BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian berfokus pada kondisi permukaan kerabang telur ayam ras, khususnya retakan kerabang untuk mendeteksi retakan pada kerabang telur ayam ras menggunakan algoritma *AdaBoost* berdasarkan fitur morfologi citra. Secara spesifik, objek penelitian dibagi menjadi dua kondisi, yaitu telur retak dan telur tidak retak. Telur retak yaitu telur yang memiliki pola retakan yang membentuk pola garis atau celah pada permukaan kerabang. Sedangkan telur tidak retak, yaitu citra telur yang tidak memiliki pola retakan atau celah pada permukaan kerabang.

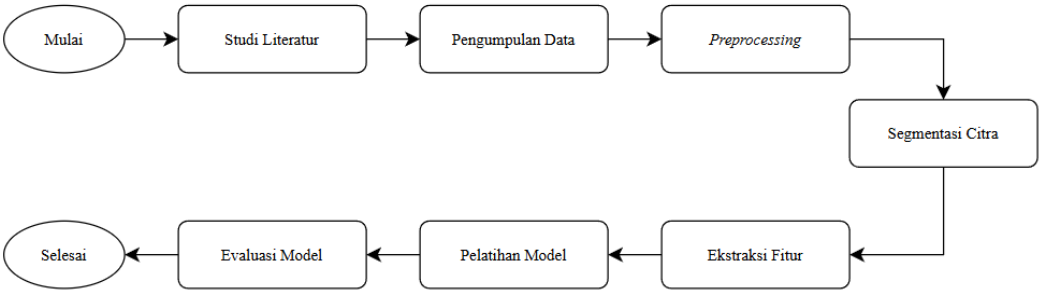
Data yang dianalisis merupakan citra kerabang telur ayam ras, yang kemudian diekstraksi fitur morfologinya untuk dijadikan input bagi algoritma *AdaBoost.* Sampel data yang digunakan pada penelitian ini berjumlah 154 citra telur retak dan 150 citra telur tidak retak, sehingga total *dataset* berjumlah 304 citra. Penelitian ini, dilakukan di lingkungan kampus Universitas Buana Perjuangan Karawang dan di rumah Penulis, yang dimulai dari bulan Oktober 2025 sampai bulai Maret 2026. Adapun rincian waktu pelaksanaan penelitian dapat dilihat pada Tabel 3.1.

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Ke-6
1	Studi Literatur						
2	Pengumpulan *Dataset*						
3	*Preprocessing* dan Pelatihan Model						
4	Evaluasi Model						
5	Penyusunan Laporan						

3.2 Prosedur Penelitian

Tahapan penelitian ini dimulai dari studi literatur, dilanjutkan dengan tahap pengumpulan data, *preprocessing*, segmentasi citra, ekstraksi fitur, pelatihan model *AdaBoost*, sampai dengan tahap evaluasi model. Adapun prosedur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Prosedur Penelitian

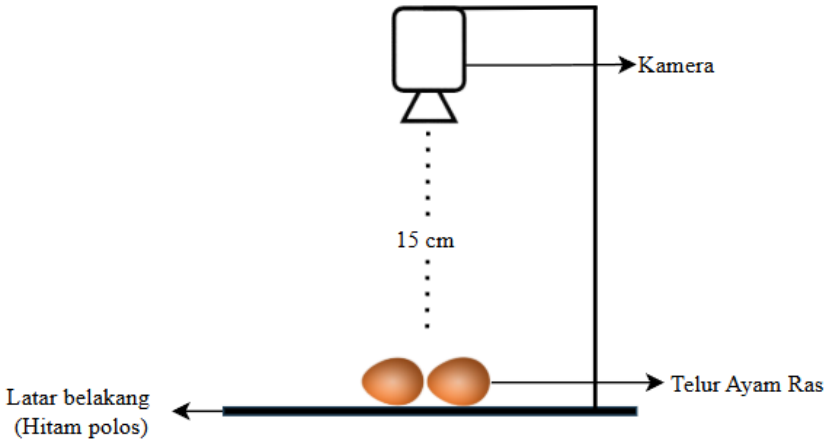
3.2.1 Studi Literatur

Pada tahap studi literatur dilakukan proses pengumpulan, pengkajian, dan analisis terhadap berbagai sumber referensi yang relevan dengan topik, yaitu deteksi retakan pada kerabang telur ayam ras menggunakan algoritma *AdaBoost* berdasarkan fitur morfologi citra. Tahap ini dilakukan untuk membangun dasar teori yang kokoh. Fokus utama dalam tahap ini, mengkaji objek telur ayam ras, *preprocessing citra*, segmentasi citra, ektraksi fitur, algoritma *AdaBoost*, dan metrik yang digunakan untuk mengevaluasi kinerja model.

3.2.2 Pengumpulan Data

Tahap pengumpulan data dilakukan untuk memperoleh data yang akan digunakan dalam penelitian, yang mana data dalam penelitian ini di peroleh dari pertenakan ayam ras yang berlokasi di Desa Karangmulya, Kec. Telukjambe Barat, Kab. Karawang. Pengumpulan data dilakukan dengan cara memotret telur ayam ras menggunakan kamera *smartphone* beresolusi 12 *MP*, dengan format file jpg. Setiap citra terdiri dari satu hingga empat butir telur ayam ras. Proses pengambilan citra dilakukan dengan memperhatikan jarak,

sudut, pencahayaan, dan latar belakang agar menghasilkan *dataset* yang seragam.



Gambar 3. 2 Ilustrasi Proses Pengambilan Dataset

Pada Gambar 3.2 menunjukan ilustrasi proses pengambilan *dataset*, yang mana kamera *smartphone* di posisi tegak lurus (90°) terhadap permukaan telur dengan jarak kamera dan telur sekitar 15 cm. Latar belakang citra menggunakan warna hitam polos untuk memudahkan proses segmentasi objek. Selain itu, pencahayaan diatur dengan memastikan cahaya merata dan meminimalkan bayangan yang dapat mempengaruhi kualitas citra. Jumlah data citra yang digunakan, yaitu 304 citra, yang terdiri dari 150 citra telur tidak retak dan 154 citra telur retak. Data yang dikumpulkan dikelompokan dalam dua kelas, yaitu retak dan tidak retak, untuk kemudian disimpan dalam direktori penyimpanan untuk diproses lebih lanjut.

3.2.3 Preprocessing

Proses *preprocessing* merupakan tahap awal yang dilakukan untuk meningkatkan kualitas citra. *Preprocessing* dilakukan untuk mengurangi gangguan visual, menekankan pengaruh *noise*, *serta* memperjelas karakteristik permukaan kerabang telur ayam ras. Pada penelitian ini dilakukan beberapa tahap *preprocessing* untuk meningkatkan kualitas citra, yang meliputi:

1. *Resize* Citra

   Resize merupakan langkah awal yang dilakukan pada tahap *preprocessing*, yang mana tahap ini dilakukan untuk meyeragamkan

dimensi citra agar ukuran citra yang digunakan konsisten, sehingga kestabilan proses komputasi dapat terjaga dan bias akibat perbedaan resolusi citra dihindari. Pada penelitian ini, ukuran citra diubah dengan menetapkan lebar citra sebesar 1000 piksel, sedangkan tinggi citra disesuaikan dengan proporsional berdasarkan rasio skala, dimana hal ini dilakukan untuk mempertahankan aspek rasio agar tidak terjadi distorsi bentuk objek. Adapun perhitungan rasio skala adalah sebagai berikut:

$$Rasio\ Skala = \frac{Wbaru}{Wasli} \quad (1)$$

Keterangan :

$Wbaru$: Lebar citra hasil *resize*

$Wbaru$: Lebar citra sebelum dilakukan proses *resize*

Setelah dilakukan perhitungan rasio skala, maka langkah selanjutnya, yaitu dilakukan perhitungan tinggi citra hasil *resize*, adapun perhitungan untuk menentukan tinggi citra, adalah sebagai berikut:

$$Hbaru = Hasli \times Rasio\ Skala \quad (2)$$

Keterangan:

$Hbaru$: Tinggi citra hasil *resize*

$Hasli$: Tinggi citra sebelum dilakukan proses *resize*

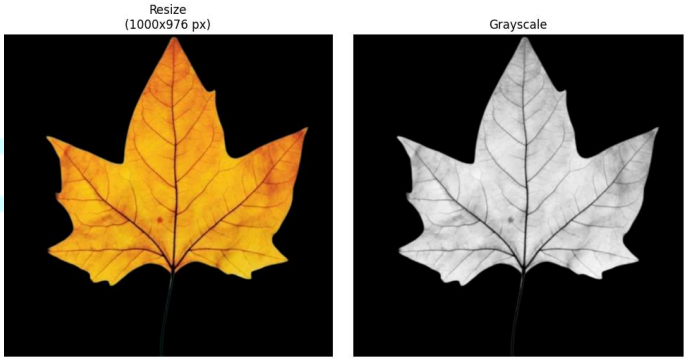
$Rasio\ Skala$: Rasio skala yang diperoleh dari perbandingan citra baru dengan citra alsi

Pada penelitian ini metode yang digunakan untuk menghitung nilai piksel baru saat dilakukan proses *resize* adalah *interpolasi area (INTER_AREA),* metode ini dipilih karena kemampuannya mempertahankan kualitas visual dengan meminimalkan distorsi.

2. Konversi *Grayscale*

Citra hasil *resize* yang memiliki ruang warna *RGB (Red, Green, Blue)* dikonversi menjadi citra *grayscale* untuk menyederhanakan informasi citra tanpa menghilangkan informasi penting yang berkaitan dengan pola retakan pada permukaan telur. Pada tahap ini citra asli *(RGB)* yang memiliki tiga kanal yang disederhanakan menjadi satu kanal tunggal, sehingga mempermudah proses komputasi karena proses selanjutnya akan dilakukan pada satu kanal. Selain itu deteksi retakan pada kerabang telur

ayam ras lebih bergantung pada perbedaan intensitas yang membentuk pola retakan pada permukaan kerabang dibandingkan informasi warna. Proses *grayscale* dilakukan menggunakan *library OpenCV* dengan parameter *COLOR_BGR2GRAY*, yang mana proses ini dijalankan menggunakan fungsi *cv2.cvColor()*

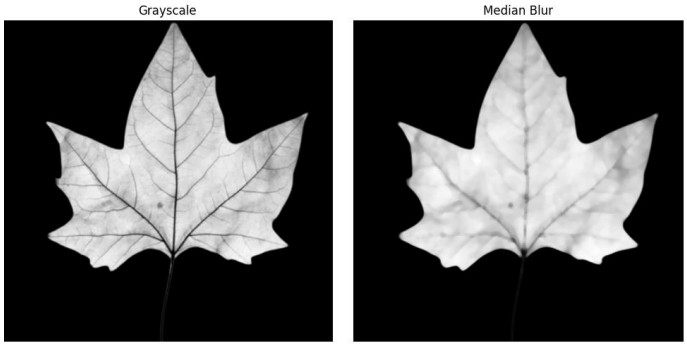


Gambar 3.3 Ilustrasi Konversi *Grayscale*

Pada Gambar 3.3 menunjukan ilustrasi proses konversi citra berwarna *(RGB)* menjadi citra *grayscale*, yang mana setelah dilakukan konversi *grayscale*, informasi warna pada citra dihilangkan dan digantikan dengan tingkat keabuan. Namun struktur dan bentuk objek tetap dipertahankan, sehingga memudahkan proses analisis pada tahap selanjutnya.

3. Reduksi *Noise*

Citra hasil konversi *grayscale*, selanjutnya dilakukan proses *Median filtering* untuk mereduksi *noise* yang ada pada citra. Penggunaan *median filtering* sebagai metode reduksi *noise* dilakukan karena *median filtering* dapat mereduksi *noise* yang terdapat pada citra, namun tetap mempertahankan tepi, yang mana hal ini sesuai dengan pengolahan citra yang membutuhkan ketelitian struktur objek. Pada penelitian ini, ukuran kernel yang dilakukan untuk menjalankan proses reduksi *noise*, yaitu 15×15, yang mana proses reduksi *noise* dijalankan menggunakan fungsi *cv2.medianBlur()*. Kernel ini dipilih untuk memberikan efek reduksi *noise* yang signifikan tanpa menghilangkan detail tepi objek secara berlebihan.

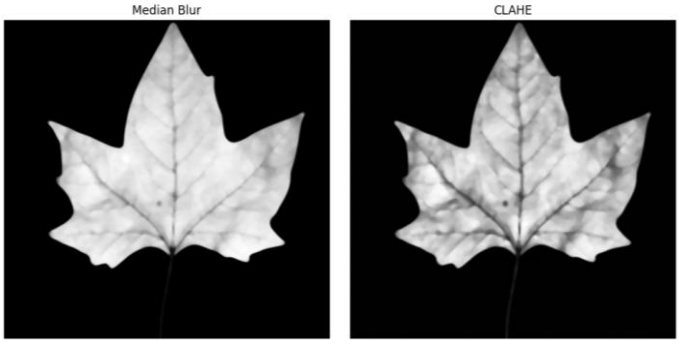


Gambar 3.4 Ilustrasi Median Filtering

Pada Gambar 3.4 menunjukan citra hasil konversi *grayscale* sebelum dilakukan reduksi *noise* dan sesudah dilakukan reduksi *noise* menggunakan *Median filtering.* Gambar tersebut digunakan untuk mengilustrasikan proses reduksi *noise.*

4. Peningkatan Kontras

Untuk memperjelas detail retakan dilakukan peningkatan kontras menggunakan metode *Contrast Limited Adaptive Histogram Equalization (CLAHE),* dengan menetapkan *tiles* dan *clip limit. CLAHE* dipilih karena kemampuannya dalam meningkatkan kontras secara lokal dan adaptif, sehingga pola retakan menjadi lebih tajam. Pada penelitian ini, diterapkan nilai *clip limit* sebesar *2.0* dan *tile grid* 8×8. Penggunaan *clip limit 2.0,* dipilih untuk membatasi peningatan kotras berlebihan, sehingga dapat mencegah penguatan *noise* pada citra, sedangan *tile grid* 8×8 dipilih untuk menjaga keseimbangan dalam peningkatan kontras lokal. Proses peningkatan kontras menggunakan *CLAHE,* pada penelitian ini dijalankan menggunakan fungsi *cv2createCLAHE()*



Gambar 3.5 Ilustrasi *CLAHE*

Pada Gambar 3.5 menampilkan perbandingan antara citra sebelum dan sesudah dilakukan peningkatan kotras menggunakan metode *CLAHE* sebagai ilustrasi tahap peningkatan kontras lokal. Citra yang sudah dilakukan peningkatan kontras menggunakan *CLAHE* akan digunakan sebagai input pada tahap segmentasi citra.

3.2.4 Segmentasi Citra

Tahap segmentasi citra dilakukan untuk memisahkan setiap butir telur ayam ras yang terdapat pada satu citra menjadi objek-objek yang terpisah. Pada penelitian ini, metode yang digunakan pada segmentasi citra adalah *marker controlled watershed segmentation. Marker controlled watershed segmentation* merupakan teknik segmentasi berbasis topografi citra yang memanfaatkan konsep air yang mengalir pada permukaan intensitas citra untuk membentuk batas antara objek. Metode ini dipilih karena kemampuannya dalam memisahkan objek yang saling berdekatan, yang mana hal ini sesuai dengan kebutuhan penelitian ini, yaitu memisahkan beberapa butir telur dalam satu citra.

Citra yang digunakan pada tahap ini merupakan citra hasil *preprocessing,* yang mana citra hasil *preprocessing* dilakukan konversi citra biner secara otomatis berdasarkan distribusi intensitas piksel menggunakan metode *otsu.* Selanjtnya dilakukan operasi morfologi opening untuk menghilangkan *noise* kecil, serta penutupan lubang *(hole filling)* agar objek telur yang dihasilkan solid. Setelah objek telur solid dilakukan penentuan area *foreground* dan *background,* dimana area *background* dihasilkan melalui proses dilasi, sedangkan area *foreground* ditentukan menggunakan transformasi jarak *(distance transform)* untuk memperoleh bagian inti objek. Selisih keduanya akan menghasilkan area tidak pasti *(unknown region)* yang digunakan dalam pembentukan *marker,* yang mana *marker* tersebut akan digunakan oleh metode *watershed,* sehingga *watershed* dapat memberikan *label* yang berbeda pada setiap objek yang terdeteksi. Hasil akhir dari proses ini yaitu citra dengan *label* segmentasi yang merepresentasikan setiap butir telur sebagai objek terpisah.

3.2.5 Ekstraksi Fitur

Proses ektraksi fitur untuk memperoleh representasi numerik yang menggambarkan karakteristik retakan pada kerabang telur ayam ras. Proses ini dilakukan dengan mengidentifikasi setiap objek telur berdasarkan label segmentasi, yang kemudian dilakukan pembentukan *region of interest (ROI)* berdasarkan *bounding box* dari kontur terbesar, dimana *ROI* digunakan untuk membatasi area analisis hanya dilakukan pada bagian telur yang relevan. Setelah, pembentukan *ROI* selesai dilakukan deteksi retakan menggunakan metode *canny edge detection* yang dilakukan pada citra *ROI*. Hasil deteksi retakan berupa citra biner yang merepresentasikan pola retakan pada objek telur, yang mana citra biner akan dianalisis menggunakan pelabelan komponen terhubung *(connected component labeling)* untuk mengidentifikasi setiap *region* retakan yang terdeteksi, sehingga *region* tersebut dilakukan perhitungan fitur morfologi. Adapun fitur morfologi yang dihitung adalah sebagai berikut:

a. Area, yaitu jumlah piksel yang merepresentasikan luas total retakan.
b. Perimeter, yaitu panjang total batas dari *region* retakan.
c. *Minor Axis*, yaitu panjang sumbu *minor* terbesar dari *region* retakan, yang menggambarkan dimensi terkecil dari retakan dominan.
d. *Rekative Area*, yaitu rasio antara luas retakan terhadap luas total objek yang dinyatakan dalam persentasi, yang mana nilai luas objek diperoleh dari jumlah piksel *mask* objek hasil segmentasi.

Seluruh fitur yang dihasilkan disusun dalam bentuk *dataset* terstruktur. Selain fitur morfologi, nama file citra, identitas objek (TELUR_ID), serta koodinat *bounding box* juga ikut disusun sebagai informasi tambahan. Setiap baris data yang dihasilkan merepresentasikan satu objek telur. Untuk memperjelas tahapan ektraksi fitur, *pseudocode* proses tersebut disajikan sebagai berikut:

Tabel 3.2 Ekstraksi Fitur

Pseudocode : **Ekstraksi Fitur**		
Input	:	Citra hasil preprocessing, Label objek hasil segmentasi, Label kelas

Output : File CVS fitur morfologi telur

START

Inisialisasi tabel data fitur kosong

For setiap objek telur berdasarkan label hasil segmentasi do

Jika label <= 1 → labjut ke objek berikutnya

Ambil mask objek dari label segmentasi

Cari kontur obejek

Jika tidak ditemukan → labjut objek berikutnya

Tentukan bounding nox (x,y,w,h)

Ektraksi ROI hasil preprocessing sesuai bounding box

Ambil mask ROI

Deteksi retakan (canny)

Lakukan erosi pada mask untuk mengurangi pengaruh tepi

Lakukan masking untuk memperoleh citra retakan

Hitung luas objek telur

Lakukan pelabelan komponen terhubung pada citra retakan

Jika tidak terdapat region retahan then

Area = 0

Perimeter = 0

Minor Axis = 0

Relative Area = 0

Else

Hitung total area retakan

Hitung total perimeter retakan

Ambil minor axis terbesar

Hitung relative area (% terhadap luas objek)

End if

Simpan data ke tabel:

{

Nama File,

Telur ID,

X, Y, W, H

Area,
Perimeter,
Minor Axis
Relative Area,
Label
}
End for
Simpan tabel data fitur ke file CSV
END

3.2.6 Pelatihan Model

Proses pelatihan model dilakukan menggunakan *dataset* hasil ektraksi fitur yang merupakan representasi satu objek telur dengan atribut *area*, *perimeter*, *minor axis*, dan *relative area*. *Dataset* dibagi menjadi dua bagian, yaitu data latih *(training set)* dan data uji *(testing set)* dengan proporsi 80:20, dimana pembagian dilakukan menggunakan metode *stratified sampling*, sehingga proporsi kelas pada data tetap seimbang. Setelah dilakukan pembagian data, dilakukan proses normalisasi fitur menggunakan metode *standardization* dengan bantuan *StandardScaler*, dimana proses ini dilakukan untuk mengubah distribusi fitur sehingga memiliki nilai rata-rata sebesar nol dan standar deviasi sebesar satu, sehingga setiap fitur memiliki skala yang sebanding agar tidak ada fitur yang mendominasi proses pelatihan model.

Pada penelitian ini, pelatihan model menggunakan model *AdaBoost (Adaptive Boosting)* dengan *base learner* berupa *Decision Tree Classifier*. Pelatihan model ini dijalankan menggunakan pohon keputusan dengan kedalaman maksimum *(max_depth)* sebesar 2, sehingga menghasilkan model dasar yang sederhana *(weak learner)*. Parameter jumlah estimator *(n_estimators)* ditetapkan sebanyak 50, yang menunjukan jumlah iterasi *boosting*, dan *learning rate* seberar 0.1 digunakan untuk mengatur konstribusi setiap model dalam proses pelatihan. Untuk merepresentasikan alur proses pelatihan model secara sistematis, *pseudocode* disajikan sebagai berikut:

Tabel 3.3 Pelatihan Model AdaBoost

Pseudocode :* Pelatihan Model *AdaBoost

Input : Dataset hasil ekstraksi fitur

Output : Model AdaBoost

Scaler

START

Load dataset hasil ektraksi fitur dari file CSV

If dataset tidak ditemukan then

Tampilkan pesan error

Stop

End if

Pisahkan data menjadi:

X ← fitur (Area, Perimeter, Minor Axis, Relative)

Y ← label kelas

Bagi dataset menjadi data latih dan data uji ← split 80:20

Gunakan stratified sampling berdasarkan label y

Hasil : x train, x test, y train, y test

Lakukan normalisasi fitur

Inisialisasi StandardScaler

Inisialisasi model AdaBoost

Base classifier ← Decisien Tree (max depth =2)

Jumlah estimator ← 50

Learning rate ← 0.1

Latih model AdaBoost menggunakan x train dan y train

Simpan model dan scaler

Simpan model ← AdaBoost model telur.joblib

Simpan scaler ← scaler telur.joblib

END

3.2.7 Evaluasi Model

Evaluasi dilakukan dengan menguji model terhadap data uji yang belum pernah dikenali oleh model. Tahap ini dilakukan untuk mengukur kinerja

model *AdaBoost* yang telah dilatih untuk mendeteksi retakan pada kerabang telur ayam ras berdasarkan fitur morfologi. Parameter yang digunakan untuk menguji model, yaitu *confusion matrix. Confusion metrix* digunakan untuk mengetahui hasil prediksi model terhadap label akutual. Berdasarkan data yang didapat dari *confusion matrix*, dilakukan beberapa perhitungan terhadap metrik untuk mengetahui kinerja model, seperti akurasi, presisi, *recall*, dan *f1-score*. Untuk menggambarkan proses evaluasi model secara sistematis, *pseudocode* disajikan sebagai berikut:

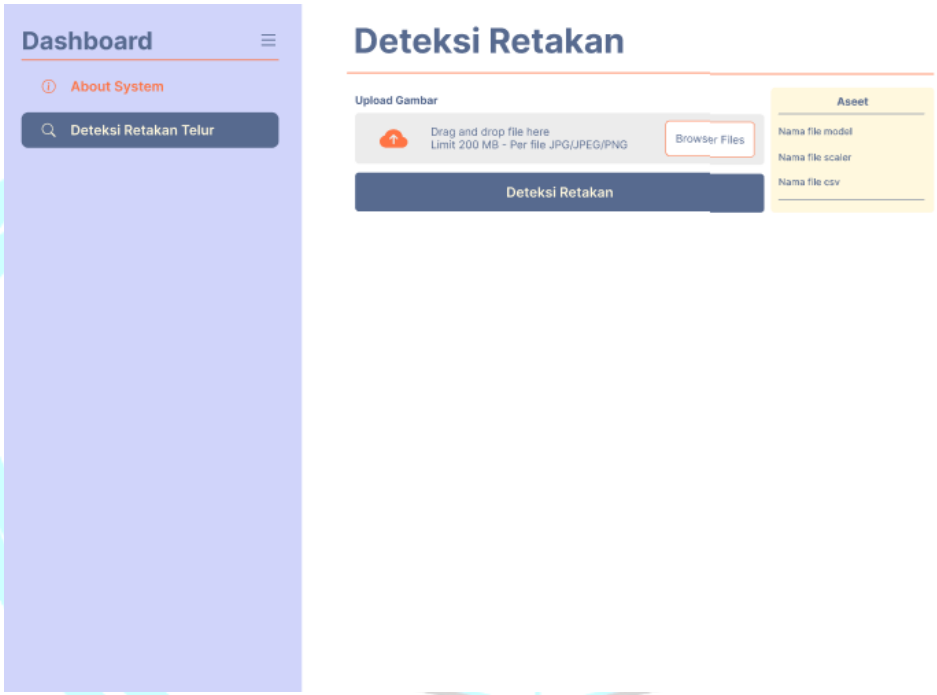
Tabel 3.4 Evaluasi Model

Pseudocode : Evaluasi Model		
Input	:	Model AdaBoost yang sudah dilatih
		Data latih dan label sebenarnya
		Data uji dan label sebenarnya
		Nama kelas (label)
Output	:	Akurasi model dan confusion matrix
START		
Load model AdaBoost yang telah dilatih		
Prediksi kelas pada data latih menggunakan model		
Prediksi kelas data uji menggunakan model		
Menghitung confusion matrix		
Menghitung akurasi		
Menghitung laporan klasifikasi		
Menampilkan akurasi, precision, recall, F1-Score		
Menampilkan confusion matrix dalam bentuk heatmap		
END		

3.2.8 Impelemntasi *Streamlit*

Model yang sudah dilatih diimplementasikan menggunakan antar muka *streamlit* agar model dapat diuji dengan menggunakan data baru. Pada antarmuka *streamlit*, sistem menyediakan fitur *upload* yang memungkinkan pengguna mengunggah citra telur, yang mana citra yang diunggah akan diproses secara otomatis melalui *pipeline* sistem yang telah diintegrasikan,

sehingga pada antarmuka *streamlit* dapat menghasilkan bentuk visualisasi *bonding box* pada objek telur beserta label prediksi untuk setiap objek deteksi. Adapun desain antarmuka pengguna *(User Interface)* dari sistem deteksi retakan pada kerabang telur ayam ras yang dibangun menggunakan *streamlit* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Desain *UI* Dari Sistem Deteksi

KARAWANG