

BAB III

METODOLOGI PENELITIAN

3.1 Subjek dan Objek Penelitian

Subjek pada penelitian ini adalah *Coffee Roaster* yang bertugas untuk menyangrai dan menentukan kematangan *roast* biji kopi. Sementara objek yang digunakan dalam penelitian ini adalah *dataset* berupa biji kopi robusta dengan 4 tingkat kematangan *roasting* yang berbeda yaitu *green*, *light roast*, *medium roast*, dan *dark roast* yang selanjutnya akan digunakan untuk mendapatkan hasil akurasi dan kecepatan dalam melakukan *object detection* dengan menggunakan algoritma YOLO.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat

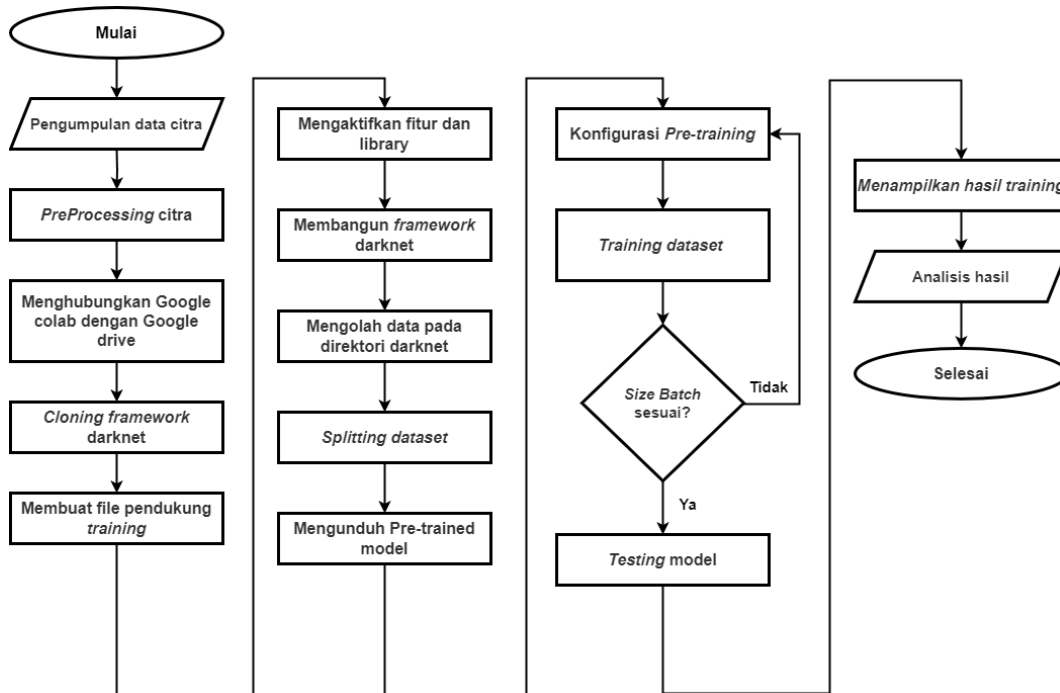
1. Laptop (CPU: AMD Ryzen 5 5500U, 2.1GHz, RAM: 16GB, OS: Windows 11-64 bit)
2. Kamera *smartphone* dengan resolusi 48MP
3. Kotak studio kecil
4. Tripod
5. *Software* Labeling
6. *Notebook* Google Colab
7. Penyimpanan Google Drive
8. Bahasa pemrograman Python
9. *Library* OpenCV, CUDNN

3.2.2 Bahan

600 citra biji kopi dengan 4 tingkat kematangan *roasting* yang sudah diproses dan diberi label dengan format YOLO.

3.3 Tahap Penelitian

Tahapan penelitian digambarkan melalui diagram pada Gambar 3.1 dibawah ini:



Gambar 3.1 Diagram alir tahap penelitian

Gambar 3.1 merupakan ilustrasi tentang tahapan penelitian yang akan dilakukan. Langkah-langkah yang dilakukan pada penelitian ini diantaranya:

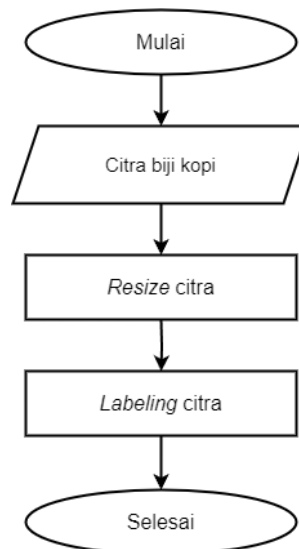
3.3.1 Pengumpulan data citra



Gambar 3.2 Diagram alir pengumpulan data citra

Gambar 3.2 menunjukkan, pengambilan *dataset* akan menggunakan biji kopi dengan 4 kematangan *roasting* yang berbeda, diantaranya *green*, *light roast*, *medium roast*, dan *dark roast*. *Dataset* diambil dengan jumlah 600 citra biji kopi, setiap kematangan *roast* memiliki 150 citra biji kopi. Citra biji kopi diambil menggunakan kamera *smartphone*.

3.3.2 *Pre-processing* citra



Gambar 3.3 Diagram alir *pre-processing* citra

Seperti pada Gambar 3.3 Terdapat dua tahap yang didapatkan dalam melakukan *pre-processing* citra, diantaranya:

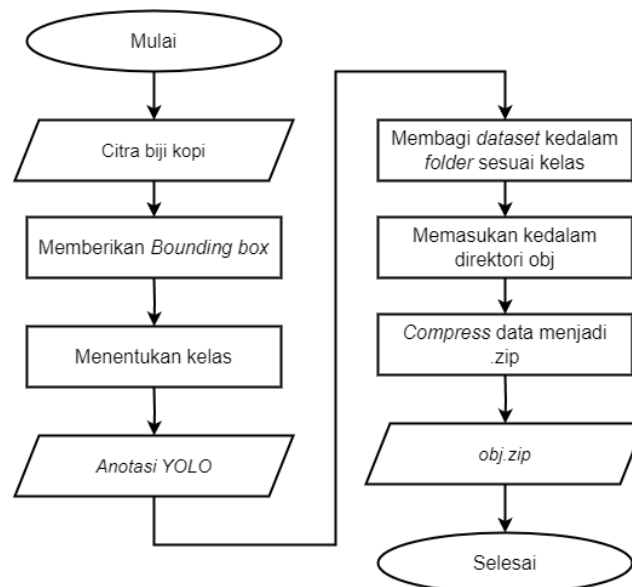
3.3.2.1 *Resize* citra



Gambar 3.4 Diagram alir *resize* citra

Gambar 3.4 merupakan diagram alir pada *resize* citra, dimana pada tahap ini, seluruh citra biji kopi yang telah diambil akan dipangkas dengan ukuran 448x448 untuk memudahkan model YOLO dalam tahap *training* dan *testing*.

3.3.2.2 *Labeling* citra



Gambar 3.5 Diagram alir *labeling* citra

Gambar 3.5 merupakan diagram alir pada tahap *labelling* citra. Dimana pada tahap ini citra biji kopi yang telah melewati tahap

resize akan diberikan *bounding box* atau garis tepi untuk menentukan lokasi objek serta kelas pada citra. *Output* dari tahap ini merupakan file anotasi dengan format .txt yang berisi kelas serta koordinat objek yang ditujukan membantu algoritma YOLOv4 untuk membantu proses *training*. *Dataset* yang telah diberi label akan dibuat masing-masing memiliki direktori sesuai dengan kelas kematangan *roasting* nya dan akan dimasukkan kedalam file bernama *obj* dengan eksistensi .zip.

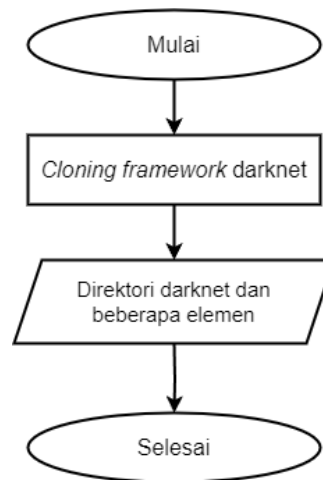
3.3.3 Menghubungkan Google Colab dengan Google Drive



Gambar 3.6 Diagram alir menghubungkan Google Colab dengan Drive

Penelitian ini menggunakan Google Colab sebagai *platform* dalam melakukan *training* dan *testing* nya, sehingga media penyimpanan untuk semua proses penelitian menggunakan Google Drive. Seperti pada Gambar 3.6, direktori yang baru akan dibuat yang nantinya direktori tersebut akan menyimpan semua data untuk pelatihan, dan setelahnya maka Google Colab dan Google Drive bisa disambungkan.

3.3.4 Cloning framework Darknet



Gambar 3.7 Diagram alir *Cloning framework* Darknet

Pada tahap ini, Gambar 3.7 menunjukkan algoritma YOLOv4 akan menggunakan *framework* Darknet, dimana *framework* ini berfungsi untuk mendukung proses *training* dan *testing* menjadi lebih mudah.

3.3.5 Membuat file pendukung *training*



Gambar 3.8 Diagram alir membuat file pendukung *training*

Gambar 3.8 merupakan susunan alir dalam melakukan pembuatan file pendukung, dimana pada tahap ini file obj.data berisikan petunjuk untuk direktori dan obj.names untuk menampilkan kelas apa saja yang digunakan untuk *dataset* akan dibuat untuk mendukung algoritma YOLOv4 dalam melakukan *training* dan *testing*.

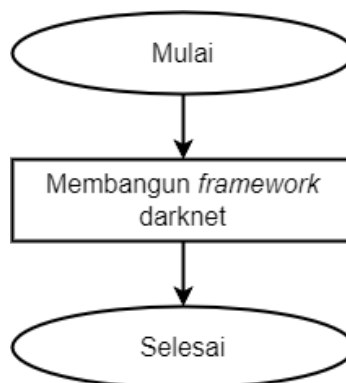
3.3.6 Mengaktifkan fitur dan *library*



Gambar 3.9 Diagram alir mengaktifkan fitur dan *library*

Gambar 3.9 menunjukkan alir pada saat melakukan aktivasi terhadap beberapa fitur dan *library* yang digunakan algoritma YOLOv4, tahap ini dilakukan menggunakan perintah pada Google Colab yang sudah terhubung dengan *framework* Darknet.

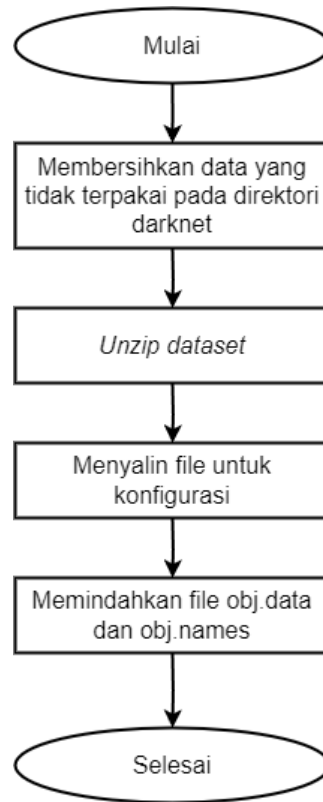
3.3.7 Membangun *framework* Darknet



Gambar 3.10 Diagram alir membangun *framework* Darknet

Seperti yang ditunjukkan pada Gambar 3.10, *framework* Darknet akan dibangun dalam arti kode yang berada pada direktori Darknet akan diubah menjadi *executable* sehingga nantinya bisa digunakan untuk *training* dan *testing*.

3.3.8 Mengolah data pada direktori Darknet



Gambar 3.11 Diagram alir mengolah data pada direktori Darknet

Pada tahap ini, Gambar 3.11 menunjukkan data yang telah dimasukkan kedalam direktori Darknet pada Google Drive akan dikelola, dimana langkah awal tahap ini adalah membersihkan data yang tidak terpakai pada direktori Darknet karena data tersebut akan diubah menjadi data yang dimiliki. Selanjutnya *dataset* berupa file dengan ekstensi .zip yang sudah dimasukkan kedalam Google Drive seperti pada akan di *unzip* menuju direktori yang sesuai. Berdasarkan langkah awal pada alir ini, file konfigurasi yang telah dibersihkan akan disalin kembali kedalam direktori konfigurasi dengan isi *default*. Langkah terakhir dalam alir tersebut adalah memindahkan file *obj.data* dan *obj.names* pada direktori *yolov4* menuju direktori data yang berada didalam direktori Darknet. Semua langkah tersebut dilakukan menggunakan perintah yang berada pada Google Colab.

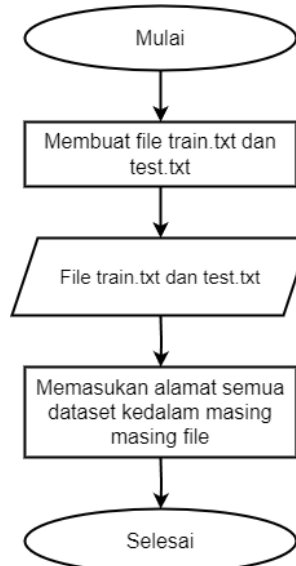
3.3.9 *Splitting dataset*

Pada penelitian ini, komposisi *splitting dataset* yang digunakan adalah 80:20 dimana 80% *dataset* digunakan untuk tahap *training* dan 20% untuk tahap *testing*. Skema pembagian *dataset* ditunjukkan pada Tabel 3.1

Tabel 3.1 Pembagian *dataset*

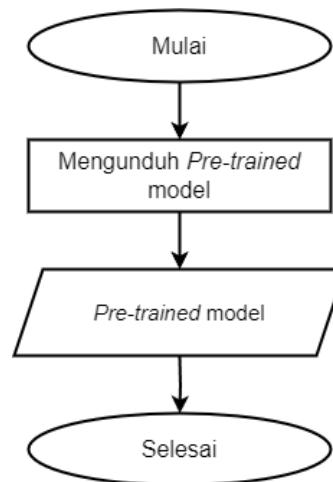
Kelas	<i>Dataset Training</i>	<i>Dataset Testing</i>
green	120	30
light	120	30
medium	120	30
dark	120	30
Jumlah	480	120
<i>Split Ratio</i>	80%	20%

Setelah didapatkan komposisi yang tepat, langkah selanjutnya adalah membuat file *train.txt* dan *test.txt* yang berisi seluruh alamat *dataset* pada direktori data didalam Google Colab, dimana file *train.txt* berisi 480 alamat *dataset*, dan file *test.txt* berisi 120 alamat *dataset*. Seluruh langkah pada tahap ini dilakukan seperti diagram alir pada Gambar 3.12 dibawah ini.



Gambar 3.12 Diagram alir *splitting dataset*

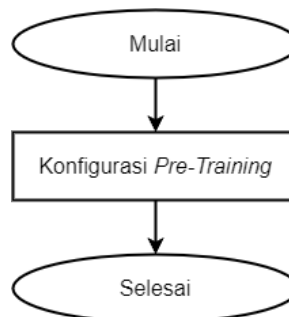
3.3.10 Mengunduh *pre-trained* model



Gambar 3.13 Diagram alir mengunduh *pre-trained* model

Dalam tahap ini, sebelum melakukan proses *training* dan *testing* dibutuhkan model yang sudah memiliki performa dengan hasil yang cukup baik sebelumnya, dengan tujuan algoritma YOLOv4 memiliki acuan dalam melakukan proses *training* dan *testing*, tahap ini akan dilakukan sesuai dengan diagram alir pada Gambar 3.13.

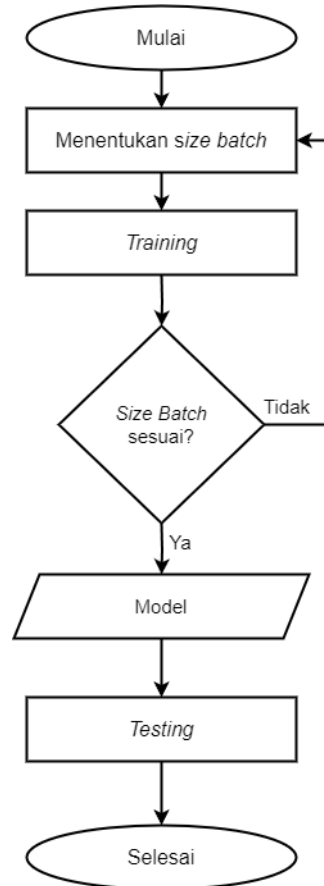
3.3.11 Konfigurasi *pre-training*



Gambar 3.14 Diagram alir konfigurasi *pre-training*

Seperti pada Gambar 3.14, tahap terakhir sebelum melakukan *training* adalah menentukan berapa saja nilai yang akan digunakan, nilai tersebut mencakup ketiga ukuran *size batch* yaitu 16, 32 dan 64, iterasi sebesar 3000 yang nantinya model akan dibentuk setiap 1000 iterasi, ukuran dimensi citra pada *dataset*, besarnya *learning rate*, dan nilai pada *filter* pada konvolusi dan yolo.

3.3.12 *Training dan testing dataset*



Gambar 3.15 Diagram alir *training* dan *testing dataset* menggunakan YOLO

Gambar 3.15 merupakan tahapan dalam melakukan proses *training* dan *testing* menggunakan algoritma YOLOv4. Setelah didapatkan *dataset* berupa citra dan label setiap citra nya, maka dilakukan tahap *training* untuk melatih algoritma YOLOv4 sesuai dengan beberapa konfigurasi yang sebelumnya telah ditentukan guna mendapatkan model yang bisa melakukan deteksi kematangan *roasting* biji kopi, tahap *training* ini dilakukan sebanyak 3 kali sesuai dengan banyaknya ukuran *size batch* yang telah ditentukan. Selanjutnya dilakukan tahap *testing* dengan melakukan deteksi terhadap citra biji kopi dengan 4 tingkat kematangan yang berbeda yang berada diluar *dataset* yang berfungsi untuk mendapatkan hasil performa dari setiap model dalam melakukan deteksi terhadap kematangan *roast* biji kopi.

3.3.13 Menampilkan hasil performa *training*

Untuk menentukan mAP (*Mean Average Precision*) atau parameter utama yang digunakan untuk mendapatkan nilai akurasi dari model yang telah dilakukan tahap *training*. mAP (*mean Average Precision*) melakukan perhitungan dengan membuat rata-rata nilai AP (*Average Precision*) pada setiap kelas yang berada pada model yang telah melakukan tahap *testing*. Untuk mengetahui nilai *accuracy*, *recall*, *precision*, dan *F1-score* maka dapat dicari menggunakan persamaan:

a. *Accuracy*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

Dengan

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

b. *Recall*

$$Recall = \frac{TP}{TP + FN} \quad (3.2)$$

Dengan

TP = *True Positive*

FN = *False Negative*

c. *Precision*

$$Precision = \frac{TP}{TP + FP} \quad (3.3)$$

Dengan

TP = *True Positive*

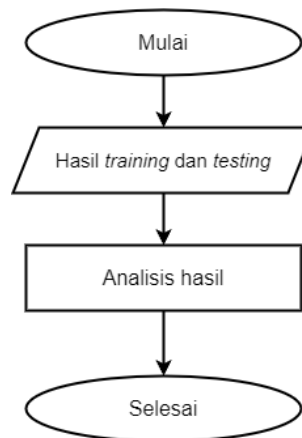
FP = *False Positive*

d. *F1-measure*

$$F1 - measure = \frac{2(Precision * Recall)}{Precision + Recall} \quad (3.4)$$

Tetapi pada tahap ini, semua persamaan tersebut akan otomatis dilakukan dan akan menemukan hasilnya secara mandiri menggunakan sebuah perintah untuk menampilkan log model pada algoritma YOLOv4. Selain *mean Average Precision* (mAP) dan *loss*, salah satu nilai persentase akan ditampilkan juga berupa *Intersection over Union* (IoU) secara rata-rata.

3.3.14 Analisis Hasil



Gambar 3.16 Diagram alir analisis hasil

Setelah melakukan tahap *testing*, hasil akan didapan berupa akurasi algoritma YOLOv4 dalam melakukan deteksi dan identifikasi kelas terhadap citra biji kopi, kemudian hasil tersebut akan dibuat menjadi sebuah tabel. Kemudian pada tabel tersebut akan ditentukan, model mana yang memiliki akurasi yang paling tepat dalam melakukan deteksi dan identifikasi pada kematangan *roast* biji kopi, apabila didapatkan beberapa model yang memiliki hasil yang terbaik tetapi setara maka dilakukan lagi analisis dengan menggunakan tabel yang didapatkan dari hasil pada tahap *training*, dimana pada tabel tersebut memiliki data performa yang lebih rinci dan pastinya nilai akurasi dan *loss* pada setiap model tidak akan memiliki nilai yang sama.