

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Tahapan - Tahapan Penelitian

#### 1. Studi Literatur

Metode pengumpulan data akan dilakukan melalui studi literatur, yang mencakup pencarian dan analisis sumber-sumber yang relevan dengan penelitian. Ini akan mencakup kajian tentang *machine learning*, CNN, YOLO, *mean Average Precision*, dan topik terkait lainnya.

#### 2. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan utama pengguna sistem deteksi jatuh berbasis YOLOv8. Ini akan membantu dalam merancang sistem yang sesuai dengan kebutuhan pengguna dan memastikan kesesuaian dengan lingkungan penerapannya. Kebutuhan perangkat keras dan lunak juga akan dianalisis untuk memastikan bahwa infrastruktur yang dibutuhkan tersedia dan sesuai dengan spesifikasi yang dibutuhkan.

#### 3. Pengumpulan Data

Penelitian ini akan menggunakan dataset orang jatuh dari situs Roboflow.com dengan nama dataset "person falling detection sdp Image Dataset" [11], yang berfokus pada orang jatuh. Dataset ini akan digunakan untuk melatih model YOLOv8.

#### 4. Pelatihan Model

Pelatihan model bertujuan untuk memperdalam pemahaman model terhadap orang jatuh. Model YOLOv8 akan dilatih sebanyak empat kali dengan konfigurasi yang berbeda:

- Model pertama akan dilatih dengan 300 *epoch* dan patience sebesar 50.
- Model kedua akan dilatih dengan 300 *epoch*.
- Model ketiga akan dilatih dengan 300 *epoch* menggunakan model yang telah dilatih sebelumnya dengan dataset orang.
- Model keempat akan dilatih dengan 500 *epoch*.

Dengan variasi konfigurasi tersebut, diharapkan model dapat mempelajari pola-pola yang kompleks dalam deteksi orang jatuh.

## 5. Implementasi

Penelitian ini akan menggunakan pendekatan ekperimental untuk menguji akurasi deteksi orang jatuh dengan menggunakan YOLOv8. Sistem akan dirancang dengan menggunakan bahasa pemograman Python. Sampel data akan diambil dari data set orang jatuh yang tersedia di internet dan juga akan melibatkan partisipan yang telah mengalami kejadian jatuh. Sistem akan menggunakan kamera dalam pengimplementasian di dunia nyata dan ditempatkan di rumah.

Implementasi dimulai dengan persiapan infrastruktur, termasuk perangkat keras dan lunak yang diperlukan, seperti laptop dengan kamera dan sistem dengan konfigurasi YOLOv8 yang benar. Kemudian, dilakukan instalasi ultralytics dan *library* lainnya yang diperlukan dan mengunduh model YOLOv8.pt serta dataset jatuh orang. Kemudian, kita siapkan kode Python untuk pengujian dan pelatihan, sesuaikan perintah, dan ubah jalur folder. Lalu model akan dilatih dengan konfigurasi yang telah ditentukan. Setelah proses pelatihan selesai, akan dilakukan pengujian dan evaluasi terhadap model dan sistem yang menggunakan perubahan tinggi badan.

## 6. Pengujian dan Evaluasi

Sistem akan diuji secara menyeluruh menggunakan video untuk mendeteksi kejadian jatuh. Evaluasi akan dilakukan untuk mengevaluasi kinerja sistem secara keseluruhan, termasuk akurasi deteksi dan respons notifikasi. Pengujian akan dilakukan dengan pengetesan sistem dengan model orang jatuh dan deteksi perubahan tinggi badan dengan video-video orang jatuh dan tidak jatuh. Untuk evaluasi akurasi antar model akan dilakukan dengan membandingkan mAP (*mean Average Precision*) dengan menggunakan ambang IoU (*Intersection over Union*) sebesar 0.5.

## 7. Dokumentasi

Sistem akan didokumentasikan melalui kode program yang dibuat serta foto-foto hasil deteksi yang dilakukan. Dokumentasi ini akan mencakup proses pengembangan sistem serta hasil-hasil yang diperoleh dari penelitian ini. Ini akan membantu dalam memudahkan pemahaman dan replikasi hasil penelitian oleh pihak lain.

### 3.2 Alur Kerja

Alur kerja untuk mendeteksi orang jatuh dengan YOLOv8 dapat dilihat pada Gambar 3.1.



Gambar 3.1. Flowchart Alur Kerja Penelitian

### 3.3 Perancangan dan Pengembangan Sistem

Perancangan dan pengembangan sistem untuk penelitian "Implementasi Algoritma YOLOv8 untuk Deteksi Orang Jatuh dalam Aktivitas Sehari-hari", terdapat beberapa bagian, yaitu instalasi YOLOv8 dan pengunduhan data set, persiapan dan proses pelatihan model, evaluasi model, dan perancangan *flowchart* sistem.

#### 3.3.1 Instalasi YOLOv8 dan Pengunduhan Dataset

Tahap ini dimulai dengan instalasi "ultralytics" untuk menggunakan YOLOv8, yang tersedia di situs resmi di [ultralytics.com](https://ultralytics.com) yang akan mengarahkan pengguna ke halaman GitHub ultralytics yang berisi YOLOv8. Setelah mengarahkan instalasi ultralytics, perintah berikut dapat dijalankan:

```
1 pip install ultralytics
```

Kode 3.1: Instalasi ultralytics

Selanjutnya, akan diunduh model dasar yang akan digunakan, yaitu "yolov8.pt" dan "yolov8-pose.pt" untuk deteksi dengan tinggi badan. Kemudian, data set akan diunduh dari Roboflow.com dengan nama data set "person falling detection sdp Image Dataset" [11], yang terdiri dari 2283 gambar. Data set akan dibagi menjadi 80% untuk data pelatihan, 10% untuk validasi, dan 10% untuk pengujian.

#### 3.3.2 Persiapan dan Proses Pelatihan Model

Tahap ini dimulai dengan perubahan konfigurasi pada *file* "data.yaml". Hal ini dilakukan agar proses pelatihan akan menggunakan struktur *folder* yang benar, yaitu pada parameter *train*, *val*, dan *test*. Kemudian untuk melakukan proses pelatihan model, dapat menggunakan perintah berikut.

```
1 yolo detect train data=data.yaml model=yolov8.pt epochs=100 imgsz  
=640 patience=50 device=cpu
```

Kode 3.2: Contoh Perintah Pelatihan Model

Pada parameter-parameter yang digunakan dapat diubah lagi sesuai kebutuhan. Untuk penjelasan parameter yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1. Penjelasan Parameter Training

Parameter	Deskripsi
data	<i>Path</i> ke file konfigurasi dataset (misalnya, data.yaml). File ini berisi parameter-parameter spesifik dataset, termasuk path ke data pelatihan dan validasi, nama kelas, dan jumlah kelas.
model	Menentukan berkas model untuk pelatihan. Mengatur sebuah jalur menuju model terlatih .pt atau berkas konfigurasi .yaml. Penting untuk menentukan struktur model atau inisialisasi bobot.
patience	Jumlah epoch yang harus ditunggu tanpa peningkatan dalam metrik validasi sebelum berhenti dini dalam pelatihan. Membantu mencegah overfitting dengan menghentikan pelatihan ketika kinerja datar.
imgsz	Ukuran gambar target untuk pelatihan. Semua gambar diubah ukurannya menjadi dimensi ini sebelum dimasukkan ke dalam model. Mempengaruhi akurasi model dan kompleksitas komputasional.
epochs	Jumlah total epoch pelatihan. Setiap epoch mewakili satu kali proses lengkap melalui seluruh dataset. Penyesuaian nilai ini dapat memengaruhi durasi pelatihan dan kinerja model.
device	Menentukan perangkat komputasi untuk pelatihan: satu GPU (device=0), beberapa GPU (device=0,1), CPU (device=cpu), atau MPS untuk Apple silicon (device=mps).

Proses pelatihan dilakukan dengan 3 macam variasi. Masing - masing memiliki variasi pada jumlah *epoch*, *patience*, dan model. Perintah yang digunakan dapat dilihat pada Tabel 3.2.

Tabel 3.2. Parameter Training Model

Variasi	Perintah
epoch 300 dan patience 50	<pre> 1 yolo detect train data=data.yaml 2 model=yolov8l.pt epochs=300 3 imgsz=640 patience=50 device=cpu </pre> <p>Kode 3.3: epoch 300 dan patience 50</p>
epoch 300 dan patience 0	<pre> 1 yolo detect train data=data.yaml 2 model=yolov8l.pt epochs=300 3 imgsz=640 patience=0 device=cpu </pre> <p>Kode 3.4: epoch 300 dan patience 0</p>
epoch 500 dan patience 0	<pre> 1 yolo detect train data=data.yaml 2 model=yolov8l.pt epochs=500 3 imgsz=640 patience=0 device=cpu </pre> <p>Kode 3.5: epoch 500 dan patience 0</p>
epoch 300 dan patience 0 dengan model orang	<pre> 1 yolo detect train data=data.yaml 2 model=modeld.pt epochs=300 3 imgsz=640 patience=0 device=cpu </pre> <p>Kode 3.6: epoch 300 dan patience 0 dengan model orang</p>

### 3.3.3 Evaluasi Model

Evaluasi dapat dilakukan dengan memperhatikan *Precision*, *Recall*, dan mAP (mean Average Precision) yang dihasilkan dari proses pelatihan menggunakan

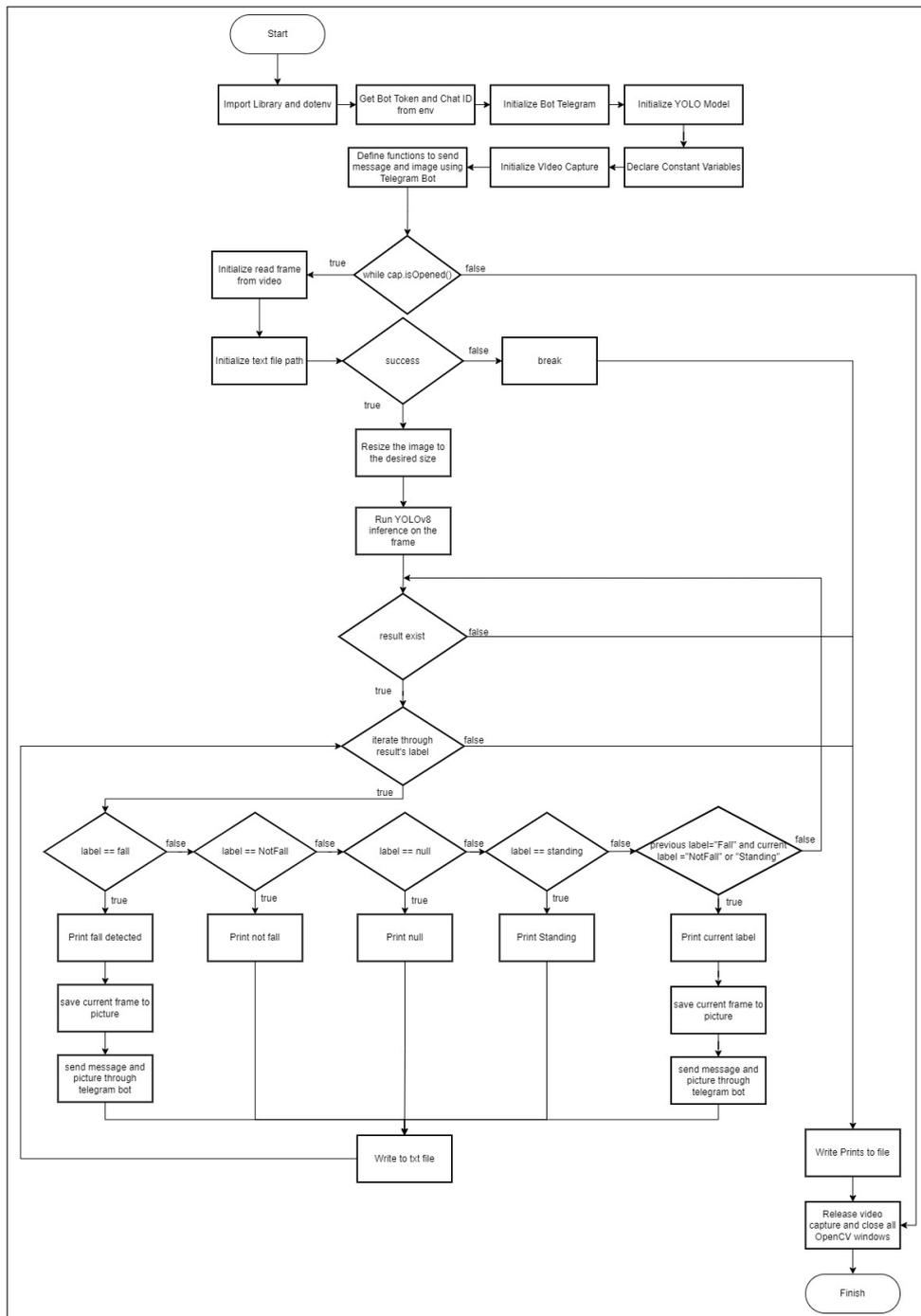
YOLOv8.

### 3.3.4 Perancangan Flowchart Sistem

Pada penelitian ini *flowchart* akan terbagi menjadi 2, yaitu *flowchart* deteksi orang jatuh menggunakan model orang jatuh, dan *flowchart* deteksi orang jatuh dengan mengukur tinggi dari orang yang jatuh.

*Flowchart* deteksi orang jatuh menggunakan model orang jatuh dapat dilihat pada Gambar 3.2. Proses dimulai dengan impor *library* dan *dotenv*. *Dotenv* digunakan untuk menyimpan data rahasia, seperti token bot Telegram dan ID chat. Kemudian, token bot dan chat ID diambil. Dilanjutkan dengan inisialisasi bot Telegram dan model YOLO. Variabel-variabel konstan yang diperlukan dideklarasikan, dan perekam video diinisialisasi. Selanjutnya, fungsi untuk mengirim pesan dan gambar via Telegram dibuat. Saat perekam video mulai, proses pembacaan *frame* dari video dan struktur file dilakukan. Jika berhasil, gambar diubah ukurannya ke ukuran yang diinginkan, dan proses *inference* oleh YOLOv8 dimulai. Jika terdapat hasil deteksi, label-label yang ditemukan akan diiterasi. Jika ditemukan label "Fall", informasi tentang kejadian orang jatuh akan dicatat dalam file txt dan dikirimkan melalui bot Telegram. Untuk label "NotFall", "null", dan "standing", informasi sesuai label juga akan dicatat dalam file txt. Jika ada perubahan dari label "Fall" menjadi "NotFall" atau "standing", sistem akan memberikan notifikasi melalui bot Telegram. Jika proses inisialisasi gagal atau tidak ada hasil deteksi, perekam video dan OpenCV (yang menampilkan video) akan dihentikan, dan sistem akan berhenti.

U M N  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

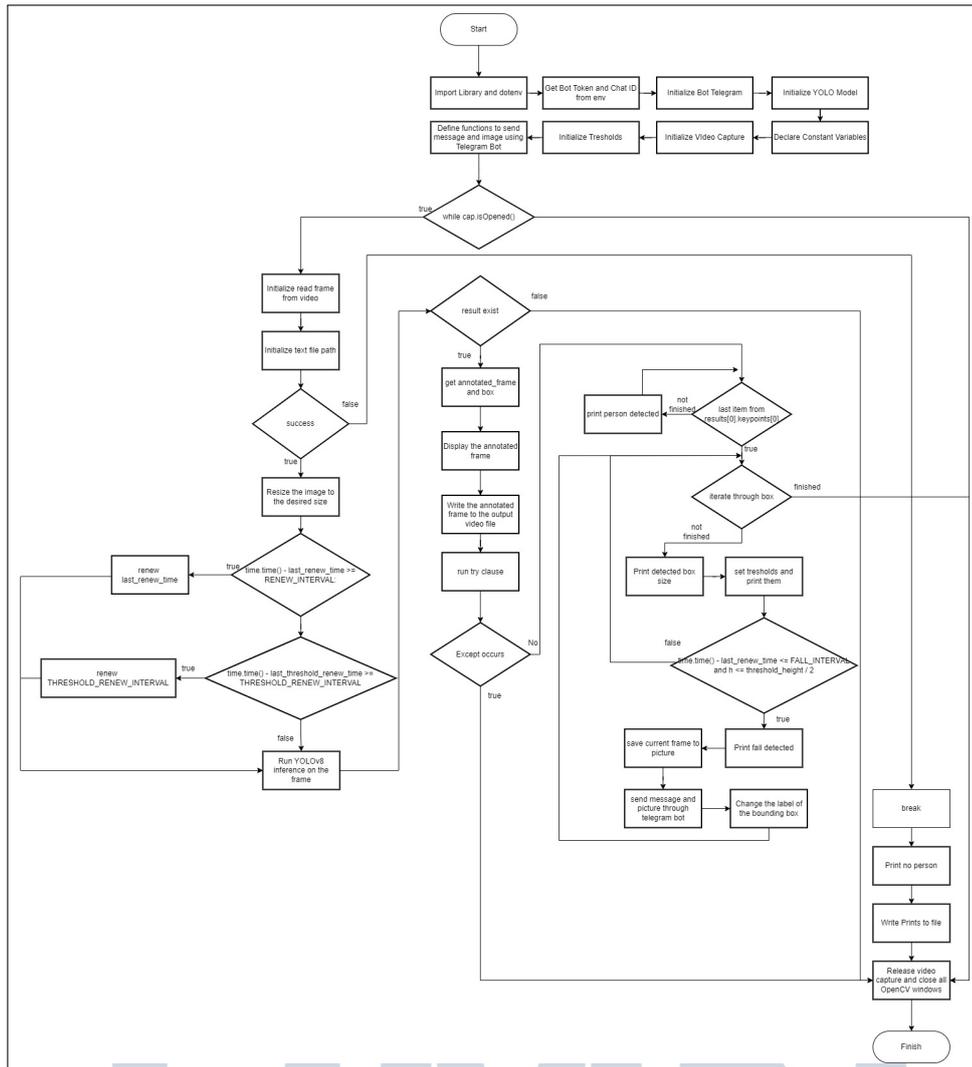


Gambar 3.2. Flowchart Deteksi Orang Jatuh Dengan Model Orang Jatuh

*Flowchart* deteksi orang jatuh dengan mengukur tinggi dari orang yang jatuh dapat dilihat pada Gambar 3.3. Proses dimulai dengan impor *library* dan *dotenv* untuk menyimpan data rahasia seperti token bot Telegram dan ID chat. Setelah itu, dilanjutkan dengan inialisasi bot Telegram dan model YOLO, diikuti dengan

deklarasi variabel konstan yang diperlukan dan inisialisasi perekam video. Fungsi untuk mengirim pesan dan gambar via Telegram dibuat, lalu saat perekam video mulai, dilakukan pembacaan *frame* dari video dan struktur file. Jika berhasil, gambar diubah ukurannya ke ukuran yang diinginkan, *renewal\_time*, dan *treshold* diset sesuai waktu interval yang telah ditentukan. Selanjutnya, dilakukan *inference* menggunakan YOLOv8. Saat mendapatkan hasil, sistem akan mendapatkan *annotated frame* dan *box* yang dihasilkan, dan dari situ, tinggi badan orang dihitung dari tinggi *box*. Jika tinggi *box* mencapai setengah atau lebih dari setengah tinggi *treshold* dan kurang dari waktu interval jatuh yang telah ditentukan, maka sistem akan mendeteksi orang jatuh, menuliskan informasi orang jatuh, dan mengirim pesan serta gambar melalui bot Telegram. Jika tidak ada hasil, *box* sudah habis, atau terjadi error, sistem akan berhenti.





Gambar 3.3. Flowchart Deteksi Orang Jatuh Dengan Mengukur Tinggi Dari Orang Yang Jatuh

