

## **BAB 3**

### **METODOLOGI PENELITIAN**

Dalam menyusun dan melaksanakan penelitian ini, obyek penelitian yang akan menjadi fokus adalah deteksi ketersediaan tempat parkir kosong pada intensitas cahaya rendah yang nantinya akan dilakukan komparasi model berdasarkan akurasi.

Untuk mencapai hal tersebut, maka metode dan tahapan yang digunakan selama penelitian ini berlangsung adalah sebagai berikut.

#### **3.1 Identifikasi Masalah**

Pada tahap ini, didapatkan permasalahan awal berupa limitasi deteksi obyek pada kondisi gelap yang sering muncul pada penelitian-penelitian sebelumnya yang menggunakan algoritma YOLO. Oleh karena itu, didapatkan tujuan dari dilakukan penelitian ini, yaitu untuk mengukur akurasi YOLOv7, YOLOv8, dan YOLOv9 dalam mendeteksi ketersediaan tempat parkir kosong pada intensitas cahaya rendah. Akurasi dari ketiga model ini akan dibandingkan untuk mengetahui model terbaik yang dapat digunakan untuk mendeteksi ketersediaan tempat parkir kosong pada intensitas cahaya rendah.

#### **3.2 Analisis Kebutuhan**

Menganalisis segala kebutuhan dan persiapan sebelum masuk ke dalam tahap perancangan sistem yang akan dibuat. Kebutuhan tersebut berupa sistem pendukung yang akan digunakan dalam proses *training*, evaluasi, maupun perancangan sistem. Pada tahap *training*, digunakan Google Colab Pro dengan mode *runtime* GPU V-100 yang menyediakan kecepatan *training* yang cepat serta RAM tinggi. Pada tahap evaluasi, digunakan GPU CUDA NVIDIA GeForce RTX 3050 4GB dengan RAM 16 GB. Pada tahap perancangan sistem, digunakan *framework* Flask yang dibuat di atas Anaconda *environment* untuk membuat situs web yang dibutuhkan.

### 3.3 Studi Literatur

Menggali lebih dalam mengenai metode dan teori-teori yang akan diterapkan pada penelitian ini. Gambaran yang jelas mengenai teori dan metode yang digunakan dapat memperjelas proses implementasi sistem dan mengurangi hambatan selama proses *training*, evaluasi, maupun perancangan.

### 3.4 Pengumpulan Data

Mencari dan menentukan *dataset* yang akan digunakan untuk proses *training* serta evaluasi dalam membentuk model yang diharapkan. Pada penelitian ini, digunakan 6 buah *dataset* sebagai berikut.

1. *Smart-Parking-System dataset*

*Dataset* ini didapatkan dari situs Roboflow dan terdiri dari 556 gambar tempat parkir pada pagi hari dengan perspektif *top view* yang telah diberi anotasi label "empty" dan "occupied" [15].

2. *Parking-Spot-Detection dataset*

*Dataset* ini didapatkan dari situs Roboflow dan terdiri dari 1614 gambar tempat parkir pada pagi hari dengan perspektif *side view* yang telah diberi anotasi label "Car" dan "Empty" [16].

3. *Dark-Parking-Lot-Top dataset*

*Dataset* ini dibuat oleh penulis dengan mengambil gambar-gambar dari internet maupun daerah sekitar tepatnya di Summarecon Mall Serpong (SMS) dan Tang City Mall yang dianotasi secara mandiri. *Dataset* ini terdiri dari 11 gambar tempat parkir yang diambil dari atas pada jarak dekat sampai dengan jauh (obyek deteksi berukuran 30-160 *pixel*) pada malam hari dengan intensitas cahaya rata-rata sebesar 56.03 *grayscale intensity value* [31].

4. *Dark-Parking-Lot-Side dataset*

*Dataset* ini dibuat oleh penulis dengan mengambil gambar-gambar dari internet maupun daerah sekitar tepatnya di Summarecon Mall Serpong (SMS), Tang City Mall, dan Flavor Bliss yang dianotasi secara mandiri. *Dataset* ini terdiri dari 10 gambar tempat parkir yang diambil dari samping pada malam hari dengan intensitas cahaya rata-rata sebesar 47.76 *grayscale intensity value* [32].

#### 5. *Parking-Lot-Bird-View dataset*

*Dataset* ini dibuat oleh penulis dengan mengambil *frame-frame* rekaman video yang didapat dari internet yang dianotasi secara mandiri. *Dataset* ini terdiri dari 30 *frame* video tempat parkir yang diambil dari atas pada jarak kamera yang jauh (obyek deteksi berukuran 20-40 *pixel*) pada pagi hari dengan intensitas cahaya rata-rata sebesar 105.13 *grayscale intensity value*. Karena kamera berjarak jauh, obyek deteksi terlihat kecil [17].

#### 6. *Dark-Parking-Lot-Bird-View dataset*

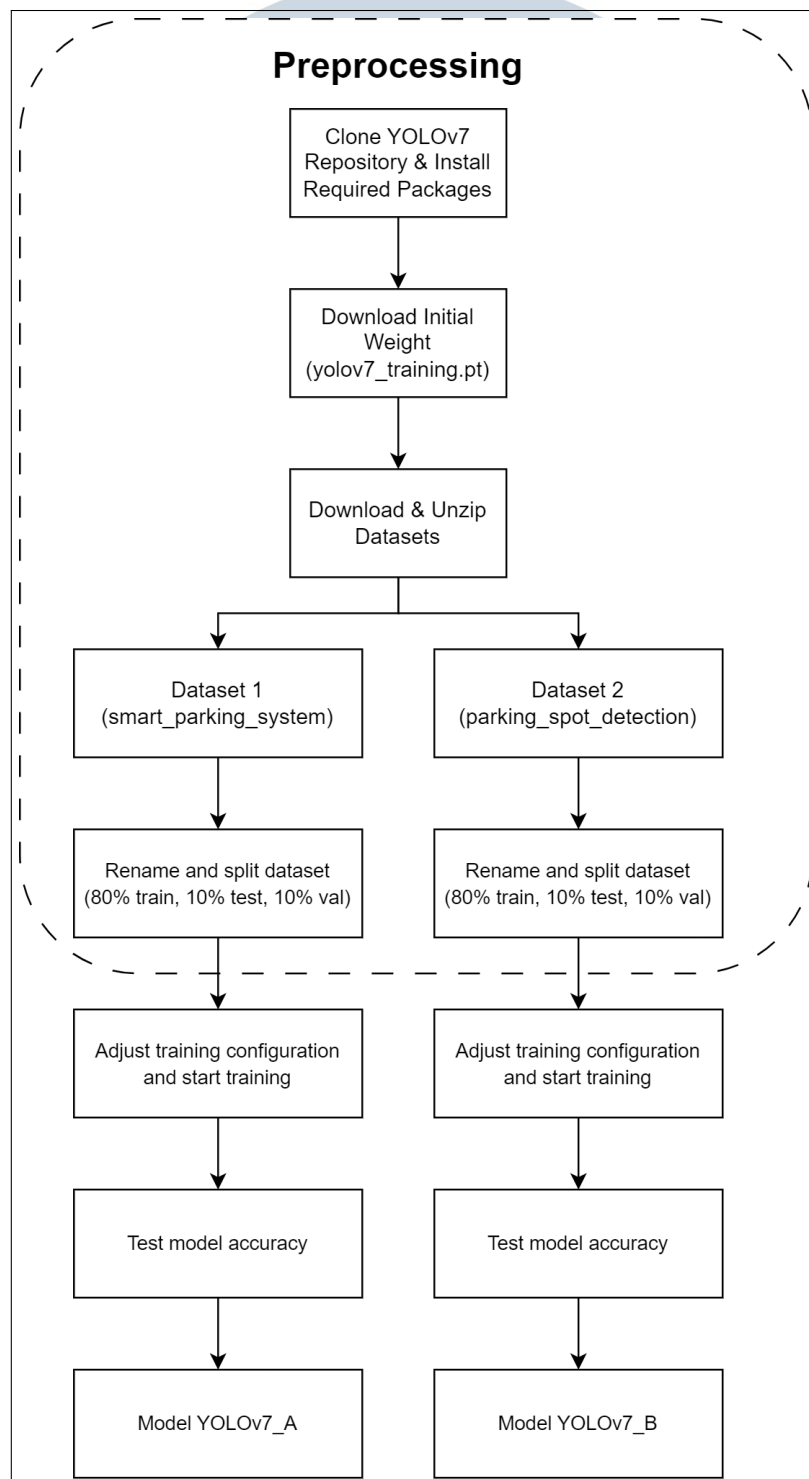
*Dataset* ini dibuat oleh penulis dengan mengambil *frame-frame* rekaman video yang didapat dari internet yang dianotasi secara mandiri. *Dataset* ini terdiri dari 30 *frame* video tempat parkir yang diambil dari atas pada jarak kamera yang jauh (obyek deteksi berukuran 30-60 *pixel*) pada malam hari dengan intensitas cahaya rata-rata sebesar 65.93 *grayscale intensity value*. Karena kamera berjarak jauh, obyek deteksi terlihat kecil [18].

Keenam *dataset* yang digunakan memiliki konten yang sama, yaitu lahan-lahan parkir yang telah dianotasi baik secara mandiri maupun *pre-annotated* dari Roboflow. Perbedaan dari keenam *dataset* berada pada jumlah gambar yang digunakan, waktu gambar diambil (intensitas cahaya normal atau rendah), perspektif gambar yang diberikan (atas atau samping), serta jarak kamera dengan obyek. *Dataset* pertama (*smart-parking-system*), ketiga (*dark-parking-lot-top*), kelima (*parking-lot-bird-view*), dan keenam (*dark-parking-lot-bird-view*) menggunakan perspektif foto tempat parkir yang diambil dari atas (*top view*), sedangkan *dataset* kedua (*parking-spot-detection*) dan keempat (*dark-parking-lot-side*) menggunakan perspektif foto tempat parkir yang diambil dari samping (*side view*). Dengan perbedaan ini, diharapkan dapat memberikan hasil model yang lebih variatif serta evaluasi yang lebih akurat.

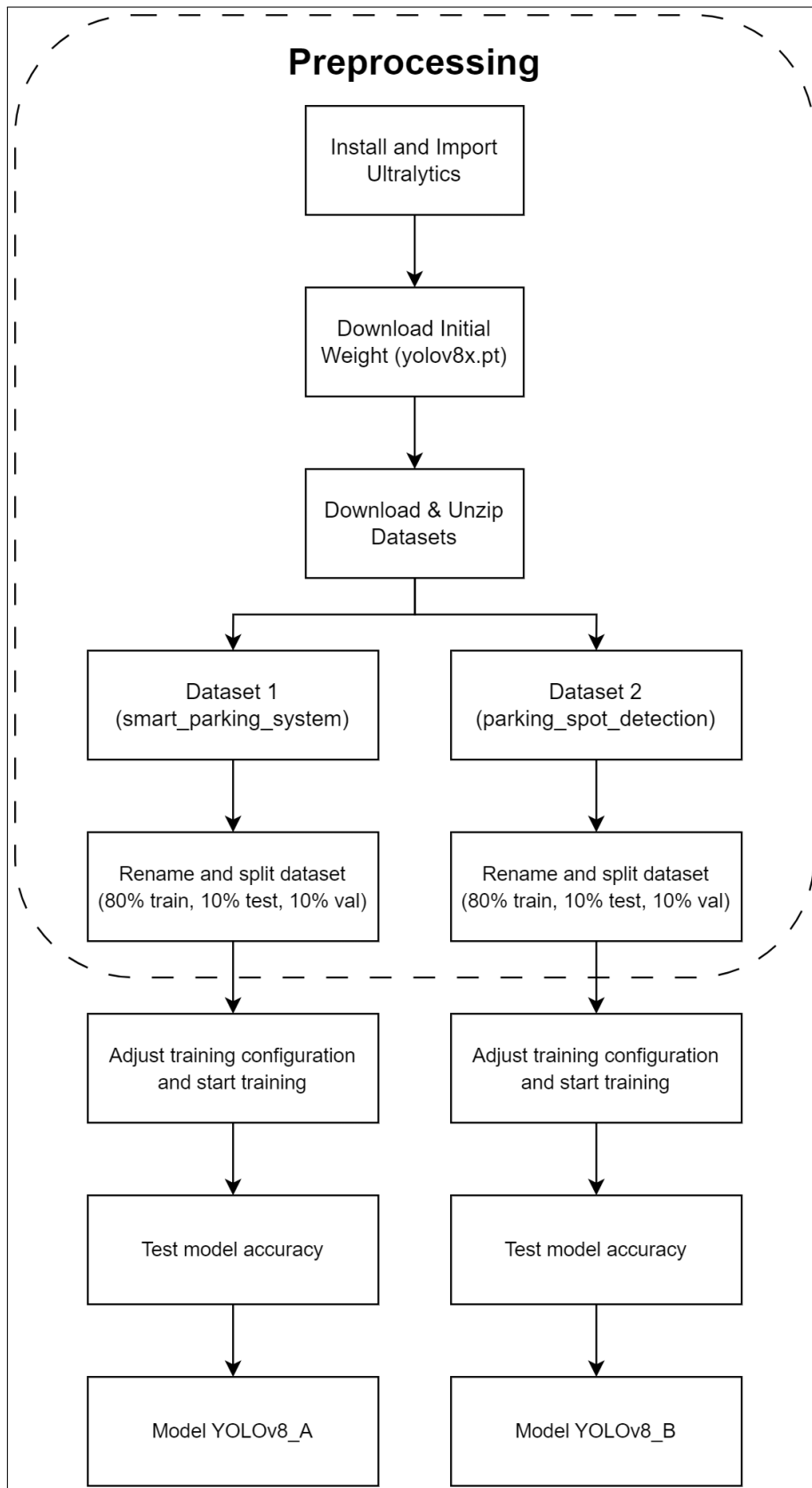
### 3.5 Training

Tahap ini ditujukan untuk menghasilkan model-model yang akan digunakan dalam proses perancangan. 2 *dataset*, yaitu "*smart-parking-system*" dan "*parking-spot-detection*" *dataset* yang digunakan dalam penelitian ini akan dilakukan *training* pada model YOLOv7, YOLOv8, dan YOLOv9. Secara total akan dihasilkan 6 jenis model dengan setiap jenis iterasi YOLO memiliki 2 jenis model berbeda. Tahapan *training* serta model yang dihasilkan untuk YOLOv7 dapat

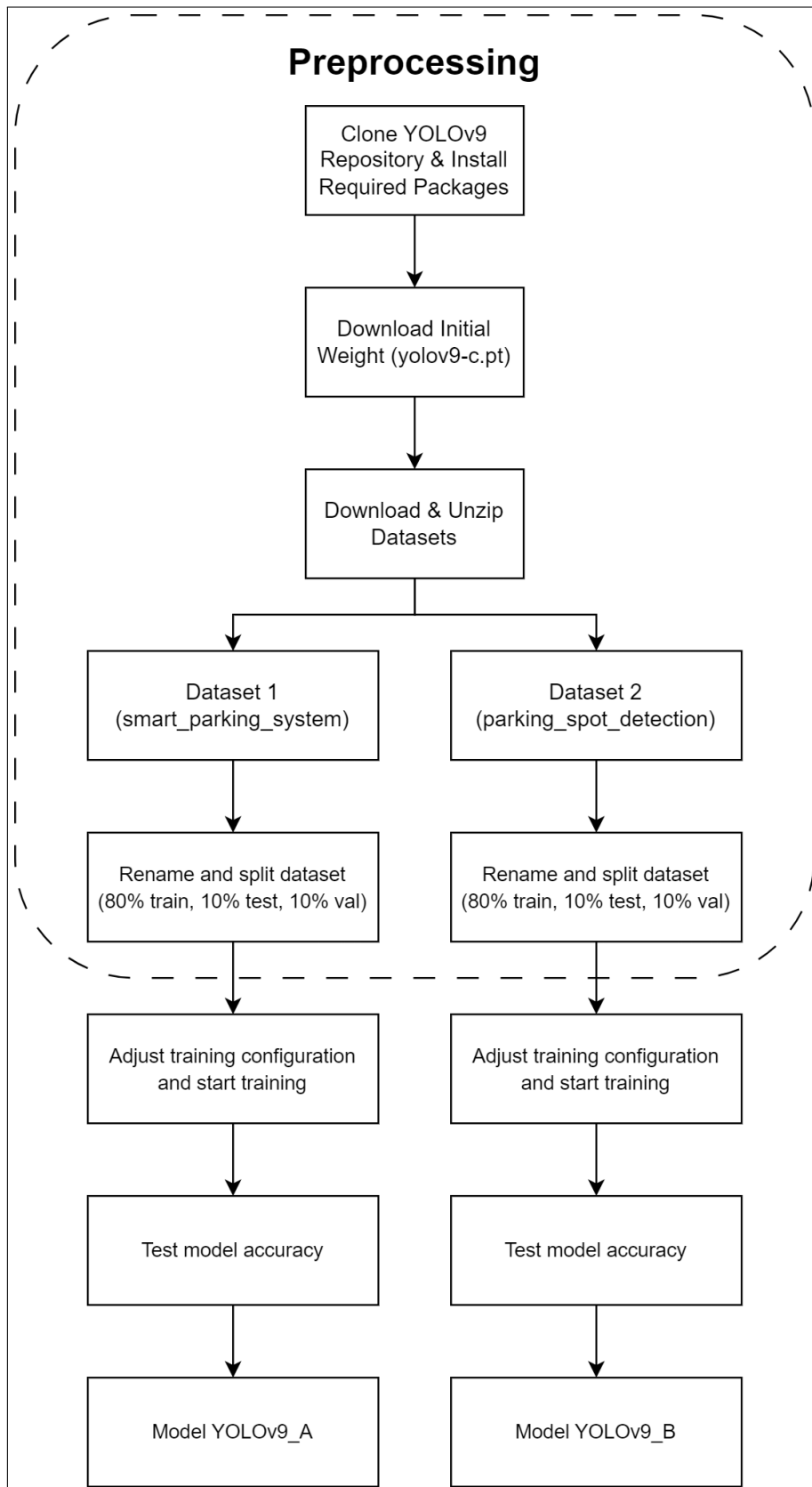
dilihat pada Gambar 3.1, untuk YOLOv8 dapat dilihat pada Gambar 3.2, dan untuk YOLOv9 dapat dilihat pada Gambar 3.3.



Gambar 3.1. YOLOv7 Model Diagram



Gambar 3.2. YOLOv8 Model Diagram



Gambar 3.3. YOLOv9 Model Diagram

Tahapan *training* terbagi menjadi 3 subtahap mulai dari persiapan sampai pada evaluasi seperti berikut.

#### 1. Persiapan *training*

Sebelum dilakukan *training* model, dilakukan *preprocessing* terlebih dahulu pada kedua *dataset* yang didapatkan. *Dataset* ini sudah memiliki gambar dengan ukuran 640 x 640 *pixel* sesuai *requirement* yang tertera pada dokumentasi YOLO. Pertama, *dataset* diunduh dari situs Roboflow dan dilakukan penamaan ulang semua *file* gambar untuk menghilangkan penggunaan titik dan menggantikannya dengan *under score*. Penamaan ulang ini dilakukan untuk mencegah kekeliruan ekstensi file yang digunakan. Setelah itu, dilakukan *data split* dengan rasio perbandingan 80% data untuk *training*, 10% data untuk *testing*, dan 10% data untuk *validation*. Kedua, dilakukan perubahan direktori *data source* serta jenis *class* untuk *YAML file* dan *file* konfigurasi *training* menyesuaikan dengan *dataset* yang digunakan. Terakhir, untuk setiap model YOLO, dilakukan pengunduhan *base model* dari masing-masing Github *repository* dengan YOLOv7 bernama "yolov7\_training.pt", YOLOv8 bernama "yolov8x.pt", dan YOLOv9 bernama "yolov9-c.pt".

#### 2. *Training* model

Tahap *training* ini dilakukan menggunakan Google Colab sehingga semua *file* yang digunakan disimpan pada Google Drive. Pertama, *dataset* yang telah dilakukan *split*, *file* konfigurasi, serta *base model* yang telah didapatkan diunggah ke Google Colab. Kedua, dilakukan *project setup* untuk setiap jenis YOLO agar proses *training* dapat dilakukan. Untuk YOLOv7 dan YOLOv9 dilakukan dengan menggunakan "git clone" dari Github *repository* masing-masing, sedangkan untuk YOLOv8 dilakukan dengan menjalankan "pip install ultralytics" untuk melakukan instalasi *package* Ultralytics yang merupakan basis dari YOLOv8. Ketiga, proses *training* dimulai dengan menjalankan perintah *train* pada terminal. Setiap model dilakukan *training* pada kedua *dataset* dengan iterasi yang sama, yaitu 300 *epoch* sehingga total *training* yang dilakukan adalah 6 kali yang menghasilkan 6 jenis model berbeda seperti pada Gambar 3.1, Gambar 3.2, dan Gambar 3.3.

Tabel 3.1. Argumen Perintah Training

Argumen	Penjelasan
- -workers	Jumlah <i>processor</i> yang digunakan untuk <i>training</i> ( <i>parallel processing</i> )
- -batch-size - -batch	Jumlah gambar yang akan diproses untuk dilakukan <i>update</i> pada model
- -device	Jenis GPU yang digunakan untuk proses <i>training</i>
- -data	Direktori data gambar yang dilakukan <i>training</i>
- -img - -imgsz	Ukuran gambar ( <i>pixel</i> ) yang digunakan selama proses <i>training</i>
- -cfg	Direktori <i>file</i> konfigurasi yang digunakan untuk <i>training</i>
- -patience	Jumlah iterasi yang dibutuhkan untuk memberhentikan <i>training</i> bila tidak terdapat perkembangan akurasi
- -min-items	Jumlah minimum gambar yang dibutuhkan untuk memulai <i>training</i>
- -close-mosaic	Jarak antar gambar dalam membentuk gambar mosaik yang digunakan sebagai data <i>training</i> baru. Ini dilakukan untuk membuat data yang lebih variatif.
- -weights - -model	Direktori <i>file</i> untuk <i>base model</i> yang digunakan untuk <i>training</i>
- -name	Nama <i>folder</i> tempat hasil model disimpan
- -hyp	Direktori <i>file</i> untuk <i>hyperparameter</i> yang digunakan untuk <i>training</i>
- -epoch - -epochs	Jumlah iterasi untuk proses <i>training</i>

Argumen perintah *training* yang dilakukan pada setiap model dapat dilihat pada Tabel 3.1. Untuk perintah-perintah yang dijalankan adalah seperti berikut.

(a) YOLOv7

Perintah *train* model YOLOv7\_A untuk YOLOv7 dapat dilihat pada



### Kode 3.1

```
1 !python train.py --batch-size 8 --device 0 --data data/
  parking_lot_dataset_1.yaml --img 640 640 --cfg cfg/
  training/yolov7-parking_lot.yaml --weights
  yolov7_training.pt --name yolov7-parking_lot-v1 --hyp
  data/hyp_scratch_custom.yaml --epochs 300
2
```

#### Kode 3.1: Train Model YOLOv7-A

Perintah *train* model YOLOv7\_B untuk YOLOv7 dapat dilihat pada Kode 3.2

```
1 !python train.py --batch-size 8 --device 0 --data data/
  parking_lot_dataset_2.yaml --img 640 640 --cfg cfg/
  training/yolov7-parking_lot.yaml --weights
  yolov7_training.pt --name yolov7-parking_lot-v2 --hyp
  data/hyp_scratch_custom.yaml --epochs 300
2
```

#### Kode 3.2: Train Model YOLOv7-B

### (b) YOLOv8

Perintah *train* model YOLOv8\_A untuk YOLOv8 dapat dilihat pada Kode 3.3

```
1 !yolo detect train data=parking_lot_dataset_1.yaml model=
  yolov8x.pt workers=8 batch=8 device=0 patience=0
  imgsz=640 name=yolov8-parking_lot-v1 epochs=300
2
```

#### Kode 3.3: Train Model YOLOv8-A

Perintah *train* model YOLOv8\_B untuk YOLOv8 dapat dilihat pada Kode 3.4

```
1 !yolo detect train data=parking_lot_dataset_2.yaml model=
  yolov8x.pt workers=8 batch=8 device=0 patience=0
  imgsz=640 name=yolov8-parking_lot-v2 epochs=300
2
```

#### Kode 3.4: Train Model YOLOv8-B

### (c) YOLOv9

Perintah *train* model YOLOv9\_A untuk YOLOv9 dapat dilihat pada Kode 3.5

```
1 !python train_dual.py --batch 8 --epochs 300 --img 640 --
  device 0 --min-items 0 --close-mosaic 15 --data data/
  parking_lot_dataset_1.yaml --weights weights/yolov9-c
```

```
. pt --cfg models/detect/yolov9-c.yaml --hyp hyp.
scratch-high.yaml
```

2

Kode 3.5: Train Model YOLOv9-A

Perintah *train* model YOLOv9\_B untuk YOLOv9 dapat dilihat pada Kode 3.6

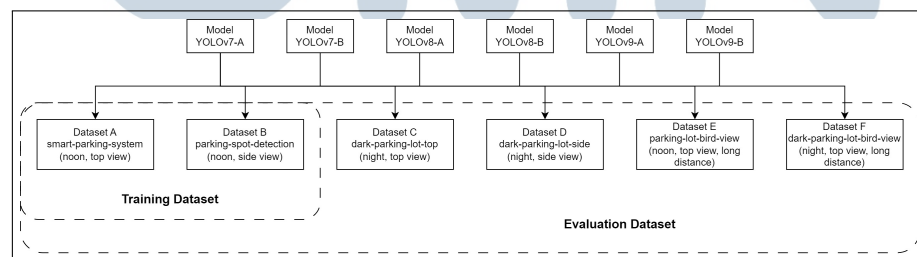
```
!python train_dual.py --batch 8 --epochs 300 --img 640 --
device 0 --min-items 0 --close-mosaic 15 --data data/
parking_lot_dataset_2.yaml --weights weights/yolov9-c
.pt --cfg models/detect/yolov9-c.yaml --hyp hyp.
scratch-high.yaml
```

2

Kode 3.6: Train Model YOLOv9-B

### 3. Evaluasi Model

Setelah model berhasil dilatih, maka pengujian dan evaluasi dilakukan terhadap setiap model dalam mendeteksi ketersediaan tempat parkir kosong untuk mengetahui akurasi dari masing-masing model. Pengujian ini dilakukan dengan menggunakan gambar-gambar tempat parkir yang diambil pada saat siang hari dan malam hari. Gambar yang digunakan terbagi menjadi 2, yaitu gambar yang diambil dari atas (*top view*) dan gambar yang diambil dari samping *side view*. Khusus untuk *top view*, terbagi menjadi 2 lagi, yaitu jarak kamera yang beragam dan jarak kamera yang jauh. Akurasi dihitung menggunakan *confusion matrix* dengan IoU 0.5 dan IoU 0.5 - 0.95 yang nantinya akan dihasilkan *precision* dan *recall* yang digunakan untuk menghitung mAP (*mean average percision*).



Gambar 3.4. Model Testing

Dapat dilihat pada Gambar 3.4, setiap model dilakukan testing terhadap 6 jenis *dataset* sehingga secara total, dilakukan 36 *testing*.

- (a) Dataset 1 (*smart-parking-system*)  
*Dataset* ini terdiri dari gambar tempat parkir yang diambil dari atas sekitar pagi dan siang hari. *Dataset* ini merupakan salah satu *dataset* yang digunakan untuk proses *training* [15].
- (b) Dataset 2 (*parking-spot-detection*)  
*Dataset* ini terdiri dari gambar tempat parkir yang diambil dari samping sekitar pagi dan siang hari. *Dataset* ini merupakan salah satu *dataset* yang digunakan untuk proses *training* [16].
- (c) Dataset 3 (*dark-parking-lot-top*)  
*Dataset* ini terdiri dari gambar tempat parkir yang diambil dari atas pada jarak dekat sampai dengan jauh (obyek deteksi berukuran 30-160 *pixel*) pada malam hari dengan intensitas cahaya rata-rata sebesar 56.03 *grayscale intensity value* [31].
- (d) Dataset 4 (*dark-parking-lot-side*)  
*Dataset* ini terdiri dari gambar tempat parkir yang diambil dari samping pada malam hari dengan intensitas cahaya rata-rata sebesar 47.76 *grayscale intensity value* [32].
- (e) Dataset 5 (*parking-lot-bird-view*)  
*Dataset* ini terdiri dari *frame-frame* video tempat parkir yang diambil dari atas pada jarak yang jauh (obyek deteksi berukuran 20-40 *pixel*) pada pagi hari dengan intensitas cahaya rata-rata sebesar 105.13 *grayscale intensity value* [17].
- (f) Dataset 6 (*dark-parking-lot-bird-view*)  
*Dataset* ini terdiri dari *frame-frame* video tempat parkir yang diambil dari atas pada jarak yang jauh (obyek deteksi berukuran 30-60 *pixel*) pada malam hari dengan intensitas cahaya rata-rata sebesar 65.93 *grayscale intensity value* [18].

Setiap *testing* yang dilakukan pada setiap jenis model memiliki konfigurasi yang sama dengan *confidence threshold* sebesar 0.01 dan IoU sebesar 0.5. Perintah-perintah yang dijalankan pada saat evaluasi model adalah sebagai berikut.

- (a) YOLOv7  
Perintah *test* model YOLOv7-A menggunakan *dataset* 1 dapat dilihat

pada Kode 3.7. Untuk perintah *test* dengan model dan *dataset* yang lain hanya perlu menyesuaikan argumen *data*, *weights*, dan *name*.

```
1 !python test.py --batch-size 2 --device 0 --data
data/parking_lot_dataset_1.yaml --weights YOLOv7-A.pt
--img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-
parking-lot-v1-test --task test
2
```

Kode 3.7: Testing Model YOLOv7-A

(b) YOLOv8

Perintah *test* model YOLOv8-A menggunakan *dataset* 1 dapat dilihat pada Kode 3.8. Untuk perintah *test* dengan model dan *dataset* yang lain hanya perlu menyesuaikan argumen *data* dan *model*.

```
1 !yolo detect val model=YOLOv8-A.pt batch=2 device
=cuda:0 data=parking_lot_dataset_1.yaml conf=0.01 iou
=0.5
2
```

Kode 3.8: Testing Model YOLOv8-A

(c) YOLOv9

Perintah *test* model YOLOv9-A menggunakan *dataset* 1 dapat dilihat pada Kode 3.9. Untuk perintah *test* dengan model dan *dataset* yang lain hanya perlu menyesuaikan argumen *data*, *weights*, dan *name*.

```
1 !python val_dual.py --data data/
parking_lot_dataset_1.yaml --img 640 --batch 2 --conf
0.01 --iou 0.5 --weights YOLOv9-A.pt --save-json --
name yolov9-parking-lot-v1-test
2
```

Kode 3.9: Testing Model YOLOv9-A

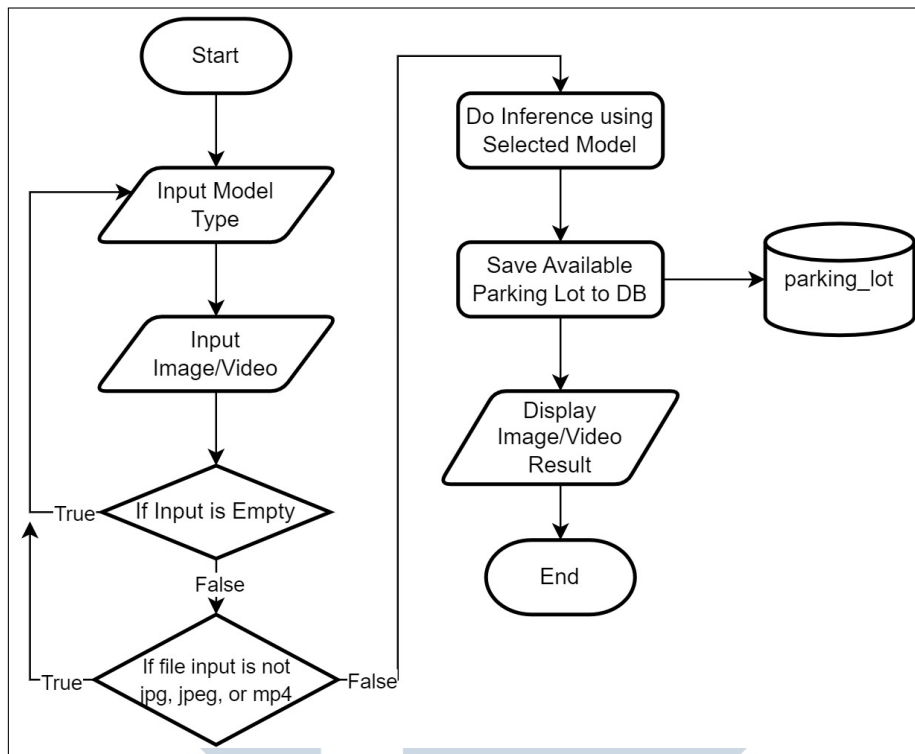
### 3.6 Perancangan

Setiap model yang telah dievaluasi diintegrasikan ke dalam sebuah situs web yang dibuat menggunakan *framework* Flask. Sebagai demo sistem, digunakan rekaman video yang diambil dari internet baik untuk pagi maupun malam hari. Secara nyata, seharusnya *input* merupakan sebuah rekaman CCTV yang berjalan secara *live*. Namun, untuk kemudahan penelitian ini, digunakan sebuah rekaman video biasa pengganti rekaman CCTV sebagai *input*. Selain itu, dibuatkan juga sebuah situs web untuk menampilkan jumlah ketersediaan tempat parkir secara *real-time* yang didapat dari hasil deteksi *input* sebelumnya. Di belakang situs web

monitor ini, akan digunakan *web socket* yang mengambil data terbaru dari *database* secara berkala. Secara total, terdapat 4 situs web dan 1 sistem *web socket* yang dibuat sebagai berikut.

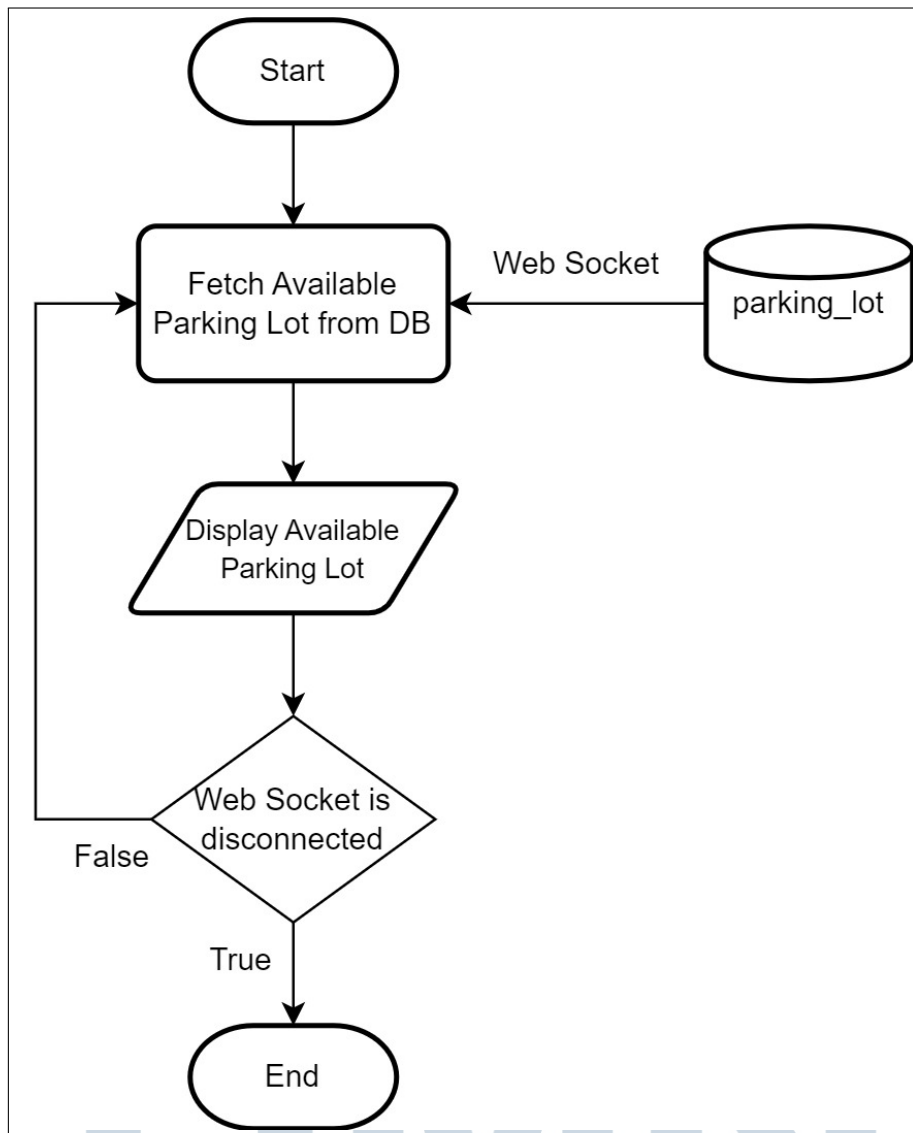
1. Situs Web Deteksi menggunakan YOLOv7  
*User* dapat memilih model YOLOv7 yang ingin digunakan (YOLOv7-A atau YOLOv7-B), memasukkan *input* gambar atau video, lalu hasil akan ditampilkan pada situs web serta disimpan pada *database*.
2. Situs Web Deteksi menggunakan YOLOv8  
*User* dapat memilih model YOLOv8 yang ingin digunakan (YOLOv8-A atau YOLOv8-B), memasukkan *input* gambar atau video, lalu hasil akan ditampilkan pada situs web serta disimpan pada *database*.
3. Situs Web Deteksi menggunakan YOLOv9  
*User* dapat memilih model YOLOv9 yang ingin digunakan (YOLOv9-A atau YOLOv9-B), memasukkan *input* gambar atau video, lalu hasil akan ditampilkan pada situs web serta disimpan pada *database*.
4. Situs Web Monitor Ketersediaan Tempat Parking  
*User* dapat melihat jumlah ketersediaan tempat parkir hasil inferensi terakhir yang didapat dari sistem deteksi YOLOv7, YOLOv8, dan YOLOv9.
5. Sistem *Web Socket*  
Sistem ini akan selalu jalan untuk memberikan *update* data terbaru dari *database* setiap 1 detik untuk ditampilkan di situs web monitor secara *live*.

U M M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

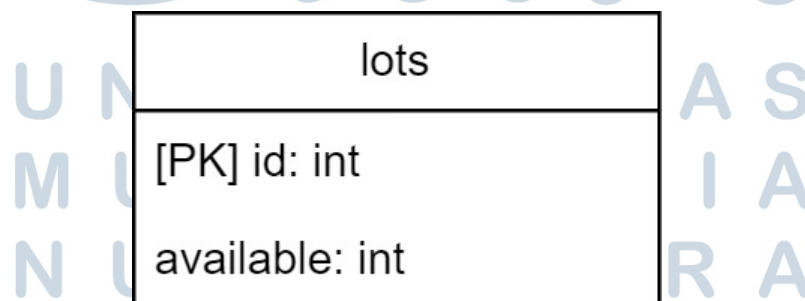


Gambar 3.5. Detection System Flowchart

UMN  
 UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA



Gambar 3.6. Monitor System Flowchart



Gambar 3.7. Struktur Tabel "lots" Pada Database

Untuk memperjelas *flow* dari sistem yang dibuat, maka dapat dilihat

*flowchart* untuk sistem deteksi pada Gambar 3.5 dan *flowchart* untuk sistem monitor pada Gambar 3.6. Selain itu, struktur tabel yang digunakan pada *database* dapat dilihat pada Gambar 3.7.

### **3.7 Dokumentasi**

Setiap proses dalam penelitian akan didokumentasi dari tahap awal sampai pada tahap akhir. Oleh karena itu, tahap ini akan dilakukan sepanjang penelitian berlangsung.

