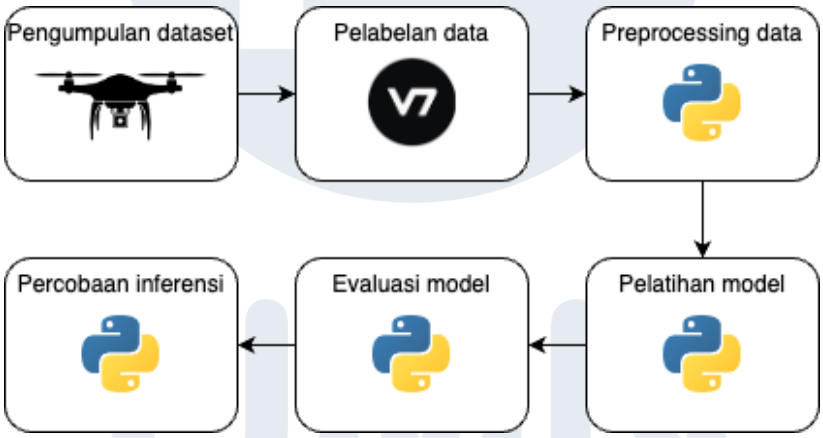
BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Penelitian ini dilakukan melalui beberapa tahap, sesuai dengan diagram pada gambar 3.1. Pertama, dilakukan pengumpulan dataset menggunakan *drone*, lalu seluruh data yang didapatkan dilabel secara manual, kemudian dilakukan *preprocessing* data untuk menyesuaikan dataset dengan ketentuan arsitektur model yang digunakan, setelah itu, proses pelatihan model dilakukan, dan seluruh model yang dihasilkan akan dievaluasi dan dibandingkan untuk menentukan yang terbaik. Model terbaik yang didapatkan akan digunakan dalam pengujian inferensi yang merupakan tahap terakhir dari keseluruhan penelitian.



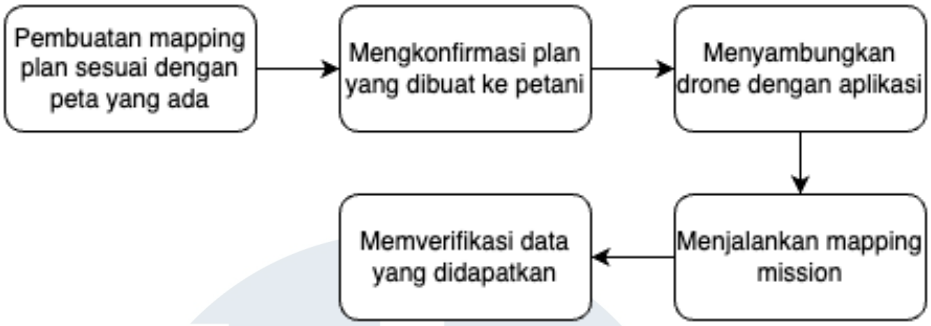
Gambar 3.1 Alur penelitian

3.2 Perancangan Modul

3.2.1 Pengumpulan Dataset

Pengambilan data dilakukan menggunakan 2 jenis *drone*, yaitu DJI Air 2S yang beresolusi 5472×3078 piksel [17], dan DJI Mini 2 dengan resolusi 4000×2250 piksel [18]. Kedua *drone* tersebut dikendalikan menggunakan aplikasi DroneLink [19], dengan memanfaatkan fitur *mapping* dari aplikasi tersebut. Seluruh data yang diambil merupakan *aerial image* dari beberapa kebun yang merupakan bagian dari

Paguyuban Mitra Turindo. Alur pengumpulan dataset dapat dilihat pada gambar 3.2.



Gambar 3.2 Alur pengumpulan dataset

Pertama, *mapping plan* akan dibuat pada aplikasi DroneLink berdasarkan peta kelompok tani yang ada. Kemudian, *mapping plan* tersebut akan diverifikasi oleh petani untuk memastikan apakah kebun-kebun yang termasuk ke dalam peta merupakan kebun yang teregistrasi, atau apakah ada kebun yang belum tercakup. Jika mapping plan sudah benar, *drone* akan dinyalakan, dan smartphone yang memiliki aplikasi DroneLink akan disambungkan dengan *controller drone* tersebut. Setelah *drone* tersambung dengan aplikasi, maka *mapping mission* akan dijalankan, dan jalannya misi akan diawasi untuk mengatasi *error* yang terjadi (jika ada). Setelah misi selesai, data yang didapatkan akan dipindahkan ke laptop, dan hasil tangkapan kebun akan diverifikasi.

3.2.2 Pelabelan Data

Dalam melakukan pelabelan data, digunakan *webapp* Darwin by V7Labs [20]. Alur kerja pada tahap ini cukup mudah, yaitu cukup dengan mengunggah dataset, lalu melakukan pelabelan, dan mengekspor dataset yang telah dilabel. Sebelum memulai proses pelabelan, kelas salak atau *snakefruit* ditentukan terlebih dahulu pada *workspace,* untuk menandakan bagian yang akan dianotasi nantinya. Kemudian, semua foto yang telah didapatkan menggunakan *drone* diunggah ke *workspace*. Setelah semua gambar telah terunggah, gambar dapat dianotasi menggunakan berbagai *tool* yang ada. Tersedia juga 4 jenis status untuk

menandai setiap data, yaitu *new*, *being annotated*, *in review*, dan *complete*. Setelah semua data telah selesai dilabel, maka seluruh hasil anotasi dapat diekspor.

Pada proses anotasi, terdapat 5 *tool* yang digunakan, yaitu:

- *brush tool*, untuk menganotasi gambar dengan *brush* yang berbentuk lingkaran dan ukuran yang dapat disesuaikan
- *eraser tool*, yang dapat digunakan untuk menghapus anotasi, dengan cara penggunaan yang sama seperti *brush tool*
- *polygon tool*, yang digunakan untuk menganotasi bagian-bagian detil dengan cara menggunakan berbagai titik yang akan membentuk garis secara otomatis, dan akan menjadi sebuah bentuk berupa anotasi ketika semua garis telah tertutup
- *auto-annotation*, yang menggunakan model SAM untuk menganotasi bagian gambar sesuai dengan karakteristik yang mirip
- *caption tool*, untuk memberikan komentar pada gambar, yang umumnya digunakan ketika tidak ada kebun salak pada data.

Contoh gambar asli dan gambar yang telah dianotasi dapat dilihat pada gambar 3.3 dan gambar 3.4.

Gambar 3.3 Gambar sebelum dianotasi

Gambar 3.4 Gambar setelah dianotasi

3.2.3 *Preprocessing* Data

Preprocessing data yang dilakukan meliputi *patch crop* menggunakan *library* patchify [21]. Semua hasil tangkapan kawasan kebun salak dan hasil anotasinya akan di-*crop* ke dalam beberapa foto berbeda dengan pengaturan ukuran 256×256 piksel dan steps 200 piksel, untuk menjamin tidak ada informasi atau area yang hilang dari gambar aslinya. Hal ini dilakukan karena ukuran resolusi asli file-file yang ada pada dataset mayoritas berukuran 5472×3648 piksel, dengan sebagian lainnya yang berukuran 4000×3000 piksel, membuat proses training menjadi sangat berat dan membuthkan resource serta waktu yang lama. Selain itu, penggunaan ukuran 256×256 piksel juga telah terbukti memiliki performa yang lebih baik menurut penelitian yang dilakukan oleh Chuang Yu, et al [9].

3.2.4 Pelatihan Model

Pelatihan model dilakukan menggunakan layanan Google Colab Pro+, dengan *hyperparameter* yang sama untuk setiap *backbone*-nya. Untuk dataset yang digunakan terdiri dari gambar hasil tangkapan *drone* dan gambar yang berisi anotasi tiap gambar, yang masing-masing berjumlah 372.000 gambar setelah melalui proses *patch crop*. *Dataset split* yang digunakan terdiri dari 70% *training*, 15% *validation*, dan 15%

testing. Dataset yang telah dipisahkan menjadi *train dataset*, *val dataset*, dan *test dataset* tersebut kemudian dimasukkan ke DataLoader untuk digunakan pada proses *training*, *validasi*, dan *testing* nantinya.

Pelatihan model dilakukan dengan basis PyTorch, lebih tepatnya menggunakan library segmentation_models.pytorch atau smp [22], dengan pilihan backbone berupa EfficientNetB0, Xception, dan MobileNetV2. Detail *hyperparameter* yang digunakan dapat dilihat pada tabel 3.1.

Tabel 3.1 *Hyperparameter* pelatihan model

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.0001
Criterion	BCEWithLogitsLoss
Batch Size	4
Epoch	10
Weights	ImageNet & None
Threshold	0.2

Untuk fungsi *callback* early_stopping, metrik val_loss dijadikan acuan dalam monitoring dengan *patience* 3. Hal ini berarti, jika dalam 3 *epoch* terakhir nilai val_loss tidak meningkat, maka proses *training* akan dihentikan.

Dalam proses *training,* informasi *train loss*, *validation loss*, *precision*, *recall*, IoU, dan F1 *Score* ditampilkan pada akhir setiap *epoch*-nya. *Epoch* dengan *validation loss* terbaik (terkecil) akan disimpan dengan format .pth. Semua informasi yang ditampilkan juga disimpan ke dalam sebuah variabel dan hasilnya akan divisualisasikan pada tahap evaluasi model.

3.2.5 Evaluasi Model

Untuk mengevaluasi hasil pelatihan seluruh model, akan digunakan beberapa pengujian berbeda, yang meliputi:

- *Learning Curve*

  Grafik *learning curve* yang didapatkan dari hasil pelatihan sebuah model menunjukkan *loss* dari setiap *epoch*. Sumbu x mengindikasikan *epoch*, dan sumbu y menunjukkan nilai *loss*. Baik hasil pelatihan maupun hasil validasi tergambarkan melalui grafik ini, yang dapat digunakan untuk menentukan apakah model berstatus *overfitting*, yang terjadi ketika training *loss* bernilai kecil, tetapi dengan *validation loss* yang besar; *underfitting*, yang merupakan kebalikan dari *overfitting*; atau *good fit*, yang memiliki *validation loss* dan *training loss* kecil, dan berarti model yang dilatih cocok untuk dataset yang digunakan.

- *Evaluation Metrics*

  Sebelum memasuki evaluation metrics yang digunakan dalam penelitian ini, perlu diketahui bahwa ada 4 nilai yang berperan sebagai basis perhitungannya, yaitu:

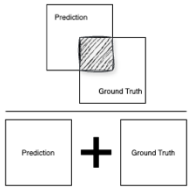
  - TP (*True Positive*), yang merupakan jumlah piksel yang berhasil diprediksi sesuai dengan ground truth
  - TN (*True Negative*), yang merupakan jumlah piksel *background* yang berhasil diprediksi
  - FP (*False Positive*), yang merupakan jumlah piksel yang seharusnya merupakan *background* tetapi diprediksi sebagai objek
  - FN (*False Negative*), yang merupakan jumlah piksel yang seharusnya merupakan bagian dari objek tetapi diprediksi sebagai *background*

  Dalam pengevaluasian performa model terhadap *test dataset*, ada 4 jenis *evaluation metrics* yang digunakan, meliputi:

  - F1 *score* yang disebut juga sebagai *Dice Coefficient*, menggunakan hasil *precision* dan *recall* dan memperhitungkan *harmonic mean* dari keduanya dengan nilai yang berkisar dari 0

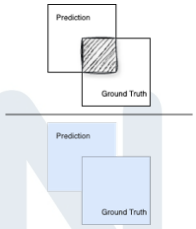
hingga 1. Semakin mendekati 1, maka rasio *precision* dan *recall* dari model tersebut semakin baik, dengan artian bahwa model tersebut memiliki ketelitian yang tinggi dalam pelabelan, karena kemungkinan terjadinya *false positive* ataupun *false negative* sangat rendah atau bahkan tidak ada sama sekali.

$$F1\ score = \frac{2 \times area\ of\ overlap}{Total\ area} = \frac{2 \times}{} \quad (3.1)$$



- IoU (*Intersection over Union*) atau Jaccard Index, mengukur sebanyak apa bagian gambar yang beririsan antara hasil prediksi model dan *ground truth* yang ada, dengan kisaran nilai dari 0 hingga 1. Semakin mendekati 1, maka hasil prediksi model semakin mendekati *ground truth*, atau posisi hasil segmentasinya mendekati daerah yang sebenarnya, atau bahkan sama persis jika nilainya berupa 1. Dengan kata lain, hasil segmentasi menjadi lebih dapat dipercaya jika nilainya semakin mendekati 1.

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} = \quad (3.2)$$



- *Precision*, merupakan perbandingan hasil prediksi TP dibandingkan dengan semua hasil positif. Nilai *precision* yang tinggi berarti tingkat *false positive* dari model yang dihasilkan rendah, atau model tidak melabeli piksel yang bukan bagian dari suatu kelas, dalam kasus ini pohon salak, menjadi objek tersebut.

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

- *Recall* atau yang disebut juga *sensitivity*, merupakan perbandingan dari hasil TP dibandingkan dengan seluruh

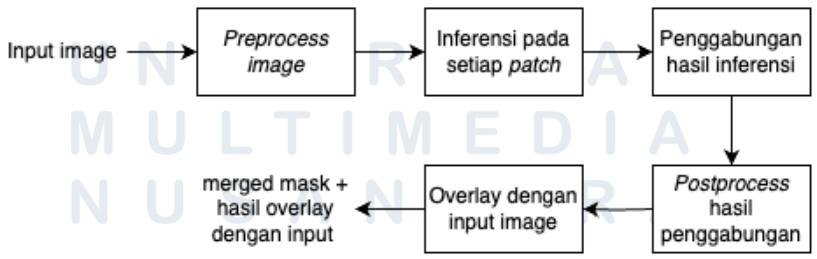
sampel yang seharusnya bernilai positif. Nilai *recall* yang tinggi menandakan bahwa model memiliki tingkat *false negative* yang rendah, atau model tidak melabeli piksel yang seharusnya merupakan bagian dari suatu kelas menjadi bagian dari kelas lain. Dalam kasus ini, melabeli pohon salak menjadi bukan pohon salak [23][24][25].

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

3.2.2 Pengujian Inferensi pada Sampel Gambar

Pada tahap ini, model dengan *validation loss* terbaik yang telah tersimpan pada proses *training* kembali di-*load* dan diubah ke mode validation. Kemudian, akan dilakukan 2 jenis inferensi, yaitu inferensi terhadap gambar yang ada pada test dataset, dan terhadap gambar hasil tangkapan *drone*. Hasil inferensi terhadap test dataset akan ditampilkan secara acak sebanyak 10 gambar, dan dibandingkan dengan ground truth yang ada.

Sedangkan hasil inferensi terhadap foto tangkapan drone akan ditampilkan berupa gambar asli yang di-*overlay* dengan hasil prediksi yang didapatkan. Dalam melakukan inferensi terhadap gambar tangkapan drone yang beresolusi tinggi, dilakukan 5 macam proses dalam keseluruhan proses inferensi, seperti yang terlihat pada gambar 3.5.



Gambar 3.5 Proses inferensi gambar beresolusi tinggi

Dalam menjelaskan proses inferensi ini, akan digunakan istilah *input* untuk gambar asli, *patch* untuk *input* yang telah dipotong, *prediction*

untuk hasil prediksi pada *patch*, *merged* untuk hasil *prediction* yang telah disatukan dan dikembalikan ke dimensi aslinya, dan *overlayed* yang merupakan hasil penggabungan *input* dengan *merged*.

Pertama, dilakukan *preprocessing* pada *input*, yang dimulai dengan melakukan *resize* pada gambar tersebut agar rasionya menjadi 1:1, kemudian melakukan *patch crop* agar gambar tersebut terbagi ke sejumlah *patch* berukuran 256×256 piksel, tanpa adanya overlap. Urutan setiap *patch* disimpan ke dalam sebuah *array*. Lalu, dilakukan inferensi terhadap setiap *patch* yang dihasilkan yang akan menghasilkan sejumlah *prediction*. Seluruh *prediction* yang dihasilkan kemudian disatukan sesuai dengan urutan yang telah tersimpan, dan hasil penggabungan yang didapatkan di-*resize* kembali ke ukuran aslinya. Hasil akhir proses ini merupakan *merged* dan *overlayed* yang keduanya disimpan dalam format .jpg.