

Nama : Hanifah Hasanah

NIM : 123140082

LINK GITHUB :

https://github.com/HANIFAHHASANAH-123140082/sig_123140082_Hanifah-Hasanah.git

TUGAS:

Tugas Praktikum 2 ini merupakan lanjutan dari Praktikum 1 yang telah membuat database spasial menggunakan PostgreSQL dan ekstensi PostGIS. Pada praktikum ini, mahasiswa diwajibkan untuk:

- Menambahkan tabel jalan dengan tipe geometri LineString untuk menyimpan data jalan di sekitar tempat tinggal.
- Menambahkan tabel wilayah dengan tipe geometri Polygon untuk menyimpan data kelurahan di sekitar tempat tinggal.
- Melakukan konversi format geometri menggunakan fungsi ST_AsText() dan ST_AsGeoJSON().
- Melakukan validasi data geometri menggunakan fungsi ST_IsValid().
- Menampilkan semua data pada QGIS dengan layer berbeda.

LANGKAH PENGERJAAN:

1. Database yang digunakan merupakan lanjutan dari Praktikum 1 dengan nama database sig_123140082. Ekstensi PostGIS telah diaktifkan sebelumnya. Database ini diakses menggunakan pgAdmin 4 yang terhubung ke server PostgreSQL 15 di localhost port 5433.
2. Tabel jalan dibuat untuk menyimpan data geometri bertipe LineString yang merepresentasikan ruas-ruas jalan di sekitar wilayah ITERA, Lampung Selatan. Berikut adalah perintah DDL untuk membuat tabel:

```
CREATE TABLE jalan (
    id      SERIAL PRIMARY KEY,
    nama_jalan VARCHAR(100),
    jenis   VARCHAR(50),
    geom    GEOMETRY(LineString, 4326)
);
```

3. Dimasukkan 3 data jalan dengan titik-titik koordinat yang berada di sekitar kawasan ITERA, Lampung Selatan (koordinat WGS84):

```
INSERT INTO jalan (nama_jalan, jenis, geom) VALUES
(
    'Jl. Terusan Ryacudu',
    'Jalan Utama',
    ST_GeomFromText('LINESTRING(
        105.2280 -5.4540,
        105.2320 -5.4550,
        105.2360 -5.4560,
        105.2400 -5.4570
    )', 4326)
),
(
    'Jl. Pahlawan',
    'Jalan Kolektor',
    ST_GeomFromText('LINESTRING(
        105.2340 -5.4570,
        105.2355 -5.4580,
        105.2370 -5.4590,
        105.2385 -5.4600
    )', 4326)
),
(
    'Jl. Kesehatan',
    'Jalan Lokal',
    ST_GeomFromText('LINESTRING(
        105.2360 -5.4540,
        105.2370 -5.4550,
        105.2380 -5.4555,
        105.2390 -5.4560
    )', 4326)
);

```

```

CREATE TABLE jalan (
    id SERIAL PRIMARY KEY,
    nama_jalan VARCHAR(100),
    jenis VARCHAR(50),
    geom GEOMETRY(LineString, 4326)
);

INSERT INTO jalan (nama_jalan, jenis, geom) VALUES
(
    'Jl. Terusan Ryacudu',
    'Jalan Utama',
    ST_GeomFromText('LINESTRING(
        105.2280 -5.4540,
        105.2320 -5.4550,
        105.2360 -5.4560,
        105.2400 -5.4570
    )', 4326)
),

```

Error: relation "jalan" already exists
SQL state: 42P07

Data berhasil dimasukkan dengan 3 baris (3 ruas jalan).

4. Tabel wilayah dibuat untuk menyimpan data geometri bertipe Polygon yang merepresentasikan batas wilayah kelurahan di sekitar ITERA. Berikut adalah perintah DDL:

```

CREATE TABLE wilayah (
    id SERIAL PRIMARY KEY,
    nama_kelurahan VARCHAR(100),
    kecamatan VARCHAR(100),
    geom GEOMETRY(Polygon, 4326)
);

```

5. Dimasukkan 2 data wilayah kelurahan dengan batas Polygon yang tertutup (titik awal = titik akhir), sesuai kaidah geometri Polygon yang valid:

```

INSERT INTO wilayah (nama_kelurahan, kecamatan, geom) VALUES
(
    'Kelurahan Way Huwi',
    'Jati Agung',
    ST_GeomFromText('POLYGON((
        105.2280 -5.4520,
        105.2420 -5.4520,
        105.2420 -5.4610,
        105.2280 -5.4610,
        105.2280 -5.4520
    )', 4326)
),

```

```

(
    'Kelurahan Hajimena',
    'Natar',
    ST_GeomFromText('POLYGON((
        105.2420 -5.4520,
        105.2560 -5.4520,
        105.2560 -5.4610,
        105.2420 -5.4610,
        105.2420 -5.4520
    ))', 4326)
);

```

The screenshot shows a database management interface with a sidebar navigation menu and a main query editor.

Navigation Menu:

- > FTS Dictionaries
- > FTS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (4) (selected)
- > fasilitas_publik
- > jalan
- > spatial_ref_sys
- > wilayah (selected)
- > Trigger Functions
- > Types
- > Views
- > Subscriptions
- Login/Group Roles
- Tablespaces

Query Editor:

```

CREATE TABLE wilayah (
    id SERIAL PRIMARY KEY,
    nama_kelurahan VARCHAR(100),
    kecamatan VARCHAR(100),
    geom GEOMETRY(Polygon, 4326)
);

INSERT INTO wilayah (nama_kelurahan, kecamatan, geom) VALUES
(
    'Kelurahan Way Huwi',
    'Jati Agung',
    ST_GeomFromText('POLYGON((
        105.2280 -5.4520,
        105.2420 -5.4520,
        105.2420 -5.4610,
        105.2280 -5.4610,
        105.2280 -5.4520
    ))', 4326)
),
(
    'Kelurahan Hajimena',
    'Natar'
)

```

Data berhasil dimasukkan dengan 2 baris (2 wilayah kelurahan).

6. Fungsi ST_AsText() digunakan untuk mengkonversi geometri dari format binary internal PostGIS menjadi format WKT (Well-Known Text) yang dapat dibaca manusia. Berikut adalah query yang dijalankan beserta hasilnya:

Jalan:

Query Query History

```
1  SELECT id, nama_jalan, ST_AsText(geom) AS wkt_text
2  FROM jalan;
```

Data Output Messages Notifications

	id [PK] integer	nama_jalan character varying (100)	wkt_text text
1	1	Jl. Terusan Ryacudu	LINESTRING(105.228 -5.454,105.232 -5.455,105.236 -5.456,105.24 -5.457)
2	2	Jl. Pahlawan	LINESTRING(105.234 -5.457,105.2355 -5.458,105.237 -5.459,105.2385 -5...
3	3	Jl. Kesehatan	LINESTRING(105.236 -5.454,105.237 -5.455,105.238 -5.4555,105.239 -5...

Wilayah:

Query Query History

```
1  SELECT id, nama_kelurahan, ST_AsText(geom) AS wkt_text
2  FROM wilayah;
```

Data Output Messages Notifications

	id [PK] integer	nama_kelurahan character varying (100)	wkt_text text
1	1	Kelurahan Way Huwi	POLYGON((105.228 -5.452,105.242 -5.452,105.242 -5.461,105.228 -5.461,105.228 -5.4...
2	2	Kelurahan Hajimena	POLYGON((105.242 -5.452,105.256 -5.452,105.256 -5.461,105.242 -5.461,105.242 -5.4...

Hasil query menunjukkan bahwa format WKT berhasil ditampilkan untuk semua data. Tipe LINESTRING ditandai dengan daftar pasangan koordinat (longitude latitude), sedangkan tipe POLYGON memiliki koordinat yang membentuk cincin tertutup.

7. Fungsi `ST_AsGeoJSON()` digunakan untuk mengkonversi geometri menjadi format GeoJSON, yang merupakan standar format pertukaran data geospasial berbasis JSON dan umum digunakan dalam aplikasi web mapping.

Jalan:

The screenshot shows a PostgreSQL database interface. At the top, there are tabs for 'Query' and 'Query History'. Below the tabs is a button labeled 'Execute script' with an 'F5' key indicator. The main area contains a SQL query:

```
1 SELECT id, nama_jalan, ST_AsGeoJSON
2 FROM jalan;
```

Below the query results, there is a 'Data Output' section with tabs for 'Messages' and 'Notifications'. The 'Data Output' tab is selected. It features a toolbar with icons for creating new tables, files, and queries, as well as download and refresh buttons. The table header includes columns for 'id' (PK integer), 'nama_jalan' (character varying(100)), and 'geojson' (text). The data table displays three rows of street information:

	id [PK] integer	nama_jalan character varying(100)	geojson text
1	1	Jl. Terusan Ryacudu	{ "type": "LineString", "coordinates": [[105.228, -5.454], [105.232, -5.455], [105.236, -5.456], [105.24, -5.457...]
2	2	Jl. Pahlawan	{ "type": "LineString", "coordinates": [[105.234, -5.457], [105.2355, -5.458], [105.237, -5.459], [105.2385, -5.460...]
3	3	Jl. Kesehatan	{ "type": "LineString", "coordinates": [[105.236, -5.454], [105.237, -5.455], [105.238, -5.4555], [105.239, -5.460...]

Wilayah:

Query History

Execute script F5 AS geojson

```
1 SELECT id, nama_kelurahan, ST_AsGeoJSON AS geojson
2 FROM wilayah;
```

Data Output Messages Notifications

	id [PK] integer	nama_kelurahan character varying (100)	geojson text	
1	1	Kelurahan Way Huwi	{"type": "Polygon", "coordinates": [[[105.228,-5.452],[105.242,-5.452],[105.242,-5.461],[105.228,-5.461],[105.228,-5.452]]]}	
2	2	Kelurahan Hajimena	{"type": "Polygon", "coordinates": [[[105.242,-5.452],[105.256,-5.452],[105.256,-5.461],[105.242,-5.461],[105.242,-5.452]]]}	

Hasil query menunjukkan bahwa semua data geometri berhasil dikonversi ke format GeoJSON. Format ini menggunakan struktur objek JSON dengan atribut type dan coordinates yang dapat langsung digunakan oleh aplikasi web seperti Leaflet, Mapbox, dan sebagainya.

8. Fungsi ST_IsValid() digunakan untuk memeriksa apakah geometri yang tersimpan dalam database merupakan geometri yang valid sesuai standar OGC (Open Geospatial Consortium). Fungsi ini mengembalikan nilai boolean: TRUE jika valid, FALSE jika tidak valid.

Jalan:

The screenshot shows a database management interface with a toolbar at the top containing various icons for file operations, search, and navigation. Below the toolbar, a menu bar has 'Query' selected. The main area contains a SQL query:

```
1 SELECT id, nama_jalan, ST_IsValid(geom) AS is_valid
2 FROM jalan;
```

Below the query results, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with three rows of data:

	id [PK] integer	nama_jalan character varying (100)	is_valid boolean
1	1	Jl. Terusan Ryacudu	true
2	2	Jl. Pahlawan	true
3	3	Jl. Kesehatan	true

Wilayah:

The screenshot shows a database management interface with a toolbar at the top containing various icons for file operations, search, and navigation. Below the toolbar, a menu bar has 'Query' selected. The main area contains a SQL query:

```
1 SELECT id, nama_kelurahan, ST_IsValid(geom) AS is_valid
2 FROM wilayah;
```

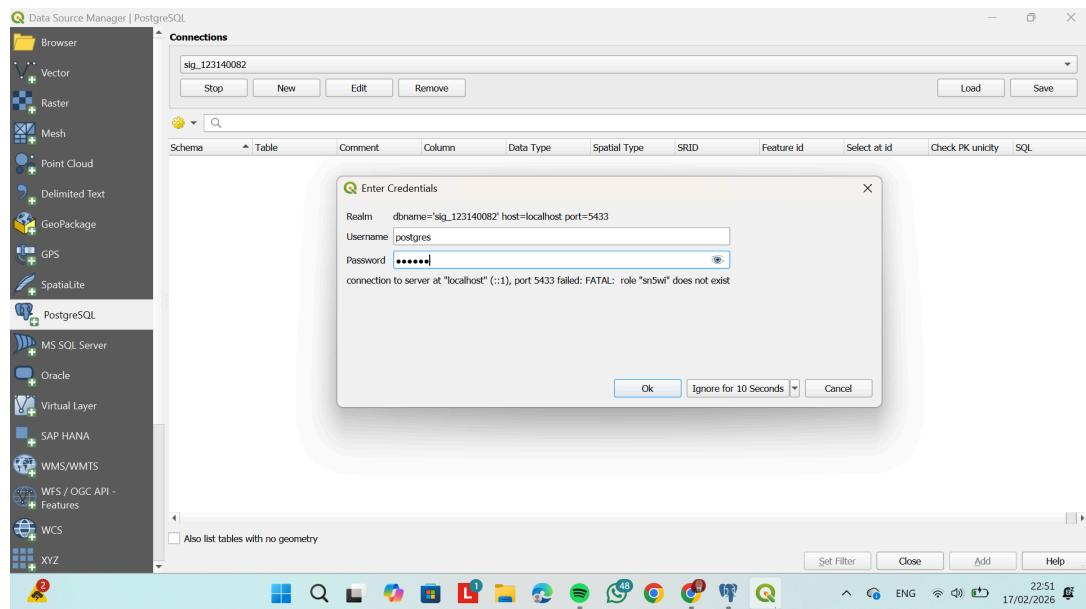
Below the query results, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two rows of data:

	id [PK] integer	nama_kelurahan character varying (100)	is_valid boolean
1	1	Kelurahan Way Huwi	true
2	2	Kelurahan Hajimena	true

A message box at the bottom right indicates: 'Successfully run. Total query runtime: 113 msec. 2 rows affected.' The status bar at the bottom shows 'Total rows: 2 Query complete 00:00:00.113' and 'CRLF Ln 2. Col 14'.

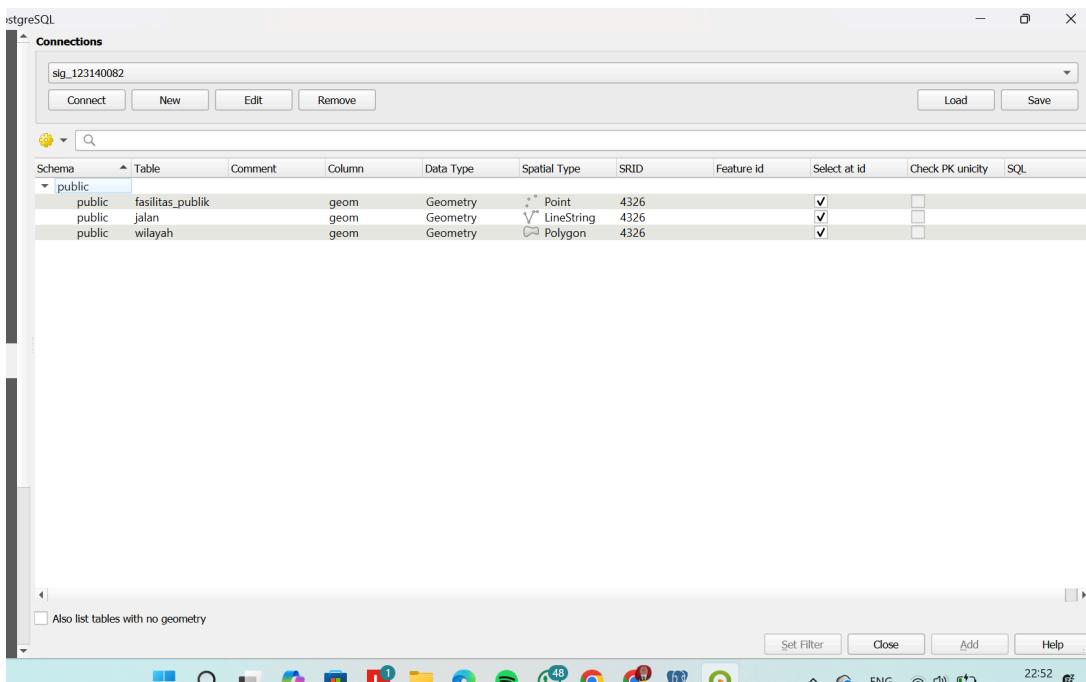
Seluruh data geometri pada tabel jalan maupun tabel wilayah mengembalikan nilai TRUE pada fungsi ST_IsValid(). Hal ini membuktikan bahwa data yang dimasukkan telah memenuhi standar OGC untuk geometri yang valid, antara lain: LineString memiliki minimal 2 titik, dan Polygon memiliki cincin yang tertutup dengan titik awal dan akhir yang sama.

9. Data yang tersimpan di database PostGIS divisualisasikan menggunakan aplikasi QGIS dengan cara menghubungkan QGIS ke database PostgreSQL melalui Data Source Manager > PostgreSQL.



Langkah koneksi QGIS ke PostGIS:

- Buka QGIS, pilih Layer > Add Layer > Add PostGIS Layers...
- Klik New, isikan Name: sig_123140082, Host: localhost, Port: 5433, Database: sig_123140082.
- Masukkan Username: postgres dan Password, lalu klik Connect.
- Pilih semua tabel (fasilitas_publik, jalan, wilayah) lalu klik Add.



Data Source Manager | PostgreSQL

Connections

Schema	Table	Comment	Column	Data Type	Spatial Type	SRID	Feature Id	Select at Id	Check PK Unicity	SQL
public	fasilitas_publik			geom	Geometry	4326		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
public	jalan			geom	Geometry	4326		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
public	wilayah			geom	Geometry	4326		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

PostgreSQL

- MS SQL Server
- Oracle
- Virtual Layer
- SAP HANA
- WMS/WMTS
- WFS / OGC API - Features
- WCS
- XYZ

Also list tables with no geometry

Set Filter Close Add Help

22:52 17/02/2026

Hasil Visualisasi:

Layers

- wilayah
- jalan
- OpenStreetMap
- fasilitas_publik
- fasilitas_publik
- jalan
- wilayah

Coordinate: -5,45479° 105,23651° Scale: 1:17870 Magnifier: 100% Rotation: 0,0 ° Render: EPSG:4326

22:54 17/02/2026

