

BAB III METODE PENELITIAN

3.1. Bahan Penelitian

Bahan penelitian ini membutuhkan dataset gambar manusia yang memiliki anotasi atau label pada setiap gambar. Anotasi dilakukan untuk mendapatkan ukuran objek dan juga posisi pada sebuah gambar dengan memberikan kotak batas (*bounding box*). Setiap gambar akan memiliki file label.txt, dan setiap gambar harus memiliki nama file yang sama misal gambar1.jpg harus memiliki label gambar1.txt dengan posisi yang di ikuti oleh data lainnya.

Selain gambar juga dibutuhkan beberapa bahan untuk melakukan training seperti *framework* darknet yang ditulis menggunakan bahasa C. File parameter konfigurasi untuk yolo yang berekstensi .cfg, file .names yang diberisikan kelas *string*, file .data yang menentukan berapa jumlah kelas yang digunakan serta lokasi dari file train, validasi, cfg, names dan backup untuk menyimpan file .backup dan juga bobot yang dilatih.

3.2. Peralatan Penelitian

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah:

1. laptop Asus dengan prosessor Intel® Core™ i5-6200U CPU @ 2.30GHz 2.40GHz, dengan RAM 8,00GB dengan Operasi sistem Window 10 Pro 64-bit dan Linux Ubuntu 18.04.
2. Webcam Logitech C525 Portable HD WEBCAM ,Resolusi : 720p
Maksimal fps : 30fps.

b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan adalah:

1. Visual Studio Code
2. Glade

3.3. Lokasi Penelitian dan Waktu Penelitian

Lokasi penelitian akan dilakukan di lobby kampus FTIK (Fakultas Teknik dan Ilmu Komputer), penelitian ini dimulai dari Desember 2019. Adapun rincian penelitian ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Tahapan Penelitian

No	Kegiatan	Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur	█															
2	Pengajuan Judul	█															
3	Analisa Kebutuhan dan Pengumpulan data									█							
4	Training													█			
5	Uji Coba Model secara Realtime													█			
6	Implentasi Perekaman													█			
7	Uji Coba Perekaman													█			
8	Evaluasi													█			

1.4. Prosedur Percobaan

Tahapan percobaan pada penelitian ini terdiri dari beberapa tahapan yang terdiri dari analisa kebutuhan dan mengumpulkan data, melakukan training dataset, uji coba model training dan implementasi dengan perekaman. Training akan dilakukan dengan metode YOLOv3. Lalu uji coba akurasi perekaman berdasarkan objek yang terdeteksi, alur prosedur dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur Prosedur Percobaan

1.5. Analisis Data

Pada penelitian ini akan dilakukan analisis data agar mengetahui kebutuhan untuk melakukan penelitian, maka akan dilakukan tahapan sebagai berikut:

1.5.1. Analisa dan Pengumpulan Data

Untuk membuat training model pada penelitian ini dibutuhkan dataset manusia. *Dataset* akan diberikan kotak batas (*bounding box*) dengan cara memberikan anotasi/label pada sebuah gambar. *Dataset* yang dipakai pada penelitian ini adalah *dataset OpenImage* dengan mengambil 1 kelas saja yaitu 'person'. *Dataset* ini sudah memiliki anotasi/label sehingga tidak diperlukan lagi aktifitas melabeli gambar secara manual menggunakan *tools* yang tersedia. Gambar yang digunakan sebanyak 500 gambar dan label menyamai jumlah gambar sedangkan anotasinya akan memiliki jumlah yang berbeda tergantung dari seberapa banyak objek 'person' pada satu gambar.

Anotasi pada gambar akan memiliki data seperti *object class id*, *center x*, *center y*, *width*, dan *height*. *Object class id* adalah sebuah *integer* yang mewakili kelas objek, karena yang digunakan hanya 1 kelas saja maka nilainya akan selalu 0. *Center x*, dan *center y* adalah koordinat dari *bounding box* atau kotak pembatas, dinormalisasi atau dibagi oleh lebar gambar dan tinggi masing-masing. *Width*, dan *height* adalah lebar dan tinggi kotak pembatas lalu dibagi lagi oleh lebar dan tinggi masing-masing (Sunita Nayak, 2019).

Dataset dan label yang sudah terkumpul kemudian akan dibagi menjadi 90% data train dan 10% data test.



Gambar 3. 2 Labeling Gambar Manual dengan Tools

Sumber: https://www.researchgate.net/figure/Using-LabelImg-to-label-the-target-and-then-a-xml-file-will-be-saved-which-contains_fig1_334489831

Selain *dataset*, penelitian ini membutuhkan beberapa *file* konfigurasi YOLO, *file* data dan juga jaringan konvolusi untuk dapat melakukan *training* menggunakan *framewok* darknet.

1. Yolo.cfg

File cfg ini merupakan file parameter konfigurasi yolo untuk melakukan training dengan jaringan konvolusi.

2. File.data

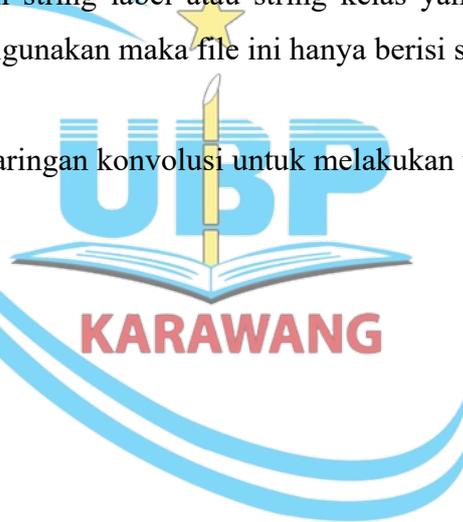
File ini berisikan informasi jumlah kelas yang digunakan serta lokasi penyimpanan file validasi, test, names, dan lokasi backup untuk menyimpan bobot hasil training.

3. Person.names

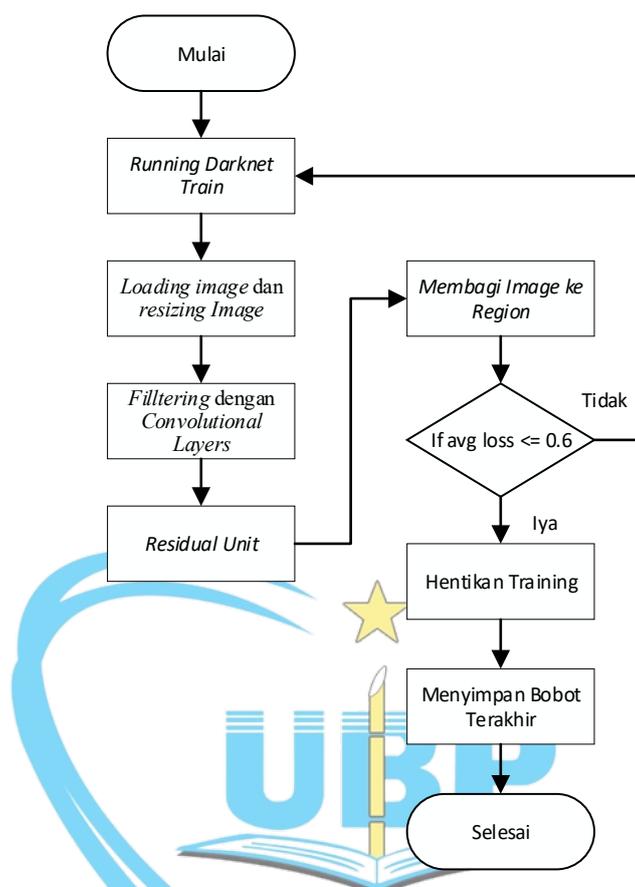
File ini meyimpan string label atau string kelas yang digunakan karena hanya 1 kelas yang digunakan maka file ini hanya berisi string 'person'.

4. Bobot konvolusi

Bobot ini adalah jaringan konvolusi untuk melakukan transfer learning.



1.6. Training Dataset



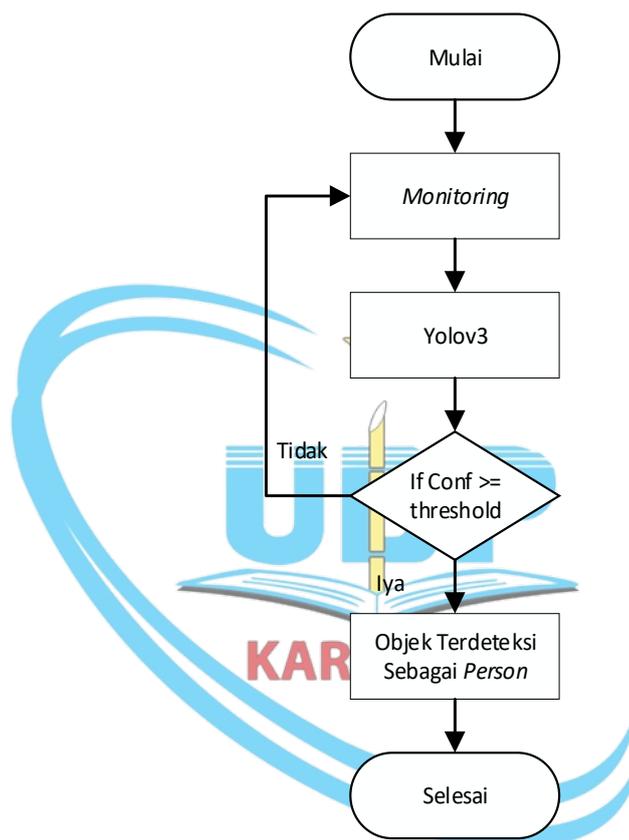
Gambar 3. 3 Alur Training Dataset

Blok pertama adalah menjalankan *training* menggunakan *framework* darknet, dengan menggunakan *command* `./darknet detector train data/person.data data/Yolov3.cfg data/darknet53.75`. Blok kedua akan dilakukan pemuatan gambar dan akan melakukan *resizing* pada gambar. Seluruh gambar yang sudah ter-*resizing* akan memasuki iterasi pertama dan masuk ke proses *filtering* dengan *Convolutional Layers*. Blok ketiga pada proses *filtering* akan terjadi ekstraksi fitur dengan menggunakan *Convolutional Layers*. Blok keempat setelah memasuki *Convolutional Layers*, maka akan memasuki residual unit dimana akan ada 5 residual blok dan hanya akan di prediksi dalam 3 *downsampled layer* terakhir. Blok kelima akan dilakukan pembagian gambar menjadi wilayah atau *region* lalu memprediksi *bounding box* atau kotak batas dan probabilitas untuk setiap *region*. Hal ini terjadi pada setiap iterasi/epoch. Blok keenam bobot atau model akan disimpan di folder backup. Proses ini akan terjadi setiap iterasi mencapai angka 200

dan kelipatannya. Proses diakhiri jika avg loss mencapai minimal 0.6 atau kurang dari itu.

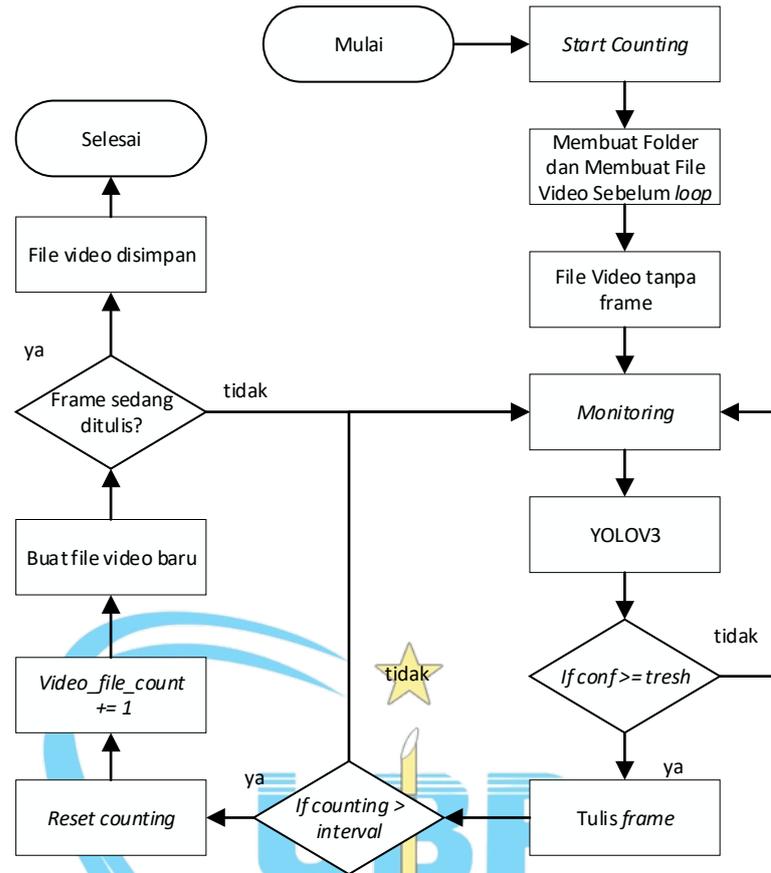
1.7. Implementasi Realtime dan Perekaman Otomatis

Hasil training dibuat *realtime* untuk mendeteksi keberadaan manusia dengan menggunakan *Webcam*. Alur pendeteksian *realtime* seperti Gambar 3.4.



Gambar 3. 4 Alur Pendeteksian Realtime

Block pertama, kamera akan memonitoring dengan menggunakan bahasa python3 dan fungsi dari opencv. Block kedua, Yolov3 merupakan dataset yang sudah dilatih dan menjadi model latih. Lalu mengklasifikasikan objek berdasarkan bobot yang sudah dilatih dan terhubung dengan file names yang akan menjadi keluaran label string. Model akan dijalankan dan membagi gambar menjadi wilayah dan skala, wilayah yang memilih skor tinggi akan dianggap sebagai objek. Block ketiga, *decision* jika (*confidence*) nilai kepercayaan lebih besar dan sama dengan (*threshol*) ambang batas maka akan dianggap sebagai objek, jika tidak maka akan tetap hanya memonitoring. Block keempat, objek terdeteksi saat syarat pada block sebelumnya terpenuhi, lalu menampilkan informasi pada *frame*.



Gambar 3. 5 Alur Realtime dan Pererekaman Otomatis

Blok pertama, akan dimulai *counting* sejak pertama kali program dijalankan. Blok kedua, akan dibuat folder dengan label tanggal dan juga waktu, dan buat file video kosong pada folder. Blok ketiga, video yang dibuat tidak akan berisikan frame. Blok ketiga, kamera akan memonitoring menggunakan bahasa python3 dan fungsi dari opencv. Blok keempat, YOLOV3 merupakan dataset yang sudah dilatih dan menjadi model latih. Lalu mengklasifikasikan objek berdasarkan bobot yang sudah dilatih dan terhubung dengan file names yang akan menjadi keluaran *label string*. Model akan dijalankan dan membagi gambar menjadi wilayah dan skala, wilayah yang memilih skor tinggi akan dianggap sebagai objek. Blok kelima *decision* jika (*confidence*) nilai kepercayaan lebih besar dan sama dengan (*Threshold*) ambang batas, maka akan dianggap objek dan menampilkan *bounding box*, label *person* dan juga nilai *confidence* dari objek yang di deteksi pada *frame*. Blok keenam tulis *frame* pada file video yang telah dibuat sebelumnya, kamera akan merekam video *frame by frame* ketika syarat dari blok sebelumnya terpenuhi. Video akan di *encoding* ke dalam bentuk avi. Blok ketujuh, *decision* cek

jika *counting* lebih besar dari nilai interval waktu yang ditentukan. Jika ya, maka lanjut ke blok delapan. Jika tidak, maka akan tetap memonitoring. Blok kedelapan, jika pada blok sebelumnya terpenuhi maka *reset counting*. Blok Sembilan, ketika *counting* di *reset* maka maka *video_file_count* akan ditambah satu. Blok sepuluh, akan dibuat file video dengan id baru. Blok sebelas, cek apakah *frame* sedang ditulis. Jika tidak, maka akan tetap memonitoring dan jika frame ditulis maka simpan video dan selesai.

1.8. Pengujian

Pengujian akan dilakukan dengan menguji akurasi model dalam mendeteksi objek manusia secara *realtime*. Berdasarkan penelitian dari (Indra, Agani, & Handayani, 2018). Pengujian dapat dilakukan dengan menentukan jumlah sample uji, lalu mengkalkulasi jumlah data sample yang benar dengan seluruh sample uji. Untuk mendapatkan nilai akurasi, maka digunakan rumus :

$$\text{Akurasi} = \frac{\text{Jumlah Data Benar}}{\text{Jumlah Seluruh Data}} \times 100\%$$

Selain pengujian model, akan dilakukan pengujian pada perekaman otomatis seberapa besar akurasi kamera mampu melakukan perekaman otomatis disaat ada objek manusia melewati atau berada dalam monitoring kamera.

1.9. Evaluasi

Evaluasi dilakukan agar mengetahui hasil dari implementasi model ke *realtime monitoring* dan perekaman otomatis. Dan mengetahui kekurangan dan juga kelebihan dari Yolo itu sendiri.