

GeoCAD Bridge: Aplikasi Desktop Berbasis Flutter Untuk Otomatisasi Konversi Data GIS ke Format CAD dengan Validasi Spasial dan Presisi Koordinat

Putri Wulan Sari^{*1}, Setyoningsih Wibowo^{*2}

¹Teknik Informatika, Universitas PGRI Semarang, Kota Semarang, putriwulansarigb24@gmail.com

²Teknik Informatika, Universitas PGRI Semarang, Kota Semarang, setyoningsihwibowo@upgrisac.id

* putriwulansarigb24@gmail.com

Abstract.

Moving data from a Geographic Information System (GIS) into Computer-Aided Design (CAD) is a recurring need across engineering, infrastructure, surveying, urban planning, and technical design worldwide. This task is still often done manually with conventional desktop software an approach that is slow, error-prone, and dependent on specialized skills that not every user has. This study develops GeoCAD Bridge, a cross-platform Flutter desktop application that automatically converts GIS data (GeoPackage, Shapefile, GeoJSON) into CAD format (DXF/DWG) while preserving spatial accuracy at a 1:1 coordinate scale and supporting multiple coordinate reference systems (WGS84 and several UTM zones). The application was built using the Agile method with the Scrum framework across five sprints over twenty weeks, and modeled with the Unified Modeling Language. Its main features include multi-format data reading, automatic spatial validation (geometry, attributes, and Coordinate Reference System), three export modes (Full Export, Selective Export, and Auto-Layering), result preview, and a seven-step workflow-based interface with a collapsible sidebar, a stepper indicator, light/dark themes, and bilingual support. Testing on ten representative datasets ranging from 500 KB to 50 MB shows an average conversion speed of 3–5 seconds per MB for Shapefiles and 5–8 seconds per MB for GeoPackages, with a 95% detection rate for geometric errors and 90% for attribute incompleteness, while all CRS mismatches were correctly detected. Compared with the manual method, which takes 30–45 minutes per 5 MB dataset, the application cut conversion time by up to 70% while making technical documentation more consistent and accurate.

Keywords: GIS; CAD; spatial data interoperability; coordinate transformation; Flutter; DXF

Abstrak

Memindahkan data dari Sistem Informasi Geografis (SIG) ke Computer-Aided Design (CAD) adalah kebutuhan yang terus berulang di banyak bidang mulai dari rekayasa, infrastruktur, survei, perencanaan kota, hingga desain teknik di berbagai belahan dunia. Selama ini pekerjaan itu kerap dilakukan secara manual dengan perangkat lunak desktop konvensional, dan caranya memakan waktu, mudah keliru, serta menuntut keahlian khusus yang tidak dimiliki semua orang. Penelitian ini mengembangkan GeoCAD Bridge, aplikasi desktop lintas platform berbasis Flutter yang mengonversi data GIS (GeoPackage, Shapefile, GeoJSON) ke format CAD (DXF/DWG) secara otomatis sambil menjaga akurasi spasial pada skala koordinat 1:1 dan mendukung beragam sistem referensi koordinat (WGS84 serta beberapa zona UTM). Aplikasi dibangun dengan metode Agile berkerangka Scrum dalam lima sprint selama dua puluh minggu, dan dimodelkan memakai Unified Modeling Language. Fitur utamanya meliputi pembacaan data multi-format, validasi spasial otomatis (geometri, atribut, dan Sistem Referensi Koordinat), tiga mode ekspor (Full Export, Selective Export, dan Auto-Layering), pratinjau hasil, serta antarmuka berbasis workflow tujuh langkah yang dilengkapi sidebar

collapsible, indikator stepper, tema terang/gelap, dan dukungan dwibahasa. Pengujian pada sepuluh dataset representatif berukuran 500 KB hingga 50 MB menunjukkan kecepatan konversi rata-rata 3–5 detik per MB untuk Shapefile dan 5–8 detik per MB untuk GeoPackage, dengan tingkat deteksi 95% untuk kesalahan geometri dan 90% untuk atribut yang tidak

lengkap, sementara seluruh ketidaksesuaian CRS berhasil dikenali. Dibandingkan cara manual yang memakan 30–45 menit per set data 5 MB, aplikasi ini memangkas waktu konversi hingga 70% sekaligus membuat dokumentasi teknis lebih konsisten dan akurat.

Kata kunci: GIS; CAD; interoperabilitas data spasial; transformasi koordinat; Flutter; DXF

1. Pendahuluan

Data spasial kini menjadi tulang punggung beragam pekerjaan rekayasa dan perencanaan dari survei pemetaan, perencanaan kota, pengelolaan utilitas, sampai desain infrastruktur teknik. Untuk mengolahnya, ada dua jenis perangkat lunak yang dominan namun bertolak dari orientasi yang berbeda: Sistem Informasi Geografis (SIG) yang menekankan analisis spasial dan basis data geografis, serta Computer Aided Design (CAD) yang mengutamakan presisi geometri dan dokumentasi desain teknik [1]. Karena keduanya saling melengkapi, memindahkan data dari GIS ke CAD menjadi kebutuhan yang muncul berulang di banyak alur kerja teknis: hasil pemetaan lapangan yang tersimpan di GIS sering kali perlu dibawa ke lingkungan CAD untuk keperluan perancangan, dokumentasi, dan koordinasi proyek.

Data spasial telah menjadi tulang punggung dalam beragam aktivitas rekayasa dan perencanaan, mulai dari survei pemetaan, perencanaan kota, manajemen utilitas, hingga desain infrastruktur teknik. Dalam ekosistem perangkat lunak, dua kelas alat mendominasi pengolahan data tersebut dengan orientasi yang berbeda: Sistem Informasi Geografis (SIG) yang berorientasi pada analisis spasial dan basis data geografis, serta Computer Aided Design (CAD) yang berorientasi pada presisi geometri dan dokumentasi desain teknik [1]. Karena kedua domain ini saling melengkapi, pertukaran data dari GIS ke CAD menjadi kebutuhan rutin di banyak alur kerja teknis: data hasil pemetaan lapangan yang dikelola dalam GIS kerap harus dibawa ke lingkungan CAD untuk perancangan, dokumentasi, dan koordinasi proyek.

Di titik perpindahan inilah persoalan sebenarnya muncul. Akarnya cukup mendasar: GIS dan CAD tumbuh sendiri-sendiri selama bertahun-tahun dengan format data, istilah, dan semantik yang berbeda, sehingga keduanya tidak mudah disatukan dan kerap memaksa pemindahan data secara manual antar sistem [2]. Tidak heran, usaha menjembatani keduanya kemudian berkembang menjadi bidang kajian tersendiri, baik pada tingkat data maupun semantik [3]. Ketika konversi GIS ke CAD dikerjakan secara manual dengan perangkat lunak konvensional, prosesnya lambat, gampang keliru, dan menuntut keahlian khusus yang tidak dimiliki semua pengguna. Masalah yang lazim terjadi pun beragam mulai dari pergeseran posisi karena sistem referensi koordinat yang tidak cocok, atribut yang hilang saat dipetakan ke entitas CAD, hingga distorsi skala. Celaknya, ketidakakuratan seperti ini bisa merembet menjadi kesalahan perancangan yang ongkos waktu dan biayanya tidak kecil.

Beberapa penelitian sebelumnya sudah memanfaatkan GIS untuk mengelola aset dan infrastruktur, tetapi sebagian besar berhenti pada visualisasi dan pengelolaan data. Kunang dan Zuhriyadi, misalnya, menggunakan GIS untuk memetakan jaringan distribusi dan gardu induk lewat ArcGIS [4]. Djatnika dkk. memetakan aset per simpul dengan Q-GIS dan Q-Field demi data lapangan yang lebih akurat [5]. Kurniawan

membangun sistem informasi aset berbasis Web GIS memakai metode Extreme Programming [6], sementara Firdaus dkk. menyusun sistem inventarisasi aset yang juga berbasis Web GIS [7]. Karya-karya itu memangmemperkuat sisi hulu pengelolaan data spasial, namun keluarannya berhenti pada peta digital dan basis data atribut belum sampai mengubah data menjadi dokumen teknik CAD.

Pada sisi konversi, beberapa pendekatan juga sudah dilaporkan. Liu dkk. melakukan konversi data CAD ke GIS secara batch dengan perkakas spatial ETL Feature Manipulation Engine (FME) yang bertumpu pada pemetaan semantik [8]. Buddhika dkk. membuat aplikasi desktop untuk transformasi koordinat batch antarsistem dan menguji hasilnya terhadap perangkat mapan seperti ArcGIS [9]. Pendekatan-pendekatan ini terbukti efektif, tetapi umumnya bergantung pada perkakas berbayar atau hanya menggarap arah CAD ke GIS, dan belum menyatukan validasi integritas data dengan pelestarian presisi koordinat dalam satu alur konversi GIS ke CAD yang bisa dijalankan tanpa keahlian khusus.

Dari seluruh tinjauan itu tampak satu celah yang belum terjawab: belum ada perangkat yang khusus mengotomatiskan konversi data GIS ke CAD lintas domain, sekaligus dilengkapi validasi spasial untuk menjamin keutuhan data sebelum konversi dan menjaga presisi koordinat. Padahal justru di tahap konversi inilah beban kerja manual dan risiko kesalahan paling besar, apa pun domain penerapannya. Penelitian ini mencoba mengisi celah tersebut dengan memadukan pembacaan data multi format, validasi spasial bertahap (geometri, atribut, dan Sistem Referensi Koordinat/CRS), serta pelestarian presisi koordinat 1:1 melalui pipeline transformasi matematis semuanya dikemas dalam antarmuka berbasis alur kerja (workflow) yang dapat digunakan tanpa harus menguasai GIS secara mendalam.

Tujuan penelitian ini adalah merancang dan membangun GeoCAD Bridge, aplikasi desktop lintas platform berbasis Flutter yang mengonversi data GIS (GeoPackage, Shapefile, GeoJSON) ke format CAD (DXF/DWG) secara otomatis, akurat, dan efisien, sekaligus menilai kinerjanya dari sisi kecepatan, akurasi deteksi kesalahan, dan penghematan waktu dibanding cara manual. Aplikasi ini dikembangkan dengan metode Agile Scrum agar umpan balik bisa diperoleh dengan cepat selama proses. Bagian-bagian berikutnya memaparkan metode pengembangan, hasil pengujian, dan pembahasannya.

2. Metode

Bagian Metode ditulis berdasarkan persoalan konversi data GIS ke CAD diselesaikan, mencakup sumber data, perangkat pengembangan, metode pengembangan sistem, perancangan model, hingga algoritma inti konversi dan validasi. Urutan pemaparannya sengaja dijaga agar selaras dengan urutan pembahasan hasil pada Bagian 3. Berikut merupakan sumber data, perangkat pengembangan, metode pengembangan sistem, perancangan model, algoritma inti konversi dan validasi :

2.1. Sumber dan Karakteristik Data

Pengujian memakai sepuluh dataset spasial representatif berukuran 500 KB hingga 50 MB dalam format Shapefile (.shp), GeoPackage (.gpkg), dan GeoJSON (.geojson). Dataset ini mencakup beragam tipe geometri, titik, garis, dan poligon, beserta atributnya, dan mewakili kasus penggunaan yang umum dijumpai seperti jaringan utilitas, infrastruktur transportasi, dan batas wilayah. Kebutuhan fungsional maupun non-fungsional sistem ditetapkan dengan menelaah alur kerja konversi GIS ke CAD yang biasa dikerjakan secara manual sekaligus kendala teknis yang menyertainya.

2.2. Perangkat Pengembangan

Pengembangan dilakukan pada laptop dengan prosesor Intel Core i7 generasi ke-11, RAM 16 GB, penyimpanan SSD 512, dan sistem operasi Windows 11. Kerangkakerja

utama yang digunakan adalah Flutter dengan bahasa pemrograman Dart, yang dipilih karena kemampuannya menghasilkan antarmuka pengguna responsif dan modern dari satu basis kode lintas platform [10], [11]. Pengelolaan state aplikasi menggunakan pustaka Provider, navigasi antarhalaman menggunakan GoRouter dengan dukungan deep linking, dan persistensi data lokal menggunakan basis data SQLite (sqflite). Transformasi sistem koordinat ditangani menggunakan pustaka proj4dart yang mendukung WGS84 dan beberapa zona UTM, sedangkan pembacaan berkas dan ekspor laporan didukung oleh file picker dan pustaka excel. Tabel 1 merangkum komponen utama tumpukan teknologi yang digunakan. Tabel 1 merangkum komponen utama tumpukan teknologi yang digunakan.

Tabel 1. Tumpukan teknologi aplikasi GeoCAD Bridge

Lapisan	Komponen	Fungsi
Antarmuka	Flutter, Material Design 3	Penyajian UI desktop lintas platform
Bahasa	Dart	Bahasa pemrograman utama
Manajemen state	Provider	Pengelolaan state reaktif
Navigasi	GoRouter	Routing berbasis URL dan deep linking
Pemrosesan spasial	proj4dart	Transformasi sistem koordinat (CRS)
Penyimpanan	SQLite (sqflite)	Persistensi sesi, layer, dan riwayat konversi
Berkas dan laporan	file_picker, excel	Impor berkas GIS dan ekspor laporan validasi

2.3. Metode Pengembangan Sistem

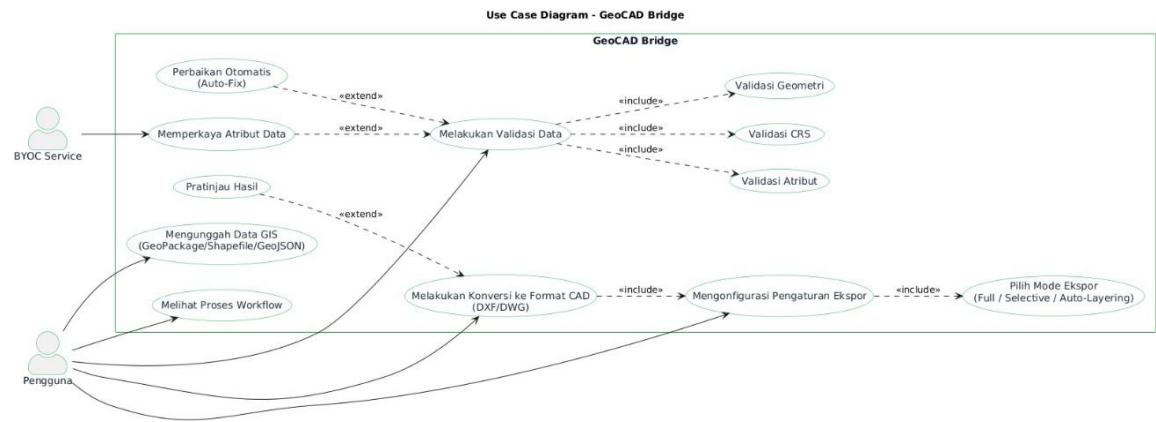
Pengembangan menggunakan metode Agile dengan kerangka kerja Scrum, yang dipilih karena memerlukan umpan balik cepat serta fleksibilitas penyesuaian fitur terhadap kebutuhan yang dinamis [12]. Proses dibagi menjadi lima sprint selama delapan minggu, dengan setiap sprint diakhiri review untuk evaluasi dan penyesuaian kebutuhan. Rincian sprint ditunjukkan pada Tabel 2.

Tabel 2. Rincian sprint pengembangan

Sprint	Durasi	Fokus Utama
Sprint 1	Minggu 1-4	Pembuatan struktur dasar aplikasi, routing, dan tata letak utama.

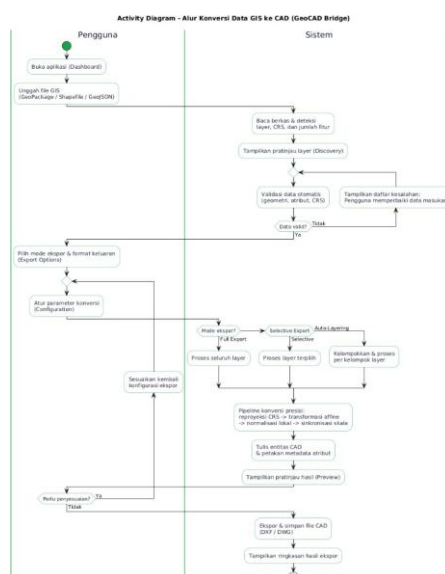
Sprint 2	Minggu 5-8	Implementasi fungsi pembacaan data GIS dan parser.
Sprint 3	Minggu 9-12	Pembangunan modul validasi data spasial.
Sprint 4	Minggu 13-16	Implementasi fungsi konversi ke CAD dan pratinjau.
Sprint 5	Minggu 17-20	Pengujian, perbaikan bug,dan finalisasi.

2.4. **Perancangan Model Sistem**
 Perancangan sistem dimodelkan menggunakan Unified Modeling Language (UML).



Gambar 1. Use Case Diagram

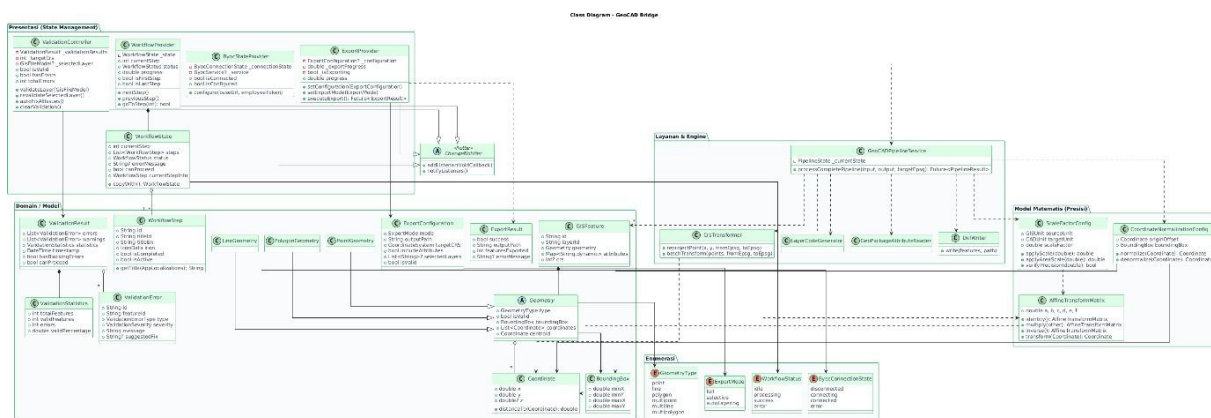
Use Case Diagram (Gambar 1) memetakan interaksi antara aktor Pengguna dengan sistem melalui lima use case utama: mengunggah data GIS, melihat proses workflow,



Gambar 2. Activity Diagram

melakukan validasi data, mengonfigurasi pengaturan ekspor, dan melakukan konversike format CAD; aktor BYOC Service berperan memperkaya atribut data secara otomatis. ke format CAD; aktor BYOC Service berperan memperkaya atribut data secara otomatis.

Activity Diagram (Gambar 2) memodelkan alur konversi mulai dari unggahberkas, identifikasi layer dan CRS, validasi otomatis, konfigurasi parameter ekspor, hingga konversi dengan titik keputusan yang mengembalikan pengguna ke perbaikandata bila validasi gagal.perbaikan data bila validasi gagal.



Gambar 3. Class Diagram

Class diagram (Gambar 3) Memetakan struktur kelas utama, yaitu WorkflowProvider, ValidationController, ExportProvider, dan ByocStateProvider beserta relasinya dengan entitas model.

Aplikasi disusun dengan arsitektur modular berlapis yang memisahkan tanggung jawab menjadi lapisan antarmuka (screens dan widgets), lapisan provider (manajemen state reaktif), lapisan layanan/aplikasi (orquestrasi pipeline konversi), serta lapisan data dan infrastruktur (parser GIS, penulis DXF, dan basis data). Pipeline konversi mengikuti urutan: masukan → validasi → transformasi koordinat → pemetaan atribut → ekspor. Antarmuka mengadopsi Material Design 3 dengan sidebar collapsible di kiri, top bar berisi indikator workflow stepper, area konten utama di tengah, dan status bardi

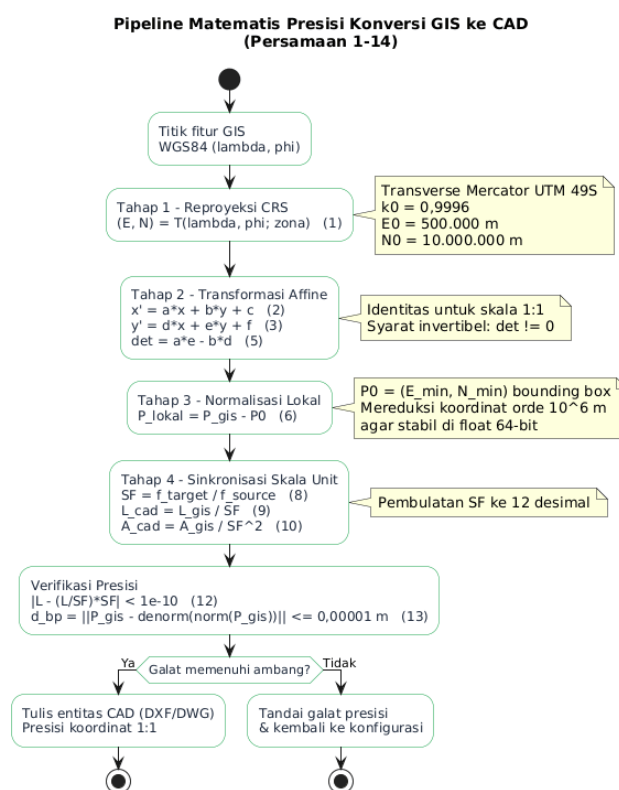
bawah, serta mendukung mode terang/gelap dan dua bahasa (Indonesia/Inggris).

2.5. Algoritma Konversi Koordinat dan Validasi

Untuk mencegah pergeseran posisi (map shifting) selama konversi, sistem mendeteksi CRS masukan, menetapkan CRS keluaran, lalu mentransformasikansetiap titik geometri menggunakan transformasi PROJ dengan presisi floating point 64bit dan toleransi 0,00001 meter. Setiap atribut fitur GIS dipetakan ke metadata entitasCAD sehingga seluruh informasi atribut dipertahankan pada keluaran. Modul validasi dijalankan sebelum konversi dan memeriksa tiga aspek secara berurutan: validitas geometri, kelengkapan dan konsistensi atribut, serta kesesuaian CRS [20], [21]. Bila ditemukan isu, sistem menampilkan ringkasan statistik dan daftar kesalahan agar pengguna memperbaiki data sebelum melanjutkan ke tahap ekspor.

2.6. Model Matematis Presisi Konversi GIS ke CAD

Presisi keluaran CAD pada skala 1:1 dijamin melalui empat tahap transformasi matematis yang dijalankan secara berurutan terhadap setiap titik koordinat fitur: (a) reprojeksi sistem koordinat, (b) transformasi affine, (c) normalisasi ke ruang koordinat lokal CAD, dan (d) sinkronisasi faktor skala unit, diakhiri tahap verifikasi presisi



Gambar 4. Pipeline Diagram

Keseluruhan alur perhitungan tersebut divisualisasikan pada Gambar 4. Seluruh perhitungan dilakukan dalam presisi floating point 64 bit (IEEE 754) dengan satuan dasar meter (SI).

2.7. Reprojeksi sistem koordinat

Data geografis WGS84 (EPSG:4326) berupa pasangan bujur–lintang (λ , ϕ) direproyeksikan ke sistem terproyeksi Universal Transverse Mercator (UTM). Aplikasi mendukung beberapa zona UTM (antara lain 48S, 49S, 50S, dan 49N); sebagai contoh perhitungan digunakan Zona 49S (EPSG:32749). Reproyeksi dinyatakan sebagaifungsi proyeksi Transverse Mercator T pada Persamaan 1:

$$(E, N) = T(\lambda, \phi; \text{zona})$$

dengan E (easting) dan N (northing) dalam meter. Proyeksi menggunakan parameterstandar UTM, yaitu faktor skala pada meridian sentral $k_0 = 0,9996$, false easting $E_0 = 500.000$ m, dan false northing $N_0 = 10.000.000$ m (belahan bumi selatan),sebagaimana dirangkum pada Tabel 3.

2.8. Transformasi affine

Transformasi affine. Setiap titik (x , y) ditransformasikan menggunakan matriksaffine enam parameter (a , b , c , d , e , f) seperti pada Persamaan 2 dan 3:

$$x' = a \cdot x + b \cdot y + c$$

$$y' = d \cdot x + e \cdot y + f$$

Dalam bentuk matriks homogen, transformasi tersebut dinyatakan pada Persamaan 4:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matriks ini menyatukan operasi penskalaan ($a = s_x$, $e = s_y$), rotasi sudut θ ($a = \cos \theta$, $b = \sin \theta$, $d = -\sin \theta$, $e = \cos \theta$), dan translasi ($c = t_x$, $f = t_y$) dalam satu komposisi. Untuk konversi presisi 1:1 tanpa distorsi, digunakan matriks identitas ($a = e = 1$; $b = c = d = f = 0$). Keterbalikan (invertibilitas) matriks diuji melalui determinan pada Persamaan 5:

$\det = a \cdot e - b \cdot d$, dengan syarat $\det \neq 0$

2.9. Normalisasi koordinat lokal

Koordinat UTM bernilai besar (orde 10^6 m) rentan terhadap floating-point drift saat dirender di CAD. Untuk menjaga stabilitas numerik, setiap titik digeser terhadap titik asal lokal $P_0 = (E_{\min}, N_{\min})$ yang diambil dari sudut minimum bounding box dataset,seperti pada Persamaan 6:

$$P_{\text{lokal}} = P_{\text{gis}} - P_0$$

Proses balik (denormalisasi) untuk verifikasi back projection diberikan padaPersamaan 7:

$$P_{\text{gis}} = P_{\text{lokal}} + P_0$$

2.10. Sinkronisasi faktor skala unit

Perbedaan satuan antara data GIS dan target CAD diselaraskan menggunakan faktor skala SF yang dihitung melalui meter sebagai satuan jembatan, sebagaimana Persamaan 8:

$$SF = \frac{f_{target}}{f_{source}}$$

dengan f_{target} dan f_{source} adalah faktor konversi unit target dan asal terhadap meter (Tabel 4). Nilai SF dibulatkan ke 12 angka desimal untuk menekan floating-point

$$L_{cad} = \frac{L_{gis}}{SF}$$

$$A_{cad} = \frac{A_{gis}}{SF^2}$$

drift. Transformasi dimensi linear (mis. panjang segmen) dan dimensi luas (mis. area poligon) diberikan pada Persamaan 9 dan 10:

Konversi nilai antarunit secara umum dinyatakan pada persamaan 11:

$$V_{cad} = \frac{(V_{gis} \cdot f_{source})}{f_{target}}$$

2.11. Verifikasi presisi

Integritas hasil diuji dengan dua kriteria. Pertama, galat round trip skala (Persamaan 12) harus lebih kecil dari ambang ε :

$$\left| L - \left(\frac{L}{SF} \right) \cdot SF \right| < \varepsilon$$

$$\varepsilon = 10^{-10}$$

Kedua, galat back projection antara titik asal dan hasil normalisasi denormalisasi (Persamaan 13) tidak boleh melampaui toleransi spasial τ :

$$d_{bp} = \|P_{gis} - \text{denorm}(\text{norm}(P_{gis}))\| \leq \tau, \tau = 0,00001 \text{ m}$$

Panjang segmen geometri dihitung dengan jarak Euclidean tiga dimensi pada Persamaan 14:

$$d = \sqrt{[(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2]}$$

Parameter proyeksi dan ambang presisi yang digunakan dirangkum pada Tabel 3, sedangkan faktor konversi unit ditunjukkan pada Tabel 4.

Tabel 3. Parameter reproyeksi UTM dan ambang presisi konversi

Parameter	Simbol	Nilai
Zona UTM (Contoh)	-	49S (EPSG:32749)
Meridian sentral	λ_0	111° BT
Faktor skala meridian	k_0	0,9996
False easting	E_0	500.000 m
False northing (selatan)	N_0	10.000.000 m
Presisi floating point	-	64 bit (IEEE 754)
Pembulatan faktor skala	-	12 angka desimal
Ambang galat skala	ϵ	10^{-10}
Toleransi spasial	τ	0,00001 m

Tabel 4. Faktor konversi unit terhadap meter (satuan jembatan)

Domain	Unit	Faktor ke meter
GIS	Meter	1,000000
GIS	Kaki	0,304800
GIS	Kilometer	1000,000000
GIS	Mil	1609,344000
GIS	Derajat (estimasi)	111319,900000
CAD	Milimeter	0,001000
CAD	Sentimeter	0,010000
CAD	Meter	1,000000
CAD	Inci	0,025400
CAD	Kaki	0,304800

3. Hasil dan Pembahasan

3.1. Penyajian Hasil

Hasil implementasi berupa aplikasi desktop GeoCAD Bridge yang berfungsi penuh dengan antarmuka berbasis workflow tujuh langkah: Dashboard, Discovery (unggah dan pratinjau layer), Validation, Export Options, Configuration, Preview, dan Export. Sidebar navigasi collapsible menyediakan akses cepat ke Dashboard, Workflow Guide, Settings, dan About, sedangkan top bar stepper menampilkan posisi pengguna dalam alur kerja. Pengguna dapat memilih tiga mode ekspor Full Export, Selective Export, dan Auto Layering serta formatkeluaran DXF atau DWG dengan presisi koordinat 1:1.

Pengujian validasi numerik dilakukan terhadap titik sampel untuk membuktikan presisi pipeline matematis pada Bagian 2.6. Sebuah titik dengan koordinat geografis ($110,4167^{\circ}$ BT; $-6,9667^{\circ}$ LS) direproyeksikan ke UTM Zona 49S, dinormalisasi terhadap titik asal lokal, lalu diskalakan ke unit CAD meter ($SF = 1$). Hasil round trip (denormalisasi dan reproyeksi balik) menunjukkan galat di bawah ambang yang ditetapkan pada Persamaan (12) dan (13). Ringkasan verifikasi disajikan pada Tabel 5.

Tabel 5. Verifikasi presisi pipeline transformasi koordinat (titik sampel)

Tahap (Persamaan)	Masukan Keluaran	Galat
isi UTM 49S(1)	(110,4167°; -6,9667°)	(± 433.000 m; ± 9.236.000 m)
Normalisasi lokal (6)	$P_0=(432.000; 9.235.000)$	(433.000;9.236.000) (1.000; 1.000)
Skala unit m→m (9)	$SF=1$ 1.000,000 m	1.000,000 m
Round trip skala (12)	1.000,000 m 1.000,000 m	$< 10^{-10}$
Back projection (13)	P_{gis} denorm(norm(P_{gis}))	$\leq 0,00001$ m

Pengujian kecepatan konversi terhadap sepuluh sampel data menunjukkan kinerja yang konsisten lintas format. Aplikasi memproses Shapefile pada kecepatan rata-rata 4 detik per MB dan GeoPackage pada 6,4 detik per MB. Ringkasan hasil pengujian kecepatan disajikan pada Tabel 6. Seluruh berkas keluaran berhasil diverifikasi dapat dibuka di perangkat lunak CAD (AutoCAD) dengan geometri yang terjaga.

Tabel 6. Hasil pengujian kecepatan konversi

Jenis data	ran rata-rata	Waktu rata-rata	Kecepatan per MB
Shapefile (.shp)	5 MB	20 detik	4,0 detik/MB
oPackage(.gpkg)	5 MB	32 detik	6,4 detik/MB
GeoJSON (.geojson)	3 MB	18 detik	6,0 detik/MB

Hasil pengujian akurasi modul validasi disajikan pada Tabel 7. Validasi geometri

mendeteksi 95% kesalahan bentuk, validasi atribut mendeteksi 90% ketidaklengkapan kolom wajib, dan validasi CRS mendeteksi seluruh (100%) ketidaksesuaian sistem referensi koordinat pada data uji.

Tabel 7. Hasil pengujian akurasi validasi data spasial

Aspek validasi	Objek pemeriksaan	Tingkat deteksi
Geometri	Validitas bentuk, overlap, gap	95%
Atribut	Kelengkapan kolom wajib	90%
CRS	Kesesuaian sistem referensi koordinat	100%

Perbandingan waktu proses antara metode manual dan GeoCAD Bridge untuk satu set data berukuran 5 MB disajikan pada Tabel 8. Proses manual membutuhkan 30–45 menit, sedangkan aplikasi menyelesaikan tugas yang sama rata-rata dalam 20 detik.

Tabel 8. Perbandingan metode manual dan GeoCAD Bridge (data 5 MB)

Aspek	Metode manual	GeoCAD Bridge
Waktu proses	30–45 menit - lebih	± 20 detik
Konsistensi hasil	Bervariasi antaroperator	Seragam (otomatis)
Kebutuhan keahlian	Tinggi	Rendah
Penghematan waktu	-	Hingga 70%

3.2. Pembahasan

Apa yang terlihat pada Bagian 3.1 sebenarnya menjawab langsung apa yang ingin dicapai penelitian ini, yakni proses konversi GIS ke CAD yang cepat, akurat, sekaligus mudah dijalankan. Namun angka penghematan waktu hingga 70% pada Tabel 8 sebaiknya tidak dibaca semata-mata sebagai "lebih cepat". Yang sesungguhnya terjadi adalah perpindahan beban kerja: pekerjaan yang dulu menuntut keahlian teknis tinggi kini dipandu langkah demi langkah dalam tujuh tahap, sehingga tugas yang tadinya hanya bisa ditangani segelintir orang berkemampuan khusus dapat dikerjakan oleh anggota tim teknis yang lebih luas.

Jika dibandingkan dengan penelitian-penelitian sebelumnya, temuan ini lebih tepat dilihat sebagai pelengkap, bukan pengganti. Kunang dan Zuhriyadi [4] maupun Djatnika dkk. [5] sudah menunjukkan bagaimana GIS dapat menghadirkan visualisasi dan pengelolaan data spasial yang baik, sementara Kurniawan [6] dan Firdaus dkk. [7] menyorot kekuatan inventarisasi aset berbasis Web GIS. Semua karya itu menguatkan sisi hulu, yaitu bagaimana data dikumpulkan dan dikelola. Di sinilah GeoCAD Bridge mengambil peran yang berbeda: ia menggarap sisi hilir yang selama ini jarang disentuh, yakni mengubah data spasial menjadi dokumen teknik CAD yang langsung siap pakai. Perbedaan itu juga tampak terhadap penelitian yang memang berfokus pada konversi. Liu

dkk. [8] mengandalkan perkakas spatial ETL berbayar seperti FME, dan Buddhika dkk. [9] berhenti pada transformasi koordinat saja; GeoCAD Bridge justru merangkai validasi integritas data, transformasi koordinat, hingga ekspor CAD menjadi satu alur tunggal yang berdiri sendiri. Karena itu, kontribusi utama penelitian ini terletak pada penyatuan tahap validasi dan konversi dalam satu alur otomatis yang tidak terikat pada domain tertentu hal yang belum menjadi perhatian pada studi-studi terdahulu.

Angka deteksi kesalahan pada Tabel 7 menegaskan bahwa pengecekan sebelum konversi memang bekerja: data yang cacat tersaring lebih dulu sebelum sempat masuk ke tahap ekspor. Hal serupa juga didukung verifikasi numerik pada Tabel 5, yang menunjukkan bahwa rangkaian perhitungan pada Persamaan (1)–(14) sanggup menjaga presisi 1:1 galat round trip skala tetap di bawah 10^{-10} dan galat back projection di bawah 0,00001 m. Di balik angka itu, langkah normalisasi koordinat terhadap titik asal lokal (Persamaan 6) ternyata berperan besar. Koordinat UTM bernilai sangat besar, berada pada orde 10^6 m, dan tanpa pereduksian semacam ini keterbatasan mantissa floating point 64 bit akan memunculkan pembulatan yang di CAD terlihat sebagai pergeseran sehalus submilimeter. Deteksi CRS yang menyentuh 100% pun bukan sekadar angka bagus: ketidaksesuaian sistem referensi

koordinat adalah biang utama bergesernya posisi pada hasil konversi, dan menanganinya lewat transformasi Transverse Mercator (Persamaan 1) dengan toleransi 0,00001 meter inilah yang menjaga presisi tetap 1:1 sebagaimana dituntut dokumen teknik. Yang sedikit tertinggal adalah deteksi atribut di angka 90%, lebih rendah dibanding geometri. Penyebabnya cukup wajar: penamaan dan kelengkapan kolom pada data lapangan tidak selalu seragam. Temuan ini sejalan dengan pedoman tata kelola data geospasial yang menekankan pentingnya standarisasi metadata [22], sekaligus menjadi celah yang masih bisa diperbaiki pada pengembangan berikutnya.

Sepanjang pengembangan, ada dua kendala teknis yang rasanya perlu disampaikan apa adanya. Pertama, pustaka Flutter belum mampu membaca berkas DWG secara langsung; jalan keluarnya adalah menuliskan entitas CAD pada level format DXF yang sifatnya terbuka. Kedua, performa tampilan sempat menurun saat data sangat banyak, di atas 10.000 fitur, dan ini diatasi dengan menerapkan virtual scrolling serta lazy loading pada komponen tabel. Perbedaan kecepatan antarformat pada Tabel 6, di mana GeoPackage tampak lebih lambat dibanding Shapefile, sebenarnya masuk akal mengingat GeoPackage adalah kontainer berbasis SQLite yang menuntut kueri basis data tambahan setiap kali atribut dibaca. Dari sini terlihat bahwa kecepatan yang ada memang sudah memadai untuk kebutuhan sehari-hari, tetapi masih terbuka ruang untuk dioptimalkan khususnya pada jalur pembacaan berbasis basis data dan pemrosesan paralel ketika berhadapan dengan data yang benar-benar besar.

4. Kesimpulan

Penelitian ini berhasil melahirkan GeoCAD Bridge, aplikasi konversi data GIS ke CAD lintas platform berbasis Flutter yang mengubah format GeoPackage, Shapefile, dan GeoJSON menjadi DXF dan DWG secara otomatis dengan presisi koordinat 1:1. Sistem validasi spasial di dalamnya sanggup menangkap kesalahan geometri hingga 95%, ketidaklengkapan atribut hingga 90%, serta seluruh ketidaksesuaian CRS, sehingga peluang lolosnya data cacat dapat ditekan sejak awal. Rangkaian transformasi matematis yang dirumuskan pada Persamaan (1)–(14) terbukti menjaga presisi tetap 1:1, dengan galat round trip di bawah 10^{-10} dan galat back projection di bawah 0,00001 m. Antarmuka berbasis workflow tujuh langkah juga terbukti menuntun pengguna langkah demi langkah tanpa menuntut penguasaan GIS yang mendalam. Secara keseluruhan, pengujian memperlihatkan waktu konversi yang terpendek hingga 70% dibanding cara manual, disertai hasil yang lebih konsisten dan akurat. Kontribusi terpenting penelitian ini adalah

menyatukan tahap validasi dan konversi dalam satu alur otomatis yang tidak terikat domain tertentu, sehingga menutup celah hilir interoperabilitas data spasial yang selama ini belum tergarap. Ke depan, beberapa hal layak dilanjutkan: transformasi CRS yang lebih luas antarsistem koordinat, batch processing untuk mengonversi data berskala besar sekaligus, serta penguatan validasi atribut yang bersandar pada standardisasi metadata.

5. Referensi

- [1] E. Prahasta, *Sistem Informasi Geografis: Konsep-Konsep Dasar*. Bandung: Informatika, 2019.
- [2] R. Peachavanish, H. A. Karimi, B. Akinci, and F. Boukamp, "An ontological engineering approach for integrating CAD and GIS in support of infrastructure management," *Adv. Eng. Informatics*, vol. 20, no. 1, pp. 71–88, 2006, doi: 10.1016/j.aei.2005.06.001.
- [3] H. A. Karimi and B. Akinci, Eds., *CAD and GIS Integration*. Boca Raton, FL, USA: CRC Press, 2010.
- [4] S. O. Kunang and I. Zuhriyadi, "Sistem Informasi Geografis Jaringan Distribusi dan GarduInduk PLN di Kota Palembang," in *Prosiding Seminar Nasional Teknologi (Semnastek)*, 2017.
- [5] R. Djatnika, I. Darmawan, M. Hidayatullah, and A. Jaya, "Pemetaan Pelanggan Per GarduBasis Q-GIS dan Q-Field," *J. Inform. Teknol. Sains*, vol. 7, no. 2, pp. 492–498, 2025.
- [6] A. Kurniawan, "Implementasi Sistem Informasi Aset Berbasis Web GIS pada PT. PLN ULPPringsewu," *EXPERT J. Manaj. Sist. Inf. dan Teknol.*, 2025.
- [7] R. Firdaus, K. Muludi, and S. P. Sinurat, "Sistem Informasi Inventarisasi Aset Berbasis WebGIS di PT. PLN (Persero) Sektor Pembangunan Tarahan," *J. Komputasi*, vol. 3, no. 1, 2015.
- [8] Z. Liu, C. Jing, and X. Chen, "Files' conversion from CAD to GIS using spatial data conversion tools provided by FME," in *Proc. 2012 Int. Conf. Comput. Sci. Serv. Syst. (CSSS)*, Aug. 2012, pp. 1939–1942, doi: 10.1109/CSSS.2012.484.
- [9] W. Buddhika, K. Premawansa, T. R. Bandara, L. Samaranayake, V. Dayananda, C. Mudannayaka, and S. Priyadarshani, "Revolutionizing spatial data analysis: Unveiling a cutting-edge approach for batch coordinate transformation," *Geomatics*, 2023.
- [10] R. Damanik, B. Sinuraya, S. P. Sipayung, and P. Tarigan, *Dasar-dasar Pemrograman Flutter di Android*. Yayasan Kita Menulis, 2024.
- [11] S. Tjandra and G. S. Chandra, "Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang," *J. Inf. Syst. Graph. Hosp. Technol.*, vol. 2, no. 2, pp. 76–81, 2020.
- [12] Tukino and Suwarno, *Pendekatan Rekayasa Perangkat Lunak Berbasis Model Agile Scrum*. 2023.
- [13] I. K. Raharjana, *Pengembangan Sistem Informasi Menggunakan Metodologi Agile*. Yogyakarta: Deepublish, 2019.
- [14] A. Faisol, *Sistem Informasi Geografis*. Yogyakarta: Deepublish, 2022.
- [15] N. N. Supuwingsih and M. Rusli, *Sistem Informasi Geografis: Konsep Dasar & Implementasi*. Yogyakarta: Penerbit Andi, 2020.
- [16] F. Ramdani, *Ilmu Geoinformatika: Observasi hingga Validasi*. Malang: UB Press, 2021.
- [17] Jumadi, Danardono, and V. N. Fikriyah, *Sistem Informasi Geografis dan Aplikasinya di Bidang Geografi*. Surakarta: UMS Press, 2021.
- [18] E. B. Setiawan, *Sistem Informasi Geografis Berbasis Web Menggunakan Google Maps dan Mapbox API*. Bandung: Informatika, 2020.
- [19] B. J. M. Putra and A. Fu'adi, *Belajar Pemrograman Mobile dengan Flutter*. Wawasan Ilmu, 2024.
- [20] N. A. Tangge, S. R. Aji, and K. Aji, *Sistem Informasi Geografis Tingkat Dasar*.

Yogyakarta:Deepublish, 2023.

[21] Sularno and N. Azizah, Buku Ajar Metode Agile. CV ITTC Indonesia Publishing, 2026. Badan Informasi Geospasial, "Peraturan Kepala BIG Nomor 6 Tahun 2021 tentang Pedoman Teknis Tata Kelola Data Geospasial," 2021.