

Tentu. Berdasarkan seluruh proses yang telah kita lalui bersama, berikut saya buat **dokumentasi implementasi Frigate dari awal hingga CCTV outlet baru dapat dimonitor dari server pusat**. Dokumen ini bisa dijadikan **SOP (Standard Operating Procedure)** untuk implementasi outlet berikutnya.

IMPLEMENTASI CCTV OUTLET BARU MENGGUNAKAN FRIGATE DAN DOCKER

Project

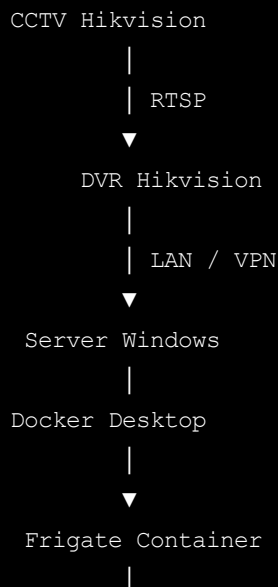
Integrasi CCTV Outlet ke Server Pusat Menggunakan Docker dan Frigate

1. Tujuan

Membangun sistem monitoring CCTV terpusat menggunakan Frigate sehingga seluruh kamera outlet dapat:

- Live View
- Recording
- AI Object Detection
- Event Review
- Penyimpanan Recording
- Monitoring dari Server Pusat

2. Arsitektur Sistem





3. Persiapan Server

Server menggunakan

- Windows 10/11
- Docker Desktop
- WSL2
- RAM minimal 8 GB
- CPU Intel/AMD

4. Install WSL

Buka PowerShell Administrator

```
wsl --install
```



Restart komputer.

Verifikasi

```
wsl --status
```



Hasil

```
Default Version: 2
```



Artinya WSL berhasil.

5. Install Docker Desktop

Install Docker Desktop.

Pastikan muncul

```
Docker Desktop is running
```



Tes

```
docker version
```



dan

```
docker ps
```



6. Membuat Folder Frigate

```
C:\frigate
```



isi

```
C:\frigate
|
├─ config
|
└─ storage
```



7. Download Image Frigate

```
docker pull ghcr.io/blakeblackshear/frigate:stable
```



8. Menjalankan Container

```
docker run -d ^
--name frigate ^
--restart unless-stopped ^
--shm-size=256m ^
-p 5000:5000 ^
-v C:\frigate\config:/config ^
-v C:\frigate\storage:/media/frigate ^
ghcr.io/blakeblackshear/frigate:stable
```



Parameter penting

```
--shm-size=256m
```



digunakan agar AI tidak kekurangan Shared Memory.

9. Login Pertama

Buka

```
http://localhost:5000
```



Frigate akan membuat akun default

```
admin
```



beserta password otomatis.

10. Membuat Config

File

```
C:\frigate\config\config.yml
```



Isi konfigurasi

YAML



```
mqtt:
  enabled: false

detectors:
  cpu1:
    type: cpu

cameras:
  outlet_test:
    ffmpeg:
      inputs:
        - path: rtsp://admin:PASSWORD@192.168.1.5:554/Streaming/Channels/101
          roles:
            - detect
            - record

detect:
  enabled: true
```

```
width: 1280
height: 720
fps: 5

objects:
  track:
    - person
    - car

filters:
  person:
    threshold: 0.5

  car:
    threshold: 0.5

record:
  enabled: true

snapshots:
  enabled: true

version: 0.17-0
```

11. Restart Container

```
docker restart frigate
```



12. Verifikasi Kamera

Masuk

```
http://localhost:5000
```



Jika Live View tampil

✓ RTSP berhasil.

13. Verifikasi AI

Masuk

System



atau

`http://localhost:5000/api/stats`



Pastikan

`detection_enabled : true`



dan

`detection_fps > 0`



Hasil implementasi

Detection Enabled : TRUE



Detection FPS : 9.1

Inference Speed : 26 ms

Artinya AI berjalan normal.

14. Review Event

Menu

Review



berhasil menampilkan

- Person
- Car

yang berarti

AI berhasil mendeteksi objek.

15. Recording

Recording otomatis tersimpan

C:\frigate\storage



struktur

recordings
clips
exports



16. Optimasi Hikvision

Lebih baik menggunakan dua stream.

Main Stream

101



digunakan

Record



Sub Stream

102



digunakan

Detect



Contoh

YAML



```
inputs:
  - path: rtsp://admin:password@IP/Streaming/Channels/102
    roles:
      - detect

  - path: rtsp://admin:password@IP/Streaming/Channels/101
    roles:
      - record
```

Ini mengurangi penggunaan CPU.

17. Monitoring

Dashboard Frigate menyediakan

- Live Camera
- Review
- Alerts
- Recording
- Storage
- CPU
- Detection Speed

Semua dapat dipantau dari browser.

18. Integrasi Server Pusat

Arsitektur implementasi outlet ke server pusat:



Setiap outlet cukup menambahkan konfigurasi kamera baru pada `config.yml`.

19. Penambahan Outlet Baru

Langkah-langkah:

- 1. Pastikan DVR dapat diakses dari server pusat.
- 2. Uji RTSP menggunakan VLC.
- 3. Tambahkan konfigurasi kamera pada `config.yml`.
- 4. Restart Frigate.
- 5. Verifikasi Live View.
- 6. Verifikasi AI Detection.
- 7. Verifikasi Recording.
- 8. Verifikasi Review Event.

Tidak perlu membuat container baru; cukup menambah kamera pada konfigurasi yang sama.

20. Hasil Implementasi

Seluruh proses berhasil dilakukan dengan hasil:

Komponen	Status
Docker Desktop	✓ Berhasil
WSL2	✓ Berhasil
Frigate	✓ Berhasil
RTSP Hikvision	✓ Terhubung
Live View	✓ Berhasil
Recording	✓ Berhasil
AI Detection	✓ Berhasil
Review Event	✓ Berhasil
Snapshot	✓ Berhasil
Storage	✓ Berhasil
Monitoring Browser	✓ Berhasil

21. Rekomendasi Pengembangan Selanjutnya

Untuk implementasi produksi pada banyak outlet, saya menyarankan:

1. **Menggunakan Docker Compose** agar konfigurasi lebih mudah dikelola dan dipulihkan.
2. **Memisahkan stream**: substream untuk deteksi AI, mainstream untuk rekaman berkualitas tinggi.
3. **Menggunakan Google Coral TPU** atau GPU jika jumlah kamera bertambah (misalnya >8 kamera) untuk meningkatkan performa AI.
4. **Mengintegrasikan MQTT** untuk notifikasi dan otomasi.
5. **Menghubungkan Frigate dengan Home Assistant** jika diperlukan untuk otomatisasi lanjutan.
6. **Menggunakan VPN** (misalnya WireGuard atau Tailscale) agar seluruh komunikasi antara outlet dan server pusat aman.
7. **Menerapkan backup berkala** untuk folder `config` dan `storage`, serta monitoring kapasitas penyimpanan.

Dokumentasi ini sudah mencakup seluruh alur implementasi dari **persiapan server**, **instalasi Docker dan Frigate**, **konfigurasi RTSP Hikvision**, **aktivasi AI Detection**, hingga **monitoring CCTV outlet dari server pusat**, sehingga dapat dijadikan acuan untuk implementasi outlet-outlet berikutnya.