



AI Engineer Challenge

PT Synapsis Sinergi Digital

PT.Qwerty ingin mengembangkan sistem **object detection, tracking** dan **People Counting** untuk memantau jumlah orang yang berada pada high risk area (berbentuk **polygon**) di dalam sebuah video feed.

A. Challenge 1 : Desain Database

1) Membangun Struktur Database (SQL/NoSQL)

Anda diminta untuk mendesain sebuah struktur database untuk menyimpan :

- **Data objek/orang** yang terdeteksi (waktu deteksi, area, dsb).
- **Data counting** (jumlah orang yang masuk dan keluar area polygon).
- **Data konfigurasi area** (opsional, jika Anda akan membuat area polygon bisa diatur dinamis oleh user, simpan pula informasi coordinates polygon, nama area, dsb).

2) Pertimbangan Desain

- Bagaimana Anda mendesain sistem untuk mengambil input video dan melakukan deteksi?
- Bagaimana relasi antara tabel deteksi dan tabel konfigurasi area polygon?

Catatan : Lampirkan diagram database dan diagram desain sistem pada *README* agar reviewer dapat memahami rancangan database Anda.

B. Challenge 2 : Pengumpulan Dataset

Anda bebas mencari atau menggunakan file video apapun sebagai sumber data. Anda

dapat menggunakan contoh video cctv Jogja berikut (nilai plus jika menggunakan *live streaming*) :

- [Malioboro_10_Kepatihan.stream](#)
- [Malioboro_30_Pasar_Beringharjo.stream](#)
- [NolKm_Utara.stream](#)

Catatan : Jelaskan di *README* dari mana video/dataset Anda diperoleh. Jika ada kendala (mis. sulitnya mendapatkan *video feed real-time*), Anda bisa menggunakan video statis atau dataset contoh dari *open source*.

C. Challenge 3 : Object Detection & Tracking

1) Deteksi Orang (Human Detection)

Kembangkan sistem untuk mendeteksi orang di dalam video. Anda diberikan kebebasan dalam menggunakan *pre-trained model* (mis. YOLO, SSD, Faster R-CNN, dsb.) dan *framework* (TensorFlow, PyTorch, OpenCV, dsb).

2) Tracking & Counting

Setelah orang terdeteksi, lakukan *tracking* sehingga Anda tahu berapa kali seseorang memasuki area polygon. Metode *tracking* bisa menggunakan *centroid tracking*, *Kalman filter*, *ByteTrack* dsb.

Sistem harus bisa menghitung jumlah orang yang masuk/keluar **detection area** berbentuk polygon (nilai plus jika polygon dapat diubah secara dinamis oleh *user*).

3) Polygon Setting (Fixed vs Dynamic)

Minimal, Anda harus memiliki 1 polygon untuk detection area

Polygon dapat berbentuk *fixed* (hard-coded).

Nilai plus jika area polygon dapat diatur ulang oleh *user* (mis. melalui *API* atau *UI* dashboard).



Catatan: Pastikan Anda menjelaskan pendekatan Anda dalam mendeteksi dan *tracking* objek (mis. alur proses: video frame → deteksi bounding box → cek apakah bounding box masuk/keluar polygon → update counter) pada diagram desain sistem.

D. Challenge 5 : Integrasi Sistem (API & Dashboard)

1) API (Application Programming Interface)

Anda harus menampilkan hasil deteksi, counting, dan prediksi melalui **response API**.

Endpoint minimal :

- **GET /api/stats/** → menampilkan history jumlah orang yang masuk/keluar area polygon (menambahkan filter seperti range waktu dan pagination adalah opsional dan akan menjadi nilai plus)
- **GET /api/stats/live** → menampilkan jumlah orang yang baru saja masuk/keluar area polygon (terbaru).
- (Opsional) **POST /api/config/area** → untuk mengatur koordinat polygon (jika ingin area deteksi dinamis, nilai plus).

2) Dashboard (opsional & nilai plus)

Jika ingin menampilkan **statistik live** (jumlah orang yang terdeteksi, graph prediksi) dalam bentuk web dashboard, itu akan menjadi nilai plus.

Menampilkan **live video** bukanlah hal wajib, tapi akan jadi nilai plus tersendiri jika dilakukan.

E. Challenge 6 : Deployment

1) Containerization (opsional & nilai plus)

Anda dapat menggunakan Docker sebagai opsi untuk mempermudah reviewer menjalan sistem.

Pastikan *Dockerfile* dan *docker-compose.yml* (jika dibutuhkan) disertakan.



2) Penjelasan Local Setup

Jika Anda tidak menggunakan docker, jelaskan cara menjalankan aplikasi secara lokal, langkah instalasi, dependency, dsb. dalam file **README.md**.

Catatan: Dokumentasi singkat tentang bagaimana menjalankan sistem akan memudahkan tim reviewer melakukan uji coba.

F. Ketentuan

1) Submit Challenge Test ke Github

- Silahkan kirimkan hasil penggeraan Anda ke github dan **invite** mufidmove@gmail.com sebagai collaborator & reviewer challenge Anda.
- Konfirmasikan ke tim HR yang menghubungi anda saat anda sudah menyelesaikan challenge ini.

2) Checklist Fitur yang Anda kerjakan

- Buat daftar *checklist* di dalam *README* untuk menandai fitur apa saja yang telah (Done) atau belum (X) Anda selesaikan, beserta kendala yang dihadapi.

Contoh Format :

1. **Desain Database** (Done)
Keterangan kendala (if any)
2. **Pengumpulan Dataset** (Done)
Keterangan kendala (if any)
3. **Object Detection & Tracking** (X)
Keterangan kendala (if any)
4. **Counting & Polygon Area** (Done)
Keterangan kendala (if any)
5. **Prediksi (Forecasting)** (Done)
Keterangan kendala (if any)
6. **Integrasi API** (API/Front End) (Done)
Keterangan problem (if any)
7. **Deployment** (X)



Keterangan kendala (if any)

3) Nilai Tambahan/Plus

- Penyelesaian sebelum batas waktu dengan kelengkapan requirement akan mendapatkan perhatian khusus dari tim Synapsis.
- Fitur-fitur opsional (seperti dashboard web, live video, polygon dinamis) juga menambah nilai plus.

4) Penjelasan Menjalankan Aplikasi

Jika Anda tidak melakukan containerization menggunakan docker, jelaskan cara menjalankan aplikasi di *local environment* (framework apa, requirements, dsb.) dalam **README.md**.

5) Deadline Challenge Test

- Batas waktu pengumpulan yaitu **4 hari** setelah Anda mendapatkan email Challenge Test dari tim Synapsis.
- Apabila dapat menyelesaikan Challenge Test sebelum batas waktu dengan kelengkapan requirementnya akan menjadi pertimbangan dan nilai plus oleh tim Synapsis.

Catatan: Gunakan file **README.md** secara komprehensif, termasuk instruksi instalasi, cara menjalankan, contoh API endpoint, dsb.

G. Penutup

- Silakan lakukan yang terbaik dalam penggerjaan challenge ini.
- Fokuslah pada *fungsionalitas utama* (object detection, tracking, dan counting).
- Fitur tambahan (dashboard web, live video, dan docker deployment) tidaklah wajib **pastikan dikerjakan di akhir** agar waktu penggerjaan dapat dimaksimalkan, dan akan menjadi nilai plus.

Jika ada pertanyaan atau kendala terkait challenge, silakan berkoordinasi dengan tim HR atau perwakilan teknis yang menghubungi Anda.

- Do Your Best -