

Panduan Lengkap Sistem: Arsitektur, Alur & Cara Menjalankan

Dokumen Tunggal Terintegrasi untuk Developer & AI Assistant | Retraining Studio & Live Counter Feedmill

1. Panduan Lengkap: Cara Menjalankan Sistem (Execution Guide)

Sistem memiliki 2 komponen operasional: **Retraining Studio** (untuk melatih & meningkatkan model) dan **Live Counter** (untuk deteksi & hitung karung di lapangan).

A. Menjalankan Retraining Studio dengan Docker (Direkomendasikan)

```
# 1. Pindah ke direktori project
cd /home/asus/feedmill/reTraining

# 2. Pastikan file .env sudah ada (masukkan HF_TOKEN)
cp .env.example .env

# 3. Jalankan container dengan GPU support & mount video archive
VIDEO_ARCHIVE_HOST=/path/ke/video/archive docker compose up -d --build

# 4. Buka Web Studio di Browser: http://localhost:8080 (API di :8000)
# 5. Cek status service & health GPU:
curl http://localhost:8000/api/health
```

B. Menjalankan Mode Development (Manual tanpa Docker)

```
# Backend (FastAPI + SAM3):
cd /home/asus/feedmill/reTraining
uv pip install -r requirements.txt
uv pip install -e sam3/
uv run uvicorn backend.main:app --reload --port 8000

# Frontend (React 19 + Vite 7) di terminal terpisah:
cd /home/asus/feedmill/reTraining/frontend
npm install
npm run dev # Buka http://localhost:5173 (Otomatis proxy /api ke :8000)
```

C. Menjalankan Live Counter Lapangan (menghitung-karung)

```
cd /home/asus/feedmill/menghitung-karung
# 1. Jalankan deteksi real-time & counter:
python3 predict.py

# 2. Jalankan dashboard monitoring operator (buka http://localhost:5000):
python3 counter_dashboard.py

# Atau kelola via systemd service:
sudo systemctl restart karung-counter.service
sudo systemctl restart karung-counter-dashboard.service
```

D. Langkah-Langkah Operasional Retraining di Web Studio

- **1. Projects Screen:** Buat project baru atau pilih project aktif. Unggah base model (.pt) dan tetapkan daftar kelas (misal: karung-pakan).
- **2. Video Archive:** Pilih rekaman video sesi muat truk (per siklus shift kerja). Tentukan rentang detik dan FPS extraction, lalu klik *Extract Frames*.
- **3. Auto-Labeling (SAM3):** Jalankan job auto-label. Model SAM3 akan melakukan segmentasi objek secara otomatis menggunakan prompt teks nama kelas.
- **4. Review Canvas:** Buka antarmuka review. Periksa setiap frame, geser/perbaiki titik polygon jika ada kesalahan, dan approve frame yang sudah valid.
- **5. Data Prep & Merge:** Terapkan filter kualitas dan konfirmasi merge ke **Master Dataset** (bersifat *immutable* dengan split validasi yang stabil).
- **6. Training & Benchmarking:** Jalankan YOLO fine-tuning. Sistem akan membandingkan metrik mAP model baru vs base model. Jika mAP lebih unggul, model best.pt siap dipakai.

2. Alur Kerja Aplikasi (Workflow)

Tahap	Komponen	Deskripsi Alur Sederhana
1. Live Detect & Record	menghitung-karung	Kamera CCTV streaming RTSP. Model mendeteksi karung & truk di conveyor serta merekam video per batch (YYYY-MM-DD/batchXXX.mp4).
2. Frame Extraction	Retraining Backend (FFmpeg)	Memotong rekaman video batch menjadi gambar frame individual sesuai interval waktu/FPS.
3. SAM3 Auto-Label	Retraining Backend (SAM3)	Model foundation SAM3 otomatis membuat bounding box & polygon label berdasarkan prompt teks kelas.
4. Review & Merge	React Web UI (:8080)	User memverifikasi anotasi di canvas review web, lalu mengunci data ke Master Dataset.
5. YOLO Retraining	Ultralytics YOLO11	Fine-tuning model YOLO dengan split validasi yang stabil (tidak pernah berubah) agar perbandingan akurat.
6. Evaluasi & Export	Benchmark Engine	Sistem membandingkan mAP model baru vs base model. Output model terbaik disimpan di best.pt.

3. Penjelasan Port & Konfigurasi Jaringan

Port	Service	Host / Mode	Fungsi & Protokol
8080	Web UI Retraining Studio	Docker / Nginx	Antarmuka visual web untuk review, anotasi, kurasi dataset, & monitor training.
8000	FastAPI Backend	Local / Docker	REST API backend untuk manajemen project, job worker (FFmpeg, SAM3, YOLO).
5173	Vite Dev Server	Local Dev	Port frontend React saat tahap coding / development lokal (proxy /api ke :8000).
5000	Counter Dashboard	Local App	Web dashboard live count karung conveyor (menghitung-karung).
8554 / 554	RTSP Live Stream	CCTV / MediaMTX	Input streaming video live kamera conveyor feedmill.

4. Peta Filesystem & Lokasi Folder

Komponen	Lokasi Path Direktori	Deskripsi & File Utama
Retraining Backend	reTraining/backend/	- main.py: Entry point FastAPI. - db.py: Schema SQLite (projects, batches, annotations). - jobs/: Worker eksekusi (extract, autolabel, merge, train).
Retraining Frontend	reTraining/frontend/	- React 19 + Vite 7 + Tailwind CSS. - src/pages/: Projects, Video Archive, Review Canvas, Dataset, Training.
Model SAM3	reTraining/sam3/	Vendor Meta SAM3 (dependency) untuk zero-shot auto-labeling.
Master Data & Weights	reTraining/data/	- app.db: Database SQLite metadata. - projects/<slug>/dataset/: Master dataset train/val. - projects/<slug>/models/: Checkpoint model best.pt.
Live Counter App	menghitung-karung/	- predict.py: Deteksi & algoritma counting karung conveyor. - counter_dashboard.py: Web UI dashboard live operator. - zones.json: ROI garis hitung & area conveyor. - jetson_counter.db: Log hitungan lokal.

5. Status: Before, Current, and Next

Fase	Kondisi Sistem	Dampak / Catatan Teknis
BEFORE	Model deteksi awal statis; jika akurasi drop, anotasi harus manual di CVAT/Roboflow tanpa pengujian terstandar.	Proses lambat, rawan drift dataset, dan tidak ada benchmark mAP komparatif yang terbukti.
CURRENT	Pipeline Retraining mandiri lokal (FastAPI + React + SAM3). Auto-label via teks prompt, kurasi data ketat, dan benchmark mAP otomatis.	Seluruh proses training berjalan mandiri di GPU workstation lokal tanpa ketergantungan cloud.
NEXT	Skrip otomatis build TensorRT (.engine) langsung setelah model baru lolos uji dan active learning untuk mining frame anomali.	Mempercepat siklus update model dan memaksimalkan FPS inference di hardware lokal.

6. Catatan Teknis & Troubleshooting Penting

- **Download Checkpoint SAM3 (~3.4 GB):** Terjadi otomatis pada job auto-label pertama kali. Disimpan dalam Docker volume / cache.
- **Package Manager:** Gunakan selalu uv (bukan pip/poetry). Eksekusi script dengan uv run.
- **Pinning Vite 7:** Frontend sengaja menggunakan Vite 7 (Vite 8 memiliki isu binding Rolldown pada lingkungan workstation ini).
- **Invarian Valid Split:** Frame yang sudah masuk ke set val akan tetap di val selamanya agar perbandingan mAP base vs new selalu valid.