

WebGIS Praktis

Perbandingan GIS, Ragam Maps API, dan Akses Publikasi Web

Program Sarjana Terapan Teknologi Survei dan Pemetaan Dasar
Departemen Teknologi Kebumihan
Sekolah Vokasi, UGM

Ismail Sunni | Geospatial Software Engineer | Camptocamp DE

Presentasi Ini

<https://github.com/ismailsunni/web-gis-2026>



Recap: Apa yang Sudah Kalian Tahu

Minggu 4:

- JavaScript: variabel, fungsi, objek, OOP
- DOM manipulation & event listener
- Mini WebGIS dari nol (tanpa library)

Minggu 5:

- Pengenalan Leaflet & OpenLayers sebagai library
- Client-side scripting dengan library GIS

Hari ini: Naik level — dari *mengenal library* ke *membangun sistem*

Tujuan Pembelajaran

Setelah sesi ini, mahasiswa mampu:

- ✓ Membandingkan **WebGIS vs Desktop GIS** secara mendalam
- ✓ Memilih **Maps API yang tepat** (Leaflet, Google, OpenLayers)
- ✓ Memahami strategi **publikasi & akses** peta interaktif di web
- ✓ Mengintegrasikan **data eksternal** (Google Sheets, WMS/WFS)
- ✓ Men-deploy WebGIS ke **GitHub Pages** atau **Vercel**

Outline

1. **WebGIS vs Desktop GIS** — Perbandingan mendalam
2. **Arsitektur WebGIS** — Stack & komponen
3. **Ragam Maps API** — Leaflet, Google Maps, OpenLayers
4. **Perbandingan Detail** — Teknis & decision matrix
5. **Publikasi & Akses** — Platform, strategi, best practices
6. **Library Lain** — Opsi tambahan di luar tiga utama
7. **Studi Kasus** — WebGIS di dunia nyata
8. **Refleksi & Tugas** — Takeaways & assignment

1 WebGIS vs Desktop GIS

Lebih dari Sekadar "Bisa Online"

Desktop GIS: Kekuatan Utama

QGIS / ArcGIS Desktop / GRASS

- Analisis spasial lengkap — buffer, overlay, network analysis
- Editing geometri presisi tinggi
- Cartographic production — peta cetak berkualitas
- Processing pipeline — batch geoprocessing
- Data format beragam — Shapefile, GeoPackage, GDB, raster

Desktop GIS = laboratorium analisis spasial

Web GIS: Kekuatan Utama

Google Maps / Mapbox / Custom WebGIS

- **Zero install** — akses via URL, tidak perlu instalasi software
- **Multi-platform** — desktop, tablet, smartphone (responsive)
- **Kolaborasi real-time** — banyak user akses bersamaan tanpa konflik
- **Integrasi seamless** — embed di website, dashboard, CMS, aplikasi
- **Update sentral** — data berubah satu kali, semua user dapat update
- **Aksesibilitas** — publik bisa akses tanpa login atau izin khusus
- **Mobile-first** — optimized untuk penggunaan di lapangan via smartphone
- **Lightweight** — tidak butuh hardware canggih untuk client

Web GIS = media komunikasi, distribusi, dan publikasi spasial

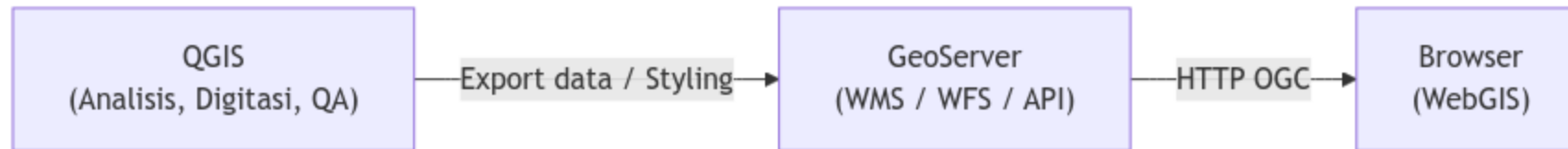
Perbandingan Mendalam - Part 1

Aspek	Desktop GIS	Web GIS
Analisis	Lengkap (500+ tools)	Terbatas visualisasi & query
Data size	GB–TB lokal	MB–ratusan MB streaming
CRS	Semua CRS	Umumnya Web Mercator
Rendering	GPU lokal	Browser (Canvas/WebGL)
Offline	✅ Penuh	⚠️ Perlu caching
Kolaborasi	❌ File-based	✅ Real-time

Perbandingan Mendalam - Part 2

Aspek	Desktop GIS	Web GIS
Distribusi	Email/FTP	Share URL instant
Maintenance	Update per mesin	1× di server
Learning curve	Curam (~40 jam)	Ringan (~4 jam)
Biaya	Dari Gratis hingga Mahal	Dari Gratis hingga Mahal
User target	GIS analyst	Publik/stakeholder

Hybrid Approach: Best of Both Worlds



Workflow modern: Analisis di desktop → Publikasi di web

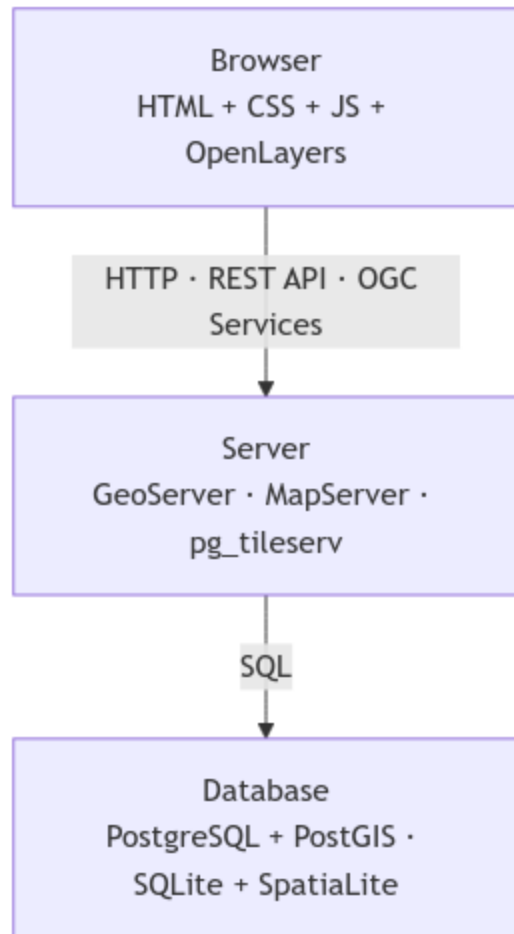
Kapan Pakai Yang Mana?

Pilihan	Pakai Saat	Contoh
 Desktop GIS	Analisis kompleks, editing presisi	Buffer, network, peta cetak
 Web GIS	Publikasi, dashboard, kolaborasi	Portal publik, monitoring
 Hybrid	Analisis + distribusi online	QGIS + Lizmap, atau QGIS -> GeoServer -> WebGIS

2 **Arsitektur WebGIS**

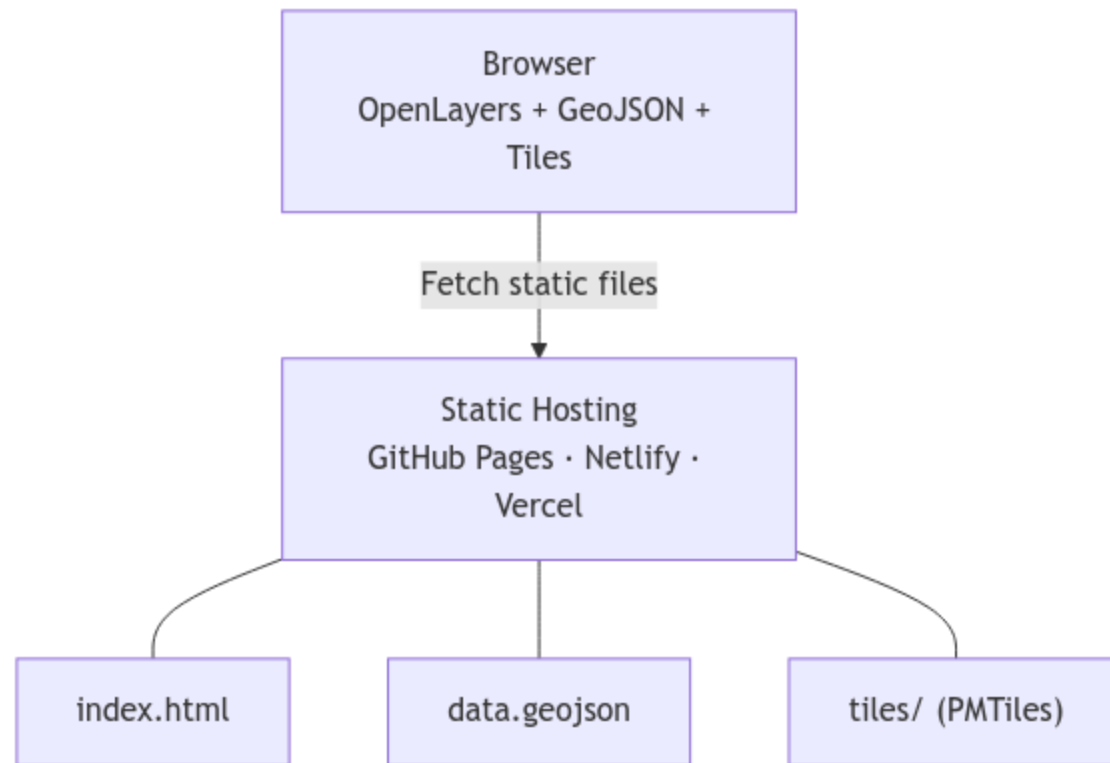
Dari Database ke Browser

Full Stack WebGIS



Alternatif: Serverless WebGIS

Tidak semua WebGIS butuh backend!



Contoh stack: GitHub Pages/Vercel + Supabase (Auth + PostgreSQL) + GeoJSON/tiles

Format Data di WebGIS

Format	Tipe	Cocok untuk
GeoJSON	Vektor	Fitur kecil-sedang (<10 MB)
TopoJSON	Vektor (compressed)	Fitur besar, batas wilayah
XYZ Tiles	Raster	Base map (OSM, satellite)
PMTiles	Vektor tiles	Offline / static hosting
WMS	Raster	Server-rendered map image
WFS	Vektor	Server feature access
COG	Raster (cloud)	Citra satelit besar

3 Ragam Maps API

Perbandingan & Pemilihan

Maps API #1: Leaflet.js

"Ringan & Cepat"

Kelebihan utama:

- Kecil, cepat dipelajari, banyak plugin
- Cocok untuk peta interaktif ringan

Cocok untuk:

- Website peta sederhana
- Dashboard monitoring
- Prototype cepat

Biaya: Gratis (open source)

Maps API #2: Google Maps API

"Siap Pakai & Powerful"

Kelebihan utama:

- Basemap premium dan data POI kuat
- Fitur siap pakai: Directions, Places, Street View

Cocok untuk:

- Aplikasi komersial
- Produk consumer (non-GIS user)
- Kebutuhan routing/geocoding cepat

Biaya: Berbayar (metered billing)

Maps API #3: OpenLayers

"Professional & Standard"

Kelebihan utama:

- Dukungan OGC native (WMS/WFS)
- Multi-CRS dan kontrol layer detail

Cocok untuk:

- Geoportals profesional
- Integrasi GeoServer/QGIS Server
- Dashboard spasial skala menengah-besar

Biaya: Gratis (open source)

Framework Keputusan

Kondisi Kebutuhan	API Disarankan
Belajar cepat, proyek sederhana	Leaflet
Perlu fitur Google siap pakai	Google Maps API
Perlu WMS/WFS, CRS kompleks, OGC	OpenLayers

Rule of thumb: simple -> Leaflet, consumer app -> Google, GIS profesional -> OpenLayers.

Leaflet vs Google Maps vs OpenLayers - Part 1

Kriteria	Leaflet	Google Maps	OpenLayers
Setup	Import CDN	API key	Import CDN
Learning curve	Mudah (2-4 jam)	Mudah (2-4 jam)	Menengah (6-10 jam)
Basemap	OSM/CARTO/Esri via tile URL	Built-in Google	OSM/CARTO/Esri via tile source
Customization	Baik	Terbatas	Sangat baik
WMS/WFS	Plugin/sulit	✗ Tidak	✓ Built-in

Leaflet vs Google Maps vs OpenLayers - Part 2

Kriteria	Leaflet	Google Maps	OpenLayers
Multi-CRS	Dengan plugin	Mercator saja	✅ Semua EPSG
3D/Terrain	❌	✅ Ada	Limited
Biaya	Gratis	~\$7-10 per 1K	Gratis
Offline	Bisa	❌ Tidak	Bisa

Google Maps: Kapan Pakai?

✓ Gunakan jika:

- Aplikasi **commercial/berbayar**
- Target **non-GIS user** (consumer)
- Perlu **Street View** atau **Directions API**
- Budget ada untuk API cost

✗ Hindari jika:

- Data proyeksi custom (non-Web Mercator)
- Butuh WMS/WFS dari server pemetaan lokal
- Budget terbatas, traffic tinggi
- Perlu open source untuk compliance

Contoh: Google Maps API

```
const map = new google.maps.Map(document.getElementById('map'), {
  zoom: 14,
  center: { lat: -7.7956, lng: 110.3695 }
});

const marker = new google.maps.Marker({
  position: { lat: -7.7956, lng: 110.3695 },
  map: map,
  title: 'Yogyakarta'
});
```

Keuntungan: Instant, polished, banyak plugin

Kerugian: API key required, metered billing

Leaflet vs OpenLayers: Teknis (Part 1)

Aksi	Leaflet	OpenLayers
Buat peta	<code>L.map('map')</code>	<code>new ol.Map({target:'map'})</code>
Set view	<code>.setView([-7.79, 110.36], 13)</code>	<code>view: new ol.View({center, zoom})</code>
Tambah tile	<code>L.tileLayer(url).addTo(map)</code>	<code>new ol.layer.Tile({source:...})</code>
Marker	<code>L.marker([lat, lng])</code>	<code>new ol.Feature({geometry: Point})</code>

Leaflet vs OpenLayers: Teknis (Part 2)

Aksi	Leaflet	OpenLayers
Popup	<code>.bindPopup(html)</code>	<code>new ol.Overlay({element:...})</code>
WMS	Plugin (sulit)	✓ <code>ol.source.TileWMS</code> (mudah)
CRS lain	Dengan proj4 plugin	✓ <code>ol.proj.fromLonLat()</code> native

Leaflet = ringkas & intuitif. OpenLayers = verbose tapi profesional.

4 Hands-on: Praktik WebGIS dengan Strategi Publikasi

Dari Local → Public dengan Akses Terkontrol

Pilihan Skenario Praktik

Opsi A (yang dipraktikkan di kelas)

- OpenLayers + Google Sheets
- Basemap switcher + filter kategori
- Deploy ke GitHub Pages

Opsi B (pengembangan mandiri)

- Leaflet + API + autentikasi sederhana
- Deploy ke Vercel

Fokus kelas: Opsi A dulu, Opsi B untuk eksplorasi.

Upgrade 1: WMS Layer dari Server OGC

OpenLayers bisa langsung konek ke GeoServer/QGIS Server/BIG.

```
const wmsLayer = new ol.layer.Tile({
  source: new ol.source.TileWMS({
    url: 'https://geoserver.example.com/wms',
    params: {
      'LAYERS': 'nama:layer',
      'TILED': true
    },
    serverType: 'geoserver'
  })
});
map.addLayer(wmsLayer);
```

Coba WMS publik: BIG, BMKG, atau GeoServer demo.

Toggle Visibility Layer

```
// Tombol toggle di HTML  
// <button id="toggle-wms">Toggle WMS</button>  
  
document.getElementById('toggle-wms').addEventListener('click', () => {  
    wmsLayer.setVisible(!wmsLayer.isVisible());  
});
```

Intinya: layer WMS bisa dihidupkan/dimatikan tanpa reload peta.

Upgrade 2: Data dari Google Sheets

Konsep: Google Sheet publik → CSV → parse di JavaScript → OL Features

Google Sheet (isi data) → Publish as CSV → Fetch di JS → Buat Feature OL

Setup Google Sheet:

nama	lon	lat	kategori	deskripsi
Tugu Yogya	110.3672	-7.7828	landmark	Ikon kota
UGM	110.3780	-7.7703	universitas	Kampus tertua

Fetch & Parse Google Sheets CSV

```
const SHEET_CSV_URL =
  'https://docs.google.com/spreadsheets/d/ID/export?format=csv&gid=0';

async function loadFromSheet(url) {
  const res = await fetch(url);
  const text = await res.text();
  const rows = text.split('\n').slice(1); // skip header

  rows.forEach(row => {
    const [nama, lon, lat, kategori] = row.split(',');
    if (!lon || !lat) return;
    const feature = new ol.Feature({
      geometry: new ol.geom.Point(
        ol.proj.fromLonLat([parseFloat(lon), parseFloat(lat)])
      ),
      nama, kategori
    });
    markerSource.addFeature(feature);
  });
}

loadFromSheet(SHEET_CSV_URL);
```





Upgrade 3: Filter Kategori

```
<div id="filter">
  <button data-cat="all">Semua</button>
  <button data-cat="landmark">Landmark</button>
  <button data-cat="universitas">Universitas</button>
</div>
```

```
document.querySelectorAll('#filter button').forEach(btn => {
  btn.addEventListener('click', () => {
    const cat = btn.dataset.cat;
    markerLayer.setStyle(feature => {
      if (cat === 'all' || feature.get('kategori') === cat) {
        return circleStyle(feature);
      }
      return null;
    });
  });
});
```

5 Strategi Publikasi & Akses Peta Interaktif

Platform Publikasi WebGIS

Platform	Untuk	Pro	Con
GitHub Pages 	Static demo	Gratis, instant	No backend
Vercel/Netlify 	Serverless	Auto-deploy, fast	Limited free
DigitalOcean/AWS 	Backend kompleks	Full control	Biaya, DevOps
GeoServer 	Geoportal	GIS-native	Learning curve

Fokus kelas: GitHub Pages (A) & Vercel (B bonus)

3 Skenario Publikasi

Skenario 1: Leaflet + GitHub Pages

→ Demo/portfolio, static, instant





Skenario 2: OpenLayers + Vercel

→ Dashboard real-time, auto-deploy

Skenario 3: GeoServer + PostGIS

→ Geoportals enterprise, powerful

Akses Peta: 4 Model

Level	Auth	Monitoring	Contoh
 Public	✗	-	OpenStreetMap
 Login	✓ SSO	Tracking	Corp dashboard
 API Key	✓ Key	Rate limit	Google Maps
 Enterprise	✓ Token	SLA	ArcGIS Online

Implementasi: Public WebGIS di GitHub Pages

```
# 1. Init repo lokal
git init
git add index.html app.js data/landmarks.geojson
git commit -m "WebGIS Yogyakarta - initial"

# 2. Buat repo di GitHub, lalu push
git remote add origin https://github.com/USERNAME/webgis-yogya.git
git push -u origin main

# 3. Aktifkan GitHub Pages
# Settings → Pages → Branch: main → Folder: root → Save

# 4. Tunggu ~1 menit, akses:
# https://USERNAME.github.io/webgis-yogya/
```

Implementasi: Autentikasi + Backend (Vercel)

```
// API endpoint: /api/landmarks?token=XXX
// Backend mengecek token sebelum return data

const token = localStorage.getItem('auth_token');
fetch(`/api/landmarks?token=${token}`)
  .then(r => r.json())
  .then(data => {
    data.features.forEach(f => {
      new L.Marker([f.geometry.coordinates[1], f.geometry.coordinates[0]])
        .bindPopup(f.properties.nama)
        .addTo(map);
    });
  });
```

Contoh Auth dengan Supabase

```
// CDN: https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2
const supabase = window.supabase.createClient(SUPABASE_URL, SUPABASE_ANON_KEY);

// Login sederhana email + password
const { data, error } = await supabase.auth.signInWithPassword({
  email: 'mahasiswa@kampus.ac.id',
  password: 'password123'
});

if (error) alert('Login gagal');
if (data?.session) {
  const token = data.session.access_token;
  fetch('/api/landmarks', { headers: { Authorization: `Bearer ${token}` } });
}
```

Backend memverifikasi JWT Supabase sebelum mengirim data.

Best Practices: UX & Teknis

Checklist dasar publikasi:

- Mobile-friendly
- Ada loading dan pesan error
- Gunakan HTTPS
- Cantumkan sumber data (attribution)
- README berisi link demo

Contoh Publikasi Nyata

Sheet Map (Open Layers + Google Sheets + GitHub Pages)

Data publik → clustering → 100% client-side

 ismailsunni.id/map/sheet-map

Route Finder (OpenLayers + Supabase + PostGIS + pgRouting + GitHub Pages)

Node.js + PostgreSQL → DigitalOcean → rate limiting

 ismailsunni.id/map/route-finder

Geo Admin (OGC Service + Geoportal)

JavaScript + WMS/WMTS → 800+ layers → 3D view

 map.geo.admin.ch

7 Library Lain

Opsi Tambahan untuk Kebutuhan Khusus

Beberapa Library Lain

- **MapLibre GL JS** — open-source untuk vector tiles dan tampilan WebGL. 
<https://maplibre.org/>
- **Mapbox GL JS / Maps SDK** — ekosistem commercial yang matang untuk peta modern berbasis vector tiles.  <https://www.mapbox.com/>
- **CesiumJS** — fokus pada globe 3D, terrain, dan visualisasi geospasial 3D. 
<https://cesium.com/platform/cesiumjs/>
- **kepler.gl** — cocok untuk eksplorasi data spasial besar dengan visualisasi cepat.  <https://kepler.gl/>
- **deck.gl** — kuat untuk visualisasi WebGL layer besar seperti trip, heatmap, dan point cloud.  <https://deck.gl/>

Alternatif Google Maps


Platform	Kelebihan Singkat	Link
Mapbox	Styling fleksibel, SDK matang	https://www.mapbox.com/
HERE Maps	Kuat untuk routing/logistik	https://www.here.com/
TomTom Maps	Traffic dan navigasi real-time	https://developer.tomtom.com/
MapTiler	Mudah dipakai dengan MapLibre	https://www.maptiler.com/

Catatan: tetap pilih platform berdasarkan **biaya**, **fitur**, dan **cakupan data**.

8 Studi Kasus

WebGIS di Dunia Nyata

Studi Kasus: map.geo.admin.ch

 **Geoportal nasional Swiss** — dibangun dengan OpenLayers

Fitur:

- 800+ layer data nasional
- 2D dan 3D view
- WMS/WMTS dari swisstopo
- Pencarian alamat & koordinat
- Cetak peta & share URL
- Open source!

 map.geo.admin.ch

Studi Kasus: Peta Interaktif Routing

 **Route Finder** — contoh WebGIS dengan routing nyata

Fitur:

- Pencarian lokasi (Photon geocoding)
- Routing jalan nyata via pgRouting
- TSP solver (Held-Karp)
- Multiple kota (Yogyakarta & München)

 ismailsunni.id/map/route-finder

Dibangun dengan OpenLayers + Supabase + PostgreSQL

Studi Kasus: Peta dari Google Sheet

 Sheet Map — WebGIS tanpa backend, data dari spreadsheet

Fitur:

- Data dari Google Sheet publik
- Clustering otomatis
- Warna per kategori
- Zero server — 100% client-side

 ismailsunni.id/map/sheet-map

Cocok untuk: survei lapangan, data crowdsource, tugas mahasiswa

9 Refleksi & Tugas

Refleksi

- 🤔 Kapan sebaiknya pakai Desktop vs Web GIS?
- 🤔 Kapan WMS lebih baik daripada GeoJSON langsung?
- 🤔 Apa keuntungan load data dari Google Sheets vs hardcode?
- 🤔 Kapan perlu memakai library lain selain Leaflet/OpenLayers?

Key Takeaways

1. Desktop GIS = **analisis**, Web GIS = **komunikasi & distribusi**
2. OpenLayers unggul di **OGC standards** — WMS/WFS, multi-CRS, native
3. **Google Sheets** → **CSV** = cara cepat punya backend data tanpa server
4. **Serverless WebGIS** bisa sangat powerful (GeoJSON + GitHub Pages)
5. Selain tiga library utama, ada opsi lain untuk kebutuhan **3D**, **vector tiles**, dan **visualisasi data besar**

Latihan Mandiri

Silakan pilih salah satu atau kombinasikan:

1. Bangun WebGIS dengan Leaflet/OpenLayers + data Google Sheets
2. Tambahkan 2 basemap + filter kategori
3. Deploy ke GitHub Pages
4. Coba integrasi 1 layer WMS publik
5. (Opsional) Coba deploy versi dengan autentikasi sederhana di Vercel

Keluaran latihan: link repo + link demo + screenshot.

Referensi

- [OpenLayers Quick Start](#)
- [OpenLayers Examples](#)
- [MapLibre GL JS Docs](#)
- [GitHub Pages Docs](#)
- [GeoJSON.io](#) — buat GeoJSON interaktif
- [ina-geoportal.go.id](#) — WMS BIG
- [Map Collection](#) — contoh proyek nyata
- [This presentation](#)

Eksplorasi dan bawa pertanyaan ke sesi berikutnya!