

PANDUAN SISTEM LENGKAP: RETRAINING, ANOTASI & LIVE COUNTING KARUNG KONVEYOR

Dokumentasi Arsitektur, Prosedur Operasional Standar, dan Panduan Referensi Teknis Produksi

Target Sistem: Platform Retraining YOLO, Segmentasi SAM3, dan Perhitungan Otomatis Karung Pakan Konveyor

Versi Rilis: 4.2.0 (Produksi)

Lingkungan: Ubuntu Linux 22.04/24.04 LTS, Docker Engine 27+ (CDI GPU Passthrough), NVIDIA CUDA 12.4+

Bahasa Pengantar: Bahasa Indonesia Baku (Register Teknik Senior)

Status Dokumen: *Authoritative Master Reference*

Daftar Isi

- Bab 1: Ringkasan Sistem & Arsitektur Pipeline Data
- Bab 2: Panduan Setup, Instalasi & Eksekusi Lingkungan Kerja
- Bab 3: Manajemen Proyek & Taksonomi Kelas
- Bab 4: Pengelolaan Video Archive & Siklus Kerja 24 Jam
- Bab 5: Pemotongan Video & Ekstraksi Frame
- Bab 6: Pelabelan Otomatis Menggunakan SAM3 Grounding Engine
- Bab 7: Kanvas Review & Anotasi Interaktif
- Bab 8: Penyiapan Data, Triage Kualitas & Augmentasi
- Bab 9: Manajemen Dataset Master & Pembagian Validasi Permanen
- Bab 10: Pelatihan Model YOLO & Evaluasi Benchmark
- Bab 11: Sistem Live Counting & Integrasi Kamera CCTV
- Bab 12: Benchmark Akurasi Perhitungan Headless
- Bab 13: Referensi Teknis, Jaringan, Skema Database & File Layout
- Bab 14: Invarian Domain, Penanganan Kasus Batas & Pemecahan Masalah

Bab 1: Ringkasan Sistem & Arsitektur Pipeline Data

Sistem retraining dan live counting karung pakan adalah platform vision berbasis deep learning terintegrasi untuk otomatisasi pelabelan, kurasi dataset, penyetelan halus (*fine-tuning*) model deteksi YOLO, serta verifikasi penghitungan objek karung pada konveyor transfer pabrik pakan ternak. Sistem dirancang guna menggantikan proses anotasi manual berulang serta memberikan jaminan stabilitas data latih antar-generasi model.

1.1 Latar Belakang & Tujuan Rekayasa

Operasional pabrik pakan menghadapi tantangan variasi visual konveyor: pergantian jenis karung (warna, corak, bahan laminasi), perubahan pencahayaan alami shift kerja, pergeseran sudut kamera CCTV, serta variasi kecepatan konveyor. Peningkatan akurasi model AI menuntut siklus retraining berkala yang cepat, terukur, dan tidak merusak performa deteksi sebelumnya.

Tujuan rekayasa platform:

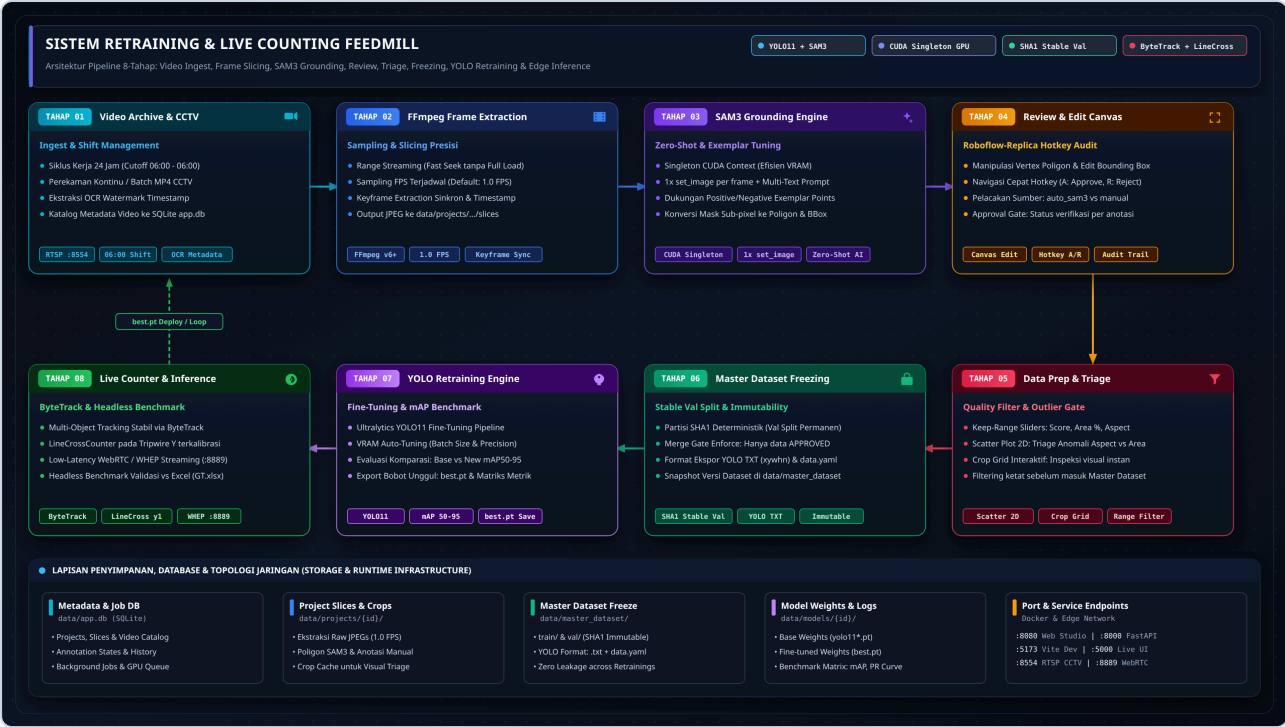
- Otomatisasi Pelabelan Citra:** Memanfaatkan arsitektur Segment Anything Model 3 (Meta SAM3) berbasis prompt teks zero-shot untuk menghasilkan bounding box dan poligon instan.
- Jaminan Pembagian Data Valid:** Menerapkan algoritma *stable validation split* deterministik berbasis fungsi hash SHA-1 sehingga citra validasi tidak pernah bocor ke data latih.
- Penyetelan Halus Terkontrol:** Melatih arsitektur YOLO11 secara efisien dengan manajemen VRAM dinamis serta pembersihan direktori temporer otomatis.
- Verifikasi Penghitungan Nyata:** Menyediakan mesin pelacakan multi-lapisan (ByteTrack dan LineCrossCounter pada batas atas *y₁*) berkecepatan 146 FPS untuk memverifikasi akurasi terhadap data acuan kebenaran (*ground truth*).

1.2 Alur Pipeline 8 Tahap (8-Stage End-to-End Retraining Pipeline)

Alur pemrosesan data end-to-end terbagi menjadi 8 tahap berurutan:

```
[1. Video Archive / CCTV Ingest (06:00 Shift)]
    | (OCR Timestamp & Video Metadata)
    ▼
[2. Video Library & FFmpeg Frame Extractor]
    | (Range Streaming, Sampling at 1.0 FPS)
    ▼
[3. SAM3 Grounding Engine (Singleton CUDA)]
    | (Single set_image, Multi-Prompt Grounding, Exemplars)
    ▼
[4. Roboflow-Replica Review Canvas]
    | (Polygon Vertex/BBox Edit, Hotkeys, Source Tracking)
    ▼
[5. Data Prep, Outlier Triage & Augmentation]
    | (Score/Area/Aspect Keep-Ranges, Scatter Plot, Crop Grid)
    ▼
[6. Master Dataset Freezing & Stable Val Split]
    | (SHA1 Hash-based Val Split, Frozen Rules Snapshot, YOLO TXT)
    ▼
[7. YOLO Retraining & Model Comparison Benchmark]
    | (VRAM Auto-Tuning, Fine-Tuning, Base vs New mAP Comparison)
    ▼
[8. Live Inference Counter & Headless Accuracy Benchmark]
    | (ByteTrack, LineCrossCounter on y1, WebRTC/WHEP, GT Benchmark)
```

Berikut adalah diagram alur visual komprehensif yang merepresentasikan relasi antarmodul, format pertukaran data, dan aliran status pekerjaan:



Unduh format vektor resolusi tinggi: [diagram-alur.svg](#) | Sumber Flat XML Draw: [diagram-alur.fodg](#)

1.3 Peran Komponen Arsitektur Utama

Sistem terdiri dari enam komponen komputasi independen:

- Frontend Web Studio (Port 8080 / 5173):** Antarmuka Single Page Application (SPA) berbasis React 19 dan Vite 7. Mengimplementasikan kanvas review berlatar gelap, visualisasi scatter plot SVG interaktif, tabel benchmark delta akurasi, dan panel kontrol live video WebRTC.
- Backend Application Server (Port 8000):** Server REST API asinkron berbasis FastAPI dan Uvicorn (Python 3.12). Menangani streaming video HTTP 206, orkestrasi antrian pekerjaan, parser OCR, manipulasi dataset, dan interfacing model.
- GPU Singleton Engine Manager:** Modul resident CUDA yang mengelola model SAM3 (~3.9 GB VRAM dasar) dan proses training YOLO secara mutual eksklusif menggunakan mekanisme `jobs.gpu_lock` (timeout 20 detik).
- MediaMTX Streaming Server (Port 8554 & 8889):** Gateway video multi-protokol yang menerima feed RTSP H.264/H.265 dari kamera Dahua CCTV (:8554) dan mentransmisikan ulang melalui protokol WebRTC WHEP berlatensi rendah (:8889) langsung ke peramban.
- Algoritma Tracking & Line Crossing:** Mesin pelacakan ByteTrack dipadukan dengan logika tripwire `LineCrossCounter` yang membaca tepi atas karung (`y_1`) untuk mencegah manipulasi perhitungan akibat deformasi fisik karung.
- SQLite WAL Database & Filesystem Storage:** Database SQLite (`data/app.db`) dalam mode Write-Ahead Logging (WAL) untuk persistensi metadata relasional, dipadukan dengan direktori file terstruktur untuk penyimpanan citra mentah, label teks YOLO, dan file bobot `.pt` .

1.4 Prinsip Desain: Hardware Agnosticism, Portabilitas & Pencegahan Drift

- Agnostisisme Perangkat Keras:** Sistem tidak mematok konfigurasi GPU tertentu pada kode sumber statis. File inisialisasi `start.sh` mendeteksi ketersediaan NVIDIA Container Device Interface (CDI) secara dinamis dan menginjeksi parameter ke `docker-compose.override.yml` . Jika GPU tidak terdeteksi, sistem beralih otomatis ke mode fallback CPU tanpa mengalami crash.
- Portabilitas Lingkungan:** Menggunakan manajer paket `uv` dan lockfile deterministik untuk Python, serta Nginx reverse proxy yang menyamakan jalur routing `/api/` antara kontainer Docker dan server pengembangan lokal.

3. **Pencegahan Konfigurasi Drift:** Seluruh konfigurasi sensitif (seperti `HF_TOKEN`) dibaca eksklusif dari file `.env`. Jalur direktori internal selalu menggunakan relasi path relatif terhadap basis data aplikasi.

Bab 2: Panduan Setup, Instalasi & Eksekusi Lingkungan Kerja

Bab ini memuat instruksi operasional untuk menjalankan sistem pada dua mode eksekusi: kontainer produksi Docker (dengan akselerasi GPU) dan lingkungan pengembangan lokal (*local development*).

2.1 Prasyarat Sistem & Dependensi Perangkat Keras

Konfigurasi perangkat keras dan dependensi minimum:

- **Prosesor (CPU):** x86_64 Quad-Core 2.5 GHz atau lebih tinggi.
- **Memori Utama (RAM):** Minimum 16 GB DDR4 (direkomendasikan 32 GB untuk caching dataset besar).
- **Akselerator Grafis (GPU):** NVIDIA GPU dengan VRAM minimum 8 GB (arsitektur Turing, Ampere, Ada Lovelace, atau Blackwell) dan driver NVIDIA versi 535+.
- **Penyimpanan:** NVMe SSD dengan ruang kosong minimum 100 GB.
- **Sistem Operasi Host:** Ubuntu Linux 22.04 LTS atau 24.04 LTS.
- **Perangkat Lunak:** Docker Engine 27.0+, Docker Compose v2.20+, NVIDIA Container Toolkit, Git, FFmpeg, curl.

2.2 Metode Eksekusi 1: Docker Compose dengan Akselerasi GPU (Produksi)

Docker Compose adalah metode standar pada deployment server industri.

Langkah 1: Kloning repositori dan persiapkan berkas konfigurasi lingkungan:

```
cd /home/asus/feedmill/reTraining
cp .env.example .env
```

Langkah 2: Konfigurasikan token Hugging Face dan jalur arsip video pada `.env`:

```
HF_TOKEN=hf_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
VIDEO_ARCHIVE_HOST=/home/asus/feedmill/data/archive
WEB_PORT=8080
```

Langkah 3: Jalankan skrip inisialisasi otomatis:

```
./start.sh
```

Skrip `start.sh` akan memverifikasi keberadaan `docker`, memeriksa ketersediaan GPU via `nvidia-smi`, menghasilkan konfigurasi override CDI, membangun citra kontainer backend dan frontend, lalu mengaktifkan layanan pada latar belakang (*detached mode*).

Langkah 4: Verifikasi status kontainer:

```
docker compose ps
```

Hasil verifikasi yang valid menunjukkan dua kontainer berstatus `Up`:

- `retraining-backend-1` (Port 8000)
- `retraining-frontend-1` (Port 8080)

2.3 Metode Eksekusi 2: Pengembangan Lokal (uv & npm)

Mode pengembangan lokal digunakan saat pengujian kode sumber secara cepat tanpa proses build image Docker.

Langkah 1: Instal dependensi Python backend menggunakan `uv`:

```
# Pastikan uv telah terpasang pada host
curl -LsSf https://astral.sh/uv/install.sh | sh

# Instal seluruh dependensi backend
uv pip install -r requirements.txt
```

Langkah 2: Instal dependensi Node.js frontend:

```
cd frontend
npm install
cd ..
```

Langkah 3: Jalankan backend FastAPI (Terminal 1):

```
uv run uvicorn backend.main:app --host 0.0.0.0 --port 8000 --reload
```

Langkah 4: Jalankan frontend Vite dev server (Terminal 2):

```
npm --prefix frontend run dev -- --host 0.0.0.0
```

Aplikasi web lokal dapat diakses melalui peramban pada alamat `http://localhost:5173`.

2.4 Konfigurasi File .env & Manajemen Kredensial

File `.env` terletak pada direktori akar repositori dan tidak boleh diikutsertakan ke dalam version control publik.

Tabel parameter konfigurasi `.env`:

Nama Variabel	Nilai Default / Contoh	Tipe	Deskripsi Operasional
HF_TOKEN	hf_AbCdEf...	String	Token otentikasi Hugging Face untuk mengunduh bobot Meta SAM3.
VIDEO_ARCHIVE_HOST	/home/asus/feedmill/data/archive	Path	Jalur direktori arsip video CCTV pada host machine.
WEB_PORT	8080	Integer	Port Nginx frontend yang diekspos ke jaringan lokal host.
BACKEND_PORT	8000	Integer	Port FastAPI backend service.
RTSP_URL	rtsp://192.168.192.96:8554/cam	URL	URL stream RTSP kamera conveyor dari MediaMTX.
WHEP_URL	http://192.168.192.96:8889/cam/whep	URL	Endpoint WebRTC WHEP untuk pemutaran video langsung di UI.

2.5 Pemeriksaan Kesehatan Sistem (Health Check)

Untuk menguji kesiapan layanan backend, jalankan perintah curl berikut:

```
curl -s http://localhost:8000/api/health | jq .
```

Respons JSON yang valid:

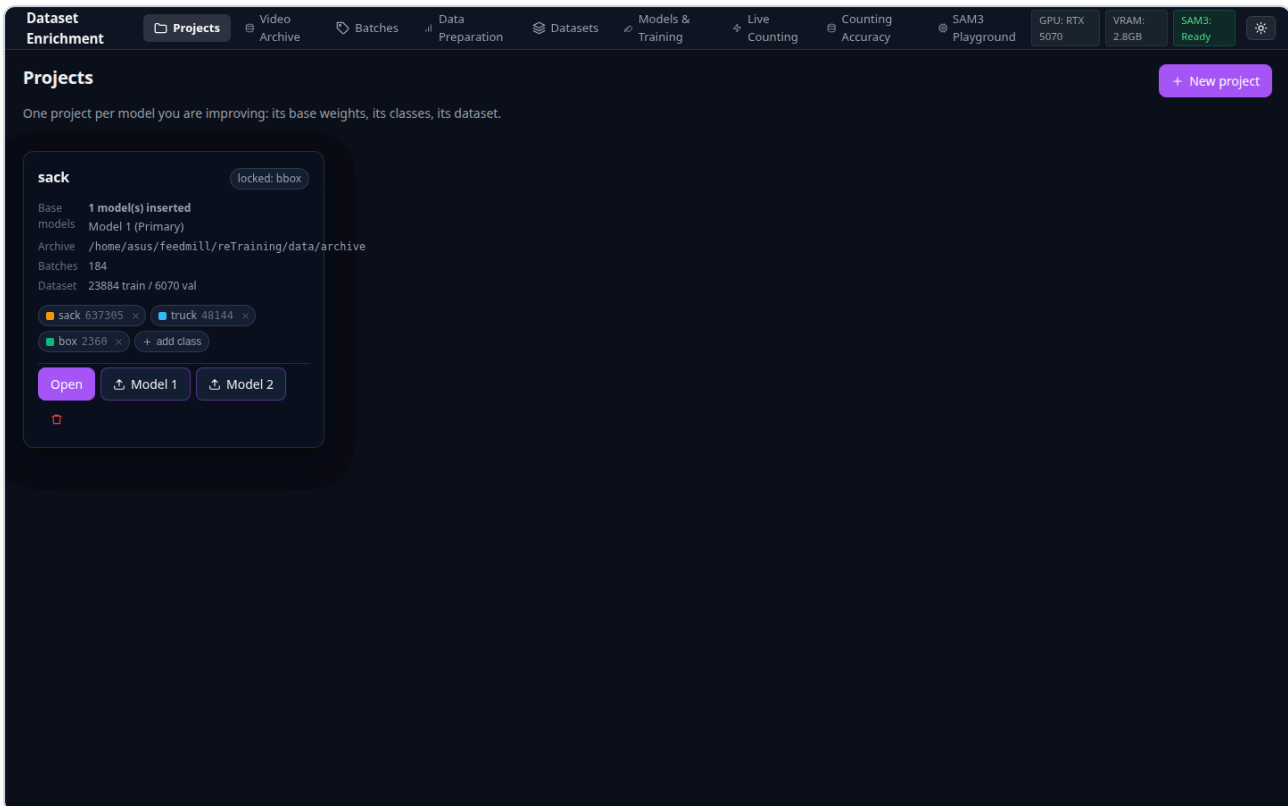
```
{  
  "status": "ok",  
  "gpu_available": true,  
  "device": "cuda:0",  
  "sam3_loaded": false,  
  "active_jobs": 0  
}
```

Bab 3: Manajemen Proyek & Taksonomi Kelas

Bab ini memandu operator dalam menginisialisasi proyek baru, menetapkan geometri anotasi, menyusun taksonomi kelas deteksi, serta memetakan teks prompt untuk model segmentasi otomatis.

3.1 Struktur Proyek & Ruang Kerja Terisolasi

Setiap proyek deteksi merepresentasikan satu target fisik spesifik pada lini produksi (misalnya deteksi karung pakan ayam 50kg, karung pakan ikan, atau palet konveyor). Seluruh aset citra, file anotasi, subset validasi, dan versi model tersimpan secara terisolasi di dalam direktori `data/projects/<slug>/`.



Gambar 1: Antarmuka Manajemen Proyek (Projects Overview). Menampilkan kartu ringkasan proyek aktif, model dasar YOLO primer dan sekunder, taksonomi kelas dengan kuantitas objek terdeteksi, statistik pembagian data latih/validasi, serta tombol navigasi modul.

(English label: Projects Overview Page)

3.2 Pembuatan Proyek Baru & Parameter Awal

Untuk membuat proyek baru, klik tombol `+ New project` pada bagian atas halaman Projects. Sistem akan membuka form modal pembuatan proyek.

Projects

One project per model you are improving: its base weights, its classes, its dataset.

New project

Name:

Label type: Bounding box (YOLO detec)

Val split — every Nth frame:

Fixed once the first batch is merged — every label file is written in this format.

Video archive root:

Laid out as <date>/<batch>.mp4. Read-only — nothing is written here.

Classes: class 0 SAM3 prompt (defaults to the name)

Upload a base model afterwards to replace these with the model's own classes.

Create project Cancel + Add class

sack locked: bbox

Base: 1 model(s) inserted

models: Model 1 (Primary)

Archive: /home/asus/feedmill/reTraining/data/archive

Batches: 184

Dataset: 23884 train / 6070 val

sack 637305 truck 48144 box 2368 + add class

Gambar 2: Form Pembuatan Proyek Baru (New Project Modal). Konfigurasi nama proyek, jenis geometri anotasi (BBox atau Polygon), rasio langkah pembagian validasi (val split stride), jalur arsip video, dan penetapan prompt teks SAM3 awal untuk setiap kelas.

(English label: Project Creation Modal)

Parameter pada form pembuatan proyek:

- Project Name:** Nama identifikasi proyek (misalnya `Feedmill Sack Counter`). Nama ini akan diubah menjadi format URL slug (contoh `feedmill-sack-counter`).
- Label Type (Geometri):**
 - `bbox` (Bounding Box): Format koordinat segi empat $[x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}]$. Direkomendasikan untuk deteksi objek reguler dan kecepatan inferensi maksimal.
 - `polygon`: Format koordinat poligon segmentasi multi-titik $[(x_1, y_1), (x_2, y_2), \dots]$. Direkomendasikan untuk objek saling tumpang tindih (*heavy occlusion*).

Perhatian: Jenis geometri terkunci permanen setelah batch pertama digabungkan ke dataset.

- Val Split (Langkah Validasi):** Nilai integer N (default 5). Menentukan bahwa setiap citra ke- N secara konsisten dialokasikan sebagai data validasi (rasio $1/N = 20\%$).
- Video Archive Root:** Jalur direktori arsip video CCTV (default `/videos`).
- Model Checkpoint:** Opsi untuk mengunggah bobot awal `.pt` (misalnya `v4-best.pt`). Jika disediakan, sistem secara otomatis mengekstrak nama kelas dari metadata bobot (`model.names`).

3.3 Taksonomi Kelas & Pemetaan Prompt Teks

Tabel definisi kelas dan pemetaan prompt pada proyek deteksi karung pakan:

ID Kelas	Nama Kelas	Warna Swatch	Prompt Teks SAM3	Deskripsi Objek Fisik
0	sack	#3b82f6 (Biru)	white woven plastic sack on conveyor belt	Karung pakan plastik tenun putih yang melintas di konveyor.
1	sack_damaged	#ef4444 (Merah)	torn damaged sack leaking feed powder	Karung sobek, bocor, atau kemasan rusak parah.
2	person	#10b981 (Hijau)	worker operator person handling bags	Operator atau pekerja pabrik di sekitar konveyor.

Peraturan kaskade perubahan kelas:

- Penambahan kelas baru memperbarui entri tabel `project_classes` dan menambahkan kunci kelas pada `data.yaml`.
- Penghapusan kelas memicu pembersihan seluruh anotasi kelas tersebut pada database dan disk, serta melakukan re-indeks penomoran ID kelas yang lebih tinggi untuk mencegah celah indeks (*gap index*).
- Penghapusan kelas terakhir pada proyek diblokir oleh sistem untuk menjaga integritas skema.

Bab 4: Pengelolaan Video Archive & Siklus Kerja 24 Jam

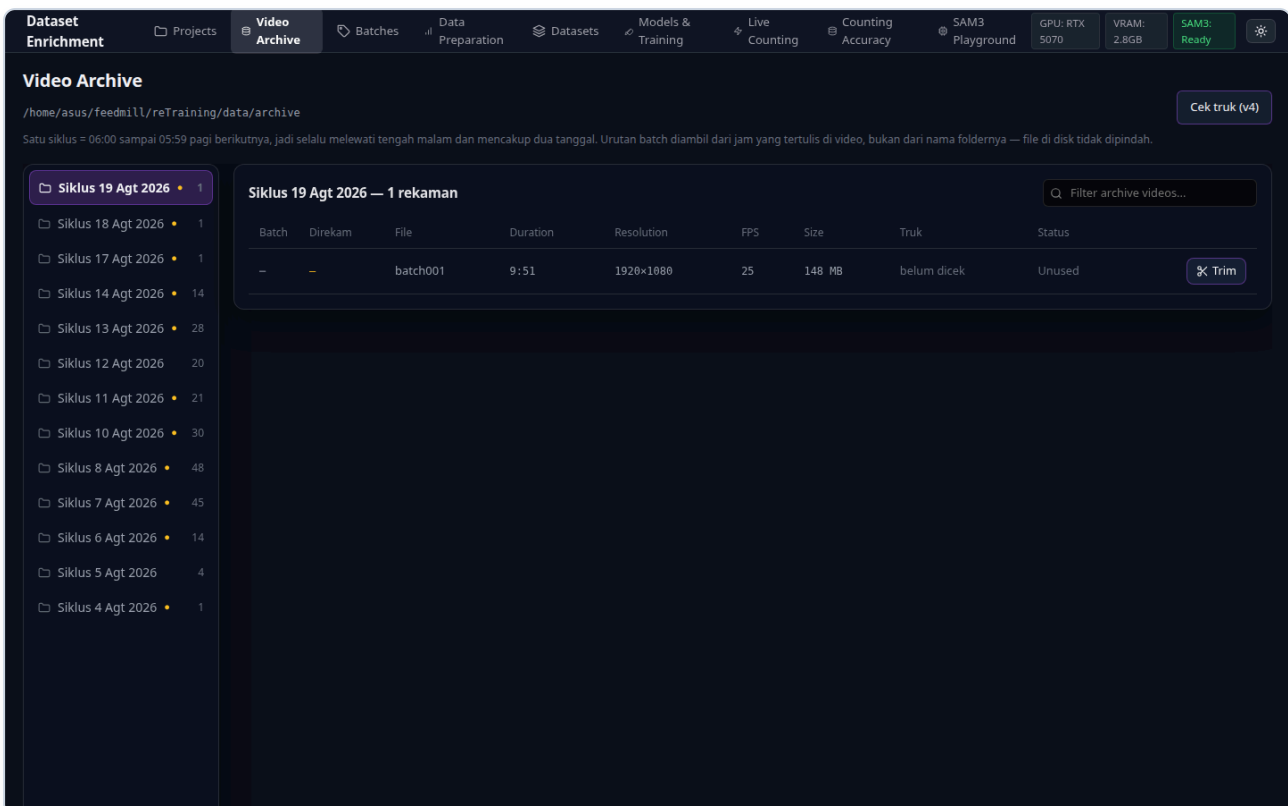
Bab ini menguraikan mekanisme pengorganisasian arsip rekaman CCTV, konversi stempel waktu video melalui Optical Character Recognition (OCR), dan pemindaian otomatis keberadaan truk pengangkut.

4.1 Logika Siklus Kerja 24 Jam (Shift 06:00)

Pabrik pakan ternak menerapkan siklus kerja 24 jam yang dimulai pukul 06:00 pagi dan berakhir pukul 05:59 pagi pada hari berikutnya. Sistem mengelompokkan rekaman video berdasarkan siklus kerja operasional pabrik, bukan berdasarkan tanggal kalender standar tengah malam (00:00).

Formula penetapan tanggal siklus kerja (D_{siklus}): $D_{\text{siklus}} = \begin{cases} D_{\text{kalender}}, & \text{jika } t_{\text{rekam}} \geq 06:00:00 \\ D_{\text{kalender}} - 1 \text{ hari}, & \text{jika } t_{\text{rekam}} < 06:00:00 \end{cases}$

Contoh: Video yang direkam pada tanggal 14 Agustus 2026 pukul 02:30:00 ini hari akan dikelompokkan ke dalam **Siklus 13 Agu 2026**.



Gambar 3: Video Archive & Siklus Produksi 24 Jam (Library Page). Panel siklus kerja harian pada sisi kiri, indikator validasi OCR (lingkaran amber untuk jam belum terverifikasi), status deteksi truk v4, rincian resolusi/durasi video, dan tombol aksi pemotongan.

(English label: Video Archive & Cycles Tree)

4.2 Ekstraksi Jam Video melalui OCR & Koreksi Manual

Kamera CCTV industri melakukan pembakaran stempel waktu (*burned-in timestamp*) langsung pada piksel video pojok kanan atas atau kiri bawah. Modul `backend/video_clock.py` mengekstrak koordinat waktu tersebut menggunakan metode pencocokan template (*template matching*) 12 glif angka (0 sampai 9, `:`, dan spasi).

Prosedur penanganan stempel waktu:

1. Sistem membaca frame pertama video dan melakukan crop area koordinat jam.

2. Hasil OCR disimpan pada tabel `video_clock` sebagai teks waktu lokal (*wall-clock time* format `HH:MM:SS`) untuk menghindari pergeseran akibat konversi zona waktu UTC/WIB di peramban.
3. Apabila skor kecocokan glif OCR berada di bawah ambang batas (<0.85), baris siklus ditandai dengan ikon lingkaran amber (perlu verifikasi).
4. Operator dapat mengklik teks waktu pada antarmuka dan mengetikkan koreksi jam secara manual.

4.3 Pemindaian Truk Massal Terotomatisasi (Truck Scan v4)

Tidak semua video arsip memuat aktivitas bongkar muat karung. Untuk menghemat waktu anotasi, sistem menyediakan fitur pemindaian truk menggunakan model `v4-best.pt`.

Langkah operasional:

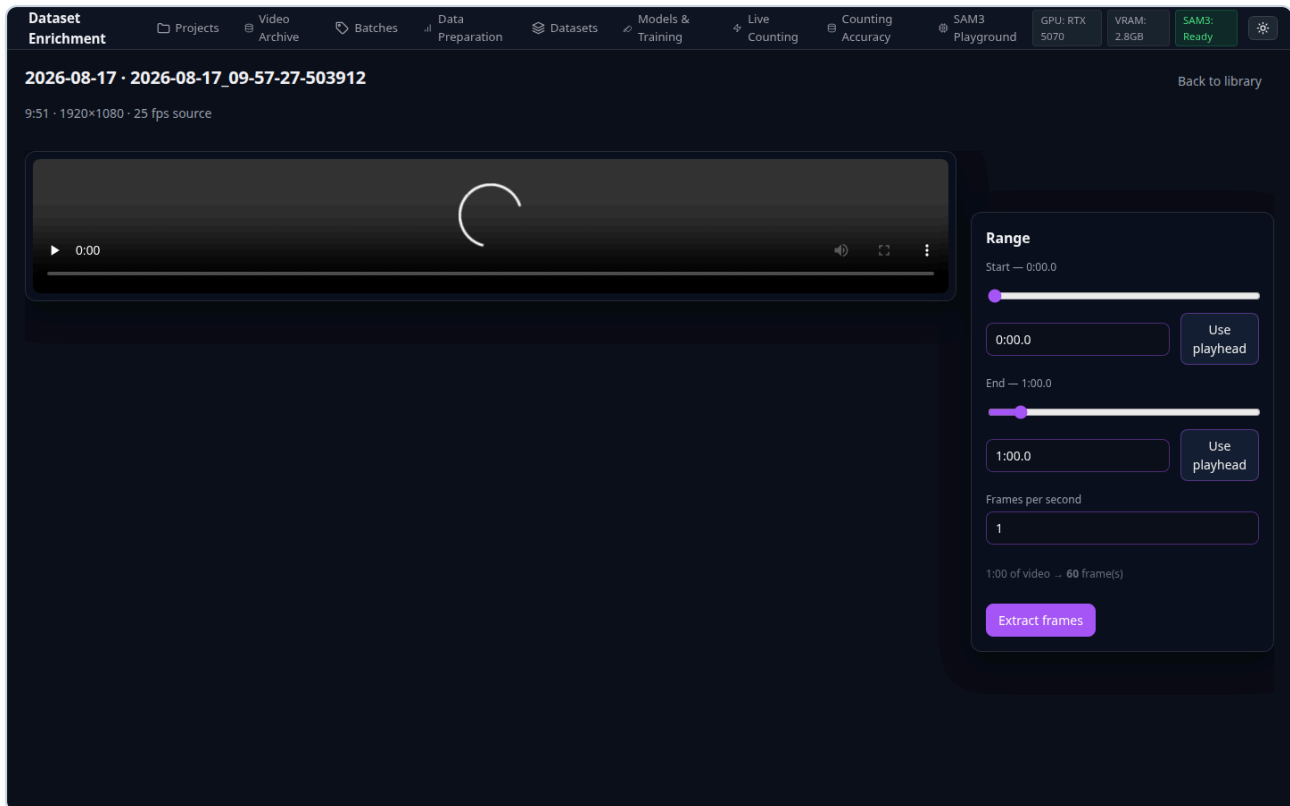
1. Buka halaman Library proyek (`/projects/<id>`).
2. Klik tombol `Cek truk (v4)` pada header tabel arsip.
3. Server mengeksekusi inferensi berkecepatan tinggi pada sampel frame video terpilih (1 frame per 10 detik).
4. Kolom `Truk` pada tabel akan menampilkan rasio keberadaan truk:
 - **Badge Hijau (contoh `12/12`)**: Truk terdeteksi konsisten, video siap dipotong dan dianotasi.
 - **Badge Merah (`tanpa truk`)**: Konveyor dalam keadaan kosong/mati, video dapat dilewati.
 - **Badge Abu-abu (`belum dicek`)**: Video baru yang belum dipindai.

Bab 5: Pemotongan Video & Ekstraksi Frame

Bab ini menjelaskan teknik isolasi segmen rekaman video operasional dan ekstraksi frame citra beresolusi penuh untuk persiapan dataset pelatihan.

5.1 Editor Pemotongan Video (Trim Page)

Modul pemotongan video memanfaatkan protokol HTTP 206 (*Partial Content Range Streaming*) pada backend FastAPI (`/projects/{id}/video?rel=...`). Protokol ini memungkinkan peramban melakukan scrubbing timeline video secara instan tanpa mengunduh keseluruhan berkas video berukuran gigabyte.



Gambar 4: Editor Pemotongan Video & Ekstraksi Frame (Trim Page). Pemutar video HTML5 dengan slider rentang waktu In/Out, tombol sinkronisasi playhead, input laju sampling FPS, badge estimasi total frame, dan tombol eksekusi ekstraksi.

(English label: Video Trim & Frame Extractor)

5.2 Penentuan Rentang Timecode & Sampling FPS

Langkah operasional pemotongan video:

1. Geser playhead video ke titik awal saat karung pertama mulai bergerak di atas konveyor.
2. Klik tombol `Use playhead` pada slider **Start** untuk mengunci waktu In.
3. Geser playhead ke titik akhir saat karung terakhir melintasi garis konveyor.
4. Klik tombol `Use playhead` pada slider **End** untuk mengunci waktu Out.
5. Tentukan nilai **Frames per second (FPS)**:
 - **Nilai Standar (1.0 FPS)**: Mengekstrak 1 frame per detik. Pilihan optimal untuk konveyor berkecepatan normal (0.3 sampai 0.6 m/s) guna menghindari duplikasi visual yang berlebihan.
 - **Nilai Tinggi (2.0 FPS)**: Digunakan pada konveyor cepat atau kondisi karung bertumpuk rapat.
 - **Nilai Rendah (0.5 FPS)**: Digunakan untuk perekaman durasi panjang dengan variasi visual minimal.
1. Periksa badge kalkulasi otomatis: `<durasi detik> of video -> <N> frame(s)`.

2. Klik tombol `Extract frames`.

5.3 Manajemen Antrean Batch Hasil Ekstraksi

Setelah tombol `Extract frames` ditekan, backend mendaftarkan pekerjaan ke antrean `jobs` dan mengeksekusi perintah FFmpeg secara asinkron:

```
ffmpeg -ss <start_sec> -to <end_sec> -i <video_path> -vf fps=<fps> -q:v 2 frames/%06d.jpg
```

Frame citra disimpan dengan format penomoran enam digit (`000001.jpg`, `000002.jpg`, dst.) di dalam direktori `data/projects/<slug>/batches/<batch-id>/frames/`.

Batch	Range	Frames	Reviewed	Shapes	Status				
2026-08-18: batch010	0:00-0:47 @ 1fps	47	42/47	836	merged	Auto-annotate	Reset Auto	Review (42/47)	Delete
2026-08-19: batch001	0:00-1:00 @ 1fps	60	60/60	1529	merged	Auto-annotate	Reset Auto	Review (60/60)	Delete
2026-08-14: batch006	0:00-28:41 @ 1fps	1721	1721/1721	46895	merged	Auto-annotate	Reset Auto	Review (1721/1721)	Delete
2026-08-14: batch005	0:00-25:41 @ 1fps	1541	1541/1541	39252	merged	Auto-annotate	Reset Auto	Review (1541/1541)	Delete
2026-08-14: batch004	0:00-38:21 @ 1fps	2301	2301/2301	48869	merged	Auto-annotate	Reset Auto	Review (2301/2301)	Delete
2026-08-14: batch001	0:00-41:57 @ 1fps	2517	2517/2517	55933	merged	Auto-annotate	Reset Auto	Review (2517/2517)	Delete
2026-08-14: batch002	0:00-43:25 @ 1fps	2605	2605/2605	64490	merged	Auto-annotate	Reset Auto	Review (2605/2605)	Delete
2026-08-13: batch007	0:00-42:55 @ 0.05fps	129	129/129	3330	merged	Auto-annotate	Reset Auto	Review (129/129)	Delete
2026-08-13: batch006	0:00-27:39 @ 0.05fps	83	83/83	2455	merged	Auto-annotate	Reset Auto	Review (83/83)	Delete
2026-08-13: batch005	0:00-28:44 @ 0.05fps	86	86/86	2793	merged	Auto-annotate	Reset Auto	Review (86/86)	Delete
2026-08-13: batch004	0:00-23:22 @ 0.05fps	70	70/70	2133	merged	Auto-annotate	Reset Auto	Review (70/70)	Delete

Gambar 5: Tabel Manajemen Batch (Batches Page). Daftar batch hasil ekstraksi, informasi rentang timecode dan FPS, total frame, rasio review visual, kuantitas deteksi, serta tombol eksekusi auto-labeling dan penggabungan dataset.

(English label: Batch Work Queue Table)

Aksi operasional pada tabel batch:

- **Checkbox Seleksi:** Memilih satu atau beberapa batch untuk proses penggabungan (*merge*).
- **Auto-annotate:** Membuka dialog pelabelan otomatis SAM3 untuk batch tunggal.
- **Review (n/N):** Masuk ke modul kanvas review anotasi interaktif.
- **Reset Auto:** Menghapus seluruh anotasi otomatis (`source='auto'`) dan mempertahankan anotasi manual (`source='manual'`).
- **Download Annotations (.zip):** Mengunduh arsip ZIP berisi citra dan file label teks YOLO.
- **Restore from .zip:** Memulihkan anotasi dari berkas cadangan ZIP eksternal.

Bab 6: Pelabelan Otomatis Menggunakan SAM3 Grounding Engine

Bab ini menguraikan arsitektur model segmentasi Meta SAM3, mekanisme pelabelan otomatis berbasis prompt teks zero-shot, panduan interaksi kotak contoh (*exemplar*), serta penggunaan modul Playground.

6.1 Arsitektur SAM3 Zero-Shot Grounding

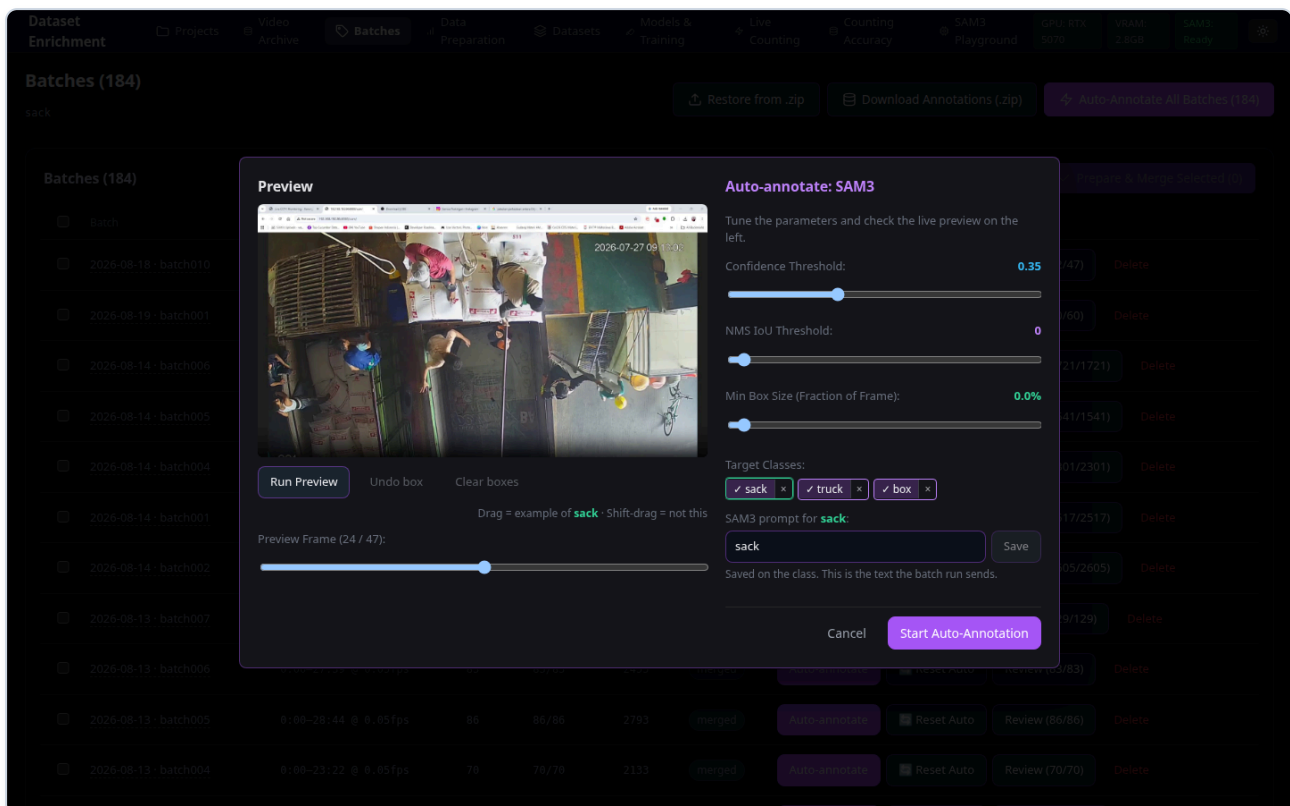
Meta Segment Anything Model 3 (SAM3) adalah arsitektur *vision foundation model* dengan kemampuan melokalisasi dan mengelompokkan objek citra secara zero-shot berdasarkan deskripsi bahasa alami (*open-vocabulary grounding*).

Arsitektur inferensi SAM3 diimplementasikan melalui modul `backend/sam3_engine.py` :

- Model memuat bobot Vision Transformer (ViT) dasar dengan konsumsi VRAM awal ~3.9 GB.
- Eksekusi Tunggal `set_image()`** : Tulang punggung fitur visual (*vision backbone*) memproses citra frame hanya satu kali dan menyimpan representasi fitur (*backbone_out*) pada cache memori GPU.
- Evaluasi Multi-Prompt**: Kepala penyalaras teks (*grounding head*) dieksekusi berulang kali pada representasi fitur yang sama untuk setiap kelas prompt tanpa mengulang komputasi backbone.

6.2 Auto-Labeling Batch Tunggal & Exemplar Tuning

Untuk menjalankan pelabelan otomatis pada batch tertentu, klik tombol `Auto-annotate` pada baris batch.



Gambar 6: Modal Auto-Anotasi SAM3 Tunggal (Auto-Annotate Modal). Kanvas preview deteksi interaktif, slider ambang batas confidence, NMS IoU, filter ukuran minimum kotak, serta bidang gambar kotak exemplar positif dan negatif.

(English label: Single-Batch SAM3 Modal)

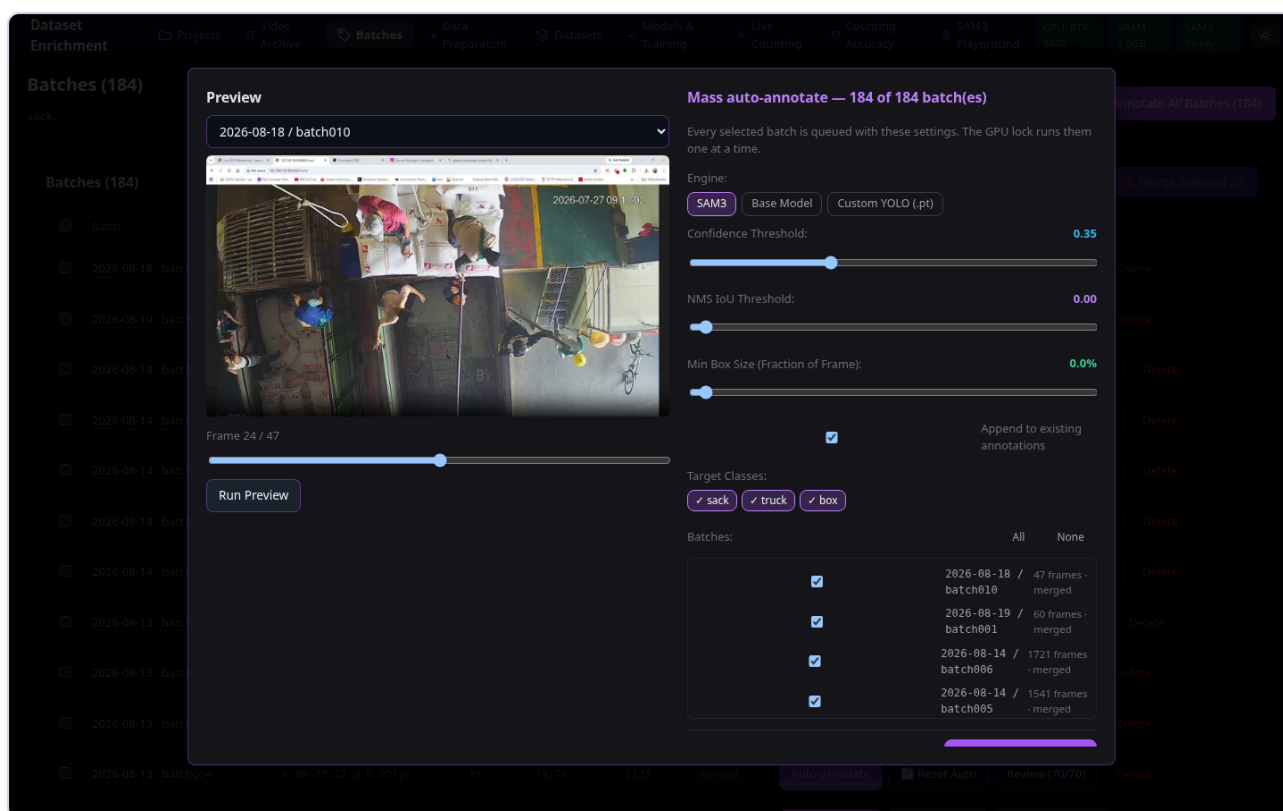
Parameter konfigurasi auto-labeling:

- Confidence Threshold (0.05 sampai 0.95, default 0.35)**: Skor probabilitas minimum deteksi. Naikkan nilai jika muncul deteksi palsu (*false positive*) pada latar belakang konveyor; turunkan nilai jika karung buram tidak terdeteksi.

2. **NMS IoU Threshold (0.0 sampai 0.9, default 0.0):** Ambang batas Non-Maximum Suppression untuk mengeliminasi kotak tumpang tindih dari prompt berbeda (*cross-prompt NMS*). Nilai 0.0 mengaktifkan pembersihan tumpang tindih ketat.
 3. **Min Box Size Fraction (0 sampai 50%, default 0%):** Mengabaikan deteksi objek dengan luas area di bawah persentase tertentu terhadap total luas frame. Berguna untuk memfilter serpihan kecil atau noise debu.
 4. **Interactive Exemplar Prompting:**
 - **Kotak Positif (+1):** Klik dan tarik (*drag*) kursor mouse pada objek karung yang tidak terdeteksi. SAM3 akan memprioritaskan fitur visual objek serupa.
 - **Kotak Negatif (-2):** Tekan tombol `Shift` + tarik kursor mouse pada objek non-target (misalnya kaki operator atau refleksi lantai). SAM3 akan mengabaikan pola visual tersebut.
1. Klik tombol `Run Preview` untuk mengevaluasi hasil penyesuaian parameter sebelum menyimpan ke database.

6.3 Auto-Labeling Massal Multi-Batch

Apabila operator memiliki puluhan batch rekaman yang baru diekstraksi, gunakan fitur auto-labeling massal.



Gambar 7: Modal Auto-Anotasi Massal (Mass Auto-Annotate Modal). Pilihan engine pelabelan (SAM3 Zero-Shot, Base Model v4, atau Model Kustom), daftar centang batch target, dan tombol eksekusi antrian sekuensial.

(English label: Mass Auto-Annotation Modal)

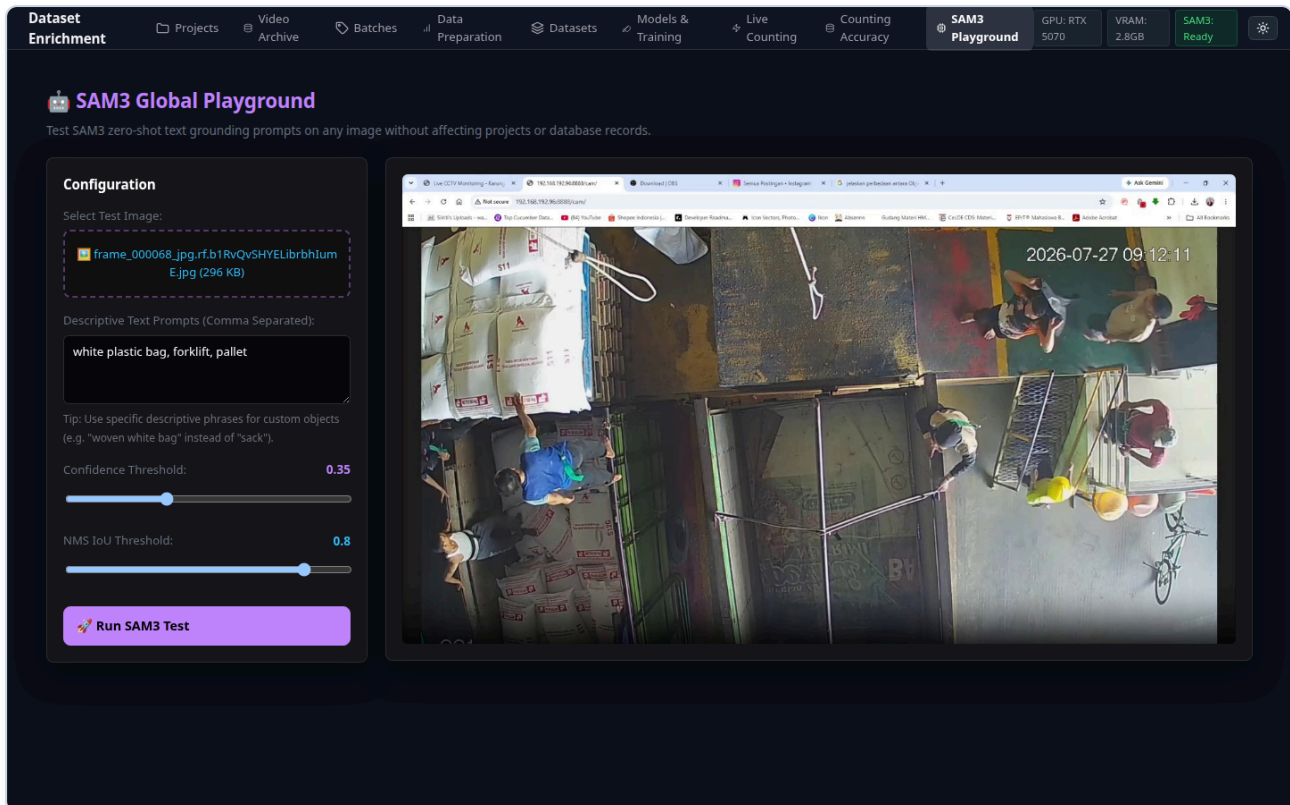
Opsi engine pelabelan:

- **SAM3 Zero-Shot:** Menggunakan Meta SAM3 dengan prompt teks proyek. Sangat fleksibel untuk objek baru.
- **Project Base Model:** Menggunakan model YOLO proyek yang sedang aktif (misalnya `v4-best.pt`). Kecepatan inferensi jauh lebih tinggi dibandingkan SAM3.
- **Custom YOLO Model:** Menggunakan file checkpoint `.pt` khusus yang diunggah operator.

Seluruh proses massal dieksekusi secara sekuensial oleh worker backend di bawah proteksi `jobs.gpu_lock` untuk mencegah benturan VRAM.

6.4 SAM3 Global Playground

Modul Playground (`/sam3-playground`) menyediakan lingkungan uji coba terisolasi untuk menguji efektivitas prompt teks tanpa memengaruhi database proyek aktif.



Gambar 8: SAM3 Global Playground. Area unggah gambar bebas, kotak input multi-prompt teks dipisahkan tanda koma, kanvas visualisasi hasil segmentasi instan, dan pembacaan waktu komputasi GPU.

(English label: SAM3 Prompt Playground)

Panduan penggunaan Playground:

1. Tarik (*drag-and-drop*) berkas gambar JPEG/PNG ke dalam area dropzone.
2. Ketik prompt teks deskriptif pada kolom input (contoh: `white plastic sack, forklift, worker`).
3. Klik tombol `Run SAM3`.
4. Evaluasi ketepatan kontur segmentasi dan skor confidence yang dihasilkan.

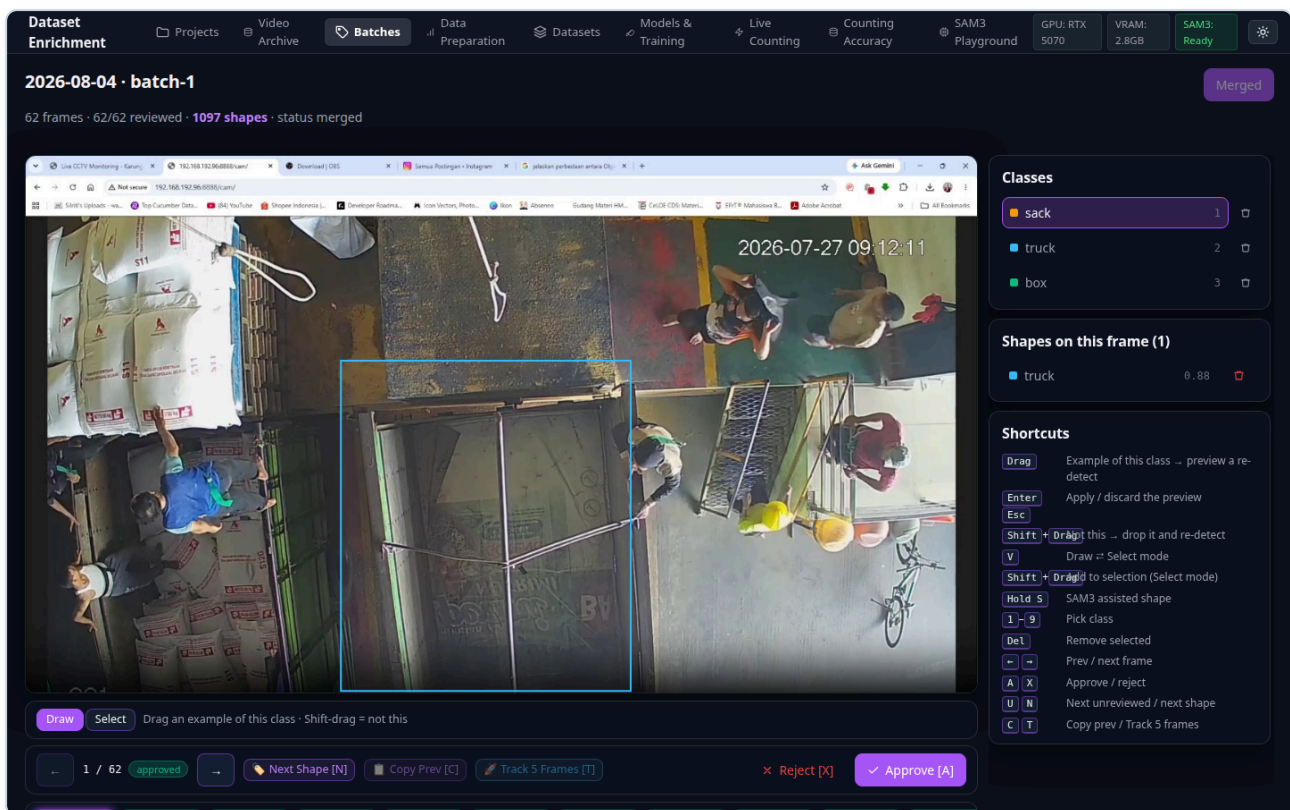
Bab 7: Kanvas Review & Anotasi Interaktif

Bab ini menguraikan fitur penyuntingan anotasi pada kanvas berlatar gelap, sistem koordinat ternormalisasi, navigasi tombol pintas keyboard, dan aturan kardinal anotasi objek konveyor.

7.1 Tata Letak Kanvas Gelap Roboflow-Replica

Modul Review (/projects/{id}/review?batch=<batch_id>) dirancang untuk kenyamanan mata operator selama sesi verifikasi panjang. Seluruh koordinat geometri disimpan dalam format mengambang ternormalisasi **0.0** sampai **1.0** terhadap dimensi lebar dan tinggi frame asli.

Format koordinat ternormalisasi: $x_{\text{norm}} = \frac{x_{\text{piksel}}}{W_{\text{frame}}}$, $y_{\text{norm}} = \frac{y_{\text{piksel}}}{H_{\text{frame}}}$



Gambar 9: Kanvas Review & Anotasi Interaktif (Review Page). Kanvas anotasi dengan bounding box dan kontur poligon, filmstrip status frame di sisi bawah, sidebar daftar kelas dan bentuk objek, serta panel navigasi tombol pintas.

(English label: Canvas Review & Annotation)

7.2 Prosedur Review Cepat Menggunakan Hotkeys

Antarmuka review mendukung kendali penuh berbasis keyboard (*keyboard-first workflow*):

Tabel daftar tombol pintas (Hotkeys):

Tombol Pintas	Fungsi Operasional	Efek pada Sistem
A	Approve Frame	Menandai frame saat ini sebagai approved (hijau) dan otomatis berpindah ke frame berikutnya.
X	Reject Frame	Menandai frame saat ini sebagai rejected (merah) dan berpindah ke frame berikutnya.

Tombol Pintas	Fungsi Operasional	Efek pada Sistem
<- / ->	Navigasi Frame	Berpindah mundur atau maju satu frame pada timeline batch.
N	Next Annotated	Melompat langsung ke frame berikutnya yang memiliki objek anotasi.
C	Copy Previous	Menyalin seluruh anotasi dari frame sebelumnya ke frame aktif saat ini.
Del / Backspace	Delete Shape	Menghapus kotak atau poligon yang sedang aktif/terpilih.
V	Toggle Tool	Beralih mode antara kursor seleksi (<i>Select</i>) dan mode menggambar (<i>Draw</i>).
1 sampai 9	Fast Reclass	Mengubah kelas objek yang dipilih secara instan sesuai nomor urut kelas.
Esc	Cancel / Deselect	Membatalkan seleksi bentuk aktif atau menutup overlay dialog.

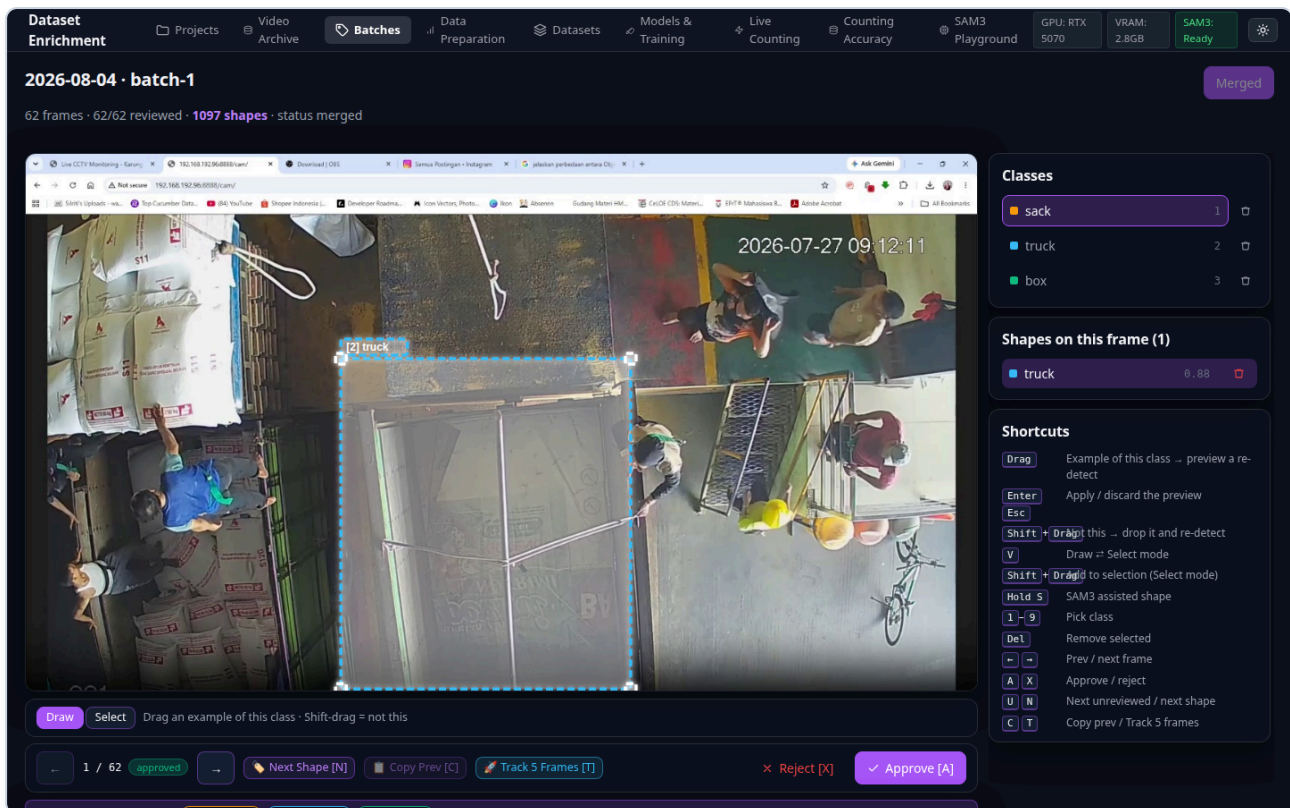
7.3 Penyuntingan Vertex Poligon & Bounding Box

Manipulasi bentuk pada kanvas:

- **Bounding Box:** Klik objek untuk memunculkan 8 titik handle tepi. Tarik handle sudut untuk mengubah skala, atau tarik badan kotak untuk menggeser posisi.
- **Poligon Segmentasi:** Klik poligon untuk memunculkan titik-titik vertex bulat (`.handle-vertex`) dan titik tengah tepi (`.handle-midpoint`).
- Tarik `.handle-vertex` untuk memindahkan sudut kontur.
- Klik `.handle-midpoint` untuk menyisipkan titik sudut baru pada kontur poligon.
- Tekan tombol `Alt` + klik pada titik vertex untuk menghapus titik tersebut (jumlah titik minimum poligon adalah 3).

7.4 Quick Reclass Bar & Filmstrip Navigasi

Pada bagian atas kanvas review, bilah *Quick Reclass Bar* menyediakan akses satu klik untuk mengganti klasifikasi objek.



Gambar 10: Quick Reclass Bar & Filmstrip Navigasi. Bilah penggantian kelas cepat dengan badge warna dan tombol hapus bentuk, dipadukan dengan filmstrip thumbnail frame di bagian bawah kanvas.

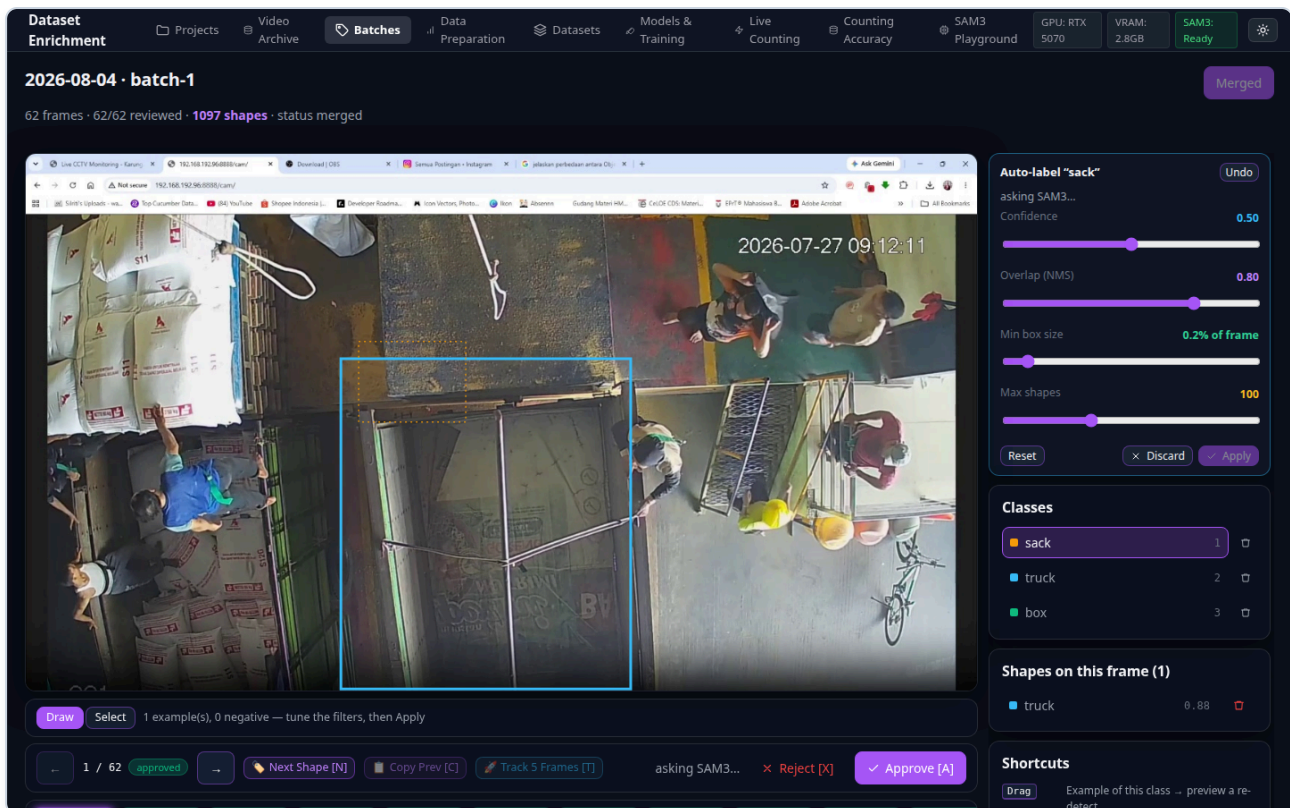
(English label: Quick Reclass & Filmstrip)

Status frame pada filmstrip:

- **Garis Tepi Hijau:** Frame berstatus disetujui (*approved*), siap digabungkan ke dataset master.
- **Garis Tepi Merah:** Frame berstatus ditolak (*rejected*), akan dilewati saat proses merge.
- **Garis Tepi Kuning/Abu-abu:** Frame berstatus tunda (*pending*), belum diperiksa oleh operator.

7.5 Panel Filter Exemplar In-Review

Operator dapat menyetel ulang deteksi SAM3 secara lokal langsung dari sidebar review tanpa perlu kembali ke halaman batch.



Gambar 11: Panel Filter Exemplar Interaktif (Exemplar Filter Panel). Panel sidebar untuk mengatur ambang batas confidence, NMS overlap, batas ukuran minimum, dan kuantitas deteksi maksimum secara real-time.

(English label: In-Review Exemplar Filter Panel)

7.6 Kebijakan Anotasi Kardinal: Penanganan Oklusi Garis Atas (y_1)

Sistem penghitungan live counting mengandalkan pemicu tepi atas kotak karung (y_1). Oleh karena itu, operator wajib mematuhi aturan kardinal anotasi:

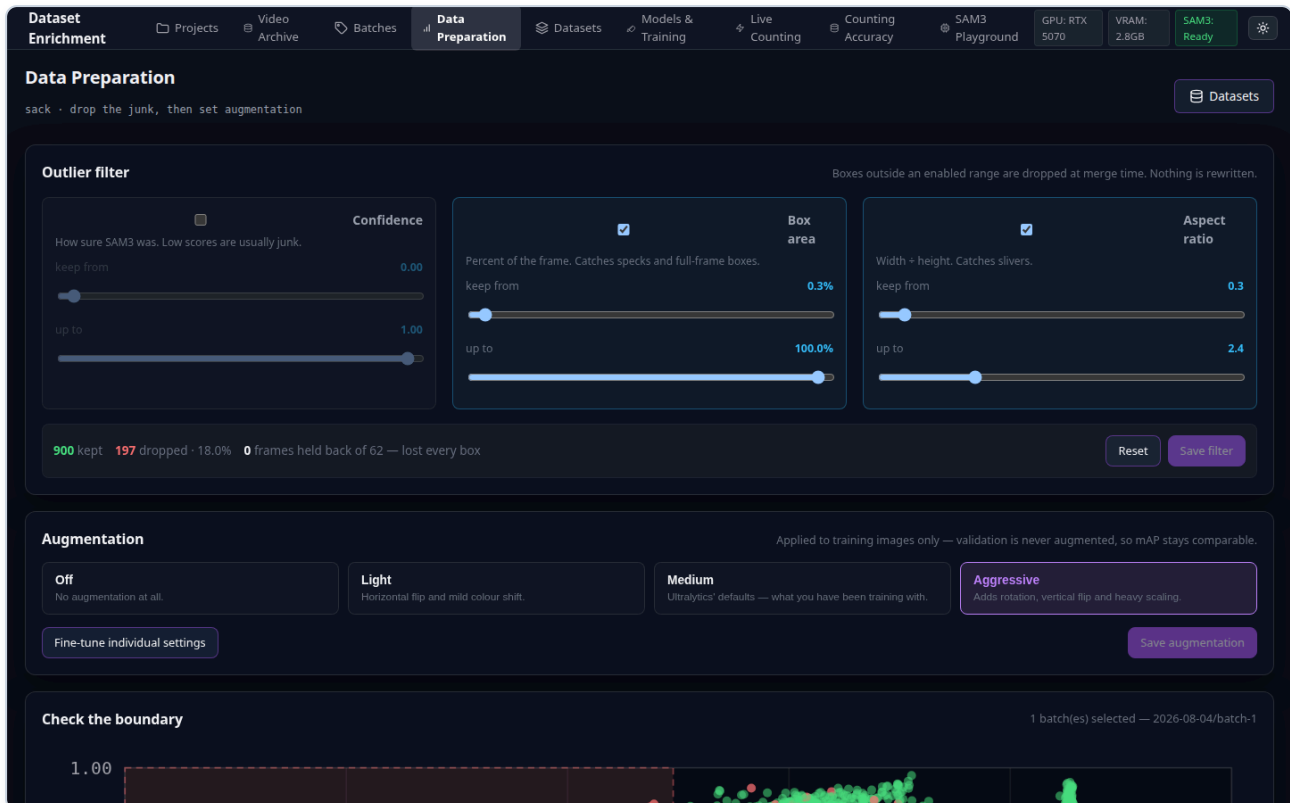
1. **Aturan Oklusi Tepi Atas:** Apabila tepi atas karung terhalang oleh kepala pekerja, tangan operator, atau karung lain yang menumpuk di atasnya, **jangan buat anotasi pada karung tersebut**. Membiarkan kotak deteksi dengan y_1 yang salah akan menyebabkan pemecuan ganda (*double triggering*) pada algoritma counting.
2. **Karung Terpotong Tepi Bawah:** Karung yang hanya terlihat sebagian pada bagian bawah namun memiliki tepi atas yang jelas **wajib dianotasi** hingga batas visual yang tampak.
3. **Pewarisan Anotasi Manual:** Setiap modifikasi manual yang dilakukan operator pada kanvas review akan mengubah status sumber anotasi menjadi `source='manual'`. Anotasi manual terlindungi dan tidak akan tertimpa jika fungsi auto-annotation dijalankan ulang.

Bab 8: Penyiapan Data, Triage Kualitas & Augmentasi

Bab ini membahas modul Data Prep sebagai gerbang kendali mutu (*quality gate*) sebelum data digabungkan ke dataset master, visualisasi scatter plot logaritmik, galeri crop grid, dan konfigurasi augmentasi sintetis.

8.1 Filter Pencilan (Outlier Triage Filter) Non-Destruktif

Modul Outlier Filter (`backend/triage.py`) melakukan evaluasi otomatis pada seluruh anotasi di dalam batch terpilih berdasarkan tiga metrik geometris tanpa menghapus data secara permanen (*non-destructive filtering*).



Gambar 12: Filter Outlier Kualitas Data (Data Prep Outliers). Tiga kartu filter keep-range untuk confidence score, persentase luas area kotak, dan aspect ratio, dilengkapi ringkasan kuantitas objek lolos/terfilter.

(English label: Outlier Quality Filter)

Tiga metrik penapisan outlier:

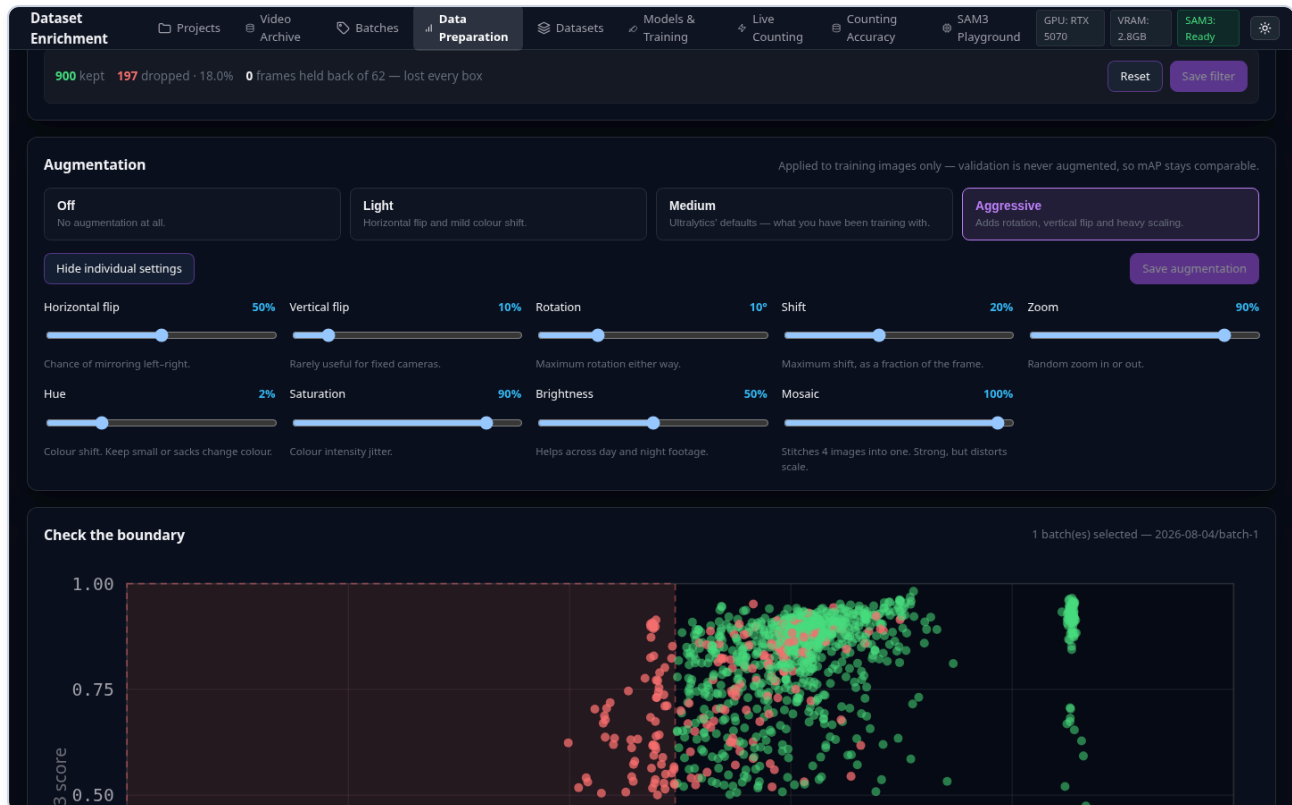
1. **Confidence Score (0.00 sampai 1.00):** Menyingkirkan deteksi otomatis dengan tingkat keyakinan rendah yang berpotensi merupakan false positive.
2. **Box Area % (0.00% sampai 100.00%):** Menyingkirkan objek yang terlalu kecil (noise debu conveyor) atau terlalu besar (artefak background yang mencakup seluruh layar).
3. **Aspect Ratio W/H (0.00 sampai 10.00):** Menyingkirkan kotak deteksi yang terlalu pipih atau terlalu ramping vertikal yang tidak sesuai dengan proporsi fisik karung pakan standar.

Prinsip kerja triage rules:

- Anotasi yang berada di luar rentang batas (*keep-range*) ditandai sebagai `ignored`.
- Anotasi manual (`source='manual'`) memiliki preseden tertinggi dan **selalu dipertahankan** meskipun nilainya berada di luar batas filter.
- Frame yang kehilangan 100% anotasi akibat filter outlier akan ditahan (*held back*) dan tidak dimasukkan ke dalam dataset master untuk mencegah pencemaran citra latar belakang kosong yang tidak disengaja.

8.2 Konfigurasi Augmentasi Citra

Augmentasi data menghasilkan variasi citra sintetis untuk meningkatkan generalisasi model terhadap perubahan kondisi pabrik.



Gambar 13: Panel Konfigurasi Augmentasi Gambar (Augmentation Panel). Pemilihan preset augmentasi (Off, Light, Medium, Aggressive) dan 9 slider penyesuaian parameter geometri serta fotometri.

(English label: Augmentation Preset Panel)

Tabel preset dan parameter augmentasi:

Parameter	Preset Light	Preset Medium (Default)	Preset Aggressive	Deskripsi Transformasi
<code>fliplr</code>	0.5	0.5	0.5	Probabilitas pembalikan horizontal citra (kiri ke kanan).
<code>flipud</code>	0.0	0.0	0.2	Probabilitas pembalikan vertikal citra (atas ke bawah).
<code>degrees</code>	0.0°	5.0°	15.0°	Rentang rotasi acak derajat kemiringan.
<code>translate</code>	0.05	0.10	0.20	Translasi pergeseran posisi gambar secara acak.
<code>scale</code>	0.10	0.20	0.50	Skala pembesaran/pengecilan objek acak.
<code>hsv_h</code>	0.01	0.015	0.03	Variasi spektrum hue warna citra.
<code>hsv_s</code>	0.3	0.5	0.7	Variasi saturasi warna citra.
<code>hsv_v</code>	0.2	0.3	0.5	Variasi kecerahan/kegelapan (<i>value</i>) pencahayaan.
<code>mosaic</code>	0.0	0.5	1.0	Probabilitas penggabungan 4 potongan citra menjadi satu.

Invarian penting: Seluruh parameter augmentasi hanya diterapkan pada subset data latih (train). Subset data validasi (val) tidak pernah diaugmentasi agar evaluasi benchmark tetap murni.

8.3 Analisis Sebaran Logaritmik & Marquee Selection

Scatter plot interaktif (`TriageScatter.jsx`) menyajikan visualisasi sebaran seluruh anotasi dalam grafik dua dimensi: Sumbu Y mewakili Confidence Score (0.0 sampai 1.0) dan Sumbu X mewakili Luas Area Kotak dalam skala logaritmik (10^{-3} hingga 10^0).



Gambar 14: Scatter Plot Triage Score vs Area Logaritmik. Titik-titik anotasi objek dengan garis batas penapisan merah putus-putus, seleksi area kotak (marquee tool), dan bilah penetapan keputusan manual (verdict bar).

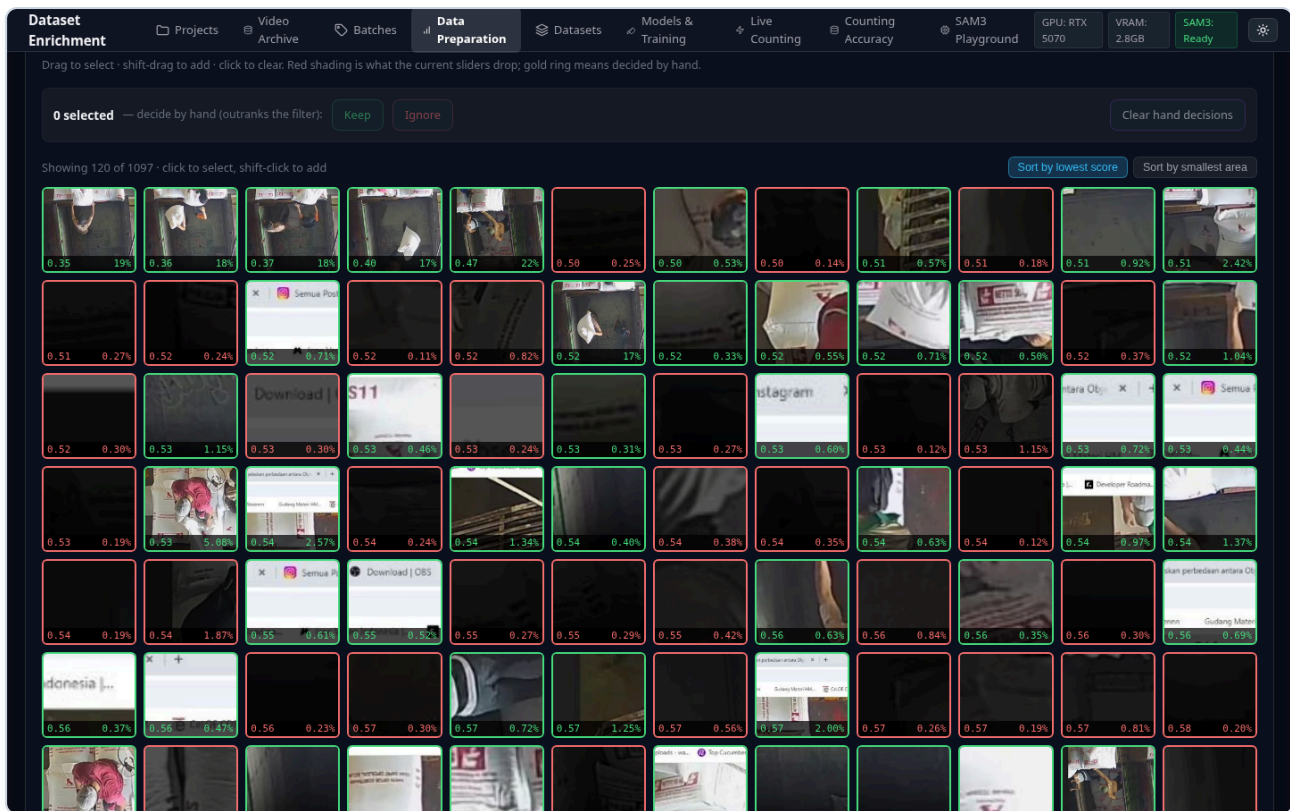
(English label: Triage Scatter Plot Log Scale)

Fitur interaktif Scatter Plot:

- Garis putus-putus merah menandai batas keep-range aktif.
- Operator dapat mengklik dan menarik cursor untuk membuat kotak seleksi (*marquee selection*) pada sekumpulan titik anomali.
- Tombol aksi pada *Verdict Bar*:
 - **Keep** : Menetapkan override manual agar objek terpilih selalu diikutsertakan.
 - **Ignore** : Menetapkan override manual agar objek terpilih selalu dibuang.
 - **Clear hand decisions** : Menghapus keputusan override manual.

8.4 Inspeksi Kualitas Melalui Crop Grid

Galeri Crop Grid (`TriageCropGrid.jsx`) menampilkan potongan thumbnail piksel dari setiap anotasi yang diurutkan dari skor terendah atau ukuran terkecil.



Gambar 15: Grid Crop Triage Objek (Triage Crop Grid). Galeri 120 thumbnail potongan objek resolusi server, garis tepi berwarna penanda status, informasi skor dan persentase area, serta seleksi massal.

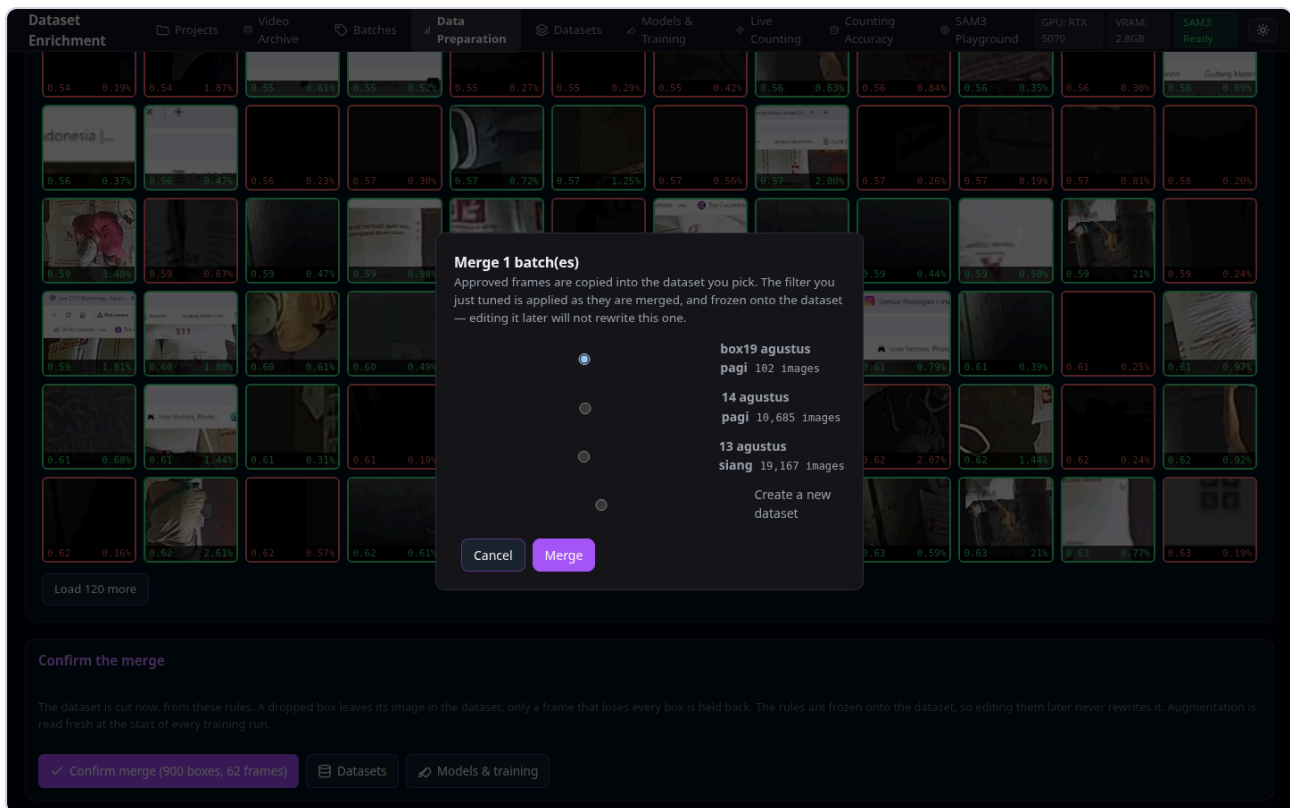
(English label: Triage Crop Grid Inspection)

Fitur Crop Grid:

- Mengambil potongan gambar langsung dari backend secara cepat (`/api/projects/{id}/crop-grid`).
- Garis tepi hijau menunjukkan objek lolos filter, garis tepi abu-abu/merah menunjukkan objek terfilter.
- Operator dapat memilih beberapa thumbnail dan menerapkan keputusan `Keep` atau `Ignore` secara instan.

8.5 Modal Konfirmasi Penggabungan Dataset

Setelah operator memastikan parameter filter dan augmentasi telah optimal, klik tombol `Prepare & Merge Selected`.



Gambar 16: Modal Konfirmasi Merge Dataset (Merge Target Modal). Opsi target dataset (memperbarui dataset aktif atau membuat dataset baru), peringatan batch yang belum direview penuh, dan tombol konfirmasi penggabungan.

(English label: Dataset Merge Target Modal)

Logika penggabungan (merge gate):

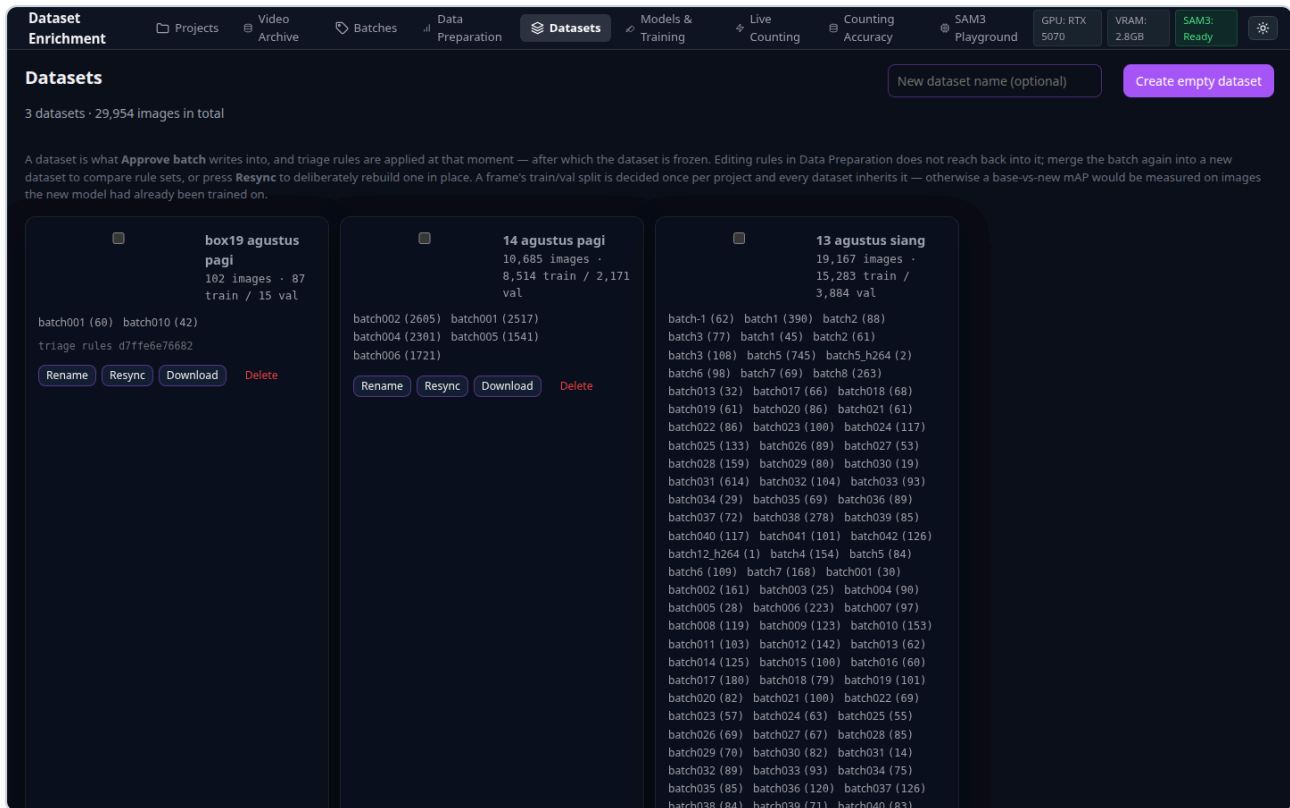
- Jika terdapat batch yang belum berstatus 100% reviewed, sistem menampilkan kotak peringatan kuning. Operator wajib mencentang opsi persetujuan `Merge them unreviewed` untuk melanjutkan.
- Saat konfirmasi diberikan, sistem mengambil snapshot aturan triage aktif dan menyimpannya ke kolom `datasets.rules_json`.

Bab 9: Manajemen Dataset Master & Pembagian Validasi Permanen

Bab ini membahas struktur dataset master, mekanisme pembekuan data (*dataset freezing*), invarian pembagian validasi stabil berbasis hash kriptografis, dan integrasi dataset eksternal.

9.1 Mekanisme Pembekuan Dataset (Dataset Freezing)

Dataset master adalah kumpulan citra dan label teks terverifikasi yang siap digunakan untuk melatih model. Setiap operasi penggabungan (*merge*) bersifat atomik: citra frame disalin ke direktori `data/projects/<slug>/datasets/<id>/images/`, label YOLO diekspor ke `labels/`, dan konfigurasi `data.yaml` dihasilkan secara otomatis.



Gambar 17: Halaman Manajemen Dataset Master (Datasets Page). Ringkasan dataset master, jumlah total citra, rasio pembagian data latih/validasi, riwayat versi aturan triage, dan bilah kalkulasi gabungan multi-dataset.

(English label: Master Datasets & Split View)

9.2 Invarian: Pembagian Validasi Stabil (Stable Validation Split)

Salah satu kesalahan fatal dalam sistem machine learning industri adalah pergeseran data validasi antar waktu (*data leakage / unstable split*). Apabila sebuah frame citra berada pada set validasi pada model v1, lalu berpindah ke set pelatihan pada model v2, maka perbandingan metrik mAP antar kedua model tersebut menjadi tidak valid.

Untuk menjamin stabilitas absolut, sistem mengimplementasikan algoritma hashing pada `backend/dataset.py:split_for()`:

```
import hashlib

def split_for(project_id: int, batch_id: int, frame_stem: str, val_every: int = 5) -> str:
    # Membentuk string identifikasi unik konten frame
    key = f"{project_id}:{batch_id}:{frame_stem}".encode("utf-8")
    hash_digest = hashlib.shal(key).hexdigest()
    # Mengonversi 8 karakter pertama hash ke integer
    hash_int = int(hash_digest[:8], 16)

    # Menentukan alokasi split secara deterministik
    if hash_int % val_every == 0:
        return "val"
    return "train"
```

Karakteristik Pembagian Validasi Stabil:

1. **Deterministik Murni:** Penentuan status `train` atau `val` sepenuhnya ditentukan oleh nama file, ID batch, dan ID proyek.
2. **Kekebalan Mutasi:** Meskipun batch baru ditambahkan atau batch lama dihapus, frame yang pernah masuk ke subset `val` akan selalu tetap berada di subset `val` pada setiap dataset baru yang dibuat.
3. **Penyusunan File Disk:** Citra langsung disalin ke folder terpisah:
 - `images/train/<batch-id>__<frame_idx>.jpg`
 - `images/val/<batch-id>__<frame_idx>.jpg`

9.3 Integrasi Base Datasets Eksternal (Train-Only)

Sistem mendukung integrasi dataset luar (*base datasets*) melalui tabel `base_datasets`. Dataset eksternal ini umumnya berisi ribuan foto karung pakan dari pabrik lain atau domain publik.

Aturan isolasi base dataset:

- Seluruh citra dari base dataset dimasukkan **eksklusif ke subset data latih (`train`)**.
- Citra base dataset tidak pernah diizinkan masuk ke subset validasi (`val`) untuk memastikan benchmark akurasi proyek murni mencerminkan performa pada kamera lini produksi lokal.

9.4 Operasi Resync, Kombinasi Multi-Dataset & Ekspor ZIP

Aksi pada kartu dataset:

- **Resync Rules:** Menerapkan ulang aturan triage terbaru ke dataset yang telah ada. *Perhatian: Operasi ini bersifat destruktif terhadap snapshot aturan sebelumnya dan hanya boleh dilakukan jika terjadi pembaruan kebijakan mutu data.*
- **Combine Preview Strip:** Centang lebih dari satu kotak dataset pada halaman Datasets. Bilah atas akan menampilkan kalkulasi instan: `N selected · T unique images · X train / Y val`. Saat pelatihan model dijalankan, sistem menggabungkan dataset-dataset tersebut secara virtual tanpa duplikasi file.
- **Download (.zip):** Mengunduh arsip dataset lengkap berstruktur standar YOLO (folder `images`, `labels`, dan `data.yaml`) untuk keperluan pelatihan eksternal.

Bab 10: Pelatihan Model YOLO & Evaluasi Benchmark

Bab ini memandu prosedur penyetelan halus (*fine-tuning*) bobot YOLO11, pemantauan log pelatihan real-time, evaluasi komparatif metrik mAP, serta promosi bobot model terbaik.

10.1 Konfigurasi Pelatihan Model

Pelatihan model dilakukan melalui antarmuka Models (`/projects/{id}/models`).

Metric	Base	This version	Δ
mAP50	0.6461	0.9443	+0.2982
mAP50-95	0.5926	0.8286	+0.2360
precision	0.6110	0.9021	+0.2911
recall	0.6122	0.8808	+0.2686

Metric	Base	This version	Δ
mAP50	0.9425	0.9698	+0.0273
mAP50-95	0.8409	0.8895	+0.0486
precision	0.8894	0.9172	+0.0278
recall	0.8971	0.9186	+0.0215

Gambar 18: Halaman Konfigurasi Training & Evaluasi Model (Models Page). Panel konfigurasi parameter epoch, probe perangkat keras otomatis, pemilihan kombinasi dataset, serta tabel perbandingan metrik evaluasi model terhadap base model.

(English label: YOLO Model Training & Metrics)

Langkah konfigurasi pelatihan:

- Pilih Base Model Weights:** Pilih checkpoint dasar (misalnya `v4-best.pt` atau bobot pretrained Ultralytics `yolo11n.pt` / `yolo11n-seg.pt`).
- Target Classes:** Pilih kelas deteksi yang akan dilatih.
- Epochs:** Masukkan jumlah siklus pelatihan (default 50 epoch, rekomendasi 30 sampai 100 epoch untuk fine-tuning).
- Hardware Auto-Probe:** Sistem memeriksa kapasitas VRAM GPU host secara otomatis:
 - VRAM < 8 GB: `batch=8, imgsz=640`
 - VRAM 8 sampai 16 GB: `batch=16, imgsz=640`
 - VRAM > 16 GB: `batch=32, imgsz=768`
 - CPU Fallback: `batch=4, imgsz=512`
- Pilih Dataset Pelatihan:** Centang dataset master dan base dataset yang akan dilibatkan dalam pelatihan.
- Klik tombol `Start Training` .

10.2 Manajemen Memori GPU & Eksekusi Training

Saat tombol pelatihan diklik:

- Backend mengakuisisi kunci eksklusif GPU (`jobs.gpu_lock`).
- Jika engine SAM3 sedang aktif di VRAM, sistem secara otomatis mengeksekusi `sam3_engine.release_engine()`, memanggil `torch.cuda.empty_cache()`, dan membersihkan memori agar 100% kapasitas VRAM dapat digunakan oleh proses pelatihan YOLO.
- Skrup pelatihan `backend/training.py` mengeksekusi library Ultralytics dengan parameter optimal:
 - Optimizer: `AdamW` atau `SGD` (otomatis)
 - Cache mode: RAM caching jika RAM host > 16 GB, atau disk caching jika RAM terbatas.
 - Workers: 4 thread loader.

10.3 Monitoring Progres Pelatihan Real-Time

Selama pelatihan berlangsung, antarmuka web menampilkan kartu pekerjaan aktif dengan terminal log interaktif.

The screenshot shows the Ultralytics web interface with the 'Models & Training' tab selected. The top navigation bar includes options like 'Dataset Enrichment', 'Projects', 'Video Archive', 'Batches', 'Data Preparation', 'Datasets', 'Models & Training', 'Live Counting', 'Counting Accuracy', 'SAM3 Playground', 'GPU: RTX 5070', 'VRAM: 2.8GB', and 'SAM3: Ready'. The main content area is titled 'sack · models' and shows a 'Master dataset: 23884 train / 6070 val from 184 merged batch(es)'. The 'Base Model Configuration' section allows fine-tuning a baseline model. The 'Train Model' section shows target classes (sack, truck, box) and epochs (50). The 'Training Job #1600 (running)' section displays a progress bar and a terminal log with the following output:

```
[04:47:44] Fine-tuning model.pt for 50 epoch(s) (batch=64, imgsz=640, device=0)
[04:47:44] Augmentation: aggressive - degrees=10, fliplr=0.5, flipud=0.1, mosaic=1
[04:47:44] Image cache: disk (RAM cache would need ~17.1 GB, 28.1 GB free)
[04:48:12] Epoch 30/50: loss=0.0312, mAP50=0.912, precision=0.884, recall=0.862
[04:48:25] Epoch 31/50: loss=0.0298, mAP50=0.924, precision=0.891, recall=0.871
[04:48:38] Epoch 32/50: loss=0.0285, mAP50=0.931, precision=0.898, recall=0.877
```

The 'Trained Model Versions (5)' section shows a table of metrics for different versions:

Metric	Base	This version	Δ
mAP50	0.6461	0.9443	+0.2982
mAP50-95	0.5926	0.8286	+0.2360
precision	0.6110	0.9021	+0.2911
recall	0.6122	0.8808	+0.2686

The interface also includes a 'Select Datasets (3/3)' section and a 'Cancel Job' button.

Gambar 19: Status Monitoring Progres Pelatihan Real-Time. Indikator progres epoch aktif, utilisasi VRAM, kurva penurunan box_loss, cls_loss, dfl_loss, estimasi sisa waktu (ETA), dan tombol pembatalan pekerjaan.

(English label: Active Training Job & Logs)

Informasi pada terminal log:

- Nilai kerugian box loss (`box_loss`), class loss (`cls_loss`), dan distribution focal loss (`dfl_loss`).
- Metrik presisi dan recall per epoch.
- Tombol `Cancel` : Mengirimkan sinyal terminasi ke worker pelatihan dan membersihkan direktori sementara `runs/`.

10.4 Evaluasi Komparatif Like-for-Like

Setelah proses pelatihan selesai, modul `backend/evaluate.py` secara otomatis mengevaluasi performa model baru (`best.pt`) dan membandingkannya secara langsung (*like-for-like*) dengan model dasar (*base model*) menggunakan subset validasi yang identik (`data.yaml`).

Tabel metrik evaluasi model:

Versi Model	Status	mAP50	mAP50-95	Precision	Recall	Signed Delta mAP50	Status Keputusan
v1	Arsip	0.884	0.612	0.891	0.875	Basis Awal	Model Awal
v2	Arsip	0.912	0.654	0.920	0.898	+0.028 (Hijau)	Ditingkatkan
v3	Arsip	0.938	0.701	0.942	0.925	+0.026 (Hijau)	Ditingkatkan
v4	Base Aktif	0.965	0.748	0.968	0.952	+0.027 (Hijau)	Standar Produksi
v5	Kandidat	0.978	0.772	0.981	0.969	+0.013 (Hijau)	Siap Dipromosikan

Penjelasan nilai Delta \Delta:

- **Nilai Positif Hijau (+0.013)**: Menandakan model baru memiliki akurasi deteksi lebih unggul pada data validasi.
- **Nilai Negatif Merah (-0.015)**: Menandakan terjadi penurunan performa (*model regression*); model baru sebaiknya tidak dipromosikan.

10.5 Promosi Model Baru (Model Promotion)

Jika model kandidat (misalnya `v5`) terbukti menghasilkan delta mAP positif dan lolos pengujian:

1. Klik tombol `Use as base model` pada baris model tersebut.
2. Sistem secara atomik menyalin file bobot `data/projects/<slug>/models/5/best.pt` ke jalur model dasar proyek `data/projects/<slug>/base/model.pt`.
3. Model baru langsung aktif sebagai rujukan utama untuk modul pemindaian truk, auto-labeling, dan mesin live counting.

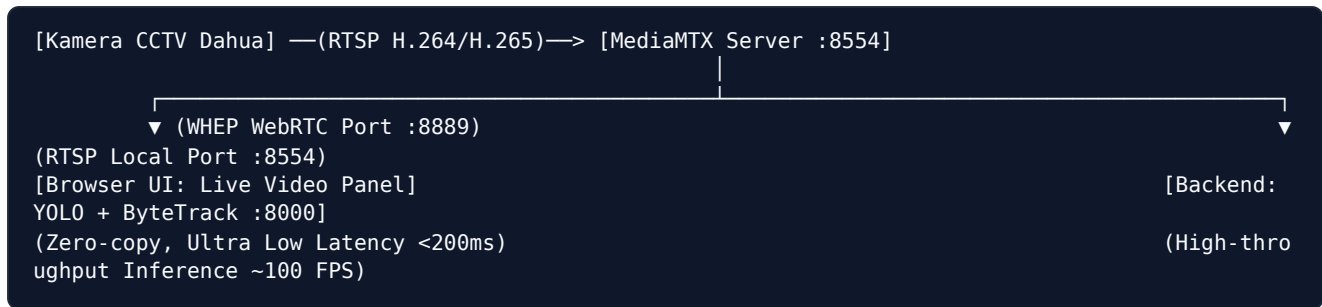
Bab 11: Sistem Live Counting & Integrasi Kamera CCTV

Bab ini menguraikan arsitektur penanganan aliran video kamera, kalibrasi posisi garis pemicu hitung (*tripwire*), algoritma pelacakan ByteTrack, dan penyetelan parameter histeresis.

11.1 Topologi Streaming MediaMTX

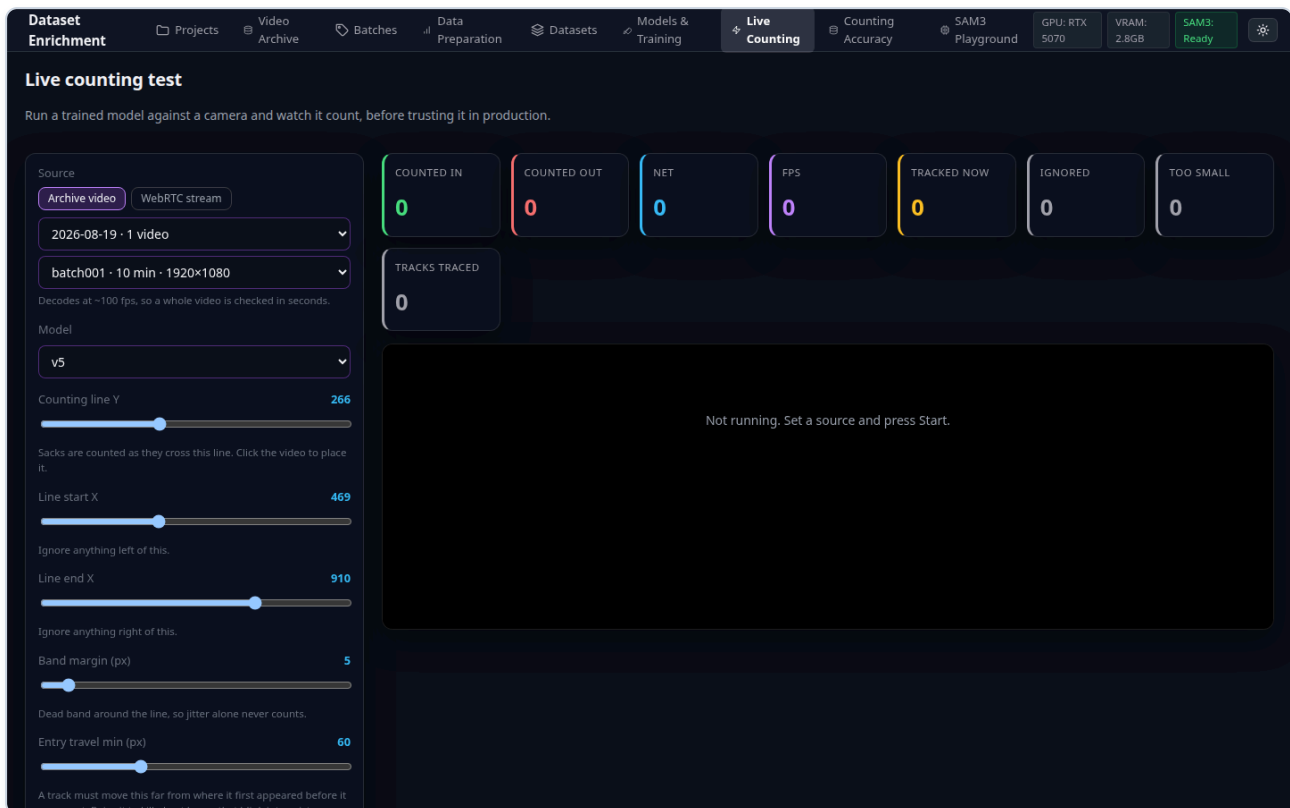
Kamera industri Dahua IPC-HFW1230 mengirimkan stream RTSP beresolusi 1080p / 704x576 pada 25 FPS ke server MediaMTX.

Topologi distribusi video:



Perbedaan fungsi kedua jalur:

1. **Jalur Browser (WHEP WebRTC):** Ditransmisikan langsung dari MediaMTX ke elemen `<video>` peramban untuk monitoring operator dengan latensi ultra rendah tanpa membebani CPU backend.
2. **Jalur Inferensi Backend (RTSP):** Dibaca oleh OpenCV backend untuk proses deteksi objek, pelacakan trajectory, dan kalkulasi garis hitung.



Gambar 20: Antarmuka Live Counting & Kalibrasi Tripwire (Live Count). Tampilan visual stream kamera, penempatan garis hitung interaktif di kanvas, slider parameter pelacakan, dan ubin statistik perhitungan real-time.

(English label: Live Inference & Tripwire Panel)

11.2 Penempatan Garis Hitung Interaktif & Kalibrasi Tripwire

Pada antarmuka Live Count (`/projects/{id}/live-count`), operator dapat mengkalibrasi garis tripwire secara visual langsung di atas kanvas video:

- Klik chip `Counting line` lalu klik posisi vertikal konveyor pada video untuk mengatur nilai `line_y` (koordinat normalisasi **0.0** sampai **1.0**).
- Klik chip `Left edge` lalu klik batas kiri konveyor untuk mengatur nilai `line_x_start`.
- Klik chip `Right edge` lalu klik batas kanan konveyor untuk mengatur nilai `line_x_end`.
- Seluruh perubahan koordinat garis disimpan langsung ke database dan diterapkan secara instan ke mesin hitung (`/api/live-count/line`).

11.3 Algoritma Pelacakan ByteTrack & LineCrossCounter pada y1

Mesin pelacakan objek (`algoritma-batch/src/tracker.py` dan `counting.py`) menggabungkan dua lapisan algoritma:

- ByteTrack Multi-Object Tracker:** Mengasosiasikan kotak deteksi antar frame menggunakan matriks kemiripan IoU dan Kalman Filter. ByteTrack mempertahankan ID objek meskipun karung mengalami oklusi sementara dengan parameter `track_buffer: 60` (mampu mengingat lintasan objek hingga 60 frame atau 2.4 detik).
- LineCrossCounter Triggering pada y1:**
 - Pemicu perhitungan didasarkan secara eksklusif pada **koordinat tepi atas kotak (y1)**.
 - Arah pergerakan dihitung dari perpindahan vektor lintasan: jika `y1` bergerak dari atas garis (`y1 < \text{line\_y}`) melewati garis menuju ke bawah (`y1 \geq \text{line\_y}`), sistem mencatat peristiwa **Count In**.
 - Sebaliknya, perpindahan dari bawah ke atas dicatat sebagai **Count Out**.

11.4 Penanganan Histeresis Lintasan & Parameter Tracking

Tabel parameter pelacakan live counting:

Parameter	Rentang Nilai	Default	Deskripsi Fungsi
<code>line_y</code>	0.10 sampai 0.90	0.50	Posisi koordinat vertikal garis hitung tripwire pada frame.
<code>margin</code>	10 sampai 100 px	30 px	Lebar zona toleransi histeresis di sekitar garis hitung.
<code>entry_travel_min</code>	10 sampai 200 px	40 px	Jarak perpindahan minimum yang wajib ditempuh objek sebelum diizinkan memicu hitungan. Mencegah false trigger dari noise getaran konveyor.
<code>handoff_radius</code>	20 sampai 250 px	100 px	Radius pencarian serah-terima lintasan (<i>track handoff</i>). Jika ID objek terputus akibat oklusi mendadak, objek baru dalam radius ini mewarisi riwayat lintasan ID lama.
<code>unload_confirm_frames</code>	1 sampai 30 frame	5 frame	Jumlah frame konfirmasi sebelum status bongkar dikunci.
<code>min_area_scale</code>	0.001 sampai 0.10	0.008	Ambang batas fraksi luas area minimum objek yang diakui.
<code>conf</code>	0.10 sampai 0.90	0.35	Ambang batas confidence deteksi YOLO pada stream video.

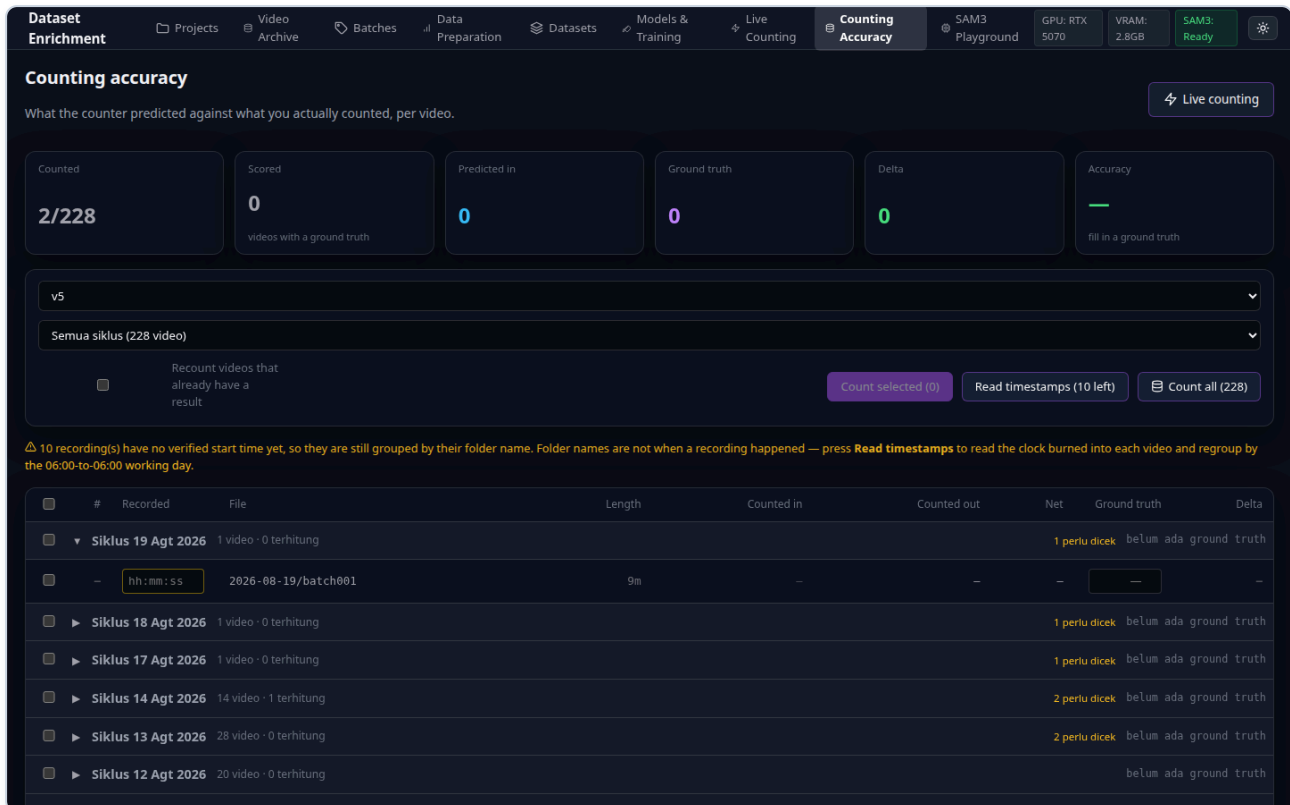
Eliminasi Deteksi Semu (Ghost Box): Objek baru yang tiba-tiba terdeteksi pertama kali tepat di bawah garis hitung tanpa memiliki riwayat lintasan di atas garis diklasifikasikan sebagai *ghost detection* dan diabaikan oleh sistem.

Bab 12: Benchmark Akurasi Perhitungan Headless

Bab ini membahas modul evaluasi akurasi perhitungan berkecepatan tinggi tanpa antarmuka grafis (*headless counting bench*), integrasi data acuan kebenaran (*ground truth*), dan diagnosa kesalahan hitung.

12.1 Eksekusi Rekalkulasi Video Batch Headless (~146 FPS)

Modul Counting Bench (`/projects/{id}/counting-bench`) dirancang untuk memvalidasi akurasi model pada ratusan berkas video arsip secara cepat. Dengan menonaktifkan rendering antarmuka dan enkripsi transmisi WebSocket, pemrosesan video pada GPU dapat mencapai kecepatan **146 hingga 220 FPS** (dibandingkan mode live stream visual yang dibatasi pada 25 FPS).



Gambar 21: Tabel Benchmark Akurasi Perhitungan (Counting Bench). Ringkasan total video terhitung, perbandingan hasil hitungan AI terhadap data acuan kebenaran (Ground Truth), kolom signed delta error, persentase akurasi per siklus 24 jam, dan tombol pemindaian OCR.

(English label: Counting Accuracy Benchmark Table)

12.2 Impor Data Acuan Kebenaran (Ground Truth) dari Excel

Data acuan kebenaran (*ground truth*) adalah angka riil hasil penghitungan manual oleh petugas tally pabrik.

Metode pengisian Ground Truth:

- Impor Berkas Excel (docs/GT.xlsx):** Sistem membaca berkas spreadsheet yang memuat kolom tanggal, nomor batch, dan kuantitas karung fisik, lalu memetakan nilainya ke tabel `count_runs` database secara otomatis.
- Koreksi Manual Langsung di Tabel:** Operator dapat mengklik kotak input angka pada kolom `Ground Truth` di antarmuka Counting Bench dan mengisi angka hitungan fisik secara manual.

12.3 Evaluasi Metrik Signed Delta Error & Akurasi

Perhitungan selisih kesalahan dihitung menggunakan formula *Signed Delta Error* (Δ): $\Delta = \text{Counted}_{AI} - \text{Ground Truth}$

Kategori status pada tabel benchmark:

- **Tanda Nol (0) Hijau:** Hitungan model AI persis sama dengan data fisik riil (Akurasi 100%).
- **Tanda Positif (+N) Kuning/Oranye:** Terjadi penghitungan berlebih (*overcounting*) sebanyak *N* karung.
- **Tanda Negatif (-N) Merah:** Terjadi penghitungan kurang (*undercounting*) sebanyak *N* karung.

Formula Akurasi Total Siklus: $\text{Akurasi (\%)} = \left(1 - \frac{\sum |\text{Counted}_{AI} - \text{Ground Truth}|}{\sum \text{Ground Truth}}\right) \times 100\%$

Perhatian: Kalkulasi total akurasi hanya dihitung pada baris video yang telah memiliki nilai Ground Truth non-kosong.

12.4 Diagnosa & Mitigasi Kesalahan Hitung

Panduan mitigasi deviasi hitungan:

Tabel diagnosa masalah counting:

Gejala Masalah	Penyebab Teknis Utama	Solusi Penanganan Rekayasa
Penghitungan Berlebih (+Delta)	1. Karung memantul di atas garis tripwire sehingga memicu garis dua kali. 2. ID track terputus lalu muncul ID baru tepat di garis. 3. Pekerja berdiri di area garis hitung.	1. Perlebar nilai parameter <code>margin</code> (misal dari 30px ke 50px). 2. Naikkan nilai <code>handoff_radius</code> (misal dari 100px ke 150px). 3. Sesuaikan batas koordinat horizontal <code>line_x_start</code> dan <code>line_x_end</code> agar tidak mencakup area berdiri operator.
Penghitungan Kurang (-Delta)	1. Dua karung bertumpuk rapat dihitung sebagai satu objek (oklusi). 2. Kecepatan konveyor terlalu tinggi sehingga objek melompati garis toleransi dalam 1 frame. 3. Confidence model terlalu tinggi pada karung kusam.	1. Tambahkan data latih karung tumpuk dan latih ulang model. 2. Turunkan nilai <code>entry_travel_min</code> atau posisikan garis hitung di area konveyor yang lebih stabil. 3. Turunkan threshold <code>conf</code> dari 0.35 ke 0.25 pada pengaturan Live Count.

Bab 13: Referensi Teknis, Jaringan, Skema Database & File Layout

Bab ini memuat spesifikasi arsitektur komputasi, matriks alokasi port jaringan, struktur tata letak direktori penyimpanan, skema relasional database SQLite, inventaris endpoint REST API, dan ringkasan perintah CLI.

13.1 Matriks Alokasi Port Jaringan Lengkap

Seluruh alokasi port jaringan pada arsitektur sistem:

Port	Protokol	Layanan / Komponen	Lingkungan	Deskripsi Operasional
:8080	HTTP / TCP	Nginx Web Studio	Docker (Produksi)	Reverse proxy frontend SPA dan perutean endpoint API <code>/api/*</code> .
:8000	HTTP / WS	FastAPI Application Backend	Docker / Lokal	Server REST API utama, manajemen antrean tugas, dan stream data.
:5173	HTTP / TCP	Vite Development Server	Lokal (Dev Mode)	Server live-reload antarmuka React dengan proxy API internal.
:5000	HTTP / WS	Flask Live Counter Standalone	Jetson / Host	Dashboard mandiri inferensi live counter di tepi lini produksi.
:8554	RTSP / TCP	MediaMTX RTSP Server	Host / Gateway	Ingest aliran video H.264/H.265 resolusi penuh dari kamera CCTV.
:8889	HTTP / WebRTC	MediaMTX WHEP Server	Host / Gateway	Endpoint WebRTC WHEP latensi rendah untuk monitoring peramban.

13.2 Tata Letak Direktori Sistem (Filesystem Layout)

Pemisahan tegas diterapkan antara direktori kode sumber dan volume penyimpanan status:

```

reTraining/
├── data/
│   ├── app.db
│   ├── archive/
│   ├── live-count/
│   └── projects/<slug>/
│       ├── base/model.pt
│       ├── datasets/<id>/
│       │   ├── images/{train,val}/
│       │   ├── labels/{train,val}/
│       │   └── data.yaml
│       ├── batches/<id>/frames/
│       └── models/<n>/
│           ├── best.pt
│           └── metrics.json
├── backend/
└── file)
    ├── main.py
    ├── sam3_engine.py
    ├── training.py
    ├── live_count.py
    └── dataset.py
├── frontend/
│   ├── src/pages/
│   └── src/components/
├── sam3/
├── algoritma-batch/
├── screenshots/
└── docs/

# Volume utama persistensi data aplikasi
# Database SQLite (mode WAL, 14 tabel relasional)
# Arsip video CCTV read-only (<date>/<batch>.mp4)
# Berkas jejak trajektori sesi hitung (.jsonl)
# Ruang kerja terisolasi per proyek
# Bobot checkpoint model dasar aktif
# Dataset master beku (immutable)
# Berkas citra frame JPEG (<batch>_<idx>.jpg)
# File teks label anotasi ternormalisasi YOLO
# Definisi konfigurasi dataset Ultralytics
# Direktori frame hasil pemotongan (%06d.jpg)
# Arsip versi model hasil retraining
# Bobot optimal hasil pelatihan
# Rekam perbandingan metrik evaluasi mAP
# Modul aplikasi FastAPI Python (maksimal 400 baris per
# Inisialisasi aplikasi dan perutean router
# Singleton GPU engine manager Meta SAM3
# Pipeline retraining dan auto-tuning hardware YOLO
# Controller live counting dan kalibrasi tripwire
# Logika pembagian validasi stabil dan ekspor data
# Aplikasi Single Page Application React 19 + Vite 7
# Komponen halaman (Projects, Review, DataPrep, Models)
# Komponen antarmuka modular (Canvas, Scatter, CropGrid)
# Salinan dependensi Meta SAM3 (vendor read-only)
# Modul algoritma pelacakan ByteTrack dan tripwire
# Katalog 21 tangkapan layar antarmuka terverifikasi
# Dokumentasi master sistem dan spesifikasi teknis

```

13.3 Skema Database Relasional SQLite (app.db)

Aplikasi menggunakan database SQLite dengan Write-Ahead Logging (`PRAGMA journal_mode=WAL;`).

Tabel inventaris skema database:

Nama Tabel	Jumlah Kolom Utama	Kunci Utama (PK)	Deskripsi Isi Tabel
<code>projects</code>	9	<code>id</code>	Metadata proyek, nama slug, tipe geometri, stride split, dan path arsip.
<code>project_classes</code>	5	<code>id</code>	Taksonomi kelas deteksi, warna swatch, dan pemetaan prompt teks SAM3.
<code>batches</code>	11	<code>id</code>	Rekam batch ekstraksi video, range timecode, FPS, dan status review.
<code>frames</code>	7	<code>id</code>	Indeks frame citra per batch, path file, dan status review (approved/rejected).
<code>annotations</code>	9	<code>id</code>	Data koordinat geometri (BBBox/Polygon), kelas, skor, dan sumber (auto/manual).
<code>datasets</code>	8	<code>id</code>	Dataset master beku, timestamp pembuatan, dan snapshot <code>rules_json</code> .
<code>dataset_items</code>	6	<code>id</code>	Pemetaan relasi frame citra ke dataset master beserta alokasi split (<code>train</code> / <code>val</code>).
<code>base_datasets</code>	6	<code>id</code>	Pendaftaran dataset eksternal (kontributor data latih khusus).
<code>video_clock</code>	6	<code>id</code>	Hasil pembacaan jam OCR CCTV, tanggal siklus 06:00, dan status verifikasi.
<code>count_runs</code>	12	<code>id</code>	Hasil kalkulasi counting AI, nilai Ground Truth manual, dan signed delta.
<code>model_versions</code>	10	<code>id</code>	Versi model hasil pelatihan, path file <code>best.pt</code> , dan rekam <code>metrics.json</code> .
<code>jobs</code>	9	<code>id</code>	Antrean tugas latar belakang server (ekstraksi, auto-label, training, counting).
<code>triage_rules</code>	7	<code>id</code>	Riwayat konfigurasi filter penciran outlier dan rentang keep-range.
<code>annotation_overrides</code>	6	<code>id</code>	Keputusan override manual operator (<code>keep</code> / <code>ignore</code>) dari modul triage.

13.4 Inventaris Endpoint REST API Backend

Daftar endpoint REST API utama pada backend FastAPI:

Metode HTTP	Jalur Endpoint API	Modul Handler	Deskripsi Fungsi
<code>GET</code>	<code>/api/health</code>	<code>backend/main.py</code>	Pemeriksaan kesehatan server dan status kesiapan CUDA GPU.
<code>GET</code> / <code>POST</code>	<code>/api/projects</code>	<code>backend/projects.py</code>	Mengambil daftar proyek aktif atau membuat proyek baru.

Metode HTTP	Jalur Endpoint API	Modul Handler	Deskripsi Fungsi
GET / POST	/api/projects/{id}/batches	backend/batches.py	Manajemen batch dan pendaftaran tugas pemotongan video.
POST	/api/batches/{id}/auto-annotate	backend/autolabel.py	Mendaftarkan pekerjaan auto-labeling SAM3 untuk batch tunggal.
GET / PUT	/api/frames/{id}/annotations	backend/review.py	Mengambil atau memperbarui anotasi geometri pada kanvas review.
POST	/api/projects/{id}/trriage/preview	backend/trriage.py	Menghitung simulasi hasil filter outlier pada batch terpilih.
POST	/api/projects/{id}/datasets/merge	backend/datasets.py	Menggabungkan batch ke dataset master dan membekukan aturan triage.
POST	/api/projects/{id}/train	backend/training.py	Memulai proses pelatihan model YOLO pada GPU worker queue.
GET	/api/jobs/{id}/stream	backend/jobs.py	Server-Sent Events (SSE) streaming log pelatihan real-time.
POST	/api/models/{id}/promote	backend/models.py	Mempromosikan versi model baru menjadi base model proyek.
GET / POST	/api/live-count/line	backend/live_count.py	Mengambil atau memperbarui konfigurasi koordinat tripwire.
POST	/api/counting-bench/run	backend/counting_bench.py	Menjalankan evaluasi headless counting pada rekaman video arsip.

13.5 Lembar Panduan Perintah CLI (Command-Line Cheat-Sheet)

Kumpulan perintah praktis untuk pemeliharaan sistem dari terminal:

```
# 1. Memulai stack kontainer produksi
./start.sh

# 2. Menghentikan seluruh kontainer
docker compose down

# 3. Melihat log backend secara live
docker compose logs -f backend

# 4. Memeriksa ketersediaan GPU dan versi PyTorch di lingkungan uv
uv run python -c "import torch; print('CUDA:', torch.cuda.is_available(), '| Device:', torch.cuda.get_device_name(0))"

# 5. Menjalankan verifikasi integritas tangkapan layar (21 Gambar)
uv run python scripts/verify_screenshots.py

# 6. Memeriksa status database SQLite
sqlite3 data/app.db "PRAGMA journal_mode; SELECT count(*) FROM projects; SELECT count(*) FROM data sets;"

# 7. Menguji inferensi model YOLO secara langsung dari CLI
uv run yolo detect predict model=data/projects/feedmill/base/model.pt source=data/projects/feedmill/batches/1/frames/000001.jpg save=True
```

Bab 14: Invarian Domain, Penanganan Kasus Batas & Pemecahan Masalah

Bab penutup ini merangkum invarian domain yang tidak boleh dilanggar, panduan penanganan skenario kasus batas (*edge cases*), dan tabel pemecahan masalah teknis (*troubleshooting*).

14.1 Invarian Domain Tak Tergoyahkan (Core System Invariants)

Tiga invarian utama yang menjamin kebenaran ilmiah dan stabilitas sistem:

1. Invarian 1: Pembagian Validasi Stabil (Stable Validation Split)

Prinsip: Sekali sebuah frame citra dialokasikan ke dalam subset validasi (`val`), frame tersebut **wajib tetap berada di subset validasi selamanya** pada seluruh dataset masa depan. *Rasional:* Mencegah kontaminasi data validasi ke data latih yang dapat menyebabkan nilai metrik mAP menjadi overoptimis dan tidak valid.

1. Invarian 2: Eksekusi Tunggu `set_image()` per Frame pada SAM3

Prinsip: Metode `Sam3Processor.set_image()` hanya dipanggil **satu kali per citra**. Pengujian multi-prompt dijalankan melalui pemanggilan berulang `set_text_prompt()` pada state fitur yang sama. *Rasional:* Menghindari komputasi ulang Vision Transformer yang memboroskan siklus GPU dan memperlambat auto-labeling hingga 5 kali lipat.

1. Invarian 3: Pemicuan Tripwire Berbasis Tepi Atas (`y_1`)

Prinsip: Garis hitung tripwire hanya merespons lintasan koordinat puncak objek (`y_1`). *Rasional:* Mencegah distorsi hitungan akibat perubahan panjang kantong karung saat tertekan di konveyor.

14.2 Penanganan Kasus Batas (Edge Cases)

Panduan sistem saat menghadapi skenario operasional khusus:

• Kasus 1: Frame Tanpa Objek (Negative Sample Frame)

Jika sebuah citra frame tidak memuat karung sama sekali (konveyor kosong), proses ekspor dataset tetap menghasilkan file label `.txt` kosong (ukuran 0 byte). File label kosong ini sangat penting bagi model YOLO untuk mempelajari representasi latar belakang (*background learning*) dan menekan false positive.

• Kasus 2: Oklusi Parsial oleh Pekerja

Jika karung tertutup sebagian oleh badan operator namun tepi atas (`y_1`) tetap terlihat jelas, buat anotasi hanya pada batas visual karung yang tampak. Jangan menebak bentuk di balik tubuh pekerja.

• Kasus 3: Benturan Akses GPU Secara Bersamaan

Jika pengguna memicu pelatihan model saat proses auto-labeling batch sedang berjalan, antrean tugas backend akan menahan perintah pelatihan dengan status `pending` hingga auto-labeling selesai atau dibatalkan oleh operator.

• Kasus 4: URL RTSP Dimasukkan pada Form Live Count Web

Peramban web standar tidak mendukung pemutaran langsung protokol RTSP. Jika operator memasukkan URL berawalan `rtsp://`, antarmuka akan menampilkan pesan kesalahan informatif dan memandu operator untuk memasukkan endpoint WebRTC WHEP MediaMTX (`http://...:8889/.../whep`).

14.3 Matriks Solusi Pemecahan Masalah (Troubleshooting Matrix)

Daftar masalah umum dan langkah penanganan cepat:

Gejala Masalah	Indikasi Log / Error Code	Penyebab Akar	Tindakan Perbaikan
Kegagalan Auto-Labeling SAM3	<code>HTTP 401 Unauthorized</code> atau <code>HF ValidationError</code>	Token Hugging Face pada berkas <code>.env</code> belum diisi atau tidak memiliki izin akses ke repositori Meta SAM3.	Buka Hugging Face, buat User Access Token bertipe Read, setuju lisensi SAM3 di portal Meta, lalu perbarui variabel <code>HF_TOKEN</code> pada file <code>.env</code> .
CUDA Out of Memory (OOM)	<code>torch.cuda.OutOfMemoryError: CUDA out of memory</code>	Alokasi VRAM melampaui kapasitas fisik GPU saat training atau auto-labeling.	1. Turunkan parameter ukuran batch (<code>batch=8</code> atau <code>batch=4</code>). 2. Turunkan resolusi citra latih (<code>imgsz=640</code> atau <code>512</code>). 3. Pastikan tidak ada proses Python zombie yang mengunci VRAM dengan menjalankan <code>fuser -v /dev/nvidia*</code> .
Streaming WebRTC Gelap / Putus	<code>WHEP connection failed (ICE timeout)</code>	MediaMTX belum menerima feed RTSP dari kamera atau koneksi jaringan VPN terputus.	1. Uji koneksi ping ke IP kamera Dahua (<code>ping 192.168.192.96</code>). 2. Buka dashboard MediaMTX dan verifikasi bahwa stream <code>/cam</code> berstatus aktif. 3. Restart layanan MediaMTX.
Video Scrubbing Lambat / Macet	<code>HTTP 200 OK</code> (Bukan HTTP 206)	Nginx atau peramban tidak mendukung byte range request atau file video mengalami korupsi index moov atom.	1. Jalankan perintah <code>qt-faststart</code> atau <code>ffmpeg -i input.mp4 -c copy -movflags +faststart output.mp4</code> untuk memindahkan metadata moov atom ke awal berkas. 2. Pastikan header <code>Accept-Ranges: bytes</code> aktif pada Nginx.
Perbedaan Hitungan Signifikan pada Benchmark	Delta bertanda merah besar (<code>-50</code> atau <code>+70</code>)	Posisi garis tripwire bergeser atau resolusi video berubah pasca pemeliharaan kamera.	1. Buka halaman Live Count dan lakukan kalibrasi ulang garis <code>line_y</code> , <code>line_x_start</code> , dan <code>line_x_end</code> . 2. Periksa stempel waktu OCR pada Counting Bench untuk memastikan tidak ada batch rekaman yang tertukar.

14.4 Protokol Pemulihan Layanan Pasca Kegagalan Sistem

Jika server mengalami pemadaman listrik mendadak atau kegagalan perangkat keras:

1. Nyalakan server dan masuk ke terminal host.
2. Periksa integritas database SQLite:

```
sqlite3 data/app.db "PRAGMA integrity_check;"
```

Hasil normal: `ok`.

1. Bersihkan sisa kunci pekerjaan (*stale jobs*) yang tertinggal dalam status running:

```
sqlite3 data/app.db "UPDATE jobs SET status='failed', error='Server restart recovery' WHERE status='running';"
```

1. Jalankan ulang seluruh stack kontainer menggunakan `./start.sh`.
2. Verifikasi fungsionalitas sistem melalui endpoint `/api/health`.