Berikut adalah penjelasan spesifikasi dan teknologi yang digunakan pada aplikasi AI ini, yang bisa kamu gunakan untuk presentasi:

---

## Spesifikasi & Teknologi Aplikasi "Pill Inspector"

### 1. \*\*Deskripsi Singkat\*\*

> "Aplikasi ini menggunakan model YOLOv8 untuk mendeteksi pil dari gambar atau webcam. Dengan antarmuka Kivy, pengguna bisa dengan mudah memilih gambar atau menggunakan webcam untuk inspeksi obat. Semua proses deteksi berjalan otomatis dan hasilnya bisa langsung dilihat serta disimpan."

### 2. \*\*Fitur Utama\*\*

- \*\*Deteksi objek pil/obat\*\* pada gambar maupun video/webcam.

- \*\*Antarmuka pengguna (GUI)\*\* berbasis Kivy yang mudah digunakan.

- \*\*Pengambilan gambar langsung dari webcam\*\* dan deteksi real-time.

- \*\*Menampilkan hasil deteksi\*\* (bounding box, label, jumlah objek).

- \*\*Menyimpan hasil deteksi\*\* ke folder hasil.

### 3. \*\*Teknologi yang Digunakan\*\*

- \*\*Python\*\*: Bahasa pemrograman utama.

- \*\*YOLOv8 (Ultralytics)\*\*: Model deep learning untuk deteksi objek.

- \*\*OpenCV\*\*: Untuk pengolahan gambar dan akses webcam.

- \*\*Kivy\*\*: Framework untuk membuat aplikasi GUI (desktop).

- \*\*Flask\*\*: Untuk backend web (jika menggunakan app.py).

- \*\*Numpy\*\*: Untuk manipulasi data numerik.

- \*\*Torch (PyTorch)\*\*: Library deep learning yang digunakan oleh YOLO.

### 4. \*\*Alur Kerja Aplikasi\*\*

1. \*\*Training Model\*\*: Dataset gambar pil dilabeli dan dilatih menggunakan YOLOv8.

2. \*\*Deteksi\*\*: Pengguna memilih gambar atau mengaktifkan webcam, lalu aplikasi melakukan deteksi objek menggunakan model yang sudah dilatih.

3. \*\*Output\*\*: Hasil deteksi ditampilkan di GUI dan bisa disimpan.

### 5. \*\*Struktur Folder Penting\*\*

- datasets: Berisi dataset gambar dan label untuk training.

- models: Berisi model hasil training.

- results: Menyimpan hasil deteksi.

- kivy\_app.py: File utama aplikasi GUI.

- app.py: Backend web (opsional).

### 6. \*\*Kebutuhan Sistem\*\*

- Python 3.9+ (disarankan 3.9 untuk kompatibilitas YOLOv8 lama)

- Library: ultralytics, opencv-python, kivy, torch, numpy, flask, werkzeug

---