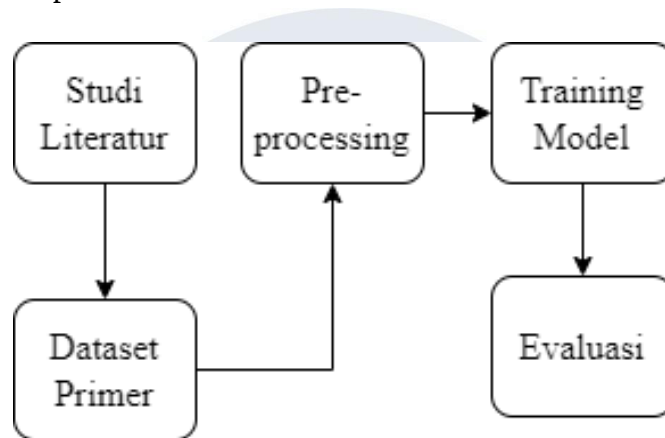


BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Metode penelitian yang akan dilakukan dapat diuraikan secara berikut, melakukan studi literatur, pembuatan dataset primer, *preprocessing*, pelatihan model, dan evaluasi performa model.



Gambar 3.1 Alur penelitian

3.2 Dataset

Pengumpulan data untuk *dataset* primer dilakukan menggunakan dua buah drone, *brand* drone yang digunakan merupakan DJI dengan model AIR 2S seperti dan model MINI 2

Data yang diambil oleh drone Air 2S memiliki resolusi sebesar 5472 x 3648 *pixel*, sedangkan data yang diambil oleh drone Mini 2 memiliki resolusi sebesar 4000 x 3000 *pixel*. walaupun kedua data memiliki resolusi yang berbeda, kedua data tetap dapat digunakan sebagai *dataset* segmentasi karena akan di *crop* menjadi ukuran yang sama.

Dikarenakan wilayah yang akan diambil datanya sangat luas, aplikasi DroneLink akan digunakan untuk membantu proses pengambilan datanya. Aplikasi DroneLink membantu pengambilan data dengan mengotomatis penerbangan dan foto drone.



Gambar 3.2 Hasil foto drone

Selama survei lapangan, diperoleh sebanyak 4.987 foto kebun, yang dapat dilihat pada Gambar 3.2. Foto-foto tersebut diambil dari kebun salak milik kelompok tani Sedyo Makmur dan Muda Jaya dari Paguyuban Mitra Turindo.

3.3 Pre-procesing

Pembuatan model segmentasi membutuhkan dataset yang terdiri dari foto dan masking. Dalam penelitian ini, dataset dibuat dengan menggunakan foto yang telah diperoleh sebelumnya dan menggunakan website Darwin V7 untuk proses masking yang menghasilkan gambar 3.3. penulis telah selesai meng-*annotate* 473 foto yang akan dijadikan dataset.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.3 Hasil masking data

Gambar yang didapatkan memiliki resolusi yang terlalu besar untuk digunakan *training* secara langsung, dengan itu digunakan *library patchify* yang dapat membagi gambar menjadi *patch* kecil berdasarkan ukuran sel *patch* yang diberikan, dan menggabungkan *patch* ke dalam gambar asli.

Dengan *library patchify*, gambar akan dipotong dari ukuran awal sebesar 4000x3000 *pixel* atau 5472x3648 *pixel* menjadi 256x256 *pixel*. Ukuran 256x256 *pixel* dipilih karena ukuran tersebut adalah ukuran yang dapat diproses oleh arsitektur DeepLabV3+ dan menurunkan kekuatan komputasi yang diperlukan untuk melakukan pelatihan model secara signifikan.

3.4 Proses pelatihan

Pelatihan model akan dilakukan menggunakan perangkat lunak Visual Studio Code. Model DeepLabV3+ dengan *backbone Xception*, *ResNet101*, dan *EfficientNetB3* akan digunakan untuk tugas *semantic segmentation*. Pembuatan model dengan masing masing *backbone* akan dilakukan dengan *hyperparameter* dan kode yang sama untuk memastikan perbedaan hanya dari *backbone*. Untuk fase pelatihan, model akan menggunakan 1000 gambar dan *mask* yang telah di

preproces menjadi ukuran 256x256 *pixel* dan dibagi menjadi tiga bagian, yaitu *training*, *validation*, dan *testing* dengan rasio 70%/15%/15%. Pelatihan model dilakukan dengan TensorFlow dengan menggunakan *hyperparameter* yang bisa dilihat pada tabel 3.1.

Tabel 3.1 Hyperparameter pelatihan model

Hyperparameter	
Optimizer	Adam
Learning Rate (lr)	0.001
Batch Size	4
Epoch	80
Weights	Imagenet

Adam dipilih karena efisien dan populer untuk tugas *deep learning*. Learning rate 0,001 dipilih karena memberikan keseimbangan antara kecepatan konvergensi dan stabilitas pelatihan. Batch size kecil digunakan karena keterbatasan daya komputasi, seperti memori GPU. Epoch sebanyak 80 dipilih untuk memberikan waktu pelatihan yang cukup agar model mencapai status *goodfit*. ImageNet digunakan untuk mempercepat konvergensi dan meningkatkan akurasi, terutama pada dataset berukuran terbatas, dengan penyesuaian sesuai karakteristik dataset pohon salak.

3.5 Evaluasi

Untuk menentukan apakah model memiliki performa yang baik, akan digunakan beragam metrik, yang meliputi:

3.5.1 *Loss curve*

Loss curve adalah grafik yang menunjukkan relasi antara epoch dan loss model, yang digunakan untuk memantau performa model selama proses *training*. *Loss* merupakan nilai hasil rumus yang mengalkulasikan perbedaan antara prediksi model dengan ground truth-nya. *Loss curve* terdiri atas dua jenis *loss*, yaitu *training loss* dan *validation loss*. Dengan menganalisis nilai kedua jenis *loss* pada setiap epoch, performa model dapat dipantau selama pelatihan. Jika nilai *training loss* jauh lebih kecil daripada

validation loss, model dinyatakan *overfitting*, yang berarti model terlalu menyesuaikan diri dengan data pelatihan dan kurang mampu melakukan generalisasi. Sebaliknya, jika *validation loss* jauh lebih kecil daripada *training loss*, model dinyatakan *underfitting*, yang menunjukkan bahwa model tidak belajar dengan baik dari data pelatihan. Sementara itu, jika kedua nilai *training loss* dan *validation loss* memiliki nilai yang seimbang, model dinyatakan *good fit*, yaitu kondisi di mana model memiliki kemampuan generalisasi yang baik.

3.5.2 Matrik pengujian

a) Intersection Over Union

Intersection Over Union (IoU) mengukur tingkat kesamaan antara area yang diprediksi oleh model dengan area sebenarnya. IoU dihitung dengan membagi luas area *intersection* antara prediksi dan *ground truth* dengan luas area *union* dari keduanya. Nilai yang dihasilkan dari rumus ini berkisar antara 0 hingga 1, semakin dekat nilai dengan 1 maka semakin mirip area prediksi dengan area *ground truth*, dengan kata lain, tingginya nilai IoU menandakan bahwa model dapat dengan akurat memprediksi area kebun salak yang benar.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Gambar 3.2 Rumus IoU

b) Precision

Precision Mengukur berapa banyak dari prediksi positif yang benar-benar positif. *Precision* dihitung sebagai rasio antara

jumlah prediksi positif yang benar (*true positive*, *TP*) dengan total jumlah prediksi positif dan prediksi yang salah (*false positive*, *FP*). *Precision* mendapatkan nilai *TP* dan *FP* dari hasil *IoU*, di mana nilai *TP* ditentukan jika nilai *IoU* sebuah gambar lebih besar dari *threshold* yang di set sebelumnya dan nilai *FP* ditentukan jika nilai *IoU* sebuah gambar lebih kecil dari nilai *threshold* yang di set. Nilai yang dihasilkan dari rumus ini berkisar antara 0 hingga 1, semakin dekat nilai dengan 1 maka menandakan banyak prediksi model yang benar, dengan kata lain, tingginya nilai *precision* menandakan model dapat lebih akurat dalam mengidentifikasi area kebun salak dibandingkan dengan area non-kebun, sehingga meminimalkan kesalahan prediksi.

$$Precision = \frac{TP}{TP + FP}$$

c) *Recall*

Recall Mengukur seberapa baik model dalam menangkap semua contoh positif yang sebenarnya. *Recall* dihitung sebagai rasio antara jumlah prediksi positif yang benar (*true positive*, *TP*) dengan total jumlah kasus positif sebenarnya, termasuk yang tidak terdeteksi dengan benar (*false negative*, *FN*). *Recall* mendapatkan nilai *TP* dan *FN* dari hasil *IoU*, di mana nilai *TP* ditentukan jika nilai *IoU* sebuah gambar lebih besar dari *threshold* yang di set sebelumnya dan nilai *FN* ditentukan jika gambar tidak ada prediced masknya atau memiliki nilai *IoU* 0. Nilai yang dihasilkan dari rumus ini berkisar antara 0 hingga 1, semakin dekat nilai dengan 1 maka menandakan model berhasil menangkap semua contoh positif yang sebenarnya, dengan kata lain, tingginya nilai *recall* menandakan model lebih sering memberikan prediksi area kebun salak yang benar dan sesuai dengan *ground truth* dari semua prediksi benar yang diberikan.

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

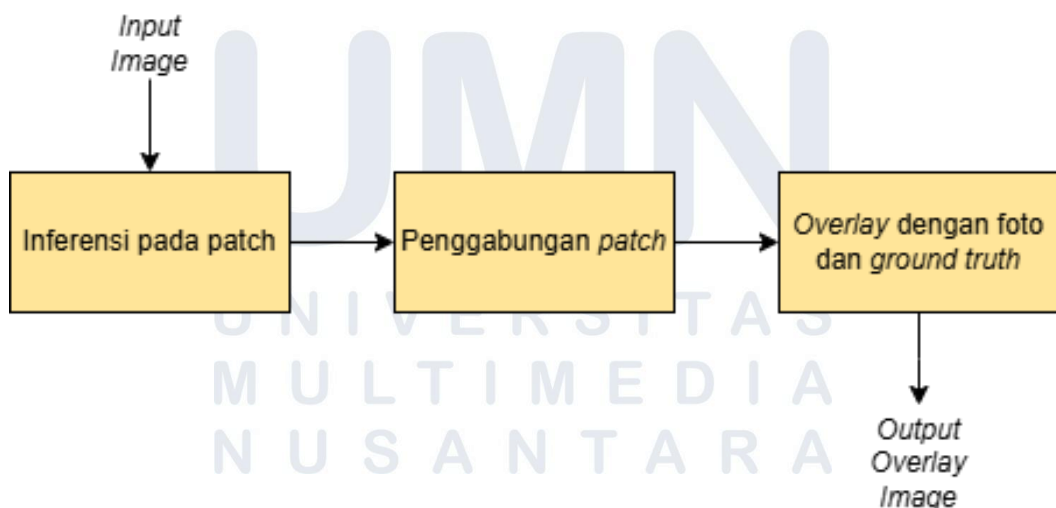
d) F1-Score.

F1-score mengukur keseimbangan antara *Precision* dan *Recall*. F1-score dihitung sebagai rasio keseimbangan antara *Recall* dengan *Precision* model, Nilai yang dihasilkan dari rumus ini berkisar antara 0 hingga 1, semakin dekat nilai dengan 1 maka menandakan kesempurnaan hasil prediksi dengan *ground truth*, dengan kata lain, tingginya nilai F1-score menunjukkan bahwa model tidak hanya mampu mengidentifikasi area kebun salak dengan akurat (*precision* tinggi) tetapi juga dapat menangkap hampir semua area kebun yang ada (*recall* tinggi), sehingga tetap ada keseimbangan antara kedua metrik tersebut [16].

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

3.6 Inferensi

Setelah semua model telah di uji, model dengan metrik terbaik akan digunakan untuk melakukan tes inferensi. Tes inferensi ini akan memprediksi masking data baru yang akan dibandingkan dengan *ground truth*, *masking* tersebut akan di *overlay* dengan data baru sebelumnya.



Gambar 3.4 Alur Inferensi

Pertama, foto dari dataset testing yang sudah dipotong akan diprediksi *masking* kebun salaknya dengan model. Kemudian prediksi akan digabungkan menjadi

ukuran semulanya. Selanjutnya, masking hasil prediksi akan di-*overlay* pada foto asli untuk memvisualisasikan hasil segmentasi. Proses ini bertujuan untuk memberikan gambaran yang jelas mengenai area mana saja dalam gambar yang berhasil dikenali oleh model sebagai kebun salak. Selain itu, masking hasil prediksi juga akan dibandingkan dengan *ground truth* yang sudah tersedia. *Ground truth* adalah referensi anotasi manual atau data label yang menunjukkan area sebenarnya dari kebun salak dalam gambar.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA