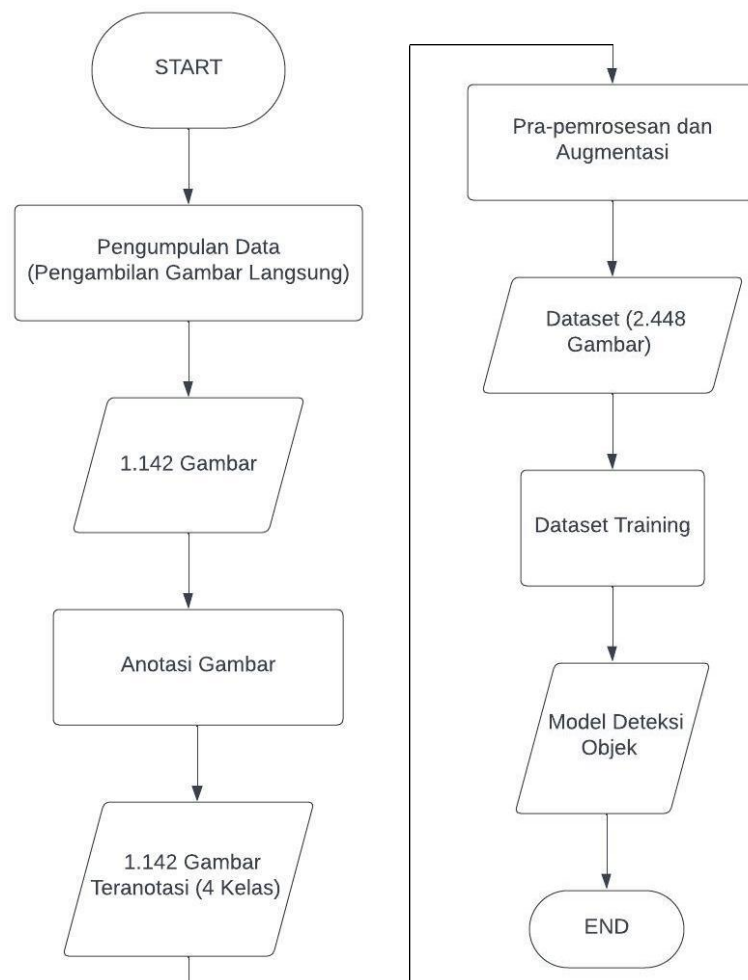


## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Pembuatan Model

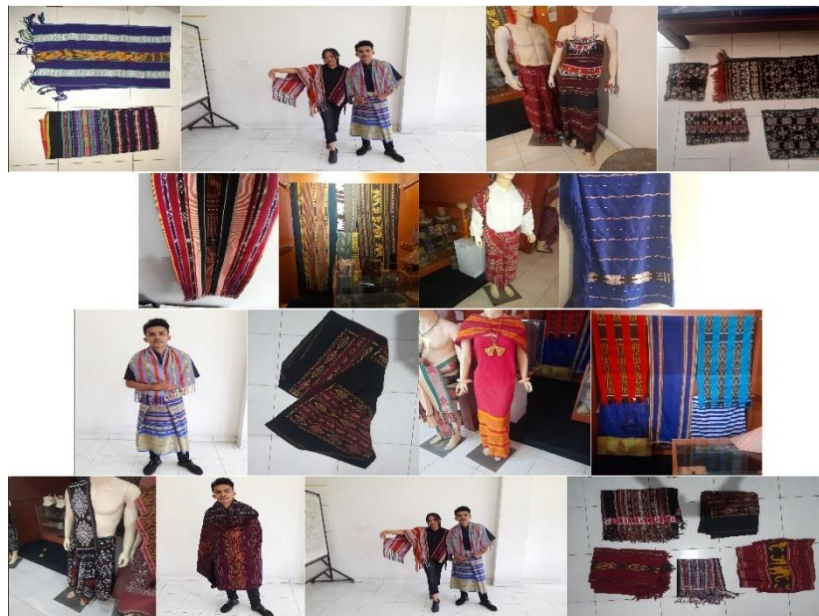
Hasil yang diperoleh dari pembuatan model adalah dataset kain tenun dan juga model deteksi objek yang akan dipakai pada aplikasi. Pembuatan model secara garis besar dapat dilihat pada Gambar 4. 1.



Gambar 4. 1. Alur pengerjaan dan hasil pembuatan model

#### 4. 1. 1. Pengumpulan Data dan Pembuatan Dataset

Penelitian ini mengumpulkan data mentah sebanyak 1142 gambar kain tenun tradisional NTT yang berasal dari seluruh kabupaten yang terletak di NTT. Beberapa contoh gambar yang dikumpulkan dapat dilihat pada Gambar 4. 2.



Gambar 4. 2. Contoh-contoh gambar terkumpul

Data kemudian dikelompokkan beberapa kabupaten ke dalam satu kelompok berdasarkan pulau-pulau besar di NTT sebagai kelas-kelas yang akan dideteksi. Hal ini dilakukan karena terdapat sejumlah kabupaten di mana sampel yang diperoleh dinilai terlalu kecil untuk bisa menjadi kelas yang dapat berdiri sendiri pada model deteksi objek sehingga dapat menurunkan kualitas dari model yang akan dibuat jika diteruskan. Data kemudian dikelompokkan berdasarkan asal kain tenun tradisional NTT ke

dalam 4 kategori, yaitu Flores, Sumba, Timor, dan Lain-lain. 4 kategori ini memiliki rincian yang dapat dilihat di Tabel 4. 1.

Tabel 4. 1. Pembagian kabupaten ke dalam kelompok pulau

<b>Pulau</b>	<b>Kabupaten / Kota (Ibu Kota)</b>
Flores	Manggarai Barat (Labuan Bajo) Manggarai (Ruteng) Manggarai Timur (Borong) Ngada (Bajawa) Nagekeo (Mbay) Ende (Ende) Sikka (Maumere) Flores Timur (Larantuka) Lembata (Lewoleba)
Sumba	Sumba Barat (Waikabubak) Sumba Barat Daya (Tambolaka) Sumba Tengah (Waibakul) Sumba Timur (Waingapu)
Timor	Belu (Atambua) Kupang (Oelamasi) Malaka (Betun) Sabu Raijua (Menia) Timor Tengah Selatan (Soe) Timor Tengah Utara (Kefamenanu)

Pulau	Kabupaten / Kota (Ibu Kota)
	Kota Kupang (-)
Lain-lain	Alor (Kalabahi) Rote Ndao (Baa)

Pembuatan dataset dilakukan kepada seluruh gambar yang telah dikumpulkan berdasarkan pembagian kelas di atas. Setelah melewati proses pra-pemrosesan dan augmentasi, jumlah gambar yang dimiliki berhasil berkembang dari 1142 gambar menjadi 2448 gambar. Seluruh gambar dibagi ke dalam 3 bagian yaitu *training*, *validation*, dan *test* dengan rasio 8:1:1 atau 1968 gambar pada *training*, 240 pada *validation*, dan 240 pada *test*.

#### 4. 1. 2. Pelatihan Dataset

Pelatihan sebanyak 50 epoch dilakukan pada Google Colab dan Jupyter Notebook untuk mengevaluasi hasil pelatihan yang akan dipilih. Hasil pelatihan dengan menggunakan Jupyter Notebook menunjukkan ketidak-konsistenan dalam 3 kali percobaan (Gambar 4. 3), sementara pelatihan dengan Google Colab memberikan hasil yang konsisten pada 3 percobaan. Oleh karena itu, diputuskan untuk melanjutkan tahap berikutnya dengan menggunakan data hasil pelatihan dari Google Colab.

```

Validating runs/train/exp2/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
  Class      Images  Instances      P          R      mAP50  mAP50-95: 100%|██████████|
    all         240        420      0.939      0.923      0.972      0.77
  Flores         240        151      0.933      0.924      0.974      0.786
    Sumba         240        113      0.911      0.903      0.959      0.74
    Timor         240         91      0.96      0.879      0.967      0.733
  Lain-lain         240         65      0.949      0.985      0.99      0.822

Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
  Class      Images  Instances      P          R      mAP50  mAP50-95: 100%|██████████|
    all         240        420      0.888         0.9      0.95      0.729
  Flores         240        151      0.934      0.894      0.958      0.744
    Sumba         240        113      0.835      0.876      0.932      0.717
    Timor         240         91      0.826         0.89      0.922      0.669
  Lain-lain         240         65      0.956      0.938      0.988      0.785

Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
  Class      Images  Instances      P          R      mAP50  mAP50-95: 100%|██████████|
    all         240        420      0.846      0.858      0.912      0.659
  Flores         240        151      0.885      0.864      0.935      0.701
    Sumba         240        113      0.815      0.814      0.901      0.653
    Timor         240         91      0.855      0.845      0.883      0.595
  Lain-lain         240         65      0.829      0.908      0.932      0.685

```

Gambar 4. 3. Hasil pelatihan dataset menggunakan Jupyter Notebook

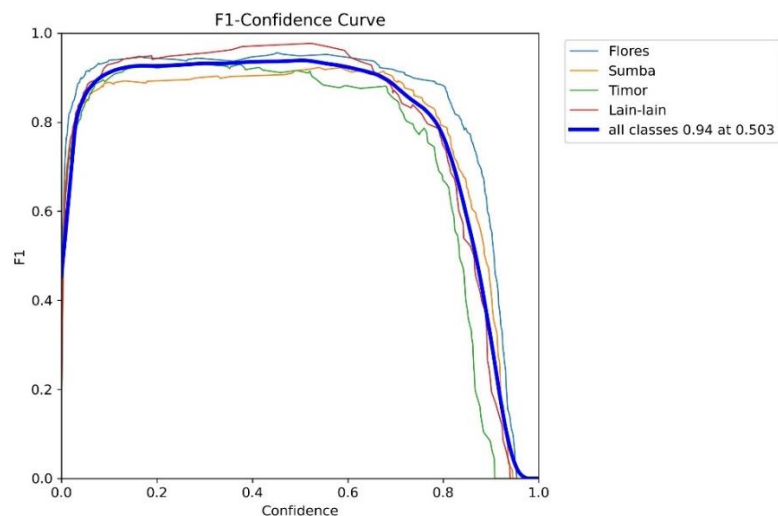
Waktu yang dibutuhkan untuk menyelesaikan pelatihan adalah selama 0,519 jam (31 menit 8 detik 400 milisekon). Pelatihan dataset menghasilkan sebuah model deteksi objek dengan ukuran 14,5 MB. Parameter-parameter terukur yang dihasilkan dari pelatihan dapat dilihat dalam Tabel 4. 3.

Tabel 4. 2. Hasil evaluasi parameter pelatihan model

Kelas	<i>Precision</i>	<i>Recall</i>	mAP50	mAP50-95
Keseluruhan	0,948	0,928	0,972	0,770
Flores	0,979	0,923	0,980	0,783
Sumba	0,918	0,903	0,950	0,735
Timor	0,929	0,901	0,970	0,750
Lain-lain	0,966	0,985	0,987	0,811

Hasil pelatihan menunjukkan bahwa secara keseluruhan, model mencapai *precision* sebesar 94,8%, *recall* sebesar 92,8%, mAP50 sebesar 97,2%, dan mAP50-95 sebesar 77,0%. Selain itu, hasil pelatihan juga menunjukkan bahwa kelas “Lain-lain” memiliki hasil pengukuran parameter yang paling tinggi pada pengukuran *recall*, mAP50, dan mAP50-95 dengan nilai masing-masing sebesar 98,5%, 98,7%, dan 81,1%. Hal ini dapat disebabkan karena gambar pada kelas “Lain-lain” memiliki jumlah yang paling rendah dari kelas lainnya sehingga model menjadi lebih cepat dan mudah dalam menyelesaikan pelatihan. Sedangkan parameter *precision* tertinggi ditunjukkan dari kelas Flores dengan nilai sebesar 97,9%.

Kualitas dari model ditunjukkan melalui hubungan *F1 score* dengan *confidence*. *F1 score* merujuk kepada titik imbang antara *precision* dan *recall*, sedangkan *confidence* merujuk kepada seberapa besar keyakinan model dari hasil prediksinya. Hubungan antara *F1 score* dan *confidence* ditunjukkan melalui Gambar 4. 4. Hasil pelatihan menunjukkan model secara keseluruhan memiliki *F1 score* sebesar 0,940 saat *confidence* sebesar 0,503. Ini berarti model memiliki nilai *F1 score* optimal saat nilai *confidence* sebesar 0,503 dengan nilai sebesar 0,940.



Gambar 4. 4. Grafik *F1 score* terhadap *confidence*

Hasil pelatihan model juga memberikan *confusion matrix* untuk menggambarkan distribusi pembelajaran pada setiap kelas pada proses *validation*. Proses *validation* sendiri merupakan proses evaluasi otomatis yang dilakukan YOLOv5 terhadap hasil pelatihan. *Confusion matrix* untuk model terlatih dapat dilihat pada Tabel 4. 4. *Confusion matrix* tersebut memiliki sebuah kelas tambahan yang diberi nama “Latar Belakang”. Kelas ini merupakan kelas yang otomatis ditampilkan oleh YOLOv5 ketika selesai melakukan pelatihan model untuk menunjukkan berapa banyak gambar yang model deteksi atau tidak deteksi.

Tabel 4. 3. *Confusion matrix* pada model terlatih

		Prediksi				
Aktual	KELAS	Flores	Sumba	Timor	Lain-lain	Latar Belakang
	Flores	143	0	0	1	7
	Sumba	0	103	2	0	8
	Timor	1	0	88	0	2

	Lain-lain	2	0	0	<b>63</b>	0
	Latar Belakang	8	10	8	5	<b>0</b>

Kesalahan prediksi terbesar dapat dilihat pada korelasi antara kelas Sumba dan kelas Latar Belakang. Model terlihat mengalami kesalahan deteksi sebanyak 8 gambar pada kelas Sumba dengan mendeteksinya sebagai kelas Latar Belakang. Hal yang sama terjadi sebaliknya, dimana model salah mendeteksi sebanyak 10 gambar dari kelas Latar Belakang sebagai kelas Sumba. Hal ini disebabkan karena salah satu tantangan dari deteksi objek, yaitu *cluttered background*, dimana banyak dari gambar-gambar yang diambil memiliki latar belakang yang bertekstur menyerupai kain tenun dari Sumba. *Confusion matrix* di atas dapat dinyatakan ke dalam persentasi seperti pada Tabel 4. 5.

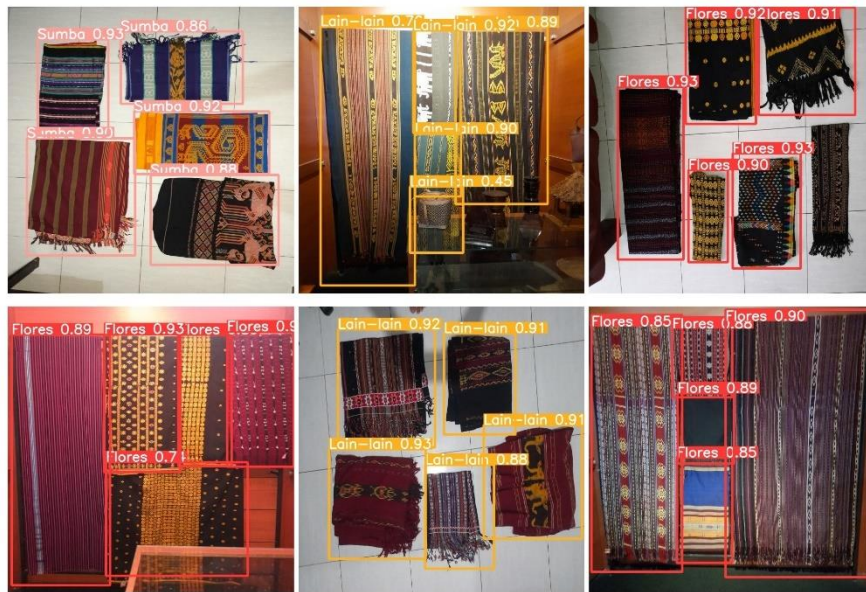
Tabel 4. 4. *Confusion matrix* dalam persentasi

Prediksi						
Aktual	KELAS	Flores	Sumba	Timor	Lain-lain	Latar Belakang
	Flores	<b>95%</b>	0%	0%	1%	5%
	Sumba	0%	<b>91%</b>	2%	0%	7%
	Timor	1%	0%	<b>97%</b>	0%	2%
	Lain-lain	2%	0%	0%	<b>97%</b>	0%
	Latar Belakang	26%	32%	26%	16%	<b>0%</b>

Uji coba model juga dilakukan secara manual dengan menggunakan gambar-gambar yang ada pada folder *test*. Hasil yang ditemukan adalah model terlihat mampu untuk mendeteksi kelas-kelas yang ada pada gambar



dengan tepat serta memiliki *confidence* umumnya di atas 0,5. Model juga terlihat mampu mendeteksi beberapa kain sekaligus atau mengabaikan kain yang bukan termasuk ke dalam kain tenun NTT di dalam suatu gambar, yang ditunjukkan pada Gambar 4. 5.



Gambar 4. 5. Model mendeteksi beberapa kain sekaligus

Model dinilai cukup tahan dalam menghadapi tantangan-tantangan yang ada pada deteksi objek. Hasil penelitian menunjukkan bahwa model mampu mendeteksi dengan tepat ketika diberikan tantangan-tantangan seperti *illumination*, *viewpoint variation*, *scale*, *occlusion*, *deformation*, dan *intra-class variation*. Hal ini ditunjukkan dengan model masih mampu mendeteksi kelas-kelas yang ada pada gambar-gambar saat kain tenun dikenakan, dilipat, digantung, diletakkan dari jauh, atau diambil gambarnya dari berbagai sisi. Hal ini bisa dilihat pada Gambar 4. 6.



Gambar 4. 6. Model menghadapi tantangan-tantangan deteksi objek

Pada penelitian ini, dapat dinilai bahwa model masih belum cukup baik dalam menghadapi tantangan *cluttered background*. *Cluttered background* pada penelitian ini merujuk kepada objek-objek yang bukan merupakan objek yang ingin dideteksi namun memiliki motif serupa dengan kain tenun tradisional NTT. Penelitian ini menemukan sejumlah *cluttered background* berupa latar belakang yang memiliki objek-objek lain seperti kerajinan tangan tradisional non-tenun, peralatan elektronik yang memiliki bagian bergaris-garis, dan meja. Gambar-gambar yang diambil saat kain tenun berada di depan objek-objek tersebut ditemukan seringkali masih terjadi kesalahan deteksi, terutama ketika objek tersebut merupakan

kerajinan tangan tradisional NTT non-tenun. Hal ini ditunjukkan melalui Gambar 4. 7.



Gambar 4. 7. Kesalahan deteksi saat model menghadapi *cluttered background*

Model yang telah dibuat kemudian dikonfigurasi sehingga dapat menerima *Input* berupa hasil penangkapan kamera perangkat secara *real-time*. Konfigurasi yang dilakukan meliputi pemuatan model deteksi objek dan mengintegrasikannya dengan kamera perangkat secara langsung. Kamera perangkat awalnya dibuka untuk menerima *Input* secara *real-time*. *Input* dari kamera perangkat tersebut kemudian dibaca pada setiap *frame* yang kemudian diubah menjadi file gambar dengan format JPEG. Gambar-

gambar tersebut dibaca dengan menggunakan model untuk mendeteksi ada tidaknya kain tenun tradisional NTT di dalamnya. Hasil deteksi kemudian dicetak dan segera ditampilkan kembali dalam bentuk file-file gambar yang kemudian kembali ditampilkan sebagai frame video dari kamera perangkat. Adapun listing code yang diperlukan untuk melakukan konfigurasi ini adalah sebagai berikut.

```
def gen():
    cap=cv2.VideoCapture(0)
    while(cap.isOpened()):
        success, frame = cap.read()
        if success == True:
            ret,buffer=cv2.imencode('.jpg',frame)
            frame=buffer.tobytes()
            img = Image.open(io.BytesIO(frame))
            results = model(img, size=640)
            results.print()
            img = np.squeeze(results.render())
            img_BGR = cv2.cvtColor(img,
cv2.COLOR_RGB2BGR)
        else:
            break
        frame = cv2.imencode('.jpg',
img_BGR)[1].tobytes()
        yield(b'--frame\r\n'b'Content-Type:
image/jpeg\r\n\r\n' + frame + b'\r\n')
```