

**PENINGKATAN TINGKAT KEDETAILAN MODEL 3D  
BANGUNAN PADA PETA KOTA DIGITAL MENGGUNAKAN  
TEKNIK MONOPLOTTING BERBASIS ARSITEKTUR  
CLIENT-SERVER WEB**

**TESIS**

**Karya tulis sebagai salah satu syarat  
untuk memperoleh gelar Magister dari  
Institut Teknologi Bandung**

**Oleh  
NUNKY FEBILIA VERANY  
NIM: 23222305  
(Program Studi Magister Teknik Elektro)**



**INSTITUT TEKNOLOGI BANDUNG  
Juni 2025**

## **ABSTRAK**

# **PENINGKATAN TINGKAT KEDETAILAN MODEL 3D BANGUNAN PADA PETA KOTA DIGITAL MENGGUNAKAN TEKNIK MONOPLOTTING BERBASIS ARSITEKTUR CLIENT-SERVER WEB**

Oleh  
**Nunky Febilia Verany**  
**NIM: 2322205**  
**(Program Studi Magister Teknik Elektro)**

Model bangunan tiga dimensi (3D) pada peta kota digital umumnya tersedia dalam *Level of Detail 2* (LoD2), yang hanya menyajikan bentuk geometri dasar bangunan dengan tambahan informasi bentuk atap namun tanpa informasi detail pada fasad. Penelitian ini bertujuan untuk meningkatkan kedetailan semantik model LoD2 dengan menambahkan informasi bukaan seperti jendela dan pintu melalui teknik monoplotting berbasis citra fasad. Sistem yang dikembangkan mengintegrasikan metode fotogrametri berbasis gambar tunggal (*monoplotting*) dalam arsitektur web *client-server*, memungkinkan pengguna melakukan anotasi langsung pada citra fasad dan memproyeksikannya ke model 3D bangunan secara spasial.

Implementasi dilakukan dengan menggunakan React.js dan CesiumJS untuk visualisasi *frontend*, serta Node.js untuk *backend* dan pengelolaan data. Citra fasad diperoleh dari kamera ponsel dan dianotasi secara manual, kemudian diproses menggunakan parameter fotogrametri untuk menentukan posisi spasial bukaan. Hasilnya menunjukkan bahwa sistem mampu memperkaya model LoD2 dengan detail fasad secara efektif, meskipun terdapat keterbatasan pada akurasi geometrik akibat absennya kalibrasi kamera dan titik kontrol tanah. Selain memperkuat representasi geometrik, informasi bukaan yang ditambahkan juga terbukti bermanfaat untuk validasi objek pajak bangunan. Dengan pendekatan ini, sistem memberikan solusi efisien dan terjangkau dalam memperkaya model kota digital untuk kebutuhan analisis lanjutan.

Kata kunci: citra fasad, client-server, LoD2, model 3D bangunan, monoplotting, Pajak Bumi dan Bangunan (PBB)

## **ABSTRACT**

### **ENHANCEMENT OF 3D BUILDING MODEL LEVEL OF DETAIL IN DIGITAL CITY MAPS USING MONOPLOTTING TECHNIQUE BASED ON WEB CLIENT-SERVER ARCHITECTURE**

By

**Nunky Febilia Verany**

**NIM: 23222305**

***(Master's Program in Electrical Engineering)***

*Three-dimensional (3D) building models in digital city maps are generally available at Level of Detail 2 (LoD2), which only presents the basic geometry of buildings along with roof shape information, but without detailed facade elements. This study aims to enhance the semantic detail of LoD2 models by adding facade opening information such as windows and doors using a monoplottting technique based on facade imagery. The system developed integrates single-image photogrammetry (monoplottting) within a web-based client-server architecture, allowing users to annotate facade images directly and project the annotations spatially onto the 3D building model.*

*The implementation uses React.js and CesiumJS for frontend visualization, and Node.js for backend services and data management. Facade images were captured using a smartphone camera and manually annotated, then processed using photogrammetric parameters to determine the spatial position of the openings. The results show that the system effectively enriches LoD2 models with detailed facade information, despite limitations in geometric accuracy due to the absence of camera calibration and ground control points. In addition to improving geometric representation, the added opening data is also proven useful for validating building tax objects. This approach offers an efficient and cost-effective solution for enhancing digital city models for further spatial and fiscal analysis.*

*Keywords: 3D building model, client-server, facade imagery, LoD2, monoplottting, property tax*

**PENINGKATAN TINGKAT KEDETAILAN MODEL 3D  
BANGUNAN PADA PETA KOTA DIGITAL MENGGUNAKAN  
TEKNIK MONOPLOTTING BERBASIS ARSITEKTUR  
CLIENT-SERVER WEB**

Oleh  
**Nunky Febilia Verany**  
**NIM: 23222305**  
**(Program Studi Magister Teknik Elektro)**

Institut Teknologi Bandung

Menyetujui  
Tim Pembimbing

Tanggal 17 Juni 2025

Ketua

Anggota

(Prof. Ir. Emir Mauludi Husni, M.Sc.,  
Ph.D.)

(Dr. Deni Suwardhi, S.T, M.T.)

## PEDOMAN PENGGUNAAN TESIS

Tesis Magister yang tidak dipublikasikan terdaftar dan tersedia di Perpustakaan Institut Teknologi Bandung, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada penulis dengan mengikuti aturan HaKI yang berlaku di Institut Teknologi Bandung. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan seizin penulis dan harus disertai dengan kaidah ilmiah untuk menyebutkan sumbernya.

Sitasi hasil penelitian Tesis ini dapat di tulis dalam bahasa Indonesia sebagai berikut:

Verany, Nunky. (2025): *Peningkatan Tingkat Kedetailan Model 3D Bangunan Pada Peta Kota Digital Menggunakan Teknik Monoplotting Berbasis Arsitektur Client-Server Web*, Tesis Program Magister, Institut Teknologi Bandung.

dan dalam bahasa Inggris sebagai berikut:

Verany, Nunky. (2025): *Enhancement of 3D Building Model Level of Detail in Digital City Maps Using Monoplotting Technique Based on Web Client-Server Architecture*, Master's Thesis, Institut Teknologi Bandung.

Memperbanyak atau menerbitkan sebagian atau seluruh tesis haruslah seizin Dekan Sekolah Pascasarjana, Institut Teknologi Bandung.

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tesis yang berjudul “Peningkatan Tingkat Kedetailan Model 3D Bangunan pada Peta Kota Digital Menggunakan Teknik *Monoplotting* Berbasis Arsitektur *Client-Server* Web” sebagai salah satu syarat untuk memperoleh gelar Magister pada Program Studi Magister Teknik Elektro, Institut Teknologi Bandung.

Tesis ini disusun sebagai bentuk kontribusi dalam pengembangan teknologi pemodelan spasial kota berbasis web, khususnya dalam meningkatkan kedetailan semantik model bangunan tiga dimensi *Level of Detail 2* (LoD2) melalui pendekatan visual berbasis citra fasad. Penulis menyadari bahwa penyelesaian penelitian ini tidak terlepas dari bantuan, dukungan, dan bimbingan dari berbagai pihak. Sehingga, penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

- Prof. Ir. Emir Mauludi Husni, M.Sc., Ph.D., selaku pembimbing utama, atas segala bimbingan, arahan, dan motivasi yang diberikan selama proses penelitian ini.
- Dr. Deni Suwardhi, S.T., M.T., selaku pembimbing kedua, atas masukan dan wawasan teknis yang sangat berharga dalam pengembangan sistem dan penyusunan tesis.
- Seluruh dosen dan staf di Program Studi Magister Teknik Elektro ITB, atas ilmu dan pengalaman yang telah diberikan selama masa studi.
- Teman-teman dan rekan seperjuangan di lingkungan kampus, atas kerja sama, diskusi, dan semangat yang saling menguatkan.
- Keluarga tercinta, atas doa, dukungan moral, dan semangat yang tak ternilai sepanjang perjalanan studi ini.

Penulis menyadari bahwa masih terdapat kekurangan dalam penyusunan tesis ini. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun untuk penyempurnaan di masa mendatang. Semoga tesis ini dapat memberikan manfaat, baik bagi pengembangan ilmu pengetahuan maupun bagi pihak-pihak yang berkepentingan dalam bidang pemetaan kota digital dan sistem informasi geografis.

Bandung, Juni 2025

Penulis

## DAFTAR ISI

ABSTRAK.....	i
<i>ABSTRACT</i> .....	<i>ii</i>
PEDOMAN PENGGUNAAN TESIS .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR LAMPIRAN.....	viii
DAFTAR GAMBAR DAN ILUSTRASI.....	ix
DAFTAR TABEL.....	xi
DAFTAR SINGKATAN DAN LAMBANG .....	xii
Bab I Pendahuluan .....	1
I.1 Latar Belakang .....	1
I.2 Rumusan Masalah .....	3
I.3 Tujuan Penelitian .....	4
I.4 Batasan Masalah .....	4
Bab II Tinjauan Pustaka .....	6
II.1 Peta Kota Digital.....	6
II.2 Fotogrammetri.....	7
II.3 Konsep CityGML dalam Representasi Bangunan .....	12
II.4 Sistem Koordinat 3 Dimensi dan Transformasi XYZ.....	13
II.5 Teknik <i>Monoplotting</i> .....	15
II.6 Arsitektur Peta Kota 3D LOD2 pada Sistem Web.....	17
II.7 Pajak Bumi dan Bangunan (PBB) Indonesia .....	19
Bab III Metodologi .....	25
III.1 Diagram Alir Penelitian.....	25
III.2 Tahapan Penelitian .....	26
III.2.1 Studi Literatur dan Analisis Kebutuhan.....	26
III.2.2 Pengumpulan dan Pra-Pemrosesan Data .....	26
III.2.3 Pengembangan Sistem Berbasis Web .....	32
III.2.4 Visualisasi dan Input Informasi Bangunan .....	37
III.2.5 Sistem Anotasi atau Penggambaran Poligon informasi Fasad pada Citra 2D.....	40
III.2.6 Visualisasi Hasil Anotasi pada 2D dan 3D .....	44
III.2.7 Penerapan Teknik <i>Monoplotting</i> .....	47
III.2.8 Perhitungan Objek Pajak.....	49
Bab IV Hasil dan Analisis Penelitian.....	51
IV.1 Tampilan Bangunan LoD2 pada CesiumJS.....	51
IV.2 Informasi Gambar tiap bangunan .....	53
IV.3 Hasil Visualisasi Bukaan Fasad Pada Model 3D Bangunan .....	57
IV.3.1 Sisi Depan Bangunan .....	57
IV.3.2 Sisi Samping Bangunan .....	59
IV.3.3 Sisi Belakang Bangunan .....	61
IV.4 Visualisasi Informasi Pajak Bumi dan Bangunan .....	64
Bab V Kesimpulan .....	66
V.1 Kesimpulan .....	66
V.2 Saran .....	66



## DAFTAR LAMPIRAN

Lampiran A Program <i>viewer building</i> 3D model pada CesiumJS .....	75
Lampiran B Program untuk viewer satu bangunan dan UI untuk penambahan informasi bangunan.....	77
Lampiran C Program untuk anotasi gambar .....	81
Lampiran D Program untuk <i>preview</i> anotasi gambar .....	85
Lampiran E Program untuk proyeksi model 3D jendela pada bangunan .....	88

## DAFTAR GAMBAR DAN ILUSTRASI

Gambar II. 1 Contoh peta digital <i>Google Maps</i> .....	6
Gambar II. 2 Prinsip pembentukan citra dalam fotogrametri (Schenk, 2005).....	8
Gambar II. 3 Dua kamera pada posisi berbeda (Station 1 dan Station 2) yang mengarah ke titik objek yang sama di dunia nyata, titik P(X, Y, Z) (Dai dan Lu, 2010) .....	11
Gambar II. 4 Representasi model LoD bangunan pada CityGML (Biljecki, 2016) .....	12
Gambar II. 5 Prinsip metode <i>monoplotting</i> (Golparvar, 2025).....	16
Gambar II. 6 Contoh tampilan bangunan 3D dengan CesiumJS (Sandcastle Cesium, 2025) .....	18
Gambar III. 1 Diagram alir penelitian.....	25
Gambar III. 2 Contoh foto tampak depan bangunan.....	27
Gambar III. 3 Metadata dari foto tampak depan bangunan .....	27
Gambar III. 4 Contoh foto tampak samping bangunan.....	28
Gambar III. 5 Metadata dari foto tampak samping bangunan .....	28
Gambar III. 6 Contoh foto tampak belakang bangunan.....	29
Gambar III. 7 Metadata dari foto tampak belakang bangunan .....	29
Gambar III. 8 Proses <i>Align Photos</i> dengan Agisoft Metashape (tampak atas).....	30
Gambar III. 9 Proses <i>Align Photos</i> dengan Agisoft Metashape (tampak samping) .....	30
Gambar III. 10 Data referensi yang diperoleh dari <i>Align Photos</i> dengan Agisoft Metashape .....	30
Gambar III. 11 Tampilan assets bangunan pada Cesium ion.....	31
Gambar III. 12 Tampilan struktur file dalam server .....	35
Gambar IV. 1 Tampilan model bangunan 3D pada Cesium ion secara keseluruhan .....	51
Gambar IV. 2 Tampilan model bangunan 3D pada Cesium Ion difokuskan pada bangunan utama yang dianalisis, dengan visualisasi yang menampilkan sisi depan, samping, dan belakang secara berurutan.....	52
Gambar IV. 3 Visualisasi antarmuka sistem untuk menambahkan informasi suatu bangunan .....	53
Gambar IV. 4 Contoh ID bangunan yang ditampilkan .....	54
Gambar IV. 5 Tampilan proses upload foto bangunan ke server.....	54
Gambar IV. 6 Tampilan list foto yang sudah diunggah oleh pengguna .....	55
Gambar IV. 7 Tampilan salah satu foto sisi bangunan yang sudah diunggah oleh pengguna .....	55
Gambar IV. 8 Tampilan salah satu foto sisi bangunan yang sudah dianotasi oleh pengguna .....	56
Gambar IV. 9 Contoh hasil anotasi pengguna pada salah satu sisi bangunan .....	57
Gambar IV. 10 Hasil anotasi pengguna pada sisi depan bangunan .....	58
Gambar IV. 11 Hasil proyeksi bukaan fasad pada model 3D bangunan (sisi depan) .....	58

Gambar IV. 12 Hasil anotasi pengguna pada sisi depan bangunan .....	60
Gambar IV. 13 Hasil proyeksi bukaan fasad pada model 3D bangunan (sisi samping).....	60
Gambar IV. 14 Hasil anotasi pengguna pada sisi belakang bangunan .....	62
Gambar IV. 15 Hasil proyeksi bukaan fasad pada model 3D bangunan (sisi belakang).....	62
Gambar IV. 16 Hasil proyeksi bukaan fasad pada model 3D bangunan secara keseluruhan .....	63
Gambar IV. 17 Visualisasi perhitungan pajak bumi dan bangunan pada bangunan yang dianalisis.....	65

## DAFTAR TABEL

Tabel III. 1 Penjelasan kode program untuk menampilkan objek 3D pada viewer berbasis ReactJS dan CesiumJS.....	33
Tabel III. 2 Penjelasan kode program pada server.....	36
Tabel III. 3 Penjelasan kode program pada visualisasi dan input informasi bangunan.....	39
Tabel III.4 Penjelasan kode program pada proses anotasi atau penggambaran <i>polygon</i> pada citra.....	43
Tabel III. 5 Penjelasan kode program pada proses visualisasi hasil anotasi atau penggambaran poligon.....	46

## DAFTAR SINGKATAN

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
SIG	<i>Sistem Informasi Geografis</i>	1
LoD	<i>Level of Detail</i>	1
GIS	<i>Geographic Information System</i>	1
CAD	<i>Computer-Aided Design</i>	1
3D	<i>Three-Dimensional</i>	1
GPS	<i>Global Positioning System</i>	4
2D	<i>Two-Dimensional</i>	8
CityGML	<i>City Geography Markup Language</i>	12
GML	<i>Geography Markup Language</i>	12
OGC	<i>Open Geospatial Consortium</i>	12
OSM	<i>OpenStreetMap</i>	13
IndoorOSM	<i>Indoor OpenStreetMap</i>	13
BIM	<i>Building Information Modeling</i>	13
WGS84	<i>World Geodetic System 1984</i>	14
LiDAR	<i>Light Detection and Ranging</i>	15
DEM	<i>Digital Elevation Model</i>	15
DSM	<i>Digital Surface Model</i>	16
CityJSON	<i>City Geography JSON Format (versi JSON dari CityGML)</i>	18
WebGL	<i>Web Graphics Library</i>	18
glTF	<i>GL Transmission Format</i>	18
WebGIS	<i>Web Geographic Information System</i>	19
PostgreSQL	<i>Post-Ingres Structured Query Language</i>	19
PostGIS	<i>Geographic Information System extension for PostgreSQL</i>	19
3DcityDB	<i>3D City Database</i>	19
API	<i>Application Programming Interface</i>	19
PBB	Pajak Bumi dan Bangunan	19
PBB-P2	Pajak Bumi dan Bangunan Perdesaan dan Perkotaan	19
UU	Undang-Undang	20
NJOP	Nilai Jual Objek Pajak	20
UU PDRD	Undang-Undang Pajak Daerah dan Retribusi Daerah	20
UU HKPD	Undang-Undang Hubungan Keuangan antara Pemerintah Pusat dan Daerah	20
Perda	Peraturan Daerah	20
PMK	Peraturan Menteri Keuangan	20
NIR	Nilai Indeks Rata-rata	20
DBKB	Daftar Biaya Komponen Bangunan	21

NJKP	Nilai Jual Kena Pajak	21
NJOPTKP	Nilai Jual Objek Pajak Tidak Kena Pajak	21
Menkeu	Menteri Keuangan	21
SISMIOP	Sistem Informasi Manajemen Objek Pajak	22
PAD	Pendapatan Asli Daerah	23
SPPT	Surat Pemberitahuan Pajak Terhutang	23
MP	<i>Megapixel</i>	26
OIS	<i>Optical Image Stabilization</i>	26
EXIF	<i>Exchangeable Image File Format</i>	26
GeoJSON	<i>Geographical JavaScript Object Notation</i>	31
HTML	<i>HyperText Markup Language</i>	33
RESTful API	<i>Representational State Transfer Application Programming Interface</i>	34
URL	<i>Uniform Resource Locator</i>	37
EPSG	<i>European Petroleum Survey Group</i>	38
HTTP	<i>HyperText Transfer Protocol</i>	38

# **Bab I     Pendahuluan**

## **I.1   Latar Belakang**

Peta digital merupakan representasi fenomena geografis yang disimpan dalam format digital dan dapat dianalisis secara komputasional (Dangermond & Goodchild, 2020). Dalam konteks perkotaan, peta kota berfungsi tidak hanya sebagai alat navigasi, tetapi juga sebagai media penyampaian informasi mengenai lokasi-lokasi penting, seperti peta jalan maupun peta destinasi wisata. Seiring dengan pesatnya perkembangan teknologi Sistem Informasi Geografis (SIG), fungsi peta kota mengalami perluasan yang signifikan. Peta kota digital tidak lagi sekedar berperan sebagai sarana visualisasi spasial, melainkan telah berkembang menjadi fondasi bagi analisis keruangan yang kompleks, mencakup penghitungan parameter geometri, identifikasi pola keterkaitan spasial, serta mendukung pengambilan keputusan dalam perencanaan, pengelolaan, dan pengembangan wilayah kota (Satar, 2014).

Salah satu wujud lanjut dari peta kota digital adalah model bangunan tiga dimensi (3D) yang dikonstruksi berdasarkan pendekatan *Level of Detail* (LoD). Konsep ini digunakan untuk menggambarkan tingkatan ketelitian baik secara geometris maupun semantik dari representasi bangunan. Model LoD1, sebagai contoh, menampilkan bangunan dalam bentuk blok sederhana dengan atap datar hasil ekstrusi denah, tanpa menyertakan detail fasad seperti jendela, pintu, atau ornamen lainnya. Kemudian untuk model yang lebih lanjut, model LoD3, menyajikan representasi yang jauh lebih rinci dengan mencakup elemen-elemen arsitektural seperti bukaan jendela dan pintu (Biljecki dkk., 2016a). Meskipun model LoD3 menawarkan tingkat kedetilan yang tinggi, model LoD1 dan LoD2 justru lebih banyak tersedia secara global karena proses pembuatannya yang relatif sederhana yaitu dengan menggunakan data denah dan ketinggian bangunan (Bieringer dkk., 2024). Akan tetapi, model-model tersebut memiliki keterbatasan signifikan, terutama pada aspek semantik, karena tidak memuat informasi detail pada bagian fasad bangunan. Padahal, elemen fasad seperti jendela dan pintu memiliki peran penting dalam berbagai jenis analisis lanjutan, antara lain simulasi pencahayaan

alami, perhitungan efisiensi energi bangunan, serta evaluasi kondisi fisik bangunan dalam konteks audit, pajak, atau perizinan.

Peningkatan semantik pada model bangunan LoD2 melalui penambahan informasi bukaan fasad memiliki potensi besar untuk memperluas kegunaan model tersebut dalam berbagai skenario aplikasi. Informasi mengenai jendela, sebagai contoh, dapat dimanfaatkan untuk analisis yang lebih mendalam terhadap karakteristik bangunan terutama pada penilaian bangunan untuk keperluan perizinan atau perpajakan. Beberapa studi menunjukkan bahwa bangunan yang memiliki informasi detail pada fasadnya dapat memberikan representasi geometris yang lebih rinci (Biljecki dkk., 2016a). Namun demikian, meskipun model LoD3 mampu menyediakan detail yang lebih rinci, ketersediaannya masih sangat terbatas. Hal ini disebabkan oleh kompleksitas dan biaya tinggi dalam proses pembuatannya, yang sering kali membutuhkan metode seperti survei lapangan, fotogrametri, atau pemindaian laser. Oleh karena itu, dibutuhkan pendekatan yang lebih efisien dan terjangkau untuk memperkaya model LoD2 dengan informasi bukaan fasad, tanpa harus membangun model LoD3 secara penuh dari awal.

Salah satu pendekatan yang menjanjikan untuk menambahkan informasi detail fasad ke dalam model LoD2 adalah melalui pemanfaatan citra fasad bangunan dan teknik pemrosesan citra berbasis fotogrametri. Dalam konteks ini, teknik *monoplotting* menjadi metode yang relevan dan efisien. *Monoplotting* adalah metode fotogrametri yang memungkinkan ekstraksi informasi spasial tiga dimensi (3D) dari sebuah citra tunggal, dengan syarat tersedia data referensi seperti model elevasi atau geometri objek dasar (Huang & Keong KWOH, 2008). Dengan menggunakan model LoD2 sebagai acuan geometris, titik atau poligon yang digambarkan secara manual pada citra fasad dapat diproyeksikan secara spasial ke posisi yang sesuai pada permukaan bangunan 3D. Pendekatan ini memanfaatkan data yang relatif mudah diakses—seperti foto bangunan—dan model dasar bangunan yang sederhana, sehingga secara signifikan lebih ekonomis dibandingkan dengan proses konstruksi model LoD3 dari nol. Terdapat studi sebelumnya yang menunjukkan bahwa metode berbasis *monoplotting* memungkinkan anotasi fasad secara semi-manual dengan ketelitian georeferensi yang memadai (Murtiyoso dkk., 2019). Studi tersebut menjadi salah satu referensi metode



*monoplotting* sebagai solusi praktis untuk meningkatkan informasi semantik model LoD2 dengan menggunakan informasi citra bangunan.

Integrasi proses ekstraksi informasi fasad dari citra dengan sistem berbasis web menawarkan keuntungan dalam hal aksesibilitas dan kemudahan penggunaan. Dengan kemajuan teknologi *web*, kini visualisasi 3D dapat dijalankan di peramban (*browser*) menggunakan WebGL atau pustaka JavaScript 3D, dan antarmuka interaktif dapat dibangun dengan kerangka kerja seperti React.js. Pada penelitian ini dikembangkan sistem prototipe dengan *frontend* React.js dan *backend* Node.js yang memungkinkan pengguna mengunggah atau menampilkan foto fasad bangunan, dengan menambahkan gambar poligon (untuk menjelaskan area jendela pada foto), kemudian sistem akan melakukan konversi poligon tersebut menjadi elemen jendela dalam model 3D LoD2 yang ditampilkan. Hasil ekstraksi (informasi jendela) disimpan di server sehingga dapat diakses dan dianalisis lebih lanjut. Melalui pendekatan *web-based* ini, proses penambahan detail bangunan dapat dilakukan secara interaktif dan real-time melalui *browser* tanpa memerlukan perangkat lunak khusus GIS (*Geographic Information System*) atau CAD (*Computer-Aided Design*) di sisi pengguna. Sebagai contoh kasus pemanfaatan, aplikasi ini dapat digunakan untuk validasi objek pajak bangunan. Meskipun demikian, perlu digarisbawahi bahwa fokus penelitian ini bukan pada aspek perpajakan itu sendiri, melainkan pada perancangan dan implementasi sistem yang mengintegrasikan pemrosesan citra fasad dan visualisasi 3D bangunan.

## **I.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

### **1. Rumusan Masalah Utama**

Bagaimana mengembangkan sistem pemodelan tiga dimensi bangunan berbasis foto fasad menggunakan teknik *monoplotting* dan memvisualisasikannya melalui antarmuka web guna mendukung validasi objek pajak?

### **2. Sub-rumusan Masalah:**

- Bagaimana pemrosesan data foto fasad bangunan sebagai input untuk teknik *monoplotting*?

- Bagaimana penerapan teknik *monoplotting* pada foto fasad dalam merekonstruksi model tiga dimensi bangunan?
- Bagaimana merancang dan mengimplementasikan visualisasi model tiga dimensi bangunan hasil rekonstruksi pada antarmuka web?
- Bagaimana pemanfaatan model tiga dimensi bangunan berbasis web hasil rekonstruksi dalam proses validasi objek pajak?

### **I.3 Tujuan Penelitian**

Penelitian ini bertujuan untuk mengembangkan sebuah sistem berbasis web yang mampu memperkaya model bangunan tiga dimensi LoD2 dengan informasi bukaan fasad (seperti jendela dan pintu) melalui teknik *monoplotting* terhadap foto fasad bangunan, serta mengintegrasikan hasilnya dalam visualisasi 3D yang dapat digunakan untuk mendukung validasi objek pajak.

Secara khusus, tujuan penelitian ini adalah sebagai berikut:

- 1) Mengembangkan metode pemrosesan foto fasad bangunan yang sesuai sebagai input sistem.
- 2) Menerapkan teknik *monoplotting* untuk mentransformasi anotasi pada citra fasad menjadi informasi spasial tiga dimensi pada model bangunan.
- 3) Merancang dan mengimplementasikan antarmuka web interaktif yang dapat memvisualisasikan model 3D bangunan hasil rekonstruksi secara *real-time*.
- 4) Menunjukkan potensi penerapan sistem ini dalam mendukung validasi objek pajak dengan menyediakan representasi geometris dan semantik bangunan yang lebih lengkap.

### **I.4 Batasan Masalah**

Agar penelitian ini terfokus dan terarah, terdapat beberapa batasan masalah yang diterapkan, yaitu:

- 1) Model bangunan yang digunakan terbatas pada model tiga dimensi *Level of Detail 2* (LoD2) yang tidak memiliki detail fasad, namun mencakup geometri dasar seperti denah, ketinggian bangunan, dan informasi atap.
- 2) Citra fasad yang digunakan merupakan foto tegak lurus (ortogonal) terhadap bidang fasad, diambil dari darat, dan mengandung informasi GPS

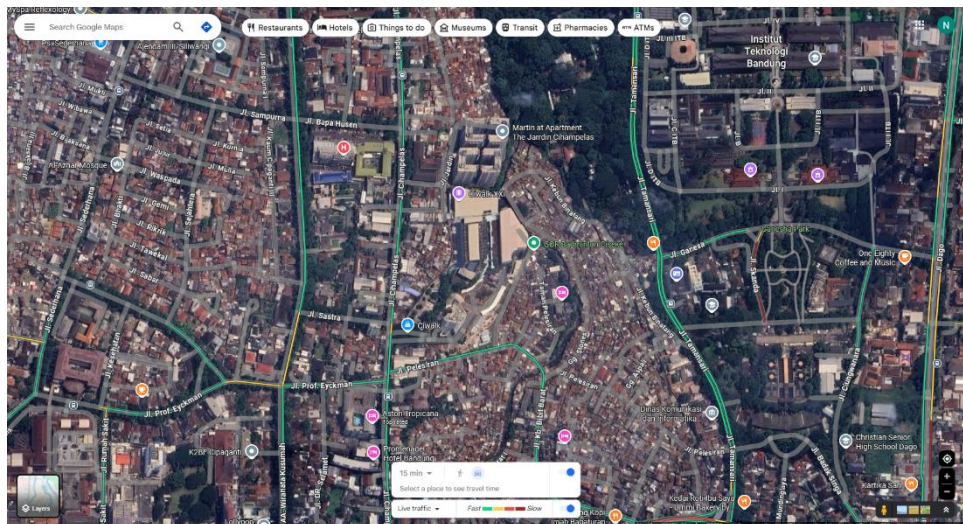
(*Global Positioning System*) serta orientasi kamera sebagai prasyarat penerapan teknik *monoplotting*.

- 3) Teknik *monoplotting* yang digunakan bersifat semi-manual, yaitu pengguna menentukan anotasi bukaan (jendela/pintu) secara interaktif pada foto fasad, kemudian sistem akan memproyeksikannya ke model 3D menggunakan referensi geometri bangunan.
- 4) Anotasi atau penggambaran poligon untuk menandai bukaan dilakukan secara manual pada citra fasad. Penelitian ini tidak mencakup otomasi pendeteksian jendela menggunakan algoritma visi komputer atau machine learning.
- 5) Sistem tidak mengimplementasikan pemrosesan citra otomatis berbasis machine learning untuk mengenali atau mengekstrak fitur fasad. Semua identifikasi elemen (jendela) bergantung pada input manual pengguna.
- 6) Jenis bukaan yang diekstraksi dibatasi pada jendela dan pintu dengan bentuk sederhana (persegi panjang), tanpa mempertimbangkan ornamen atau bentuk arsitektural yang kompleks.
- 7) *Platform* pengembangan sistem dibatasi pada teknologi berbasis web, dengan *frontend* menggunakan React.js untuk visualisasi dan *backend* menggunakan Node.js untuk pemrosesan data dan penyimpanan hasil.
- 8) Keluaran sistem difokuskan pada visualisasi dan penyimpanan elemen bukaan dalam format geospasial (GeoJSON), tanpa melakukan simulasi lanjutan seperti pencahayaan atau energi.
- 9) Penerapan validasi objek pajak dibatasi pada demonstrasi potensi sistem sebagai pendukung visualisasi informasi spasial bangunan yang lebih kaya, bukan pada aspek perhitungan atau kebijakan perpajakan itu sendiri.

## Bab II Tinjauan Pustaka

### II.1 Peta Kota Digital

Peta digital adalah representasi komputerisasi dari informasi geografis, di mana data geospasial direferensikan secara geografis berdasarkan lokasinya. Salah satu kegunaan utama dari peta digital adalah kemampuan dalam merekam dan melestarikan informasi geografis dari berbagai periode waktu, termasuk peta historis yang tidak lagi tersedia dalam bentuk fisik asli (Chiang dkk., 2014). Menurut Walter dan Luo, peta digital menyimpan banyak informasi implisit yang tidak direpresentasikan secara eksplisit dalam data, namun dapat diekstraksi melalui proses interpretasi peta. Informasi implisit tersebut mencakup struktur permukiman, pusat kota, atau kepadatan penduduk, yang dahulu hanya dapat dilakukan oleh kartografer, kini bisa diekstrak melalui metode komputasi otomatis (*automatic map interpretation*) (Walter & Luo, 2011). Berbeda dengan peta konvensional (cetak), peta digital memungkinkan interaksi pengguna secara dinamis. Pengguna dapat mengubah skala tampilan, menampilkan atau menyembunyikan berbagai lapisan informasi, serta menghubungkan elemen-elemen pada peta dengan basis data eksternal untuk mendapatkan informasi tambahan seperti sejarah bangunan atau informasi spasial lainnya (Jones dkk., 2004).



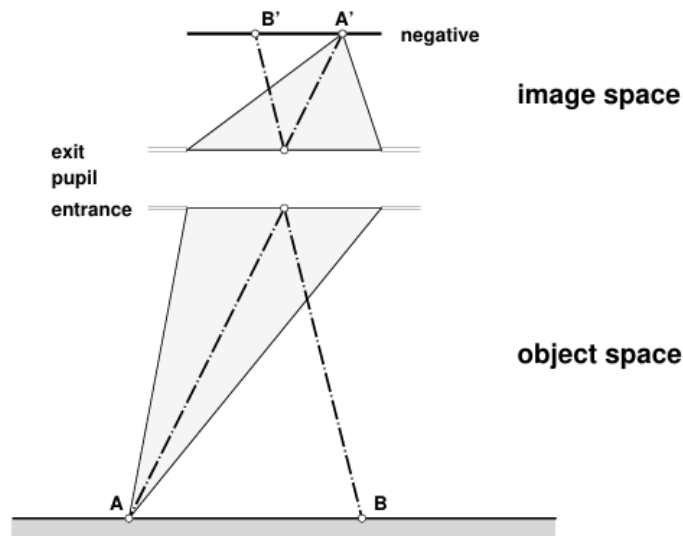
**Gambar II. 1** Contoh peta digital *Google Maps* (Google, 2025)

Peta kota digital merupakan representasi spasial dari berbagai entitas kota seperti bangunan, jalan, vegetasi, dan infrastruktur lainnya dalam bentuk digital yang dapat dianalisis dan divisualisasikan secara komputasional (Chiang dkk., 2014). Representasi ini menjadi landasan bagi berbagai aplikasi SIG, termasuk pemetaan tematik, pemodelan kota 3D, serta simulasi dan visualisasi urban. Salah satu fungsi utama peta kota digital adalah untuk membangun lingkungan virtual yang merepresentasikan kondisi riil suatu kawasan urban (Yoo & Han, 2006).

Contoh penerapan peta kota digital diteliti oleh Yang, Byungyun mengenai pengembangan kota cerdas (smart city). Dalam penelitiannya peta kota digital dalam bentuk model kota tiga dimensi memiliki peranan penting dalam meningkatkan persepsi spasial dan pemahaman tata ruang baik bagi perencana kota maupun masyarakat umum. Model 3D memungkinkan visualisasi area perkotaan secara virtual dan spasial sehingga mendukung proses pengambilan keputusan dalam perencanaan urban. Model kota 3D dapat menyajikan informasi secara fotorealistik dan memperkuat persepsi spasial terhadap objek geografis (Yang, 2019).

## **II.2 Fotogrammetri**

Fotogrametri merupakan ilmu yang mempelajari hubungan antara titik-titik di ruang objek dengan titik-titik yang bersesuaian pada bidang citra (*image plane*), terutama melalui proses pemotretan dan pengolahan gambar (Roos, 1951). Menurut Schenk, konsep dasar fotogrametri didasarkan pada relasi geometris antara ruang gambar (*image space*) dan ruang objek (*object space*) (Schenk & Quarter, 2005).



**Gambar II. 2** Prinsip pembentukan citra dalam fotogrametri (Schenk & Quarter, 2015)

Dengan:

- Titik A dan B: Merupakan titik-titik objek di permukaan bumi.
- Garis dari A dan B menuju *entrance* pupil: Menunjukkan jalur sinar cahaya yang dipantulkan dari objek menuju kamera.
- *Entrance* pupil: Titik masuk sinar ke dalam sistem optik kamera.
- *Exit* pupil: Titik keluar sinar dari sistem optik menuju sensor atau film.
- Dalam kamera ideal (seperti kamera fotogrametri), *entrance* pupil dan *exit* pupil digunakan untuk mendefinisikan jalur sinar dalam sistem optik.
- Titik A' dan B': Merupakan hasil proyeksi dari titik A dan B pada bidang negatif (film atau sensor)
- Garis putus-putus dari A dan B ke A' dan B': Menunjukkan jalur sinar proyeksi perspektif (dasar utama dalam fotogrametri untuk menentukan kedudukan objek di ruang dari citra dua dimensi (2D)).
- Negatif: Merujuk pada bidang tempat citra direkam (bisa berupa film negatif dalam fotografi konvensional atau sensor digital dalam kamera modern). Orientasi objek yang dihasilkan terbalik (seperti B menjadi B', dan A menjadi A').

Fotogrametri bertujuan untuk memperoleh informasi yang dapat diandalkan tentang objek melalui pengukuran dan interpretasi data citra. Fotogrametri dilakukan dengan menggunakan prinsip dasar perspektif yaitu, sinar dari kamera menuju titik objek akan saling berpotongan pada lokasi titik tersebut di ruang 3D (Grussenmeyer & Al Khalil, 2002).

Willneff menyatakan bahwa prinsip utama dalam fotogrametri adalah persamaan kolinearitas, yang menjelaskan hubungan geometris antara titik objek di dunia nyata, titik citra, dan pusat proyeksi kamera (Willneff dkk., 2005). Persamaan kolinearitas fotogrammetry dijelaskan oleh Stockdale sebagai berikut (Stockdale dkk., 2015):

$$x_a - x_0 = -f \cdot \frac{r_{11}(X_A - X_C) + r_{21}(Y_A - Y_C) + r_{31}(Z_A - Z_C)}{r_{13}(X_A - X_C) + r_{23}(Y_A - Y_C) + r_{33}(Z_A - Z_C)} \quad (\text{II.1})$$

$$y_a - y_0 = -f \cdot \frac{r_{12}(X_A - X_C) + r_{22}(Y_A - Y_C) + r_{32}(Z_A - Z_C)}{r_{13}(X_A - X_C) + r_{23}(Y_A - Y_C) + r_{33}(Z_A - Z_C)} \quad (\text{II.2})$$

Dengan:

$x_a, y_a$  = Koordinat titik pada citra (*image plane*)

$x_0, y_0$  = Posisi pusat utama (principal point) citra

$f$  = Panjang fokus kamera

$X_A, Y_A, Z_A$  = Koordinat titik di dunia nyata (*object point*)

$X_C, Y_C, Z_C$  = Posisi kamera di ruang (*camera center*)

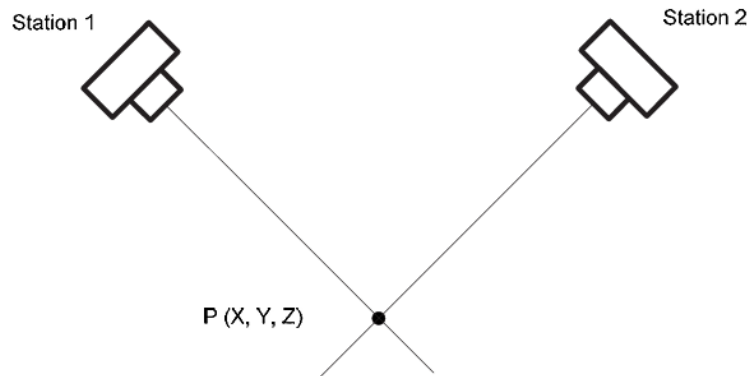
$r_{ij}$  = Elemen dari matriks rotasi yang diturunkan dari tiga sudut rotasi kamera ( $\omega, \varphi, \kappa$ )

Dalam hal ini, posisi sebuah titik di bidang gambar berhubungan langsung dengan posisi kamera dan orientasinya terhadap sistem koordinat dunia. Persamaan ini memuat parameter orientasi interior, seperti panjang fokus dan titik utama kamera, serta parameter orientasi eksterior, yaitu posisi kamera dalam ruang dan rotasinya (Willneff dkk., 2005).

Konsep fotogrametri secara umum merujuk pada teknik pengukuran yang memungkinkan ekstraksi titik-titik 3D dari citra 2D. Dalam konteks pemodelan bangunan, fotogrametri digital memungkinkan pembangunan model 3D yang akurat tanpa perlu pengukuran langsung pada objek fisik (Shashi dkk., 2007). Fotogrametri melibatkan teknik untuk memperoleh informasi geometrik seperti posisi, ukuran, dan bentuk objek dari citra foto. Metode ini memungkinkan perekaman dan rekonstruksi tiga dimensi (3D) dari suatu objek menggunakan citra digital, baik dari kamera digital maupun hasil pemindaian foto analog (Hanke dkk., 2002). Teknologi ini sangat berguna terutama dalam rekonstruksi arsitektural hingga aplikasi perencanaan dan visualisasi kota. Dibandingkan dengan survei lapangan konvensional yang lebih memakan waktu dan tenaga, fotogrametri menawarkan alternatif yang lebih efisien (Shashi dkk., 2007).

Dalam penelitian Dai dan Lu penerapan metode fotogrametri dilakukan untuk pengukuran elemen-elemen konstruksi secara presisi, seperti dimensi jendela, pilar, dan panel bangunan. Tujuan utama dari pendekatan ini adalah untuk memverifikasi kuantitas pekerjaan yang telah diselesaikan, menilai kesesuaian produk bangunan terhadap spesifikasi teknis, serta menyediakan bukti untuk mendukung permintaan pembayaran oleh kontraktor. Salah satu aspek penting dalam penelitian tersebut menjelaskan bahwa metode fotogrametri dapat diaplikasikan pada pasangan foto dari dua sudut berbeda, yang memungkinkan ekstraksi informasi dimensi objek melalui prinsip triangulasi, yaitu proses menghitung posisi titik 3D berdasarkan persilangan dua garis pandang dari kamera yang berbeda (Dai dkk., 2010).





**Gambar II. 3** Dua kamera pada posisi berbeda (Station 1 dan Station 2) yang mengarah ke titik objek yang sama di dunia nyata, titik  $P(X, Y, Z)$  (Dai dkk., 2010)

Dengan:

- *Station 1* dan *Station 2* masing-masing mengambil gambar dari arah yang berbeda terhadap titik P.
- Titik potong atau titik terdekat antar kedua garis ini dihitung sebagai posisi 3D titik P.
- Ketika posisi relatif antara kamera dan parameter internal kamera diketahui (kalibrasi), maka garis pandang (*rays*) dari masing-masing kamera ke titik objek dapat dibentuk. Titik potong kedua garis pandang inilah yang menghasilkan koordinat 3D dari titik tersebut di dunia nyata.

Konsep dasar fotogrametri dipahami sebagai teknik survei tiga dimensi yang memanfaatkan citra dua dimensi untuk membangun model 3D, dan memiliki keunggulan dalam hal non-invasif, efisiensi biaya, dan kemampuan menghasilkan informasi geometri serta kedalaman permukaan secara kuantitatif. Fotogrametri, khususnya digital fotogrametri, digunakan untuk menganalisis dan memonitor kondisi permukaan bangunan, terutama pada konteks pelestarian bangunan bersejarah (Galantucci & Fatiguso, 2019). Dalam penelitian Galantucci dan Fatiguso pendekatan fotogrametri digunakan untuk membangun model 3D dari fasad bangunan bersejarah menggunakan perangkat lunak Agisoft PhotoScan. Model yang dihasilkan terdiri dari point cloud, polygonal mesh, dan ortho-photo beresolusi tinggi. Setiap model digunakan untuk mendeteksi kerusakan seperti retakan dan

kehilangan material dengan bantuan filter spasial dan algoritma ekstraksi kontur. Proses ini menghasilkan peta berwarna untuk membantu identifikasi jenis kerusakan secara visual dan kuantitatif.

### II.3 Konsep CityGML dalam Representasi Bangunan

Dalam konteks pemodelan kota secara tiga dimensi, *City Geography Markup Language* atau CityGML merupakan standar internasional yang dirancang untuk merepresentasikan dan memfasilitasi pertukaran data model kota 3D secara semantik. CityGML dikembangkan berdasarkan *Geography Markup Language* versi 3.1.1 (GML3) dan ditetapkan sebagai standar oleh *Open Geospatial Consortium* (OGC). Berbeda dengan format geometrik murni, CityGML tidak hanya menekankan pada bentuk dan tampilan visual dari objek kota, tetapi juga menyertakan representasi semantik, struktur ontologis, atribut tematik, dan hubungan antar objek (Kolbe, 2009).

Secara umum, CityGML mendukung empat aspek utama dari model kota virtual yaitu, semantik, geometri, topologi, dan penampilan. Dalam hal ini, setiap objek dalam CityGML dapat direpresentasikan dalam lima tingkat LoD yang terstandar, yakni LoD0 hingga LoD4, dengan kompleksitas dan akurasi geometrik yang meningkat di setiap tingkatannya. Misalnya, LoD1 direpresentasikan sebagai blok bangunan ekstrusi sederhana, LoD2 sudah mencakup bentuk atap dan instalasi utama seperti balkon, LoD3 menyediakan detail arsitektural seperti jendela dan pintu, sementara LoD4 mencakup struktur interior seperti ruangan dan tangga (Kolbe, 2009).



**Gambar II. 4** Representasi model LoD bangunan pada CityGML (Biljecki dkk., 2016)

Dalam penelitian Goetz yang berjudul "*Towards generating highly detailed 3D CityGML models from OpenStreetMap*", dijelaskan bahwa penerapan CityGML difokuskan untuk meningkatkan tingkat keterperincian model kota tiga dimensi, hingga mencapai Level of Detail 4 (LoD4). Pendekatan ini memanfaatkan data geospasial yang bersumber dari komunitas, seperti OpenStreetMap (OSM). Studi tersebut menunjukkan bahwa model CityGML yang lebih detail dapat dihasilkan dengan menambahkan informasi tambahan ke data dasar contohnya, melalui pemanfaatan skema IndoorOSM (*Indoor OpenStreetMap*) untuk melengkapi struktur internal bangunan. Goetz menyebutkan bahwa model dengan tingkat detail LoD3 dan LoD4 mampu merepresentasikan fitur-fitur seperti bukaan (jendela dan pintu), serta struktur interior bangunan secara lengkap. Penerapan skema ini bertujuan untuk memenuhi kebutuhan akan model kota digital dalam kerangka inisiatif *Digital Earth* yang diprakarsai oleh Al Gore, dengan penekanan bahwa informasi detail dapat diakses dan dimanfaatkan secara lebih luas (Goetz, 2013).

#### **II.4 Sistem Koordinat 3 Dimensi dan Transformasi XYZ**

Salah satu dasar utama dalam sistem pemetaan dan pemodelan geospasial tiga dimensi adalah penggunaan sistem koordinat global. Sistem ini memungkinkan posisi suatu objek direpresentasikan secara presisi dalam ruang, menggunakan sumbu kartesian tiga dimensi (X, Y, Z) yang berpusat pada pusat massa bumi (Van Sickel, 2025).

Dalam pemanfaatannya, transformasi dari sistem koordinat lokal ke sistem koordinat global dilakukan dengan model transformasi tujuh parameter (7-parameter transformation), atau dikenal juga sebagai transformasi Bursa–Wolf, yaitu model matematika untuk memindahkan koordinat tiga dimensi dari satu sistem datum geosentris ke sistem datum lainnya. Model ini mempertahankan sifat konformal (yakni menjaga sudut tetap), namun memungkinkan perubahan skala secara global (Chang dkk., 2018). Secara matematis, hubungan antara koordinat dalam dua sistem referensi dapat dinyatakan sebagai:

$$y = t + s \cdot R \cdot x \quad (\text{II.3})$$

Dengan:

$x$  = Vektor koordinat dalam sistem sumber

$y$  = Vektor koordinat dalam sistem target

$t = [t_x, t_y, t_z]^T$  adalah vektor translasi (meter)

$s$  = faktor skala global

$R$  = matriks rotasi  $3 \times 3$  (*Direction Cosine Matrix*) yang ortogonal dengan determinan +1

Persamaan ini berasal dari formulasi dasar transformasi Helmert tiga dimensi dan digunakan sebagai pendekatan dalam banyak aplikasi geodesi, termasuk transformasi datum antar sistem koordinat lokal dan global (Chang dkk., 2018). Transformasi ini menjadi langkah krusial agar data dari lingkungan desain atau pengukuran lokal dapat digunakan bersama dengan sistem peta dunia seperti WGS 84, yang merupakan sistem acuan pada GPS dan aplikasi SIG global.

*World Geodetic System 1984* (WGS 84) merupakan kerangka acuan geodetik global modern yang dikembangkan oleh *Defense Mapping Agency* (DMA), dan menjadi standar utama untuk pemetaan, navigasi satelit, serta sistem informasi geografis (GIS). WGS 84 mendefinisikan sistem koordinat kartesian tiga dimensi yang berpusat di massa bumi (*geocentric*), dengan sumbu Z sejajar dengan kutub konvensional dan sumbu X berada di bidang ekuator terhadap meridian nol (Kumar, 1988).

Salah satu aspek penting dari WGS 84 adalah kemampuannya dalam mengintegrasikan berbagai datum lokal ke dalam sistem global melalui parameter transformasi koordinat, yang mencakup pergeseran sumbu X, Y, dan Z. Dengan lebih dari 80 datum lokal yang dikaitkan menggunakan titik kontrol dari stasiun *Doppler*, sistem ini memberikan solusi global terhadap keragaman sistem referensi lokal yang selama ini menyulitkan proses integrasi spasial antarwilayah. Selain itu, WGS84 menyediakan definisi elipsoid dan model gravitasi bumi dengan akurasi tinggi, yang penting untuk

menentukan tinggi geodetik ( $h$ ), tinggi ortometrik ( $H$ ), dan undulasi geoid ( $N$ ), sebagaimana dijelaskan dalam rumus (Kumar, 1988):

$$H = h - N \quad (\text{II.4})$$

Dengan:

$H$  = Tinggi ortometrik

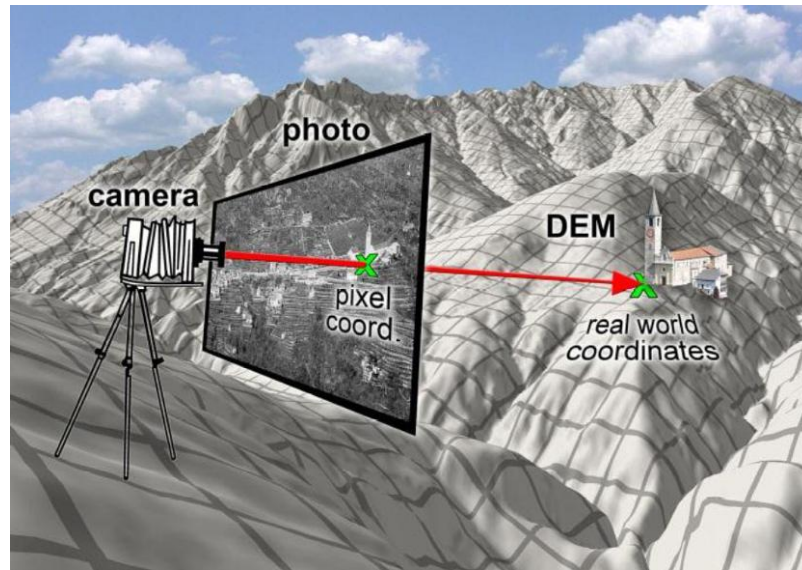
$h$  = Tinggi geodetik

$N$  = Undulasi geoid

Dalam proses ekstraksi informasi bangunan, Zhou menjelaskan bahwa transformasi koordinat XYZ dari data citra (hasil fotogrametri atau LiDAR (*Light Detection and Ranging*)) ke sistem referensi dunia nyata diperlukan untuk merekonstruksi bentuk bangunan secara presisi. Hal ini dilakukan melalui tahapan registrasi dan koreksi geometrik, di mana koordinat lokal hasil observasi diubah menjadi sistem koordinat dunia seperti WGS84 (Zhou dkk., 2016).

## II.5 Teknik Monoplotting

*Monoplotting* adalah teknik fotogrametri monositik yang menggunakan satu citra dan *Digital Elevation Model* (DEM) untuk memperoleh informasi spasial tiga dimensi. Dengan *monoplotting*, garis pandang (*ray*) dari pusat proyeksi kamera melalui tiap titik piksel pada citra diproyeksikan ke permukaan tanah melalui DEM (*Digital Elevation Model*). Posisi objek pada citra 2D dapat dihitung dengan mengintersepsikan sinar tersebut dengan DEM, sehingga diperoleh koordinat 3D objek di bumi (Huang & Keong KWOH, 2008).



**Gambar II. 5** Prinsip metode *monoplanotting* (Golparvar & Wang, 2025)

Prinsip *monoplanotting* mengasumsikan bahwa pusat proyeksi kamera, titik citra (2D), dan titik dunia nyata (3D) terletak pada satu garis lurus. Persamaan yang dapat menjelaskan hubungan tersebut adalah persamaan kolinearitas fotogrametri (Persamaan II.1 dan II.2) (Stockdale dkk., 2015). Persamaan kolinearitas menjadi dasar utama dalam proses rekonstruksi tiga dimensi dari satu atau lebih citra, dan menjadi bagian penting dalam berbagai pendekatan pemetaan fotogrametri, termasuk *monoplanotting* (Willneff dkk., 2005). Dengan memahami posisi dan orientasi kamera saat pengambilan gambar, maka setiap titik yang terdapat pada citra dapat ditransformasikan ke sistem koordinat dunia melalui proses proyeksi ke permukaan digital (DSM (*Digital Surface Model*) atau DEM), sebagaimana dilakukan dalam *monoplanotting*.

Metode *monoplanotting* menjadi relevan terutama ketika hanya tersedia arsip citra lama yang bersifat non-metrik dan memiliki parameter orientasi kamera yang tidak diketahui secara pasti. Mihaela Triglav-Čekada et al. dalam penelitiannya menyatakan bahwa, berkat kemajuan teknologi penginderaan jauh seperti LiDAR dan DEM resolusi tinggi, bahkan citra tunggal *oblique* dengan sudut pandang yang minim informasi pun dapat digunakan untuk ekstraksi data spasial melalui proses *monoplanotting* (Triglav-Čekada dkk., 2014). Langkah pertama dalam prosedur *monoplanotting* adalah melakukan orientasi eksternal citra terhadap sistem koordinat bumi dengan

menggunakan DEM. Dalam banyak kasus, terutama untuk citra lama, orientasi otomatis tidak dapat dilakukan karena jumlah titik kontrol atau fitur umum (*tie points*) sangat terbatas. Oleh karena itu, penulis mengembangkan prosedur orientasi interaktif secara manual, di mana pengguna menyesuaikan parameter rotasi dan posisi kamera dengan mencocokkan citra dengan tampilan *back-projection* dari DEM (Triglav-Čekada dkk., 2014).

Bozzini et al. dalam penelitiannya mengembangkan metode *monoplotting* berbasis citra tunggal untuk mendokumentasikan proses penggalian arkeologi dengan memanfaatkan citra foto digital dan model permukaan digital (DSM). Pendekatan ini memungkinkan pengguna untuk memproyeksikan data spasial berupa titik, garis, atau area secara langsung dari citra ke dalam sistem koordinat dunia nyata tanpa memerlukan proses kalibrasi kamera yang kompleks maupun penggunaan perangkat lunak fotogrametri komersial (Bozzini dkk., 2012).

Golparvar-Fard et al. menggunakan metode *monoplotting* untuk mendukung pemantauan kemajuan konstruksi secara otomatis. Pendekatan ini bertumpu pada prinsip bahwa satu citra saja, jika diketahui parameter orientasi kameranya serta didukung oleh model permukaan (atau model BIM 3D), dapat digunakan untuk mengekstraksi informasi spasial dan membandingkannya dengan model perencanaan. Proses ini melibatkan penggunaan parameter orientasi kamera, baik interior maupun eksterior, untuk menghubungkan koordinat piksel pada citra dengan posisi nyata dalam ruang tiga dimensi melalui proyeksi kolinearitas (Golparvar & Wang, 2025).

## **II.6 Arsitektur Peta Kota 3D LOD2 pada Sistem Web**

Visualisasi kota 3D berbasis web telah menjadi pendekatan penting dalam pengembangan sistem informasi geografis modern. Dalam konteks representasi digital kota, Level of Detail 2 (LoD2) menjadi salah satu tingkatan model kota yang paling banyak diadopsi karena mampu menyajikan bentuk atap dan struktur geometris bangunan secara lengkap namun tetap efisien secara komputasional (Biljecki dkk., 2016). LoD2 mengacu pada model bangunan yang menyertakan volume dasar dan bentuk atap, serta

mendukung segmentasi semantik permukaan dinding dan atap. Model ini banyak digunakan karena memiliki keseimbangan antara kompleksitas geometri dan kelayakan visualisasi web (Kolbe, 2007). Format standar seperti CityGML dan CityJSON (*CityGML in JSON*) umum digunakan untuk merepresentasikan data ini (Abdul Rahman dkk., 2025).

Kemajuan teknologi web memungkinkan data spasial tiga dimensi dapat diakses secara *real-time* melalui *browser* tanpa memerlukan *plugin* tambahan. Perkembangan pustaka berbasis WebGL (*Web Graphics Library*) seperti CesiumJS menjadi contoh dalam hal ini, karena mampu mendukung *streaming* dan *rendering* data geospasial secara efisien dan interaktif (Y. Chen dkk., 2018). CesiumJS mendukung berbagai format visualisasi seperti glTF (*GL Transmission Format*) dan 3D Tiles, yang sangat cocok untuk menampilkan model LoD2 dari CityGML dengan struktur ubin spasial yang memungkinkan pemuatan bertahap berdasarkan sudut pandang kamera (Zhu dkk., t.t.). Proses ini juga dioptimalkan dengan teknik pemisahan area berdasarkan *grid* dan penggunaan input *Level of Detail* (LoD) (Q. Chen dkk., 2024).



Gambar II. 6 Contoh tampilan bangunan 3D dengan CesiumJS (Sandcastle Cesium, 2025)

Dalam implementasinya, data LoD2 dapat diperoleh dari berbagai sumber seperti pemetaan fotogrametri, citra *oblique*, atau pemindaian LiDAR. Data tersebut selanjutnya diproses ke dalam format interoperabel seperti CityGML,



lalu dikonversi menjadi 3D Tiles untuk ditampilkan menggunakan CesiumJS (Zhu dkk., 2022). Beberapa studi juga menerapkan teksturisasi otomatis berbasis citra untuk meningkatkan realisme model (Lunde, 2024), atau integrasi informasi dari model BIM melalui proses transformasi semantik dan geometrik (Xia & Rüppel, 2024).

Arsitektur sistem WebGIS (*Web-based Geographic Information System*) 3D biasanya disusun dalam tiga struktur utama. Pada struktur *backend*, digunakan basis data spasial seperti PostgreSQL (*Post Ingres Structured Query Language*) yang diperluas dengan PostGIS (PostgreSQL dan GIS) serta manajemen model CityGML dengan 3DcityDB (*3D City Database*) (Abdul Rahman dkk., 2019). Di sisi *middleware*, digunakan server API (*Application Programming Interface*) seperti Node.js atau Flask untuk melayani permintaan data dan pemrosesan. Sementara di sisi *frontend*, CesiumJS digunakan sebagai mesin visualisasi yang mendukung fitur interaktif seperti kueri objek, navigasi spasial, pengukuran geometri, hingga animasi waktu (Li dkk., 2024).

Berbagai studi implementasi menunjukkan keberhasilan sistem WebGIS 3D dalam berbagai konteks aplikatif. Kilsedar et al. dalam penelitiannya memanfaatkan CesiumJS untuk membuat simulasi banjir berbasis permukaan yang dapat disesuaikan ketinggiannya (Kilsedar dkk., 2019). Kemudian Liu et al. (2023) memanfaatkan sistem ini untuk mendukung pendidikan geospasial berbasis web (Liu dkk., 2023). Semua studi ini menunjukkan bahwa Cesium bukan hanya alat visualisasi, tetapi *platform* interaktif yang dapat diintegrasikan dengan data tematik dan mendukung partisipasi pengguna dalam pengambilan keputusan spasial.

## **II.7 Pajak Bumi dan Bangunan (PBB) Indonesia**

Pajak Bumi dan Bangunan (PBB) merupakan pajak kebendaan yang dikenakan atas objek berupa tanah dan/atau bangunan yang memberikan manfaat ekonomi bagi orang pribadi atau badan. Dalam sistem perpajakan Indonesia, pengelolaan PBB-P2 (Perdesaan dan Perkotaan) mengalami perubahan signifikan sejak diberlakukannya undang-undang atau UU No. 28

Tahun 2009 yang memberikan kewenangan pemungutan kepada pemerintah daerah (Republik Indonesia, 2009). Perubahan ini menuntut tersedianya data objek pajak yang valid, akurat, dan mutakhir agar penetapan nilai jual objek pajak (NJOP) dapat mencerminkan kondisi riil dan menjamin keadilan fiskal (Rizka, 2014).

Pajak Bumi dan Bangunan (PBB) diatur oleh Undang-Undang Nasional dan peraturan pelaksanaannya. UU No.12/1985 tentang PBB (sudah diubah oleh UU No.12/1994) menetapkan konsep umum PBB. Sejak berlakunya UU No.28/2009 (UU PDRD (Undang-Undang Pajak Daerah dan Retribusi Daerah)), PBB Perdesaan dan Perkotaan (PBB-P2) menjadi pajak daerah (dikelola Kabupaten/Kota), sementara PBB untuk sektor perkebunan, kehutanan, pertambangan dan lainnya (PBB-P3) tetap dipungut oleh pemerintah pusat (Republik Indonesia, 2009). Undang-Undang Hubungan Keuangan antara Pemerintah Pusat dan Pemerintahan Daerah atau UU HKPD No.1/2022 kemudian menetapkan bahwa tarif PBB-P2 maksimal 0,5% (Republik Indonesia, 2022). Di tingkat daerah, Kota Cimahi mengatur hal umum PBB melalui Peraturan Daerah atau Perda Kota Cimahi No.8/2023 (Pajak Daerah dan Retribusi Daerah), sedangkan rincian teknis penilaian NJOP dan pelaksanaan pungutan mengikuti petunjuk Peraturan Kementerian Keuangan (PMK/Perdirjen Pajak) dan peraturan wali kota terkait (Pemerintah Kota Cimahi, 2023).

NJOP (Nilai Jual Objek Pajak) adalah dasar pengenaan PBB didefinisikan sebagai harga rata-rata transaksi wajar atas objek pajak (atau nilai perolehan baru/pengganti jika tidak ada transaksi). NJOP dibagi menjadi dua komponen utama yaitu, NJOP Bumi dan NJOP Bangunan. Sesuai Pedoman Penilaian PBB (PMK No.85/2024), NJOP Bumi dihitung sebagai luas tanah (atau lahan) dikali NJOP per meter persegi (Nilai Indikasi Rata-rata/NIR di Zona Nilai Tanah). Demikian pula, NJOP Bangunan merupakan luas bangunan dikali NJOP bangunan per meter persegi (Republik Indonesia, 2024). NJOP bangunan per meter ditentukan menurut klasifikasi bangunan (jenis penggunaan dan kualitas konstruksi) yang diatur oleh daerah melalui Daftar

Biaya Komponen Bangunan (DBKB). DBKB adalah tabel standar biaya komponen bangunan (struktur, material, fasilitas) untuk tiap tipe bangunan. Sehingga,

$$\text{NJOP Bumi} = \text{Luas tanah} \times \text{NJOP per m}^2 \text{ (NIR pada zona nilai tanah) (II.5)}$$

$$\text{NJOP Bangunan} = \text{Luas bangunan} \times \text{NJOP per m}^2 \text{ (dari klasifikasi DBKB (II.6))}$$

Karena karakteristik bangunan (misalnya material, desain, umur) sudah tercakup dalam klasifikasi DBKB, faktor-faktor seperti jumlah jendela otomatis mempengaruhi NJOP melalui komponen material/finishing tersebut.

Dalam PBB tradisional (sebelum PBB dialihkan ke daerah) dikenal Nilai Jual Kena Pajak (NJKP) sebagai persentase tertentu dari NJOP setelah dikurangi *threshold*. Namun untuk PBB-P2 modern, rumus dasar adalah:

$$\text{NJKP} = (\text{NJOP}_{\text{Bumi}} + \text{NJOP}_{\text{Bangunan}}) - \text{NJOPTKP} \text{ (II.7)}$$

Dengan kata lain, nilai kena pajak (dasar pengenaan) umumnya diperoleh dengan mengurangi NJOPTKP (Nilai Jual Objek Pajak Tidak Kena Pajak) dari total NJOP. Sebagai ilustrasi (menurut contoh perhitungan PBB-P2), NJKP sering dihitung sebagai persentase 20–40% dari (NJOP – NJOPTKP), namun secara aturan daerah biasanya cukup dihitung langsung.

NJOPTKP adalah batas NJOP yang bebas pajak. Setiap wajib pajak mendapat potongan NJOPTKP satu kali setahun pada objek senilai terbesar. UU PDRD menetapkan batas bawah NJOPTKP minimal Rp10.000.000 per tahun pajak. Pemerintah pusat (Menkeu) mengeluarkan PMK khusus untuk menyesuaikannya. Berdasarkan PMK terbaru, dasar NJOPTKP nasional ditetapkan Rp12.000.000. Artinya, pada perhitungan PBB, nilai NJOP hingga Rp12 juta pertama tidak kena pajak. Daerah boleh menetapkan nilai lebih tinggi, tetapi tidak di bawah Rp10 juta (sebagaimana diatur UU PDRD).

UU HKPD (UU No.1/2022) menyatakan tarif PBB-P2 maksimal 0,5% dari NJKP. Pemerintah daerah menetapkan tarif spesifik melalui Perda pajak daerah. Sebagai contoh, banyak daerah menetapkan tarif sekitar 0,1–0,2% untuk rumah tinggal dan hingga 0,5% untuk properti komersial. (Sebelum HKPD, tarif semula ditetapkan maksimum 0,3% oleh UU PDRD (Republik Indonesia, 2022). Setiap Perda/Perwali daerah dapat menetapkan tarif lebih rendah atau memberikan insentif tertentu, misalnya keringanan PBB (seperti Perwali Cimahi No.1/2021 tentang pengurangan akibat kenaikan NJOP).

Dalam evaluasi nilai bangunan PBB, penilaian kualitas dilakukan melalui DBKB dan indeks bangunan. DBKB menguraikan komponen bangunan utama (struktur, dinding, atap, lantai, pintu/jendela, dll.) beserta biaya baku tiap komponen. Misalnya, tipe material jendela (kayu, aluminium, kaca) sudah tercermin dalam kategori DBKB dan koefisien bangunan. Pemerintah daerah juga dapat menetapkan koefisien khusus untuk variasi kualitas (usia bangunan, perawatan, tipe lantai, jenis atap, dan lain-lain) dalam pedoman penilaian.

Validasi data objek pajak menjadi kunci utama dalam perhitungan PBB yang tepat sasaran. Dalam konteks ini, konsep kadaster fiskal memainkan peran penting sebagai sistem pendataan bidang tanah dan bangunan yang fokus pada aspek ekonomi. Kadaster fiskal mencakup informasi teknis seperti luas tanah, luas bangunan, jumlah lantai, dan kelengkapan fasilitas. Ketidaksesuaian data di lapangan dengan database administratif seperti SISMIOP (Sistem Informasi Manajemen Objek Pajak) dapat menyebabkan NJOP yang tidak akurat dan menimbulkan ketidakadilan dalam besaran PBB yang ditetapkan (Hidayat & Wati, 2022).

Salah satu aspek fisik yang penting dalam validasi objek pajak adalah fasad bangunan. Fasad, sebagai tampilan luar bangunan, memuat elemen-elemen seperti jendela, pintu, balkon, dan ornamen yang mencerminkan kualitas dan kelas bangunan. Elemen seperti jumlah jendela, tipe kusen, material kaca, dan bentuk bukaan dapat menjadi indikator nilai arsitektural dan ekonomis suatu bangunan. Penelitian Mukhlisin menegaskan bahwa keberadaan banyak

jendela atau desain jendela yang kompleks dapat menaikkan klasifikasi bangunan, sehingga berdampak pada nilai NJOP yang lebih tinggi (Mukhlisin, 2021). Hal ini sejalan dengan Peraturan Dirjen Pajak PER-24/PJ/2016 yang menyatakan bahwa penilaian bangunan mempertimbangkan jenis bahan bangunan dan kelengkapan fasilitas.

Dalam mendukung akurasi validasi ini, teknologi pemodelan bangunan tiga dimensi dengan LoD 3 menjadi sangat krusial. LoD 3 mampu menampilkan bentuk bangunan secara lengkap, termasuk elemen fasad seperti jendela dan pintu, serta menghasilkan visualisasi yang mendekati kondisi riil. Dengan menggunakan LoD 3, dimensi jendela dapat diukur, jumlahnya dapat dihitung, dan distribusinya pada fasad dapat dianalisis secara spasial. Model ini juga dapat digunakan dalam audit digital terhadap objek pajak, meminimalkan kesalahan manual, dan memperkuat integrasi dengan sistem informasi fiskal berbasis geospasial (Arnold, 2012).

Perhitungan PBB sendiri mengikuti rumus:

$$\text{PBB} = \text{Tarif PBB} \times \text{NJKP} \quad (\text{II.8})$$

Dengan:

NJKP = persentase dari (NJOP - NJOPTKP).

Tarif PBB = persen tarif (maks 0,5%)

NJOP terdiri atas nilai tanah dan nilai bangunan, di mana nilai bangunan dihitung berdasarkan luas, material, jumlah lantai, serta kelengkapan seperti jendela dan balkon (Siahaan, 2018). Dengan kata lain, jumlah jendela bukan hanya menjadi elemen arsitektural, tetapi juga bagian dari parameter teknis yang memengaruhi klasifikasi dan perhitungan nilai bangunan dalam NJOP. Semakin banyak jendela atau semakin rumit desain fasad, maka semakin tinggi nilai bangunan yang dihitung, dan berimplikasi pada besarnya PBB terutang.

Berbagai studi lain juga menunjukkan bahwa efektivitas dan kontribusi PBB terhadap pendapatan asli daerah (PAD) sangat bergantung pada kualitas pendataan objek pajak. Kurangnya validasi dapat menyebabkan rendahnya

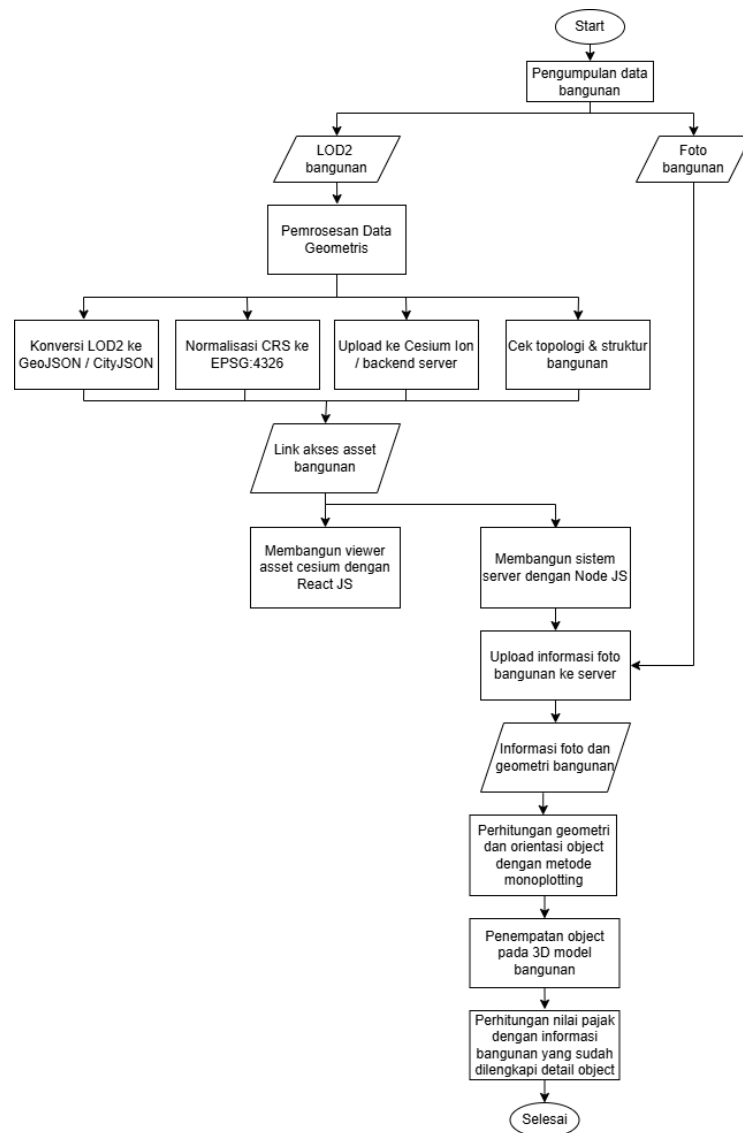
realisasi penerimaan dan ketidaksesuaian antara SPPT (Surat Pemberitahuan Pajak Terutang) dan nilai pasar bangunan (Dari, 2022). Oleh karena itu, strategi validasi data fisik objek pajak berbasis model 3D dan visualisasi fasad harus berjalan beriringan dengan pendekatan administratif dan sistem pembayaran digital (Riningsih, 2024).

### Bab III Metodologi

Penelitian ini menggunakan pendekatan rekayasa sistem (*system development*) dengan tahapan metodologis yang dirancang untuk mengintegrasikan teknik *monoplotting* berbasis citra fasad ke dalam sistem visualisasi bangunan 3D berbasis web. Secara umum, metodologi penelitian ini terdiri dari beberapa tahapan yang dijelaskan sebagai berikut,

#### III.1 Diagram Alir Penelitian

Berikut dijelaskan dalam grafik untuk diagram alir penelitian yang dilakukan:



Gambar III. 1 Diagram alir penelitian

### III.2 Tahapan Penelitian

Dijelaskan untuk tahapan penelitian yang dilakukan sebagai berikut:

#### III.2.1 Studi Literatur dan Analisis Kebutuhan

Dilakukan kajian terhadap pustaka yang relevan mengenai model LoD2, teknik *monoplotting*, visualisasi 3D berbasis web, aplikasi pemodelan bangunan dengan sistem web modern, serta kajian sistem pajak bumi dan bangunan di Indonesia dan tata cara perhitungannya.

#### III.2.2 Pengumpulan dan Pra-Pemrosesan Data

Tahap awal dalam penelitian ini difokuskan pada proses akuisisi dan persiapan data, baik dalam bentuk citra fasad bangunan maupun model bangunan tiga dimensi LoD2. Aktivitas dalam tahap ini meliputi:

##### 1. Mengumpulkan citra fasad bangunan yang dilengkapi dengan metadata (GPS, orientasi kamera) dari perangkat pemotret.

Pengambilan citra dilakukan di wilayah Kota Cimahi, pada beberapa bangunan yang memiliki karakteristik fasad yang dapat dijangkau secara visual dari permukaan tanah. Kamera yang digunakan adalah **iPhone 14 Pro**, yang memiliki sistem kamera dengan spesifikasi utama sebagai berikut:

- *Triple-camera system (Pro):*
  - o 48 MP (*Megapixel*) *Main camera* (*f/1.78 aperture*, *24mm*, *sensor-shift OIS (Optical Image Stabilization)*)
  - o 12 MP *Ultra Wide* (*f/2.2*, *13mm*)
  - o 12 MP *Telephoto* (*f/2.8*, *77mm*, *3x optical zoom*)
- *Focal Length (Main): 24mm (equivalent)*
- *Sensor Size: 1/1.28"*
- *GPS Metadata: Lokasi otomatis tersimpan dalam EXIF (Exchangeable Image File Format) (latitude, longitude, altitude)*

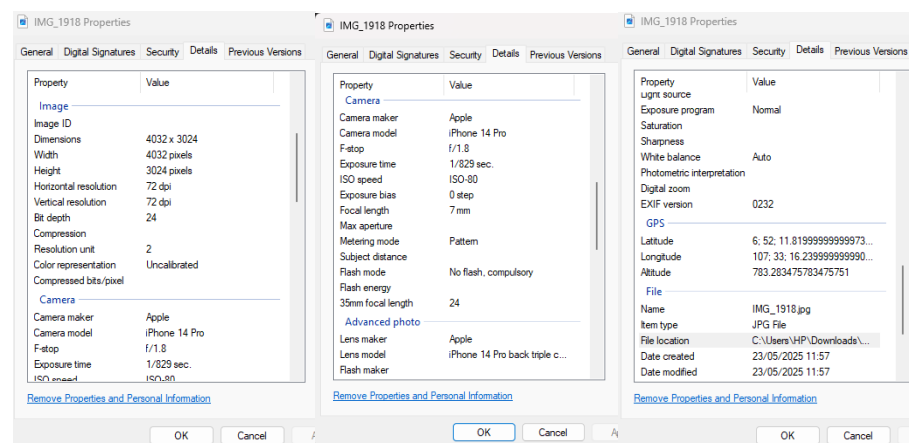
Karena keterbatasan jangkauan visual dari permukaan tanah, citra diambil dari berbagai sudut dan sisi bangunan yang memungkinkan, agar seluruh bagian fasad didapatkan secara menyeluruh. Beberapa sisi bangunan yang tidak terjangkau langsung, seperti bagian belakang atau



area tertutup oleh vegetasi, dapat menyebabkan keterbatasan dalam pengambilan data visual. Berikut merupakan foto dan detail dari foto yang diambil di lapangan:



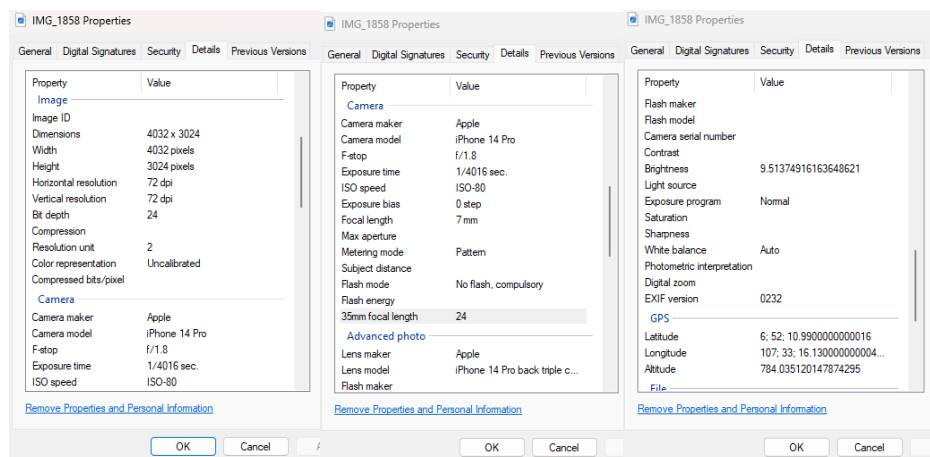
**Gambar III. 2** Contoh foto tampak depan bangunan



**Gambar III. 3** Metadata dari foto tampak depan bangunan



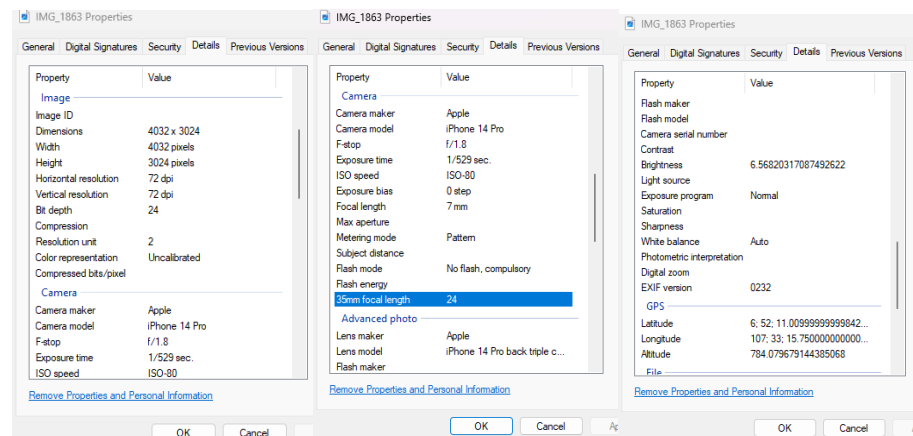
**Gambar III. 4** Contoh foto tampak samping bangunan



**Gambar III. 5** Metadata dari foto tampak samping bangunan

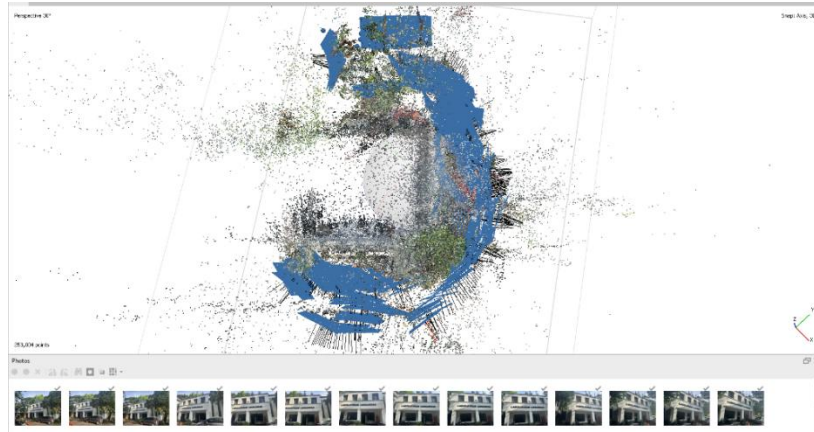


**Gambar III. 6** Contoh foto tampak belakang bangunan



**Gambar III. 7** Metadata dari foto tampak belakang bangunan

- Untuk memperoleh informasi posisi dan orientasi kamera yang diperlukan dalam teknik *monoplotting*, dilakukan proses referensi fotogrametri menggunakan perangkat lunak Agisoft Metashape. Dari proses ini diperoleh parameter posisi kamera (latitude, longitude, altitude) serta rotasi kamera dalam bentuk omega, phi, dan kappa (atau dalam sistem lain: *roll*, *pitch*, *yaw*). Informasi ini disimpan bersama nama file foto untuk digunakan dalam transformasi spasial.



**Gambar III. 8** Proses *Align Photos* dengan Agisoft Metashape (tampak atas)



**Gambar III. 9** Proses *Align Photos* dengan Agisoft Metashape (tampak samping)

Reference													
Cameras	Long. est	Lat. est	Alt. est (m)	Accuracy (m)	Error (m)	Omega est (°)	Phi est (°)	Kappa est (°)	Accuracy (°)	Error (°)	Projections	Error (pix)	
✓ IMG_2_...	107.554327	-6.869712	782.444911	10.000000	0.397746	-176.430	-25.781	133.677	10.000	220.663	1267	1.780	
✓ IMG_1_...	107.554441	-6.869938	782.498892	10.000000	0.878236	100.523	3.052	-1.779	10.000	90.558	2896	3.574	
✓ IMG_1_...	107.554467	-6.869934	782.504194	10.000000	1.377767	97.562	5.420	0.135	10.000	87.661	2905	2.385	
✓ IMG_1_...	107.554475	-6.869927	782.511635	10.000000	0.628330	99.027	3.682	1.980	10.000	89.102	2856	3.617	
✓ IMG_1_...	107.554463	-6.869936	782.512825	10.000000	1.603787	101.279	-3.719	0.016	10.000	91.434	2950	2.969	
✓ IMG_1_...	107.554457	-6.869941	782.513931	10.000000	1.654171	98.277	-4.796	2.541	10.000	88.530	2535	2.539	
✓ IMG_1_...	107.554455	-6.869945	782.523566	10.000000	1.456723	98.297	1.367	4.437	10.000	88.384	2420	3.713	
✓ IMG_1_...	107.554449	-6.869942	782.525547	10.000000	1.707134	97.316	18.161	-2.741	10.000	88.913	3094	2.571	
✓ IMG_1_...	107.554442	-6.869940	782.528142	10.000000	0.939310	101.590	2.607	0.064	10.000	91.607	3167	3.470	
✓ IMG_1_...	107.554473	-6.869930	782.529743	10.000000	0.460278	97.125	7.237	3.441	10.000	87.464	2674	3.101	
✓ IMG_1_...	107.554482	-6.869921	782.529875	10.000000	0.744115	101.049	3.164	0.991	10.000	91.082	2885	3.130	
✓ IMG_1_...	107.554479	-6.869924	782.533870	10.000000	0.451727	96.541	7.847	2.450	10.000	86.880	2940	2.830	
✓ IMG_1_...	107.554445	-6.869950	782.536666	10.000000	0.922813	97.757	20.298	-5.663	10.000	90.018	2045	2.381	
✓ IMG_1_...	107.554445	-6.869950	782.541300	10.000000	1.785895	98.208	14.835	-4.119	10.000	89.205	2152	2.312	
✓ IMG_1_...	107.554437	-6.869961	782.548571	10.000000	1.641104	98.777	-5.045	-1.222	10.000	89.030	2651	2.763	
✓ IMG_1_...	107.554437	-6.869961	782.548569	10.000000	1.553786	97.910	-4.081	-1.792	10.000	88.092	3160	2.092	
✓ IMG_1_...	107.554446	-6.869942	782.551496	10.000000	0.390789	99.224	-3.674	1.126	10.000	89.322	2869	3.179	
✓ IMG_1_...	107.554445	-6.869941	782.552381	10.000000	1.291531	101.802	1.874	-1.140	10.000	91.830	2888	3.405	
✓ IMG_1_...	107.554432	-6.869969	782.552847	10.000000	21.740408	100.917	-9.661	3.706	10.000	96.917	767	2.567	
✓ IMG_1_...	107.554452	-6.869943	782.558884	10.000000	1.806851	98.406	11.250	-2.810	10.000	88.984	3288	2.642	
✓ IMG_1_...	107.554439	-6.869959	782.565262	10.000000	2.109326	98.857	-1.942	-2.064	10.000	88.952	2917	3.061	

**Gambar III. 10** Data referensi yang diperoleh dari *Align Photos* dengan Agisoft Metashape



- Untuk Model bangunan tiga dimensi LoD2 diperoleh dalam format CityGML, yang menyajikan geometri dasar bangunan meliputi bentuk denah, tinggi bangunan dan atap, namun tanpa detail fasad. Untuk memastikan kompatibilitas dengan sistem visualisasi berbasis web (CesiumJS), model ini dikonversi ke format GeoJSON (*Geographical JavaScript Object Notation*) menggunakan alat bantu citygml-tools. Hasil konversi kemudian diunggah ke Cesium ion Assets untuk digunakan dalam visualisasi tiga dimensi.



**Gambar III. 11** Tampilan assets bangunan pada Cesium ion

- Melakukan pra-pemrosesan citra untuk memastikan kelayakan data citra dan LoD2. Pra-pemrosesan dilakukan untuk memastikan integritas dan kelayakan data sebelum digunakan dalam proses *monoplotting*, dengan langkah-langkah berikut:
  1. Memastikan setiap foto mencakup bidang fasad secara utuh.
  2. Mengecek keberadaan metadata geospasial (GPS), panjang fokus (*focal length*), dimensi gambar, dan parameter optik lainnya di dalam file EXIF.
  3. Menampilkan hasil konversi GeoJSON di platform CesiumJS. Kemudian melakukan inspeksi visual untuk memastikan bahwa

model dapat dimuat tanpa kesalahan geometri atau *error rendering* di Cesium.

### III.2.3 Pengembangan Sistem Berbasis Web

Dalam membangun sistem berbasis web, diperlukan tahap dalam pembuatan sistem *frontend* dan *backend* yang dijelaskan sebagai berikut,

#### a. *Frontend*

Tahap pengembangan *frontend* dilakukan dengan menggunakan kerangka kerja React.js yang dipadukan dengan pustaka visualisasi geospasial CesiumJS. Tujuan dari tahapan ini adalah untuk membangun sistem visualisasi interaktif berbasis web yang mampu menampilkan model bangunan tiga dimensi *Level of Detail 2* (LoD2) secara *real-time* di peramban (*browser*), tanpa ketergantungan pada perangkat lunak GIS atau CAD eksternal.

Komponen utama dalam arsitektur *frontend* adalah CesiumViewer, yang dirancang sebagai wadah utama tampilan peta 3D. Data bangunan dimuat dalam format GeoJSON dari layanan Cesium Ion dan ditampilkan di dalam viewer menggunakan API dari CesiumJS. Setiap entitas bangunan yang dimuat dianalisis secara spasial berdasarkan atribut ketinggian (*elevation*). Proses ini dilakukan dengan cara memeriksa nilai *height* dari setiap titik dalam geometri bangunan. Bangunan dengan elevasi mendekati nol diasumsikan berada di permukaan tanah dan diberikan warna putih solid, sedangkan bangunan dengan nilai elevasi lebih tinggi diberi warna yang berbeda. Pewarnaan ini diterapkan secara otomatis saat *rendering* berlangsung, dan ditujukan untuk meningkatkan kontras visual dalam membedakan struktur bangunan berdasarkan posisinya dalam ruang vertikal.

*Frontend* juga dirancang untuk menangani interaksi pengguna secara langsung. Setiap kali pengguna melakukan klik pada suatu bangunan dalam *viewer*, sistem akan membaca atribut uid dari objek

tersebut dan melakukan navigasi otomatis ke halaman detail bangunan (/zoom/:uid). Navigasi ini digunakan untuk mengarahkan pengguna ke tampilan interaktif selanjutnya yang memuat citra fasad, hasil anotasi, serta opsi untuk memproyeksikan jendela ke model 3D.

Seluruh proses pemanggilan data, pewarnaan elemen, dan manajemen interaksi pengguna dikelola melalui *React lifecycle hooks (useEffect)* dan API bawaan CesiumJS. Proses pembersihan *viewer* juga dilakukan secara eksplisit saat komponen dibongkar untuk menghindari kebocoran memori dan mempertahankan performa aplikasi selama *runtime*. Tahapan ini menjadi dasar visualisasi utama dalam sistem, dan berfungsi sebagai antarmuka awal sebelum pengguna beralih ke tahap input data dan anotasi fasad bangunan.

**Tabel III. 1** Penjelasan kode program untuk menampilkan objek 3D pada viewer berbasis ReactJS dan CesiumJS.

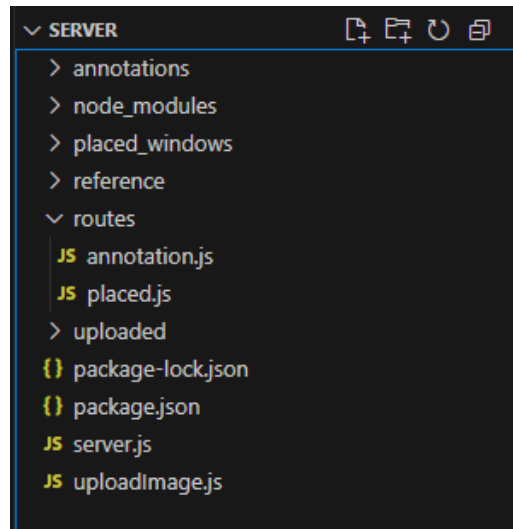
Bagian kode pada viewer React.js dan Cesium	Penjelasan fungsi kode
<code>Viewer(viewerRef.current)</code>	Membuat instansi peta 3D Cesium dalam elemen HTML ( <i>HyperText Markup Language</i> ) yang diacu oleh <code>viewerRef</code>
<code>IonResource.fromAssetId(3410848)</code>	Mengambil data GeoJSON bangunan dari Cesium Ion Asset
<code>GeoJsonDataSource.load(...)</code>	Memuat data GeoJSON ke dalam Cesium sebagai entitas visual

<code>Cartographic.fromCartesian(...)</code>	Mengonversi posisi Cartesian ke koordinat geospasial (lon, lat, height)
<code>entity.polygon.material = ...</code>	Mengatur warna entitas berdasarkan nilai elevasi (putih untuk tanah, merah untuk atap)
<code>ScreenSpaceEventHandler.setInputAction(...)</code>	Menangani <i>event</i> klik pada objek dan membaca properti untuk navigasi ke halaman detail
<code>navigate (/zoom/\${uid})</code>	Mengarahkan pengguna ke halaman /zoom/:uid berdasarkan properti bangunan yang diklik
<code>viewer.destroy()</code> <code>clickHandler.destroy()</code>	Membersihkan <i>viewer</i> dan <i>handler</i> saat komponen React dihentikan untuk mencegah kebocoran memori.

### ***b. Backend***

Pengembangan sisi *backend* dalam penelitian ini dilakukan dengan menggunakan Node.js dan kerangka kerja Express.js. Tahapan ini bertujuan untuk membangun layanan RESTful API (*Representational State Transfer Application Programming Interface*) yang mampu menangani alur data antara antarmuka *frontend* dengan sistem penyimpanan dan pemrosesan spasial. *Backend* dirancang untuk mendukung tiga fungsi utama dalam system yaitu, pengunggahan citra fasad bangunan, penyimpanan dan manajemen file hasil anotasi fasad dalam format GeoJSON, serta pemrosesan dan penyajian hasil konversi anotasi menjadi objek jendela 3D pada model bangunan LoD2.





**Gambar III. 12** Tampilan struktur file dalam server

File utama `server.js` dikembangkan sebagai *entry point server* yang mengelola inisialisasi aplikasi, pemanggilan *middleware*, konfigurasi rute, dan penyajian file statis. *Server* dijalankan pada port 5000 dan menggunakan *middleware cors()* untuk mengizinkan permintaan lintas domain, khususnya dari aplikasi *frontend* berbasis React. Selain itu, `express.json()` diaktifkan sebagai *middleware* untuk memproses *payload* yang dikirim dalam format JSON, seperti metadata anotasi atau daftar file yang akan diolah lebih lanjut.

Struktur rute *backend* dibagi menjadi tiga modul utama, masing-masing menangani proses unggah gambar (`uploadImage.js`), anotasi (`routes/annotation.js`), dan hasil proyeksi jendela (`routes/placed.js`). Setiap rute dipanggil dengan awalan *namespace /api* untuk menjaga konsistensi penamaan *endpoint*. Server juga dikonfigurasi untuk menyajikan file statis secara langsung dari tiga direktori utama yaitu, `uploaded/` untuk gambar, `annotations/` untuk hasil anotasi, dan `placed/` untuk objek jendela yang telah diproyeksikan ke bangunan 3D.

Seluruh alur *backend* ini dirancang agar dapat berkomunikasi secara asinkron dengan *frontend* melalui permintaan `fetch()`, sehingga proses seperti unggah foto, tampilkan anotasi, atau render hasil jendela dapat dilakukan secara real-time dalam satu kesatuan sistem berbasis web.

**Tabel III. 2** Penjelasan kode program pada server

Bagian kode pada server	Penjelasan fungsi
<code>const express = require("express");</code>	Mengimpor pustaka Express untuk membuat server <i>backend</i> .
<code>const cors = require("cors");</code>	Mengimpor middleware CORS agar <i>frontend</i> dapat mengakses API tanpa kendala lintas origin.
<code>const multer = require("multer");</code>	Digunakan untuk menangani pengunggahan file.
<code>const fs = require("fs");</code>	Digunakan untuk operasi file sistem, seperti membaca atau menyimpan file.
<code>const path = require("path");</code>	Digunakan untuk mengelola path file/direktori secara <i>cross-platform</i> .
<code>const app = express();</code>	Membuat instance aplikasi <i>Express</i> .
<code>const PORT = 5000;</code>	Menetapkan port server (5000) yang akan digunakan saat menjalankan server lokal.
<code>app.use(cors());</code>	Mengaktifkan middleware CORS agar API dapat diakses dari frontend React.
<code>app.use(express.json()) ;</code>	Middleware untuk parsing data JSON yang dikirim dari frontend.

<code>app.use("/uploaded", express.static(...))</code>	Menyajikan file gambar hasil unggah sebagai file statis yang dapat diakses dari <i>browser</i> .
<code>app.use("/annotations", express.static(...))</code>	Menyajikan file GeoJSON hasil anotasi pengguna.
<code>app.use("/placed", express.static(...))</code>	Menyajikan file GeoJSON hasil penempatan jendela 3D dari proses <i>monoplotting</i> .
<code>const uploadRouter = require("./uploadImage" );</code>	Memuat rute pengunggahan gambar dari file eksternal ( <code>uploadImage.js</code> ).
<code>app.use("/api", uploadRouter);</code>	Input rute pengunggahan di bawah namespace <code>/api</code> .
<code>const annotationRouter = require("./routes/annot ation");</code>	Memuat rute untuk menyimpan dan membaca anotasi.
<code>app.use("/api", annotationRouter);</code>	Input rute anotasi ke dalam namespace <code>/api</code> .
<code>const placeWindowRouter = require("./routes/place d");</code>	Input rute anotasi ke dalam namespace <code>/api</code> .
<code>app.use("/api", placeWindowRouter);</code>	Input rute hasil proyeksi jendela ke dalam namespace <code>/api</code> .
<code>app.listen(PORT, () =&gt; { ... });</code>	Menjalankan server Express dan menampilkan log saat server aktif di <code>http://localhost:5000</code> .

### III.2.4 Visualisasi dan Input Informasi Bangunan

Pada tahap ini, sistem dirancang untuk memungkinkan pengguna melakukan input data fasad bangunan serta memvisualisasikan dan memproyeksikan hasil anotasi secara langsung ke dalam model 3D bangunan LoD2. Proses dimulai dari pemilihan bangunan yang ingin dianalisis. Parameter uid dari URL (*Uniform Resource Locator*) digunakan

untuk menampilkan hanya satu objek bangunan tertentu yang dimuat dari sumber data GeoJSON melalui layanan Cesium Ion. Model bangunan ditampilkan menggunakan pustaka CesiumJS dalam bentuk tampilan 3D, dan setiap bangunan yang dimuat akan difokuskan kamera secara otomatis menggunakan fungsi `flyTo`.

Setelah bangunan berhasil divisualisasikan, pengguna dapat melakukan proses input data berupa foto-foto fasad bangunan yang diunggah melalui antarmuka pengguna. Foto-foto tersebut kemudian dianotasi secara manual menggunakan komponen khusus bernama `ImageAnnotator`, yang memungkinkan pengguna menggambar poligon pada area bukaan (jendela atau pintu) langsung di atas citra. Hasil anotasi disimpan dalam format GeoJSON dan dilengkapi dengan informasi tambahan seperti ukuran bukaan, orientasi sisi, dan koordinat pusat citra. File GeoJSON ini kemudian digunakan dalam tahap transformasi dan proyeksi geometri ke permukaan fasad 3D.

Setelah anotasi selesai dilakukan, sistem menyediakan fitur pemetaan hasil ke model bangunan 3D. Fungsi `handlePlace` akan membaca data hasil anotasi dan melakukan proyeksi ke permukaan bangunan berdasarkan informasi geometri dan sisi bangunan yang telah ditentukan sebelumnya. Proyeksi mempertimbangkan skala berdasarkan ukuran asli jendela (dalam satuan meter) dan mengonversinya ke skala spasial peta dalam sistem koordinat *European Petroleum Survey Group* atau EPSG:4326.

Sistem juga menyediakan fitur tambahan berupa pemetaan dan pelabelan sisi bangunan (`handleShowSides`), yang secara otomatis mengekstraksi bidang vertikal (permukaan  $z=0$ ) dari geometri model LoD2. Semua interaksi dilakukan secara dinamis melalui permintaan HTTP (*HyperText Transfer Protocol*) menggunakan `fetch` dari sisi klien ke server Node.js, sehingga proses input, anotasi, dan visualisasi dapat dilakukan secara real-time langsung dari *browser* tanpa perangkat lunak eksternal.

**Tabel III. 3** Penjelasan kode program pada visualisasi dan input informasi bangunan

Bagian kode pada sistem visualisasi informasi bangunan	Penjelasan fungsi
<code>useParams()</code>	Mengambil uid dari URL agar hanya satu bangunan LoD2 yang ditampilkan di viewer.
<code>Viewer(viewerRef.current)</code>	Membuat instansi Cesium viewer untuk menampilkan peta 3D dan objek-objek bangunan.
<code>GeoJsonDataSource.load(resource)</code>	Memuat data GeoJSON dari Cesium Ion dan menampilkannya sebagai bangunan 3D.
<code>handleUpload()</code>	Mengunggah foto-foto fasad ke backend dengan <i>endpoint</i> <code>/api/upload/:uid</code> .
<code>loadImages()</code>	Mengambil daftar foto dari server ( <code>/api/images/:uid</code> ) untuk ditampilkan dan dianotasi.
<code>loadAnnotations()</code>	Mengambil daftar file GeoJSON hasil anotasi dari <i>backend</i> ( <code>/api/annotations/:uid</code> ).
<code>loadPlacedWindows()</code>	Mengambil daftar hasil jendela yang telah diproyeksikan ke 3D ( <code>/api/placed_windows/:uid</code> ).
<code>handleShowSides()</code>	Menampilkan sisi-sisi dinding ( $z=0$ ) dari bangunan dalam bentuk <i>polygon</i> berlabel.

<code>handlePlace(viewer, placedWindowUrl, uid)</code>	Menempatkan jendela ke fasad berdasarkan file .geojson hasil proyeksi.
<code>setSelectedImage(...)</code>	Mengatur gambar terpilih untuk ditampilkan dalam mode <i>preview</i> atau mode anotasi.
<code>ImageAnnotator</code>	Komponen untuk menggambar <i>polygon</i> jendela di atas foto fasad dan menyimpannya sebagai GeoJSON.
<code>ShowAnnotation</code>	Menampilkan kembali gambar dan anotasi <i>polygon</i> jendela hasil sebelumnya untuk keperluan validasi.

### III.2.5 Sistem Anotasi atau Penggambaran Poligon informasi Fasad pada Citra 2D

Untuk mendukung ekstraksi bukaan fasad secara manual, penelitian ini mengembangkan sebuah modul anotasi citra yang memungkinkan pengguna menggambarkan poligon berbentuk persegi panjang secara interaktif pada foto fasad bangunan. Modul ini diimplementasikan sebagai komponen React bernama `ImageAnnotator`, dengan dukungan pustaka `React-Konva` untuk menggambar elemen grafis 2D berbasis kanvas, serta `exifr.js` untuk mengekstrak metadata spasial dari gambar.

Ketika sebuah gambar dimuat ke dalam komponen, sistem secara otomatis membaca metadata GPS, orientasi, dan informasi kamera dari file gambar menggunakan pustaka `exifr`. Informasi seperti latitude, longitude, altitude, focal length, dan image dimension akan digunakan dalam proses konversi anotasi dari sistem piksel ke sistem koordinat spasial dunia nyata (WGS84).

Pengguna melakukan anotasi dengan cara klik dan seret untuk membuat persegi panjang di atas gambar. Setiap anotasi disimpan sebagai *array* koordinat piksel. Sistem kemudian menghitung skala spasial (meter per piksel) menggunakan parameter kamera, panjang fokus, dan asumsi jarak kamera-ke-fasad. Skala ini digunakan untuk mengubah posisi poligon menjadi koordinat geografis 3D (longitude, latitude, altitude), yang kemudian disimpan sebagai objek GeoJSON berisi properti geometri dan informasi ukuran jendela (dalam meter).

Sistem mengubah posisi anotasi (yang semula dalam satuan piksel gambar) ke dalam sistem koordinat geospasial dunia nyata (longitude, latitude, altitude). Proses konversi ini menggunakan prinsip dasar perspektif kamera dan skala spasial, dengan asumsi bahwa permukaan bangunan sejajar dengan bidang gambar dan memiliki jarak tetap terhadap kamera. Penjelasan langkah konversi dijelaskan dengan:

#### 1. Ekstraksi Parameter Kamera dan Gambar

Metadata yang diambil dari citra melalui EXIF meliputi:

- latitude, longitude → posisi pusat kamera
- GPSTimeStamp → ketinggian kamera
- FocalLength (dalam mm) → panjang fokus lensa
- ExifImageWidth → lebar gambar (piksel)
- sensorWidth (5.6 mm untuk iPhone wide camera) → lebar sensor kamera secara fisik

#### 2. Konversi Resolusi Sensor ke Skala Ruang

Dihitung berapa milimeter per piksel (ukuran piksel dalam dunia nyata) (GeoCue Group, 2025):

$$mmPerPixel = \frac{sensorWidth}{imageWidth} \quad \text{III.1}$$

#### 3. Perkiraan jarak pengambilan foto dengan kamera ke fasad bangunan diasumsikan tetap,

$$distance = 10 \text{ meter} \quad \text{III.2}$$

4. Konversi ke meter per piksel di dunia nyata

Menggunakan rumus perspektif kamera pinhole (Park, 2025):

$$meterPerPixel = \frac{mmPerPixel \times distance}{FocalLength} : 1000 \quad \text{III.3}$$

5. Transformasi Piksel ke Koordinat Geospasial

Dengan mengasumsikan bahwa citra berada pada bidang tegak lurus permukaan bumi, konversi dari piksel ke posisi dunia nyata dilakukan dengan menggeser posisi tengah gambar secara proporsional (Roukema & Lew, 2004):

$$dx = x - \frac{width}{2} \quad \text{III.4}$$

$$dy = y - \frac{height}{2} \quad \text{III.5}$$

$$lon = longitude + \frac{dx \times meterPerPixel}{111320 \times \cos \left( latitude \times \frac{\pi}{180} \right)} \quad \text{III.6}$$

$$lat = latitude - \frac{dy \times meterPerPixel}{110540} \quad \text{III.7}$$

Nilai pembagi 111320 dan 110540 digunakan untuk mengubah meter ke derajat lintang dan bujur (konversi berdasarkan pendekatan ellipsoid WGS84).

6. Pembuatan Poligon GeoJSON

Setiap anotasi (persegi panjang) dikonversi ke empat titik sudut dalam bentuk koordinat [lon, lat, alt], kemudian dibuat dalam struktur GeoJSON dengan properti tambahan seperti:

1. widthInMeters
2. heightInMeters
3. meterPerPixel
4. buildingId
5. ImageName



Hasil anotasi atau gambar poligon dapat disimpan dengan menekan tombol "Save Annotation", yang akan mengirimkan data dalam format GeoJSON ke *backend* melalui *endpoint* API sesuai *buildingId* dan *imageName*. Anotasi yang telah disimpan nantinya digunakan dalam proses pemetaan (*monoplotting*) ke model 3D bangunan.

**Tabel III.4** Penjelasan kode program pada proses anotasi atau penggambaran *polygon* pada citra

Bagian kode pada sistem anotasi	Penjelasan fungsi
<code>useImage(imageUrl)</code>	Memuat citra secara asinkron ke dalam kanvas menggunakan Konva
<code>exifr.parse(blob, {...})</code>	Mengekstrak metadata EXIF (GPS, orientasi kamera, ukuran gambar) dari file citra
<code>stageRef.current.getPointerPosition()</code>	Mengambil posisi kursor untuk menentukan titik awal dan akhir saat menggambar persegi panjang
<code>setRects([...rects, newRect])</code>	Menyimpan anotasi persegi panjang yang telah digambar ke dalam daftar anotasi aktif
<code>convertToGeoJSON()</code>	Mengubah setiap anotasi (dalam piksel) menjadi fitur GeoJSON berdasarkan posisi GPS dan skala spasial
<code>pixelToLatLonAlt(pt)</code>	Fungsi konversi dari posisi piksel ke koordinat longitude, latitude, dan altitude dunia nyata
<code>meterPerPixel</code>	Menghitung rasio skala spasial berdasarkan <i>focal length</i> dan ukuran sensor kamera

<code>handleSave()</code>	Mengirim GeoJSON anotasi ke backend melalui API dan menyimpannya di direktori server
<code>&lt;Stage&gt;</code> , <code>&lt;Layer&gt;</code> , <code>&lt;Rect&gt;</code>	Komponen dari Konva untuk menggambar gambar dan persegi panjang anotasi pada kanvas

### III.2.6 Visualisasi Hasil Anotasi pada 2D dan 3D

Tahap visualisasi hasil anotasi bertujuan untuk menampilkan ulang poligon jendela yang sebelumnya telah dibuat oleh pengguna pada citra fasad, sekaligus menyediakan fungsi untuk memproyeksikannya ke model bangunan 3D. Untuk keperluan ini, dikembangkan sebuah komponen React bernama `ShowAnnotation` yang menangani pembacaan file citra dan file anotasi dalam format GeoJSON, serta merekonstruksi ulang posisi poligon pada bidang gambar asli.

Proses dimulai dengan memuat citra dan metadata spasial yang tersimpan dalam file EXIF menggunakan pustaka `exifr`. File anotasi dalam format GeoJSON kemudian diambil dari server dan diproses untuk dikonversi kembali ke sistem koordinat piksel gambar, menggunakan metode transformasi koordinat geospasial ke citra. Transformasi ini didasarkan pada asumsi bahwa citra memiliki pusat di tengah, dan setiap titik koordinat lon/lat dihitung jaraknya terhadap pusat berdasarkan derajat lintang dan bujur, lalu dikonversi ke piksel menggunakan nilai `meterPerPixel` yang sebelumnya dihitung pada saat anotasi.

Setiap poligon dalam GeoJSON dikonversi ke bentuk persegi panjang berdasarkan nilai minimum dan maksimum dari koordinat x dan y hasil konversi. Hasil persegi panjang kemudian digambarkan ulang di atas gambar asli menggunakan pustaka `React-Konva`, dengan warna hijau limau transparan dan garis tepi hijau untuk menandai jendela yang telah dianotasi.

Pada tahap visualisasi hasil anotasi, sistem perlu menampilkan ulang bentuk jendela yang sebelumnya telah dianotasi dan disimpan dalam format GeoJSON. Namun, karena file GeoJSON menyimpan posisi jendela dalam sistem koordinat geospasial (longitude, latitude), sementara citra fasad berada dalam sistem koordinat piksel (x, y), maka perlu dilakukan transformasi dari koordinat dunia nyata kembali ke koordinat citra. Proses ini bersifat invers dari proses *monoplotting*.

Perhitungan transformasi ini didasarkan pada prinsip proyeksi ekuidistan lokal, di mana diasumsikan bahwa:

1. Kamera menghadap tegak lurus ke dinding fasad bangunan.
2. Gambar memiliki pusat optik di tengah (pusat piksel mewakili posisi GPS kamera).
3. Skala spasial antara dunia nyata dan gambar bersifat linier (tetap), dan diwakili oleh parameter *meterPerPixel*.

Untuk setiap titik koordinat (longitude, latitude) dari polygon jendela, dihitung selisih posisi relatif terhadap pusat gambar (*centerLon*, *centerLat*) dalam satuan meter menggunakan konversi geodetik lokal (Booth, 2025):

Selisih horizontal (bujur):

$$\Delta x = (lon - centerLon) \times 111320 \times \cos(centerLat) \quad \text{III.8}$$

Selisih vertikal (lintang):

$$\Delta y = (centerLat - lat) \times 110540 \quad \text{III.9}$$

Setelah diperoleh selisih dalam meter, posisi tersebut dikonversi ke sistem piksel menggunakan skala spasial:

$$x = \frac{imageWidth}{2} + \frac{\Delta x}{meterPerPixel} \quad \text{III.10}$$

$$y = \frac{imageHeight}{2} + \frac{\Delta y}{meterPerPixel} \quad \text{III.11}$$

Sistem memproyeksikan titik anotasi kembali ke gambar dengan asumsi bahwa titik pusat citra adalah posisi GPS kamera, dan setiap meter di dunia nyata dikonversi ke sejumlah piksel dalam citra menggunakan nilai  $\text{meterPerPixel}$ .

Komponen ini juga menyediakan fungsi `Place All Windows`, yang akan mengirimkan kembali data GeoJSON ke server melalui endpoint `/api/placeAnnotationOnBuilding`. Server akan memproses data tersebut dan memproyeksikan poligon jendela ke fasad bangunan 3D yang ditampilkan di viewer Cesium.

**Tabel III. 5** Penjelasan kode program pada proses visualisasi hasil anotasi atau penggambaran poligon

Bagian kode pada sistem visualisasi poligon	Penjelasan fungsi
<code>useImage(imageUrl)</code>	Memuat citra fasad ke dalam elemen canvas berbasis Konva
<code>exifr.parse(imgBlob, { gps: true })</code>	Mengekstrak metadata posisi kamera (latitude, longitude) dari EXIF
<code>setScale(...)</code>	Menyesuaikan ukuran tampilan gambar agar sesuai dengan area kanvas
<code>latLonToPixel()</code>	Mengonversi titik koordinat GeoJSON (lon, lat) ke koordinat piksel gambar
<code>pixelRects = geo.features.map(...)</code>	Membentuk ulang setiap <i>polygon</i> GeoJSON menjadi persegi panjang berdasarkan distribusi koordinat piksel
<code>&lt;KonvaImage /&gt;</code>	Menampilkan citra sebagai latar belakang pada canvas

<code>&lt;Rect /&gt;</code>	Menampilkan anotasi jendela sebagai persegi panjang semi-transparan di atas citra
<code>handlePlaceOnBuilding()</code>	Mengirim ulang seluruh file GeoJSON ke server untuk diproyeksikan ke model 3D bangunan
<code>setRawGeoJSON(geo)</code>	Menyimpan objek GeoJSON asli yang akan dikirim ulang ke <i>backend</i> saat diperlukan

### III.2.7 Penerapan Teknik *Monoplotting*

Dalam sistem ini, teknik *monoplotting* digunakan untuk memetakan hasil anotasi jendela dari citra 2D ke model bangunan 3D. Setiap anotasi jendela berupa poligon yang telah dikonversi ke koordinat geospasial (longitude, latitude, dan estimasi tinggi) akan ditempatkan secara geometris pada permukaan dinding bangunan tiga dimensi (LoD2) di lingkungan CesiumJS.

Proses dimulai dengan memuat metadata kamera dari file `reference_lab.txt`, yang berisi parameter posisi dan orientasi kamera saat pengambilan gambar, berupa nilai omega (rotasi sumbu X), phi (rotasi sumbu Y), dan kappa (rotasi sumbu Z). Ketiga nilai ini dikonversi ke radian dan digunakan untuk membentuk rotasi spasial objek dalam bentuk Heading-Pitch-Roll (HPR), yang merepresentasikan arah hadap objek ketika diproyeksikan ke dinding.

Sistem kemudian mencari dinding (wall) bangunan terdekat berdasarkan rata-rata posisi anotasi (centroid) dan membandingkannya dengan pusat geometri masing-masing segmen dinding dari data `building_with_parts.geojson`. Sistem memilih dinding terdekat dan menghitung heading (arah hadap) dari vektor sisi terpanjang dinding tersebut menggunakan fungsi `computeHeadingFromWall`. Heading ini dikoreksi dengan nilai offset untuk mencocokkan orientasi visual.

Setelah arah dan posisi dasar diketahui, anotasi jendela kemudian diposisikan pada ketinggian (Z) yang dihitung berdasarkan urutan baris (row) dari bawah ke atas. Lebar dan tinggi setiap jendela diperoleh dari properti `widthInMeters` dan `heightInMeters` hasil perhitungan skala `meterPerPixel` saat anotasi dilakukan.

Anotasi diproyeksikan menjadi box 3D yang ditampilkan flush (menempel) pada dinding bangunan. Penyesuaian dilakukan dengan menggeser objek ke arah dinding (mengikuti arah heading) menggunakan vektor dorong (`pushVector`) yang dihitung dari lebar jendela dan arah normal permukaan dinding. Semua transformasi ini dilakukan melalui kombinasi matriks transformasi rotasi dan translasi dalam ruang 3D menggunakan `Cesium.Matrix4` dan `Cesium.HeadingPitchRoll`, sehingga jendela divisualisasikan dengan presisi terhadap posisi dan orientasi sebenarnya.

### III.2.8 Perhitungan Objek Pajak

Dalam sistem ini, perhitungan Pajak Bumi dan Bangunan (PBB) dilakukan secara otomatis dengan menggabungkan data spasial digital (berupa file GeoJSON) dan parameter NJOP berdasarkan regulasi daerah dan nasional. Proses perhitungan dimulai dengan pengambilan data jendela yang telah ditempatkan pada masing-masing bangunan. Data ini diperoleh melalui *endpoint* API `http://localhost:5000/api/placed_windows/{buildingId}` yang mengembalikan daftar file GeoJSON jendela untuk satu bangunan tertentu.

Setiap file GeoJSON kemudian diunduh dan dianalisis untuk menghitung total jumlah jendela yang terdapat pada bangunan tersebut. Jumlah jendela digunakan sebagai indikator kualitas bangunan yang berdampak terhadap nilai jual bangunan. Dalam implementasi ini, apabila jumlah jendela  $\geq 10$ , maka sistem memberikan bonus kualitas sebesar 10%; jika jumlahnya antara 5–9 jendela, maka diberikan bonus sebesar 5%. Bonus kualitas ini digunakan untuk menyesuaikan nilai NJOP bangunan secara proporsional, sebagaimana dicerminkan dalam praktik penilaian bangunan berdasarkan komponen material dan fasilitas (mengacu pada DBKB).

Selanjutnya, sistem mengambil nilai dasar properti berupa NJOP bangunan per meter persegi dan NJOP tanah per meter persegi. Jika pengguna tidak memberikan input langsung, maka sistem akan menggunakan nilai rata-rata Cimahi sebagai *fallback* — masing-masing Rp1.200.000/m<sup>2</sup> untuk bangunan dan Rp900.000/m<sup>2</sup> untuk tanah. Luas tanah juga dapat diinput secara manual atau menggunakan nilai standar (*default*) jika kosong.

Luas bangunan dihitung otomatis berdasarkan data bentuk dasar bangunan (*footprint*) yang disimpan dalam GeoJSON. Sistem memanggil fungsi `calculateFootprintAreaFromGeoJSON` untuk mendapatkan luas permukaan bangunan secara horizontal (pada sumbu  $z=0$ ). Setelah seluruh data dihimpun, sistem menghitung untuk perhitungan pajak dengan persamaan II.6 hingga II.8. Sistem kemudian menerapkan potongan NJOPTKP sebesar Rp15.000.000 sebagai pengurang nilai yang tidak dikenakan pajak, dan menghitung Nilai Jual Kena Pajak (NJKP)

berdasarkan tarif 20% untuk objek di bawah Rp1 miliar, dan 40% untuk objek di atasnya. Tarif ini mencerminkan pendekatan progresif sebagaimana diatur dalam Undang-Undang No.1 Tahun 2022 tentang Hubungan Keuangan antara Pemerintah Pusat dan Daerah (UU HKPD) dan Perda Kota Cimahi No.8 Tahun 2023. PBB terutang dihitung sebagai 0,5% dari NJKP. Nilai akhir yang ditampilkan sistem mencerminkan kewajiban pajak atas objek yang diperiksa, dengan mempertimbangkan elemen-elemen spasial (luas, bentuk bangunan) dan administratif (tarif NJOP, NJOPTKP, bonus kualitas).



## Bab IV Hasil dan Analisis Penelitian

Bab ini menyajikan hasil dan analisis dari sistem yang telah dikembangkan untuk memperkaya model bangunan tiga dimensi dengan detail fasad, khususnya informasi mengenai bukaan jendela. Analisis dilakukan untuk mengevaluasi kinerja sistem dalam menerapkan metode monoplottting fotogrametri guna meningkatkan tingkat kedetailan model bangunan LoD2, serta menilai efektivitas sistem dalam menghitung estimasi pajak jendela secara otomatis berdasarkan hasil proyeksi spasial.

### IV.1 Tampilan Bangunan LoD2 pada CesiumJS

Gambar IV.1 menunjukkan hasil visualisasi bangunan tiga dimensi (3D) LoD2 pada kawasan studi di Kota Cimahi, yang telah diproses dan divisualisasikan menggunakan platform CesiumJS. Dalam visualisasi ini, objek bangunan ditampilkan dalam bentuk model blok ekstrusi berdasarkan data geospasial, dan ditampilkan di atas citra satelit resolusi tinggi untuk memperoleh informasi lokasi dan lingkungan sekitar.

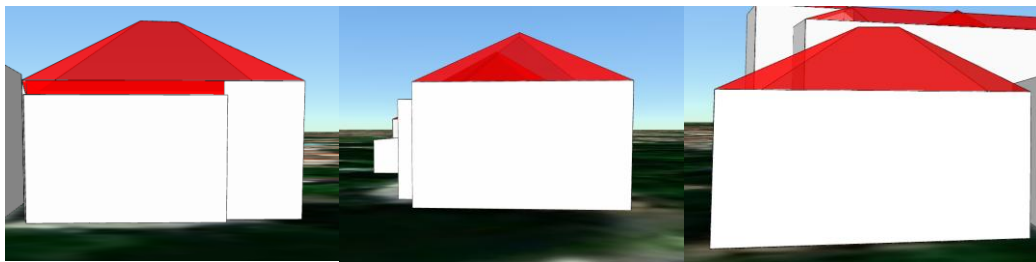


**Gambar IV. 1** Tampilan model bangunan 3D pada Cesium ion secara keseluruhan

Setiap bangunan dimodelkan sebagai blok geometri 3D sederhana berdasarkan footprint dan tinggi bangunan yang tersedia dalam format GeoJSON, kemudian diwarnai untuk membedakan elemen struktur dengan warna putih

merepresentasikan dinding dan warna merah sebagai representasi atap. Pewarnaan ini tidak hanya memperkuat persepsi spasial terhadap struktur vertikal dan horizontal bangunan, tetapi juga mendukung segmentasi semantik yang diperlukan dalam proses anotasi fasad dan pemetaan elemen jendela.

Pemodelan LoD2 dalam konteks penelitian ini dilakukan sebagai baseline untuk meningkatkan semantik bangunan tanpa harus membangun LoD3 secara penuh. Dengan menggunakan teknik *monoplotting*, detail bukaan fasad seperti jendela ditambahkan berdasarkan anotasi manual pada citra fasad bangunan. Walaupun dalam gambar belum terlihat langsung elemen jendela yang diproyeksikan, namun gambar ini merupakan tahap awal dari integrasi sistem visualisasi 3D, di mana hasil proyeksi jendela akan ditempatkan secara spasial pada bidang dinding model berdasarkan informasi hasil anotasi dan parameter orientasi kamera.



**Gambar IV. 2** Tampilan model bangunan 3D pada Cesium Ion difokuskan pada bangunan utama yang dianalisis, dengan visualisasi yang menampilkan sisi depan, samping, dan belakang secara berurutan.

Visualisasi yang ditunjukkan pada gambar IV.1 dan IV.2 merepresentasikan tahapan awal dalam *pipeline* pemrosesan sistem, yakni model dasar bangunan yang akan dianalisis lebih lanjut menggunakan teknik *monoplotting*. Pada tahap ini, belum terdapat detail fasad seperti jendela atau pintu yang terintegrasi ke dalam model. Namun, bangunan telah siap untuk dianotasi melalui modul sistem yang memungkinkan pengguna menandai elemen fasad pada citra 2D, lalu memproyeksikannya ke permukaan model 3D.

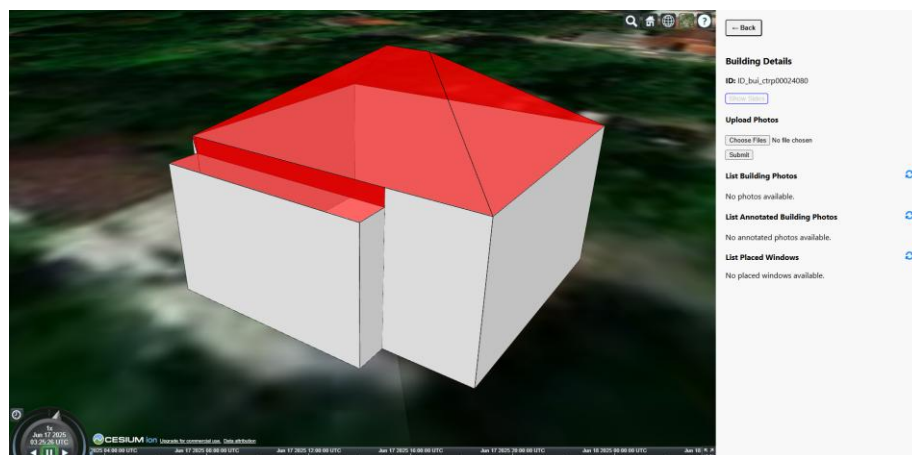
Dari sisi visual dan struktur, model yang ditunjukkan telah memenuhi kebutuhan dasar sebagai kerangka kerja geometris untuk mendukung proses transformasi spasial hasil anotasi foto fasad. Bangunan tersebut telah ditempatkan pada lokasi

spasial yang benar, sesuai dengan data geospasial bangunan riil. Hal ini penting untuk memastikan ketepatan pemetaan anotasi jendela dalam sistem koordinat dunia.

Bangunan pada gambar IV.2 akan menjadi target dari proses peningkatan detail melalui *monoplotting*, di mana hasil anotasi dari foto fasad bangunan akan diproyeksikan ke bidang dinding putih yang terlihat dalam gambar. Selanjutnya, hasil tersebut dapat digunakan untuk visualisasi dan validasi objek pajak, sebagaimana dijabarkan dalam tujuan penelitian.

#### IV.2 Informasi Gambar tiap bangunan

Gambar IV.3 menampilkan antarmuka sistem untuk pengelolaan interaksi pengguna dalam menambahkan detail bangunan. Dalam tahap ini, pengguna telah memilih salah satu bangunan dari kumpulan model LoD2 yang tersedia, yang kemudian divisualisasikan dalam tampilan 3D dengan dinding putih dan atap merah berfokus pada bangunan tersebut saja. Model bangunan yang ditunjukkan berfungsi sebagai titik awal untuk proses penambahan informasi semantik, seperti bukaan jendela, melalui mekanisme anotasi berbasis citra fasad.

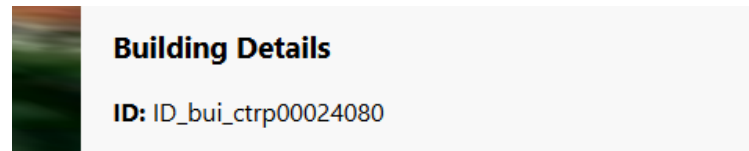


**Gambar IV. 3** Visualisasi antarmuka sistem untuk menambahkan informasi suatu bangunan

Pada sisi kanan antarmuka terdapat panel informasi bangunan dengan beberapa elemen kontrol utama yaitu,

- 1. Building Details (ID: ID\_bui\_city00024080)**

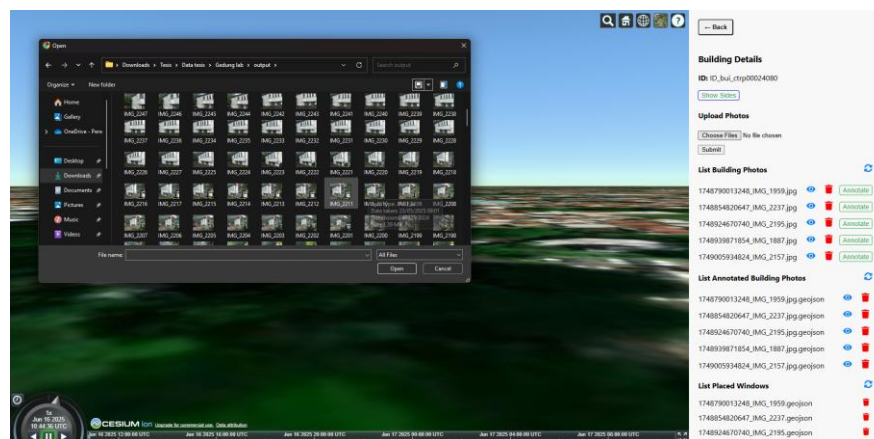
Menyediakan identitas unik dari bangunan yang sedang dianalisis. ID ini digunakan untuk mengelola asosiasi antara foto, anotasi, dan hasil proyeksi jendela ke model 3D.



**Gambar IV. 4** Contoh ID bangunan yang ditampilkan

## 2. Upload Photos

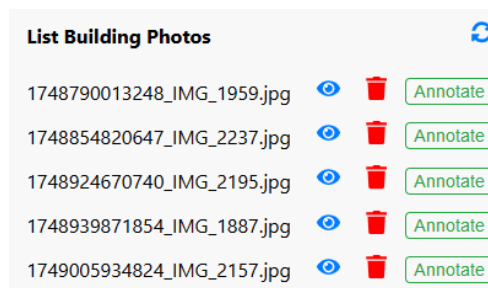
Merupakan komponen untuk pengguna dapat mengunggah citra fasad bangunan yang akan dianotasi. Tombol "Submit" memungkinkan proses unggah dijalankan, dan gambar disimpan ke direktori server untuk diproses lebih lanjut.



**Gambar IV. 5** Tampilan proses upload foto bangunan ke server

## 3. List Building Photos

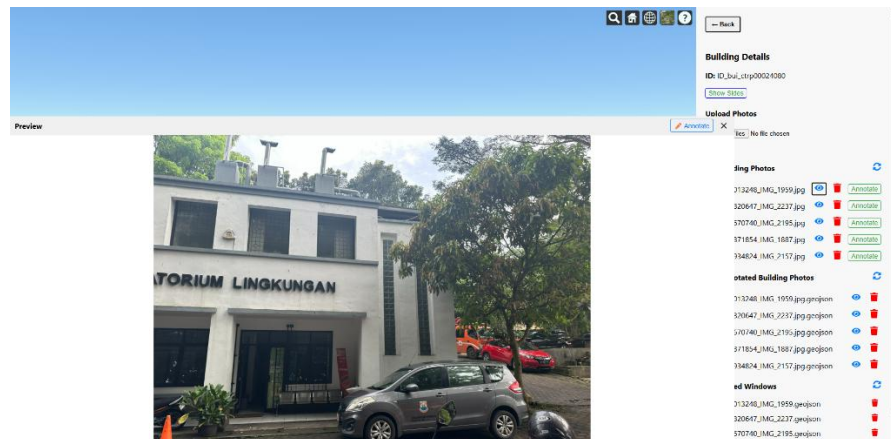
Menampilkan daftar foto yang telah diunggah oleh pengguna. Apabila pengguna sudah pernah mengupload foto ke server maka akan muncul id foto yang sudah diunggah pada list.



**Gambar IV. 6** Tampilan list foto yang sudah diunggah oleh pengguna

Pada menu tersebut pengguna dapat melihat foto yang sudah diunggah dengan mengklik ikon mata, atau menghapus dengan mengklik ikon tong sampah, dan juga dapat melakukan proses anotasi dengan mengklik tombol “*annotate*”

#### 4. Tombol Annotate

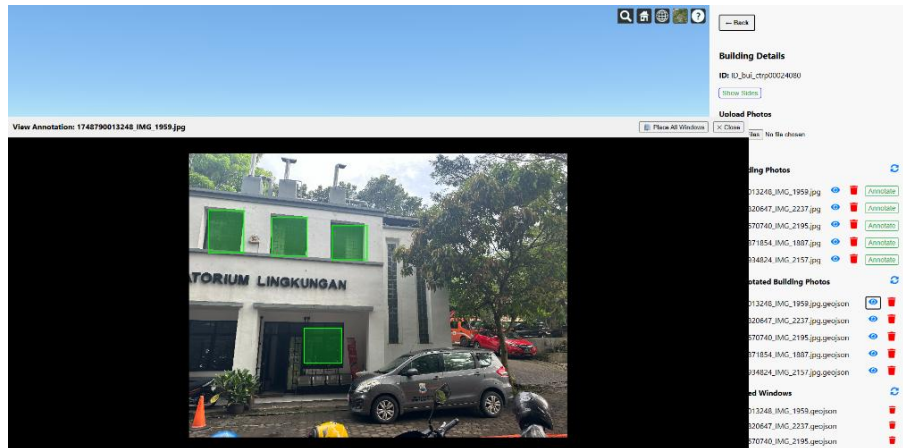


**Gambar IV. 7** Tampilan salah satu foto sisi bangunan yang sudah diunggah oleh pengguna

Gambar IV.7 menunjukkan antarmuka sistem pada tahap *preview photo*, yaitu tahap untuk melihat foto yang sudah diunggah oleh pengguna ke server. Untuk tahap yang dapat dilakukan oleh pengguna selanjutnya yaitu proses anotasi atau proses penandaan manual terhadap elemen bukaan bangunan (seperti jendela) yang ditampilkan langsung pada citra fasad bangunan. Foto ditampilkan dalam resolusi penuh di bagian kiri layar untuk memudahkan pengguna dalam menandai area jendela secara akurat.

Proses anotasi dilakukan dengan menggambar poligon persegi panjang langsung pada citra menggunakan komponen interaktif dalam antarmuka.





**Gambar IV. 8** Tampilan salah satu foto sisi bangunan yang sudah dianotasi oleh pengguna

Gambar IV.8 menampilkan antarmuka sistem pada tampilan anotasi citra fasad, di mana pengguna sudah selesai melakukan penandaan area jendela secara manual pada foto bangunan. Dalam hasil anotasi tersebut, tampak lima jendela telah berhasil dianotasi pada fasad depan bangunan “Laboratorium Lingkungan”, yang ditandai dengan kotak hijau transparan. Kotak anotasi tersebut merepresentasikan area bukaan (jendela) yang akan diproses dan diproyeksikan ke model 3D bangunan.

Setiap anotasi disimpan dalam format GeoJSON di server, yang mencakup informasi spasial seperti posisi piksel, dimensi anotasi, serta metadata kamera (GPS, focal length, ukuran sensor). Format ini memungkinkan sistem mentransformasikan posisi anotasi dari citra 2D ke koordinat spasial dunia nyata (longitude, latitude, altitude) melalui proses *monoplotting*.

## 5. List Annotated Building Photos

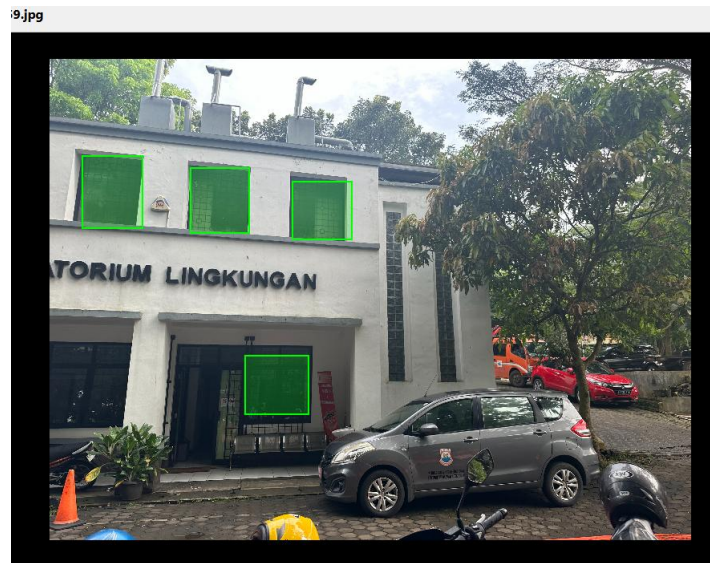
Berfungsi untuk menampilkan foto-foto fasad yang telah diberikan anotasi jendela oleh pengguna. Kolom ini akan terisi setelah proses anotasi dilakukan di antarmuka anotator.

## 6. List Placed Windows

Merupakan hasil akhir dari proses monoplotting, yaitu daftar elemen jendela yang telah diproyeksikan dari foto ke model 3D.

### IV.3 Hasil Visualisasi Bukaán Fasad Pada Model 3D Bangunan

Visualisasi bukaan fasad merupakan tahapan lanjutan dari proses anotasi citra bangunan yang bertujuan untuk memperkaya model 3D bangunan dengan elemen semantik, khususnya jendela sebagai indikator kualitas struktur. Gambar IV.9 memperlihatkan contoh hasil anotasi bukaan fasad yang dilakukan pada citra bangunan. Anotasi ditampilkan dalam bentuk kotak hijau transparan yang menunjukkan area jendela, baik pada lantai dasar maupun lantai atas bangunan.



**Gambar IV. 9** Contoh hasil anotasi pengguna pada salah satu sisi bangunan

Sebagaimana ditunjukkan pada Gambar IV.9, terdapat empat jendela yang telah berhasil dianotasi secara manual pada foto fasad bangunan. Setiap kotak anotasi berisi informasi posisi piksel, dimensi objek, serta metadata kamera yang digunakan untuk keperluan transformasi spasial. Hasil anotasi ini selanjutnya digunakan untuk proses proyeksi ke model 3D bangunan.

Proses proyeksi fasad bukaan ke model 3D bangunan dilakukan pada beberapa sisi bangunan yaitu,

#### IV.3.1 Sisi Depan Bangunan

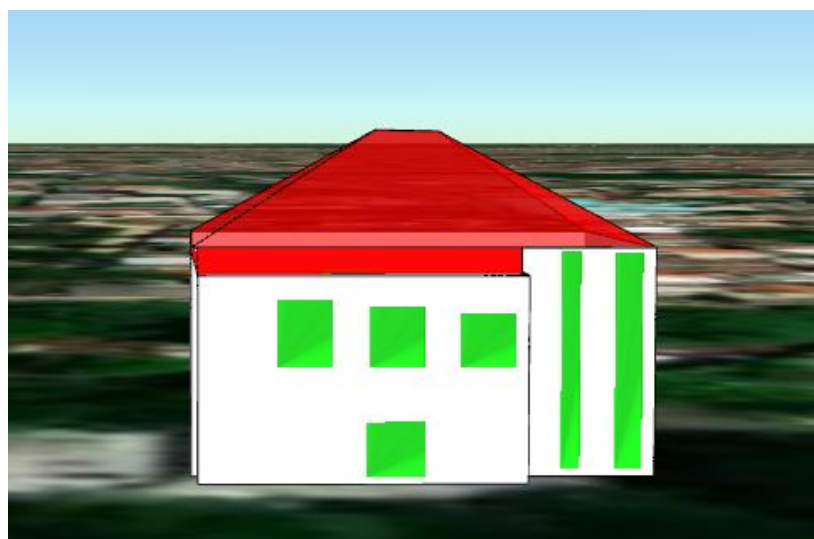
Salah satu sisi yang diproses adalah sisi depan bangunan, di mana terdapat total enam jendela yang dianotasi dari dua citra yang berbeda (Gamnbar IV.10). Penggunaan dua citra diperlukan karena pada salah satu gambar,

sebagian jendela tertutup oleh pohon, sehingga anotasi tambahan dilakukan menggunakan citra alternatif yang memperlihatkan area tersebut secara lebih jelas. Pemilihan citra-citra tersebut penting untuk memastikan seluruh bukaan dapat teridentifikasi dan diproyeksikan secara lengkap pada permukaan dinding model.



**Gambar IV. 10** Hasil anotasi pengguna pada sisi depan bangunan

Gambar IV.11 menunjukkan hasil proyeksi jendela dari citra fasad ke permukaan dinding model 3D bangunan. Proyeksi dilakukan terhadap sisi depan dan sebagian sisi samping bangunan, dengan total enam jendela yang ditampilkan dalam warna hijau transparan. Secara keseluruhan, jumlah jendela yang berhasil diproyeksikan sesuai dengan jumlah anotasi yang telah dibuat oleh pengguna, menunjukkan bahwa proses transformasi dari koordinat 2D ke spasial 3D berjalan dengan baik secara kuantitatif.



**Gambar IV. 11** Hasil proyeksi bukaan fasad pada model 3D bangunan (sisi depan)



Namun, secara geometris terdapat sedikit ketidaksesuaian pada posisi vertikal hasil proyeksi, khususnya pada beberapa jendela yang tampak melenceng dari batas dinding atau terlihat sedikit turun dari posisi yang seharusnya. Hal ini kemungkinan besar disebabkan oleh sudut pengambilan gambar yang terlalu rendah (menghadap ke atas) oleh pengguna saat memotret fasad. Perspektif kamera yang ekstrem dari bawah mengakibatkan distorsi bentuk jendela dalam citra, yang kemudian berdampak pada kesalahan estimasi saat proses *monoplotting*, terutama dalam penentuan elevasi dan sudut jatuh garis pandang terhadap permukaan dinding model.

Meskipun demikian, sistem tetap mampu memetakan posisi jendela sesuai dengan area yang dianotasi oleh pengguna, baik dari segi jumlah maupun letak horizontal. Hal ini menunjukkan bahwa komponen transformasi horizontal (longitude dan latitude) telah bekerja dengan cukup akurat, sementara koreksi pada komponen vertikal (elevasi) dapat menjadi fokus pengembangan lebih lanjut, misalnya dengan memasukkan estimasi tinggi kamera atau parameter bantuan (*ground control*).

Secara umum, hasil ini menggambarkan bahwa sistem mampu menjalankan proses proyeksi fasad dengan baik dalam kondisi citra yang terbatas, namun tetap sensitif terhadap kualitas dan sudut pandang input. Oleh karena itu, pemilihan sudut pengambilan foto yang lebih tegak lurus terhadap bidang fasad sangat disarankan agar posisi jendela pada model 3D dapat ditampilkan secara lebih presisi.

#### **IV.3.2 Sisi Samping Bangunan**

Gambar IV.12 memperlihatkan hasil anotasi manual pada sisi samping bangunan, di mana pengguna telah berhasil menandai total tujuh bukaan jendela, dengan lima jendela di lantai atas dan dua jendela besar di lantai bawah. Anotasi ditandai dengan kotak hijau transparan, menunjukkan area jendela yang akan diproses untuk diproyeksikan ke permukaan model 3D.



**Gambar IV. 12** Hasil anotasi pengguna pada sisi depan bangunan

Gambar IV.13 merupakan hasil dari proyeksi anotasi tersebut ke permukaan model 3D LoD2 menggunakan teknik *monoplotting*. Secara umum, jumlah dan posisi horizontal jendela pada model 3D sesuai dengan hasil anotasi pengguna, menunjukkan keberhasilan transformasi spasial dari citra 2D ke representasi 3D. Jendela besar pada bagian bawah berhasil diposisikan dengan tepat di zona bawah dinding model, sementara jendela vertikal lantai atas juga terletak di lokasi yang konsisten dengan struktur fisik pada foto aslinya.



**Gambar IV. 13** Hasil proyeksi bukaan fasad pada model 3D bangunan (sisi samping)

Namun, terdapat sedikit perbedaan dalam ukuran atau proporsi vertikal pada beberapa jendela, terutama pada bagian atas, yang tampak lebih tinggi dari seharusnya. Hal ini dapat disebabkan oleh beberapa faktor, antara lain:

1. Distorsi perspektif dari sudut kamera dengan gambar diambil dari posisi yang sedikit rendah dengan sudut elevasi menghadap ke atas, menyebabkan jendela bagian atas tampak lebih kecil atau lebih rapat dalam citra.
2. Kurangnya kalibrasi terhadap tinggi kamera atau ground reference yang mengakibatkan pergeseran skala vertikal saat diproyeksikan.
3. Tidak adanya referensi struktur geometri vertikal di model 3D yang dapat dijadikan acuan absolut saat proses penempatan.

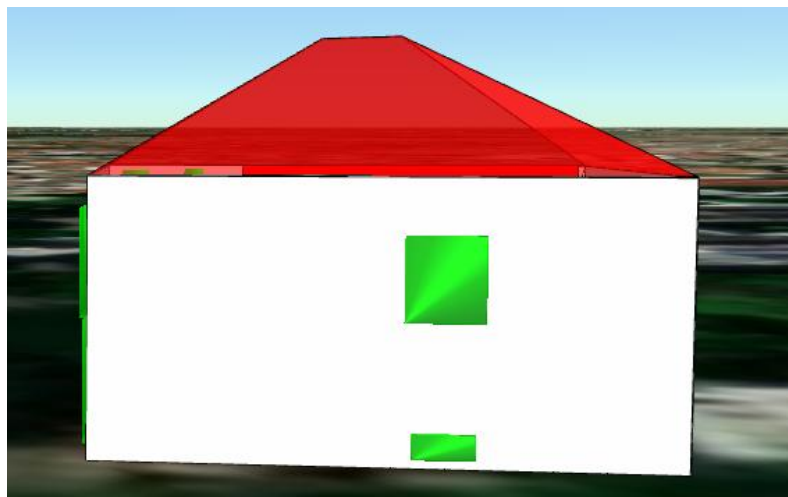
#### **IV.3.3 Sisi Belakang Bangunan**

Gambar IV.14 menampilkan citra bagian belakang bangunan dengan dua jendela yang telah dianotasi menggunakan kotak berwarna hijau. Jendela pertama berada di lantai dua dengan dimensi yang lebih besar, sedangkan jendela kedua terletak di lantai dasar dan berukuran lebih kecil. Citra ini diambil dari sudut samping karena akses langsung ke sisi belakang bangunan cukup terbatas. Faktor-faktor seperti kepadatan vegetasi, kemiringan tanah, dan keberadaan pagar besi menghambat mobilitas untuk pengambilan gambar secara menyeluruh. Meskipun demikian, fitur-fitur arsitektural utama, khususnya bukaan jendela, masih dapat dikenali secara visual sehingga memungkinkan dilakukannya anotasi.



**Gambar IV. 14** Hasil anotasi pengguna pada sisi belakang bangunan

Gambar IV.15 merupakan visualisasi model 3D dari sisi belakang bangunan yang sama, dengan dua bukaan jendela yang telah direkonstruksi secara digital. Penempatan jendela pada model 3D dilakukan berdasarkan hasil anotasi dari citra lapangan, dengan acuan utama pada posisi relatif dan sisi bangunan yang sesuai. Meskipun terdapat keterbatasan dalam penggunaan titik kontrol tanah (*ground control point*) serta ketidaktepatan dalam skala jarak absolut, kuantitas dan sisi bangunan tempat bukaan berada telah direpresentasikan dengan benar. Warna hijau yang digunakan pada model membantu menandai lokasi fitur yang telah dianotasi, sehingga tetap memungkinkan proses validasi visual secara kasar antara hasil observasi dua dimensi dan representasi tiga dimensi, meskipun belum sepenuhnya akurat secara geometrik.

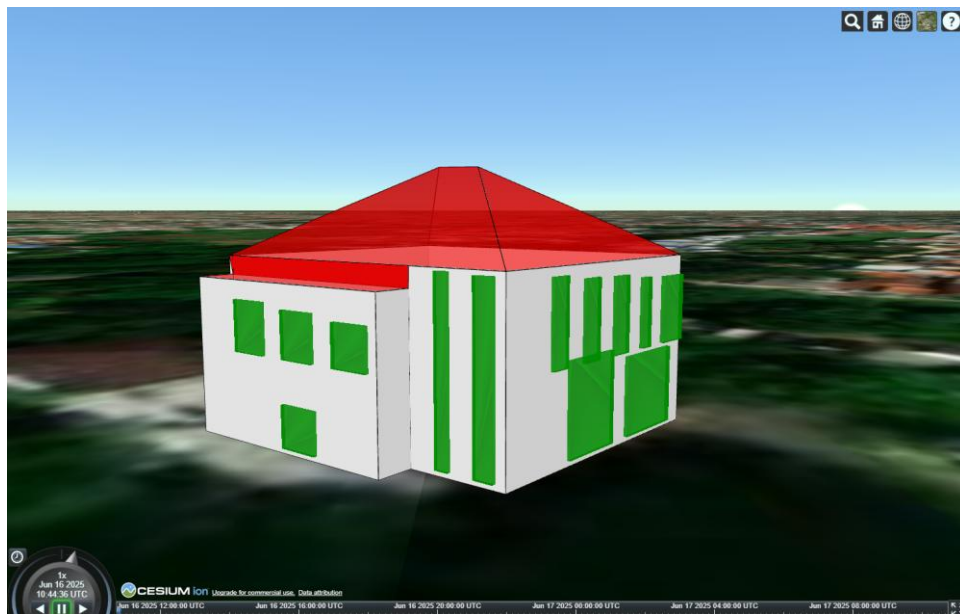


**Gambar IV. 15** Hasil proyeksi bukaan fasad pada model 3D bangunan (sisi belakang)

Secara keseluruhan, kedua gambar menunjukkan kecocokan yang cukup baik antara data anotasi dari citra nyata dengan hasil pemetaan pada model 3D, khususnya dalam hal jumlah bukaan dan kesesuaian sisi bangunan. Meskipun pengambilan gambar dilakukan dari area dengan akses yang terbatas serta tanpa dukungan titik kontrol tanah (*ground control*) yang memadai, posisi jendela tetap dapat diidentifikasi dan dianotasi secara visual dengan tingkat keandalan yang cukup. Proses ini mengindikasikan bahwa pendekatan berbasis pengamatan visual masih dapat digunakan untuk menghasilkan

representasi spasial yang informatif, meskipun akurasi geometrik belum sepenuhnya terjamin. Integrasi data observasi lapangan dengan model digital tetap memberikan kontribusi penting dalam mendokumentasikan elemen-elemen arsitektural, terutama dalam kondisi kerja yang menantang di lapangan.

Hasil secara keseluruhan tahap anotasi dan proyeksi ditunjukkan pada Gambar IV.16. Secara visual, distribusi anotasi tampak mencakup keseluruhan sisi fasad, termasuk sisi depan, samping kiri, dan belakang. Jumlah dan posisi bukaan relatif mencerminkan hasil observasi dari citra lapangan, meskipun terlihat adanya ketidaksesuaian minor dalam proporsi atau penempatan vertikal beberapa elemen. Warna hijau yang digunakan sebagai penanda anotasi berhasil menonjolkan fitur-fitur utama bangunan dan memudahkan proses evaluasi secara visual terhadap keberhasilan integrasi antara data citra 2D dan model 3D.



**Gambar IV. 16** Hasil proyeksi bukaan fasad pada model 3D bangunan secara keseluruhan

Namun, dalam proses pembuatan model ini terdapat beberapa kendala teknis yang mempengaruhi akurasi spasial anotasi. Salah satu tantangan utama adalah tidak digunakannya kalibrasi kamera, yang menyebabkan distorsi perspektif tidak terkoreksi dan berdampak langsung pada keakuratan rekonstruksi 3D. Selain itu, seluruh citra diambil dari permukaan tanah menggunakan kamera ponsel tanpa

pengaturan sudut tegak lurus atau tripod, sehingga menghasilkan citra dengan sudut miring dan variasi tinggi rendah yang tidak konsisten. Tidak adanya Ground Control Points (GCP) juga turut menyulitkan proses georeferensi dan orientasi absolut model dalam ruang tiga dimensi. Akibatnya, posisi bukaan pada model bersifat relatif dan belum sepenuhnya presisi terhadap posisi sebenarnya di dunia nyata.

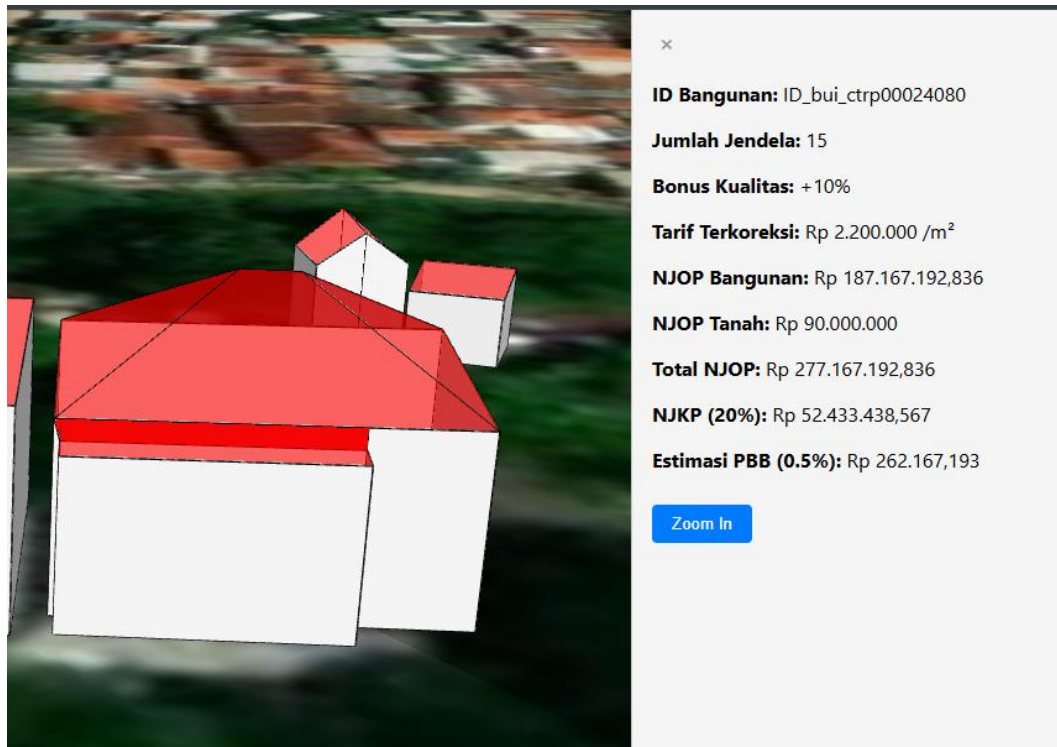
Dari segi visual, distribusi jendela tampak menyebar secara proporsional pada sisi-sisi fasad bangunan yang relevan, mencerminkan hasil anotasi dari beberapa sudut foto yang berbeda. Hal ini menunjukkan bahwa sistem mampu menggabungkan hasil dari berbagai perspektif citra menjadi satu kesatuan representasi spasial yang utuh dan akurat. Integrasi antara *backend* dan *frontend* yang mendukung pengunggahan foto, penyimpanan anotasi, serta proyeksi spasial menjadikan sistem ini sebagai solusi yang praktis dan efisien untuk memperkaya model 3D LoD2.

Secara keseluruhan, gambar ini menegaskan bahwa sistem berhasil mewujudkan proses transformasi data fasad dari citra ke dalam bentuk spasial 3D secara menyeluruh. Hal ini tidak hanya meningkatkan kualitas semantik model bangunan, tetapi juga membuka peluang untuk dimanfaatkannya sistem ini dalam proses validasi objek pajak atau audit bangunan secara digital.

#### **IV.4 Visualisasi Informasi Pajak Bumi dan Bangunan**

Gambar IV.17 menunjukkan antarmuka visualisasi 3D bangunan yang dilengkapi dengan panel informasi perpajakan secara rinci. Model bangunan ditampilkan dalam representasi geometris sederhana dengan atap berwarna merah dan dinding putih, sementara panel di sebelah kanan memberikan data yang mencakup ID Bangunan, jumlah jendela, bonus kualitas, tarif terkoreksi, serta perhitungan lengkap mulai dari NJOP hingga estimasi PBB.





**Gambar IV. 17** Visualisasi perhitungan pajak bumi dan bangunan pada bangunan yang dianalisis

Salah satu komponen penting yang ditampilkan adalah jumlah jendela (15 buah), yang secara langsung memberikan kontribusi pada komponen bonus kualitas sebesar +10%. Dalam konteks penilaian bangunan, jumlah jendela bisa diinterpretasikan sebagai indikator kualitas arsitektural dan kenyamanan penghuni merupakan dua faktor yang biasanya meningkatkan nilai bangunan. Dengan asumsi bahwa semakin banyak jendela berarti bangunan lebih terang dan sirkulasi udara lebih baik. Peningkatan ini berdampak pada nilai akhir NJOP bangunan, yang kemudian berpengaruh terhadap NJOP total, NJKP, dan akhirnya estimasi PBB. Hal ini menunjukkan bahwa data geometris dan anotasi elemen arsitektural seperti jendela bukan hanya penting untuk dokumentasi visual, tetapi juga berimplikasi langsung pada aspek fiskal dalam kebijakan pemetaan dan perpajakan properti.

## Bab V    Kesimpulan

### V.1   Kesimpulan

Penelitian ini berhasil mengembangkan sebuah sistem berbasis web yang mampu meningkatkan tingkat kedetailan model 3D bangunan Level of Detail 2 (LoD2) melalui penambahan informasi semantik berupa bukaan fasad bangunan (jendela) menggunakan teknik *monoplotting*. Teknik ini memungkinkan proyeksi spasial elemen 2D dari citra fasad ke permukaan model 3D secara semi-manual dengan input pengguna, dan menghasilkan representasi 3D yang lebih informatif tanpa membangun model LoD3 dari awal.

Proses pengembangan dilakukan dengan pendekatan sistem client-server, di mana *frontend* menggunakan React.js dan CesiumJS untuk visualisasi interaktif, sementara *backend* berbasis Node.js menangani penyimpanan, pemrosesan, serta manajemen data anotasi dan hasil proyeksi. Hasil sistem menunjukkan bahwa pengguna dapat melakukan anotasi bukaan langsung pada citra fasad, dan hasilnya berhasil dipetakan ke model 3D dengan akurasi spasial yang memadai, meskipun terdapat keterbatasan dalam orientasi kamera dan *ground control*. Jumlah dan distribusi jendela yang diperoleh terbukti dapat digunakan untuk memperkaya informasi geometrik dan semantik bangunan, serta memiliki implikasi potensial dalam validasi objek pajak. Sistem ini juga membuktikan bahwa pendekatan berbasis foto fasad dan pemetaan visual dapat menjadi alternatif yang efisien dan ekonomis untuk meningkatkan kualitas peta kota digital.

### V.2   Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diajukan untuk pengembangan lebih lanjut diantaranya,

1. **Peningkatan Akurasi Geometrik melalui Kalibrasi Kamera dan Titik Kontrol**

Proses *monoplotting* dalam penelitian ini masih memiliki keterbatasan akurasi akibat tidak dilakukannya kalibrasi kamera secara eksplisit dan tidak tersedianya *Ground Control Points* (GCP) di lapangan. Oleh karena itu, disarankan untuk menggunakan kamera dengan parameter kalibrasi yang



diketahui atau melakukan proses kalibrasi terlebih dahulu, serta menambahkan GCP sebagai referensi spasial agar hasil proyeksi lebih presisi secara geometrik.

## **2. Automasi Deteksi Bukaannya Menggunakan Visi Komputer**

Anotasi bukaan jendela masih dilakukan secara manual oleh pengguna. Untuk mempercepat dan menyederhanakan proses ini, disarankan agar pengembangan sistem selanjutnya mengintegrasikan algoritma visi komputer atau machine learning guna melakukan deteksi otomatis terhadap elemen fasad pada citra.

## **3. Penambahan Ragam dan Kompleksitas Bentuk Arsitektural**

Saat ini, bentuk bukaan dibatasi pada geometri sederhana (persegi panjang). Pengembangan sistem yang mendukung anotasi bentuk bebas atau bukaan non-ortogonal seperti lengkungan atau elemen dekoratif akan meningkatkan fleksibilitas dan representasi semantik model 3D secara lebih realistis.

## **4. Integrasi dengan Sistem Administratif Pajak dan Kadaster**

Meskipun sistem ini telah menunjukkan potensi untuk mendukung validasi objek pajak, pengembangan lebih lanjut perlu mempertimbangkan integrasi dengan sistem kadaster fiskal dan basis data pajak daerah seperti SISMIOIP untuk menciptakan sistem validasi yang operasional dan sinkron dengan kebijakan fiskal daerah.

## **5. Pengujian Skala Luas dan Uji Pengguna**

Untuk menilai performa sistem dalam kondisi dunia nyata, diperlukan uji coba pada area perkotaan yang lebih luas serta pengujian oleh pengguna non-teknis (seperti petugas pajak atau masyarakat umum). Hal ini bertujuan untuk mengevaluasi aspek kegunaan (*usability*), efisiensi kerja, dan skalabilitas sistem.

## DAFTAR PUSTAKA

- Abdul Rahman, A., Rashidan, H., Musliman, I. A., Buyuksalih, G., Bayburt, S., & Baskaraca, P. (2019). *3D Geospatial Database Schema For Istanbul 3D City Model*. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4/W16), 11–16. <https://doi.org/10.5194/isprs-archives-XLII-4-W16-11-2019>
- Abdul Rahman, A., Zamzuri, A., & Rashidan, H. (2025). *3D Geoinformation – Then and Now*. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-M-6–2025, 1–7. <https://doi.org/10.5194/isprs-archives-XLVIII-M-6-2025-1-2025>
- Arnold, J. M. (2012). *Improving the Tax System in Indonesia* (OECD Economics Department Working Papers, Vol. 998). <https://doi.org/10.1787/5k912j3r2qmr-en>
- Bieringer, A., Wysocki, O., Tuttas, S., Hoegner, L., & Holst, C. (2024). *Analyzing the impact of semantic LoD3 building models on image-based vehicle localization*. <https://doi.org/10.5194/isprs-annals-X-4-W5-2024-55-2024>
- Biljecki, F., Ledoux, H., & Stoter, J. (2016a). *An improved LOD specification for 3D building models*. *Computers, Environment and Urban Systems*, 59, 25–37. <https://doi.org/10.1016/j.compenvurbsys.2016.04.005>
- Biljecki, F., Ledoux, H., & Stoter, J. (2016b). *Generation Of Multi-Lod 3D City Models In CityGML With The Procedural Modelling Engine Random3dcity*. *Isprs Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4W1), 51–59. <https://doi.org/10.5194/isprs-annals-IV-4-W1-51-2016>
- Booth, B. (2025). *Determining a Z-factor for scaling linear elevation units to match geographic coordinate values*. ArcGIS Blog. Diakses 18 Juni 2025, <https://www.esri.com/arcgis-blog/products/arcgis-desktop/defense/determining-a-z-factor-for-scaling-linear-elevation-units-to-match-geographic-coordinate-values?>
- Bozzini, C., Conedera, M., & Krebs, P. (2012). *A New Monoplotting Tool to Extract Georeferenced Vector Data and Orthorectified Raster Data from Oblique Non-Metric Photographs in the Digital Era*. *A New Monoplotting Tool to Extract Georeferenced Vector Data and Orthorectified Raster Data from Oblique Non-Metric Photographs*. Dalam *International Journal of Heritage* (Vol. 1, Nomor 3).

- Chang, G., Xu, T., & Wang, Q. (2018). *M-estimator for the 3D symmetric Helmert coordinate transformation*. *Journal of Geodesy*, 92(1), 47–58. <https://doi.org/10.1007/s00190-017-1043-9>
- Chen, Q., Chen, J., Sun, K., Huang, M., Chen, G., & Liu, H. (2024). *A Parallel-Optimized Visualization Method for Large-Scale Multiple Video-Augmented Geographic Scenes on Cesium*. *ISPRS International Journal of Geo-Information*, 13(12). <https://doi.org/10.3390/ijgi13120463>
- Chen, Y., Shooraj, E., Rajabifard, A., & Sabri, S. (2018). *From IFC to 3D tiles: An integrated open-source solution for visualising BIMs on cesium*. *ISPRS International Journal of Geo-Information*, 7(10). <https://doi.org/10.3390/ijgi7100393>
- Chiang, Y. Y., Leyk, S., & Knoblock, C. A. (2014). *A survey of digital map processing techniques*. Dalam *ACM Computing Surveys (Vol. 47, Nomor 1)*. Association for Computing Machinery. <https://doi.org/10.1145/2557423>
- Dai, F., Lu, M., & Asce, M. (2010). *Assessing the Accuracy of Applying Photogrammetry to Take Geometric Measurements on Building Products*. *Journal Of Construction Engineering And Management*, DOI: 10.1061/(ASCE)CO.1943-7862.0000114. <https://doi.org/10.1061/ASCECO.1943-7862.0000114>
- Dangermond, J., & Goodchild, M. F. (2020). *Building geospatial infrastructure*. *Geo-Spatial Information Science*, 23(1), 1–9. <https://doi.org/10.1080/10095020.2019.1698274>
- Dari, W. (2022). *Tax Sustainability: The Effectiveness and Contribution of Land and Building Taxes to Regional Original Revenue in Yogyakarta City, Indonesia*. *Journal of Economics, Finance And Management Studies*, 05(03). <https://doi.org/10.47191/jefms/v5-i3-01>
- Galantucci, R. A., & Fatiguso, F. (2019). *Advanced damage detection techniques in historical buildings using digital photogrammetry and 3D surface anlysis*. *Journal of Cultural Heritage*, 36, 51–62. <https://doi.org/10.1016/j.culher.2018.09.014>
- GeoCue Group. (2025). *Converting focal length from pixels to millimeters to use in Bentley ContextCapture*. GeoCue Support. Diakses 18 Juni 2025, from <https://support.geocue.com/converting-focal-length-from-pixels-to-millimeters-to-use-in-bentley-context-capture/>.
- Goetz, M. (2013). *Towards generating highly detailed 3D CityGML models from OpenStreetMap*. *International Journal of*

- Geographical Information Science*, 27(5), 845–865.  
<https://doi.org/10.1080/13658816.2012.721552>
- Golparvar, B., & Wang, R. Q. (2025). *Advancing Earth science applications through a semi-automatic monoplotting framework for efficient 3D geo-referencing of monocular oblique visual Data. Computers and Geosciences*, 200.  
<https://doi.org/10.1016/j.cageo.2025.105915>
- Google. (2025). *Google Maps*. diakses pada 18 Juni 2025  
<https://www.google.com/maps> <https://www.google.com/maps>
- Grussenmeyer, P., & Al Khalil, O. (2002). *Solutions for exterior orientation in photogrammetry: A review. Photogrammetric Record*, 17(100), 615–634. <https://doi.org/10.1111/j.1477-9730.2002.tb01907.x>
- Hidayat, R., & Wati, S. R. (2022). Pengaruh kesadaran Wajib Pajak dan kualitas pelayanan pajak terhadap kepatuhan Wajib Pajak dalam membayar Pajak bumi dan Bangunan di kota Bandung. *Owner*, 6(4), 4009–4020. <https://doi.org/10.33395/owner.v6i4.1068>
- Huang, X., & Keong KWOH, L. (2008). *Monoplotting a Semi-automated Approach for 3D Reconstruction from Single Satellite Image. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.
- Jones, A., Blake, C., Davies, C., & Scanlon, E. (2004). *Digital maps for learning: A review and prospects. Computers and Education*, 43(1-2 SPEC ISS.), 91–107.  
<https://doi.org/10.1016/j.compedu.2003.12.007>
- Kilsedar, C. E., Fissore, F., Pirotti, F., & Brovelli, M. A. (2019). *Extraction and visualization of 3d building models in urban areas for flood simulation. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2/W11), 669–673.  
<https://doi.org/10.5194/isprs-Archives-XLII-2-W11-669-2019>
- Kolbe, T. H. (2008). *Representing and Exchanging 3D City Models with CityGML. Dalam Representing and Exchanging 3D City Models with CityGML* (hlm. 15–31). [https://doi.org/DOI:10.1007/978-3-540-87395-2\\_2](https://doi.org/DOI:10.1007/978-3-540-87395-2_2)
- Kumar, M. (1988). *World geodetic system 1984: a modern and accurate global reference frame. Marine Geodesy*, 12(2), 117–126.  
<https://doi.org/10.1080/15210608809379580>

- Li, H., Chen, Z., & Li, Z. (2024). *Research and Application of 3D Modeling and Rendering on CIM Basic Platform in Cesium Engine*. <https://doi.org/10.20944/preprints202410.2170.v1>
- Liu, C., Song, B., Fu, M., Meng, X., Zhao, Y., Wang, X., Li, X., Liu, Z., & Han, Y. (2023). *Cesium-MRS: A Cesium-based Platform for Visualizing Multi-source Remote Sensing Data*. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 48(5/W1-2023), 15–19. <https://doi.org/10.5194/isprs-archives-XLVIII-5-W1-2023-15-2023>
- Lunde, V. (2022). *Analysing methods for visualizing time-series datasets in open-source web mapping*. Lund University.
- Mukhlisin, M. (2021). Peranan Pajak Bumi Dan Bangunan Perdesaan Dan Perkotaan Melalui Efektivitas Dan Kontribusi Terhadap Pendapatan Asli Daerah Kabupaten Situbondo. *Journal of Law, Administration, and Social Science*.
- Murtiyoso, A., Grussenmeyer, P., & Suwardhi, D. (2019). *Technical considerations in low-cost heritage documentation*. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(2/W17), 225–232. <https://doi.org/10.5194/isprs-archives-XLII-2-W17-225-2019>
- Park, H. S. (2025). *Camera model*. CSCI 5980 Multiview 3D Geometry in Computer Vision, University of Minnesota. University of Minnesota. Diakses 18 Juni 2025 dari <https://www-users.cse.umn.edu/~hspark/CSci5980/ch1.pdf>.
- Pemerintah Kota Cimahi. (2023). Peraturan Daerah Kota Cimahi Nomor 8 Tahun 2023 tentang Pajak Daerah dan Retribusi Daerah. Dalam *Lembaran Daerah Kota Cimahi Tahun 2023 Nomor 303*. Diundangkan di Cimahi, 29 Desember 2023.
- Hanke, K. (2002). *Architectural photogrammetry*. <https://www.researchgate.net/publication/277296138>
- Republik Indonesia. (2009). Undang-Undang Nomor 28 Tahun 2009 tentang Pajak Daerah dan Retribusi Daerah. Dalam *Lembaran Negara Republik Indonesia Tahun 2009 Nomor 130*. Jakarta: Sekretariat Negara.
- Republik Indonesia. (2022). Undang-Undang Nomor 1 Tahun 2022 tentang Hubungan Keuangan antara Pemerintah Pusat dan Pemerintahan Daerah. Dalam *Lembaran Negara Republik Indonesia Tahun 2022 Nomor 4; Tambahan Lembaran Negara Nomor 6757*. Diundangkan di Jakarta, 5 Januari 2022.

- Republik Indonesia. (2024). Peraturan Menteri Keuangan Nomor 85 Tahun 2024 tentang Penilaian Pajak Bumi dan Bangunan Perdesaan dan Perkotaan. Dalam *Berita Negara Republik Indonesia Tahun 2024 Nomor 881*. Ditetapkan di Jakarta, 25 Oktober 2024; Diundangkan 26 November 2024.
- Riningsih, D. (2024). Pelatihan Perhitungan dan Pembayaran Pajak Bumi dan Bangunan. *Journal of Education Research*.
- Pertiwi, R. N. (2014). Analisis Efektivitas Pemungutan Pajak Bumi Dan Bangunan (Studi Pada Dinas Pendapatan, Pengelolaan Keuangan dan Aset Kota Probolinggo). *Jurnal Perpajakan Vol.3 No.1*.
- Roos, W. (t.t.). *951 Photogrammetria On the definition of fundamental concepts in photogrammetry*.
- Roukema, B. F., & Lew, B. (2004). *A Solution to the Isolatitude, Equi-area, Hierarchical Pixel-Coordinate System*.
- Satar, M. (2013). *Manual Penggunaan GIS Manual ArcGIS Tingkat Dasar*. Padang: Universitas Negeri Padang.
- Schenk, T., & Quarter, A. (2005). *Introduction to Photogrammetry*.
- Shashi, M., Jain, K., & Shashi, M. (2007). *Use of photogrammetry in 3D modeling and visualization of buildings*. 2(2). <https://www.researchgate.net/publication/255651498>
- Siahaan, M. (2018). Penerapan Penghitungan Pajak dan Pencatatan serta Pelaporan Pajak Bumi dan Bangunan Apartemen X. *Jurnal Riset Manajemen dan Bisnis (JRMB) Fakultas Ekonomi UNIAT*, 3(1), 121–134.
- Stockdale, C. A., Bozzini, C., Macdonald, S. E., & Higgs, E. (2015). *Extracting ecological information from oblique angle terrestrial landscape photographs: Performance evaluation of the WSL Monoplotting Tool*. *Applied Geography*, 63, 315–325. <https://doi.org/10.1016/j.apgeog.2015.07.012>
- Triglav-Čekada, M., Bric, V., & Zorn, M. (2014). *How to decide which oblique image has the highest mapping potential for monoplotting method: A case studies on river erosion and floods*. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(5), 379–384. <https://doi.org/10.5194/isprsannals-II-5-379-2014>
- Van Sickle, J. (2025). *Lesson 6: State Plane Coordinates and Heights*. Dalam *GEOG 862: GPS and GNSS for Geospatial Professionals*. The Pennsylvania State University. Diakses dari <https://www.e-education.psu.edu/geog862/print/book/export/html/1793>.

- Walter, V., & Luo, F. (2011). *Automatic interpretation of digital maps. ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4), 519–528. <https://doi.org/10.1016/j.isprsjprs.2011.02.010>
- Willneff, J., Poon, J., & Fraser, C. (2005). *Monoplotting applied to high-resolution satellite imagery. Journal of Spatial Science*, 50(2), 1–11. <https://doi.org/10.1080/14498596.2005.9635045>
- Xia, Z., & Rüppel, U. (2024). *LoD2BIM: A New Workflow for reconstructing and converting LoD2 Model to Information-rich IFC Model for Existing Building. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10(4/W5-2024), 325–332. <https://doi.org/10.5194/isprs-annals-X-4-W5-2024-325-2024>
- Yang, B. (2019). *Developing a Mobile Mapping System for 3D GIS and Smart City Planning. MDPI Sustainability Journal*. <https://doi.org/doi:10.3390/su11133713>
- Yoo, B., & Han, S. (2006). *Image-Based Modeling of Urban Buildings Using Aerial Photographs and Digital Maps. Dalam Transactions in GIS* (Vol. 10, Nomor 3). <http://www.u-data.com>
- Zhou, G., Luo, Q., Xie, W., Yue, T., Huang, J., & Shen, Y. (2016). *Transformation model with constraints for high-accuracy of 2D-3D building registration in aerial imagery. Remote Sensing*, 8(6). <https://doi.org/10.3390/rs8060507>
- Zhu, G., Yuan, Z., Wang, N., Cheng, Y., & Li, J. (t.t.). *Research on the Development and Application of 3D Cloud Platform Based on Cesium. World Scientific Research Journal*, 8, 2022. [https://doi.org/10.6911/WSRJ.202210\\_8\(10\).0018](https://doi.org/10.6911/WSRJ.202210_8(10).0018)

## **LAMPIRAN**



## Lampiran A Program *viewer building 3D* model pada CesiumJS

### 1. Inisialisasi Viewer dan Token Cesium

```
Ion.defaultAccessToken = "eyJ..."; // Token akses Cesium
const viewer = new Viewer(viewerRef.current, {
  infoBox: false,
  selectionIndicator: false,
});
```

### 2. Memuat dan Menampilkan GeoJSON Bangunan

```
IonResource.fromAssetId(3410848)
  .then((resource) =>
    GeoJsonDataSource.load(resource, {
      stroke: Color.BLACK,
      strokeWidth: 1,
      clampToGround: false,
    })
  )
  .then((dataSource) => {
    viewer.dataSources.add(dataSource);
    viewer.zoomTo(dataSource);
    ...
  });
```

### 3. Event Handler Saat Klik Bangunan

```
clickHandler.setInputAction(async (movement) => {
  const picked = viewer.scene.pick(movement.position);
  if (!picked || !picked.id ||
    !defined(picked.id.properties)) return;

  const buildingId = ...;
  ...
}, ScreenSpaceEventType.LEFT_CLICK);
```

#### 4. Perhitungan Jumlah Jendela Bangunan

```
const res = await
fetch(`http://localhost:5000/api/placed_windows/${buildingId}`);
...
totalCount += geojson.features?.length || 0;
```

#### 5. Perhitungan Luas Bangunan (*Footprint*)

```
let footprintArea = await
calculateFootprintAreaFromGeoJSON(buildingId);
```

#### 6. Hitung NJOP dan Estimasi PBB

```
const buildingNJOP = footprintArea * njopBangunanPerM2;
const landNJOP = landArea * njopTanahPerM2;
...
const njkp = njkpRate * Math.max(0, totalNJOP -
njoptkp);
const pbbDue = 0.005 * njkp;
```

#### 7. Panel UI Info Bangunan

```
{buildingInfo && (
  <div style={{ position: "absolute", top: 0, right: 0,
... }}>
    ...
    <p><strong>Jumlah Jendela:</strong>
{buildingInfo.windowCount ?? "N/A"}</p>
    <p><strong>Estimasi PBB:</strong>
{formatRupiah(buildingInfo.pbbDue)}</p>
    ...
  </div>
)}
```

## Lampiran B Program untuk viewer satu bangunan dan UI untuk penambahan informasi bangunan

### 1. Inisialisasi Viewer Cesium

```
useEffect(() => {
  Ion.defaultAccessToken = "your_token_here";
  const viewer = new Viewer(viewerRef.current);
  viewerInstanceRef.current = viewer;

  // Load building data
  IonResource.fromAssetId(3410848)
    .then((resource) => {
      GeoJsonDataSource.load(resource, {
        stroke: Color.BLACK,
        strokeWidth: 1,
        clampToGround: false,
      })
    })
    .then((dataSource) => {
      viewer.dataSources.add(dataSource);
      // ... (pemrosesan entitas bangunan)
    });

  return () => {
    if (viewer && !viewer.isDestroyed()) {
      viewer.destroy();
    }
  };
}, [uid]);
```

### 2. Visualisasi Dinding Bangunan

```
const handleShowSides = async () => {
  // ... (persiapan data)

  const wallGeoJSON = {
    type: "FeatureCollection",
```

```

        features: wallPolygons,
    };

    const dataSource = await
    GeoJsonDataSource.load(wallGeoJSON, {
        clampToGround: false,
    });

    // Visualisasi dinding
    dataSource.entities.values.forEach((entity, i) => {
        entity.polygon.material =
    Color.ORANGE.withAlpha(0.5);
        entity.polygon.outline = true;

        // Menambahkan label
        viewerInstanceRef.current.entities.add({
            position: center,
            label: {
                text: `Side ${i + 1}`,
                font: "14px sans-serif",
                fillColor: Color.BLACK,
            },
        });
    });
};

```

### 3. Penanganan Anotasi Jendela

```

useEffect(() => {
    // ... (persiapan)

    annotations.forEach((file) => {
        fetch(url)
            .then((res) => res.json())
            .then((geojson) => {
                // Skala koordinat berdasarkan ukuran asli
            })
    })
})

```

```

        const scaledCoords = coords.map(([lon, lat]) =>
[
            centroid[0] + (lon - centroid[0]) *
scaleFactor,
            centroid[1] + (lat - centroid[1]) *
scaleFactor,
        ]);

        return GeoJsonDataSource.load(scaledGeojson, {
            clampToGround: false,
        });
    })
    .then((dataSource) => {
        // Visualisasi jendela
        entity.polygon.height = heightOffset;
        entity.polygon.extrudedHeight = heightOffset +
0.2;

        entity.polygon.material =
Color.LIME.withAlpha(0.6);
    });
});
}, [annotations, uid]);

```

#### 4. Upload dan Manajemen Gambar

```

const handleUpload = async () => {
    const formData = new FormData();
    for (let file of files) {
        formData.append("images", file);
    }

    await
fetch(`http://localhost:5000/api/upload/${uid}`, {
    method: "POST",
    body: formData,
});

```

```

    loadImages();
};

const loadImages = async () => {
    const res = await
fetch(`http://localhost:5000/api/images/${uid}`);
    const data = await res.json();
    setImages(data);
};

```

## 5. Komponen Antarmuka Pengguna

```

return (
  <div style={{ display: "flex", height: "100vh" }}>
    { /* 3D Viewer */ }
    <div ref={viewerRef} style={{ flex: 15 }} />
    <div style={{ flex: 4, padding: "20px" }}>
      { /* ... berbagai kontrol UI */ }
    </div>
    {selectedImage && !isAnnotating && (
      <div style={{ /* styling */ }}>
        <img src={selectedImage} alt="preview" />
      </div>
    )}
    { /* Mode Anotasi */ }
    {selectedImage && isAnnotating && (
      <ImageAnnotator
        imageUrl={selectedImage}
        buildingId={uid}
        onClose={() => setIsAnnotating(false)}
      />
    )}
  </div>
);

```

## Lampiran C Program untuk anotasi gambar

### 1. Ekstraksi Metadata EXIF

```
useEffect(() => {
  fetch(imageUrl)
    .then((res) => res.blob())
    .then((blob) => exifr.parse(blob, { gps: true,
xmp: true, tiff: true }))
    .then((data) => {
      // Konversi koordinat GPS
      let lat = data.latitude;
      let lon = data.longitude;

      if (Array.isArray(data.GPSLatitude)) {
        const [d, m, s] = data.GPSLatitude;
        lat = d + m/60 + s/3600;
        if (data.GPSLatitudeRef === "S") lat = -lat;
      }

      setImageMetadata({
        ...data,
        latitude: lat,
        longitude: lon
      });
    });
}, [imageUrl]);
```

### 2. Fungsi Pembuatan Anotasi

```
const handleMouseDown = () => {
  const pos = stageRef.current.getPointerPosition();
  const unscaled = { x: pos.x/scale, y: pos.y/scale };
  setNewRect({ x: unscaled.x, y: unscaled.y, width: 0,
height: 0 });
};

const handleMouseMove = () => {
  if (!newRect) return;
  const pos = stageRef.current.getPointerPosition();
```

```

    const unscaled = { x: pos.x/scale, y: pos.y/scale };
    setNewRect({ ...newRect, width: unscaled.x-newRect.x,
height: unscaled.y-newRect.y });
  };
  const handleMouseUp = () => {
    if (newRect && Math.abs(newRect.width) > 5 &&
Math.abs(newRect.height) > 5) {
      setRects([...rects, newRect]);
    }
    setNewRect(null);
  };
};

```

### 3. Konversi ke GeoJSON

```

const convertToGeoJSON = () => {
  const meterPerPixel = (mmPerPixel * distance) / focal
/ 1000;
  const pixelToLatLonAlt = (pt) => {
    const dx = pt.x - img.width/2;
    const dy = pt.y - img.height/2;
    const lon = longitude +
(dx*meterPerPixel)/(111320*Math.cos(latitude*Math.PI/18
0));
    const lat = latitude - (dy*meterPerPixel)/110540;
    return [lon, lat, alt];
  };
  return {
    type: "FeatureCollection",
    features: rects.map((r) => ({
      type: "Feature",
      geometry: {
        type: "Polygon",
        coordinates: [[
          pixelToLatLonAlt({x: r.x, y: r.y}),
          pixelToLatLonAlt({x: r.x+r.width, y: r.y}),

```



```

        pixelToLatLonAlt({x: r.x+r.width, y:
r.y+r.height})),
        pixelToLatLonAlt({x: r.x, y: r.y+r.height})),
        pixelToLatLonAlt({x: r.x, y: r.y})
    ]]
  },
  properties: {
    widthInMeters: Math.abs(r.width) *
meterPerPixel,
    heightInMeters: Math.abs(r.height) *
meterPerPixel
  }
}))
};
};

```

#### 4. Komponen Visualisasi

```

<Stage
  width={img.width * scale}
  height={img.height * scale}
  scale={{ x: scale, y: scale }}
  ref={stageRef}
  onMouseDown={handleMouseDown}
  onMouseMove={handleMouseMove}
  onMouseUp={handleMouseUp}
>
  <Layer>
    <KonvaImage image={img} />
    {rects.map((r, idx) => (
      <Rect
        key={idx}
        x={r.x}
        y={r.y}
        width={r.width}
        height={r.height}

```

```

        stroke="red"
        fill="rgba(255,0,0,0.2) "
    />
  )))
  {newRect && (
    <Rect
      x={newRect.x}
      y={newRect.y}
      width={newRect.width}
      height={newRect.height}
      stroke="yellow"
      dash={[10, 5]}
    />
  )}
</Layer>
</Stage>

```

## 5. Penyimpanan Anotasi

```

const handleSave = async () => {
  const geojson = convertToGeoJSON();
  const res = await fetch(
    `http://localhost:5000/api/annotations/${buildingId}/${
      imageName}.geojson`,
    {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify(geojson),
    }
  );

  if (res.ok) {
    alert("Annotation saved!");
    onClose();
  }
};

```

## Lampiran D Program untuk *preview* anotasi gambar

### 1. Inisialisasi dan Skalasi Gambar

```
const [img] = useImage(imageUrl);
const [scale, setScale] = useState(1);
useEffect(() => {
  if (!img) return;
  const maxWidth = 900;
  const maxHeight = 600;
  const scaleW = maxWidth / img.width;
  const scaleH = maxHeight / img.height;
  setScale(Math.min(scaleW, scaleH, 1));
}, [img]);
```

### 2. Proyeksi Koordinat GeoJSON ke Pixel

```
useEffect(() => {
  const loadAndProject = async () => {
    const geo = await fetch(geojsonUrl).then((res) =>
      res.json());
    const gps = await exifr.parse(imgBlob, { gps: true
    });
    const latLonToPixel = ([lon, lat]) => {
      const dx = (lon - centerLon) * 111320 *
        Math.cos(centerLat * Math.PI / 180);
      const dy = (centerLat - lat) * 110540;
      return {
        x: imageWidth / 2 + dx / meterPerPixel,
        y: imageHeight / 2 + dy / meterPerPixel,
      };
    };
    const pixelRects = geo.features.map(f => {
      const coords = f.geometry.coordinates[0].map(c =>
        latLonToPixel(c));
      return {
        x: Math.min(...coords.map(c => c.x)),
        y: Math.min(...coords.map(c => c.y)),
```

```

        width: Math.max(...coords.map(c => c.x)) -
Math.min(...coords.map(c => c.x)),
        height: Math.max(...coords.map(c => c.y)) -
Math.min(...coords.map(c => c.y))
    };
    });
    setRects(pixelRects);
};
if (img) loadAndProject();
}, [geojsonUrl, imageUrl, img]);

```

### 3. Penempatan Anotasi pada Model 3D Bangunan

```

const handlePlaceOnBuilding = async () => {
    const geojsonToSend =
JSON.parse(JSON.stringify(rawGeoJSON));
    geojsonToSend.features.forEach((feature) => {
        feature.properties.imageName = imageName;
        feature.properties.buildingId = buildingId;
    });
    const res = await
fetch("http://localhost:5000/api/placeAnnotationOnBuild
ing", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify(geojsonToSend),
    });
    if (res.ok) alert("Annotation polygon placed on
building!");
};

```

### 4. Komponen Visualisasi Anotasi

```

return (
    <Stage
        width={img.width * scale}
        height={img.height * scale}

```

```

    scale={{ x: scale, y: scale }}
  >
  <Layer>
    <KonvaImage image={img} />
    {rects.map((r, idx) => (
      <Rect
        key={idx}
        x={r.x}
        y={r.y}
        width={r.width}
        height={r.height}
        stroke="lime"
        fill="rgba(0,255,0,0.2)"
      />
    ))}
  </Layer>
</Stage>
);

```

## 5. Antarmuka Pengguna

```

<div style={{ padding: 10, background: "#f0f0f0" }}>
  <strong>View Annotation: {imageName}</strong>
  <div>
    <button onClick={handlePlaceOnBuilding}>
      Place All Windows
    </button>
    <button onClick={onClose}>X Close</button>
  </div>
</div>

```

## Lampiran E Program untuk proyeksi model 3D jendela pada bangunan

### 1. Inisialisasi dan Pemrosesan Data Dasar

```
const loadReferenceMap = async () => {
  const response = await fetch("/reference_lab.txt");
  const text = await response.text();
  const refMap = {};
  // ... parsing data
  return refMap;
};

const loadWallPolygons = async (uid) => {
  const response = await
fetch("/building_with_parts.geojson");
  const data = await response.json();
  // ... ekstrak dinding bangunan
  return walls;
};
```

### 2. Algoritma Penempatan Window

```
features.sort((a, b) => a._avgLat - b._avgLat);
let rowGroups = [];
for (const feature of features) {
  // ... logika pengelompokan baris
}
for (const row of rowGroups) {
  row.items.sort((a, b) => a._avgLon - b._avgLon);
}
```

### 3. Perhitungan Posisi dan Orientasi

```
if (wallPolygons.length) {
  const heading = computeHeadingFromWall(nearest);
  hpr = new HeadingPitchRoll(heading + headingOffset,
0, 0);
}
else if (orientation) {
```

```

    hpr = new HeadingPitchRoll(kappaRad, phiRad,
    omegaRad);
}
const wallAnchor = Cartesian3.fromDegrees(wallLon,
wallLat, zOffset);
const baseMatrix =
Transforms.headingPitchRollToFixedFrame(wallAnchor,
hpr);
const layoutShift = Matrix4.fromTranslation(new
Cartesian3(xOffset, 0, 0));

```

#### 4. Visualisasi 3D Window

```

const box = new BoxGeometry({
  vertexFormat: VertexFormat.ALL,
  maximum: new Cartesian3(width / 2, FIXED_DEPTH / 2,
height / 2),
  minimum: new Cartesian3(-width / 2, -FIXED_DEPTH / 2,
-height / 2)
});
const instance = new GeometryInstance({
  geometry: box,
  modelMatrix,
  attributes: {
    color:
ColorGeometryInstanceAttribute.fromColor(Color.LIME.wit
hAlpha(0.85))
  }
});
viewer.scene.primitives.add(
  new Primitive({
    geometryInstances: instance,
    appearance: new PerInstanceColorAppearance({
translucent: true })
  })
);

```

## 5. Penanganan Ketinggian Dinamis

```
const getFeatureZRange = (feature) => {  
  // ... ekstrak nilai Z min/max  
};  
  
const zRanges = row.items.map(getFeatureZRange);  
const currentMaxZ = Math.max(...zRanges.map(r =>  
  r.maxZ));  
const dynamicZGap = nextMinZ - currentMaxZ;  
currentZ += Math.max(dynamicZGap, 0.003 *  
  SCALE_FACTORZ);
```