

BAB 3

METODOLOGI PENELITIAN

Berikut merupakan metodologi penelitian yang akan dilakukan selama penelitian berlangsung.

3.1 Tahapan-tahapan penelitian

Tahapan-tahapan perancangan sistem dari penelitian "Implementasi framework YOLOV7 untuk mendeteksi penyusup dengan citra digital" adalah sebagai berikut.

1. Identifikasi dan perumusan masalah

Masalah yang teridentifikasi dan dirumuskan adalah mendeteksi penyusup yang memasuki rumah disaat rumah sedang kosong dengan menggunakan citra digital.

2. Studi Literatur

Mempelajari dan memahami dasar-dasar dari teori yang digunakan untuk meneliti permasalahan yang telah dirumuskan. Tahap ini menjadikan referensi pustaka sebagai acuan. Referensi pustaka dapat bersumber dari seluruh kajian tertulis seperti artikel atau buku yang berkaitan dengan Implementasi framework YOLOV7 untuk mendeteksi penyusup dengan citra digital.

3. Pengumpulan dan pengolahan *dataset*

Dataset yang digunakan dalam penelitian ini didapatkan dari situs Roboflow yang diunggah oleh *alive*[16]. *Dataset* berisi citra manusia yang diambil secara *general*. Anotasi label yang digunakan berbentuk *.txt* dimana file *.txt* berisi kelas dan koordinat *bounding box* dari citra gambarnya. *Dataset* juga telah diolah dengan ukuran gambarnya telah di *stretch* menjadi size 640x640. Setelah itu, dilakukan *quality check* satu persatu pada gambar dalam *dataset*, sehingga gambar yang tidak jelas seperti gambar patung atau sepatu tidak digunakan sebagai gambar untuk *training* model nantinya.

4. Analisa dan Perancangan Sistem

Analisa dilakukan untuk menentukan algoritma yang akan diimplementasikan untuk mengatasi masalah yang telah dirumuskan.

Pada penelitian ini, YOLOv7 akan digunakan untuk mengatasi masalah yang telah dirumuskan. YOLOv7 yang akan melakukan klasifikasi dari citra gambar.

5. Pengembangan Sistem

Pengembangan sistem dimulai dengan pelatihan model dengan menggunakan *dataset* yang telah diolah terlebih dahulu. Setelah itu, hasil dari pelatihan model akan digunakan untuk mendeteksi penyusup yang terekam oleh kamera di dalam rumah.

6. Evaluasi Sistem

Sistem yang telah dikembangkan akan dievaluasi kembali untuk mengetahui kinerja model yang telah dilatih dan digunakan di dalam sistem.

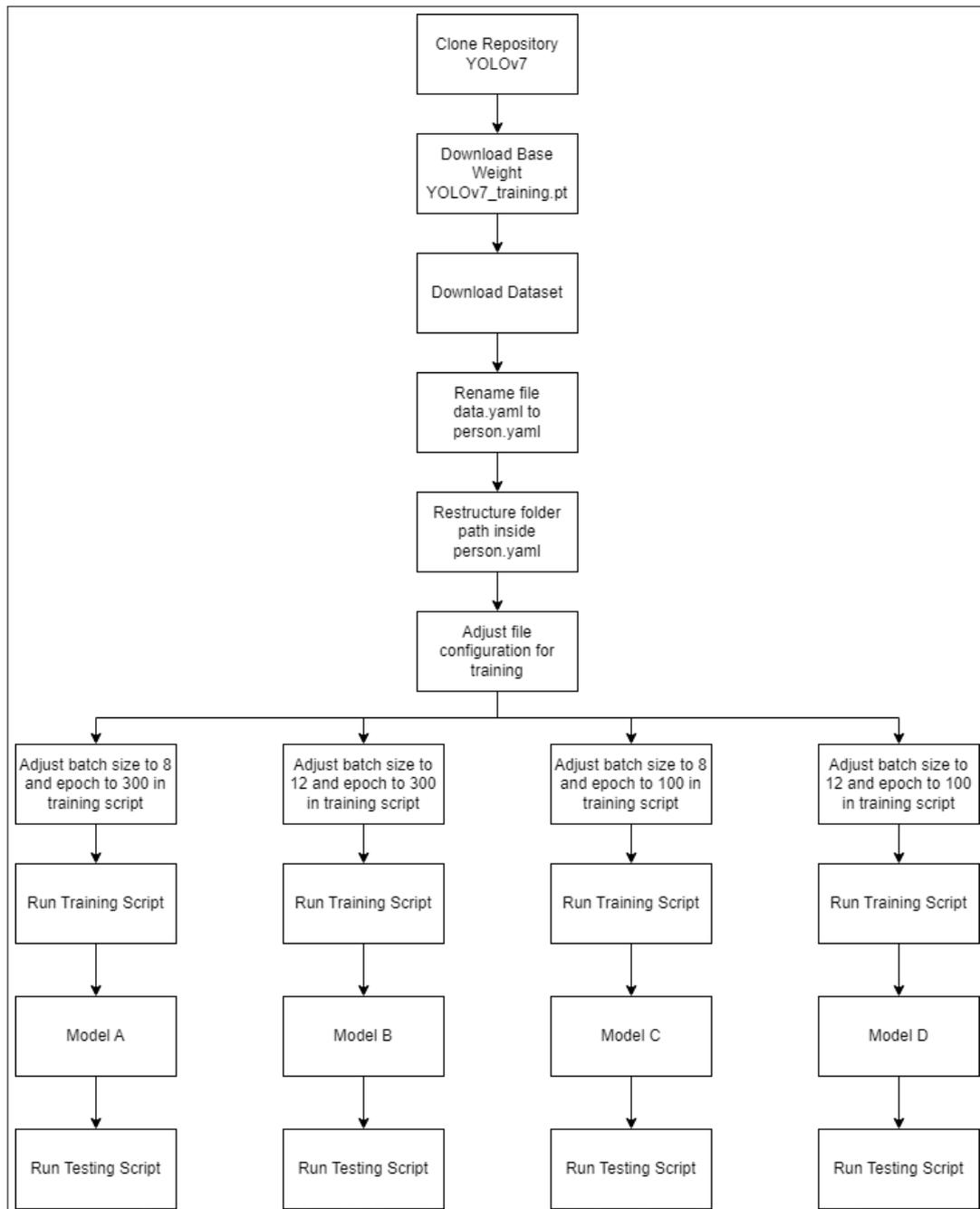
7. Penyusunan Laporan

Penulisan laporan adalah sebuah dokumentasi dari awal proses penelitian hingga selesai. Laporan akan berisi jawaban dari rumusan masalah berdasarkan hasil yang didapatkan selama penelitian berlangsung.

3.2 Alur Kerja

Gambar 3.1 adalah representasi dari alur kerja dari penelitian "Implementasi framework YOLOV7 untuk mendeteksi penyusup dengan citra digital".





Gambar 3.1. Alur Kerja Penelitian

3.3 Perancangan dan Pengembangan Sistem

Perancangan dan pengembangan sistem pada penelitian "Implementasi framework YOLOV7 untuk mendeteksi penyusup dengan citra digital" terbagi menjadi beberapa bagian. Pertama, instalasi YOLOv7 dan pengunduhan *dataset*.

Kedua, persiapan dan proses *training model*. Ketiga, evaluasi model. Keempat, perancangan *flowchart* sistem.

3.3.1 Instalasi YOLOv7 dan Pengunduhan Dataset

Pertama-tama YOLOv7 diunduh dari *repository* YOLOv7 yang diunggah oleh WongKinYiu yang terdapat pada situs GitHub. Lalu instalasi dilakukan dengan menjalankan perintah sebagai berikut.

```
pip install -r requirements.txt
```

Setelah itu, unduh model *base weight* "yolov7_training.pt" yang akan digunakan untuk training dari *repository* YOLOv7 yang sama seperti sebelumnya. Setelah selesai mengunduh modelnya, unduh *dataset* melalui link berikut.

Setelah *dataset* selesai diunduh, *dataset* di ekstrak dan dimasukkan ke dalam *folder* YOLOv7 khususnya pada bagian *folder* data. *Dataset* yang digunakan terdiri dari 4999 gambar dan 4999 anotasinya yang terbagi menjadi 3 *folder*. *Folder* Train berisi 3499 gambar dan 3499 anotasi, Test berisi 500 gambar dan 500 anotasi, dan Valid berisi 1000 gambar dan 1000 anotasi.

3.3.2 Persiapan Dataset

Pada tahap ini, mulanya diperlukan pengubahan nama *file* data.yaml menjadi "person.yaml". Hal ini dilakukan untuk mempermudah penelitian dalam mengenali nama *file* dan *dataset* yang digunakan. Setelah diubah, sesuaikan struktur *folder* pada parameter *train*, *test*, dan *valid* sesuai dengan struktur *folder dataset* yang akan digunakan (jika belum sesuai). Setelah itu, buka *folder* yolov7/cfg/training dan duplicate *file* yolov7.yaml lalu ubah nama *file* tersebut menjadi "yolov7-person.yaml". Lalu buka *file* yolov7-person.yaml tersebut, lalu sesuaikan parameter *nc* di dalamnya sesuai dengan jumlah kelas yang terdapat pada *dataset*. Pada *dataset* ini, jumlah kelas yang ada hanya satu kelas. *Dataset* yang digunakan memiliki anotasi dalam bentuk *file* .txt yang berisi koordinat-koordinat dari *bounding box* pada gambar.

3.3.3 Proses Training Model

Pada tahap ini, proses *training* dapat dilakukan. Untuk melakukan proses *training*, diperlukan perintah-perintah agar proses *training* dapat berjalan dengan

lancar dan baik. Berikut perintah-perintah dan penjelasannya secara singkat.

Tabel 3.1. Perintah-perintah *Training*

Perintah	Deskripsi
-workers	Jumlah prosesor yang digunakan untuk membagi gambar sejumlah <i>batch</i> yang ditentukan
-batch-size	Jumlah dari pembagian gambar yang akan diproses secara bersamaan dalam satu waktu saat proses <i>training</i>
-device	Menentukan perangkat yang digunakan untuk melakukan <i>training</i>
-data	<i>Path file</i> data berbentuk <i>.yaml</i> yang akan digunakan pada saat <i>training</i>
-img	Ukuran piksel dari gambar <i>dataset</i> yang akan digunakan untuk melakukan <i>training</i>
-cfg	<i>Path file</i> konfigurasi berbentuk <i>.yaml</i> yang digunakan untuk melakukan <i>training</i>
-weights	<i>Path file base weight</i> berbentuk <i>.pt</i> yang akan digunakan untuk melakukan <i>training</i>
-name	Nama dari <i>folder</i> tempat penyimpanan hasil <i>training</i>
-hyp	<i>Path file</i> hyperparameter berbentuk <i>.yaml</i> yang akan digunakan untuk melakukan <i>training</i>
-epoch	Jumlah iterasi dari pemrosesan seluruh gambar pada <i>dataset</i> saat melakukan <i>training</i> model

Proses *training* akan dilakukan dengan menggunakan 3 macam perintah. Masing-masing perintah akan divariasi dari *batch-size* dan juga *epochnya* untuk melihat perbedaan akurasi dari variasi kombinasi *batch-size* dan *epoch*. Perintah yang digunakan untuk melakukan proses *training* dapat dilihat pada Tabel 3.2.

Tabel 3.2. Variasi perintah *training*

Variasi	Perintah
Batch-size 8 dan epoch 300	<pre>python train.py --workers 0 --batch-size 8 --device 0 --data data\person.yaml --img 640 640 --cfg cfg\training\yolov7-person.yaml --weights yolov7_training.pt --name yolov7- person --hyp data\hyp.scratch.custom.yaml -- epochs 300</pre>
Batch-size 12 dan epoch 300	<pre>python train.py --workers 0 --batch-size 12 --device 0 --data data\person.yaml --img 640 640 --cfg cfg\training\yolov7-person.yaml --weights yolov7_training.pt --name yolov7- person --hyp data\hyp.scratch.custom.yaml -- epochs 300</pre>
Batch-size 8 dan epoch 100	<pre>python train.py --workers 0 --batch-size 8 --device 0 --data data\person.yaml --img 640 640 --cfg cfg\training\yolov7-person.yaml --weights yolov7_training.pt --name yolov7- person --hyp data\hyp.scratch.custom.yaml -- epochs 100</pre>
Lanjut pada halaman berikutnya	

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.2 Variasi perintah *training*

Variasi	Perintah
Batch-size 12 dan epoch 100	<pre>python train.py --workers 0 --batch-size 12 --device 0 --data data\person.yaml --img 640 640 --cfg cfg\training\yolov7-person.yaml --weights yolov7_training.pt --name yolov7- person --hyp data\hyp.scratch.custom.yaml -- epochs 100</pre>

3.3.4 Evaluasi Model

Setelah model hasil *training* didapatkan, model akan dievaluasi dengan cara melakukan *test* pada model untuk melihat akurasi yang dihasilkan. Untuk melakukan evaluasi dengan *testing*, jalankan perintah pada Tabel 3.3.

Tabel 3.3. Variasi perintah untuk evaluasi model

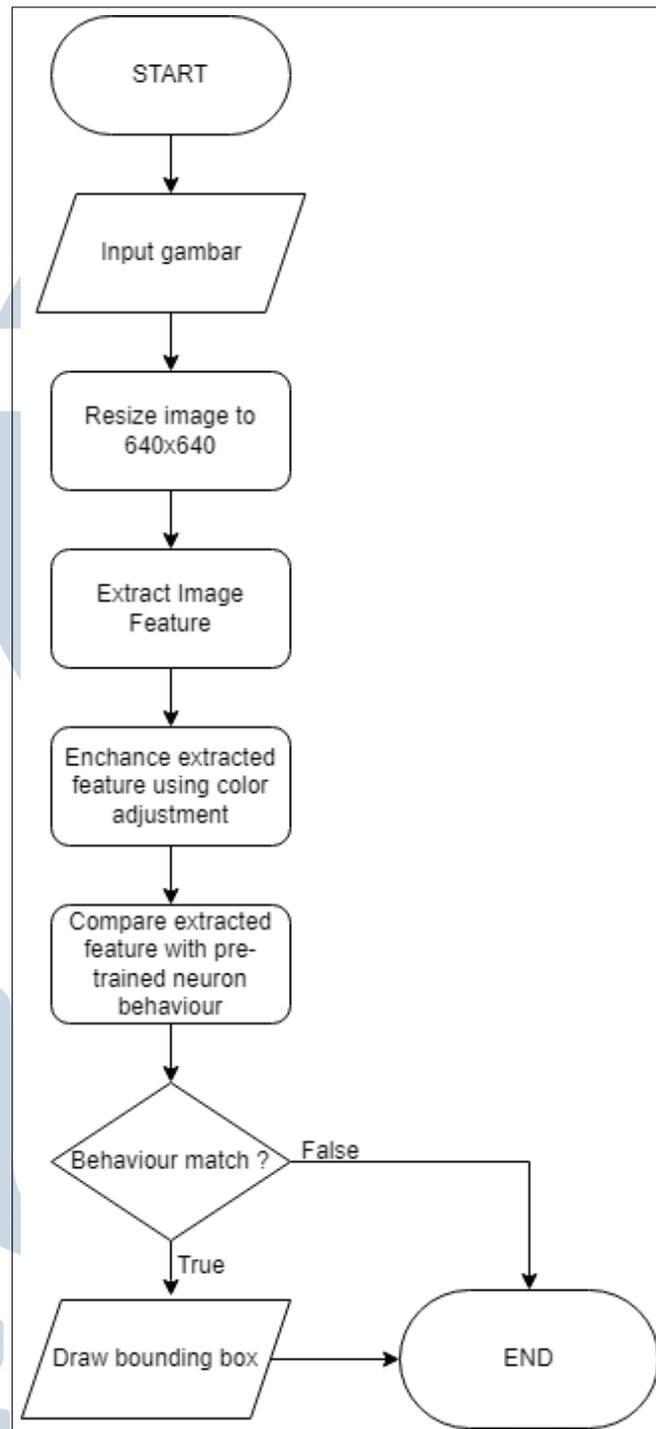
Variasi	Perintah
Batch-size 8 dan epoch 300	<pre>python test.py --weights runs\train\ person_dataset_yolo_training_batch_8_300\ weights\best.pt --batch-size 2 --device 0 -- data data\person.yaml --img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-person-val -- task val</pre>
	Lanjut pada halaman berikutnya

Tabel 3.3 Variasi perintah untuk evaluasi model

Variasi	Perintah
Batch-size 12 dan epoch 300	<pre>python test.py --weights runs\train\ person_dataset_yolo_training_batch_12 -300\ weights\best.pt --batch-size 2 --device 0 -- data data\person.yaml --img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-person-val -- task val</pre>
Batch-size 8 dan epoch 100	<pre>python test.py --weights runs\train\ person_dataset_yolo_training_batch_8 -100\ weights\best.pt --batch-size 2 --device 0 -- data data\person.yaml --img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-person-val -- task val</pre>
Batch-size 12 dan epoch 100	<pre>python test.py --weights runs\train\ person_dataset_yolo_training_batch_12 -100\ weights\best.pt --batch-size 2 --device 0 -- data data\person.yaml --img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-person-val -- task val</pre>

3.3.5 Perancangan Flowchart Sistem

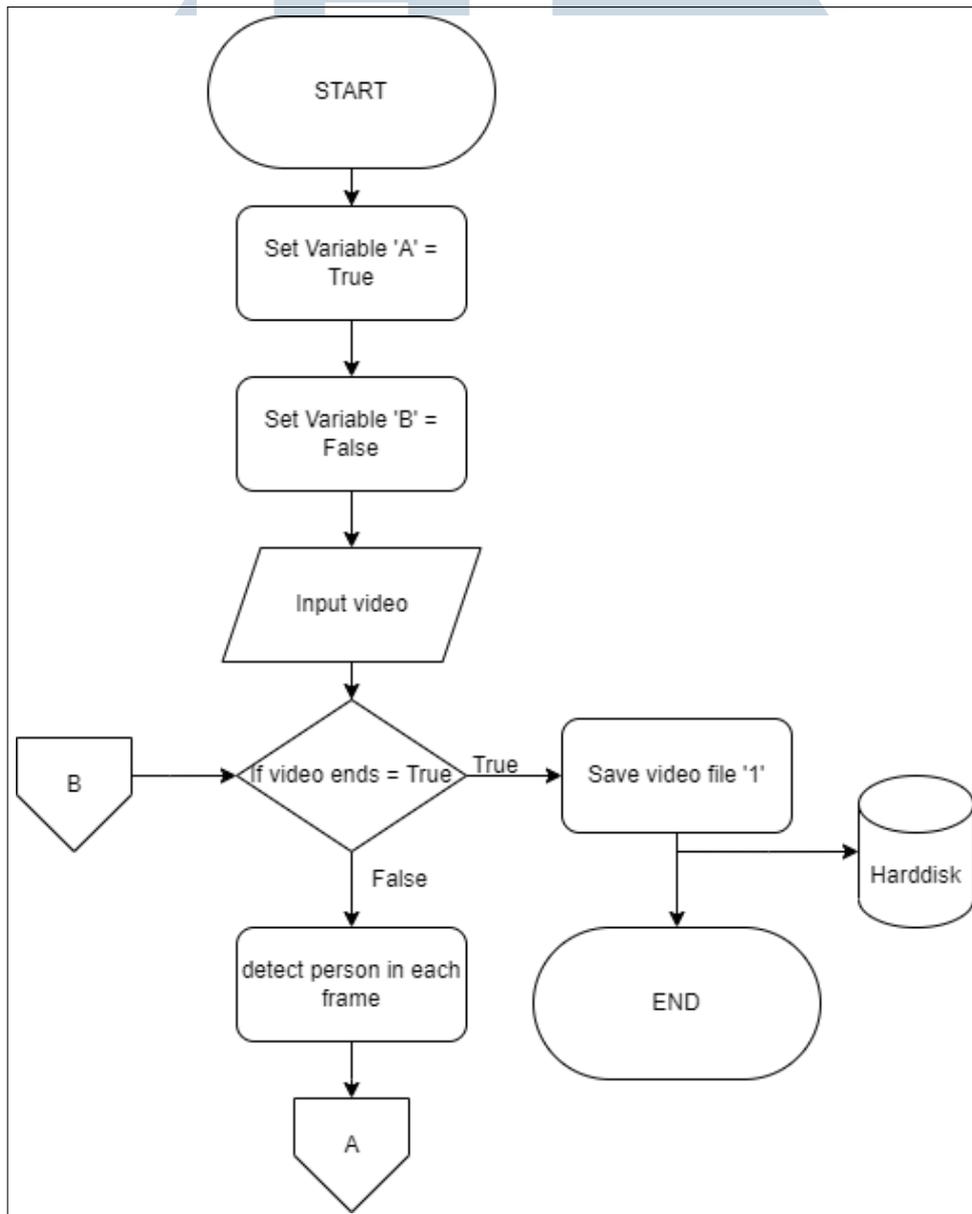
Pada penelitian ini, *Flowchart* akan terbagi menjadi 3, *flowchart* CNN, *flowchart* sistem keamanan dan *flowchart* dari *bot* telegram.



Gambar 3.2. Flowchart CNN

Flowchart dari sistem CNN dapat dilihat pada Gambar 3.2. Sistem dimulai dengan menerima gambar yang akan dideteksi, lalu gambar yang diterima akan dirubah ukurannya menjadi 640x640. Setelah ukuran gambar dirubah, sistem akan melakukan ekstraksi fitur gambar. Hasil ekstraksi fitur setelah itu akan di-

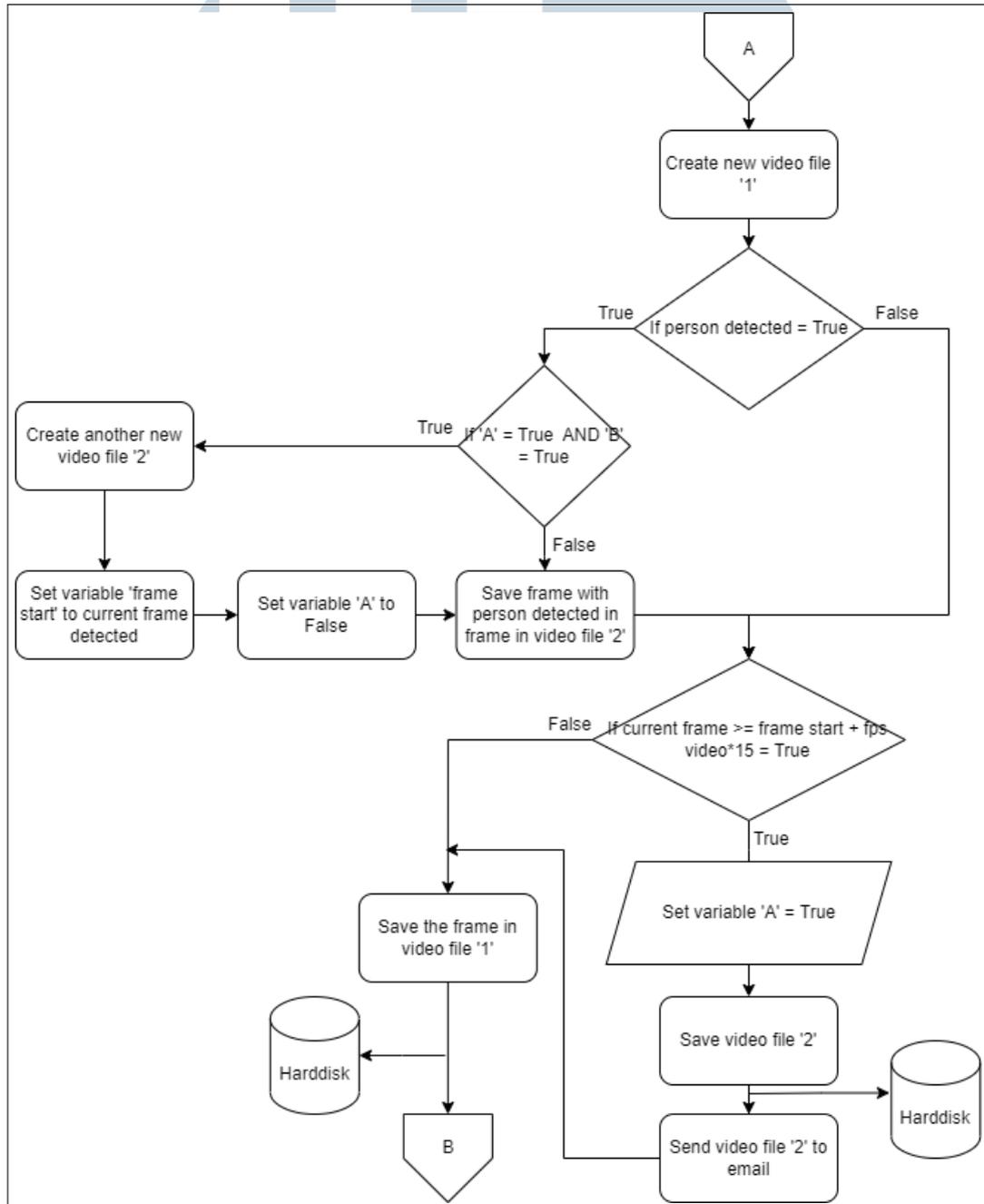
enhance agar fitur dapat terlihat jelas dan mudah dikenali *behaviour*-nya oleh *artificial neuron*. Setelah fitur di-*enhance*, fitur gambar akan dilewati melalui *layer-layer artificial neuron* yang telah dilatih terlebih dahulu. Jika fitur gambar memiliki *behaviour* yang sama dengan *artificial neuron*-nya, maka *bounding box* akan digambar untuk memberikan identifikasi dari gambar tersebut.



Gambar 3.3. Flowchart Sistem Keamanan

Flowchart dari sistem keamanan dimulai dengan mendeklarasikan variabel A menjadi *True* dan variabel B menjadi *False*. Setelah itu dilanjutkan dengan video dimasukkan ke dalam sistem pendeteksian. Sistem dari YOLOv7 akan

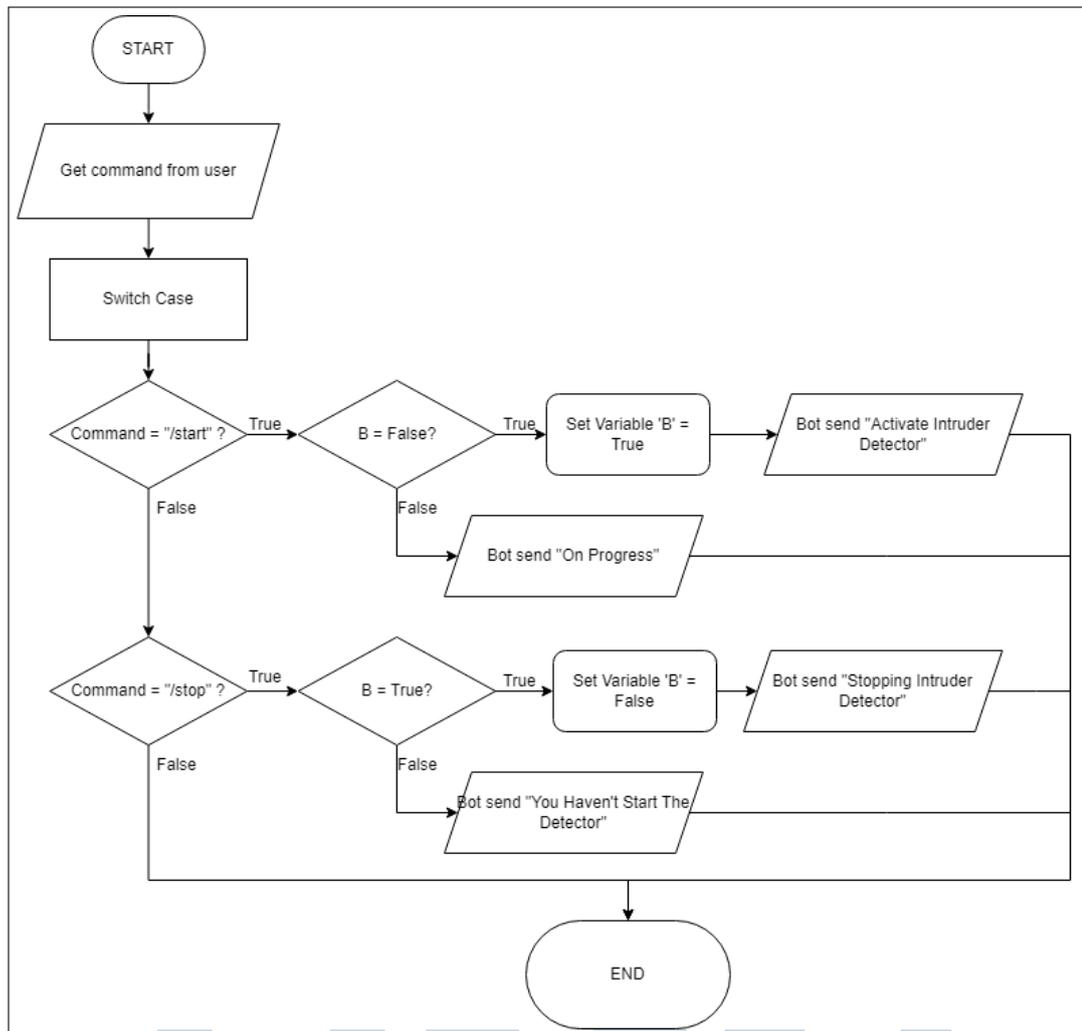
melakukan *looping* untuk melakukan pendeteksian. *Looping* sistem akan berjalan terus menerus hingga *video* yang ingin dideteksi selesai. Pendeteksian dimulai dengan pendeteksian fitur pada setiap *frame*. Setelah itu *flowchart* sistem keamanan dilanjutkan dengan *off-page reference* menuju *flowchart* selanjutnya seperti pada Gambar 3.4.



Gambar 3.4. Flowchart Sistem Keamanan

Sistem dari YOLOv7 akan melakukan pendeteksian sekaligus membuat *file video '1'* untuk menyimpan rekam jejak pendeteksian video selama berlangsung. Jika terdeteksi adanya manusia, maka sistem akan mengecek variabel A yang mana jika variabel A dan B adalah True, maka sistem akan membuat *file video* baru, lalu akan membuat variabel "*framestart*" menjadi *frame* dimana manusia terdeteksi. Setelah itu *frame* tersebut akan disimpan oleh sistem ke dalam *file video '2'* yang baru dibentuk dan variabel A akan diubah menjadi *False*. Jika sudah, terlepas dari terdeteksinya manusia atau tidak, sistem akan mengecek jika *frame* yang saat ini sedang dideteksi lebih besar sama dengan dari variabel "*framestart*" dijumlahkan dengan *fps video* yang dikalikan 15, maka variabel A akan diubah kembali menjadi *True* agar pada saat pendeteksian manusia berlangsung, batas maksimal durasi dari *file video '2'* yang merekam pendeteksian manusia hanya 15 detik dan jika terdeteksi lebih dari 15 detik maka akan membentuk *file video* baru. Lalu setelah variabel A dirubah kembali menjadi *True*, sistem akan menyimpan *file video '2'* menjadi file jadi dan mengirimkan *file video '2'* tersebut menuju email pengguna sebagai *Alert*. Setelah itu, alur akan kembali pada Gambar 3.3 dimana setelah itu akan dilakukan pengecekan apakah durasi dari *video* telah selesai atau belum, jika belum maka sistem akan terus melakukan *looping* hingga durasi *video* selesai.





Gambar 3.5. Flowchart bot telegram

Flowchart dari bot telegram dimulai dengan menerima *input command* pengguna. Lalu jika *command* yang diterima adalah */start* maka sistem akan mengecek isi dari variabel B. Jika variabel B adalah *False* maka bot akan mengubah variabel B menjadi *True* lalu bot akan mengirimkan pesan "*Activate Intruder Detector*". Jika isi variabel B adalah *True*, maka bot akan mengirimkan pesan "*On Progress*". Jika *command* yang diterima adalah */stop* maka sistem akan mengecek isi dari variabel B. Jika variabel B adalah *True* maka bot akan mengubah variabel B menjadi *False* lalu bot akan mengirimkan pesan "*Stopping Intruder Detector*". Jika isi variabel B adalah *False*, maka bot akan mengirimkan pesan "*aYou haven't Start The Detector*".