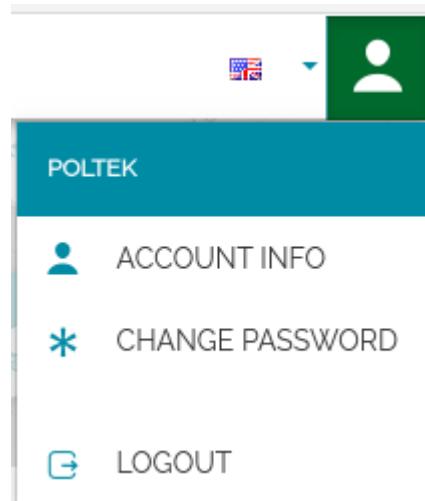
Untuk login ke dalam MapStore2 yang terinstal dalam komputer Anda, gunakan tautan berikut `http://localhost/mapstore/`. Untuk menggunakan web SIGRO, silakan kunjungi `https://map.sigro.org/mapstore`. Gunakan kredensial ‘poltek’ untuk login.

2. Tampilan awal MapStore2
3. Informasi Pengguna

Klik ikon di bagian kanan atas dasbor untuk mengetahui informasi pengguna.



Gambar 7.2: Open Attribute Table



Gambar 7.3: Open Attribute Table

Bagian IV

Visualisasi dan Penyajian Data

Bab 8

Visualisasi Data

8.1 QGIS

Mengumpulkan dan mengatur data dari sumber yang berbeda hanya setengah dari pekerjaan. Langkah selanjutnya adalah mempresentasikannya di peta, mengungkapkan konten dan fitur tematik secara akurat. Untuk tujuan ini, berbagai teknik desain visual, atau lapisan penataan berdasarkan atributnya, digunakan dalam GIS. QGIS memiliki kemampuan visualisasi kartografi dan penataan data yang luas dan fleksibel.

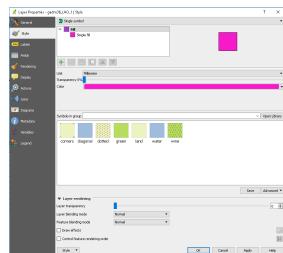
Di bagian ini, kita akan membahas topik dan keterampilan berikut:

- Good practices with respect to data organization within a single working document (project)
- Visual representation of data, revealing its thematic and spatial features
- Layer labeling, which is done to enhance data readability
- Control over styles
- Base maps used to provide spatial context and a background for your data

Hasil utama dari bagian ini adalah proyek, yang dirancang sesuai dengan persyaratan dasar visualisasi kartografi.

8.1.1 Grouping dan Ordering

Secara default, layer dimuat dalam urutan abjad (terbalik jika Anda menambahkannya dari panel browser). Setiap layer baru diletakkan di atas layer sebelumnya dan menutupinya. Semua lapisan dihidupkan secara default, ditata



Gambar 8.1: Layter Toolbar

dengan simbol seragam sederhana, dan diberi warna acak. Ini adalah alasan ketika orang yang berbeda memvisualisasikan data yang sama, mereka akan melihat warna yang berbeda. Urutan lapisan dapat diubah hanya dengan menyeret dan menjatuhkannya ke atas dan ke bawah legenda. Juga disarankan untuk memesan dan mengatur lapisan dalam beberapa kelompok logis, karena menyederhanakan navigasi dan pemahaman data. Khususnya dalam istilah GIS, lapisan bawah akan berupa fitur raster atau poligon. Di atas poligon akan ada fitur garis dan kemudian arahkan fitur di atas.

Untuk mengelola dan mengatur ulang layer dan mempertahankan visibilitasnya, gunakan Layer toolbar di panel Layers , yang diperlihatkan dalam screenshot berikut dan berisi tombol-tombol yang dijelaskan sesudahnya (sesuai urutan tampilan dari kiri ke kanan):

Tambah Grup : Ini membuat grup lapisan kosong.

Kelola Visibilitas Lapisan : Ini memungkinkan kami untuk dengan cepat menampilkan dan menyembunyikan lapisan dan juga menyesuaikan visibilitasnya dengan kombinasi lapisan yang telah ditentukan — yang disebut preset.

Filter Legenda Menurut Konten Peta : Jika filter aktif, legenda lapisan hanya menunjukkan item-item yang benar-benar terlihat di dalam kanvas peta. Semua simbol lainnya disembunyikan dari legenda.

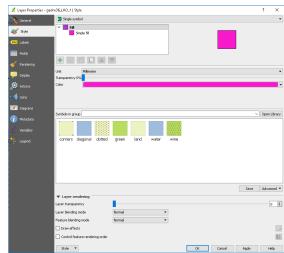
Rentangkan Semua / Tutup Semua : Tombol digunakan untuk membuka atau menutup lapisan, legenda simbologi, dan grup lapisan, jika ada.

Remove Layer / Group : Ini menghapus entri legenda yang dipilih.

Ada dua cara membuat grup layer:

Klik pada ikon tombol Tambah Grup , dari Layers toolbar. Grup baru akan muncul di bagian bawah daftar lapisan. Ketikkan nama yang sesuai lalu seret dan letakkan lapisan ke dalam grup. Hati-hati jika Anda menyorot grup tertentu, itu akan membuat Sub-grup baru di bawah grup yang aktif. Jika ini terjadi, Anda dapat menyeret sub-grup untuk menjadikannya sebagai grup.

Pilih beberapa layer sambil menahan tombol Ctrl, dan gunakan pintasan konstektual klik kanan Kelompok yang Dipilih untuk menempatkannya dalam satu



Gambar 8.2: Layer Property

grup.

Dimungkinkan untuk mengembangkan hierarki bertingkat dengan subkelompok dengan memilih grup dan menerapkan Add Group . Item apa pun dari legenda lapisan, apakah itu layer tunggal atau grup, dapat diganti namanya dengan nama pendek klik kanan. Mengganti nama tidak memengaruhi dataset itu sendiri, tetapi memungkinkan kami menerapkan padanya nama yang tepat dan bermakna dalam suatu proyek. Sekarang coba atur sendiri layer-layer itu menjadi beberapa grup yang bermakna, dan beri nama dengan tepat.

8.1.2 Styling

Dalam QGIS , gaya adalah cara visualisasi kartografi yang memperhitungkan fitur individu dan tematik layer. Ini mencakup karakteristik dasar simbologi, seperti warna dan keberadaan isian, parameter garis besar, penggunaan marker, rendering skala-tergantung, transparansi lapisan, interaksi dengan lapisan lain, dan label.

Gaya yang dipilih dengan baik sangat menyederhanakan persepsi dan keterbaaan data, sehingga penting untuk mempelajari cara bekerja dengan gaya agar dapat mewakili data Anda dengan cara terbaik. Dalam istilah sederhana, pelajari cara membuat peta, “eye-catching”. Pada bagian ini, kita akan membahas layer vektor dan raster secara terpisah, karena stilosiasi mereka memiliki beberapa fitur unik.

8.1.3 Vektor

The Style menu dari Lapisan Properti dialog menyediakan Anda dengan semua alat yang diperlukan untuk melambangkan dan gaya data Anda. Untuk membukanya, klik dua kali pada nama layer di panel Layers , atau gunakan Properties klik kanan kontekstual, dan pilih bagian Style . Anda akan melihat sesuatu yang mirip dengan 8.2.

8.2 GeoServer

8.2.1 Styling

8.2.2 Menggunakan SLD

Bab 9

Penyajian Data

Bibliografi