

BASIS DATA SPASIAL & ANALISIS LANJUT

MASTER CLASS | SERI M2 | KEHUTANAN & PERTANIAN



DAFTAR ISI

Pendahuluan: Dari Peta Berkas ke Gudang Data	1
Prasyarat dan hubungan antarseri	1
Cara membaca buku ini	1
Kenalan lagi dengan KPH Contoh	1
Perangkat dan data	1
Hak cipta dan penggunaan	2
Unduhan: buku dan paket data	3
BAB 1: Satu Gudang untuk Semua Peta: PostGIS dan Geodatabase Perusahaan	4
Studi kasus: "Petak yang mana yang benar?"	4
Konsep: basis data spasial dalam tiga kalimat	4
Bagian A: Membangun gudang data dengan PostGIS dan QGIS	6
A1. Menyiapkan basis data dan skema	6
A2. Memasukkan berkas ke tabel	6
A3. Memeriksa isi gudang	7
A4. Menambah aturan di dalam basis data	7
A5. Menghubungkan QGIS	8
Bagian B: ArcGIS Pro dan geodatabase perusahaan	9
Bagian C: ArcMap 10.8	9
BAB 2: Bertanya pada Peta: SQL Spasial	11
Studi kasus: "Berapa kejadian di tiap petak?"	11
Konsep: SQL spasial dalam tiga kalimat	11
Bagian A: SQL spasial di PostGIS, dijalankan lewat QGIS atau psql	12
A1. Mengukur: luas tiap petak	12
A2. Menggabung berdasarkan letak: kejadian per petak	12
A3. Berjarak paling jauh: tebangan liar dekat jalan	12
A4. Tetangga terdekat: pos jaga untuk tiap titik panas	13
A5. Buffer, gabung, dan potong	13
A6. Mengelompokkan titik	14
A7. Indeks spasial dalam angka	14
A8. View dan view terwujud	15
A9. SQL yang sama, tanpa server	15
Bagian B: ArcGIS Pro	16
Bagian C: ArcMap 10.8	16
BAB 3: Jalan Tercepat dan Jangkauan Layanan: Analisis Jaringan	18
Studi kasus: "Berapa menit dari pos ke lokasi kejadian?"	18
Konsep: jaringan dalam tiga kalimat	18
Bagian A: Jaringan di QGIS dan PostGIS	19
A1. Rute terpendek dan tercepat di QGIS	19
A2. Celah kecil, dampak besar	20
A3. Area layanan di QGIS	21
A4. Jaringan di PostGIS dengan pgRouting	21
A5. Apakah QGIS dan pgRouting sepakat?	22
Bagian B: ArcGIS Pro	23
Bagian C: ArcMap 10.8	23

BAB 4: Ke Mana Air Mengalir: Hidrologi dari DEM	25
Studi kasus: "Petak mana yang dilewati sungai?"	25
Konsep: hidrologi DEM dalam tiga kalimat	25
Bagian A: Hidrologi di QGIS	26
A1. Mengisi lubang	26
A2. Arah dan akumulasi aliran	26
A3. Membentuk sungai	26
A4. Daerah aliran di satu titik keluar	26
A5. Memeriksa hasil dengan hitungan sendiri	27
A6. Sungai dan DAS dikawinkan dengan petak lewat SQL	28
Bagian B: ArcGIS Pro	29
Bagian C: ArcMap 10.8	29
BAB 5: Menebak di Antara Titik: Interpolasi	31
Studi kasus: "Hujan di petak yang tidak punya stasiun"	31
Konsep: interpolasi dalam tiga kalimat	31
Bagian A: Interpolasi di QGIS	32
A1. IDW	32
A2. TIN	32
A3. Spline	32
A4. Kriging	32
A5. Regresi dengan elevasi lalu sisa dengan IDW	32
A6. Menilai: siapa yang paling dekat dengan kenyataan?	33
Bagian B: ArcGIS Pro	35
Bagian C: ArcMap 10.8	35
BAB 6: Mencari Lahan Terbaik: Kesesuaian Lahan Multi-Kriteria	37
Studi kasus: "Di mana kita menanam jenis pohon yang baru?"	37
Konsep: tumpang tindih berbobot dalam tiga kalimat	37
Bagian A: Kesesuaian lahan di QGIS	39
A1. Menurunkan bobot dengan AHP	39
A2. Menyiapkan lima raster kriteria	39
A3. Kendala dan penjumlahan berbobot	40
A4. Hasil	40
A5. Menimbang: analisis kepekaan	42
A6. Angka per petak	43
Bagian B: ArcGIS Pro	43
Bagian C: ArcMap 10.8	44
BAB 7: Menyatukan dan Menjaga: Proyek Terpadu dan Tata Kelola	46
Studi kasus: "Satu tabel untuk rapat besok"	46
Konsep: dari analisis ke layanan data	46
Bagian A: Proyek terpadu dan tata kelola di PostGIS	47
A1. Menyiapkan bahan	47
A2. Lembar keputusan dengan satu kueri	47
A3. Hak akses: siapa boleh apa	48
A4. Jejak perubahan	49
A5. Cadangan dan pemulihan	49
A6. Perawatan ringan	50
Bagian B: ArcGIS Pro	50
Bagian C: ArcMap 10.8	50

Ceklis kesiapan menuju M3 dan M4.....	50
Lanjut belajar.....	52
Glosarium.....	53
Daftar Pustaka.....	58
Dokumentasi resmi.....	58
Makalah dan buku.....	59
Aplikasi Live Penulis.....	60

Pendahuluan: Dari Peta Berkas ke Gudang Data

Buku ini adalah **Seri M2** dari kelas Master. Judulnya **Basis Data Spasial dan Analisis Lanjut**. Di seri sebelumnya Anda bekerja dengan berkas. Di sini Anda pindah ke gudang data bersama, lalu memakainya untuk lima analisis yang sering diminta di kantor kehutanan dan pertanian.

Setelah selesai, Anda bisa memasang data di PostGIS, bertanya kepadanya dengan SQL, mencari rute dan jangkauan layanan, menurunkan sungai dari DEM, menebak curah hujan di antara stasiun, dan menilai kesesuaian lahan dengan banyak kriteria.

Prasyarat dan hubungan antarseri

- **Seri I3 (Survei Lapangan dan Basis Data).** Anda sudah mengenal GeoPackage, domain, dan SQL dasar. Di M2 semuanya dipakai lagi di server.
- **Seri I4 (Otomasi Dasar dengan Python).** Anda sudah bisa menjalankan skrip PyQGIS dan memanggil `processing.run`. Di M2 hampir semua analisis juga ada dalam bentuk skrip.
- **Seri sesudahnya.** M3 (deret waktu dan pemodelan) dan M4 (pengembangan dan publikasi) tidak mewajibkan seri ini. Anda boleh membacanya dalam urutan apa pun, dan gudang data M2 bisa dipakai sebagai sumber data di keduanya.

Seri M1 tidak wajib dibaca lebih dulu. Bila Anda sudah membacanya, data survei dan citra dari sana bisa dimasukkan ke gudang yang dibuat di Bab 1.

Cara membaca buku ini

Satu bab terbagi dalam bagian dengan pola yang sama:

- **Studi kasus:** masalah kecil dari "KPH Contoh", sebuah kantor pengelola hutan fiktif.
- **Konsep:** inti gagasan dalam tiga kalimat, dengan analogi.
- **Langkah:** **Bagian A** untuk QGIS (acuan utama), **Bagian B** untuk ArcGIS Pro, **Bagian C** untuk ArcMap 10.8, bila ada padanannya.
- **Cek paham, Kesalahan umum, Ringkasan dan latihan,** dan **tabel perbandingan**.

Anda boleh memilih satu perangkat saja. Bila sebuah fitur tidak ada di perangkat tertentu, buku ini menyebutnya terus terang.

Kenalan lagi dengan KPH Contoh

Cerita tetap terjadi di **KPH Contoh**. Semua nama, tempat, dan peristiwa fiktif. Tokohnya hanya dikenal lewat peran: **Analisis Muda** (itu Anda) dan **Kepala Seksi**.

Pada seri ini wilayah kerja diperluas menjadi **2 km x 2 km (400 ha)**. Sudut barat dayanya sama dengan area contoh di seri sebelumnya. Wilayah ini berisi **25 petak** (P-01 sampai P-25), jaringan jalan, stasiun hujan, dan 150 catatan kejadian. Semua data buatan, jadi aman dipakai. Hasil analisis **bukan** kondisi lapangan sebenarnya.

Perangkat dan data

- **QGIS 4.0.2.** Semua skrip diuji di versi ini dengan data sintetis.

- **PostgreSQL 17 dengan PostGIS 3.6 dan pgRouting 4.0.** Itu versi yang dipakai penulis. Versi lain bisa sedikit berbeda.
- **ArcGIS Pro dan ArcMap 10.8.** Langkah bertanda **[CEK]** perlu Anda cocokkan dengan perangkat Anda. Laporkan bila ada beda.
- **Paket data `paket-m2`**, semuanya dalam WGS 84 / UTM zona 49S (EPSG:32749):

Berkas	Isi
Petak.gpkg	25 poligon petak: kode, blok, jenis tegakan, tahun tanam
Jalan.gpkg	37 ruas jalan: kelas, kecepatan, status (satu ruas ditutup)
Jalan_celah.gpkg	salinan jalan dengan satu celah 4 m (untuk latihan)
Fasilitas.gpkg	Kantor KPH, 4 pos jaga, 1 TPK
Kejadian.gpkg	150 titik: titik panas, terbang liar, longsor
Stasiun_Hujan.gpkg	30 stasiun: 24 untuk melatih, 6 untuk menguji
Tanah.gpkg	8 poligon jenis tanah
DEM_10m.tif	model tinggi, piksel 10 m, 200 x 200
hasil/	keluaran skrip Bab 4 sampai 6 (sudah jadi, boleh dipakai langsung)

Skrip pembuat data (`buat_data_m2.py`) dan semua skrip bab ada di folder `skrip`. Skrip yang memakai QGIS dijalankan dengan `python-qgis.bat` dari folder QGIS, atau ditempel di Python Console QGIS.

Catatan tentang ilustrasi: Gambar di buku ini adalah **skema buatan penulis** atau peta dari data sintetis, bukan tangkapan layar. Letak tombol di komputer Anda bisa sedikit berbeda. Ikuti nama menu yang tertulis.

Apa yang sudah dijalankan: Perintah SQL, rute, hidrologi, interpolasi, dan kesesuaian lahan dijalankan dan angkanya dicatat di buku ini. Yang dijalankan dan dicatat angkanya adalah alat dan skrip padanannya. Letak klik di jendela bisa sedikit berbeda di versi Anda.

Kontak penulis: badarmubarok@gmail.com.

Hak cipta dan penggunaan

Hak cipta © 2026 Badar Mubarak Yogaswara. All Rights Reserved.

Hak cipta dilindungi undang-undang. Dilarang mengutip atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penulis. Untuk penggunaan lain, misalnya bahan ajar berbayar atau penerbitan ulang, hubungi penulis di badarmubarok@gmail.com.

Data latihan di buku ini buatan (sintetis), bukan data lapangan. Nama produk dan merek, misalnya QGIS dan ArcGIS, milik pemiliknya masing-masing.

Unduhan: buku dan paket data

Buku ini dan semua berkas latihannya tersedia gratis di situs penulis. Alamat lengkapnya tertulis di bawah, jadi bisa Anda ketik langsung bila membaca versi cetak.

Berkas	Tautan	Isi
Halaman unduhan M2	https://badarmubarok.id/buku-gis/m2/unduh.html	semua berkas seri ini
Buku (PDF)	https://badarmubarok.id/buku-gis/m2/unduh/BUKU-M2.pdf	buku ini dalam PDF
Paket data latihan	https://badarmubarok.id/buku-gis/m2/unduh/data-sintetis-m2.zip	data sintetis latihan
Skrip (zip)	https://badarmubarok.id/buku-gis/m2/unduh/skrip-m2.zip	skrip Python dan lainnya
Seluruh seri	https://badarmubarok.id/buku-gis/	12 buku Basic sampai Master

Pindai kode QR ini dengan kamera ponsel untuk membuka halaman unduhan M2.



BAB 1: Satu Gudang untuk Semua Peta: PostGIS dan Geodatabase Perusahaan

Studi kasus: "Petak yang mana yang benar?"

Di folder bersama KPH Contoh ada tiga berkas petak: `petak_v3`, `petak_FINAL`, dan `petak_revisi_baru`. Dua rekan sama-sama menyunting dan saling menimpa. Kepala Seksi bertanya, "Yang resmi yang mana?" Tak ada yang bisa menjawab dengan pasti. Anda diminta membuat satu gudang data, dengan aturan dan jejak siapa mengubah apa.

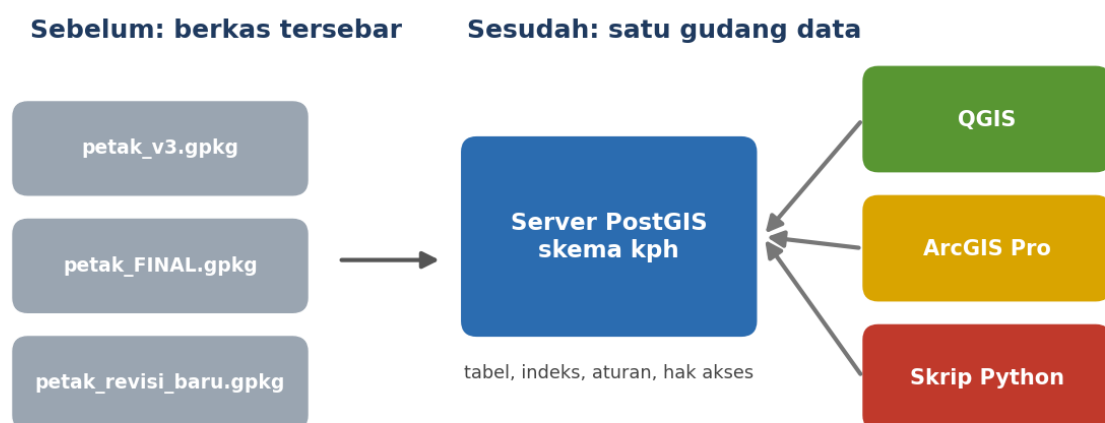
Konsep: basis data spasial dalam tiga kalimat

Basis data spasial adalah gudang data yang dijaga satu petugas, yaitu **server**. Berkas biasa seperti kertas di meja yang boleh dicoret siapa saja. Gudang data seperti lemari berkunci: ada rak, ada aturan, dan ada buku tamu.

Istilah baru bab ini:

- **PostGIS**: tambahan untuk PostgreSQL yang memberi kemampuan menyimpan titik, garis, dan poligon.
- **Skema**: "rak" di dalam basis data tempat tabel dikelompokkan.
- **SRID**: nomor sistem koordinat yang menempel pada tiap geometri (untuk UTM 49S: 32749).
- **Indeks spasial**: daftar isi yang membuat pencarian berdasarkan lokasi cepat.
- **Peran**: kelompok hak akses, misalnya "boleh baca saja".

Ilustrasi 1.1: Dari berkas tersebar ke satu gudang data



Siapa yang benar? Siapa menimpa siapa?

Skema; bukan tangkapan layar

Dunia Esri memakai istilah **geodatabase perusahaan** (enterprise geodatabase) untuk gagasan serupa. Isinya tetap tabel di server basis data. Esri menambahkan fitur seperti domain, subtype, dan

penyuntingan berversi. PostGIS punya padanan untuk sebagian fitur itu, dan tidak untuk yang lain. Tabel di akhir bab menjelaskannya.

BAGIAN A | QGIS

Bagian A: Membangun gudang data dengan PostGIS dan QGIS

A1. Menyiapkan basis data dan skema

Langkah ini dikerjakan pemilik server, bukan analis biasa. Anda memerlukan PostgreSQL dengan PostGIS terpasang. Basis data kosong bernama `kph_contoh` dibuat lebih dulu.

1. Buka terminal. Buat basis data kosong:

terminal

```
createdb -h localhost -U postgres kph_contoh
```

2. Jalankan berkas `m2_01a_skema.sql` dari folder skrip:

terminal

```
psql -h localhost -U postgres -d kph_contoh -f m2_01a_skema.sql
```

Hasilnya: ekstensi PostGIS aktif, skema `kph` ada, dan dua peran (`pembaca`, `penyunting`) siap dipakai. Isi pokok berkasnya:

sql

```
CREATE EXTENSION IF NOT EXISTS postgis;
CREATE SCHEMA IF NOT EXISTS kph;
CREATE ROLE pembaca NOLOGIN;
CREATE ROLE penyunting NOLOGIN;
GRANT USAGE ON SCHEMA kph TO pembaca, penyunting;
ALTER DEFAULT PRIVILEGES IN SCHEMA kph GRANT SELECT ON TABLES TO pembaca, penyunting;
ALTER DEFAULT PRIVILEGES IN SCHEMA kph GRANT INSERT, UPDATE, DELETE ON TABLES TO penyunting;
```

Kata sandi: Jangan menulis kata sandi di skrip atau di berkas yang Anda bagikan. Pakai variabel lingkungan `PGPASSWORD`, berkas `pgpass`, atau biarkan program menanyakannya.

A2. Memasukkan berkas ke tabel

Anda punya tiga cara. Pilih salah satu.

Cara 1: satu perintah `ogr2ogr`. Perintah ini membaca GeoPackage dan menulis tabel PostGIS. Contoh untuk petak:

terminal

```
ogr2ogr -f PostgreSQL PG:"dbname=kph_contoh" Petak.gpkg Petak -nln kph.petak -nlt MULTIPOLYGON -lco GEOMETRY_NAME=geom -lco FID=gid -overwrite
```

Keterangan singkat: `-nln` memberi nama tabel (`kph.petak = skema.tabel`), `-nlt MULTIPOLYGON` menyeragamkan jenis geometri, `-lco GEOMETRY_NAME=geom` menamai kolom bentuk, `-lco FID=gid` menamai kunci utama.

Cara 2: skrip Python. Berkas `m2_01b_impor.py` memasukkan ketujuh layer sekaligus:

terminal

```
python-qgis.bat m2_01b_impor.py D:\data\paket-m2
```

Hasil uji penulis: tujuh tabel terbentuk (`petak`, `tanah`, `jalan`, `fasilitas`, `kejadian`, `stasiun_hujan`, `batas_wilayah`).

Cara 3: alat Processing. Di QGIS buka **Processing ► Toolbox**, cari **Export to PostgreSQL**. Alat GDAL padanannya, **Export to PostgreSQL (new connection)**, diuji lewat skrip `m2_01c_qgis_muat.py` dan menghasilkan tabel dengan jumlah baris yang sama.

A3. Memeriksa isi gudang

Buka `psql` dan jalankan:

sql

```
SELECT f_table_name, srid, type
FROM geometry_columns
WHERE f_table_schema = 'kph'
ORDER BY 1;
```

Hasil uji: tujuh tabel, semuanya `srid = 32749`. Jenisnya `MULTIPOLYGON` untuk `petak`, `tanah`, dan `batas wilayah`, `MULTILINESTRING` untuk `jalan`, dan `POINT` untuk `fasilitas`, `kejadian`, dan `stasiun hujan`.

Periksa juga indeks spasial. `ogr2ogr` membuatnya otomatis:

sql

```
SELECT tablename, indexname FROM pg_indexes WHERE schemaname = 'kph' ORDER BY 1;
```

Hasil uji: 14 indeks, yaitu satu kunci utama dan satu indeks `gist` untuk tiap tabel.

A4. Menambah aturan di dalam basis data

Aturan di basis data menolak data salah **sebelum** masuk. Ini padanan "domain" di geodatabase.

Jalankan `m2_01d_aturan_data.sql`. Isi pokoknya:

sql

```
CREATE TABLE kph.ref_jenis_tegakan (jenis text PRIMARY KEY, keterangan text);
INSERT INTO kph.ref_jenis_tegakan VALUES ('Jati','Tectona grandis'), ('Mahoni','Swietenia'),
('Sengon','Falcataria'), ('Akasia','Acacia'), ('Lahan kosong','Belum ditanami');
ALTER TABLE kph.petak ADD CONSTRAINT petak_kode_unik UNIQUE (kode);
ALTER TABLE kph.petak ADD CONSTRAINT petak_jenis_fk
FOREIGN KEY (jenis_tegakan) REFERENCES kph.ref_jenis_tegakan (jenis);
ALTER TABLE kph.petak ADD CONSTRAINT petak_tahun_wajar
CHECK (tahun_tanam IS NULL OR tahun_tanam BETWEEN 1900 AND 2100);
```

```
ALTER TABLE kph.petak ADD CONSTRAINT petak_geom_sah CHECK (ST_IsValid(geom));
```

Sekarang coba memasukkan data yang salah. Hasil uji penulis:

Percobaan	Jawaban server
Jenis tegakan "Pinus" (tidak ada di daftar)	ditolak: melanggar foreign key petak_jenis_fk
Titik dimasukkan ke kolom poligon	ditolak: Geometry type (Point) does not match column type (MultiPolygon)
Geometri ber-SRID 4326	ditolak: Geometry SRID (4326) does not match column SRID (32749)
Kode P-01 dimasukkan dua kali	ditolak: melanggar petak_kode_unik
Tahun tanam 3022	ditolak: melanggar petak_tahun_wajar

Hasilnya: jumlah petak tetap 25. Gudang menjaga dirinya sendiri.

A5. Menghubungkan QGIS

1. Buka **Layer ► Data Source Manager** (Ctrl+L), lalu tab **PostgreSQL**.
2. Klik **New**. Isi **Name** (misalnya KPH Contoh), **Host**, **Port**, dan **Database** (kph_contoh).
3. Pada **Authentication**, pilih **Basic**, lalu isi nama pengguna dan kata sandi Anda sendiri. Simpan dalam konfigurasi otentikasi QGIS. [CEK: label menu]
4. Klik **Test Connection**, lalu **OK**. Klik **Connect**.
5. Buka skema kph, pilih tabel petak, lalu klik **Add**.

Hasilnya: layer petak muncul di peta dan tabel atributnya berisi 25 baris. Setiap penyuntingan Anda langsung tersimpan di server.

Lewat Python, layer yang sama dimuat begini (skrip m2_01c_qgis_muat.py):

```
python
from qgis.core import QgsDataSourceUri, QgsVectorLayer
uri = QgsDataSourceUri()
uri.setConnection("localhost", "5432", "kph_contoh", "analisis", "")
uri.setDataSource("kph", "petak", "geom", "", "gid")
lyr = QgsVectorLayer(uri.uri(False), "Petak (PostGIS)", "postgres")
print(lyr.isValid(), lyr.featureCount())
```

Hasil uji: True 25. Dengan `uri.setSql("jenis_tegakan = 'Jati'")` jumlahnya menjadi 10. Kata sandi sengaja dikosongkan; QGIS akan menanyakannya.

Jendela DB Manager: QGIS juga punya **Database ► DB Manager**. Di sana Anda bisa melihat tabel, mengetik SQL, dan memuat hasilnya sebagai layer. Bab 2 memakai jendela ini. [CEK: letak menu]

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro dan geodatabase perusahaan

Dokumentasi Esri menyebut alat **Create Enterprise Geodatabase** (kelompok **Data Management Tools**). Alat ini mendukung Oracle, PostgreSQL, dan SQL Server. Untuk PostgreSQL ada pilihan jenis ruang (**Spatial Type**): ST_Geometry atau PostGIS. Penulis membaca ini dari dokumentasi resmi, belum menjalankannya.

1. Siapkan server PostgreSQL dan berkas otorisasi (keycodes) dari ArcGIS Server. [CEK]
2. Jalankan **Create Enterprise Geodatabase**. Isi **Database Platform** (PostgreSQL), **Instance, Database**, pengelola basis data, pengelola geodatabase (umumnya *sde*), dan berkas otorisasi. [CEK: nama dan urutan kolom]
3. Di panel **Catalog**, buat koneksi basis data baru ke server tersebut. [CEK]
4. Salin feature class dari GeoPackage ke koneksi itu (klik kanan, **Import**). [CEK]
5. Untuk aturan nilai, buat **domain** (alat **Create Domain**), lalu pasang ke kolom lewat **Assign Domain To Field**. [CEK]

Bila Anda memilih jenis ruang PostGIS, tabel dapat dibaca QGIS seperti pada Bagian A. Bila memilih ST_Geometry, QGIS tidak membacanya dengan cara yang sama. [CEK]

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

1. Alat **Create Enterprise Geodatabase** juga dikenal di seri 10.x. [CEK: ada di 10.8 dan opsi PostgreSQL]
2. Di **ArcCatalog**, buat koneksi lewat **Database Connections ► Add Database Connection**. [CEK]
3. Seret feature class ke koneksi itu untuk mengimpornya. [CEK]
4. Domain dibuat lewat properti geodatabase (**Domains**) atau alat **Create Domain**. [CEK]

Cek paham

1. Apa gunanya indeks spasial?
2. Mengapa aturan seperti "jenis tegakan harus ada di daftar" lebih baik diletakkan di basis data daripada di formulir QGIS saja?
3. Anda memasukkan data ber-SRID 4326 ke tabel ber-SRID 32749 dan ditolak. Apa arti penolakan itu?

Jawaban:

1. Mempercepat pencarian berdasarkan lokasi. Tanpa indeks, server memeriksa semua baris satu per satu.
2. Karena aturan di server berlaku untuk semua program dan semua orang, termasuk skrip. Aturan di formulir hanya berlaku bagi pemakai formulir itu.

3. Basis data menjaga agar semua geometri satu tabel memakai satu sistem koordinat. Ubah dulu koordinatnya (reproyeksi), baru masukkan.

Kesalahan umum

- **Memasukkan data tanpa memeriksa SRID.** Letak petak bisa meleset ratusan kilometer. Periksa `geometry_columns` setelah impor.
- **Mencampur jenis geometri.** GeoPackage mengizinkan poligon tunggal dan ganda dalam satu layer, PostGIS kolom bertipe tidak. Seragamkan dengan `-nlt MULTIPOLYGON`.
- **Menulis kata sandi di skrip yang dibagikan.** Pakai `pgpass` atau variabel lingkungan.
- **Memberi semua orang hak penyunting.** Beri hak baca saja kepada kebanyakan orang (Bab 7).

Ringkasan dan latihan

Ringkasan: basis data spasial memusatkan data, menjaga aturan, dan melayani banyak pengguna. PostGIS menambahkan geometri ke PostgreSQL. Geodatabase perusahaan Esri adalah gagasan serupa dengan fitur tambahan.

Latihan: tambahkan tabel `kph.ref_blok` berisi Blok A, B, dan C. Pasang foreign key dari `petak.blok`. Coba masukkan "Blok Z" dan catat jawaban server.

Tabel perbandingan: file dan server

Hal	GeoPackage (berkas)	PostGIS	Geodatabase perusahaan Esri
Tempat data	satu berkas	server PostgreSQL	server (Oracle, PostgreSQL, SQL Server)
Banyak penyunting bersamaan	terbatas	ya	ya
Aturan nilai	pembatas dan trigger, bentuk lain terbatas	CHECK, FOREIGN KEY, trigger	domain, subtype, relasi
Penyuntingan berversi	tidak ada	tidak bawaan (bisa dirancang sendiri, Bab 7)	ada
Dibaca QGIS	ya	ya	ya untuk PostGIS, [CEK] untuk ST_Geometry

BAB 2: Bertanya pada Peta: SQL Spasial

Studi kasus: "Berapa kejadian di tiap petak?"

Kepala Seksi mengirim tiga pertanyaan sekaligus. Berapa kejadian di tiap petak? Berapa tebangan liar yang terjadi dekat jalan? Pos jaga mana yang paling dekat dengan tiap titik panas? Dengan SQL spasial, tiap pertanyaan itu cukup satu kueri, dan jawabannya keluar dalam hitungan milidetik.

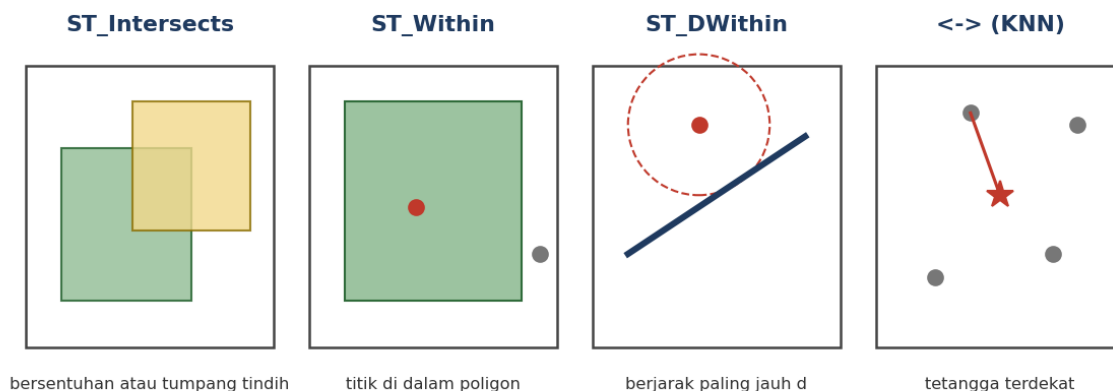
Konsep: SQL spasial dalam tiga kalimat

SQL spasial adalah SQL biasa ditambah fungsi yang mengerti letak dan bentuk, dan namanya selalu berawalan `ST_`. Bayangkan bertanya kepada pustakawan: "Buku mana yang ada di rak ini?" Di sini rak itu adalah poligon petak. Satu kueri bisa menggabungkan tabel berdasarkan **letak**, bukan hanya berdasarkan nilai kolom.

Istilah baru bab ini:

- **Predikat spasial:** fungsi yang menjawab ya atau tidak, misalnya `ST_Intersects` (bersentuhan) dan `ST_DWithin` (berjarak paling jauh sekian).
- **Pengukur:** fungsi yang menjawab angka, misalnya `ST_Area` dan `ST_Distance`.
- **Tetangga terdekat (KNN):** mengurutkan hasil menurut kedekatan dengan operator `<->`.
- **View:** kueri yang disimpan dan bisa dibuka seperti tabel.
- **EXPLAIN:** perintah untuk melihat cara server menjalankan kueri.

Ilustrasi 2.1: Predikat spasial



Skema; bukan tangkapan layar

Satuan ukuran: `ST_Area` dan `ST_Distance` memakai satuan sistem koordinatnya. Karena data latihan ini memakai UTM (meter), luas keluar dalam meter persegi dan jarak dalam meter. Pada data berkoordinat derajat, hasilnya tidak berguna.

BAGIAN A | QGIS

Bagian A: SQL spasial di PostGIS, dijalankan lewat QGIS atau psql

Berkas `m2_02a_kueri_dasar.sql` berisi semua kueri di bawah. Anda bisa menjalankannya dengan `psql -f`, atau menempelkannya di **DB Manager QGIS: Database ► DB Manager**, pilih koneksi, lalu buka jendela **SQL**. Klik **Execute**. Centang **Load as new layer** bila hasilnya punya kolom bentuk dan kunci unik. [CEK: letak menu]

A1. Mengukur: luas tiap petak

sql

```
SELECT kode, jenis_tegakan, ROUND((ST_Area(geom) / 10000.0)::numeric, 2) AS luas_ha
FROM kph.petak ORDER BY kode;
```

Hasil uji: 25 baris, mulai P-01 (Sengon, 17,52 ha). Jumlah luas semua petak 400,00 ha, sama dengan luas wilayah 2 km x 2 km. Angka itu bisa dipakai sebagai alat pemeriksa: bila totalnya tidak 400, ada petak tumpang tindih atau berlubang.

A2. Menggabung berdasarkan letak: kejadian per petak

sql

```
SELECT p.kode, COUNT(k.gid) AS jumlah
FROM kph.petak AS p
LEFT JOIN kph.kejadian AS k ON ST_Intersects(p.geom, k.geom)
GROUP BY p.kode
ORDER BY jumlah DESC, p.kode
LIMIT 5;
```

LEFT JOIN menjaga petak yang tidak punya kejadian tetap muncul dengan angka nol. Hasil uji:

Petak	Kejadian
P-04	13
P-09	12
P-12	12
P-08	9
P-01	7

Pemeriksaan: jumlah semua petak = 150 = jumlah baris `kph.kejadian`. Setiap kejadian jatuh di tepat satu petak.

A3. Berjarak paling jauh: tebaran liar dekat jalan

sql

```
SELECT k.jenis,
COUNT(*) AS total,
```

```

COUNT(*) FILTER (WHERE EXISTS (
    SELECT 1 FROM kph.jalan j WHERE ST_DWithin(k.geom, j.geom, 100))) AS dekat_jalan
FROM kph.kejadian k
GROUP BY k.jenis ORDER BY k.jenis;

```

ST_DWithin(a, b, 100) berarti "a dan b berjarak paling jauh 100 meter". Hasil uji:

Jenis	Total	Dalam 100 m dari jalan	Persen
Longsor	40	24	60,0
Tebangan liar	60	59	98,3
Titik panas	50	36	72,0

Tebangan liar nyaris selalu dekat jalan (data sintetis sengaja dibuat begitu). Pada data nyata, pola seperti ini menjadi dasar menentukan rute patroli.

Pakai ST_DWithin, bukan ST_Distance di WHERE: ST_Distance(a, b) < 100 memaksa server menghitung jarak untuk semua pasangan. ST_DWithin bisa memakai indeks spasial, jadi jauh lebih cepat.

A4. Tetangga terdekat: pos jaga untuk tiap titik panas

```

sql
SELECT pos.nama AS pos_terdekat, COUNT(*) AS jumlah_titik_panas,
    ROUND(AVG(pos.jarak)::numeric, 1) AS rata_jarak_m
FROM kph.kejadian AS k
CROSS JOIN LATERAL (
    SELECT f.nama, ST_Distance(f.geom, k.geom) AS jarak
    FROM kph.fasilitas AS f
    WHERE f.jenis = 'Pos'
    ORDER BY f.geom <-> k.geom
    LIMIT 1) AS pos
WHERE k.jenis = 'Titik panas'
GROUP BY pos.nama ORDER BY pos.nama;

```

LATERAL membuat subkueri dijalankan sekali untuk tiap titik panas. Di dalamnya, ORDER BY f.geom <-> k.geom LIMIT 1 mengambil pos terdekat. Hasil uji: Pos Jaga 4 melayani 29 dari 50 titik panas (rata-rata jarak 629,0 m), Pos Jaga 1 hanya 5 titik. Beban pos tidak seimbang.

A5. Buffer, gabung, dan potong

Berapa luas tiap petak yang berada dalam 50 m dari jalan?

```

sql
WITH sabuk AS (SELECT ST_Union(ST_Buffer(geom, 50)) AS g FROM kph.jalan)
SELECT p.kode, ROUND((ST_Area(ST_Intersection(p.geom, s.g)) / 10000.0)::numeric, 2) AS ha_dekat_jalan
FROM kph.petak p, sabuk s
ORDER BY ha_dekat_jalan DESC, p.kode LIMIT 5;

```

Hasil uji: P-03 9,28 ha, P-02 8,73 ha, P-09 8,21 ha, P-08 8,15 ha, P-17 7,67 ha. `ST_Union` melebur semua buffer jadi satu bidang, jadi bagian yang tumpang tindih tidak dihitung dua kali.

Dua kueri serupa menjawab pertanyaan "dissolve" dan "overlay" dari Seri I1:

sql

```
-- luas tiap jenis tegakan (dissolve)
SELECT jenis_tegakan, COUNT(*) AS jumlah_petak,
       ROUND((ST_Area(ST_Union(geom)) / 10000.0)::numeric, 2) AS luas_ha
FROM kph.petak GROUP BY jenis_tegakan ORDER BY luas_ha DESC;
```

Hasil uji: Jati 160,97 ha (10 petak), Mahoni 82,71 ha, Sengon 78,86 ha, Akasia 42,62 ha, Lahan kosong 34,84 ha. Overlay petak dengan tanah memakai `ST_Intersection` dan hasilnya: Aluvial 46,58 ha, Latosol 146,02 ha, Podsolik 116,70 ha, Regosol 90,70 ha. Totalnya 400,00 ha.

A6. Mengelompokkan titik

sql

```
SELECT cid, COUNT(*) FROM (
  SELECT ST_ClusterDBSCAN(geom, eps := 150, minpoints := 3) OVER () AS cid
  FROM kph.kejadian WHERE jenis = 'Titik panas') s
GROUP BY cid ORDER BY cid NULLS LAST;
```

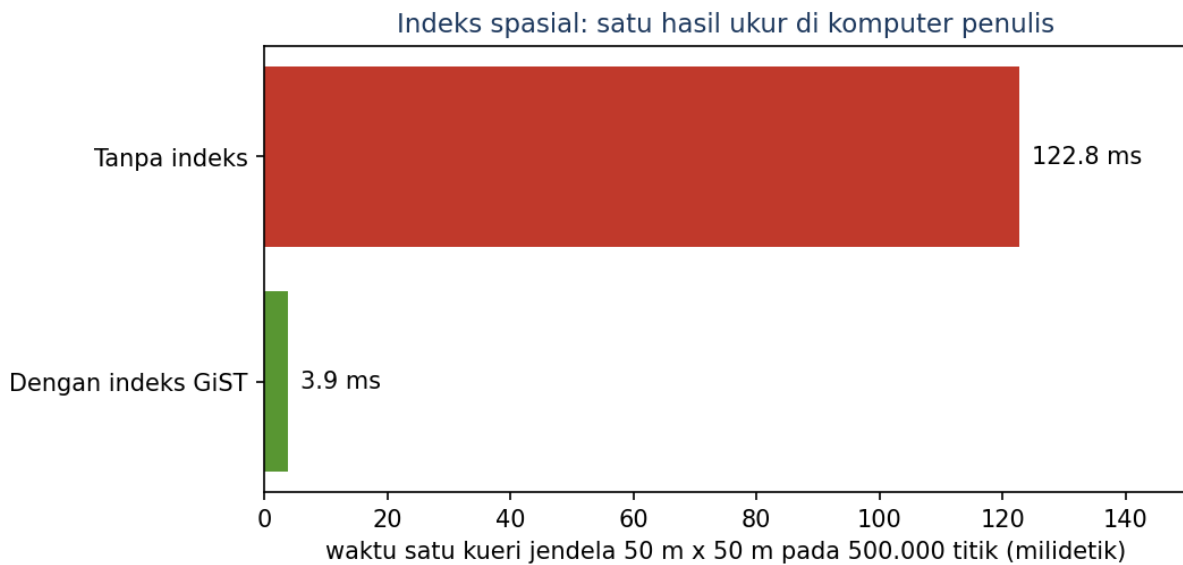
Hasil uji: satu gerombol berisi 25 titik panas, dan 25 titik lain tersebar (`cid` kosong). Gerombol itulah yang layak dipantau lebih dulu.

A7. Indeks spasial dalam angka

Penulis membuat tabel uji 500.000 titik acak, lalu mencari titik di jendela 50 m x 50 m. Skrip: `m2_02b_indeks_dan_view.sql`. Ketik `EXPLAIN` di depan kueri untuk melihat rencananya.

Keadaan	Rencana server	Waktu (satu kali ukur)
Tanpa indeks	Parallel Seq Scan (baca semua baris)	122,8 ms
Dengan indeks <code>gist</code>	Bitmap Index Scan pada indeks	3,9 ms

Keduanya mengembalikan jumlah titik yang sama. Selisihnya sekitar 30 kali pada ukuran ini, dan makin besar pada tabel yang makin besar. Waktu di komputer Anda akan berbeda. Yang penting adalah pola: indeks mengubah "baca semua" menjadi "baca bagian yang perlu".

Ilustrasi 2.2: Hasil ukur indeks spasial

Skema dari data sintetis; bukan tangkapan layar. Angka milik Anda akan berbeda.

Indeks tidak selalu menolong: Pada percobaan penulis, kueri yang menggabungkan 25 petak dengan seluruh 500.000 titik hanya sedikit lebih cepat dengan indeks (sekitar 1,3 detik menjadi 0,9 detik), karena hampir semua titik memang terpakai. Indeks paling berguna bila Anda mencari bagian kecil dari tabel besar.

A8. View dan view terwujud

View menyimpan kueri. Isinya selalu terbaru karena dihitung ulang tiap dibuka. **View terwujud** (materialized view) menyimpan hasilnya, cepat tetapi harus disegarkan.

sql

```
CREATE VIEW kph.v_kejadian_per_petak AS
SELECT p.gid, p.kode, p.geom, COUNT(k.gid) AS jumlah_kejadian
FROM kph.petak p LEFT JOIN kph.kejadian k ON ST_Intersects(p.geom, k.geom)
GROUP BY p.gid, p.kode, p.geom;

CREATE MATERIALIZED VIEW kph.mv_kejadian_per_petak AS SELECT * FROM kph.v_kejadian_per_petak;
REFRESH MATERIALIZED VIEW kph.mv_kejadian_per_petak;
```

Hasil uji: jumlah awal 150. Setelah satu kejadian ditambahkan, view biasa menunjukkan 151, view terwujud masih 150, dan baru menjadi 151 setelah `REFRESH`. Kejadian uji itu lalu dihapus.

A9. SQL yang sama, tanpa server

Bila PostGIS belum tersedia, GeoPackage bisa dikueri dengan SQL bergaya Spatialite lewat GDAL. Skrip `m2_02c_silang_geopackage.py` menjalankan kueri yang sama di sana. Hasil uji: total luas 400,0 ha, tiga petak teratas (P-04 13, P-09 12, P-12 12), dan 59 terbangun liar dekat jalan, **semuanya sama persis** dengan PostGIS.

Dua perbedaan yang perlu Anda ketahui. Pada uji penulis, jarak ditulis `ST_Distance(...) <= 100`, karena `ST_DWithin` dan operator `<->` tidak dicoba di dialek itu. Kolom bentuk bernama `geom`, dan kolom kunci bernama `fid`.

QGIS punya dua padanan tanpa server (skrip `m2_02d_qgis_setara.py`):

- **Count points in polygon** (Processing ► Toolbox): hasil P-04 13, P-09 12, P-12 12.
- **Execute SQL** (kueri pada layer maya): tabel masukan bernama `input1`, `input2`, dan seterusnya. Hasilnya sama.

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro

Pro punya dua jalan. Jalan pertama memakai alat bawaan tanpa menulis SQL. Jalan kedua memakai SQL.

1. **Menggabung berdasarkan letak:** alat **Spatial Join** atau **Summarize Within**. Pilih petak sebagai sasaran dan kejadian sebagai gabungan. Hasilnya jumlah kejadian per petak. [CEK]
2. **Berjarak paling jauh:** alat **Select Layer By Location** dengan hubungan **Within a distance**. [CEK]
3. **Tetangga terdekat:** alat **Near** atau **Generate Near Table**. [CEK]
4. **SQL:** untuk tabel di geodatabase perusahaan, Pro memakai fungsi SQL milik Esri (`ST_Geometry`) bila jenis ruangnya `ST_Geometry`. Tabel PostGIS dapat dibaca lewat **Query Layer (Map ► Add Data ► Query Layer)**. Fungsi `ST_PostGIS` dijalankan oleh PostgreSQL di dalam kueri itu. [CEK]

Pro tidak punya padanan langsung untuk `ST_ClusterDBSCAN`. Alat **Density-based Clustering** dalam kotak peralatan Spatial Statistics mengerjakan hal serupa. [CEK]

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

1. **Spatial Join** (klik kanan layer ► **Joins and Relates** ► **Join** ► **Join data from another layer based on spatial location**) menghitung kejadian per petak. [CEK]
2. **Select By Location** dengan pilihan **are within a distance of the source layer feature**. [CEK]
3. **Near** (kotak peralatan Analysis) menghitung jarak ke fasilitas terdekat. [CEK]
4. **Query Layer: File ► Add Data ► Add Query Layer** untuk SQL ke basis data. [CEK]

Cek paham

1. Mengapa kueri A2 memakai `LEFT JOIN`, bukan `JOIN` biasa?
2. Mana yang lebih baik di klausa `WHERE: ST_Distance(a, b) < 100` atau `ST_DWithin(a, b, 100)`? Mengapa?

3. View terwujud menunjukkan 150 kejadian, padahal sudah ada 151 di tabel. Apa yang harus dilakukan?

Jawaban:

1. `LEFT JOIN` mempertahankan petak tanpa kejadian dengan hitungan nol. `JOIN` biasa membuang petak itu dari hasil.
2. `ST_DWithin`, karena bisa memakai indeks spasial.
3. Jalankan `REFRESH MATERIALIZED VIEW`. View terwujud hanya diperbarui saat disegarkan.

Kesalahan umum

- **Menghitung luas pada data berkoordinat derajat.** Hasilnya satuan derajat persegi. Reproyeksi dulu ke UTM, atau pakai tipe `geography`.
- **Lupa indeks spasial pada tabel hasil buatan sendiri.** Tabel baru dari `CREATE TABLE AS` tidak punya indeks. Buat dengan `CREATE INDEX ... USING gist (geom)`.
- **Memakai `JOIN` padahal ingin semua petak muncul.** Pakai `LEFT JOIN`.
- **Menggabung dua tabel ber-SRID berbeda.** PostGIS akan menolak. Samakan dengan `ST_Transform`.

Ringkasan dan latihan

Ringkasan: SQL spasial menggabung tabel berdasarkan letak. Predikat menjawab ya atau tidak, pengukur menjawab angka, `<->` mengurutkan menurut kedekatan. Indeks spasial dan `ST_DWithin` membuatnya cepat.

Latihan: tulis kueri yang menghitung, untuk tiap petak, jumlah longsor dalam jarak 200 m dari sungai. (Tabel sungai dibuat di Bab 4; sementara itu pakai jalan.) Periksa bahwa jumlah semua petak tidak melebihi 40.

Tabel perbandingan: pekerjaan yang sama di tiga perangkat

Pekerjaan	QGIS dan PostGIS	ArcGIS Pro	ArcMap 10.8
Hitung kejadian per petak	<code>ST_Intersects + GROUP BY</code> , atau Count points in polygon	Summarize Within [CEK]	Spatial Join [CEK]
Dalam jarak tertentu	<code>ST_DWithin</code>	Select Layer By Location [CEK]	Select By Location [CEK]
Terdekat	<code>ORDER BY <-> LIMIT 1</code>	Generate Near Table [CEK]	Near [CEK]
SQL pada basis data	DB Manager, Execute SQL	Query Layer [CEK]	Add Query Layer [CEK]

BAB 3: Jalan Tercepat dan Jangkauan Layanan: Analisis Jaringan

Studi kasus: "Berapa menit dari pos ke lokasi kejadian?"

Ada laporan tebangan liar di sisi timur wilayah. Kepala Seksi ingin tahu rute terbaik dari Pos Jaga 1 ke Pos Jaga 4 dan berapa lama perjalanannya. Ia juga bertanya, "Dari kantor, bagian mana yang bisa dicapai dalam jarak 500 meter dan 1 kilometer lewat jalan?" Garis lurus di peta tidak menjawabnya, karena kendaraan harus mengikuti jalan. Satu ruas jalan juga sedang ditutup.

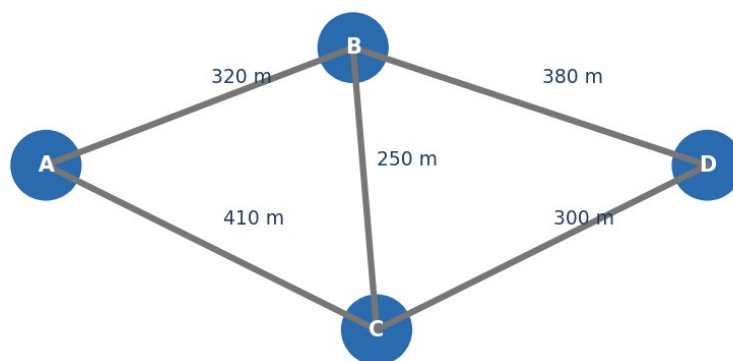
Konsep: jaringan dalam tiga kalimat

Analisis jaringan menghitung perjalanan menyusuri jalan, bukan melintas bebas. Bayangkan peta bus: titik-titik pemberhentian dihubungkan garis, dan tiap garis punya "ongkos" berupa jarak atau waktu. Komputer mencari jalur dengan ongkos total paling kecil.

Istilah baru:

- **Jaringan (graf):** kumpulan simpul dan ruas.
- **Simpul:** persimpangan atau ujung jalan.
- **Ruas:** potongan jalan di antara dua simpul.
- **Biaya (cost):** ongkos melewati ruas, bisa panjang (meter) atau waktu (detik).
- **Area layanan:** bagian jaringan yang bisa dicapai dari satu titik dengan biaya tertentu.

Ilustrasi 3.1: Simpul, ruas, dan biaya



Tabel ruas

id	source	target	cost
1	A	B	320
2	A	C	410
3	B	D	380
4	C	D	300
5	B	C	250

Simpul = persimpangan; ruas = potongan jalan; biaya = panjang atau waktu

Skema; bukan tangkapan layar

Syarat utama: **ruas harus bersambung di simpul**. Jalan yang berhenti 4 m sebelum persimpangan, bagi komputer, tidak tersambung.

BAGIAN A | QGIS

Bagian A: Jaringan di QGIS dan PostGIS

A1. Rute terpendek dan tercepat di QGIS

1. Muat `Jalan.gpkg` dan `Fasilitas.gpkg`.
2. Buka **Processing ► Toolbox**, cari **Shortest path (point to point)**.
3. Isi **Vector layer representing network** dengan `Jalan`. Pilih **Path type to calculate: Shortest**.
4. Klik tombol titik di **Start point**, lalu klik di dekat Pos Jaga 1. Lakukan hal serupa untuk **End point** di Pos Jaga 4.
5. Klik **Run**. Satu garis rute muncul. Catat nilai **Travel cost** di log.

Untuk waktu tempuh, ubah **Path type** menjadi **Fastest**, lalu isi **Speed field** dengan `kecepatan_kmj`. Biaya yang keluar dalam **jam**, jadi kalikan 60 untuk menit.

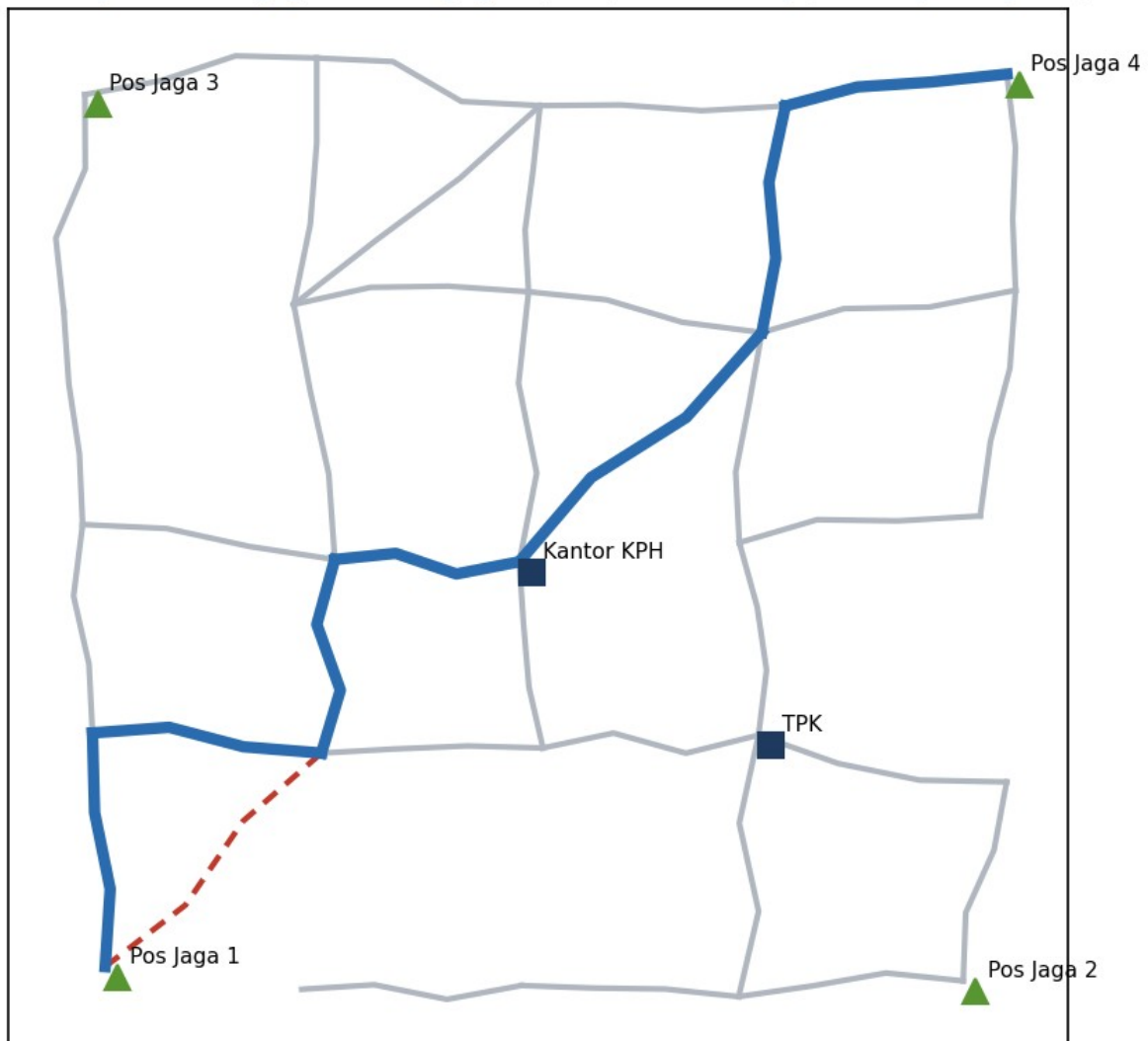
Alat yang sama dipanggil lewat skrip `m2_03a_rute_qgis.py` (`native:shortestpathpointtopoint`). Hasil uji:

Skenario	Hasil
Terpendek, semua ruas (termasuk yang ditutup)	2.987,7 m
Terpendek, ruas ditutup dibuang	3.316,2 m
Tercepat, ruas ditutup dibuang	6,4 menit

Satu ruas yang ditutup menambah jarak sekitar 330 m. Cara membuangnya: sebelum memuat layer, beri **filter** `status <> 'ditutup'` (klik kanan layer ► **Filter**). Lewat skrip, tambahkan `subset=status<>'ditutup'` di jalur layer.

Ilustrasi 3.2: Rute terpendek di jaringan sintetis

Rute terpendek Pos Jaga 1 ke Pos Jaga 4 (biru); ruas ditutup (merah putus-putus)



Skema dari data sintetis; bukan tangkapan layar

A2. Celah kecil, dampak besar

Muat `Jalan_celah.gpkg`. Satu ruas di dalamnya berhenti 4 m sebelum simpul. Ulangi rute terpendek (ruas ditutup dibuang). Hasil uji:

Keadaan	Jarak
Jaringan benar	3.316,2 m
Jaringan bercelah 4 m, Topology tolerance = 0	3.351,3 m
Jaringan bercelah 4 m, Topology tolerance = 5	3.316,2 m

Dengan toleransi 0, rute memutar karena ruas itu tak tersambung (selisih 35,1 m). **Topology tolerance** (toleransi topologi) menganggap ujung yang berjarak hingga 5 satuan peta sebagai satu simpul. Satuan

di sini meter, karena layernya UTM. Cara lain: rapikan datanya dengan **Snap geometries to layer** sebelum analisis.

A3. Area layanan di QGIS

1. Cari **Service area (from point)**.
2. Pilih jaringan, strategi **Shortest**, dan **Travel cost** 500 (meter).
3. Isi **Start point** di dekat Kantor KPH. Centang **Include upper/lower bound points** bila Anda ingin titik batasnya.
4. Klik **Run**. Anda mendapat garis jalan yang terjangkau.

Catatan hasil uji: garis keluaran berupa potongan jalan antar-titik-belok yang ujung awalnya terjangkau. Potongan terakhir bisa sedikit melewati batas. Jangan menghitung panjang jalan terjangkau dari garis itu begitu saja. Untuk batas yang tepat, pakai titik batas (bound points) atau pgRouting di bawah. [kemungkinan: kesimpulan dari pengamatan keluaran penulis]

A4. Jaringan di PostGIS dengan pgRouting

pgRouting adalah ekstensi PostgreSQL untuk analisis jaringan. Skrip `m2_03b_pgrouting.sql` mengerjakan semuanya.

1. Aktifkan ekstensi:

```
sql
CREATE EXTENSION IF NOT EXISTS pgrouting;
```

2. Buat tabel simpul dari ujung-ujung ruas, lalu tabel ruas yang tahu simpul awal (`source`) dan akhir (`target`):

```
sql
CREATE TABLE kph.jalan_simpul AS
SELECT ROW_NUMBER() OVER (ORDER BY ST_X(g), ST_Y(g))::int AS id, g AS geom
FROM (SELECT DISTINCT ST_StartPoint(ST_GeometryN(geom, 1)) AS g FROM kph.jalan
      UNION
      SELECT DISTINCT ST_EndPoint(ST_GeometryN(geom, 1)) FROM kph.jalan) AS t;

CREATE TABLE kph.jalan_ruas AS
SELECT j.gid AS id, a.id AS source, b.id AS target,
       ST_Length(j.geom) AS panjang_m,
       ST_Length(j.geom) / (j.kecepatan_kmj * 1000.0 / 3600.0) AS waktu_dtk,
       j.status, j.geom
FROM kph.jalan j
JOIN kph.jalan_simpul a ON ST_Equals(a.geom, ST_StartPoint(ST_GeometryN(j.geom, 1)))
JOIN kph.jalan_simpul b ON ST_Equals(b.geom, ST_EndPoint(ST_GeometryN(j.geom, 1)));
```

Hasil uji: 37 ruas dan 25 simpul. Cara ini cocok bila ujung ruas **persis sama**. Untuk data yang ujungnya hampir sama, bersihkan dulu (Bagian A2).

3. Cari simpul terdekat dari tiap fasilitas:

```
sql
SELECT f.nama, (SELECT s.id FROM kph.jalan_simpul s ORDER BY s.geom <-> f.geom LIMIT 1) AS simpul_id
```

```
FROM kph.fasilitas f ORDER BY f.nama;
```

Hasil uji: Kantor KPH simpul 11, Pos Jaga 1 simpul 5, Pos Jaga 4 simpul 24.

4. Rute terpendek dengan algoritma Dijkstra:

sql

```
SELECT MAX(agg_cost) AS panjang_m, COUNT(*) FILTER (WHERE edge > 0) AS jumlah_ruas
FROM pgr_dijkstra(
  'SELECT id, source, target, panjang_m AS cost FROM kph.jalan_ruas WHERE status <> ''ditutup'',
  5, 24, false);
```

Argumen `false` berarti jaringan tak berarah (jalan dua arah). Hasil uji: 3.296,9 m lewat 7 ruas. Tanpa membuang ruas ditutup: 2.976,2 m lewat 6 ruas. Tercepat (kolom `waktu_dtk` sebagai biaya): 6,37 menit.

5. Jangkauan layanan dari Kantor KPH:

sql

```
SELECT COUNT(*) FROM pgr_drivingDistance(
  'SELECT id, source, target, panjang_m AS cost FROM kph.jalan_ruas WHERE status <> ''ditutup'',
  11, 500::float, false);
```

Hasil uji: 3 simpul terjangkau dalam 500 m dan 10 simpul dalam 1.000 m.

6. Periksa apakah jaringan terputus:

sql

```
SELECT component, COUNT(*) FROM pgr_connectedComponents(
  'SELECT id, source, target, panjang_m AS cost FROM kph.jalan_ruas')
GROUP BY component;
```

Hasil uji: satu komponen berisi 25 simpul. Artinya semua simpul saling terhubung. Bila ada dua komponen atau lebih, ada jalan yang terputus.

A5. Apakah QGIS dan pgRouting sepakat?

Titik awal di QGIS berasal dari fasilitas yang berjarak sekitar 25 m dari jalan, sehingga angkanya sedikit berbeda dari pgRouting (yang mulai dari simpul). Agar adil, skrip `m2_03c_silang_jaringan.py` memberi kedua alat **titik yang sama persis** (simpul 5 dan 24). Hasil uji:

Ukuran	QGIS	pgRouting
Terpendek	3.296,93 m	3.296,93 m
Tercepat	6,373 menit	6,373 menit
Persimpangan terjangkau 500 m	3	3
Persimpangan terjangkau 1.000 m	10	10

Hasilnya sama. Dua alat berbeda yang sepakat memberi rasa aman bahwa jaringan dan biayanya dibaca dengan benar.

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro

Analisis jaringan di Pro memerlukan **kumpulan data jaringan** (network dataset) dan ekstensi **Network Analyst**. Dokumentasi Esri menyebut alat **Make Route Analysis Layer** dan **Make Service Area Analysis Layer** untuk membuat lapisan analisis. Pada alat yang kedua, parameter pentingnya adalah sumber data jaringan, **Travel Mode**, **Travel Direction**, dan **Cutoffs**. Penulis membaca ini dari dokumentasi, belum menjalankannya.

1. Salin `Jalan` ke geodatabase, di dalam sebuah **feature dataset**. [CEK]
2. Jalankan alat **Create Network Dataset**, pilih feature class jalan sebagai sumber, dan atur atribut biaya (panjang dan waktu). Bangun jaringan dengan **Build Network**. [CEK]
3. Jalankan **Make Route Analysis Layer** dengan jaringan itu. Tambahkan lokasi lewat **Add Locations** (Pos Jaga 1 dan 4), lalu **Solve**. [CEK]
4. Untuk area layanan, pakai **Make Service Area Analysis Layer**, isi **Cutoffs** dengan 500 dan 1000, lalu **Add Locations** dan **Solve**. [CEK]
5. Ruas ditutup dimasukkan sebagai **barrier** atau lewat pembatas (restriction) pada atribut jaringan. [CEK]

Pro tidak membaca tabel `kph.jalan_ruas` dari pgRouting sebagai jaringan. Tidak ada padanan langsung; analisis jaringan Pro bergantung pada network dataset.

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

1. Aktifkan ekstensi **Network Analyst** (**Customize ► Extensions**), lalu tampilkan toolbar **Network Analyst**. [CEK]
2. Buat network dataset dari feature class jalan di ArcCatalog (klik kanan feature dataset ► **New ► Network Dataset**). [CEK]
3. Di toolbar, pilih **New Route** atau **New Service Area**, tambahkan lokasi, lalu klik **Solve**. [CEK]

Cek paham

1. Mengapa rute dari QGIS dan pgRouting bisa berbeda beberapa belas meter bila titik awalnya tidak sama?
2. Satu ruas jalan berhenti 4 m sebelum persimpangan. Apa akibatnya pada rute, dan apa dua cara memperbaikinya?
3. `pgr_connectedComponents` mengembalikan dua komponen. Apa artinya?

Jawaban:

1. QGIS menempelkan titik ke titik jalan terdekat, sedangkan pgRouting mulai dari simpul. Selisih kecil itu adalah ruas pendek di antara keduanya.
2. Ruas itu tidak dianggap tersambung, jadi rute memutar atau tidak ditemukan. Perbaiki dengan toleransi topologi atau dengan merapikan geometri (snap) sebelum analisis.
3. Jaringan terbagi dua bagian yang tidak saling terhubung. Ada jalan yang terputus atau data yang salah.

Kesalahan umum

- **Mengira jalan yang tampak bersambung pasti tersambung.** Periksa dengan komponen terhubung atau coba rute di dua sisi celah.
- **Lupa membuang ruas yang ditutup.** Rute melewati jalan yang tak bisa dipakai.
- **Menjumlahkan panjang garis area layanan QGIS sebagai jarak terjangkau.** Potongan terakhir dihitung utuh. Pakai titik batas.
- **Mengira biaya "Fastest" QGIS dalam menit.** Satuannya jam.

Ringkasan dan latihan

Ringkasan: jaringan = simpul + ruas + biaya. Rute terpendek mencari biaya total terkecil. Area layanan mencari semua yang terjangkau dalam batas biaya. Syaratnya ruas bersambung di simpul.

Latihan: hitung rute terpendek dari Kantor KPH ke Pos Jaga 2 dan ke TPK, dengan dan tanpa ruas ditutup. Bandingkan hasil QGIS dan pgRouting memakai simpul yang sama.

Tabel perbandingan: analisis jaringan

Pekerjaan	QGIS	PostGIS dan pgRouting	ArcGIS Pro	ArcMap 10.8
Rute terpendek	Shortest path (point to point)	<code>pgr_dijkstra</code>	Make Route Analysis Layer [CEK]	New Route [CEK]
Area layanan	Service area (from point)	<code>pgr_drivingDistance</code>	Make Service Area Analysis Layer [CEK]	New Service Area [CEK]
Bentuk jaringan	layer garis biasa	tabel <code>source</code> dan <code>target</code>	network dataset	network dataset
Syarat tambahan	tidak ada	ekstensi pgRouting	ekstensi Network Analyst	ekstensi Network Analyst

BAB 4: Ke Mana Air Mengalir: Hidrologi dari DEM

Studi kasus: "Petak mana yang dilewati sungai?"

Kepala Seksi akan merencanakan gorong-gorong dan jalur patroli. Ia butuh peta sungai dan batas daerah tangkapan air di wilayah KPH Contoh. Survei lapangan untuk semua sungai kecil terlalu mahal. Padahal Anda sudah punya DEM. Dari tinggi tanah saja, komputer bisa menebak ke mana air mengalir.

Konsep: hidrologi DEM dalam tiga kalimat

Hidrologi DEM mengikuti satu aturan sederhana: air di sebuah kotak (piksel) mengalir ke tetangga yang paling curam menurun. Bayangkan menuang segelas air di atas papan miring berpetak. Petak yang banyak menerima kiriman air dari hulu adalah calon sungai, dan seluruh petak hulu yang mengirim air ke satu titik membentuk satu daerah tangkapan.

Istilah baru:

- **Lubang (sink)**: sel yang lebih rendah dari semua tetangganya, jadi air "terjebak".
- **Arah aliran**: ke tetangga mana tiap sel mengalirkan air (metode D8: delapan arah).
- **Akumulasi aliran**: jumlah sel hulu yang mengalir lewat sebuah sel.
- **Ambang sungai**: batas akumulasi yang menentukan sel mana disebut sungai.
- **DAS** (daerah aliran sungai): seluruh area yang mengalir ke satu titik keluar (outlet).

Ilustrasi 4.1: Alur hidrologi dari DEM

Air selalu mengalir ke tetangga yang paling curam menurun



Skema; bukan tangkapan layar

Resolusi DEM: DEM latihan berpiksel 10 m (200 x 200 sel). Satu sel = 100 m². Jadi 500 sel = 5 ha. Ambang sungai 500 sel berarti "alur yang menampung minimal 5 ha".

BAGIAN A | QGIS

Bagian A: Hidrologi di QGIS

Semua langkah ada di skrip `m2_04a_hidrologi.py`. Lewat jendela QGIS, alat yang sama ada di **Processing ► Toolbox**. Alat GRASS (`r.watershed`, `r.stream.extract`, `r.water.outlet`) memerlukan penyedia GRASS aktif. [CEK: ketersediaan GRASS di instalasi Anda]

A1. Mengisi lubang

1. Muat `DEM_10m.tif`.
2. Buka **Fill sinks (Wang & Liu)**. Isi **Minimum slope** dengan 0,1 (derajat, nilai awal).
3. Klik **Run**. Anda mendapat DEM terisi (`DEM_terisi.tif`).

Hasil uji: DEM asli punya **167 lubang**. Sesudah diisi, jumlahnya **0**. Sebanyak 1.994 sel naik, dengan kenaikan terbesar 13,67 m. Mengapa sebanyak itu? Delapan lubang sengaja ditanam saat DEM dibuat, sisanya muncul dari derau acak pembuatan DEM. Pada DEM nyata hal serupa terjadi karena kesalahan ukur.

A2. Arah dan akumulasi aliran

1. Buka **r.watershed** (GRASS).
2. **Elevation**: `DEM_terisi`. **Minimum size of exterior watershed basin**: 500.
3. Centang **Enable Single Flow Direction (D8) flow** dan **Use positive flow accumulation even for likely underestimates**.
4. Simpan **Number of cells that drain through each cell** (`Akumulasi.tif`), **Drainage direction** (`Arah.tif`), dan **Unique label for each watershed basin**.
5. Klik **Run**.

Hasilnya: akumulasi maksimum 25.581 sel (255,8 ha). Sel dengan angka besar membentuk alur sungai.

A3. Membentuk sungai

1. Buka **r.stream.extract**.
2. **Input map: elevation map**: `DEM_terisi`. **Input map: accumulation map**: `Akumulasi`. **Minimum flow accumulation for streams**: 500.
3. Simpan **Unique stream ids (vect)**. Pada **v.out.ogr output type** pilih **line**.
4. Klik **Run**.

Hasil uji: **30 segmen sungai, total 7,75 km**. Catatan: bila Anda membiarkan pilihan jenis keluaran pada **auto**, berkas ikut memuat titik simpul dan memasukkannya ke PostGIS gagal. Pilih **line**.

A4. Daerah aliran di satu titik keluar

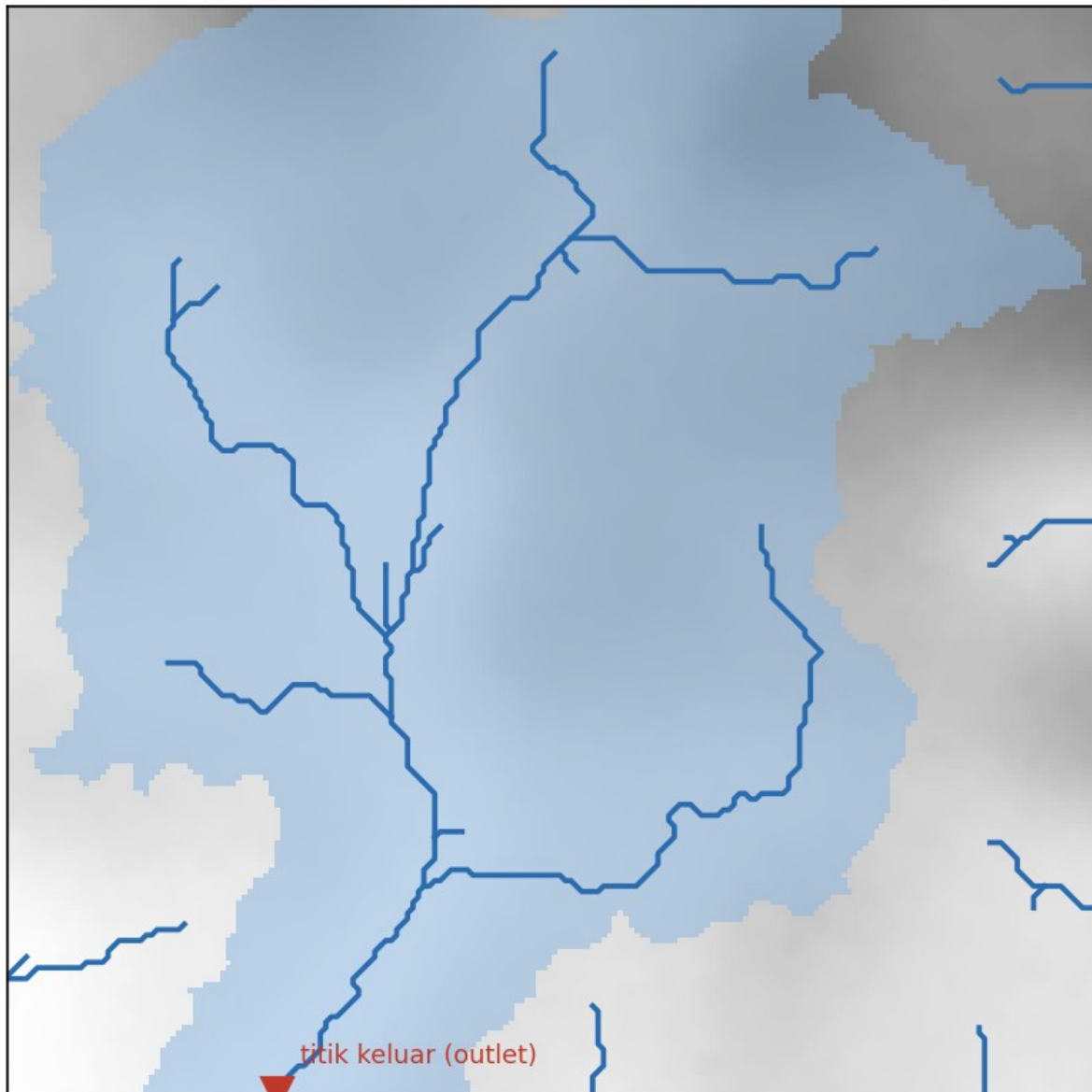
1. Tentukan titik keluar. Skrip memilih sel tepi dengan akumulasi terbesar: baris 199, kolom 49, koordinat X = 312495,0 dan Y = 9996005,0.
2. Buka **r.water.outlet**. **Drainage direction raster**: `Arah`. **Coordinates of outlet point**: isi koordinat di atas.

3. Klik **Run**, lalu ubah hasil menjadi poligon dengan **Polygonize (raster to vector)** dan lebur dengan **Dissolve**.

Hasil uji: DAS berisi **26.211 sel = 262,11 ha**, atau 65,5% wilayah.

Ilustrasi 4.2: Sungai dan DAS di KPH Contoh

Sungai (garis biru) dan DAS di titik keluar (biru muda)



Skema dari data sintetis; bukan tangkapan layar

A5. Memeriksa hasil dengan hitungan sendiri

Jangan percaya hasil alat tanpa diperiksa. Penulis menulis ulang aliran D8 dengan NumPy murni pada DEM terisi yang sama. Hasilnya:

Ukuran	Alat QGIS dan GRASS	Hitungan NumPy
Sel hulu titik keluar (DAS)	26.211	26.230
Selisih		19 sel (0,07%)
Sel DAS yang sama di kedua hasil	26.201	

Luas DAS cocok dengan sangat baik. Tetapi **nilai raster akumulasi** dari `r.watershed` di titik keluar adalah 25.581, sekitar 2,5% lebih rendah daripada 26.230. Sebanyak 86,1% sel punya akumulasi identik di kedua hasil. Penyebab selisih itu belum penulis telusuri. [belum diverifikasi] Pegangan praktis: **hitung luas DAS dari raster DAS (`r.water.outlet`)**, bukan dari nilai maksimum akumulasi.

A6. Sungai dan DAS dikawinkan dengan petak lewat SQL

Masukkan hasil ke PostGIS:

terminal

```
ogr2ogr -f PostgreSQL PG:"dbname=kph_contoh" Sungai.gpkg Sungai -nln kph.sungai -nlt MULTILINESTRING -lco GEOMETRY_NAME=geom -lco FID=gid -overwrite
ogr2ogr -f PostgreSQL PG:"dbname=kph_contoh" DAS_utama.gpkg -nln kph.das_utama -nlt MULTIPOLYGON -lco GEOMETRY_NAME=geom -lco FID=gid -overwrite
```

Lalu jalankan kueri dari `m2_04b_sungai_petak.sql`:

sql

```
SELECT p.kode,
       ROUND(COALESCE(SUM(ST_Length(ST_Intersection(p.geom, s.geom))), 0)::numeric, 0) AS sungai_m
FROM kph.petak p LEFT JOIN kph.sungai s ON ST_Intersects(p.geom, s.geom)
GROUP BY p.kode ORDER BY sungai_m DESC LIMIT 5;
```

Hasil uji:

Petak	Panjang sungai (m)
P-12	1.089
P-17	965
P-22	662
P-03	647
P-19	571

Hasil lain dari berkas yang sama:

- **Petak di luar DAS:** hanya P-25.
- **Longsor dalam 30 m dari sungai:** 3 dari 40 kejadian.
- **Titik potong jalan dan sungai:** 26. Itulah calon lokasi gorong-gorong atau jembatan untuk Kepala Seksi.

Kepala Seksi mendapat jawabannya: P-12 dan P-17 paling banyak dilalui sungai, dan ada 26 titik yang perlu dicek untuk gorong-gorong.

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro

Dokumentasi Esri mencantumkan alat hidrologi dalam kotak peralatan **Spatial Analyst Tools** (perlu ekstensi Spatial Analyst):

- **Fill**: mengisi lubang pada raster permukaan.
- **Flow Direction**: arah aliran dengan metode D8, MFD, atau D-Infinity.
- **Flow Accumulation**: akumulasi aliran, bisa diberi bobot.
- **Stream Order**: memberi urutan pada ruas alur.
- **Stream to Feature**: mengubah raster alur menjadi garis.
- **Snap Pour Point**: menempelkan titik keluar ke sel akumulasi tertinggi dalam jarak tertentu.
- **Watershed**: daerah tangkapan di atas titik keluar.
- **Basin**: semua cekungan sekaligus.

Urutan kerja yang setara dengan Bagian A: [CEK]

1. **Fill** pada DEM.
2. **Flow Direction** (D8) pada DEM terisi.
3. **Flow Accumulation** pada raster arah.
4. Ambil sel akumulasi ≥ 500 dengan **Con** atau **Raster Calculator**, lalu **Stream Order** dan **Stream to Feature**.
5. **Snap Pour Point** untuk titik keluar, lalu **Watershed**, dan ubah ke poligon dengan **Raster to Polygon**.

Pro memberi urutan alur (Strahler) lewat **Stream Order**. Keluaran `r.stream.extract` yang disimpan penulis hanya memuat nomor segmen. Nilai urutan alur tidak diperiksa di QGIS. [belum diverifikasi]

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

Alat dan urutannya sama, di **Spatial Analyst Tools** ► **Hydrology**: **Fill**, **Flow Direction**, **Flow Accumulation**, **Stream Order**, **Stream to Feature**, **Snap Pour Point**, **Watershed**. [CEK]

Ekstensi **Spatial Analyst** harus aktif (**Customize** ► **Extensions**). [CEK]

Cek paham

1. Mengapa DEM harus diisi lubangnya sebelum menghitung aliran?
2. Ambang sungai 500 sel pada piksel 10 m berarti berapa hektare?
3. Mengapa luas DAS sebaiknya dihitung dari raster DAS, bukan dari nilai akumulasi terbesar?

Jawaban:

1. Lubang menjebak air. Aliran berhenti di dalam lubang dan alur sungai terputus.
2. $500 \text{ sel} \times 100 \text{ m}^2 = 50.000 \text{ m}^2 = 5 \text{ ha}$.
3. Pada uji penulis, nilai akumulasi maksimum lebih rendah 2,5% daripada jumlah sel yang benar-benar mengalir ke titik keluar. Raster DAS cocok dengan hitungan perbandingan.

Kesalahan umum

- **Melewatkan pengisian lubang.** Sungai terpotong-potong.
- **Ambang sungai terlalu kecil.** Jaringan sungai jadi seperti urat daun dan memuat alur yang tidak nyata. Coba ambang yang cocok dengan pengamatan lapangan.
- **Titik keluar tidak tepat di alur.** DAS yang keluar sangat kecil. Pilih sel dengan akumulasi tertinggi di sekitarnya (itu gunanya **Snap Pour Point**).
- **Memakai DEM berkoordinat derajat.** Satuan jarak dan luas jadi salah. Pakai proyeksi UTM.

Ringkasan dan latihan

Ringkasan: isi lubang, hitung arah dan akumulasi, tentukan ambang sungai, pilih titik keluar, ambil DAS. Selalu periksa hasil dengan cara lain.

Latihan: ulangi dengan ambang 250 dan 1.000 sel. Catat jumlah segmen dan panjang total sungai. Mana yang menurut Anda paling masuk akal, dan mengapa?

Tabel perbandingan: hidrologi DEM

Langkah	QGIS	ArcGIS Pro	ArcMap 10.8
Isi lubang	Fill sinks (Wang & Liu)	Fill	Fill [CEK]
Arah dan akumulasi	<code>r.watershed</code> (GRASS)	Flow Direction, Flow Accumulation	sama dengan Pro [CEK]
Sungai	<code>r.stream.extract</code>	Con, Stream Order, Stream to Feature [CEK]	sama dengan Pro [CEK]
DAS di satu titik	<code>r.water.outlet</code>	Snap Pour Point, Watershed [CEK]	sama dengan Pro [CEK]
Urutan alur (Strahler)	tidak diuji (hanya nomor segmen disimpan)	Stream Order	Stream Order [CEK]

BAB 5: Menebak di Antara Titik: Interpolasi

Studi kasus: "Hujan di petak yang tidak punya stasiun"

KPH Contoh punya 30 stasiun hujan, tetapi 25 petaknya tidak punya satu pun. Untuk menilai kesesuaian lahan, Kepala Seksi butuh peta curah hujan yang menutup seluruh wilayah. Anda harus menebak nilai di tempat yang tidak diukur. Pertanyaannya: metode tebakan mana yang paling bisa dipercaya?

Konsep: interpolasi dalam tiga kalimat

Interpolasi menebak nilai di tempat yang tidak diukur dari nilai di tempat yang diukur. Bayangkan menebak suhu kamar dari tiga termometer di sudut-sudutnya. Tebakan yang baik memberi bobot lebih besar kepada termometer yang dekat.

Istilah baru:

- **IDW** (inverse distance weighting): bobot berbanding terbalik dengan jarak, dipangkatkan **p**. Makin besar **p**, makin dekat titik makin dominan.
- **TIN**: jaring segitiga yang menghubungkan titik, nilai di dalam segitiga dihitung lurus dari ketiga sudutnya.
- **Kriging**: tebakan berbobot yang bobotnya dihitung dari pola kemiripan antartitik (semivariogram).
- **Titik latih dan titik uji**: sebagian titik dipakai menebak, sebagian disimpan untuk menilai tebakan.
- **RMSE**: rata-rata besar selisih tebakan dan kenyataan. Makin kecil makin baik.

Untuk latihan ini, 30 stasiun dibagi: **24 titik latih** dan **6 titik uji**. Nilai 6 stasiun uji tidak boleh dipakai saat menebak. Anda hanya mengintip mereka saat menilai.

Hanya enam titik uji: Enam titik adalah contoh yang sangat kecil. Peringkat metode bisa berubah bila titik ujinya diganti. Dalam pekerjaan nyata, pakai validasi silang (tinggalkan satu titik) sebagai pendamping, seperti di A5.

BAGIAN A | QGIS

Bagian A: Interpolasi di QGIS

Semua metode di bawah dijalankan oleh skrip `m2_05a_interpolasi.py`. Keluarannya ada di folder `hasil`.

A1. IDW

1. Buka **Processing ► Toolbox**, cari **IDW interpolation**.
2. Pada **Input layer(s)**, klik tombol **+**, pilih layer stasiun latih, atribut `ch_tahunan_mm`, dan jenis **Points**.
3. **Distance coefficient P**: 2. **Extent**: samakan dengan `DEM_10m` (**Calculate from layer**). **Output raster size**: 10.
4. Klik **Run**.

Ulangi dengan $P = 1$ dan 3. Perhatikan: kolom pertama di daftar atribut yang dilihat QGIS bisa berupa `fid`, sehingga nomor kolom bergeser satu. Pastikan atribut yang dipilih memang `ch_tahunan_mm`. (Saat membuat skrip, penulis sempat memilih kolom keliru dan hasilnya memuat angka elevasi.)

A2. TIN

Alat **TIN interpolation**. Pilih metode **Linear** atau **Clough-Toucher (cubic)**. TIN hanya menghitung di dalam jaring segitiga. Titik di luar tepi jaring tidak mendapat nilai.

A3. Spline

QGIS memuat alat GRASS **v.surf.bspline** (B-spline) dan **v.surf.rst** (spline dengan tegangan). Nilai awal skrip: `v.surf.bspline` dengan langkah 500 m dan metode bikubik, `v.surf.rst` dengan `tension` 40 dan `smooth` 0,5. Parameter ini hasil pilihan penulis, bukan angka baku.

A4. Kriging

QGIS 4.0.2 yang diuji penulis tidak punya alat kriging bawaan di toolbox. Skrip penulis menyediakan kriging biasa dengan NumPy:

1. Hitung selisih nilai antarpasang titik, kelompokkan menurut jarak (semivariogram empiris).
2. Cocokkan model eksponensial ke kelompok itu.
3. Selesaikan sistem persamaan kriging untuk tiap sel.

Hasil penyesuaian untuk data latih: *nugget* 0, *sill* parsial sekitar 66.870 mm², jangkauan 5.000 m. Angka jangkauan menempel pada batas atas yang penulis izinkan dalam penyesuaian. Artinya pola kemiripan berlangsung sangat jauh, wajar karena hujan di data ini naik mengikuti elevasi (suatu tren). Di luar skrip ini, kriging dapat memakai plugin atau alat pihak ketiga. [CEK]

A5. Regresi dengan elevasi lalu sisa dengan IDW

Hujan sering naik bersama ketinggian. Jadi Anda menyesuaikan garis lurus **hujan terhadap elevasi dan koordinat X**, lalu menginterpolasi sisanya (residu) dengan IDW $p = 2$ dan menjumlahkannya kembali. Hasil penyesuaian dari 24 titik latih:

KODE

$$CH = 1944,2 + 3,04 \times \text{elevasi} + 0,1371 \times (X - X_0) \quad R^2 = 0,912$$

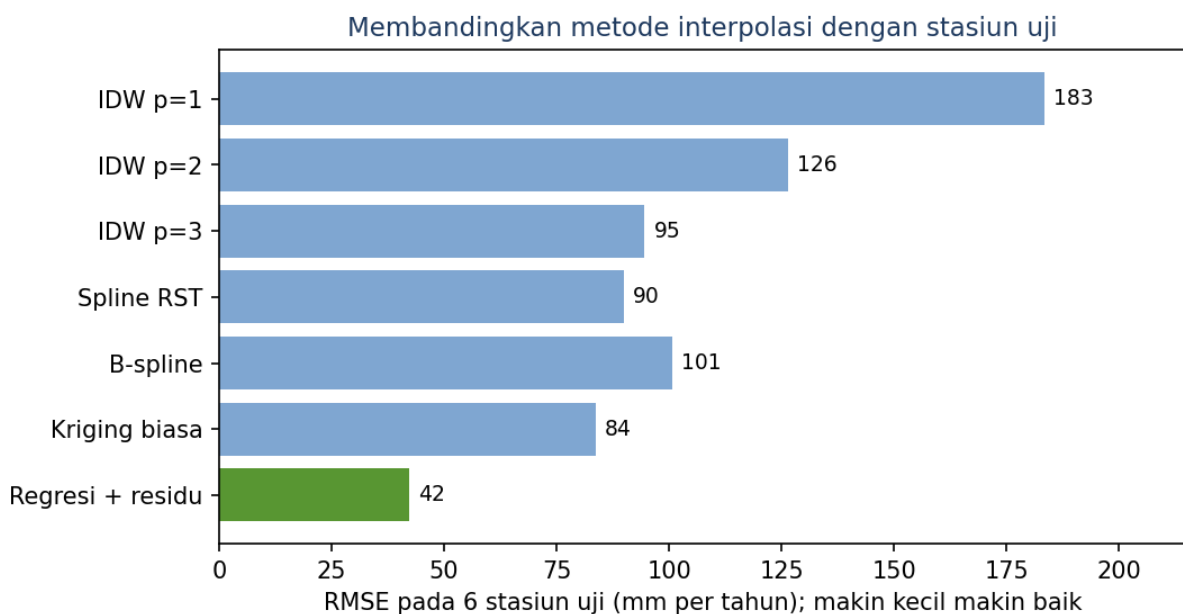
Artinya: tiap naik 1 m, hujan tahunan naik sekitar 3 mm. Elevasi tiap sel diambil dari DEM. Metode ini dikerjakan lewat skrip, bukan lewat tombol QGIS.

A6. Menilai: siapa yang paling dekat dengan kenyataan?

Skrip mengambil nilai tiap raster hasil di enam stasiun uji dan menghitung selisih terhadap pengamatan. Hasil uji:

Metode	RMSE (mm/th)	MAE	Bias	Titik uji terjangkau
IDW p = 1	183,4	175,2	+150,1	6
IDW p = 2	126,5	119,7	+97,4	6
IDW p = 3	94,6	83,8	+61,0	6
TIN linear	55,9	40,5	+38,6	2
TIN Clough-Tocher	98,3	74,6	-45,6	3
B-spline bikubik	100,7	84,1	+9,5	6
RST (spline tegangan)	89,9	76,2	+41,0	6
Kriging biasa	83,7	68,4	+33,5	6
Regresi elevasi + residu IDW	42,2	33,3	+9,6	6

Ilustrasi 5.1: Galat metode interpolasi



TIN tidak ditampilkan: hanya menutupi 2 dari 6 titik uji. Skema dari data sintetis; bukan tangkapan layar

Bacalah tabel ini dengan hati-hati:

- **TIN tidak adil dibandingkan.** Ia hanya mencakup 2 atau 3 dari 6 titik uji. RMSE-nya dihitung dari titik yang sedikit itu, jadi tidak ikut dalam peringkat.
- **IDW punya bias positif (menebak terlalu tinggi).** Penyebabnya belum diperiksa. Dugaan penulis: IDW tidak bisa menebak di luar rentang nilai titik latih, padahal hujan naik mengikuti elevasi. [kemungkinan]
- **Metode regresi menang karena datanya memang dibuat naik bersama elevasi.** Ini data sintetis. Pada data nyata, hubungan itu harus dibuktikan dulu.

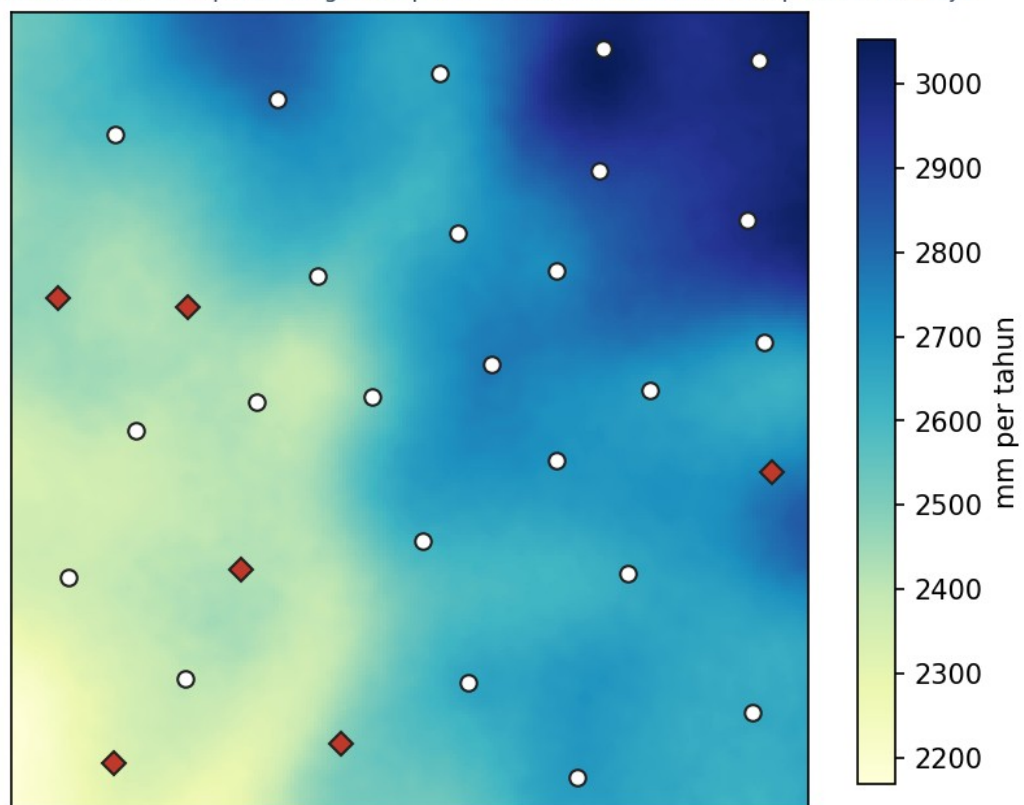
Validasi silang tinggalkan-satu pada 24 titik latih untuk IDW (RMSE menurut pangkat p):

p	0,5	1	2	3	4
RMSE	169,0	149,4	115,3	96,0	90,1

Pada data ini, p yang lebih besar memberi galat lebih kecil. Pangkat 2 adalah nilai awal umum, bukan jaminan terbaik.

Ilustrasi 5.2: Peta curah hujan hasil interpolasi

Curah hujan tahunan hasil interpolasi (lingkaran putih = stasiun latih, belah ketupat merah = uji)



Skema dari data sintetis; bukan tangkapan layar

Peta pemenang disimpan sebagai `hasil/CH_tahunan.tif` (200 x 200 sel, 10 m) dan dipakai di Bab 6.

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro

Kotak peralatan **Spatial Analyst Tools** memuat kelompok **Interpolation: IDW, Kriging, Spline, Natural Neighbor**, dan **Trend**. Pada IDW, dokumentasi Esri menyebut parameter **Power** dengan nilai awal 2, serta pilihan radius pencarian (variabel atau tetap) dan **barrier** berupa garis patahan. [Kriging, Spline, dan Trend: nama alat dari pengetahuan umum, belum dibaca di dokumentasi; Uji]

1. Siapkan dua layer dari stasiun: latih dan uji. Pisahkan lewat **Select By Attributes** (`peran = 'latih'`) dan ekspor. [CEK]
2. Jalankan **IDW** pada stasiun latih, `ch_tahunan_mm` sebagai **Z value field**. Isi **Power**. [CEK]
3. Jalankan **Kriging** (pilih model Spherical atau Exponential) dan **Spline**. [CEK]
4. Ambil nilai di titik uji dengan **Extract Values to Points**, lalu hitung selisihnya di tabel. [CEK]
5. Ekstensi **Geostatistical Analyst** menyediakan wizard dan validasi silang yang lebih lengkap, termasuk kriging bayes empiris. [CEK: ekstensi dan nama alat]

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

1. **Spatial Analyst Tools ► Interpolation: IDW, Kriging, Spline, Natural Neighbor, Trend**. [CEK]
2. **Extract Values to Points** untuk membaca nilai di titik uji. [CEK]
3. **Geostatistical Analyst ► Geostatistical Wizard** untuk kriging dengan validasi silang. [CEK]

Cek paham

1. Mengapa nilai enam stasiun uji tidak boleh ikut dipakai saat menebak?
2. TIN memberi nilai hanya di 2 dari 6 titik uji. Apa penyebabnya?
3. Mengapa metode regresi menang pada latihan ini, dan mengapa itu belum tentu berlaku pada data nyata?

Jawaban:

1. Bila ikut dipakai, tebakannya "mencontek". Nilai galat jadi terlalu bagus dan tidak menunjukkan kemampuan menebak tempat baru.
2. TIN hanya menghitung di dalam segitiga yang dibentuk titik latih. Titik uji di luar tepi jaring tidak punya nilai.
3. Data dibuat sehingga hujan naik bersama elevasi, jadi regresi elevasi cocok. Pada data nyata, hubungan itu harus diperiksa dulu.

Kesalahan umum

- **Menilai metode dengan titik yang dipakai untuk melatihnya.** Hasilnya selalu tampak sempurna (interpolator eksak).

- **Membandingkan RMSE dari jumlah titik yang berbeda.** Seperti pada TIN. Laporkan juga jumlah titik.
- **Memilih kolom atribut yang keliru.** Periksa nilai terkecil dan terbesar hasil. Hujan 120 sampai 275 jelas bukan milimeter, itu elevasi.
- **Memakai IDW untuk menebak jauh di luar sebaran titik.** Hasil tertarik ke rata-rata dan tidak pernah melampaui rentang data.

Ringkasan dan latihan

Ringkasan: interpolasi menebak nilai di tempat tak terukur. Sisihkan titik uji, bandingkan metode dengan RMSE, laporkan jumlah titik, dan curigai hasil yang terlalu bagus.

Latihan: jalankan IDW $p = 2$ dan $p = 3$ pada data latih, lalu hitung sendiri RMSE di enam stasiun uji dengan **Sample raster values**. Cocokkan dengan tabel A6.

Tabel perbandingan: interpolasi

Metode	QGIS	ArcGIS Pro	ArcMap 10.8
IDW	IDW interpolation	IDW [CEK]	IDW [CEK]
TIN	TIN interpolation	Create TIN, TIN to Raster [CEK]	sama [CEK]
Spline	<code>v.surf.bspline</code> , <code>v.surf.rst</code> (GRASS)	Spline [CEK]	Spline [CEK]
Kriging	tidak ada alat bawaan yang diuji; skrip NumPy	Kriging , Geostatistical Analyst [CEK]	Kriging , Geostatistical Analyst [CEK]
Nilai di titik uji	Sample raster values	Extract Values to Points [CEK]	sama [CEK]

BAB 6: Mencari Lahan Terbaik: Kesesuaian Lahan Multi-Kriteria

Studi kasus: "Di mana kita menanam jenis pohon yang baru?"

KPH Contoh akan mencoba satu jenis pohon baru. Kepala Seksi meminta daftar petak yang paling cocok. Lahan yang baik itu landai, tanahnya subur, hujannya cukup, dan dekat jalan. Tidak ada lahan yang unggul di semuanya. Anda harus menimbang empat hal sekaligus, dan menolak lahan yang memang tidak boleh ditanami.

Kriteria ini fiktif: Semua kriteria, skor, bobot, dan ambang di bab ini dibuat untuk latihan. Itu **bukan** standar kesesuaian lahan resmi dan **bukan** saran penanaman.

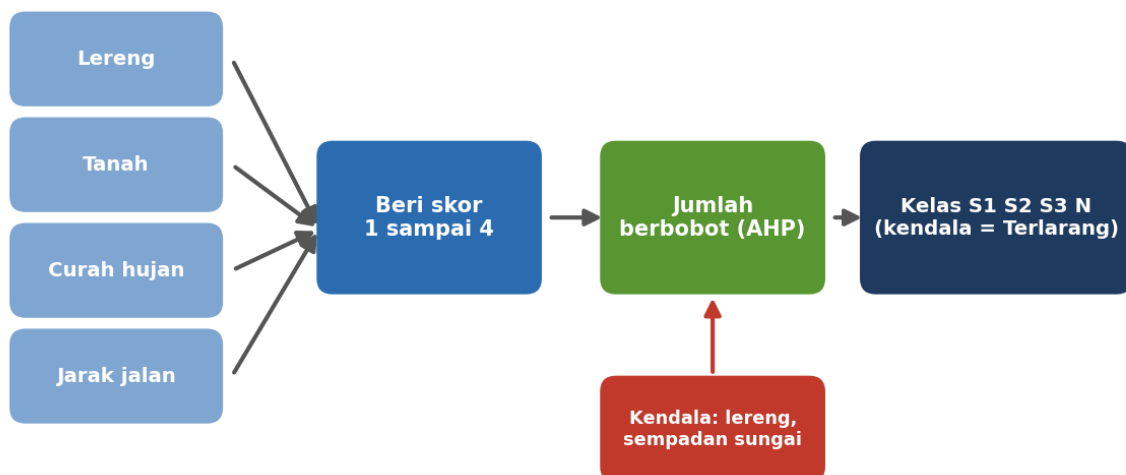
Konsep: tumpang tindih berbobot dalam tiga kalimat

Kesesuaian lahan multi-kriteria memberi tiap kriteria nilai (skor), mengalikannya dengan bobot kepentingan, lalu menjumlahkannya. Bayangkan menilai calon rumah: harga, jarak, dan keamanan masing-masing diberi angka, lalu dijumlah dengan bobot menurut selera Anda. **Kendala** adalah syarat mutlak, seperti "tidak boleh di tepi sungai", yang mencoret lokasi tanpa ditimbang.

Istilah baru:

- **Skor:** nilai 1 sampai 4 untuk tiap kelas kriteria (4 terbaik).
- **Bobot:** seberapa penting satu kriteria; jumlah semua bobot 1.
- **AHP:** cara menurunkan bobot dari perbandingan berpasangan antarkriteria.
- **Kendala:** syarat yang meniadakan lokasi, apa pun skornya.
- **Analisis kepekaan:** mencoba bobot lain untuk melihat apakah kesimpulan berubah.

Ilustrasi 6.1: Alur kesesuaian lahan



Skema; bukan tangkapan layar

Aturan skor yang dipakai di bab ini:

Kriteria	Skor 4	Skor 3	Skor 2	Skor 1	Kendala
Lereng (%)	0 sampai 8	8 sampai 15	15 sampai 25	25 sampai 40	40 atau lebih
Curah hujan (mm/th)	2.800 atau lebih	2.600 sampai 2.800	2.400 sampai 2.600	kurang dari 2.400	tidak ada
Tanah	Aluvial	Latosol	Podsolik	Regosol	tidak ada
Jarak ke jalan (m)	0 sampai 100	100 sampai 250	250 sampai 500	lebih dari 500	tidak ada
Jarak ke sungai (m)					30 atau kurang

BAGIAN A | QGIS

Bagian A: Kesesuaian lahan di QGIS

Semua langkah ada di skrip `m2_06a_kesesuaian.py`. Prasyaratnya `Sungai.gpkg` (Bab 4) dan `CH_tahunan.tif` (Bab 5), keduanya sudah ada di hasil.

A1. Menurunkan bobot dengan AHP

Bandingkan kriteria dua-dua memakai skala 1 sampai 9 (1 = sama penting, 3 = sedikit lebih penting, dan seterusnya). Pilihan penulis untuk latihan:

	Lereng	Tanah	Hujan	Jalan
Lereng	1	2	3	4
Tanah	1/2	1	2	3
Hujan	1/3	1/2	1	2
Jalan	1/4	1/3	1/2	1

Bobot adalah vektor eigen utama matriks ini, dinormalkan agar berjumlah 1. Hasil hitung skrip:

Kriteria	Bobot
Lereng	0,467
Tanah	0,277
Curah hujan	0,160
Jarak jalan	0,095

Pemeriksaan konsistensi: λ maksimum = 4,031, CI = 0,0103, **CR = 0,0115**. Batas kebiasaan CR adalah 0,1, jadi perbandingan ini cukup konsisten. (RI untuk empat kriteria = 0,90, dari tabel Saaty.)

A2. Menyiapkan lima raster kriteria

Semua raster memakai grid yang sama: 200 x 200 sel berukuran 10 m.

- Lereng.** Buka **Raster ► Analysis ► Slope**, pilih `DEM_10m`, dan centang **Express slope as percent instead of degrees**. Hasil: `Lereng_pct.tif`.
- Skor lereng.** Buka **Reclassify by table**. Isi tabel: min 0, maks 8, nilai 4; 8, 15, 3; 15, 25, 2; 25, 40, 1. Pilih **Range boundaries: min <= value < max**. Pilih **Use NoData when no range matches value** dan isi **Output NoData value** dengan 0. Sel dengan lereng 40 atau lebih tidak cocok ke baris mana pun, jadi menjadi 0.
- Skor hujan.** Reclassify `CH_tahunan` dengan tabel dalam bab ini (tipe keluaran Byte).
- Skor tanah.** Hitung kolom skor dari `jenis_tanah` dengan **Field calculator**:

KODE

```
CASE WHEN "jenis_tanah"='Aluvial' THEN 4 WHEN "jenis_tanah"='Latosol' THEN 3
  WHEN "jenis_tanah"='Podsolik' THEN 2 ELSE 1 END
```

Lalu ubah ke raster dengan **Rasterize (vector to raster)**: **Field to use for a burn-in value**: skor, lebar dan tinggi 200 sel, **Output extent** sama dengan DEM.

- Jarak ke jalan dan sungai.** Rasterkan garis (burn-in 1), lalu jalankan **Proximity (raster distance)** dengan **Distance units: Georeferenced coordinates**. Reclassify jarak jalan menjadi skor. Jarak sungai dipakai untuk kendala.

Jaga agar ruas jalan yang ditutup tidak ikut (filter `status<>'ditutup'`). Jarak raster dihitung dari pusat sel ke sel garis terdekat, jadi ketelitiannya sekitar satu sel (10 m).

A3. Kendala dan penjumlahan berbobot

Buka **Raster calculator** dari **Processing ► GDAL**, atau dari menu **Raster ► Raster Calculator**. Rumus kendala (A = skor lereng, B = jarak sungai):

KODE

```
logical_and(A>0, B>30)
```

Rumus jumlah berbobot (A lereng, B tanah, C hujan, D jalan):

KODE

```
A*0.467 + B*0.277 + C*0.160 + D*0.095
```

Rumus kelas akhir (A = skor, B = kendala; kelas 1 = S1, 2 = S2, 3 = S3, 4 = N, 0 = Terlarang):

KODE

```
(B==1)*(1*(A>=3.2) + 2*logical_and(A>=2.6, A<3.2) + 3*logical_and(A>=2.0, A<2.6) + 4*(A<2.0))
```

Jebakan NoData: Pada percobaan pertama penulis, 435 sel (4,35 ha) hilang dari hasil. Sel dengan lereng 40 atau lebih adalah NoData di skor lereng, dan kalkulator GDAL membuat semua hasil NoData bila satu masukannya NoData. Pemeriksaan dengan NumPy menunjukkan kecocokan hanya 98,9%. Perbaikan: tambahkan `--hideNoData` pada kolom **Additional command-line parameters** kalkulator. Setelah itu kecocokan 100%.

A4. Hasil

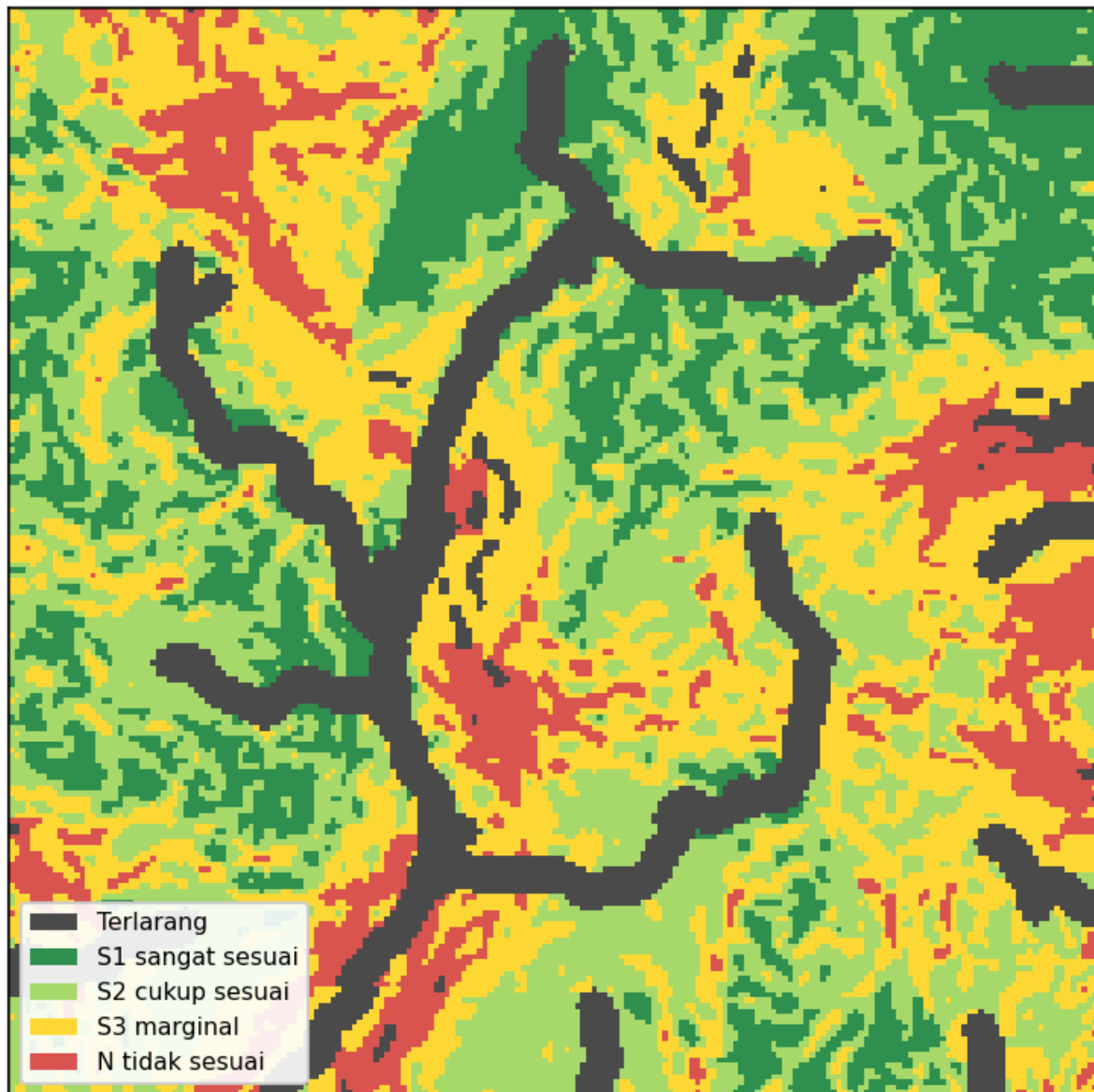
Hasil uji (1 sel = 0,01 ha):

Kelas	Luas (ha)	Persen
Terlarang (kendala)	53,21	13,3
S1 sangat sesuai	67,33	16,8
S2 cukup sesuai	123,45	30,9
S3 sesuai marginal	116,27	29,1
N tidak sesuai	39,74	9,9
Jumlah	400,00	100

Jumlahnya tepat 400 ha, tanpa sel NoData. Hasil kelas dari alat GDAL **sama persis (100% sel)** dengan hitungan NumPy murni.

Ilustrasi 6.2: Peta kesesuaian lahan

Kesesuaian lahan jenis pohon contoh (kriteria fiktif)



Skema dari data sintetis; bukan tangkapan layar

A5. Menimbang: analisis kepekaan

Bagaimana bila Anda tidak percaya bobot AHP dan memakai bobot sama rata (0,25 untuk tiap kriteria)? Hasil uji:

Skenario	Luas S1
Bobot AHP	67,33 ha
Bobot sama rata	81,76 ha
Sel yang pindah kelas	31,6%

Hampir sepertiga wilayah pindah kelas. Pesannya: **bobot adalah keputusan, bukan fakta**. Tulis dasar penentuannya di laporan, dan tunjukkan kepekaannya kepada pengambil keputusan.

A6. Angka per petak

Gunakan **Zonal histogram** dengan raster kelas dan layer `Petak`. Lima petak dengan S1 terluas:

Petak	Jenis	S1 (ha)	S2	S3	N	Terlarang
P-05	Jati	11,95	5,36	0,36	0,01	1,47
P-03	Mahoni	9,02	5,27	1,63	0,18	3,91
P-08	Jati	4,46	5,46	2,96	0,00	2,95
P-17	Jati	4,26	4,78	3,55	1,47	5,84
P-06	Akasia	4,24	4,95	1,83	0,01	2,53

Jumlah S1 di 25 petak 67,33 ha, sama dengan luas S1 seluruh wilayah. Kepala Seksi mendapat jawabannya: P-05 dan P-03 paling layak dilihat lebih dulu.

Bila kelas diubah menjadi poligon (**Polygonize** lalu **Dissolve** menurut kelas) dan dimasukkan ke PostGIS sebagai `kph.kesesuaian`, luas tiap kelas dari poligon sama dengan hasil raster (misalnya S1 = 67,33 ha). Bab 7 memakainya.

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro

Dokumentasi Esri menjelaskan alat **Weighted Overlay** (Spatial Analyst): raster masukan harus bertipe bilangan bulat, **bobot pengaruh dalam persen dan berjumlah 100**, tiap kelas diberi nilai pada **skala evaluasi** 1 sampai 9, dan alat ini memerlukan ekstensi Spatial Analyst.

1. **Slope** pada DEM, keluaran dalam persen. [CEK]
2. **Reclassify** lereng, hujan, dan jarak menjadi skor 1 sampai 4. Lereng 40 atau lebih diberi **NODATA**. [CEK]
3. **Euclidean Distance** dari jalan dan sungai. Beri kendala sungai lewat **Reclassify** atau **Con**. [CEK]
4. Untuk tanah, ubah poligon menjadi raster dengan **Polygon to Raster** memakai kolom skor. [CEK]
5. Jalankan **Weighted Overlay**. Isi bobot dalam persen: 47, 28, 16, 9 (pembulatan dari bobot AHP, jumlah 100). Hasil Pro bisa sedikit berbeda dari QGIS karena pembulatan bobot dan skala evaluasi. [CEK]
6. Pro juga menyediakan **Weighted Sum** (bobot desimal tanpa skala evaluasi). [CEK: parameter]
7. Perhitungan AHP (vektor eigen dan CR) tidak ada sebagai alat bawaan yang penulis ketahui. Hitung di luar, lalu masukkan bobotnya. [CEK]

Pro juga memiliki antarmuka **Suitability Modeler**. Penulis belum membaca dokumentasinya. [CEK]

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

1. **Spatial Analyst Tools ► Surface ► Slope** (keluaran Percent Rise). [CEK]
2. **Spatial Analyst Tools ► Reclass ► Reclassify**. [CEK]
3. **Spatial Analyst Tools ► Distance ► Euclidean Distance**. [CEK]
4. **Spatial Analyst Tools ► Overlay ► Weighted Overlay** atau **Weighted Sum**. [CEK]
5. Ekstensi **Spatial Analyst** harus aktif. [CEK]

Cek paham

1. Apa beda skor dan bobot?
2. Mengapa lereng 40% atau lebih dijadikan kendala dan bukan skor 0?
3. Pada hasil kelas muncul sel NoData yang seharusnya "Terlarang". Apa penyebab paling mungkin?

Jawaban:

1. Skor menilai kelas di dalam satu kriteria (lereng landai skor 4). Bobot menilai kepentingan antarkriteria (lereng lebih penting daripada jarak jalan).
2. Skor 0 tetap ikut dijumlahkan dengan kriteria lain, sehingga lokasi bisa lolos. Kendala mencoret lokasi apa pun skor lainnya.
3. Salah satu raster masukan punya NoData di sel itu, dan kalkulator meneruskan NoData ke hasil. Perlakukan NoData sebagai nilai (`--hideNoData`) atau isi dulu.

Kesalahan umum

- **Mengira bobot AHP sebagai kebenaran.** Itu pendapat yang dirumuskan. Selalu uji kepekaan.
- **Skor yang tidak sejajar.** Pastikan 4 selalu berarti terbaik di semua kriteria.
- **Grid raster berbeda-beda.** Semua kriteria harus sama ukuran sel dan sama batasnya. Pakai **Output extent** dan jumlah sel yang sama.
- **Memperlakukan kendala sebagai skor.** Kendala harus mencoret, bukan hanya menurunkan nilai.

Ringkasan dan latihan

Ringkasan: beri skor tiap kriteria, tentukan bobot (AHP), jumlahkan, coret lokasi yang terkena kendala, bagi menjadi kelas, hitung luas, dan uji kepekaan.

Latihan: ubah ambang kendala sungai dari 30 m menjadi 50 m. Hitung ulang luas S1 dan catat perubahannya. Jangan lupa memeriksa bahwa jumlah semua kelas tetap 400 ha.

Tabel perbandingan: kesesuaian lahan

Langkah	QGIS	ArcGIS Pro	ArcMap 10.8
Lereng	Slope (persen)	Slope [CEK]	Slope [CEK]
Skor	Reclassify by table	Reclassify [CEK]	Reclassify [CEK]
Jarak	Proximity (raster distance)	Euclidean Distance [CEK]	Euclidean Distance [CEK]
Jumlah berbobot	Raster calculator	Weighted Overlay, Weighted Sum	Weighted Overlay [CEK]
Bobot AHP	dihitung lewat skrip	di luar alat [CEK]	di luar alat [CEK]
Luas per petak	Zonal histogram	Tabulate Area [CEK]	Tabulate Area [CEK]

BAB 7: Menyatukan dan Menjaga: Proyek Terpadu dan Tata Kelola

Studi kasus: "Satu tabel untuk rapat besok"

Rapat perencanaan dimulai besok pagi. Kepala Seksi tidak mau membuka lima peta. Ia ingin **satu tabel**: tiap petak, berapa luas lahan sangat sesuai, berapa kejadian, berapa panjang sungai, dan berapa jauh dari kantor lewat jalan. Anda juga harus memastikan tabel itu aman: tidak bisa diubah sembarang orang, bisa dilacak bila ada perubahan, dan bisa dipulihkan bila server rusak.

Konsep: dari analisis ke layanan data

Proyek terpadu menggabungkan hasil banyak analisis dalam satu kueri. **Tata kelola** menjaga gudang data tetap benar dari waktu ke waktu. Bayangkan perpustakaan: bukan hanya rak buku (data), tetapi juga kartu anggota (hak akses), buku tamu (jejak perubahan), dan salinan arsip di gedung lain (cadangan).

Istilah baru:

- **Hak akses**: siapa boleh membaca, siapa boleh mengubah.
- **Trigger**: aturan yang berjalan otomatis saat data diubah.
- **Jejak perubahan (audit)**: catatan siapa mengubah apa dan kapan.
- **Cadangan (backup)**: salinan data untuk memulihkan bila terjadi kerusakan.
- **Pemulihan (restore)**: memasukkan kembali cadangan ke server.

Ilustrasi 7.1: Daur data di gudang



Skema; bukan tangkapan layar

BAGIAN A | QGIS

Bagian A: Proyek terpadu dan tata kelola di PostGIS

A1. Menyiapkan bahan

Pastikan tabel berikut sudah ada di skema kph: petak, kejadian, jalan_ruas, jalan_simpul, fasilitas (Bab 1 dan 3), sungai (Bab 4), dan kesesuaian (Bab 6). Tabel terakhir dibuat dari poligon kelas kesesuaian:

terminal

```
ogr2ogr -f PostgreSQL PG:"dbname=kph_contoh" Kesesuaian_kelas.gpkg Kesesuaian_kelas -nln kph.kesesuaian -nlt MULTIPOLYGON -lco GEOMETRY_NAME=geom -lco FID=gid -overwrite
```

Hasil uji: lima baris (kelas 0 sampai 4). Luasnya sama dengan hasil raster: kelas 1 (S1) 67,33 ha dan jumlah semua 400,00 ha.

A2. Lembar keputusan dengan satu kueri

Skrip m2_07a_lembiar_keputusan.sql menggabungkan empat bahan memakai tabel sementara (CTE). Intinya:

sql

```
CREATE TABLE kph.lembiar_keputusan AS
WITH
  s1 AS (SELECT p.gid, COALESCE(SUM(ST_Area(ST_Intersection(p.geom, k.geom))), 0) / 10000.0 AS s1_ha
        FROM kph.petak p LEFT JOIN kph.kesesuaian k
              ON k.kelas = 1 AND ST_Intersects(p.geom, k.geom) GROUP BY p.gid),
  kej AS (SELECT p.gid, COUNT(e.gid) AS kejadian
        FROM kph.petak p LEFT JOIN kph.kejadian e ON ST_Intersects(p.geom, e.geom) GROUP BY p.gid),
  sg AS (SELECT p.gid, COALESCE(SUM(ST_Length(ST_Intersection(p.geom, r.geom))), 0) AS sungai_m
        FROM kph.petak p LEFT JOIN kph.sungai r ON ST_Intersects(p.geom, r.geom) GROUP BY p.gid)
SELECT p.gid, p.kode, p.jenis_tegakan, p.geom,
       ROUND(s1.s1_ha::numeric, 2) AS s1_ha, kej.kejadian, ROUND(sg.sungai_m::numeric, 0) AS sungai_m,
       RANK() OVER (ORDER BY s1.s1_ha DESC) AS peringkat_s1
FROM kph.petak p JOIN s1 USING (gid) JOIN kej USING (gid) JOIN sg USING (gid);
```

Berkas lengkapnya juga menambah kolom jarak_kantor_m. Jaraknya dihitung dengan pgr_dijkstraCost dari simpul terdekat Kantor KPH ke simpul terdekat titik di dalam tiap petak (ST_PointOnSurface). Hasil uji, lima teratas menurut S1:

Peringkat	Petak	Jenis	Luas (ha)	S1 (ha)	S1 (%)	Kejadian	Sungai (m)	Jarak kantor (m)
1	P-05	Jati	19,17	11,97	62,5	6	182	1.580
2	P-03	Mahoni	19,97	8,98	45,0	6	647	921
3	P-08	Jati	15,85	4,51	28,4	9	452	547
4	P-06	Akasia	13,56	4,25	31,3	4	437	1.317
5	P-17	Jati	19,70	4,21	21,4	7	965	783

Pemeriksaan: jumlah kejadian semua petak 150 (cocok dengan tabel kejadian). Jumlah S1 dari poligon 67,31 ha, sedangkan dari raster 67,33 ha. Selisih 0,02 ha timbul karena poligon dipotong tepat di batas petak, sedangkan **Zonal histogram** menghitung sel menurut pusatnya.

Kueri pertanyaan rapat: petak dengan S1 minimal 40%, kejadian paling banyak 8, dan jarak lewat jalan paling jauh 1,5 km:

sql

```
SELECT kode, jenis_tegakan, s1_persen, kejadian, jarak_kantor_m
FROM kph.lembar_keputusan
WHERE s1_persen >= 40 AND kejadian <= 8 AND jarak_kantor_m <= 1500
ORDER BY s1_persen DESC;
```

Hasil uji: hanya **P-03** (S1 45,0%, 6 kejadian, 921 m). P-05 yang S1-nya paling luas gugur karena jaraknya 1.580 m. Jawaban untuk Kepala Seksi: P-03 memenuhi semua syarat, P-05 menarik tetapi terlalu jauh dari kantor.

Tabel, bukan view terwujud: Pada PostgreSQL 17 yang dipakai penulis, membuat atau menyegarkan **view terwujud** berjalan dengan `search_path` terbatas. Fungsi `pgRouting pgr_dijkstraCost` memanggil fungsi lain di skema `public`, dan hasilnya galat `function _pgr_get_statement(text) does not exist`. Penulis memakai `CREATE TABLE AS`. Bila Anda perlu view terwujud, hitung rute di tabel terpisah lebih dulu. [terbukti pada PostgreSQL 17.10; versi lain belum dicoba]

Tampilkan tabel ini di QGIS (layer dari PostGIS, kolom `geom`) dan beri simbologi menurut `s1_persen`. Itulah peta rapat Anda.

A3. Hak akses: siapa boleh apa

Skrip `m2_07b_tata_kelola.sql` membuat dua akun contoh yang masing-masing masuk ke satu peran:

sql

```
CREATE ROLE analis_baca LOGIN IN ROLE pembaca;
CREATE ROLE analis_edit LOGIN IN ROLE penyunting;
GRANT SELECT ON ALL TABLES IN SCHEMA kph TO pembaca, penyunting;
GRANT INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA kph TO penyunting;
```

Kata sandi **tidak** ditulis di skrip. Pada server nyata, atur dengan `\password analis_baca di psql`.

Hasil uji (`m2_07c_uji_hak.sql`, memakai `SET ROLE`):

Akun	Percobaan	Hasil
analisis_baca	membaca jumlah petak	25
analisis_baca	mengubah jenis tegakan P-01	ditolak: permission denied for table petak
analisis_edit	mengubah tahun tanam P-01	berhasil (1 baris)

Tabel lama perlu diberi hak lagi: `ALTER DEFAULT PRIVILEGES` hanya berlaku untuk tabel yang dibuat **sesudahnya**. Untuk tabel yang sudah ada, jalankan `GRANT ... ON ALL TABLES` seperti di atas.

A4. Jejak perubahan

Esri punya penyuntingan berversi dan arsip. PostGIS tidak punya fitur bawaan yang setara. Versi sederhana dibuat dengan trigger yang menyalin baris lama ke tabel riwayat:

sql

```
CREATE TABLE kph.riwayat_petak (id bigserial PRIMARY KEY, waktu timestampz NOT NULL DEFAULT now(),
    pengguna text NOT NULL DEFAULT current_user, aksi text NOT NULL, gid integer, baris_lama jsonb);

CREATE FUNCTION kph.catat_riwayat_petak() RETURNS trigger LANGUAGE plpgsql AS $$
BEGIN
    INSERT INTO kph.riwayat_petak (aksi, gid, baris_lama) VALUES (TG_OP, OLD.gid, to_jsonb(OLD) - 'geom');
    IF TG_OP = 'DELETE' THEN RETURN OLD; END IF;
    RETURN NEW;
END $$;

CREATE TRIGGER petak_riwayat BEFORE UPDATE OR DELETE ON kph.petak
FOR EACH ROW EXECUTE FUNCTION kph.catat_riwayat_petak();
```

Hasil uji: setelah `analisis_edit` mengubah tahun tanam P-01, tabel riwayat mencatat satu baris: aksi UPDATE, pengguna `analisis_edit`, kode P-01, tahun lama 2014. Nilai awal lalu dikembalikan dari catatan itu.

Batasannya jelas: ini hanya catatan nilai lama (tanpa bentuk geometri, karena - 'geom' membuangnya agar catatan ringan). Ini **bukan** penyuntingan berversi dengan cabang dan penggabungan. Bila Anda butuh versi penuh, itu salah satu alasan memilih geodatabase Esri.

A5. Cadangan dan pemulihan

Skrip `m2_07d_cadangan.sh` mengerjakannya:

terminal

```
pg_dump -Fc -n kph -f kph_contoh.dump kph_contoh
createdb kph_pulih
psql -d kph_pulih -c "CREATE EXTENSION postgis" -c "CREATE EXTENSION pgrouting"
pg_restore -d kph_pulih kph_contoh.dump
```

-Fc memilih format kustom yang terkompres, dan -n kph hanya mengambil skema kph. Ekstensi dibuat dulu di basis data tujuan.

Hasil uji: berkas cadangan 126.116 byte. Jumlah baris asal dan hasil pulih **sama** untuk enam tabel yang diperiksa (petak 25, kejadian 150, jalan 37, sungai 30, kesesuaian 5, lembar keputusan 25). Jumlah indeks juga sama (31) dan total luas petak sama (4.000.000 m²).

Satu aturan emas: **cadangan yang belum pernah dicoba dipulihkan belum bisa disebut cadangan**. Jadwalkan latihan pemulihan secara berkala.

A6. Perawatan ringan

- `VACUUM ANALYZE`; membersihkan sisa baris lama dan memperbarui statistik agar server memilih rencana kueri yang baik.
- Periksa ukuran: `SELECT pg_size_pretty(pg_database_size('kph_contoh'));`
- Setelah memasukkan data baru dalam jumlah besar, jalankan `ANALYZE` pada tabel itu. [CEK]

BAGIAN B | ArcGIS Pro

Bagian B: ArcGIS Pro

Pada geodatabase perusahaan, tata kelola dipegang oleh fitur geodatabase dan alat administrasi:

1. **Hak akses:** klik kanan feature class di panel **Catalog**, lalu atur **Privileges**, atau pakai alat **Change Privileges**. [CEK]
2. **Versi: Register As Versioned** pada dataset, lalu penyunting bekerja di versi sendiri dan menggabungkannya dengan **Reconcile Versions**. Ini fitur yang tidak punya padanan bawaan di PostGIS. [CEK: nama alat dan jenis versi]
3. **Arsip: Enable Archiving** mencatat riwayat perubahan. [CEK]
4. **Cadangan:** dilakukan pada tingkat basis data (misalnya `pg_dump` untuk PostgreSQL), bukan oleh Pro. [CEK]
5. **Kueri gabungan** seperti lembar keputusan dapat dijalankan di basis data lalu dibaca lewat **Query Layer**. [CEK]

BAGIAN C | ArcMap 10.8

Bagian C: ArcMap 10.8

1. Hak akses lewat **ArcCatalog** ► klik kanan dataset ► **Privileges**. [CEK]
2. Versi lewat **Register as Versioned** dan toolbar **Versioning**. [CEK]
3. Arsip lewat **Enable Archiving**. [CEK]
4. Cadangan dilakukan pada tingkat basis data, di luar ArcMap. [CEK]

Ceklis kesiapan menuju M3 dan M4

Beri tanda bila Anda sudah bisa:

- Memuat data ke PostGIS dengan SRID benar dan indeks spasial.
- Menulis kueri dengan `ST_Intersects`, `ST_DWithin`, dan `<->`.

- Menyiapkan jaringan dan menghitung rute atau area layanan.
- Menurunkan sungai dan DAS dari DEM, lalu memeriksanya dengan hitungan lain.
- Membandingkan metode interpolasi dengan titik uji.
- Menyusun kesesuaian lahan dengan skor, bobot, kendala, dan uji kepekaan.
- Memberi hak akses, mencatat perubahan, dan mencoba memulihkan cadangan.

Cek paham

1. Mengapa `analisis_baca` bisa membaca tetapi tidak bisa mengubah?
2. Jumlah S1 dari poligon (67,31 ha) sedikit berbeda dari raster (67,33 ha). Mengapa itu tidak apa-apa?
3. Apa kelemahan jejak perubahan buatan sendiri dibanding penyuntingan berversi?

Jawaban:

1. Perannya hanya diberi hak `SELECT`. Percobaan mengubah ditolak oleh server.
2. Dua cara menghitung memakai aturan berbeda untuk sel di batas petak. Selisih 0,02 ha dari 67 ha adalah wajar, dan jumlah kedua cara sama-sama sesuai luas wilayah.
3. Ia hanya mencatat nilai lama. Tidak ada cabang, tidak ada penggabungan, dan tidak melacak bentuk geometri dalam contoh ini.

Kesalahan umum

- **Memberi semua orang peran penyunting.** Beri hak baca saja kecuali yang memang perlu mengubah.
- **Menyimpan kata sandi di skrip atau repositori.** Pakai `pgpass` atau variabel lingkungan.
- **Cadangan yang tidak pernah diuji pulih.** Latihlah pemulihan.
- **Lupa membuat ekstensi di basis data tujuan sebelum `pg_restore`.** Tabel bergeometri kemungkinan gagal dibuat. [kemungkinan: tidak dicoba]

Ringkasan dan latihan

Ringkasan: satu kueri dapat menyatukan hasil banyak analisis menjadi tabel keputusan. Gudang yang baik punya hak akses, jejak perubahan, dan cadangan yang teruji.

Latihan: buat peran `tamu` yang hanya boleh membaca tabel `lembar_keputusan` (bukan tabel lain). Uji dengan `SET ROLE tamu`.

Tabel perbandingan: tata kelola

Hal	PostGIS	Geodatabase perusahaan Esri
Hak akses	peran dan <code>GRANT</code>	Privileges pada dataset [CEK]
Jejak perubahan	trigger buatan sendiri	Enable Archiving [CEK]
Penyuntingan berversi	tidak bawaan	Register As Versioned [CEK]
Cadangan	<code>pg_dump</code> dan <code>pg_restore</code>	di tingkat basis data [CEK]
Kueri gabungan	SQL langsung	Query Layer [CEK]

Lanjut belajar

Seri berikutnya adalah **M3: Deret Waktu dan Pemodelan**. Pindai kode QR ini untuk membuka halaman seri tersebut dan unduhannya.



Glosarium

Glosarium ini memuat istilah teknis yang dipakai di buku Seri M2. Istilah diurutkan menurut abjad, dan angka di ujung tiap entri menunjuk bab tempat istilah itu pertama kali dijelaskan.

AHP (analytic hierarchy process): cara menurunkan bobot kriteria dari perbandingan berpasangan antarkriteria. Hasilnya diperiksa dengan rasio konsistensi. *Bab 6.*

Akumulasi aliran (flow accumulation): jumlah sel hulu yang mengalir lewat sebuah sel. Sel dengan angka besar membentuk alur sungai. *Bab 4.*

Ambang sungai (stream threshold): batas akumulasi yang menentukan sel mana disebut sungai. Pada piksel 10 m, ambang 500 sel berarti alur yang menampung minimal 5 ha. *Bab 4.*

Analisis jaringan (network analysis): perhitungan perjalanan menyusuri jalan, bukan melintas bebas. Hasilnya berupa rute dan area layanan. *Bab 3.*

Analisis kepekaan (sensitivity analysis): mencoba bobot lain untuk melihat apakah kesimpulan berubah. *Bab 6.*

Arah aliran (flow direction): ke tetangga mana tiap sel mengalirkan air. Metode D8 memakai delapan arah. *Bab 4.*

Area layanan (service area): bagian jaringan yang bisa dicapai dari satu titik dengan biaya tertentu, misalnya jarak 500 m. *Bab 3.*

Biaya (cost): ongkos melewati satu ruas jaringan, bisa berupa panjang (meter) atau waktu (detik). *Bab 3.*

Bobot (weight): tingkat kepentingan satu kriteria dibanding kriteria lain; jumlah semua bobot adalah 1. *Bab 6.*

Buffer: sabuk dengan lebar tertentu di sekeliling objek. Di Bab 2 dipakai untuk mencari bagian petak yang berada dalam 50 m dari jalan. *Bab 2.*

Cadangan (backup): salinan data untuk memulihkan gudang bila terjadi kerusakan. Cadangan yang belum pernah dicoba dipulihkan belum bisa disebut cadangan. *Bab 7.*

CHECK: pembatas di basis data yang menolak nilai yang tidak wajar, misalnya tahun tanam 3022. *Bab 1.*

CTE (common table expression): tabel sementara di dalam satu kueri, dipakai untuk menggabungkan beberapa bahan menjadi satu hasil. *Bab 7.*

D8: metode arah aliran dengan delapan arah; tiap sel mengalir ke satu tetangga yang paling curam menurun. *Bab 4.*

DAS (daerah aliran sungai; watershed): seluruh area yang mengalir ke satu titik keluar. *Bab 4.*

DB Manager: jendela di QGIS untuk melihat tabel, mengetik SQL, dan memuat hasilnya sebagai layer. *Bab 1.*

DBSCAN: cara mengelompokkan titik yang berdekatan, dijalankan dengan fungsi `ST_ClusterDBSCAN`. Titik yang terpencil tidak masuk kelompok mana pun. *Bab 2.*

DEM (digital elevation model): model tinggi permukaan berbentuk raster. Di buku ini berpixel 10 m. *Pendahuluan.*

Dijkstra: algoritma yang mencari rute dengan biaya total paling kecil, dijalankan dengan fungsi `pgr_dijkstra`. *Bab 3.*

Dissolve (peleburan): menggabung poligon yang nilai atributnya sama menjadi satu bidang. Di SQL dikerjakan dengan `ST_Union`. *Bab 2.*

Domain: aturan nilai di geodatabase Esri. Padanannya di PostGIS adalah pembatas seperti `CHECK` dan `FOREIGN KEY`. *Bab 1.*

Ekstensi (extension): tambahan pada PostgreSQL, misalnya PostGIS dan pgRouting. Ekstensi harus dibuat dulu di basis data tujuan. *Bab 1.*

EXPLAIN: perintah untuk melihat cara server menjalankan sebuah kueri. *Bab 2.*

Feature dataset: wadah di dalam geodatabase tempat feature class jalan disalin sebelum dibuat menjadi network dataset di ArcGIS Pro. *Bab 3.*

Foreign key (kunci asing): pembatas yang memaksa nilai sebuah kolom ada dalam tabel acuan, misalnya jenis tegakan harus ada di daftar. *Bab 1.*

Geodatabase perusahaan (enterprise geodatabase): gagasan Esri untuk basis data spasial di server, dengan fitur tambahan seperti domain, subtype, dan penyuntingan berversi. *Bab 1.*

GRASS: penyedia alat di QGIS yang dipakai untuk hidrologi, misalnya `r.watershed`. Penyedia ini harus aktif di instalasi Anda. *Bab 4.*

Hak akses: aturan siapa boleh membaca dan siapa boleh mengubah data. *Bab 7.*

Hidrologi DEM: cara menebak aliran air dari tinggi tanah, dengan aturan bahwa air mengalir ke tetangga yang paling curam menurun. *Bab 4.*

IDW (inverse distance weighting): interpolasi dengan bobot berbanding terbalik dengan jarak, dipangkatkan p . Makin besar p , makin dekat titik makin dominan. *Bab 5.*

Indeks spasial (spatial index): daftar isi yang membuat pencarian berdasarkan lokasi cepat. *Bab 1.*

Interpolasi (interpolation): menebak nilai di tempat yang tidak diukur dari nilai di tempat yang diukur. *Bab 5.*

Jaringan (graf; network): kumpulan simpul dan ruas. Syarat utamanya, ruas harus bersambung di simpul. *Bab 3.*

Jejak perubahan (audit): catatan siapa mengubah apa dan kapan. *Bab 7.*

Kendala (constraint, pada kesesuaian lahan): syarat mutlak yang mencoret lokasi apa pun skornya, misalnya jarak 30 m atau kurang dari sungai. *Bab 6.*

Kesesuaian lahan multi-kriteria: penilaian lahan dengan memberi skor tiap kriteria, mengalikannya dengan bobot, menjumlahkannya, lalu mencoret lokasi yang terkena kendala. *Bab 6.*

Komponen terhubung (connected component): bagian jaringan yang semua simpulnya saling terhubung. Bila ada dua komponen atau lebih, ada jalan yang terputus. *Bab 3.*

KPH Contoh: kantor pengelola hutan fiktif, tempat semua studi kasus di buku ini terjadi. *Pendahuluan.*

Kriging: tebakan berbobot yang bobotnya dihitung dari pola kemiripan antartitik (semivariogram). *Bab 5.*

Kunci utama (primary key): kolom yang menandai tiap baris tabel secara unik. Di contoh bab ini bernama `gid`. *Bab 1.*

LATERAL: kata kunci SQL yang membuat subkueri dijalankan sekali untuk tiap baris, misalnya untuk mencari pos terdekat bagi tiap titik panas. *Bab 2.*

LEFT JOIN: penggabungan tabel yang mempertahankan semua baris tabel kiri, sehingga petak tanpa kejadian tetap muncul dengan hitungan nol. *Bab 2.*

Lubang (sink): sel yang lebih rendah dari semua tetangganya, sehingga air terjebak. Lubang diisi sebelum aliran dihitung. *Bab 4.*

Network Analyst: ekstensi ArcGIS Pro dan ArcMap untuk analisis jaringan. Ekstensi ini memerlukan network dataset. *Bab 3.*

Network dataset (kumpulan data jaringan): bentuk jaringan yang dibaca ArcGIS Pro dan ArcMap, dibuat dari feature class jalan. *Bab 3.*

NoData: sel raster yang dianggap tidak bernilai. Bila satu masukan kalkulator GDAL berisi NoData di sebuah sel, hasil di sel itu ikut NoData. *Bab 6.*

ogr2ogr: perintah yang membaca sebuah berkas spasial dan menuliskannya ke format lain, misalnya dari GeoPackage ke tabel PostGIS. *Bab 1.*

Overlay (tumpang susun): menumpuk dua lapisan untuk mengetahui irisannya, misalnya petak dengan tanah. Di SQL dikerjakan dengan `ST_Intersection`. *Bab 2.*

Pengukur: fungsi spasial yang menjawab dengan angka, misalnya `ST_Area` dan `ST_Distance`. *Bab 2.*

Penyuntingan berversi (versioning): fitur geodatabase Esri tempat tiap penyunting bekerja di versi sendiri lalu menggabungkannya. PostGIS tidak punya fitur bawaan yang setara. *Bab 1, 7.*

Peran (role): kelompok hak akses di basis data, misalnya "boleh baca saja". *Bab 1.*

pg_dump dan **pg_restore:** perintah untuk membuat cadangan dan memulihkannya. Berkas cadangan memakai format kustom yang terkompres. *Bab 7.*

pgRouting: ekstensi PostgreSQL untuk analisis jaringan. *Bab 3.*

Piksel (pixel): kotak terkecil dalam raster; tiap kotak menyimpan satu angka. *Bab 4.*

PostGIS: tambahan untuk PostgreSQL yang memberi kemampuan menyimpan titik, garis, dan poligon. *Bab 1.*

Predikat spasial: fungsi yang menjawab ya atau tidak, misalnya `ST_Intersects` dan `ST_DWithin`. *Bab 2.*

Proyek terpadu: penggabungan hasil banyak analisis dalam satu kueri, misalnya lembar keputusan per petak. *Bab 7.*

Query Layer: cara ArcGIS Pro dan ArcMap membaca hasil kueri SQL dari basis data sebagai layer. *Bab 2.*

Regresi (regression): menyesuaikan garis lurus antara dua hal, misalnya hujan terhadap elevasi. *Bab 5.*

- Reklasifikasi** (reclassify): mengubah nilai raster menjadi kelas atau skor menurut tabel rentang. *Bab 6.*
- Reproyeksi** (reprojection): mengubah koordinat data ke sistem koordinat lain sebelum dimasukkan atau digabung dengan data lain. *Bab 1.*
- Residu** (residual): sisa selisih antara nilai pengamatan dan garis regresi. Sisa ini diinterpolasi dengan IDW lalu dijumlahkan kembali. *Bab 5.*
- RMSE** (root mean square error): ukuran rata-rata besar selisih tebakan dan kenyataan. Makin kecil makin baik. *Bab 5.*
- Ruas** (edge): potongan jalan di antara dua simpul. *Bab 3.*
- Semivariogram**: grafik kemiripan antartitik menurut jarak, dipakai untuk menentukan bobot kriging. *Bab 5.*
- Server**: petugas penjaga gudang data yang melayani banyak pengguna sekaligus. *Bab 1.*
- Simpul** (node): persimpangan atau ujung jalan. *Bab 3.*
- Skema** (schema): "rak" di dalam basis data tempat tabel dikelompokkan. *Bab 1.*
- Skor** (score): nilai 1 sampai 4 untuk tiap kelas sebuah kriteria; 4 berarti terbaik. *Bab 6.*
- Snap Pour Point**: alat ArcGIS Pro dan ArcMap yang menempelkan titik keluar ke sel akumulasi tertinggi dalam jarak tertentu. *Bab 4.*
- Spatial Analyst**: ekstensi ArcGIS Pro dan ArcMap yang memuat alat raster seperti hidrologi, interpolasi, dan Weighted Overlay. *Bab 4.*
- SQL spasial**: SQL biasa ditambah fungsi yang mengerti letak dan bentuk; namanya berawalan `ST_`. *Bab 2.*
- SRID**: nomor sistem koordinat yang menempel pada tiap geometri. Untuk UTM 49S nomornya 32749. *Bab 1.*
- Tetangga terdekat** (KNN): mengurutkan hasil menurut kedekatan dengan operator `<->`. *Bab 2.*
- TIN** (triangulated irregular network): jaring segitiga yang menghubungkan titik. Nilai di dalam segitiga dihitung lurus dari ketiga sudutnya. *Bab 5.*
- Titik keluar** (outlet): titik tempat seluruh air dari sebuah DAS keluar. *Bab 4.*
- Titik latih dan titik uji**: sebagian titik dipakai untuk menebak, sebagian disimpan untuk menilai tebakan. *Bab 5.*
- Toleransi topologi** (topology tolerance): batas jarak yang membuat dua ujung ruas dianggap satu simpul. Nilai 5 berarti ujung yang berjarak hingga 5 satuan peta dianggap bersambung. *Bab 3.*
- Trigger**: aturan yang berjalan otomatis saat data diubah, dipakai di sini untuk mencatat riwayat perubahan. *Bab 7.*
- UTM** (Universal Transverse Mercator): sistem koordinat berukuran meter yang dipakai data latihan (zona 49S, EPSG:32749). *Pendahuluan.*
- VACUUM ANALYZE**: perintah perawatan yang membersihkan sisa baris lama dan memperbarui statistik agar server memilih rencana kueri yang baik. *Bab 7.*

Validasi silang (cross-validation): menilai metode interpolasi dengan meninggalkan satu titik, menebaknya dari titik lain, lalu membandingkan. *Bab 5*.

View: kueri yang disimpan dan bisa dibuka seperti tabel; isinya dihitung ulang tiap dibuka. *Bab 2*.

View terwujud (materialized view): view yang menyimpan hasilnya. Cepat dibuka, tetapi harus disegarkan dengan `REFRESH`. *Bab 2*.

Weighted Overlay: alat Spatial Analyst yang menjumlahkan raster bertipe bilangan bulat dengan bobot persen yang berjumlah 100. *Bab 6*.

Zonal histogram: alat QGIS yang menghitung luas tiap kelas raster di dalam tiap poligon, misalnya per petak. *Bab 6*.

Daftar Pustaka

Daftar ini memuat sumber yang dipakai menyusun buku ini. Semua tautan dibuka dan dicek pada 7 Oktober 2026. Halaman dokumentasi yang terus diperbarui ditulis tanpa tahun (t.t.), dan nomor versi yang tertera di halaman dicatat pada judulnya.

Dokumentasi resmi

- Esri. (t.t.). *An overview of the Hydrology toolset*. ArcGIS Pro documentation. <https://doc.esri.com/en/arcgis-pro/latest/tool-reference/spatial-analyst/an-overview-of-the-hydrology-tools.html>.
- Esri. (t.t.). *Create Enterprise Geodatabase (Data Management Tools)*. ArcGIS Pro documentation. <https://doc.esri.com/en/arcgis-pro/latest/tool-reference/data-management/create-enterprise-geodatabase.html>.
- Esri. (t.t.). *IDW (Spatial Analyst Tools)*. ArcGIS Pro documentation. <https://doc.esri.com/en/arcgis-pro/latest/tool-reference/spatial-analyst/idw.html>.
- Esri. (t.t.). *Make Service Area Analysis Layer (Network Analyst Tools)*. ArcGIS Pro documentation. <https://doc.esri.com/en/arcgis-pro/latest/tool-reference/network-analyst/make-service-area-analysis-layer.html>.
- Esri. (t.t.). *Weighted Overlay (Spatial Analyst Tools)*. ArcGIS Pro documentation. <https://doc.esri.com/en/arcgis-pro/latest/tool-reference/spatial-analyst/weighted-overlay.html>.
- GRASS Development Team. (t.t.). *r.stream.extract* (GRASS 8.5). GRASS manual. <https://grass.osgeo.org/grass-stable/manuals/r.stream.extract.html>.
- GRASS Development Team. (t.t.). *r.water.outlet* (GRASS 8.5). GRASS manual. <https://grass.osgeo.org/grass-stable/manuals/r.water.outlet.html>.
- GRASS Development Team. (t.t.). *r.watershed* (GRASS 8.5). GRASS manual. <https://grass.osgeo.org/grass-stable/manuals/r.watershed.html>.
- pgRouting Project. (t.t.). *pgr_dijkstra* (pgRouting Manual 4.0). https://docs.pgrouting.org/latest/en/pgr_dijkstra.html.
- PostgreSQL Global Development Group. (t.t.). *CREATE TRIGGER*. PostgreSQL 17 Documentation. <https://www.postgresql.org/docs/17/sql-createtrigger.html>.
- PostgreSQL Global Development Group. (t.t.). *GRANT*. PostgreSQL 17 Documentation. <https://www.postgresql.org/docs/17/sql-grant.html>.
- PostgreSQL Global Development Group. (t.t.). *Materialized Views*. PostgreSQL 17 Documentation. <https://www.postgresql.org/docs/17/rules-materializedviews.html>.
- PostgreSQL Global Development Group. (t.t.). *pg_dump*. PostgreSQL 17 Documentation. <https://www.postgresql.org/docs/17/app-pgdump.html>.
- PostgreSQL Global Development Group. (t.t.). *pg_restore*. PostgreSQL 17 Documentation. <https://www.postgresql.org/docs/17/app-pgrestore.html>.
- QGIS Project. (t.t.). *Interpolation*. QGIS Documentation (bagian Processing algorithms; halaman menunjukkan versi 3.44). https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/interpolation.html.

- QGIS Project. (t.t.). *Network analysis*. QGIS Documentation (bagian Processing algorithms; halaman menunjukkan versi 3.44). https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/networkanalysis.html.
- QGIS Project. (t.t.). *Opening Data*. QGIS Documentation (halaman menunjukkan versi 3.44). https://docs.qgis.org/latest/en/docs/user_manual/managing_data_source/opening_data.html.

Makalah dan buku

- Saaty, T. L. (1977). *A scaling method for priorities in hierarchical structures*. Journal of Mathematical Psychology, 15(3), 234-281. [https://doi.org/10.1016/0022-2496\(77\)90033-5](https://doi.org/10.1016/0022-2496(77)90033-5).
- Wang, L., & Liu, H. (2006). *An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling*. International Journal of Geographical Information Science, 20(2), 193-213. <https://doi.org/10.1080/13658810500433453>.

Aplikasi Live Penulis

Berikut aplikasi web milik penulis yang bisa Anda buka langsung. Daftar ini dicek pada 7 Oktober 2026. Sumbernya bagian Aplikasi Live di <https://badarmubarok.id/#aplikasi-live>.

No.	Aplikasi	Tautan
1	SIM RHL — Sistem Informasi Manajemen Rehabilitasi Hutan dan Lahan (Fakultas Kehutanan UGM), Versi PC	https://simrhl.badarmubarok.id
2	SIM RHL, Versi HP	https://m.simrhl.badarmubarok.id
3	ScienceThink — RAD Tool (Rapid Application Development)	https://sciencethink.badarmubarok.id
4	Peta Landsystem Indonesia — peta landsystem dan peta tematik lainnya	https://landsystem.badarmubarok.id
5	Analisis Kebutuhan Unsur Hara — aplikasi Web GIS analisis unsur hara	https://unsurhara.badarmubarok.id
6	Analisa Karbon — Web GIS estimasi cadangan karbon dari citra satelit	https://karbon.badarmubarok.id
7	Aplikasi Deforestasi — deteksi perubahan tutupan hutan (2001–2025)	https://deforestasi.badarmubarok.id
8	Deteksi Area Terbakar — pemetaan tingkat keparahan area terbakar (NBR/dNBR)	https://kebakaran.badarmubarok.id
9	Mata Elang — platform deteksi konservasi berbasis AI	https://mataelang.badarmubarok.id
10	SIM Konflik — Sistem Informasi Manajemen Konflik	https://simkonflik.badarmubarok.id
11	Seri Buku GIS Kehutanan dan Pertanian — 12 buku praktik GIS (dalam bahasa Indonesia)	https://badarmubarok.id/buku-gis/

MASTER CLASS · SERI M2

BASIS DATA SPASIAL DAN ANALISIS LANJUT

Dari berkas ke gudang data: SQL spasial, jaringan, hidrologi DEM, interpolasi, dan kesesuaian lahan (kehutanan dan pertanian).

Yang akan Anda pelajari

- 1** Satu Gudang untuk Semua Peta: PostGIS dan Geodatabase Perusahaan
- 2** Bertanya pada Peta: SQL Spasial
- 3** Jalan Tercepat dan Jangkauan Layanan: Analisis Jaringan
- 4** Ke Mana Air Mengalir: Hidrologi dari DEM
- 5** Menebak di Antara Titik: Interpolasi
- 6** Mencari Lahan Terbaik: Kesesuaian Lahan Multi-Kriteria
- 7** Menyatukan dan Menjaga: Proyek Terpadu dan Tata Kelola



**Pindai untuk membuka
seluruh seri buku GIS**

Buku PDF, paket data, dan skrip
<https://badarmubarok.id/buku-gis/>