

TF4063 Sains Data Rekayasa

Laporan Tugas Besar

**PREDIKSI BERAT AYAM BERDASARKAN MODEL
MACHINE LEARNING DENGAN MASK RCNN R50
FPN DAN *FRAMEWORK* DETECTRON2**



Oleh:

Fernando Alvirianda Yoga	13319103
Jihad Zakki Darajad	13320064
Jacky Wu	10022143

**PROGRAM STUDI TEKNIK FISIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI BANDUNG**

2022

DAFTAR ISI

1	PENDAHULUAN	2
1.1	Latar Belakang	2
1.2	Tujuan & Sasaran	2
1.3	Metodologi	2
2	STUDI PUSTAKA	3
3	PERANCANGAN & IMPLEMENTASI	6
3.1	Perancangan	6
3.2	Implementasi	7
3.3	Eksekusi	7
4	PENGUJIAN & ANALISIS	12
5	KESIMPULAN & HIKMAH	19
5.1	Kesimpulan	19
5.2	Hikmah	19
6	DAFTAR PUSTAKA	21

NOMENKLATUR

Buatlah tabel singkatan dan atau simbol yang digunakan di dokumen ini

Istilah	Keterangan
ML	Machine Learning, proses pengembangan algoritma yang bisa memproses jenis data tertentu untuk menghasilkan output yang spesifik.
CCTV	Closed Circuit Television, jenis perekam video yang mengawasi suatu lokasi dalam jangka waktu panjang
COCO	Common Objects In Context, format dataset yang umum digunakan untuk machine learning berbasis visual
DNN	Deep Neural Network

1 PENDAHULUAN

1.1 Latar Belakang

Pengerjaan tugas besar ini secara garis besar ditujukan untuk:

- Meningkatkan keilmuan dalam bidang ML.
- Membandingkan model yang dipakai.
- Memberi manfaat praktis kepada mitra, yaitu untuk mampu memprediksi berat ayam melalui pantauan CCTV.

1.2 Tujuan & Sasaran

Tujuan dan sasaram kami melaksanakn tugas besar ini adalah:

- Mempelajari dan wawasan keilmuan di bidang ML.
- Membandingkan model.
- Memberikan sebuah model yang mampu memprediksi berat ayam.

1.3 Metodologi

1. Manual Annotation : Mengambil data K10 dan melakukan anotasi manual menggunakan Tool LabelMe.
2. Config : Melakukan konfigurasi dasar pada kode seperti instalasi Dependencies, Setup Drive tempat pengumpulan dataset, Melakukan inisialisasi kode, serta menunjukkan anotasi yang telah dibuat.
3. Training : Melakukan pelatihan kepada kode yang telah dibuat dengan melakukan konfigurasi pelatihan, pelatihan, serta analisis pelatihan.
4. Predict : Mengambil model dari pelatihan untuk memprediksi berat ayam pada Dataset K11.

2 STUDI PUSTAKA

2.1 Machine Learning

Machine Learning merupakan sebuah ilmu pengembangan algoritme dan model yang digunakan sistem komputer untuk menjalankan tugas tanpa instruksi eksplisit, tetapi Machine learning akan mengandalkan pola serta inferensi sebagai gantinya. Sistem komputer menggunakan algoritme machine learning untuk memproses data historis berjumlah besar dan mengidentifikasi pola data. Hal ini memungkinkannya untuk memprediksi hasil yang lebih akurat dari set data input yang diberikan.

Istilah Machine Learning pertama kali dikemukakan oleh beberapa ilmuwan matematika seperti Adrien Marie Legendre, Thomas Bayes dan Andrey Markov pada tahun 1920-an dengan mengemukakan dasar-dasar Machine Learning dan konsepnya. Sejak saat itu ML banyak yang mengembangkan. Salah satu contoh dari penerapan ML yang cukup terkenal adalah Deep Blue yang dibuat oleh IBM pada tahun 1996.

Teknik Supervised Learning merupakan teknik yang bisa kamu terapkan pada pembelajaran mesin yang bisa menerima informasi yang sudah ada pada data dengan memberikan label tertentu. Diharapkan teknik ini bisa memberikan target terhadap output yang dilakukan dengan membandingkan pengalaman belajar di masa lalu.

Selain itu juga terdapat teknik Unsupervised Learning merupakan teknik yang bisa kamu terapkan pada machine learning yang digunakan pada data yang tidak memiliki informasi yang bisa diterapkan secara langsung. Diharapkan teknik ini dapat membantu menemukan struktur atau pola tersembunyi pada data yang tidak memiliki label. Sedikit berbeda dengan Supervised Learning, kamu tidak memiliki data apapun yang akan dijadikan acuan sebelumnya.

2.2 Alat Machine Learning

Zaman sekarang membuat machine learning cukup mudah karena bisa menggunakan program yang ada di internet untuk memudahkan proses pembuatan dan pelatihan model.

Pada Tugas Besar ini, kami menggunakan Detectron2 sebagai alat machine learning utama. Detectron2 adalah *library* yang dibuat oleh tim riset Facebook yang mampu membuat model machine learning untuk mendeteksi objek dalam bentuk visual [1]. Detectron2 adalah sistem yang berbasis PyTorch, yaitu pustaka Machine Learning yang berbasis Python.

Model utama yang kami gunakan pada Detectron2 untuk tugas ini adalah Model Mask R-CNN, atau Mask Region-based Convolution Neural Network [2]. Mask R-

CNN adalah Model DNN yang mampu mendeteksi objek. Output yang dihasilkan berupa batas objek, klasifikasi, serta mask overlay yang menutupi objek secara mendetil.

2.3 Regresi Linear

Regresi linear adalah teknik analisis data yang memprediksi nilai data yang tidak diketahui dengan menggunakan nilai data lain yang terkait dan diketahui. Secara matematis memodelkan variabel yang tidak diketahui atau tergantung dan variabel yang dikenal atau independen sebagai persamaan linier. Misalnya, anggaplah Anda memiliki data tentang pengeluaran dan pendapatan Anda untuk tahun lalu. Teknik regresi linier menganalisis data ini dan menentukan bahwa pengeluaran Anda adalah setengah dari penghasilan Anda. Mereka kemudian menghitung biaya masa depan yang tidak diketahui dengan mengurangi separuh pendapatan yang diketahui di masa depan.

Model regresi linier relatif sederhana dan memberikan rumus matematika yang mudah ditafsirkan untuk menghasilkan prediksi. Regresi linier adalah teknik statistik yang sudah ada dan mudah diterapkan pada perangkat lunak dan komputasi. Bisnis menggunakannya untuk mengonversi data mentah secara andal dan dapat diprediksi menjadi kecerdasan bisnis serta wawasan yang dapat ditindaklanjuti. Para ilmuwan di berbagai bidang, termasuk biologi serta ilmu perilaku, lingkungan, dan sosial menggunakan regresi linier untuk melakukan analisis data awal dan memprediksi tren masa depan. Banyak metode ilmu data, seperti machine learning dan kecerdasan buatan, menggunakan regresi linier untuk memecahkan masalah yang kompleks.

Teknik regresi linier sederhana mencoba untuk menyusun grafik garis antara dua variabel data, yaitu x dan y . Sebagai variabel independen, x berada di sepanjang sumbu horizontal. Variabel independen juga disebut variabel eksplanatori atau variabel prediktor. Variabel dependen, y , berada pada sumbu vertikal. Anda juga dapat merujuk ke nilai y sebagai variabel respons atau variabel yang diprediksi.

Dalam Machine Learning, program komputer yang disebut algoritme menganalisis set data besar dan bekerja mundur dari data tersebut untuk menghitung persamaan regresi linier. Ilmuwan data melatih algoritme pada set data yang diketahui atau diberi label terlebih dahulu, kemudian menggunakan algoritme tersebut untuk memprediksi nilai yang tidak diketahui. Data dalam kehidupan nyata lebih rumit daripada contoh sebelumnya.

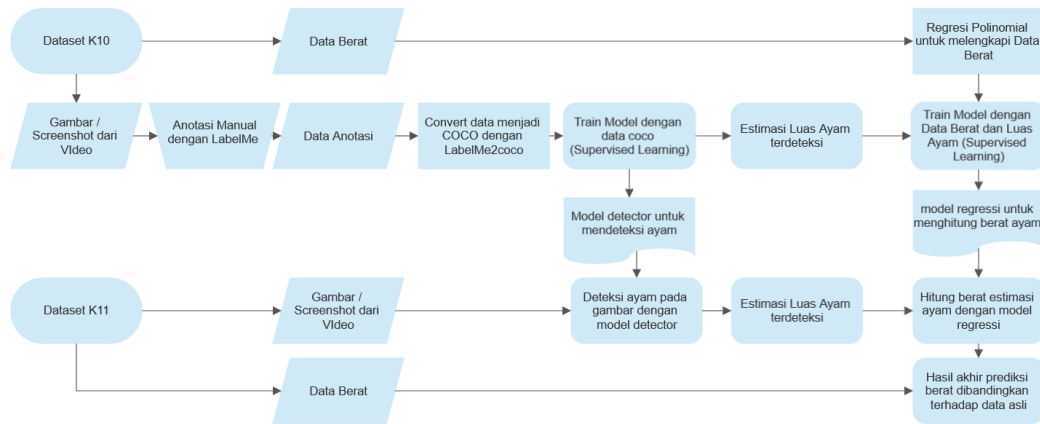
Estimasi berat ayam diperoleh dari relasi antara luas ayam yang diperoleh dengan jumlah dan usia ayam yang terdeteksi. Model regresi berat menggunakan transformasi logaritmik dengan rumusan sebagai berikut.

$$\ln(\hat{w}) = \boldsymbol{\theta}^T \mathbf{x} = \theta_0 + \theta_1 \ln(\beta^2 \overrightarrow{A_m}) + \theta_2 \ln(t) + bias$$

Dengan \mathbf{x} merupakan vektor fitur, $\boldsymbol{\theta}$ merupakan vektor parameter, \hat{w} merupakan prediksi berat, $\overrightarrow{A_m}$ merupakan rata-rata luas segmentasi, dan t merupakan usia ayam.

3 PERANCANGAN & IMPLEMENTASI

3.1 Perancangan



Dataset K10 merupakan dataset yang kami buat berdasarkan video rekaman CCTV. Gambar didapat dari mengambil beberapa Screenshot pada setiap video. Setiap gambar diberikan anotasi manual dengan LabelMe untuk menunjukkan ayam yang dideteksi, menghasilkan data anotasi. Data tersebut diubah menjadi COCO dengan LabelMe2coco, lalu digunakan untuk training model dengan metode supervised learning (data sudah diberi label untuk menentukan output yang diinginkan). Training menghasilkan model detektor serta estimasi luas ayam pada gambar.

Dataset K10 juga memiliki data berat. Regresi polinomial digunakan untuk melengkapi data berat tersebut. Data berat lengkap serta estimasi luas ayam dari training sebelumnya digunakan untuk training model dengan metode yang sama (supervised learning). Training menghasilkan model regresi untuk menghitung berat ayam.

Dataset K11 merupakan dataset yang dibuat oleh kelompok 11, yang akan digunakan sebagai data pengujian model. Model detektor memindai gambar dari dataset, menunjukkan ayam yang terdeteksi di gambar serta estimasi luas ayam. Model regresi menghitung berat berdasarkan estimasi luas yang didapat. Hasilnya prediksi berat oleh model dibandingkan terhadap data berat untuk menentukan performa model, seberapa presisi dan akurat hasilnya terhadap data asli.

3.2 Implementasi

a. Persiapan data training

Gambar / screenshot diambil dari 13 video yang diberikan, dengan tujuan minimal 100 gambar secara total, atau sekitar 8 gambar per video.

Anotasi ayam pada gambar / screenshot menggunakan LabelMe, dengan tujuan anotasi seminimal mungkin 5 ayam per gambar. Jumlah titik per poligon tidak dibatasi, digunakan sebanyak yang diperlukan agar hasil deteksi seakurat mungkin.

Data Json yang didapat dari LabelMe diubah menjadi format COCO dengan LabelMe2coco agar dapat dibaca oleh Training model. [3]

Data berat ayam yang diperlukan adalah pada hari ke-11 sampai ke-24. Karena data yang diberikan tidak lengkap, digunakan regresi polinomial orde 2 untuk mendapatkan data interpolasi berat yang diperlukan.

b. Error pada data yang tidak terdeteksi

Pada pemrosesan data sempat terjadi kesalahan dimana model tidak mendeteksi ayam sama sekali dari data, hal ini dikarenakan penamaan file yang tidak sesuai antara gambar dan coco. Dalam file anotasi LabelMe dan COCO, terdapat nama file gambar sebagai referensi. Ketika nama file gambar dan Json diubah agar seragam, nama file gambar dalam anotasi tidak berubah, sehingga dianggap tidak sesuai. Untuk mengatasi masalah tersebut, nama file referensi dalam Json diubah secara manual agar sesuai dengan nama file gambar. Cara mudahnya adalah dengan mencari "imagePath" dalam file Json untuk mendapat nama file referensi.



3.3 Eksekusi

Pada program yang kami buat, terdapat beberapa susunan folder dan file. Berikut beberapa rincian dari folder dan file yang telah dibuat.

1. K10	22/12/2022 11:52	File folder	
2. K11	22/12/2022 4:58	File folder	
3. train	22/12/2022 23:30	File folder	
4. modeltraining	22/12/2022 13:04	File folder	
5. testing	23/12/2022 0:12	File folder	
6. fullcode	23/12/2022 0:03	File folder	
0. README	22/12/2022 23:31	Text Document	1 KB

Berikut penjelasan dari folder yang tampak pada gambar di atas.

1. Folder K10 merupakan data training yang kelompok 10 buat dengan bersumber dari 13 hari sampel video. Folder ini berisi coco dataset dari tangkapan gambar cctv dan file json dari segmentasi yang telah dilakukan pada aplikasi labelme. Dalam pembuatan coco dataset ini, perlu diperhatikan penamaan file yang sinkron antara gambar dengan file gambar di dalam labelme json yang dibuat.
2. Folder K11 merupakan data testing yang kelompok 11 miliki. Folder ini berisi gambar yang nantinya akan digunakan sebagai data testing dari model training yang kami dapat.
3. Folder train berisi kode program untuk melakukan training data sehingga bisa didapat modelnya. Kode program tersedia dalam file format py dan ipynb.
4. Folder model training berisi file model_final.pth. File ini merupakan model hasil data yang telah dilakukan training. Dengan mempunyai model ini, testing data dari kelompok 11 akan siap dilakukan.
5. Folder testing berisi kode program untuk menjalankan testing data dari kelompok 11. Kode program tersedia dalam file format py dan ipynb.
6. Folder fullcode merupakan kode program keseluruhan dari program yang kami buat. Kami menjalankannya di googlecolab sehingga untuk menjalankan keseluruhan program perlu di-*upload* folder K10 dan K11 ke *drive* dari user yang akan menjalankan program.
7. Terakhir, terdapat readme yang merupakan panduan dalam menjalankan kode program untuk mendapatkan kalkulasi berat ayam per harinya.

Perlu diingat bahwa kode program utama ada pada folder testing, training, dan fullcode. Agar lebih mudah untuk menjalankan pelatihan data dan pengujian data, dapat langsung dijalankan di Google Colaboratory¹ karena memudahkan menjalankan proses pelatihan ataupun pengujian data. Berikut merupakan kode utama dari *fullcode* yang telah dibuat.

¹ Tautan terlampir

Tugas Besar SDR

0. Config

- Install Dependencies
- Drive setup
- Init
- Show the Annotations

1. Train

- Training Config
- Trainer
- Training Analysis

2. Predict

- a. Default Predictor (Train Data K10)

- b. Predictor (Testing Data K11)

- Data 1

- Data 2

- Data 3

- Data 4

- Data 5

- Data 6

- Data 7

- Data 8

- Data 9

- Data 10

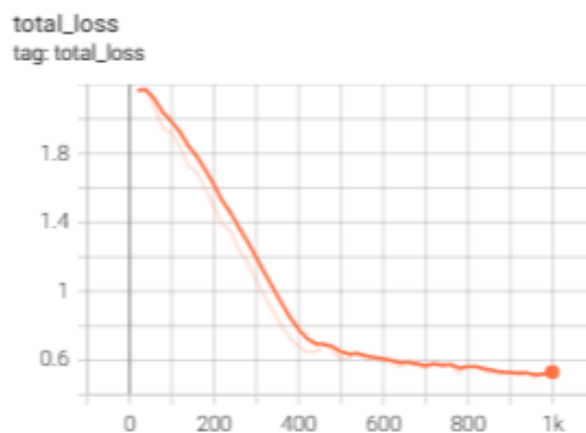
- Data 11

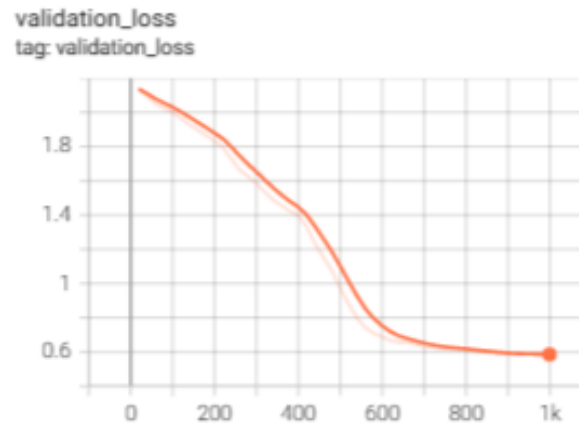
- Data 12

- Data 13

Pada bagian pelatihan, dijalankan kode utama pada bagian *train*. Hasil dari pelatihan ini akan membuahkan model machine learning pada yang terdapat pada folder *modeltraining* sebelumnya. Pada *training analysis*, dilakukan analisis berdasarkan *total loss* dan *validation loss* yang didapat dari TensorBoard seperti berikut.

```
%load_ext tensorboard
%tensorboard --logdir output
```





Sebelumnya telah dilakukan training data dengan 1000 iterasi. Disimpulkan dari grafik di atas, nilai optimal akurasi sebelum overfitting diperoleh ketika training dengan 700 iterasi (ketika loss validasi mendekati elbow). Kemudian dilakukan modifikasi pada kode program training dengan mengubah iterasinya menjadi 700 dan didapatkanlah model machine learning yang akan digunakan untuk pengujian data.

Pada bagian pengujian, digunakan kode pada bagian predict. Pada *default predictor*, ditampilkan 3 sampling visualisasi sebagai berikut.

```
for d in random.sample(dataset_val_dicts, 3):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im)
    v = Visualizer(im[:, :, ::-1],
                  metadata=broilercv_val_metadata,
                  scale=0.8,
                  instance_mode=ColorMode.IMAGE_BW # remove the colors of unsegmented pixels
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    cv2_imshow(v.get_image()[:, :, ::-1])
```

Kemudian diuji data dari kelompok 11 dengan menjalankan kode program bagian *predictor* (Testing Data K11). Ditambahkan filter 80% akurasi agar tampilan visualisasi membuang objek yang terdeteksi kurang dari 80% dengan kode seperti berikut.

```
instances = predictions['instances'].to('cpu')
detections = instances[instances.scores > 0.8]
predictions['instances'] = detections
```

Kemudian, diambil sampling sebanyak 10 gambar per harinya dan dilakukan deteksi berat yang akan dirata-ratakan untuk memperoleh nilai deteksi berat pada

masing-masing harinya (usia ayam). Perlu dibuat kode secara manual dengan menyesuaikan direktori dari gambar dan usia ayam pada tanggal tersebut, sebagai contoh pada hari pertama seperti berikut.

```
#Data Video 1

image_src = np.array(['drive/My Drive/Colab Notebooks/K11/Data1/M1_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M2_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M3_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M4_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M5_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M6_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M7_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M8_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M9_1.png',
                      'drive/My Drive/Colab Notebooks/K11/Data1/M10_1.png'])

age = 11
```

Setelah itu, juga ditampilkan pada *line command* hasil berat per gambar dan rata-rata berat di hari itu beserta dengan visualisasi segmentasi, luas ayam, dan bobot dengan kode seperti berikut.

```
for i in range(len(image_src)):
    im = cv2.imread(image_src[i])
    outputs = predictor(im, age)
    v = WeightVisualizer(im[:, :, ::-1],
                          metadata=broilercv_val_metadata,
                          scale=0.8,
                          age=f'{age}')
    instances = outputs['instances']
    detections = instances[instances.scores > 0.9]

    w = v.draw_instance_predictions(instances.to("cpu"))
    u = v.draw_weight_predictions(outputs)

    cv2_imshow(u.get_image()[:, :, ::-1])

    print(f'perhitungan berat pada gambar ke-{i+1} adalah {outputs["weight"]}')
    weightData[i] = outputs["weight"]

print("===== Memulai perhitungan berat rata-rata =====")
print(weightData)
print("Berat rata-rata: ", np.average(weightData))
```

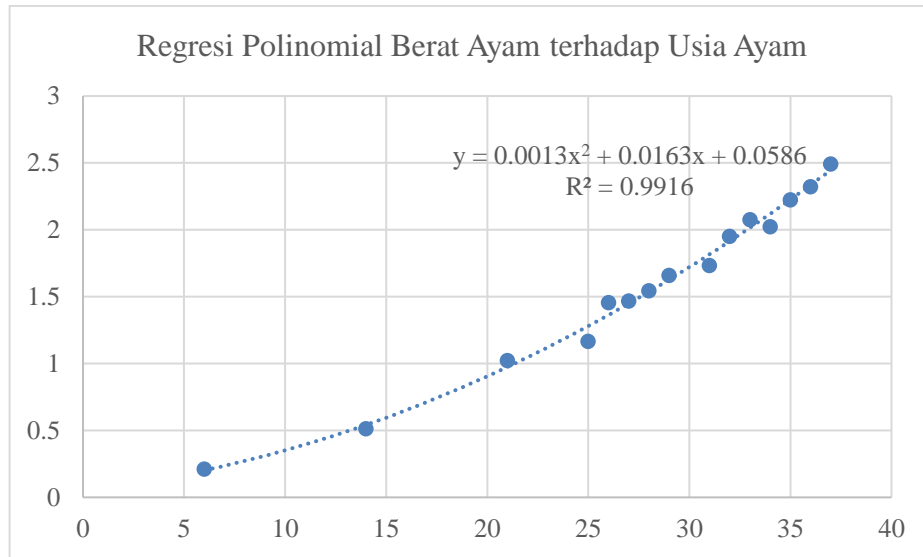
Setelah menjalankan pengujian ini, dapat dicatat berat rata-rata ayam di tiap harinya sehingga dapat dianalisis lebih lanjut error terhadap berat sebenarnya (data diberikan).

4 PENGUJIAN & ANALISIS

Pada bagian akan dijelaskan mengenai pengujian beserta analisisnya. Tahapan pengujian data dilakukan dengan memanfaatkan dataset kelompok setelah kami, yaitu K11. Pada pengujian model yang kami peroleh, digunakan dataset berupa 10 *frame sampling* (gambar) masing-masing harinya dan akan dihitung berat tiap *frame*/gambar tersebut dengan kode program yang ada. Tidak berhenti di situ, ditambahkan juga kode program untuk merata-ratakan hasil deteksi berat dari 10 gambar tersebut. Di akhir program akan didapat perhitungan berat rata-rata ayam setiap harinya. Sebagai pembanding (acuan), digunakan hasil data berat secara langsung yang telah diberikan oleh pihak kandang ayam sebagai berikut.

Tanggal Timbang	Hari	Berat (kg)
21/08/2022	6	0.2108333
29/08/2022	14	0.5116667
05/09/2022	21	1.0216667
09/09/2022	25	1.1644501
10/09/2022	26	1.4552
11/09/2022	27	1.4664831
12/09/2022	28	1.5430357
13/09/2022	29	1.659
15/09/2022	31	1.733
16/09/2022	32	1.951
17/09/2022	33	2.076
18/09/2022	34	2.023
19/09/2022	35	2.2222222
20/09/2022	36	2.322
21/09/2022	37	2.492

Namun, data berat yang diberikan di atas masih kurang karena tidak tersedia untuk semua usia. Kemudian, dilakukanlah pendekatan regresi polinomial dari berat ayam terhadap usia ayam untuk mendapatkan berat dari usia yang diinginkan sebagai pembanding. Berikut merupakan hasil regresi polinomial yang diperoleh dari data tabel di atas.



Dari persamaan regresi yang diperoleh di atas, didapatkan nilai perhitungan dari berat ayam sebenarnya pada tiap usia yang dibutuhkan sebagai pembanding (nilai referensi) sebagai berikut.

Data Video	Hari	True Weight (kg)
26/08/2022	11	0.3952
27/08/2022	12	0.4414
28/08/2022	13	0.4902
29/08/2022	14	0.5416
31/08/2022	16	0.6522
01/09/2022	17	0.7114
02/09/2022	18	0.7732
03/09/2022	19	0.8376
04/09/2022	20	0.9046
05/09/2022	21	0.9742
06/09/2022	22	1.0464
07/09/2022	23	1.1212
08/09/2022	24	1.1986

Tahap selanjutnya, akan dilakukan perhitungan berat dengan model machine learning yang telah didapat dari proses pelatihan data. Untuk melakukan deteksi dan segmentasi individu ayam pada tiap *frame*/gambar yang tersedia, digunakan Mask R-CNN dengan *backbone* ResNet-50 FPN ditambah bantuan *framework* Detectron dan Pustaka PyTorch.

```
!pip install -U torch torchvision
!pip install git+https://github.com/facebookresearch/fvcore.git
```



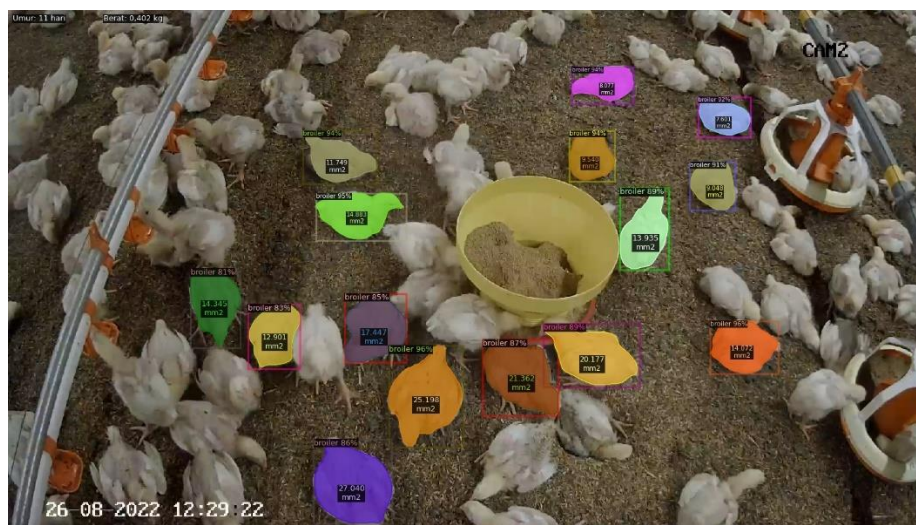
```
!python -m pip install pyyaml==5.1
import sys, os, distutils.core
!git clone 'https://github.com/facebookresearch/detectron2'
dist = distutils.core.run_setup("./detectron2/setup.py")
!python -
m pip install {' '.join([f"'{x}'" for x in dist.install_requires])}
sys.path.insert(0, os.path.abspath('./detectron2'))
```


Pada segmentasi di atas, dilakukan filter 50% keakuratan dengan kode program seperti berikut.

```
predictions = super().__call__(original_image)

instances = predictions['instances'].to('cpu')
detections = instances[instances.scores > 0.5]
predictions['instances'] = detections
```

Supaya hasil deteksi berat lebih akurat, kami menggunakan filter 80%. Berikut ditampilkan juga *frame*/gambar dengan persentase deteksi objek yang lebih dari 80% (dengan data *frame*/gambar yang sama).



Pada deteksi dan segmentasi di atas telah dilakukan perhitungan area serta estimasi berat ayam. Estimasi berat ayam diperoleh dari relasi antara luas ayam yang diperoleh dengan jumlah dan usia ayam yang terdeteksi. Model regresi berat menggunakan transformasi logaritmik dengan rumusan sebagai berikut.

$$\ln(\hat{w}) = \theta^T x = \theta_0 + \theta_1 \ln(\beta^2 \vec{A}_m) + \theta_2 \ln(t) + bias$$

Dengan x merupakan vektor fitur, θ merupakan vektor parameter, \hat{w} merupakan prediksi berat, \vec{A}_m merupakan rata-rata luas segmentasi, dan t merupakan usia ayam. Kemudian, agar dapat ditampilkan hasil berat rata-rata pada program, dijalankan kode berikut.

```
image_src = np.array(['drive/My Drive/Colab Notebooks/K11/Data1/M1_1.pn
g',.....]) #direktori gambar
age = 11 #usia ayam

weightData = np.zeros(len(image_src))
```

```

for i in range(len(image_src)):
    im = cv2.imread(image_src[i])
    outputs = predictor(im, age)
    v = WeightVisualizer(im[:, :, ::-1],
                          metadata=broilercv_val_metadata,
                          scale=0.8,
                          age=f'{age}'
    )
    instances = outputs['instances']
    detections = instances[instances.scores > 0.9]

    w = v.draw_instance_predictions(instances.to("cpu"))
    u = v.draw_weight_predictions(outputs)

    cv2_imshow(u.get_image()[:, :, ::-1])

    print(f'perhitungan berat pada gambar ke-
{i+1} adalah {outputs["weight"]}')
    weightData[i] = outputs["weight"]

print("===== Memulai perhitungan berat rata-rata =====")
print(weightData)
print("Berat rata-rata: ", np.average(weightData))

```

Diperoleh hasil berat rata-rata pada tanggal 26 Agustus 2022 seperti berikut.

```

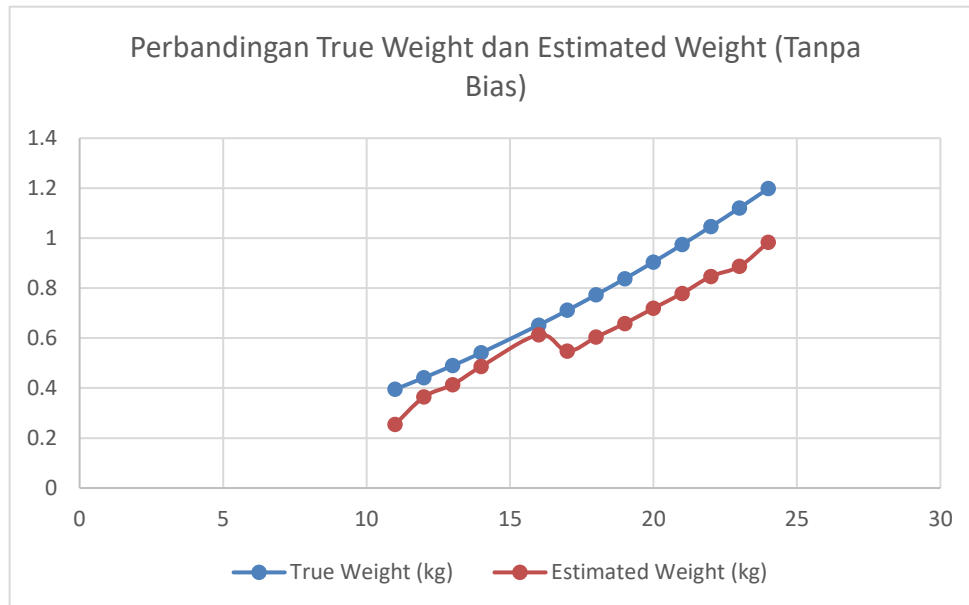
perhitungan berat pada gambar ke-10 adalah 0.3977400181418048
===== Memulai perhitungan berat rata-rata =====
[0.40182903 0.40290672 0.40280045 0.40524428 0.39885042 0.39940343
 0.40805192 0.40733677 0.40462098 0.39774002]
Berat rata-rata: 0.4028784011969

```

Secara manual, dijalankan kode program di atas untuk seluruh hari yang disediakan dengan menyesuaikan *image_src* dan *age*. Setelah dilakukan pengolahan data, diperoleh *estimated weight* dari model machine learning beserta errornya sebagai berikut.

Data Video	Hari	True Weight (kg)	Estimated Weight (kg)	Error
26/08/2022	11	0.3952	0.25455	14%
27/08/2022	12	0.4414	0.364679	8%
28/08/2022	13	0.4902	0.41398596	8%
29/08/2022	14	0.5416	0.486685	5%
31/08/2022	16	0.6522	0.613288	4%
01/09/2022	17	0.7114	0.547663412	16%

02/09/2022	18	0.7732	0.603717678	17%
03/09/2022	19	0.8376	0.658610569	18%
04/09/2022	20	0.9046	0.719756716	18%
05/09/2022	21	0.9742	0.778966519	20%
06/09/2022	22	1.0464	0.846982478	20%
07/09/2022	23	1.1212	0.887289281	23%
08/09/2022	24	1.1986	0.98335919	22%
Average				15%



Hasil error yang diperoleh cukup besar dan coba ditambahkan bias model yang ada (karena terlihat ada offset pada grafik perbandingan di atas). Perhitungan bias dilakukan dengan merata-ratakan selisih dari *true weight* dan *estimated weight*² sehingga diperoleh nilai bias sebesar 0.149. Kemudian, dilakukan penyesuaian kode program dengan penambahan bias pada bagian di bawah ini.

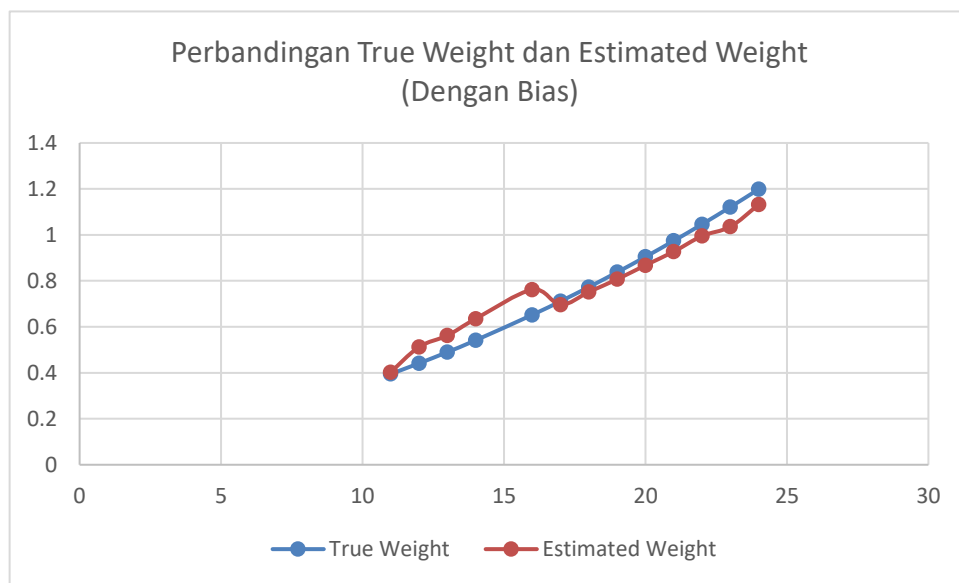
```
...
x_in = np.array([1.0, np.log(age), np.log(areas.numpy().mean())])
theta = np.array(self.cfg.WEIGHT.THETA, dtype=x_in.dtype)
y = theta.T @ x_in
predictions["weight"] = np.exp(y).item() + 0.148328
return predictions
```

Berikut ditampilkan *estimated weight* setelah dilakukan penambahan bias.

Dengan Bias

² Pengolahan data secara lengkap terlampir pada file Excel

Data Video	Hari	True Weight	Estimated Weight	Error
26/08/2022	11	0.3952	0.402878401	1%
27/08/2022	12	0.4414	0.513007352	7%
28/08/2022	13	0.4902	0.562313964	7%
29/08/2022	14	0.5416	0.635013286	9%
31/08/2022	16	0.6522	0.761616063	11%
01/09/2022	17	0.7114	0.695991412	2%
02/09/2022	18	0.7732	0.752045678	2%
03/09/2022	19	0.8376	0.806938569	3%
04/09/2022	20	0.9046	0.868084716	4%
05/09/2022	21	0.9742	0.927294519	5%
06/09/2022	22	1.0464	0.995310478	5%
07/09/2022	23	1.1212	1.035617281	9%
08/09/2022	24	1.1986	1.13168719	7%
Average Error				5%



Pada tabel di atas tampak bahwa rata-rata error keseluruhan yang diperoleh sudah cukup bagus, dengan error rata-rata sebesar 5%.

5 KESIMPULAN & HIKMAH

5.1 Kesimpulan

Berikut merupakan beberapa kesimpulan yang dapat diambil dari proses pengerjaan tugas besar ini:

1. Telah dipelajari keilmuan di bidang machine learning. Pada keseluruhan pengerjaan tugas besar ini, kami telah mengimplementasikan algoritma machine learning yang dapat memprediksi berat ayam dengan Pustaka Detectron untuk mendeteksi ayam dari gambar atau rekaman CCTV.
2. Telah dibandingkan model machine learning yang telah diperoleh. Dilakukam iterasi data latih sebanyak 700 kali yang merupakan nilai optimal akurasi sebelum overfitting. Hal ini diperoleh dari analisis pada TensorBoard. Kemudian, telah dilakukan uji model pada data K11. Telah dilakukan juga pengaplikasian filter kepercayaan dengan nilai 80%. Nilai ini memberikan hasil yang cukup akurat pada rata-rata berat ayam per harinya.
3. Telah diberikan sebuah model yang mampu memprediksi berat ayam. Setelah dilakukan proses pelatihan dan diperoleh model menggunakan 2-layer model, metode ini menghasilkan prediksi yang cukup baik dengan error relatif rendah, yaitu sebesar 5%.

5.2 Hikmah

Berikut merupakan pembelajaran dan wawasan yang kami dapat selama melaksanakan tugas ini:

1. Preparasi data adalah bagian paling penting dari machine learning, karena input data akan sangat mempengaruhi output yang dihasilkan. Keakurasian input data akan mempengaruhi akurasi output model, sehingga perlu dipastikan data input sudah cukup akurat. Begitu juga dengan tingkat kesulitan. Data yang relatif sulit untuk dianotasi seperti ayam yang bertumpukan adalah data yang bagus bagi model, karena bisa mengajari model untuk mendeteksi ayam yang bertumpukan juga di fase testing.
2. Analisa dan *troubleshooting* juga penting dalam keberlangsungan proses machine learning. Pada percobaan ini sempat ada kesalahan dalam penamaan file dalam json, sehingga perlu mengedit data json secara langsung untuk memperbaiki kesalahan nama tersebut.
3. Teamwork dan komunikasi sangat penting untuk membentuk kerjasama yang baik dalam kelompok. Akses kode dan dokumen bersama secara

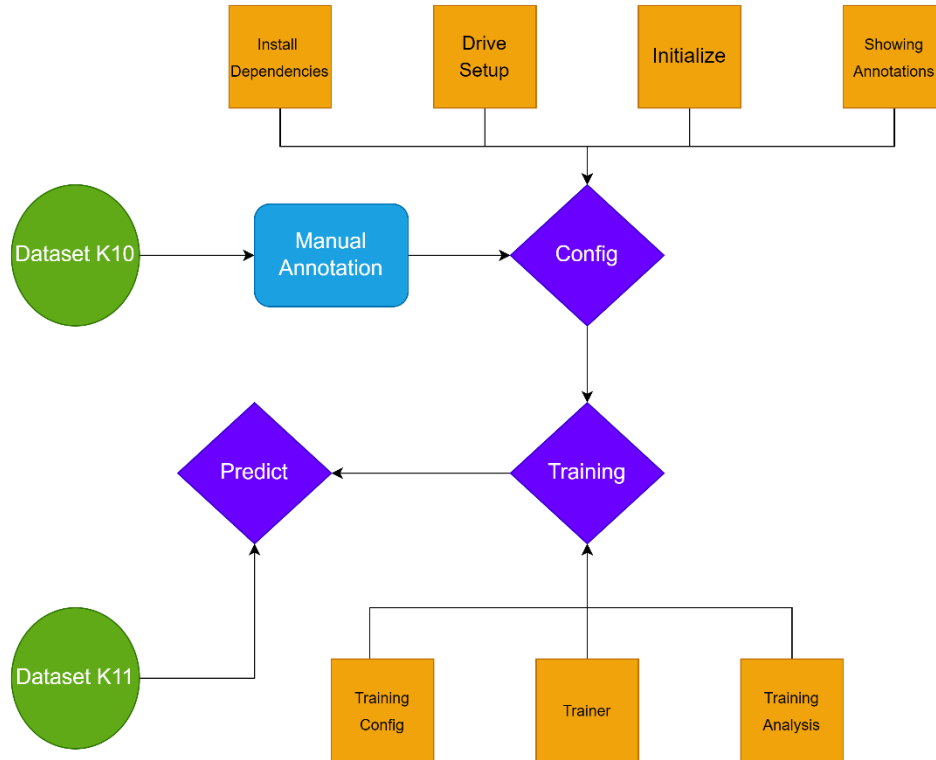
online sangat membantu untuk memudahkan kerjasama. Time Management atau disiplin waktu juga sangat penting agar anggota kerja kelompok tidak melalaikan tugasnya.

6 DAFTAR PUSTAKA

- [1] Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., & Girshick, R. (2019). Detectron2. Retrieved from <https://github.com/facebookresearch/detectron2>
- [2] Abdulla, W. (2017). Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. GitHub Repository. Retrieved from https://github.com/matterport/Mask_RCNN
- [3] Fatih. Labelme to Coco (labelme2coco) data transformation library written using Python. GitHub Repository. Retrieved from <https://github.com/fcakyon/labelme2coco>

LAMPIRAN

A. Flowchart Metodologi



B. Source Code di Google Colab

https://colab.research.google.com/drive/1LV7PCZXJpuGb5-SmB7o_yyHnoXAQ0l4?usp=sharing

C. Excel Sheet untuk perhitungan regresi data berat serta bias prediksi berat

https://itbdsti-my.sharepoint.com/:x:/g/personal/13320003_mahasiswa_itb_ac_id/EbPs7xtO0tJBudXF7WO-hAsBnKwixrdrdvCZpObTroRIKg?e=mlvANG

D. Salinan online dokumen laporan tugas besar

https://itbdsti-my.sharepoint.com/:w:/g/personal/13320003_mahasiswa_itb_ac_id/EfPD2j_oUhhMmVi-jtU8zToBMKyx37MfTrl09piXlkZ9YA?e=QYMwGD

MANAJEMEN PROYEK

			Bobot	Waktu (minggu)							
No	Pekerjaan	Pekerja	(%)	1	2	3	4	5	6	7	8
1	Diskusi alur pengerjaan dan bagi tugas	A, B, C	5								
2	Anotasi data video ayam	A, B, C	10								
3	Konversi data anotasi menjadi data yang dapat dilatih (coco dataset)	B, C	15								
4	Pelatihan data dan mendapatkan model machine learning	A, B	15								
5	Pengujian data (program prediksi berat)	A, B, C	20								
5	Analisis data dan akurasi	A, C	15								
6	Pembuatan laporan tugas besar	A, B, C	15								

Pekerja	Nama	Jumlah Bobot (%)
A	Fernando Alvirianda Yoga	80
B	Jihad Zakki Darajad	80
C	Jacky Wu	80