

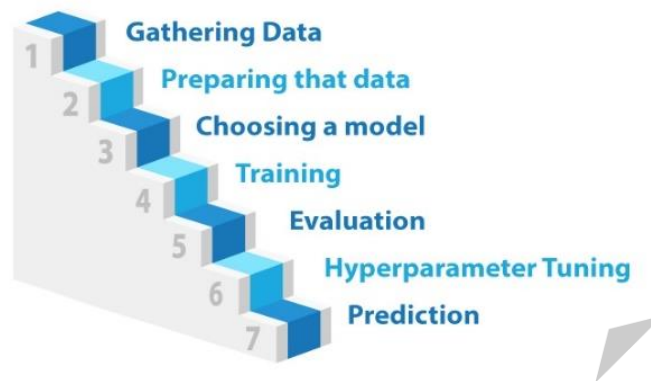
BAB 3

METODE PENELITIAN

3.1 PEMODELAN SISTEM

Pada perancangan sistem pendeteksi kesiapan panen tanaman Pakcoy digunakan bahasa pemrograman python dan java, data dilatih menggunakan *framework Tensorflow Lite* dengan arsitektur *model EfficienDet lite2*.

Model perancangan yang digunakan pada penelitian ini dijalankan pada *interpreter google colaboratory, roboflow* sebagai *database* dalam proses olah data *augmentation*, *android visual studio* sebagai *deployment* ke aplikasi. Semua aplikasi yang digunakan dalam perancangan ini berbasis aplikasi *web open-source* yang dapat digunakan untuk kode, persamaan, visualisasi dan teks naratif. Penelitian ini dilaksanakan mengacu pada alur *7 steps of Machine Learning* yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tampilan *7 steps of machine learning*

3.1.1 *Gathering Data*

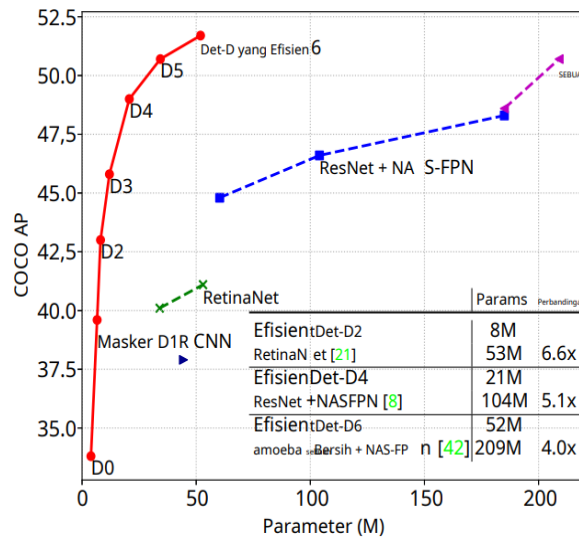
Tahap ini merupakan tahap pengumpulan data, tahap ini sangat penting karena kualitas dari data akan berpengaruh secara langsung terhadap model pendeteksian objek yang akan di buat. Pada penelitian ini terdapat 325 data gambar (*real time*) terdiri dari sayuran Pakcoy siap panen dan sayuran Pakcoy yang belum siap panen yang didapatkan dari memfoto secara langsung.

3.1.2 Preparing Data

Tahap ini merupakan tahap persiapan data, setelah data gambar di kumpulkan selanjutnya di lakukan pelabelan pada gambar menggunakan *tools* yang ada pada *website roboflow* dan kemudian dilakukan ekspor dataset berupa gambar yang telah di labeli tadi dengan menerapkan *data preprocessing* dan *augmentations* dengan pembagian dataset yaitu sebesar 60% untuk data latih, 20% untuk data validasi dan 20% untuk data uji.

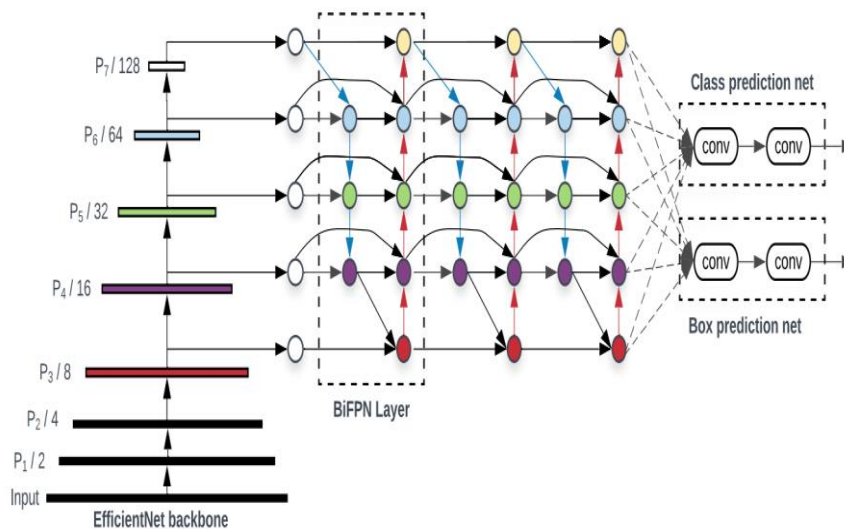
3.1.3 Choosing a Model

Langkah selanjutnya adalah pemilihan model, penelitian ini menggunakan arsitektur model *EfficientDet Lite 2*, model ini merupakan turunan dari model *EfficientDet* yang secara konsisten mencapai akurasi dan efisiensi yang lebih baik dari model-model sebelumnya (Tan et al., 2020, p.10788). *EfficientDet* memiliki delapan model yaitu D0-D7 yang mana akurasi dan kompleksitas waktu pada model meningkat seiring dengan ukuran model [27]. Perbandingan ukuran model dapat dilihat pada Gambar 3.2



Gambar 3.2 Perbandingan Ukuran Model[27]

Selanjutnya pada Gambar 3.3 menunjukkan gambaran dari arsitektur *model EfficientDet*.



Gambar 3.3 Arsitektur Model *EfficientDet* [27]

Model EfficientDet menggunakan *model EfficientNet* sebagai jaringan tulang punggung, BiFPN sebagai jaringan fitur, dan jaringan prediksi kelas atau kotak bersama. Lapisan BiFPN dan lapisan jaring kelas/kotak diulang beberapa kali berdasarkan batasan sumber daya yang berbeda [27].

Model EfficientDet menggunakan *model EfficientNet* sebagai jaringan tulang punggung, BiFPN sebagai jaringan fitur, dan jaringan prediksi kelas atau kotak bersama. Lapisan BiFPN dan lapisan jaring kelas atau kotak diulang beberapa kali berdasarkan batasan sumber daya yang berbeda [27]

3.1.4 Training

Selanjutnya masuk ke proses pelatihan, pelatihan dilakukan di Colab Notebooks menggunakan *framework Tensorflow Lite*, *batch size* di atur ke 8 dan menggunakan 100 *epoch* untuk proses pelatihan, dari konfigurasi ini pemrosesan per *epoch* rata-rata memakan waktu sebanyak 24 detik, sehingga proses pelatihan model keseluruhan memakan waktu kurang lebih sekitar 40 menit.

3.1.5 Evaluation

Setelah pelatihan selesai, selanjutnya dilakukan evaluasi menggunakan data validasi yang telah dipisahkan sebelumnya untuk melihat apakah modelnya sudah bagus, pada tahap evaluasi ini model pendeteksi objek yang telah dilatih akan diuji

menggunakan data gambar yang belum pernah dilihat sebelumnya, di tahap ini kita dapat melihat kinerja model sebelum di terjunkan langsung di dunia nyata.

3.1.6 *Hyperparameter Tuning*

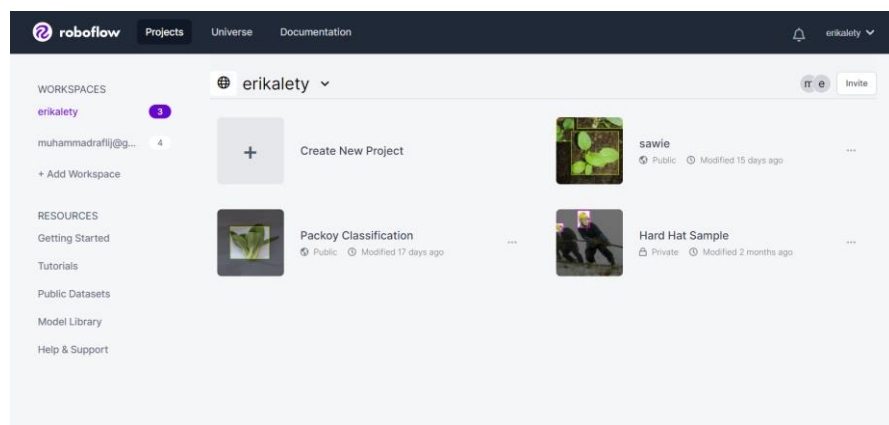
Hyperparameter Tuning dilakukan untuk lebih meningkatkan lagi akurasi dari model setelah dilakukan pelatihan dengan menyetel beberapa parameter, pada penelitian ini tidak di lakukan penyetelan ulang pada parameter karena akurasi yang di dapatkan usai pelatihan di rasa sudah cukup baik.

3.1.7 *Prediction*

Setelah melalui berbagai macam proses, tahap ini adalah inti dari semua proses yang telah di lalui, di mana nilai pembelajaran mesin di realisasikan untuk mendeteksi tanaman Pakcoy siap panen dan Pakcoy belum siap panen, model yang telah selesai dilatih kemudian di *deploy* ke perangkat seluler android untuk secara nyata melihat akurasi dari prediksi model saat diterapkan pada perangkat dan agar dapat digunakan oleh masyarakat.

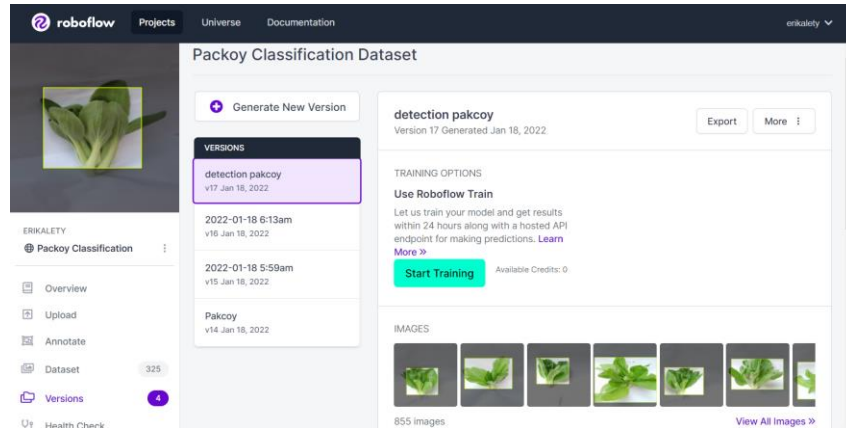
3.2 ROBOFLOW

Database yang digunakan untuk komunikasi pengolahan citra dengan *smartphone* pada sistem ini yaitu Roboflow. Konsep yang digunakan pada *database* ini adalah *Real-time*. *Platform Roboflow* ini mempermudah pada pembuatan sistem tugas akhir ini dalam hal penyimpanan data. Penggunaan *Roboflow* ini juga nantinya dapat terintergrasikan langsung dengan *TF Lite App Inventor*.



Gambar 3.4 Tampilan Awal Roboflow

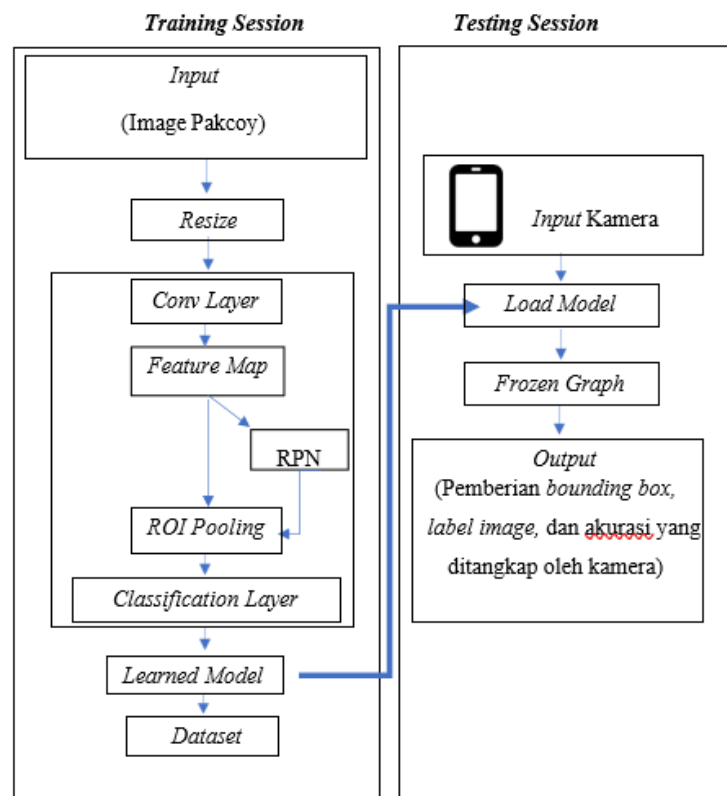
Merupakan tampilan awal dari Roboflow. Untuk melakukan proses pembuatan *database* dapat dengan klik mulai atau jika Roboflow sudah *login* bisa klik pada bagian *project*.



Gambar 3. 5 Tampilan Pembuatan *Project Roboflow*

3.3 ARSITEKTUR UMUM

Adapun arsitektur umum dari aplikasi ini dapat dilihat pada Gambar



Gambar 3.6 Tampilan Arsitektur Umum

Keterangan Arsitektur Umum (Gambar 3.6) adalah :

1. *Training Session*

a) *Input:*

Pada tahap ini *input* diberikan citra berupa tanaman Pakcoy.

b) *Resize:*

Pada tahap awal proses *preprocessing* yaitu dilakukan *resizing* untuk mengubah ukuran citra dengan memperkecil ukuran citra secara *horizontal* dan atau *vertical*, gambar akan di-*resize* agar ukuran file tidak lebih 200 KB dan dimensi *image* tidak lebih dari 720 x 1280.

c) *Convolutional Layer:*

Layer yang terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dua dimensi dengan panjang dan tinggi (*pixels*).

d) *Feature Map:*

Feature Map adalah sebuah map yang dibuat oleh *convolutional layer* yang berisi informasi tentang representasi *vector* dari *image* yang ditangkap, *convolution layer* akan menghasilkan dua *feature map* yang sama, *feature map* pertama akan diolah di RPN untuk menghasilkan *region proposal* dan *feature map* lainnya akan langsung dikirim ke *pooling layer*.

e) RPN (*Region Proposal Network*):

RPN merupakan sebuah modul yang bekerja untuk mengolah *feature map* yang telah dibuat pada *convolution layer* untuk memprediksi bagian yang dianggap sebagai objek dan melakukan prediksi *bounding box* dari objek tersebut,

f) ROI *Pooling*:

ROI merupakan *layer* yang bertanggung jawab untuk menyamakan ukuran dari *feature map* dan *region proposal* yang telah diolah oleh RPN serta mengirim informasi *feature map* dan *proposal* untuk diklasifikasi pada *classification layer*.

g) *Classification Layer:*

Classification layer berfungsi untuk mengelompokkan objek yang telah dideteksi pada RPN dan melakukan pelabelan terhadap objek tersebut serta memberikan *bounding box* pada objek tersebut.

h) *Learned Model*:

Learned model adalah model yang telah selesai dipelajari oleh sistem.

i) *Dataset*:

Kumpulan dari *learned* model lain yang tidak dipakai dalam pengujian yang berisi informasi bobot *dataset*.

2. *Testing Session*

a) *Input*:

Pada tahap ini, sistem diberikan *input* data berupa citra tanaman Pakcoy.

b) *Load Model*:

Load model adalah tahap dimana sistem akan memuat kembali model yang telah disimpan pada masa *training session*.

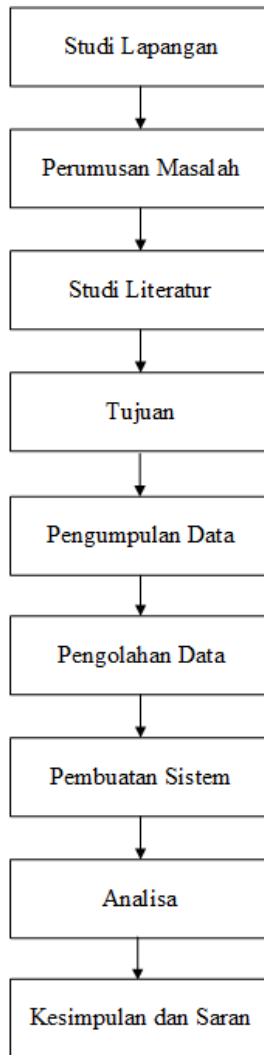
c) *Frozen Graph*:

Pada tahap ini data inputan yang diterima melalui kamera akan diproses pada *graph* yang tersimpan pada *frozen* model untuk dilakukan identifikasi dan pemberian *bounding box* berdasarkan *weight* yang telah tersimpan pada model yang telah dilatih.

d) *Output*:

Output merupakan hasil dari identifikasi yang telah dilakukan, dimana objek yang telah diklasifikasi diberikan *bounding box* dan label.

3.4 ALUR PENELITIAN



Gambar 3.7 Diagram Alur Penelitian

3.4.1 Studi Lapangan

Salah satu cara untuk memperoleh data dengan melakukan pengamatan langsung terhadap objek yang akan diteliti untuk mendapatkan informasi mengenai

3.4.2 Perumusan Masalah

Perumusan masalah dibuat dengan mempertimbangkan beberapa hal yaitu dari segi metode, hasil penelitian, dan analisis penelitian yang akan dilakukan. Tujuan dibuatnya perumusan masalah dalam skripsi ini yaitu untuk mengetahui bagaimana sistem yang seharusnya dibuat untuk mendeteksi kesiapan matang tanaman Pakcoy.

3.4.3 Studi Literatur

Cara yang digunakan untuk mencari sumber pustaka dengan pembahasan mengenai tanaman Pakcoy, ekstraksi ciri, metode *EfficienDet Lite 2*. Studi literatur dibutuhkan sebagai referensi dalam mendapatkan gambaran dan informasi terkait dengan penelitian sebelumnya.

3.4.4 Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah menggunakan teknik pengumpulan data dengan cara mengambil data foto secara langsung (*lifetime*) serta menggunakan augmentasi data.

3.4.5 Pengolahan Data

Setelah data dikumpulkan, selanjutnya dilakukan pengolahan data. Data antara tanaman Pakcoy sudah matang dan tanaman Pakcoy yang belum matang dijadikan dalam satu *database*.

Database tersebut kemudian diolah dengan menggunakan roboflow untuk dicari ekstraksi fitur tanaman Pakcoy. Ekstraksi fitur ini digunakan untuk pembandingan antar citra tanaman Pakcoy yang akan dihitung berapa tingkat akurasi kesiapan panen pada tanaman tersebut. Dengan penambahan metode augmentasi pada citra tanaman Pakcoy maka dapat menentukan kesiapan panen dari tanaman Pakcoy lebih tinggi.

3.4.6 Pembuatan Sistem

Pada tahap ini, pembuatan sistem yang akan dibuat merupakan sistem yang digunakan untuk mendeteksi kesiapan panen pada tanaman Pakcoy dengan menggunakan indikator warna kehijauan pada tanaman tersebut. Pada sistem ini akan menggunakan fitur ekstraksi ciri untuk mengukur intensitas akurasi warna kehijauan dengan menggunakan metode *Tensorflow Lite* dan pemodelan *EfficienDet Lite 2*.

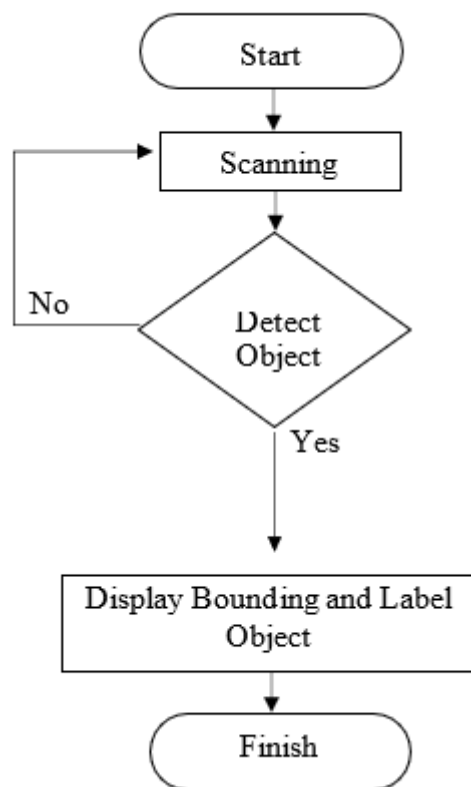
3.4.7 Analisis

Analisis data dihasilkan dari tahapan pengumpulan data kemudian masuk kedalam sistem lalu diuji untuk tingkat kematangan tanaman Pakcoy hingga tanaman tersebut sudah dapat dipanen.

3.4.8 Kesimpulan dan Saran

Tahap selanjutnya adalah menyimpulkan terhadap penggunaan fitur ekstraksi ciri menggunakan *framework Tensorflow Lite* dengan arsitektur model *EfficienDet Lite 2* yang menunjukkan bahwa tanaman Pakcoy sudah siap panen atau belum.

3.5 PERANCANGAN SISTEM



Gambar 3.8 *Flowchart* Sistem

Flowchart pada Gambar 3.8 menjelaskan secara detail proses deteksi objek. Pertama harus melakukan eksekusi program, lalu program akan melakukan inisialisasi atau pengecekan terhadap komponen-komponen perangkat pendukung

sistem maupun kode program. Jika terdapat *error* maka program akan dihentikan, namun jika tidak terdeteksi *error* maka program akan mengaktifkan perangkat kamera.

Perangkat kamera yang telah aktif lalu mengambil video secara *real time* dan kemudian sistem membaca setiap *frame* pada video tersebut. *Frame* lalu diolah menggunakan metode SSD, jika terdapat objek manusia, maka hasil deteksi akan ditampilkan ke *display* lalu disimpan kedalam media penyimpanan. Jika tidak ada objek manusia yang terdeteksi, maka program akan memproses *frame* selanjutnya.

3.6 CONFUSION MATRIX

Confusion matrix merupakan pengukuran yang dapat digunakan untuk menghitung kinerja atau tingkat kebenaran pada proses klasifikasi. Tabel *confusion matrix* ditunjukkan pada Tabel 3.1 berikut.

Tabel 3. 1 *Confusion Matrix*

N = 200		<i>Actual Values</i>	
		<i>Positive (1)</i>	<i>Negative (0)</i>
<i>Predicted Values</i>	<i>True (1)</i>	TP	FP
	<i>False (0)</i>	FN	TN

Keterangan:

- TP (*True Positive*) merupakan banyaknya data yang memiliki prediksi matang kemudian terbaca oleh sistem sebagai matang.
- FP (*False Positive*) merupakan banyaknya data yang memiliki prediksi matang kemudian terbaca oleh sistem sebagai belum matang.
- FN (*False negative*) merupakan banyaknya data yang memiliki prediksi belum matang kemudian terbaca oleh sistem sebagai belum matang.
- TN (*True Negative*) merupakan banyaknya data yang memiliki prediksi belum matang kemudian terbaca oleh sistem sebagai matang.