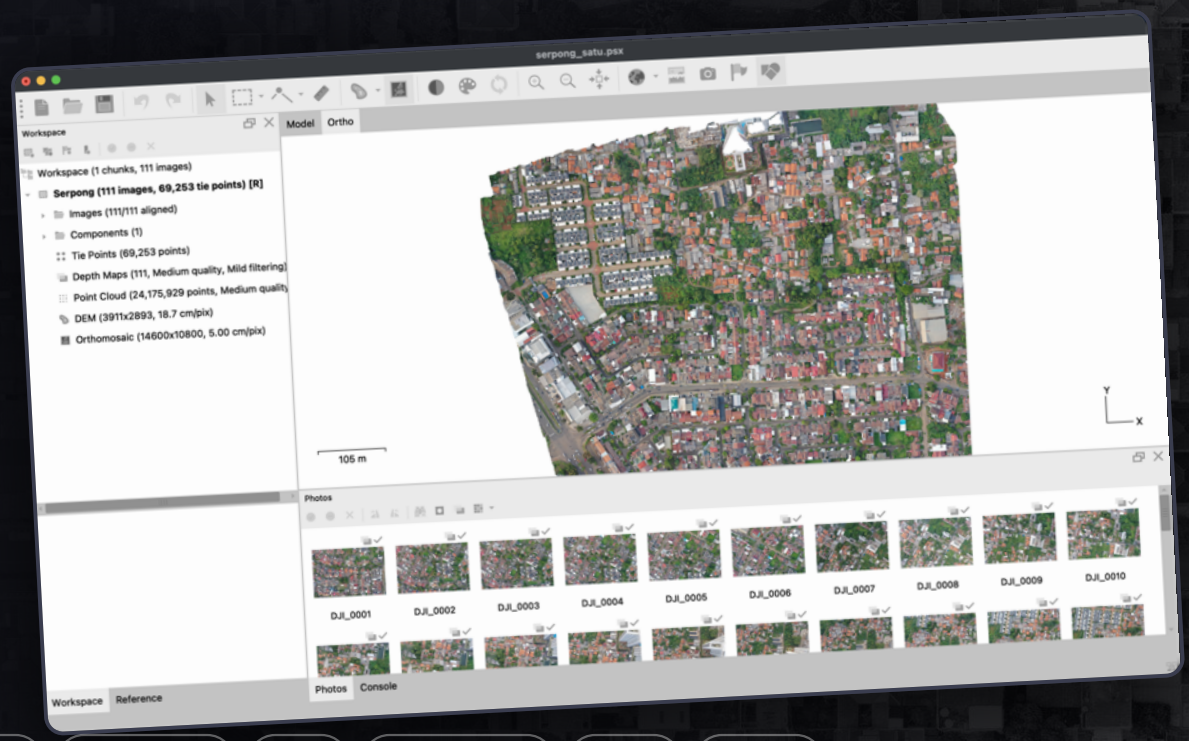
liupurnomo

EBOOK GRATIS

OLAH DATA DRONE DI AGISOFT METASHAPE

Panduan Kilat dari Foto Udara Menjadi Ortomosaik dan DSM



Align | Point Cloud | DEM | Orthomosaic | Batch | Python

Liu Purnomo, S.Kom.

liupurnomo.com

liupurnomo

Olah Data Drone di Agisoft Metashape

Panduan Kilat dari Foto Udara Menjadi Ortomosaik dan DSM

Penulis: Liu Purnomo, S.Kom.
Edisi pertama, November 2026

Ebook ini dibagikan gratis. Boleh disalin dan dibagikan seluas-luasnya dalam bentuk utuh tanpa diubah dan tanpa dipungut biaya. Tidak untuk diperjualbelikan.

Disarikan dari buku *Aerial Mapping* karya penulis. Data contoh berupa 111 foto DJI Mavic 2 Pro kawasan Serpong. Tangkapan layar diambil dari Agisoft Metashape Professional 2.1.3. Agisoft dan Metashape adalah merek dagang Agisoft LLC.

Situs: liupurnomo.com
Email: hi@liupurnomo.com



Kata Pengantar

Agisoft Metashape Professional adalah perangkat lunak fotogrametri yang paling banyak dipakai untuk mengolah foto drone di Indonesia, baik di instansi pemerintah, perguruan tinggi, maupun perusahaan survei. Banyak pemula membuka Metashape, mengklik semua tahap dengan pengaturan bawaan, lalu bingung mengapa hasilnya miring, berlubang, atau tidak tepat di peta.

Ebook ini disusun sebagai panduan kilat. Setiap tahap dijelaskan dengan tangkapan layar, parameter yang disarankan beserta alasannya, dan hasil nyata dari data contoh, yaitu 111 foto DJI Mavic 2 Pro kawasan Serpong. Isinya disarikan dari buku *Aerial Mapping* karya penulis, lalu diringkas menjadi lembar-lembar yang mudah dibuka kembali saat bekerja.

Halaman **Resep Cepat** di bagian akhir merangkum seluruh parameter dalam satu halaman. Cetak halaman tersebut dan tempel di dekat komputer pengolahan.

Jakarta, November 2026

Liu Purnomo, S.Kom.





Daftar Isi

Kata Pengantar 3

Daftar Gambar 5

Bagian I — Persiapan 6

Bab 01 — Alur Kerja dalam Satu Halaman 7

Bab 02 — Mengenal Antarmuka 9

Bab 03 — Pengaturan Preferences 11

Bagian II — Pengolahan Langkah demi Langkah 12

Bab 04 — Project dan Foto 13

Bab 05 — Sistem Koordinat dan Akurasi Kamera 14

Bab 06 — Kalibrasi Kamera dan Rolling Shutter 17

Bab 07 — Tahap 1, Align Photos 18

Bab 08 — Tahap 2, Build Point Cloud 20

Bab 09 — Tahap 3, Build DEM 21

Bab 10 — Tahap 4, Build Orthomosaic 22

Bab 11 — Ekspor GeoTIFF 24

Bab 12 — Laporan Pengolahan 26

Bagian III — Otomatisasi 28

Bab 13 — Batch Process 29

Bab 14 — Skrip Python 31

Bab 15 — Masalah yang Sering Muncul 32

Resep Cepat — seluruh parameter dalam satu halaman 33

Lampiran — Skrip batch_metashape.py 34

Tentang Penulis 37





Daftar Gambar

Gambar 02.1 Antarmuka Metashape setelah 111 foto dimasukkan. Workspace di kiri, tampilan Model di tengah, dan Photos di bawah 9

Gambar 03.1 Tiga tab Preferences yang perlu diperiksa 11

Gambar 04.1 Chunk Serpong berisi 111 foto. Lingkaran biru menandai posisi kamera dari GPS 13

Gambar 05.1 Panel Reference dengan koordinat kamera dalam derajat 14

Gambar 05.2 Reference Settings untuk data contoh 15

Gambar 05.3 Koordinat kamera setelah dikonversi ke UTM zona 48S 16

Gambar 06.1 Dialog Camera Calibration 17

Gambar 07.1 Pengaturan Align Photos 18

Gambar 07.2 Proses dan hasil Align Photos 19

Gambar 08.1 Pengaturan dan hasil Build Point Cloud 20

Gambar 09.1 Pengaturan dan hasil Build DEM 21

Gambar 10.1 Pengaturan dan hasil Build Orthomosaic 22

Gambar 11.1 Dialog ekspor dengan batas persegi panjang yang sama 24

Gambar 11.2 Ortofoto hasil ekspor, 620 × 470 m pada GSD 5 cm, tanpa bagian kosong 25

Gambar 12.1 Membuat laporan pengolahan 26

Gambar 13.1 Menyusun daftar job Batch Process 29

Gambar 13.2 Parameter job yang perlu diperiksa 30



Bagian I

PERSIAPAN

Mengenal alur, antarmuka, dan pengaturan yang perlu diperiksa sebelum mengolah data pertama kali.



Bab 01

Alur Kerja dalam Satu Halaman

Metashape menerapkan alur *Structure from Motion* dan *Multi-View Stereo*. Foto yang bertampalan dicocokkan untuk menghitung posisi kamera, lalu setiap piksel diproyeksikan menjadi titik tiga dimensi, permukaan, dan akhirnya ortomosaik. Urutannya selalu sama.

01 **Add Photos**
memasukkan foto ke chunk

02 **Reference**
sistem koordinat dan akurasi GPS

03 **Align Photos**
posisi kamera dan titik ikat

04 **Point Cloud**
titik awan padat

05 **Build DEM**
model permukaan (DSM)

06 **Orthomosaic**
foto tegak tanpa distorsi

07 **Export**
GeoTIFF dan laporan

Angka penting data contoh

111	foto DJI Mavic 2 Pro, kamera Hasselblad L1D-20c 20 MP
204 m	tinggi terbang rata-rata di atas titik lepas landas
4,67 cm	resolusi tanah (GSD) rata-rata
0,359 km²	luas cakupan foto
69.253	titik ikat hasil Align Photos
0,535 px	galat proyeksi ulang rata-rata, di bawah ambang 1 piksel
24,2 juta	titik pada point cloud kualitas Medium
16 menit	total waktu pengolahan di Apple M5, RAM 16 GB



Spesifikasi komputer yang disarankan Agisoft

Komponen	Minimal	Disarankan
Prosesor	**4 inti, 64-bit**	8 sampai 16 inti
Memori	**16 GB**	32 sampai 64 GB, lebih untuk ribuan foto
Kartu grafis	**mendukung OpenCL atau Metal**	NVIDIA GeForce RTX, AMD Radeon, atau Apple silicon
Penyimpanan	**SSD**	SSD NVMe, ruang kosong 3 sampai 5 kali ukuran foto

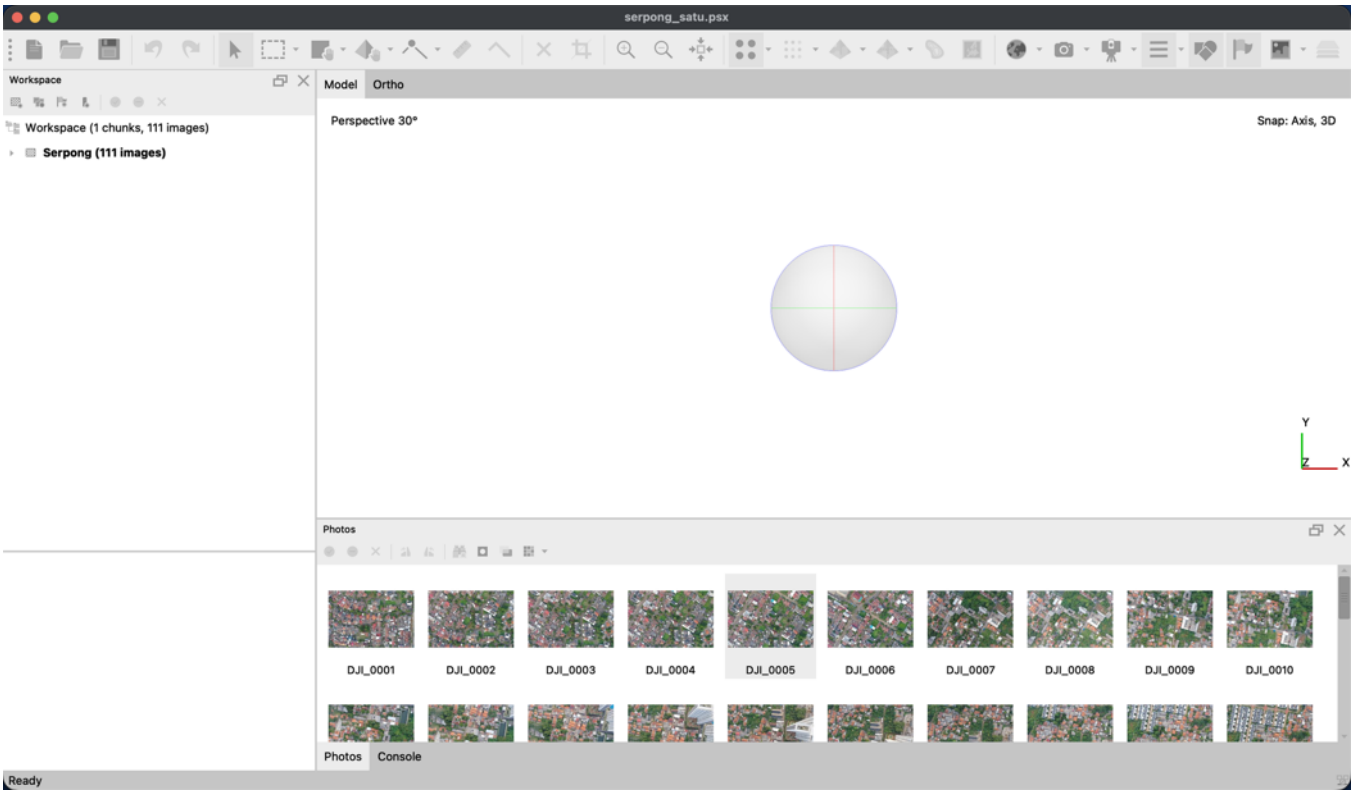




Bab 02

Mengenal Antarmuka

Jendela Metashape terdiri atas beberapa panel yang bisa dibuka dan ditutup dari menu **View**. Tangkapan layar di buku ini diambil dari Metashape Professional 2.1.3 di macOS. Tampilan di Windows sama, kecuali menu **Preferences** yang berada di menu **Tools**.



Gambar 02.1 Antarmuka Metashape setelah 111 foto dimasukkan. Workspace di kiri, tampilan Model di tengah, dan Photos di bawah.



Panel utama

Workspace	Struktur project, yaitu chunk, foto, dan produk yang sudah dibuat.
Model	Tampilan tiga dimensi untuk kamera, titik ikat, point cloud, dan mesh.
Ortho	Tampilan dua dimensi untuk DEM dan ortomosaik.
Photos	Thumbnail foto. Tanda centang berarti foto sudah selaras.
Reference	Koordinat kamera, titik kontrol (GCP), dan galatnya.
Console	Catatan setiap proses dan tempat menjalankan perintah Python.

Menu utama dan fungsinya

Menu	Fungsi utama
File	membuat, membuka, dan menyimpan project, impor dan ekspor data, membuat laporan
Workflow	tahap pengolahan utama dari Add Photos sampai Build Orthomosaic dan Batch Process
Model	alat seleksi, pengukuran, mode tampilan, dan transformasi region
Photo, Ortho	alat khusus tampilan foto dan ortomosaik, misalnya masker dan seamline
Tools	kalibrasi kamera, optimasi kamera, kontur, statistik survei, dan skrip





Bab 03

Pengaturan Preferences

Periksa pengaturan aplikasi sekali sebelum pengolahan pertama melalui **Metashape Pro** lalu **Preferences** di macOS, atau **Tools** lalu **Preferences** di Windows. Pengaturan ini berlaku untuk semua project.



General



GPU



Advanced

Gambar 03.1 Tiga tab Preferences yang perlu diperiksa.

Yang perlu diatur

General	Aktifkan **Write log to file**. Log sangat membantu saat menelusuri masalah pada pengolahan yang panjang.
GPU	Pastikan kartu grafis tercentang. Pencocokan foto dan depth map beberapa kali lebih cepat dengan GPU. Pada kartu grafis diskrit, matikan **Use CPU when performing GPU accelerated processing**.
Advanced	Biarkan **Keep depth maps** aktif bila produk akan dibangun ulang dengan parameter berbeda. Matikan bila ruang disk terbatas.



Bagian II

PENGOLAHAN LANGKAH DEMI LANGKAH

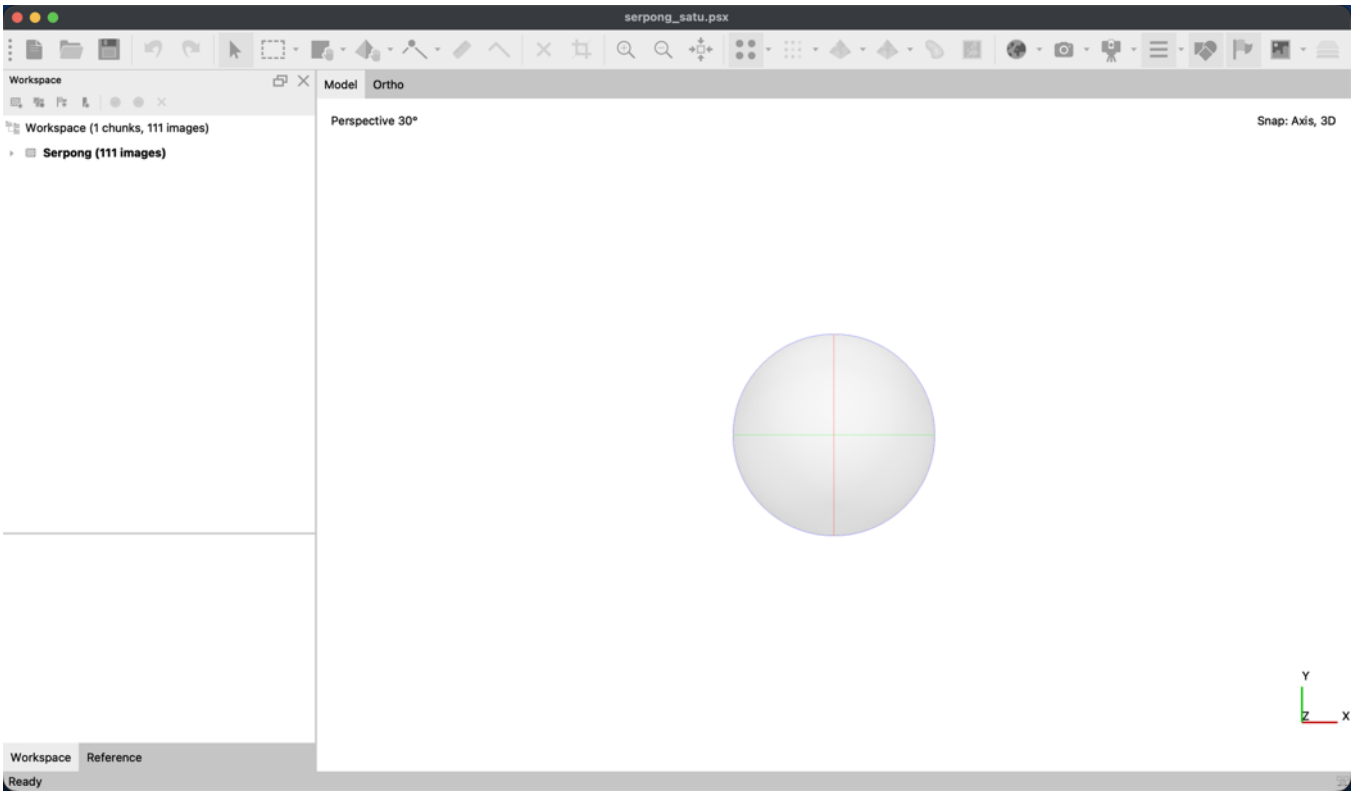
Dari memasukkan foto sampai ortomosaik, lengkap dengan parameter yang disarankan dan hasil data contoh.



Bab 04

Project dan Foto

1. Pilih **File** lalu **Save As** untuk menyimpan project baru berekstensi .psx di folder kerja, misalnya `05_olah/serpong_satu.psx`. Data olahan disimpan di folder `.files` dengan nama yang sama.
2. Pilih **Workflow** lalu **Add Folder** atau **Add Photos**, kemudian pilih folder foto. Metashape membuat chunk baru bila belum ada. Chunk adalah wadah satu blok pengolahan.
3. Klik dua kali nama chunk di Workspace untuk menggantinya, misalnya Serpong.
4. Periksa jumlah foto dan thumbnail di panel Photos. Foto kabur atau tidak relevan, misalnya saat lepas landas, dapat dinonaktifkan dengan klik kanan lalu **Disable Cameras**.



Gambar 04.1 Chunk Serpong berisi 111 foto. Lingkaran biru menandai posisi kamera dari GPS.

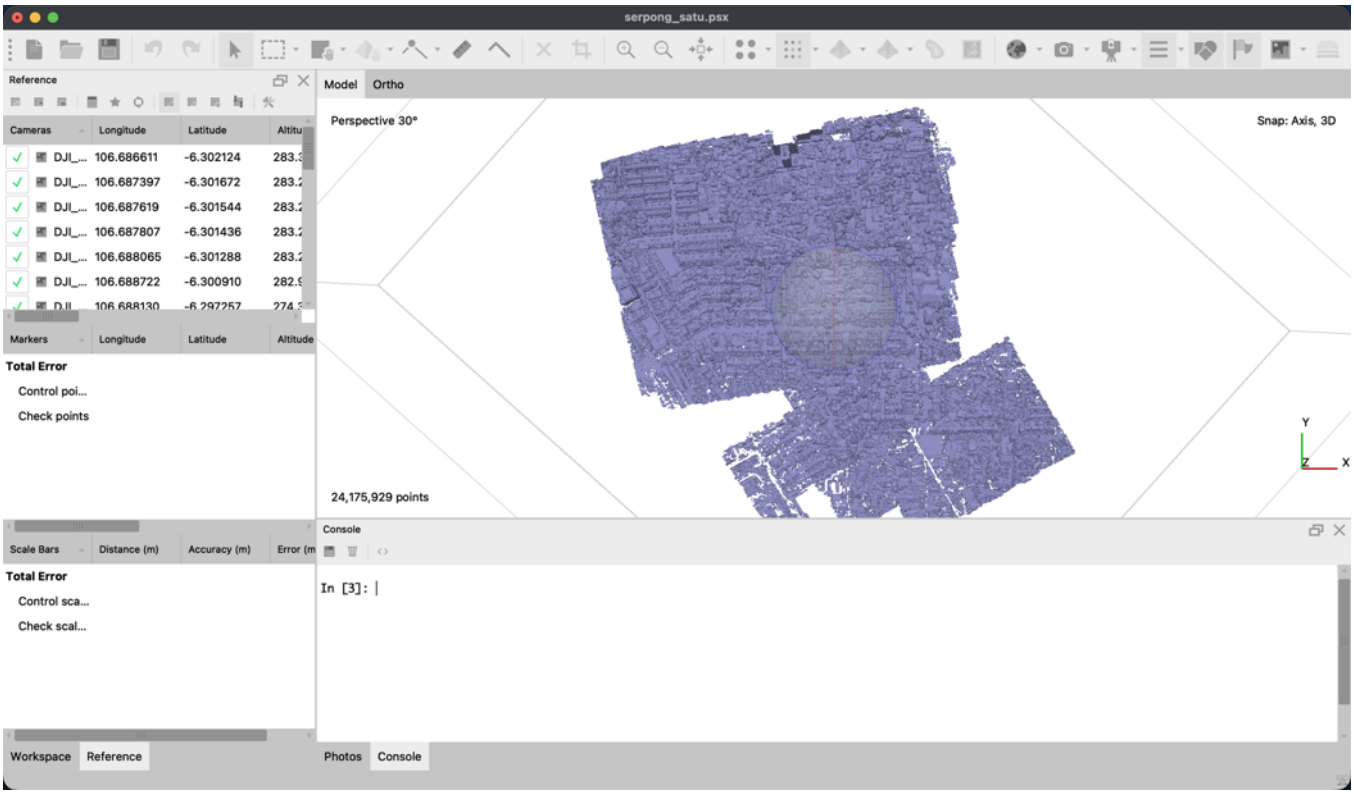




Bab 05

Sistem Koordinat dan Akurasi Kamera

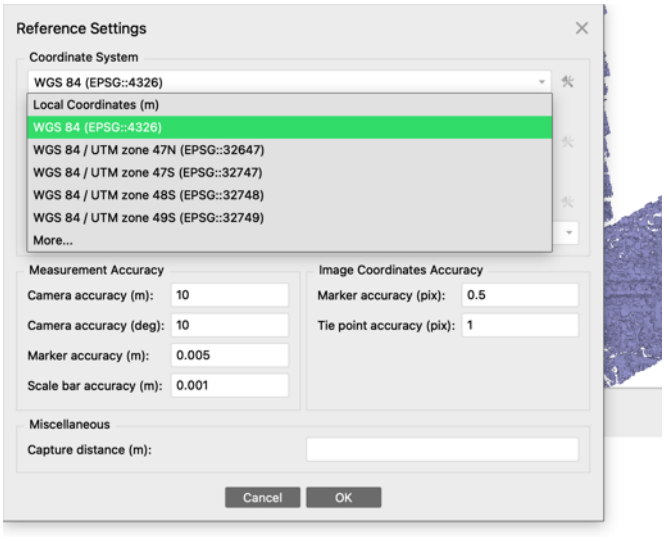
Panel **Reference** menampilkan koordinat kamera dari metadata GPS dalam WGS 84 geografis (EPSG:4326). Untuk pemetaan, produk sebaiknya memakai proyeksi dengan satuan meter, yaitu UTM pada zona lokasi. Data contoh berada di zona 48S.



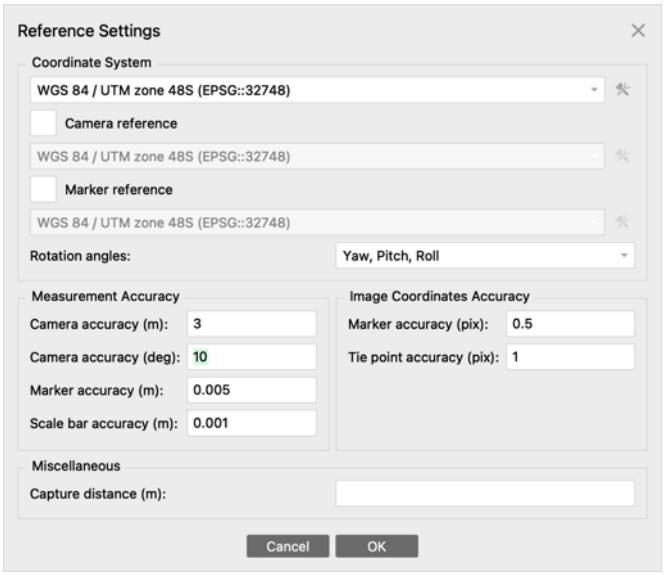
Gambar 05.1 Panel Reference dengan koordinat kamera dalam derajat.

1. Klik ikon **Reference Settings** (kunci pas) di panel Reference.
2. Pada **Coordinate System**, pilih **WGS 84 / UTM zone 48S (EPSG::32748)**. Zona lain dicari melalui **More...**.
3. Isi **Camera accuracy (m)** sesuai ketelitian GPS drone, lalu klik OK.
4. Pilih seluruh kamera di panel Reference, klik kanan, lalu **Convert** ke sistem UTM.





Memilih UTM zona 48S



Akurasi kamera 3 m

Gambar 05.2 Reference Settings untuk data contoh.

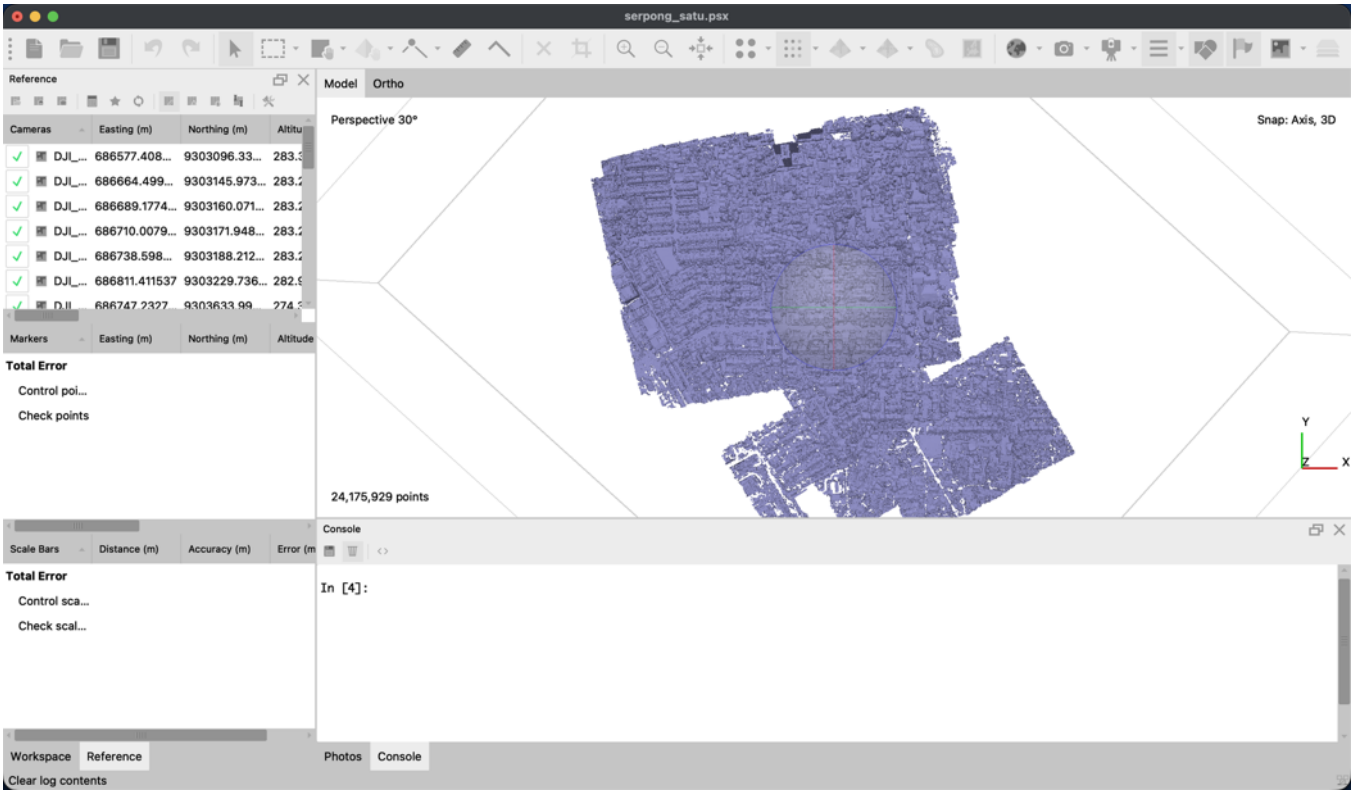
Nilai akurasi yang disarankan

Jenis data	Accuracy	Keterangan
Drone tanpa RTK	3 m	nilai bawaan 10 m terlalu longgar untuk drone modern
Drone RTK status FIX	0,05 m	pastikan foto memang tersimpan dengan status FIX
GCP dari GNSS geodetik	0,005 m	diisi pada Marker accuracy

⚠ Peringatan

Mengganti sistem koordinat di Reference Settings hanya mengubah cara koordinat ditampilkan. Bila koordinat kamera di tabel masih dalam derajat, Metashape menganggap angka derajat itu sebagai meter dan model menjadi rusak. Selalu lakukan **Convert** sebelum Align Photos. Setelah dikonversi, kolom berubah menjadi Easting dan Northing.





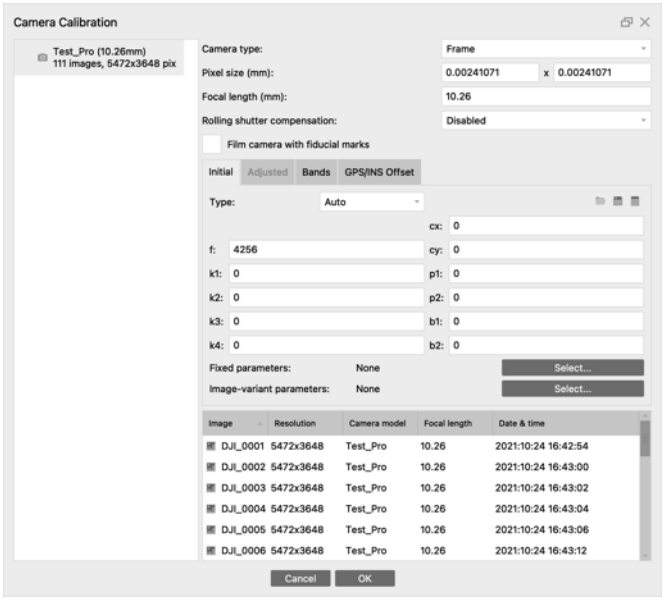
Gambar 05.3 Koordinat kamera setelah dikonversi ke UTM zona 48S.



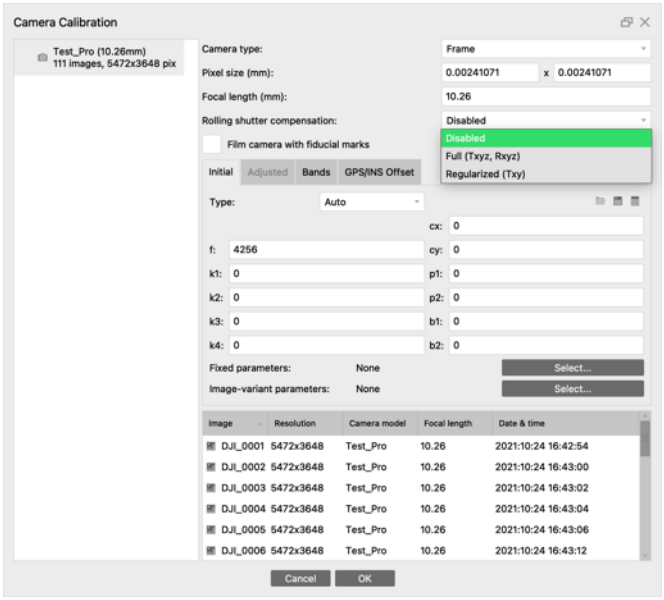
Bab 06

Kalibrasi Kamera dan Rolling Shutter

Pilih **Tools** lalu **Camera Calibration** untuk memeriksa parameter kamera dari EXIF. Untuk data contoh, Metashape membaca Hasselblad L1D-20c dengan resolusi 5.472 × 3.648 piksel, ukuran piksel 2,41 µm, dan panjang fokus 10,26 mm. Parameter distorsi dibiarkan nol karena dihitung saat Align Photos.



Parameter kamera dari EXIF



Pilihan rolling shutter

Gambar 06.1 Dialog Camera Calibration.

Kompensasi rolling shutter

Jenis rana	Pilihan	Contoh drone
Elektronik (rolling)	Regularized (Txy)	Mavic 2 Pro, Mavic Air, Mini series
Elektronik, gerakan besar	Full	lebih lengkap tetapi lebih rawan tidak stabil
Mekanis (global)	Disabled	Phantom 4 Pro, Mavic 3 Enterprise

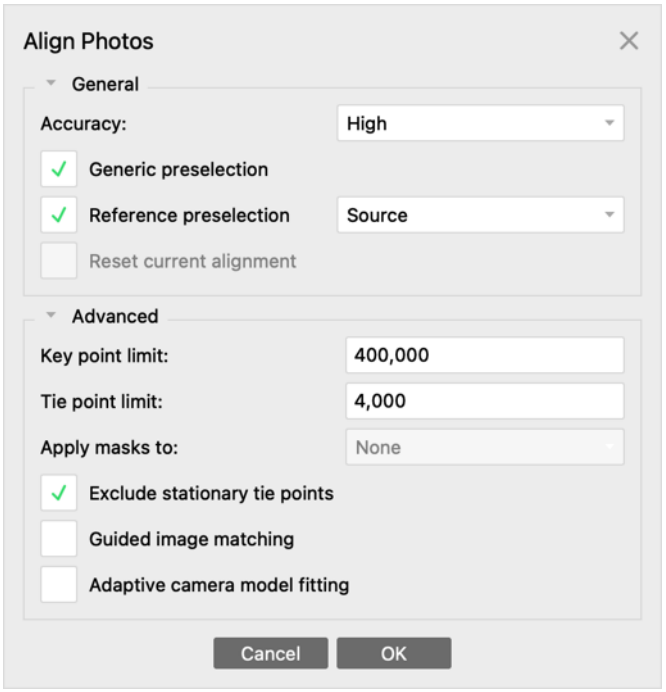




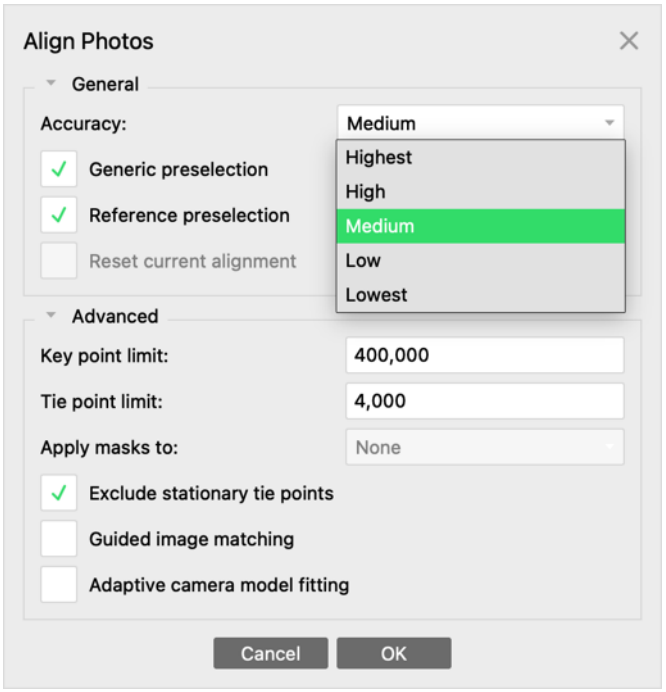
Bab 07

Tahap 1, Align Photos

Pilih **Workflow** lalu **Align Photos**. Tahap ini mendeteksi titik fitur, mencocokkannya antarfoto, lalu menjalankan *bundle adjustment* untuk menghitung posisi dan orientasi setiap kamera serta parameter kalibrasinya.



Dialog Align Photos



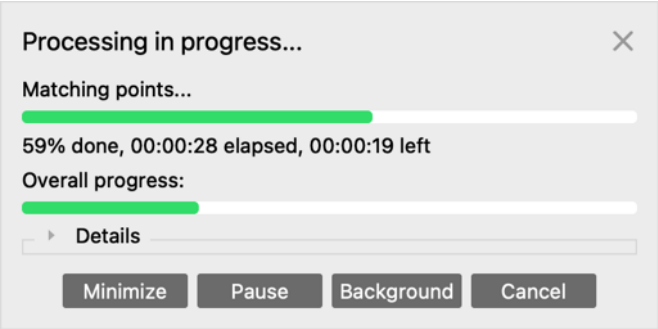
Pilihan Accuracy

Gambar 07.1 Pengaturan Align Photos.

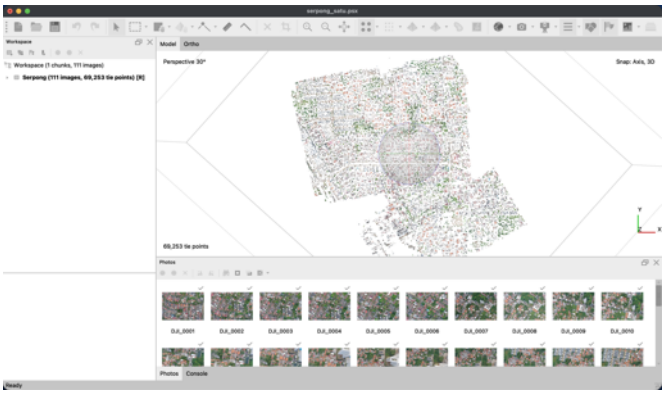


Parameter Align Photos untuk pemetaan

Parameter	Nilai	Penjelasan
Accuracy	High	memakai foto pada resolusi asli. Medium memperkecil 2 kali per sisi
Generic preselection	aktif	mencari pasangan foto dengan versi kecil, mempercepat proses
Reference preselection	aktif, Source	memakai koordinat GPS untuk memilih foto yang berdekatan
Key point limit	40.000 sampai 400.000	jumlah maksimum titik fitur per foto
Tie point limit	4.000	jumlah maksimum titik ikat per foto
Exclude stationary tie points	aktif	membuang titik yang tidak bergerak, misalnya noda di lensa
Adaptive camera model fitting	nonaktif	aktifkan hanya bila kamera belum dikenal



Progres pencocokan



69.253 titik ikat

Gambar 07.2 Proses dan hasil Align Photos.

Pencocokan berlangsung 5 menit 51 detik dan penyelarasan 30 detik. Seluruh 111 foto selaras dengan 69.253 titik ikat dan galat proyeksi ulang rata-rata 0,535 piksel. Foto yang gagal ditandai NA di Workspace.

Tips

Bila ada foto NA, ulangi Align Photos dengan **Reset current alignment** aktif dan Accuracy dinaikkan, atau tambahkan foto dari jalur di sekitarnya. Foto permukaan air dan atap seng yang seragam memang sering gagal dan boleh dibiarkan bila jumlahnya sedikit.

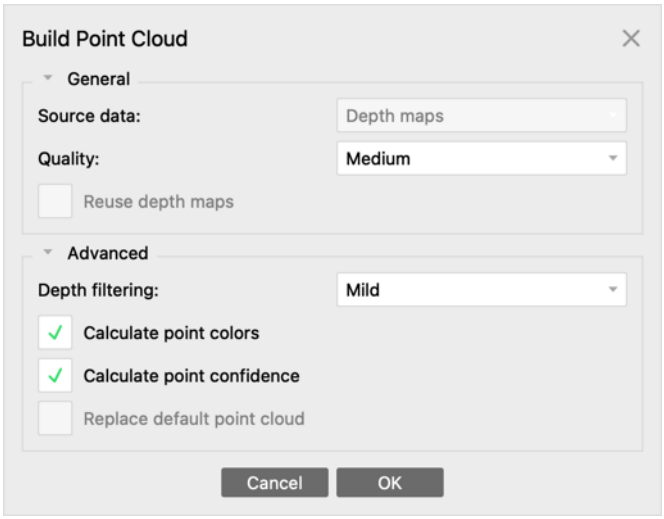




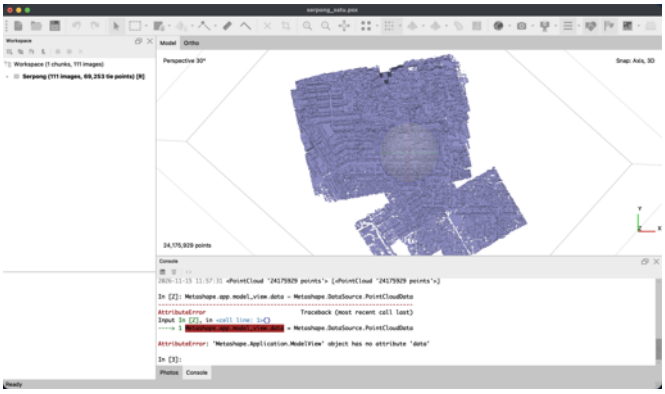
Bab 08

Tahap 2, Build Point Cloud

Pilih **Workflow** lalu **Build Point Cloud**. Depth map dibentuk untuk setiap foto dari posisi kamera hasil Align Photos, lalu digabungkan menjadi titik awan padat.



Quality Medium, filter Mild



24.175.929 titik

Gambar 08.1 Pengaturan dan hasil Build Point Cloud.

Parameter Build Point Cloud

Parameter	Pilihan	Penjelasan
Quality	Medium	cukup untuk ortomosaik dan DSM. High dan Ultra High untuk model detail
Depth filtering	Mild	mempertahankan detail kecil seperti tepi atap. Aggressive untuk permukaan halus
Calculate point colors	aktif	memberi warna RGB pada setiap titik
Calculate point confidence	aktif	menyimpan tingkat keyakinan setiap titik untuk menyaring derau

Pada data contoh, tahap ini menghasilkan 24.175.929 titik dalam 4 menit 42 detik dengan kerapatan rata-rata 28,7 titik per meter persegi.

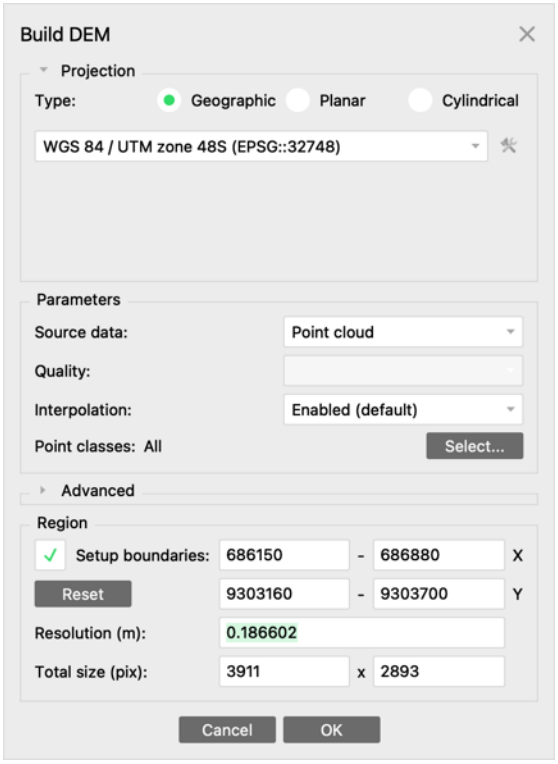


Bab 09

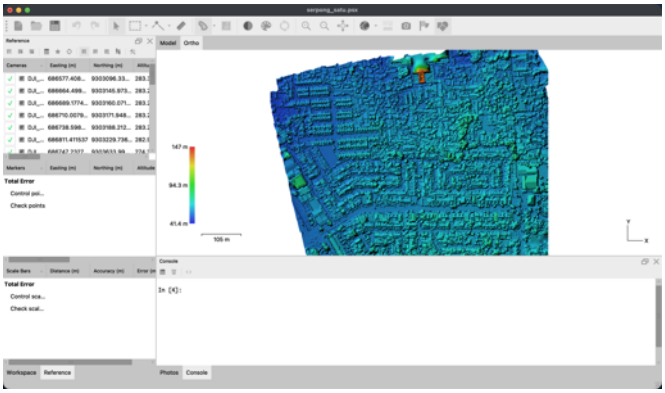
Tahap 3, Build DEM

Pilih **Workflow** lalu **Build DEM**. DEM dari point cloud adalah DSM, yaitu model permukaan termasuk bangunan dan vegetasi. DSM ini juga menjadi permukaan ortorektifikasi pada tahap berikutnya.

1. Pada **Projection**, pastikan sistem koordinat UTM zona 48S.
2. Pada **Source data**, pilih **Point cloud**. **Interpolation** dibiarkan **Enabled** agar celah kecil terisi.
3. Pada **Region**, centang **Setup boundaries** lalu isi batas persegi panjang area misi dalam UTM, misalnya X 686.150 sampai 686.880 dan Y 9.303.160 sampai 9.303.700.
4. Biarkan **Resolution (m)** sesuai usulan Metashape, sekitar 4 kali GSD, lalu klik OK.



Batas persegi panjang UTM



DSM 18,7 cm

Gambar 09.1 Pengaturan dan hasil Build DEM.

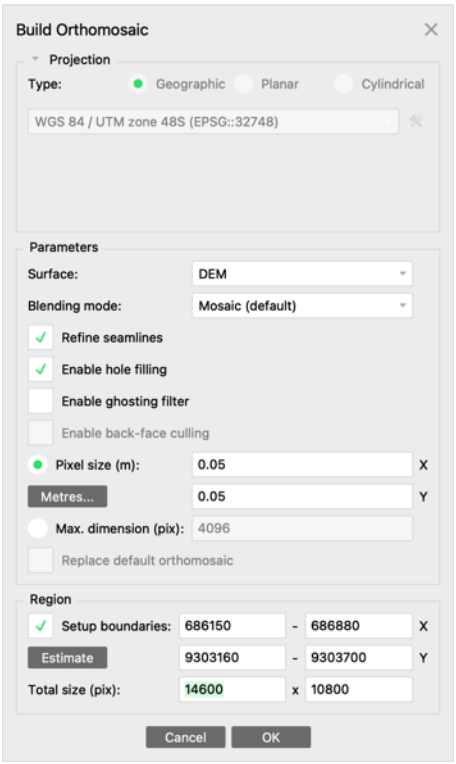
DEM data contoh berukuran 3.911 × 2.893 piksel dengan resolusi 18,7 cm. Gradasi warna biru ke merah menunjukkan ketinggian 41 sampai 147 m.



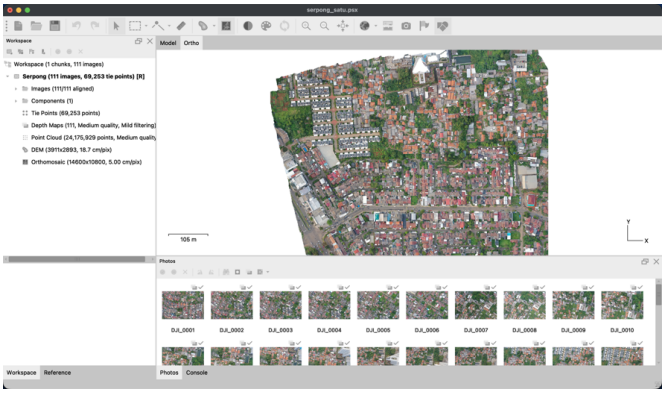
Bab 10

Tahap 4, Build Orthomosaic

Pilih **Workflow** lalu **Build Orthomosaic**. Setiap foto diproyeksikan ulang ke permukaan DEM lalu digabungkan menjadi satu ortomosaik.



Surface DEM, piksel 5 cm



Ortomosaik di tab Ortho

Gambar 10.1 Pengaturan dan hasil Build Orthomosaic.

Parameter Build Orthomosaic

Parameter	Nilai	Penjelasan
Surface	DEM	permukaan ortorektifikasi. Mesh memerlukan Build Model terlebih dahulu
Blending mode	Mosaic	mengambil piksel dari foto paling tegak. Average lebih kabur
Refine seamlines	aktif	garis sambungan mengikuti tepi objek, atap dan mobil tidak terpotong
Enable hole filling	aktif	mengisi celah kecil
Pixel size (m)	0,05	dibulatkan dari GSD rata-rata 4,67 cm
Setup boundaries	persegi panjang	hasil berbentuk persegi panjang utara-selatan

Ortomosaik data contoh berukuran 14.600 × 10.800 piksel dan selesai dalam 3 menit 8 detik dengan pemakaian memori puncak 1,81 GB.



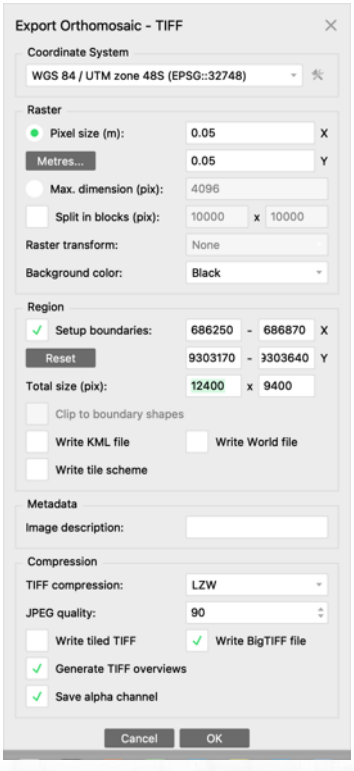


Bab 11

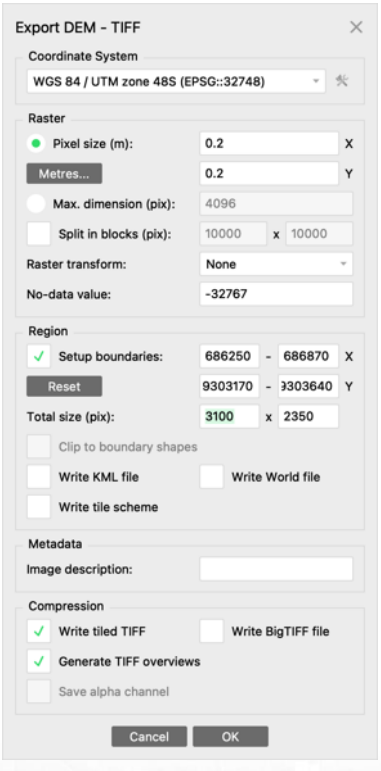
Ekspor GeoTIFF

Hasil di dalam project perlu diekspor ke GeoTIFF agar bisa dibuka di QGIS dan ArcGIS. Agar rapi tanpa tepi bergerigi, pilih batas ekspor berupa persegi panjang terbesar yang seluruhnya tertutup data. Pada data contoh, batas itu adalah X 686.250 sampai 686.870 dan Y 9.303.170 sampai 9.303.640, atau 620 × 470 m.

1. Pilih **File**, **Export**, lalu **Export Orthomosaic**. Isi nama berkas, misalnya `serpong_ortho_5cm`, dan biarkan format **TIFF/GeoTIFF**.
2. Pastikan sistem koordinat UTM zona 48S dan **Pixel size** 0,05 m.
3. Centang **Setup boundaries** dan isi batas persegi panjang di atas.
4. Pilih **Compression LZW**, aktifkan **Write BigTIFF file**, **Generate TIFF overviews**, dan **Save alpha channel**, lalu klik OK.
5. Ulangi dengan **Export DEM** untuk DSM memakai batas yang sama dan ukuran piksel 0,2 m.



Export Orthomosaic



Export DEM

Gambar 11.1 Dialog ekspor dengan batas persegi panjang yang sama.



Gambar 11.2 Ortofoto hasil ekspor, 620 × 470 m pada GSD 5 cm, tanpa bagian kosong.

Ringkasan hasil data contoh

Produk	Ukuran	Keterangan
Titik ikat	69.253 titik	111 dari 111 foto selaras, galat 0,535 piksel
Point cloud	24.175.929 titik	Medium, Mild, 28,7 titik/m²
DEM (DSM)	3.911 × 2.893 piksel	resolusi 18,7 cm
Ortomosaik	14.600 × 10.800 piksel	5 cm, GSD rata-rata 4,67 cm
Ekspor ortofoto	12.400 × 9.400 piksel	620 × 470 m, 590 MB, LZW
Ekspor DSM	3.100 × 2.350 piksel	0,2 m, ketinggian 51 sampai 142 m
Total waktu	sekitar 16 menit	Apple M5, RAM 16 GB

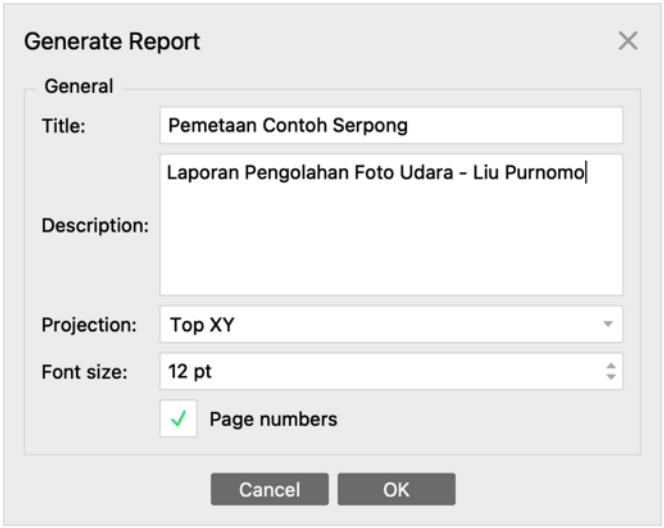




Bab 12

Laporan Pengolahan

Pilih **File**, **Export**, lalu **Generate Report** untuk membuat laporan PDF. Isi **Title** dengan nama pekerjaan dan **Description** dengan keterangan singkat. Metashape mengingat judul terakhir, sehingga periksa kolom ini setiap kali membuat laporan agar judul pekerjaan sebelumnya tidak terbawa.



Dialog Generate Report



Halaman pertama laporan

Gambar 12.1 Membuat laporan pengolahan.

Isi laporan yang perlu diperhatikan

Tinggi terbang	204 m rata-rata pada data contoh.
Resolusi tanah	4,67 cm per piksel.
Luas cakupan	0,359 km².
Galat lokasi kamera	4,44 m rata-rata, sesuai ketelitian GPS drone tanpa RTK. Nilai ini menunjukkan perlunya GCP untuk akurasi absolut.

Tips

Sertakan laporan dalam setiap penyerahan data. Laporan adalah bukti mutu pengolahan dan memuat seluruh parameter yang dipakai.



Bagian III

OTOMATISASI

Menjalankan seluruh tahap sekaligus dengan Batch Process dan skrip Python untuk pekerjaan rutin.

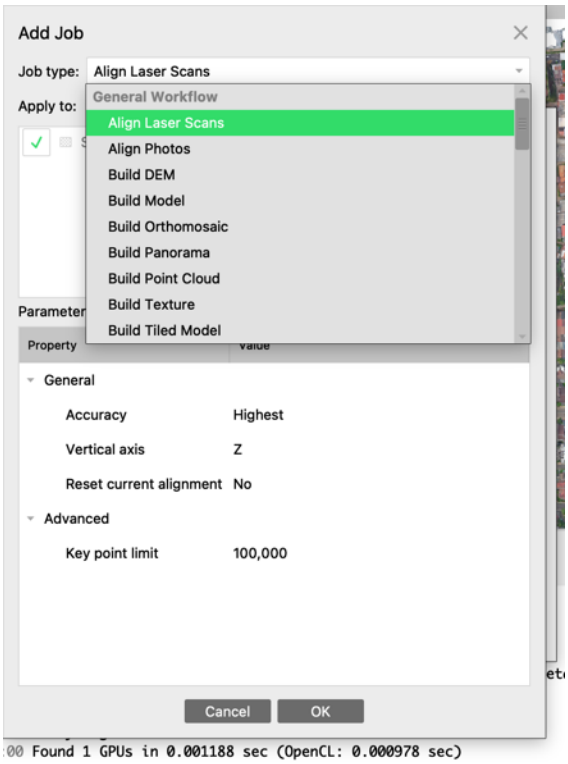


Bab 13

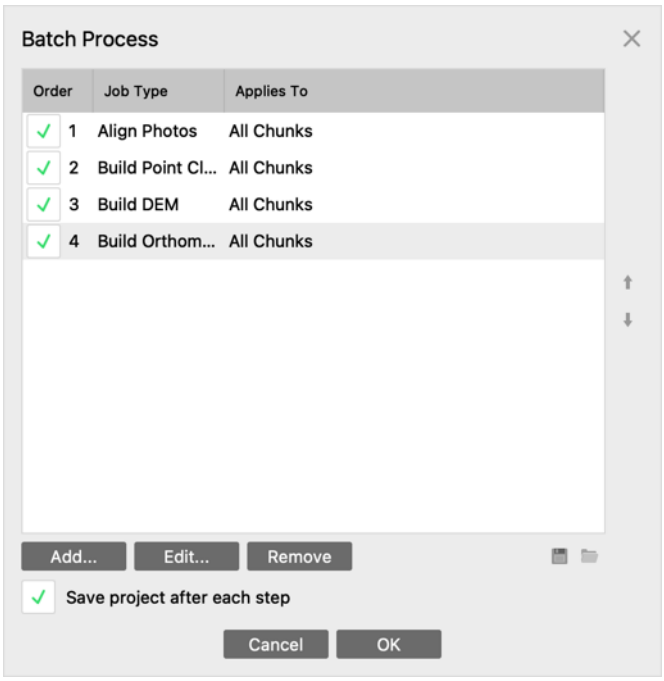
Batch Process

Pilih **Workflow** lalu **Batch Process**. Dialog ini berisi daftar job yang dijalankan berurutan. Daftar job dapat disimpan dan dibuka kembali sebagai berkas XML untuk dipakai di project lain.

1. Klik **Add...**, pilih **Job type**, misalnya **Align Photos**.
2. Pada **Apply to**, pilih **All Chunks** atau **Selected Chunks**.
3. Periksa tabel **Parameters** dan klik kolom **Value** untuk mengubahnya.
4. Ulangi untuk Build Point Cloud, Build DEM, Build Orthomosaic, Export Orthomosaic, Export DEM, dan Generate Report.
5. Aktifkan **Save project after each step**, lalu klik OK.



Daftar jenis job



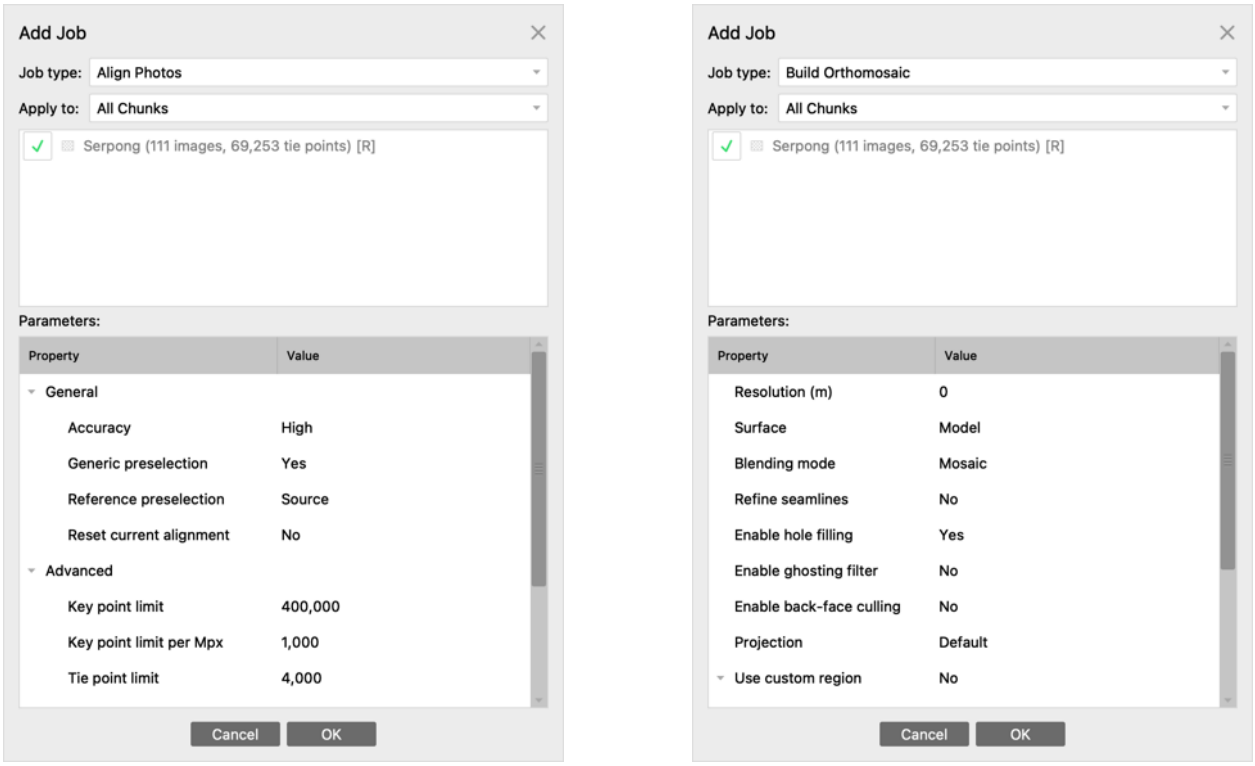
Empat job berurutan

Gambar 13.1 Menyusun daftar job Batch Process.



⚠ Perhatian

Job **Build Orthomosaic** pada Batch Process secara bawaan memakai **Surface Model**, resolusi otomatis, dan tanpa batas wilayah. Bila project tidak memiliki model 3D, job gagal atau memakai permukaan yang salah. Selalu periksa tabel parameter setiap job.



Parameter job Align Photos

Surface harus diganti DEM

Gambar 13.2 Parameter job yang perlu diperiksa.

Parameter job untuk ortofoto

Job	Parameter	Nilai
Align Photos	Accuracy	High
Build Point Cloud	Quality, Filter mode	Medium, Mild
Build DEM	Source data, Projection	Point cloud, UTM zona lokasi
Build Orthomosaic	Surface	DEM
Build Orthomosaic	Resolution (m)	sesuai GSD, misalnya 0,05
Export Orthomosaic dan DEM	Use custom region	Yes, batas persegi panjang





Bab 14

Skrip Python

Metashape Professional dilengkapi modul Python bernama `Metashape`. Skrip di Lampiran memproses setiap subfolder foto menjadi satu chunk, menjalankan empat tahap dengan parameter yang sama seperti di ebook ini, lalu mengekspor ortofoto, DSM, dan laporan. Bila satu folder gagal, folder lain tetap dilanjutkan.

Skrip juga menghitung persegi panjang utara-selatan terbesar yang seluruhnya tertutup data, lalu membulatkannya ke kelipatan 10 m. Hasil ekspor selalu rapi tanpa piksel kosong walaupun bentuk area misi berbeda-beda.

1. Salin skrip ke folder kerja, lalu ubah bagian PENGATURAN, yaitu folder foto, folder hasil, kode EPSG zona UTM, akurasi GPS, dan GSD keluaran.
2. Susun foto dalam subfolder, satu subfolder untuk satu blok atau tanggal, misalnya `Foto/2026-01_Blok_A`.
3. Jalankan dari Terminal dengan perintah di bawah. Di Windows, gunakan `metashape.exe -r batch_metashape.py` dari folder instalasi.

```
/Applications/MetashapePro.app/Contents/MacOS/MetashapePro -r batch_metashape.py
```

Pengolahan manual dan skrip pada data contoh

Aspek	Manual	Skrip
Waktu proses	sekitar 16 menit ditambah waktu operator	16 menit 13 detik tanpa operator
Batas ekspor	dipilih dari tampilan DEM	dihitung otomatis, kelipatan 10 m
Ortofoto	12.400 × 9.400 piksel, 5 cm	12.400 × 9.600 piksel, 5 cm
Piksel kosong	0%	0%
Cocok untuk	belajar dan data bermasalah	pekerjaan rutin, banyak blok, server

Tips

Saat mengekspor dengan skrip, gunakan parameter `resolution_x` dan `resolution_y`. Dengan `resolution` saja, Metashape dapat menyesuaikan ukuran piksel sedikit, misalnya menjadi 0,0500015 m, sehingga hasil tidak tepat bertampalan dengan data lain.





Bab 15

Masalah yang Sering Muncul

Gejala, penyebab, dan solusi

Model miring atau rusak	Koordinat kamera masih derajat padahal sistem diganti ke UTM. Lakukan **Convert** di panel Reference sebelum Align Photos.
Banyak foto NA	Tampalan kurang, foto kabur, atau permukaan air. Naikkan Accuracy, aktifkan Reset current alignment, atau tambah foto.
Ortomosaik berbayang	Blending Average atau Refine seamlines mati. Gunakan Mosaic dan aktifkan Refine seamlines.
Batch ortho gagal	Surface bawaan Batch Process adalah Model. Ganti menjadi DEM.
Hasil tidak tepat di peta	Galat GPS drone beberapa meter. Tambahkan GCP atau gunakan drone RTK.
Pengolahan sangat lambat	GPU tidak aktif atau RAM kurang. Periksa tab GPU di Preferences dan turunkan Quality.
Tepi ortofoto bergerigi	Ekspor tanpa batas. Gunakan Setup boundaries dengan persegi panjang yang seluruhnya tertutup data.
Judul laporan salah	Metashape mengingat judul terakhir. Periksa kolom Title sebelum Generate Report.



LEMBAR REFERENSI

RESEP CEPAT

Cetak halaman ini dan tempel di dekat komputer pengolahan.

Preferences

Log	**Write log to file**
GPU	**tercentang**
Depth maps	**Keep, bila ruang cukup**

Reference

Sistem	**UTM zona lokasi**
Camera acc.	**3 m non RTK, 0,05 m RTK**
Marker acc.	**0,005 m**
Wajib	**Convert sebelum Align**

Kamera

Rana elektronik	**Regularized (Txy)**
Rana mekanis	**Disabled**

Align Photos

Accuracy	**High**
Preselection	**Generic dan Reference**
Key point	**40.000 sampai 400.000**
Tie point	**4.000**

Point Cloud

Quality	**Medium**
Filtering	**Mild**
Colors, confidence	**aktif**

DEM

Source	**Point cloud**
Interpolation	**Enabled**
Region	**persegi panjang UTM**

Orthomosaic

Surface	**DEM**
Blending	**Mosaic**
Seamlines	**Refine**
Pixel	**sesuai GSD, misalnya 0,05 m**

Ekspor

Format	**GeoTIFF, LZW**
Opsi	**BigTIFF, overviews, alpha**
Batas	**persegi panjang tertutup data**
DSM	**0,2 m**



Lampiran

Skrip batch_metashape.py

Skrip lengkap untuk Metashape Professional 2.1. Ubah bagian PENGATURAN sesuai komputer Anda. Skrip hanya memakai Python bawaan Metashape sehingga tidak perlu memasang pustaka tambahan.

batch_metashape.py

```
"""Pengolahan batch Agisoft Metashape Pro 2.1: setiap subfolder foto menjadi satu chunk, lalu
Align Photos → Build Point Cloud → Build DEM → Build Orthomosaic → ekspor ortofoto, DSM, dan laporan.

Batas ortofoto dan DSM dibuat persegi panjang utara-selatan (UTM), yaitu persegi panjang terbesar yang
seluruhnya tertutup data, dibulatkan ke kelipatan 10 m, sehingga hasil rapi tanpa tepi bergerigi.
Skrip hanya memakai Python bawaan Metashape, tanpa pustaka tambahan.

Cara menjalankan (Terminal macOS):
  /Applications/MetashapePro.app/Contents/MacOS/MetashapePro -r batch_metashape.py
Atur variabel di bagian PENGATURAN sebelum dijalankan.
"""
import math
import os

import Metashape

# ---------------------------------------------------------------- PENGATURAN
FOLDER_FOTO = "/path/ke/Foto"            # satu folder, atau folder berisi beberapa subfolder
FOLDER_HASIL = "/path/ke/Hasil"
EPSG = "EPSG::32748"                                # WGS 84 / UTM zona 48S
AKURASI_GPS = 3                                     # meter, untuk drone tanpa RTK
GSD_ORTHO = 0.05                                    # meter
GSD_DSM = 0.20                                      # meter
# ------------------------------------------------------------------------------


def daftar_foto(folder):
    return sorted(os.path.join(folder, f) for f in os.listdir(folder)
                  if f.lower().endswith((".jpg", ".jpeg", ".tif", ".tiff", ".dng")))


def kelompok_foto(folder):
    sub = [os.path.join(folder, d) for d in sorted(os.listdir(folder)) if os.path.isdir(os.path.join(folder, d))]
    sub = [s for s in sub if daftar_foto(s)]
    return {os.path.basename(s): daftar_foto(s) for s in sub} or {os.path.basename(folder): daftar_foto(folder)}


def ubah_ke_utm(chunk, crs):
    """Konversi koordinat kamera dari GPS (WGS 84 geografis) ke UTM, lalu set sistem koordinat chunk."""
    wgs = Metashape.CoordinateSystem("EPSG::4326")
    for cam in chunk.cameras:
        loc = cam.reference.location
        if loc is not None:
            cam.reference.location = Metashape.CoordinateSystem.transform(loc, wgs, crs)
    chunk.crs = crs
    chunk.camera_location_accuracy = Metashape.Vector([AKURASI_GPS] * 3)
    chunk.updateTransform()


def persegi_penuh(chunk, crs):
    """Persegi panjang utara-selatan terbesar yang seluruhnya tertutup DEM (kelipatan 10 m).
    DEM diekspor sementara sebagai ASCII grid 1 m agar dapat dibaca tanpa pustaka tambahan."""
    tmp = os.path.join(FOLDER_HASIL, "_dem_sementara.asc")
    proj = Metashape.OrthoProjection()
```

```
    proj.crs = crs
    chunk.exportRaster(tmp, source_data=Metashape.ElevationData, projection=proj, resolution=1.0,
                       nodata_value=-9999)
    kepala, baris = {}, []
    with open(tmp) as f:
        for i, line in enumerate(f):
            if i < 6:
                k, v = line.split()
                kepala[k.lower()] = float(v)
            else:
                baris.append([float(x) =≠ -9999 for x in line.split()])
    os.remove(tmp)
    if os.path.exists(tmp[:-4] + ".prj"):
        os.remove(tmp[:-4] + ".prj")
    x0 = kepala["xllcorner"]
    sel = kepala["cellsize"]
    y_atas = kepala["yllcorner"] + sel * len(baris)
    tinggi = [0] * len(baris[0])
    terbaik = (0, None)
    for r, ok in enumerate(baris):
        tinggi = [t + 1 if o else 0 for t, o in zip(tinggi, ok)]
        tumpuk = []
        for c in range(len(tinggi) + 1):
            h = tinggi[c] if c < len(tinggi) else 0
            awal = c
            while tumpuk and tumpuk[-1][1] ⩾ h:
                awal, hh = tumpuk.pop()
                if hh * (c - awal) > terbaik[0]:
                    terbaik = (hh * (c - awal), (r - hh + 1, r, awal, c - 1))
            tumpuk.append((awal, h))
    r0, r1, c0, c1 = terbaik[1]
    xmin = math.ceil((x0 + c0 * sel) / 10) * 10
    xmax = math.floor((x0 + (c1 + 1) * sel) / 10) * 10
    ymax = math.floor((y_atas - r0 * sel) / 10) * 10
    ymin = math.ceil((y_atas - (r1 + 1) * sel) / 10) * 10
    return Metashape.BBox(Metashape.Vector([xmin, ymin]), Metashape.Vector([xmax, ymax]))


def olah(doc, nama, foto, crs):
    chunk = doc.addChunk()
    chunk.label = nama
    chunk.addPhotos(foto)
    print(f"[{nama}] {len(chunk.cameras)} foto")
    ubah_ke_utm(chunk, crs)
    for sensor in chunk.sensors:
        sensor.rolling_shutter = Metashape.Shutter.Model.Regularized
    chunk.matchPhotos(downscale=1, generic_preselection=True, reference_preselection=True,
                      keypoint_limit=400000, tiepoint_limit=4000)
    chunk.alignCameras()
    doc.save()
    chunk.buildDepthMaps(downscale=4, filter_mode=Metashape.MildFiltering)
    chunk.buildPointCloud(point_confidence=True)
    doc.save()
    chunk.buildDem(source_data=Metashape.PointCloudData, interpolation=Metashape.EnabledInterpolation)
    batas = persegi_penuh(chunk, crs)
    chunk.buildOrthomosaic(surface_data=Metashape.ElevationData, blending_mode=Metashape.MosaicBlending,
                           refine_seamlines=True, fill_holes=True, resolution=GSD_ORTHO)
    doc.save()
    proj = Metashape.OrthoProjection()
    proj.crs = crs
    dasar = os.path.join(FOLDER_HASIL, nama.replace(" ", "_"))
    # resolution_x dan resolution_y menjaga ukuran piksel tepat, sehingga jumlah piksel = lebar / GSD
    chunk.exportRaster(dasar + "_ortho.tif", source_data=Metashape.OrthomosaicData, projection=proj,
                       resolution_x=GSD_ORTHO, resolution_y=GSD_ORTHO, region=batas, save_alpha=True,
                       white_background=False)
    chunk.exportRaster(dasar + "_dsm.tif", source_data=Metashape.ElevationData, projection=proj,
                       resolution_x=GSD_DSM, resolution_y=GSD_DSM, region=batas)
    chunk.exportReport(dasar + "_laporan.pdf", title=nama, description="Pengolahan batch")
    print(f"[{nama}] selesai, batas {batas.min.x:.0f}-{batas.max.x:.0f} E, {batas.min.y:.0f}-{batas.max.y:.0f} N")


def main():
    os.makedirs(FOLDER_HASIL, exist_ok=True)
    crs = Metashape.CoordinateSystem(EPSG)
    doc = Metashape.Document()
    doc.save(os.path.join(FOLDER_HASIL, "batch.psx"))
    for nama, foto in kelompok_foto(FOLDER_FOTO).items():
        try:
            olah(doc, nama, foto, crs)
        except Exception as e:          # satu folder gagal tidak menghentikan folder lain
```

```
            print(f"[{nama}] GAGAL: {e}")
    doc.save()


main()
Metashape.app.quit()
```



Tentang Penulis



Liu Purnomo, S.Kom. adalah praktisi teknologi dan *software engineer* yang telah berkecimpung di dunia pesawat tanpa awak sejak 2013 dan menjadi instruktur drone profesional sejak 2014. Ia memegang sertifikasi Master Instructor dari Badan Nasional Sertifikasi Profesi (BNSP) serta sertifikasi Remote Pilot dari Direktorat Jenderal Perhubungan Udara.

Selain melatih pilot dan menangani pekerjaan pemetaan udara di berbagai daerah, ia menulis sejumlah buku tentang drone, di antaranya *The Remote Pilot*, *Seni dan Filosofi Penerbangan Drone*, dan *Aerial Mapping*. Ebook ini disarikan dari buku *Aerial Mapping* dan dibagikan gratis agar lebih banyak operator drone mampu mengolah datanya sendiri dengan benar.

Pertanyaan, masukan, dan tawaran kerja sama pelatihan pemetaan dapat disampaikan melalui saluran berikut.

Situs liupurnomo.com

Email hi@liupurnomo.com

liupurnomo

SELAMAT MENGOLAH.

Bagikan ebook ini kepada rekan operator drone, mahasiswa, dan tim pemetaan Anda. Pertanyaan dan masukan dapat dikirim melalui situs atau email di bawah.

liupurnomo.com
hi@liupurnomo.com