

Deteksi Jenis Sampah Menggunakan Metode Transfer Learning YOLO-V8

Nandang Suwela¹, Muhammad Zidan Hedriyadi²

^{1,2}Program Studi Teknik Informatika
Universitas Indraprasta PGRI

e-mail : nandang.s@gmail.com¹, zidanhendriadi45@gmail.com²

ABSTRAK

Permasalahan besar yang dihadapi oleh hampir semua negara termasuk Indonesia adalah masalah sampah, dimana jumlah dan jenis sampah yang beredar terus meningkat setiap tahunnya seiring dengan bertambahnya jumlah penduduk. Hal ini mengakibatkan penambahan peningkatan jenis dan jumlah timbunan sampah. Pemilahan jenis sampah menjadi tantangan utama dalam pengelolaan sampah karena keberadaan sampah yang ada di lingkungan masyarakat selalu dalam keadaan tercampur atau tidak terpilah. Pemilahan sampah yang baik sangat diperlukan sehingga dapat membedakan sampah yang dapat didaur ulang dan meminimalkan sampah yang tidak dapat didaur ulang. Penelitian ini merencanakan model kecerdasan buatan Convolutional Neural Network (CNN) untuk membantu mengidentifikasi sampah melalui gambar, dan untuk meningkatkan akurasi peneliti menggunakan teknik transfer learning dari modul You Only Look Once (YOLO). YOLO-V8 merupakan versi terbaru dari model YOLO yang telah dikembangkan untuk meningkatkan akurasi dan kecepatan deteksi objek dengan akurasi tinggi. Penggunaan metode transfer learning pada YOLO-V8 memungkinkan untuk menyesuaikan model yang sudah ada dengan dataset baru yang spesifik, seperti dataset sampah, tanpa perlu melatih model dari awal, sedangkan untuk memudahkan analisa secara visual, digunakan aplikasi web. Perhitungan mAP didasarkan pada perhitungan IoU, pada mAP50, hasil prediksi memiliki minimal terdapat 50% bagian kotak tumpang tindih terhadap kotak target, sehingga model dapat mencapai nilai 0.88. Sedangkan pada mAP50-95 hasil prediksi minimal memiliki 95%, untuk nilai ini model mampu mencapai angka 0.70. Penelitian menggunakan dataset yang sangat terbatas, untuk penelitian selanjutnya dapat dilakukan penambahan dataset baik dalam jumlah kelas maupun jumlah dataset itu sendiri. Integrasi dengan sistem lain juga bisa dilakukan, seperti perhitungan berat sampah dan pelaporan mengenai pendapatan yang didapat dari menjual sampah untuk meningkatkan penghasilan masyarakat.

Kata Kunci : Deteksi Sampah, Transfer Learning, YOLO-8

I. PENDAHULUAN

Permasalahan besar yang dihadapi oleh hampir semua negara termasuk Indonesia adalah masalah sampah, dimana jumlah dan jenis sampah yang beredar terus meningkat setiap tahunnya seiring dengan bertambahnya jumlah penduduk. Kegiatan pada sektor produksi dan peningkatan kebutuhan konsumsi masyarakat berdampak pada penambahan peningkatan jenis dan jumlah timbunan sampah. Berdasarkan data dari Sistem Informasi Penanggulangan Sampah Nasional (SIPSN) yang dikutip dari situs website resmi SIPSN per 2023, terdapat timbunan sampah Indonesia sebesar 19,07 juta ton/tahun dan ada sekitar 32,51% atau 6,2 juta ton/tahun sampah yang tidak terkelola[1]. Pemilahan jenis sampah menjadi tantangan utama dalam pengelolaan sampah karena keberadaan sampah yang ada di lingkungan masyarakat selalu dalam keadaan tercampur atau tidak terpilah. Pemilahan sampah yang baik sangat diperlukan sehingga dapat membedakan sampah yang dapat didaur ulang dan meminimalkan sampah yang tidak dapat didaur ulang. Teknologi dapat berperan besar dalam proses pemilahan sampah, salah satunya adalah dengan menggunakan kecerdasan buatan dan *computer vision*. Penelitian yang telah dilakukan diantaranya adalah penelitian yang menggunakan algoritma KNN (*k-nearest neighbor*) dengan

menggunakan tiga sampel jenis sampah yang relatif serupa dan tidak dilakukan secara *real time* pada lingkungan terbuka. Akurasi dari penelitian ini sebesar 98,33%[2]. Penelitian lain menggunakan metode *convolutional neural network (CNN)* dalam mengidentifikasi jenis sampah dengan menambah hyperparameter mereka berhasil meningkatkan akurasi dari 67,6% menjadi 91,2%[3]. Setelah mengevaluasi penelitian terdahulu tersebut, peneliti dalam penelitian ini merencanakan model kecerdasan buatan *Convolutional Neural Network (CNN)* untuk membantu mengidentifikasi sampah melalui gambar, dan untuk meningkatkan akurasi peneliti menggunakan teknik transfer learning dari modul *You Only Look Once (YOLO)* yang memiliki akurasi tinggi, sedangkan untuk memudahkan analisa secara visual digunakan aplikasi web dan dapat memberikan hasil yang lebih baik sehingga dapat memberikan informasi yang lebih efisien.

II. LANDASAN TEORI

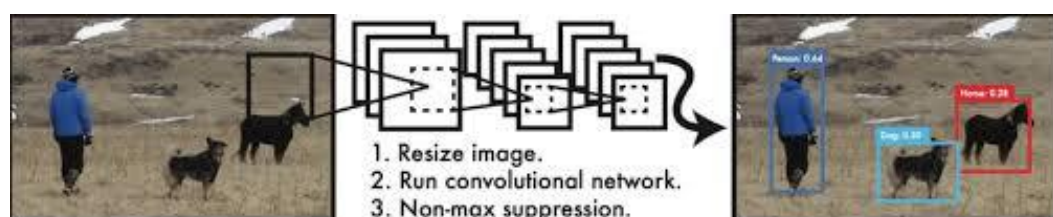
2.1 Sampah

Menurut *World Health Organization (WHO)* sampah adalah sesuatu yang tidak digunakan, tidak dipakai, tidak disenangi atau sesuatu yang dibuang yang berasal dari kegiatan manusia dan tidak terjadi dengan sendirinya. Sampah merupakan material sisa baik dari hewan, manusia, maupun tumbuhan yang tidak terpakai lagi dan dilepaskan kedalam dalam bentuk padat, cair, maupun gas yang selalu ada dalam kehidupan sehari-hari [4]. Sementara itu menurut (Noelaka, 2008) dalam [5] bahwa sampah dibagi menjadi tiga bagian yakni:

- Sampah organik yaitu sampah yang berasal dari sisa sayuran, sisa daging, sampah taman, daun-daunan dan lain-lain yang mudah terurai melalui proses alami.
- Sampah anorganik yaitu limbah yang tidak mudah rusak dan sulit terurai seperti plastik, logam, karet, kaca, bahan bangunan bekas, dsb.
- Sampah B3 (bahan berbahaya beracun) yaitu limbah ini berasal dari bahan kimia organik, anorganik, atau logam berat yang umumnya berasal dari limbah industri. Perlu penanganan khusus untuk limbah B3 ini, dan tidak boleh tercampur dengan sampah organik dan anorganik.

2.2 *You Only Look Once (YOLO)*

YOLO adalah algoritma yang melakukan deteksi objek secara real-time yang mulai diperkenalkan pada tahun 2015 oleh Joseph Redmon, dkk. Mereka melakukan penelitian dengan judul "*You Only Look Once: Unified, Real-Time Object Detection*". YOLO dapat menerapkan satu jaringan saraf ke gambar penuh dan kemudian membagi gambar menjadi beberapa wilayah yang selanjutnya dapat diprediksi melalui kotak terikat dan probabilitas. YOLO digemari karena selain mampu melakukan deteksi secara *real time* tetapi juga dapat menghasilkan data yang akurat. Metode yang digunakan model YOLO yaitu setelah mendeteksi semua objek yang telah terdefinisi lalu akan membuat kotak pembatas di sekelilingnya. Untuk memprediksi kotak pembatas serta probabilitas kelas untuk semua objek yang terdapat dalam gambar, YOLO memanfaatkan CNN. Model ini diberi nama YOLO karena dengan hanya sekali melihat gambar dapat mengidentifikasi objek dan posisinya dengan bantuan kotak pembatas. YOLO terus mengalami perkembangan dari waktu ke waktu hingga saat ini, versi terakhir yang dipublikasikan adalah versi 10.



Gambar 1 The YOLO Detection System
Sumber : pjreddie.com

2.3 Deep Learning

Setelah *Support Vector Machine* (SVM), *multilayer perceptron* (MLP), jaringan saraf buatan (ANN), dan jaringan saraf dangkal lainnya menjadi populer, kemudian pada awal tahun 2000-an *deep learning* mulai diperkenalkan. Menurut banyak peneliti, itu merupakan bagian dari pembelajaran mesin (*Machine Learning*), yang pada gilirannya merupakan bagian dari kecerdasan buatan (AI). *Deep learning* memungkinkan model komputasi yang terdiri dari banyak lapisan pemrosesan untuk memahami representasi data dengan tingkat abstraksi yang berbeda. Dengan menggunakan algoritma *backpropagation*, *deep learning* menemukan struktur kompleks dalam kumpulan data yang sangat besar, yang memungkinkan mesin untuk menyesuaikan parameter internalnya untuk membuat representasi untuk setiap lapisan berdasarkan representasi dari lapisan sebelumnya [6].

Ada tiga lapisan dalam *deep learning*. Lapisan masukan berfungsi untuk menerima data, lapisan output berfungsi untuk menghasilkan hasil pengolahan data, dan lapisan tersembunyi berfungsi untuk mengekstrak pola di dalam data. JST yang dalam berbeda dari JST superfisial (JST dengan lapisan tersembunyi tunggal) karena memiliki banyak lapisan tersembunyi dan dapat melakukan tugas yang lebih rumit. Saat data ditransfer dari satu lapisan tersembunyi ke lapisan berikutnya, fitur sederhana digabungkan kembali dan dikomposisi ulang menjadi fitur yang kompleks. Sederhananya, DL bekerja dengan baik dengan data yang tidak terstruktur dan lebih akurat daripada ML; namun, itu membutuhkan banyak data pelatihan, perangkat lunak, dan perangkat keras yang mahal [7].

Algoritma *deep learning* telah digunakan untuk berbagai macam tujuan. Misalnya, algoritma kategori pemrosesan bahasa dapat digunakan untuk mengidentifikasi pola yang ada dengan menerima input berupa kata-kata atau kalimat, dalam kasus ini, input yang diterima adalah angka dari piksel gambar, yang kemudian diidentifikasi sebagai kotak dan penandaan objek, sehingga kasus ini masuk sebagai *computer vision*.

2.4 Computer Vision

Subbidang kecerdasan buatan (AI) yang dikenal sebagai *computer vision* berfokus pada pembuatan sistem komputer yang memiliki kemampuan untuk "melihat" dan memahami dunia fisik dengan cara yang mirip dengan manusia. *Computer vision* mengajarkan mesin untuk melakukan hal-hal seperti membedakan objek, seberapa jauh jaraknya, apakah bergerak, dan mengidentifikasi apakah ada kesalahan pada gambar.

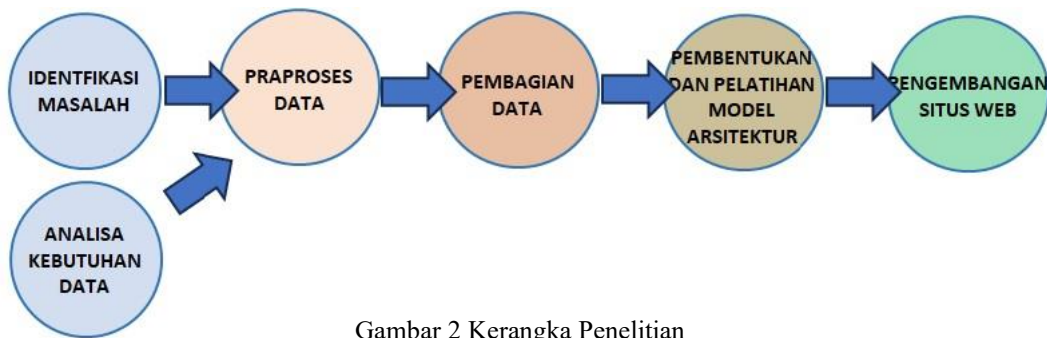
Model matematika untuk pemrosesan gambar dikenal sebagai algoritma *computer vision*. Di bidang kecerdasan buatan, pengertian ini lebih umum dan memiliki manfaat teknis yang signifikan. Algoritma pada dasarnya mendapatkan informasi dari gambar atau data. Jika manusia dihadapkan pada gambar yang berbeda, mereka dapat langsung menganalisisnya sesuai dengan pemahaman mereka sendiri; namun, jika analisis gambar dilakukan menggunakan komputer, metode interpretasi yang lebih banyak akan dihasilkan, yang secara keseluruhan lebih rumit. Algoritma *computer vision* dibuat sebelum model probabilitas fisik digunakan untuk menginterpretasikan gambar. Dengan menggunakan metode ini, komputer dapat secara akurat mengidentifikasi gambar berdasarkan fungsinya dan membuat model tiga dimensi dan simulasi prediktif berdasarkan gambar tersebut untuk memberikan hasil yang diperlukan [8].

2.5 Python

Python merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum dan pertama kali dirilis pada tahun 1991. *Python* saat ini adalah bahasa yang sangat populer. *Python* juga merupakan bahasa pemrograman yang serbaguna, *Python* dapat digunakan untuk *deep learning* dan *machine learning*. *Python* memiliki perpustakaan yang lengkap dan sintaks yang mudah dibuat. Anda dapat menggunakan IDE seperti VsCode, sublime text, PyCharm, atau IDE online seperti Jupyter Notebook atau Google Colab untuk membuat kode *Python* [9]. *Python* adalah bahasa pemrograman yang digunakan oleh Google, Instagram, dan Netflix [10].

III. METODE PENELITIAN

3.1 Kerangka Penelitian



Gambar 2 Kerangka Penelitian
Sumber : penulis

Tahapan pertama dalam penelitian ini fokus pada permasalahan pengelolaan sampah di Indonesia, tahapan ini mencakup pengenalan variabel-variabel yang relevan yang akan digunakan dalam pendeteksian jenis sampah. Setelah memahami masalah dan membuat ide dan model untuk menyelesaikannya, langkah selanjutnya adalah mengumpulkan data. Data yang dibutuhkan mencakup informasi tentang gambar sampah yang terdiri dari berbagai jenis bahan, seperti styrofoam, plastik, kaleng, dan kertas. Data ini diambil dari dataset yang dimiliki Alistrair King di Kaggle.com.

Ada kemungkinan bahwa data yang dikumpulkan dari Kaggle.com tidak selalu dalam format yang dibutuhkan oleh peneliti. Oleh karena itu, untuk membuat data siap untuk diolah lebih lanjut, proses diperlukan pra-pemrosesan. Pra-pemrosesan data ini mencakup berbagai langkah. Ini termasuk memilih kategori yang dibutuhkan, memilih gambar yang dapat digunakan, dan menentukan jumlah data yang ingin digunakan. Data yang telah dibersihkan dan siap untuk langkah analisis selanjutnya dihasilkan sebagai hasil dari proses pra-pemrosesan ini.

Hasil data yang telah diolah akan dibagi menjadi tiga bagian, masing-masing dengan empat kategori yang diperlukan untuk pelatihan model. Tiga bagian tersebut adalah train, test, dan val. Empat kategori yang digunakan adalah styrofoam, plastik, kaleng, dan kertas.

Untuk sistem pendeteksian, pembentukan model menggunakan YOLOv8 yang telah dilatih sebelumnya. Dimulai dengan menyiapkan dataset dan arsitektur model yang akan digunakan, serta library dan lingkungan kerja yang akan digunakan. Setelah itu, peneliti memasukkan model yang telah dilatih sebelumnya ke dalam lembar kerja, yang dapat ditemukan di situs resmi YOLO, dan menentukan alamat dataset yang akan dituju. Peneliti menggunakan Google Colab untuk membuat model ini.

Proses pelatihan pada model YOLO yang telah dilatih sebelumnya dimulai dengan memasukkan alamat dataset dan menetapkan jumlah epochs dan ukuran gambar yang akan dilatih. Setelah pelatihan selesai, validasi model dilakukan dengan menggunakan berat yang tersedia pada model YOLO. Dengan demikian, peneliti dapat mengetahui parameter yang digunakan dari file.txt yang dibuat dari pelatihan.

Pengembangan situs web yang berfungsi sebagai antarmuka untuk menampilkan hasil deteksi jenis sampah kepada pengguna adalah bagian dari tahap ini. Pengembangan ini mencakup desain antarmuka pengguna yang efisien dan responsif, integrasi dengan backend untuk mengelola dan menampilkan data deteksi sampah secara dinamis, dan implementasi fitur tambahan seperti filter data dan visualisasi grafis. Dengan menjamin keamanan, kecepatan, dan ketersediaan situs web di berbagai perangkat, tujuan utamanya adalah meningkatkan pengalaman pengguna.

3.2 Hardware dan Software

Penelitian menggunakan PC dengan spesifikasi prosesor core i8 dengan RAM sebesar 16 GB dan harddisk 1 TB. Software yang digunakan seperti Tensorflow sebuah open source platform untuk machine learning yang menyediakan berbagai API pada berbagai bahasa pemrograman yang memudahkan proses pembuatan model AI. Komputasi yang dirumuskan dengan Tensorflow dapat dieksekusi dengan sedikit atau tanpa modifikasi pada berbagai sistem yang heterogeny. Kemudian digunakan juga You Only Look Once (YOLO) sebuah algoritma mutakhir untuk mendeteksi objek secara real-time YOLO digunakan karena tidak hanya menghasilkan data yang akurat, tetapi juga mampu melakukannya secara real-time. Sementara itu bahasa pemrograman yang digunakan adalah Python yang merupakan bahasa yang sangat populer saat ini. Selain itu, Python juga merupakan bahasa pemrograman yang serbaguna, misalnya Python dapat digunakan untuk Machine Learning dan Deep Learning.

IV. ANALISIS DAN HASIL PERANCANGAN

4.1 Pembahasan Algoritma

4.1.1 Akuisisi Data

Penulis menggunakan dataset gambar sampah yang diambil dari Roboflow.com. Penggunaan dataset tersebut berdasar pada hasil pencarian yang telah dilakukan Roboflow Universe yang telah dirilis pada bulan Maret 2024, dataset tersebut berjudul Waste-Detection. Untuk menambah variasi jenis sampah yang ditemukan, penulis menambahkan satu jenis sampah lagi yang berasal dari dataset Trash Detection yang dirilis pada bulan Mei 2024, sehingga dataset menjadi sebagai berikut :

Tabel 1 Daftar Jenis Sampah

No	Jenis Sampah	Jumlah Gambar	Set Pelatihan	Set Validasi	Set Pengujian
1	Metal	505	378	81	46
2	Plastik	666	491	102	73
3	Kaca	542	417	78	47
4	Kardus	529	424	48	57
5	Styrofoam	528	386	99	43

4.1.2 Praproses Data

Untuk menyesuaikan input yang diterima oleh model, tahapan praproses data diperlukan sebelum dataset yang sudah dikumpulkan dapat digunakan untuk proses pelatihan. Teknik yang digunakan dalam tahapan ini juga dapat mempengaruhi kualitas segmentasi gambar yang dihasilkan.

4.1.2.1 Pembagian Dataset

Dimulai dengan membagi dataset menjadi tiga bagian, yaitu pelatihan, validasi, dan pengujian, penelitian ini menggunakan 2770 gambar data. Semakin banyak dataset yang dilatih, semakin banyak model yang belajar fitur sampel untuk melakukan prediksi pada dataset pengujian. Proporsi pembagian dataset pelatihan adalah 70%, dataset validasi 20%, dan dataset pengujian 10%.

4.1.2.2 Anotasi Objek

Untuk melakukan anotasi objek sampah, harus tahu bentuknya terlebih dahulu. Berikut ini adalah sampel objek sampah berdasarkan kelasnya:



Gambar 3 Sampel Gambar Sampah
 Sumber: Penulis

Selanjutnya, dataset yang telah dikumpulkan diunggah ke penyimpanan cloud Roboflow untuk proses anotasi. Anotasi dilakukan dengan memberikan kotak pembatas pada objek sampah pada gambar. Setelah anotasi selesai, Roboflow akan membuat folder label yang menyimpan lokasi koordinat titik kotak pembatas untuk setiap gambar dalam format teks file (txt). Format kotak pembatas YOLO menggunakan nilai yang telah dinormalisasi untuk setiap gambar. Ini adalah perubahan nilai gambar yang telah dinormalisasi:

Tabel 2 Normalisasi Ukuran Objek

No	X	Y	Width	Height	X_center	Y_center	Width_norm	Height_norm
1	126.92	118.85	113.85	216.15	0.49609375	0.4640625	0.44453125	0.84453125
2	128.08	118.85	63.08	164.23	0.5	0.50390625	0.24609375	0.64140625
3	345	208.5	522	403	0.5390625	0.434375	0.815625	0.83984375
4	319	222.5	506	363	0.4984375	0.46328125	0.790625	0.75625
5	99.63	125	132.59	184.81	0.3890625	0.48828125	0.51796875	0.721875
6	134.44	132.41	205.91	84.81	0.525	0.5171875	0.8015625	0.33125
7	130.56	138.89	215.93	132.59	0.51015625	0.5421875	0.84375	0.51796875
8	130.37	125.19	197.04	129.63	0.509375	0.4890625	0.76953125	0.50625
9	125.56	130.74	157.04	203.70	0.490625	0.5109375	0.61328125	0.7953125

Gambar berikut menunjukkan proses pelabelan Roboflow menggunakan bounding box:



Gambar 4 Visualisasi Anotasi Objek
 Sumber: Penulis

Setelah proses anotasi selesai, gambar diubah ukurannya untuk sesuai dengan input model YOLOv8, yang membutuhkan gambar berukuran 640 x 640. Untuk mengubah ukuran gambar awal yang belum diubah, digunakan metode penarikan (stretch to fit).

4.1.2.3 Augmentasi

Setelah proses anotasi selesai, gambar diperbesar. Proses memperbesar dataset menggunakan teknik seperti penarikan dan rotasi penggeseran untuk mencegah overfitting. Namun, penulis hanya menggunakan teknik peningkatan geometris, yaitu rotasi horizontal. Dataset berikut dihasilkan dari proses augmentasi:

Tabel 3 Dataset Hasil Augmentasi

No	Jenis Sampah	Jumlah Gambar	Set Pelatihan	Set Validasi	Set Pengujian
1	Metal	1600	1210	249	141
2	Plastik	1560	1206	198	156
3	Kaca	1399	1032	216	151
4	Kardus	1293	977	130	186
5	Styrofoam	897	655	157	85

4.1.3 Pelatihan dan Evaluasi

Dalam satu iterasi, dilakukan proses pelatihan untuk meningkatkan kemampuan prediksi model dan proses evaluasi untuk mengevaluasi kinerja model. Kedua proses ini dilakukan berdasarkan dataset masing-masing dan diulang sebanyak jumlah iterasi yang dimasukkan.

4.1.3.1 Konfigurasi Model

Sebelum memulai pelatihan, diperlukan persiapan. Persiapan ini berkaitan dengan model yang digunakan; backbone, neck, dan head adalah blok penting dalam algoritma YOLOv8. Backbone, yang juga disebut sebagai feature extractor, bertugas mengekstrak fitur-fitur penting dari data yang dimasukkan. Pada lapisan awal, backbone menangkap pola-pola sederhana seperti tepi dan tekstur, dan seiring kedalaman lapisan, ia dapat merepresentasikan data dalam skala yang berbeda, yang memungkinkannya menangkap fitur pada berbagai tingkat abstraksi. Hasil akhirnya adalah representasi data input hierarkis yang kaya.

4.1.3.2 Pelatihan Model

Iterasi pelatihan dilakukan dengan jumlah iterasi yang sesuai dengan input yang diberikan setelah menyelesaikan persiapan model, dataset, dan semua elemen yang diperlukan untuk proses pelatihan model. Selain melakukan pelatihan menggunakan optimizer SGD, parameter kinerja seperti IoU (Intersection over Union) dan ketepatan rata-rata dievaluasi. Peneliti juga membuat file.yaml yang digunakan dalam proses pelatihan untuk mengintegrasikan notebook Kaggle dengan dataset. File ini memudahkan konfigurasi pelatihan dengan menghubungkan script ke dataset. Setelah membuat file data.yaml, proses pelatihan dilakukan di Kaggle.com.

```

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
97/100 2.35G 0.5202 0.3176 0.9465 21 640: 100%|██████████
██████████ 228/228 [00:35<00:00, 6.50it/s]
Class Images Instances Box(P R mAP50 mAP50-95):
100%|██████████ 13/13 [00:02<00:00, 4.56it/s]
all 406 950 0.87 0.817 0.879 0.704

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
98/100 2.35G 0.5178 0.3101 0.9439 37 640: 100%|██████████
██████████ 228/228 [00:35<00:00, 6.50it/s]
Class Images Instances Box(P R mAP50 mAP50-95):
100%|██████████ 13/13 [00:02<00:00, 4.56it/s]
all 406 950 0.863 0.825 0.885 0.709

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
99/100 2.35G 0.5188 0.3095 0.9455 28 640: 100%|██████████
██████████ 228/228 [00:35<00:00, 6.36it/s]
Class Images Instances Box(P R mAP50 mAP50-95):
100%|██████████ 13/13 [00:02<00:00, 4.62it/s]
all 406 950 0.846 0.84 0.883 0.709

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
100/100 2.33G 0.5113 0.312 0.9375 16 640: 100%|██████████
██████████ 228/228 [00:35<00:00, 6.50it/s]
Class Images Instances Box(P R mAP50 mAP50-95):
100%|██████████ 13/13 [00:02<00:00, 4.47it/s]
all 406 950 0.845 0.837 0.881 0.709

100 epochs completed in 1.177 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 6.3MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 6.3MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.13 torch-2.1.2 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 168 layers, 3006623 parameters, 0 gradients, 8.1 GFLOPs
Class Images Instances Box(P R mAP50 mAP50-95):
100%|██████████ 13/13 [00:05<00:00, 2.32it/s]
all 406 950 0.833 0.837 0.894 0.714
Kaca 406 216 0.83 0.866 0.923 0.735
Kardus 406 130 0.831 0.794 0.863 0.625
Metal 406 249 0.9 0.9 0.945 0.781
Plastik 406 198 0.824 0.874 0.897 0.753
Styrofoam 406 157 0.782 0.752 0.842 0.677

Speed: 0.2ms preprocess, 2.1ms inference, 0.8ms loss, 1.5ms postprocess per image
Results saved to runs/detect/train2
    
```

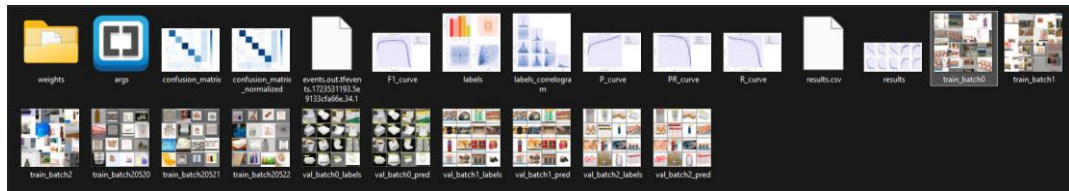
Gambar 5
 Proses training

Selama pelatihan berlangsung, kita dapat melihat bahwa ada total 100 epoch yang akan dilalui oleh model. Metrik seperti box_loss, obj_loss dan cls_loss—yang mengukur kesalahan dalam berbagai aspek deteksi—mulai menunjukkan nilai-nilai awal dan akan terus ditingkatkan seiring berjalannya waktu pelatihan. Selanjutnya, perhatian diberikan pada angka yang dihasilkan. Box_loss, yang mengevaluasi seberapa baik model memprediksi bounding box objek, menunjukkan penurunan yang konsisten, yang menunjukkan bahwa model lebih baik dalam melokalisasi objek dalam gambar. Sementara itu, obj_loss dan cls_loss, yang mengukur keakuratan dalam menemukan keberadaan objek dan mengklasifikasikannya ke dalam kategori yang tepat, juga menunjukkan tren penurunan yang menunjukkan bahwa model tersebar. Tidak ada

kebocoran memori atau penggunaan sumber daya yang tidak efisien, karena tingkat pemakaian GPU tetap stabil.

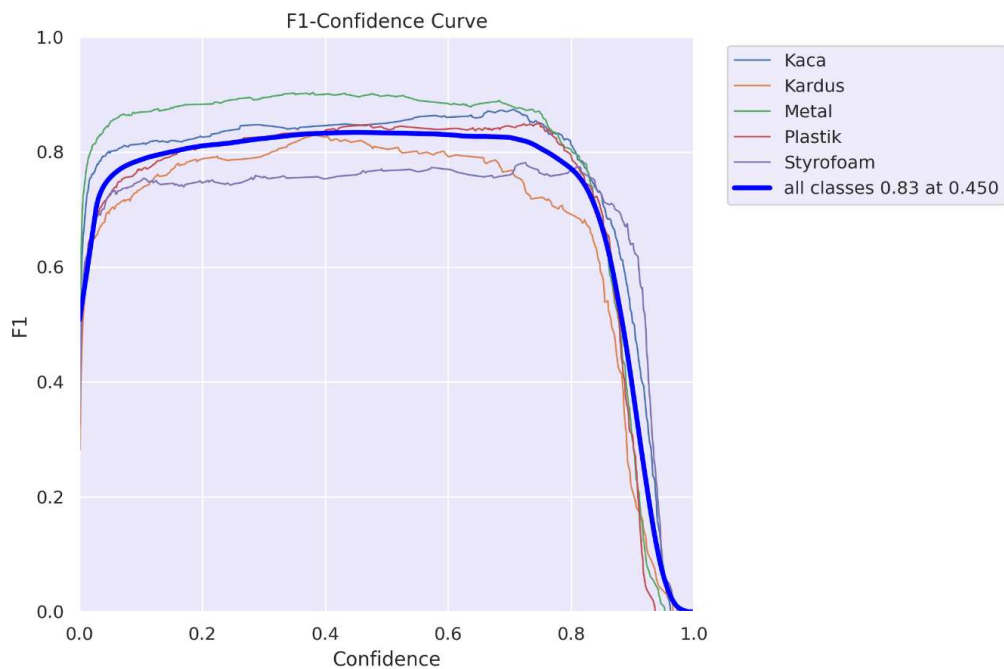
4.1.3.3 Hasil Training

Berbagai file penting yang terkait dengan penelitian ini dibuat selama proses pelatihan, yang disimpan di folder "runs/train" di direktori output Kaggle. Hasil pelatihan ini juga membentuk berbagai file metrik evaluasi dan data yang digunakan untuk mengevaluasi dan mengelola proses pelatihan selanjutnya.



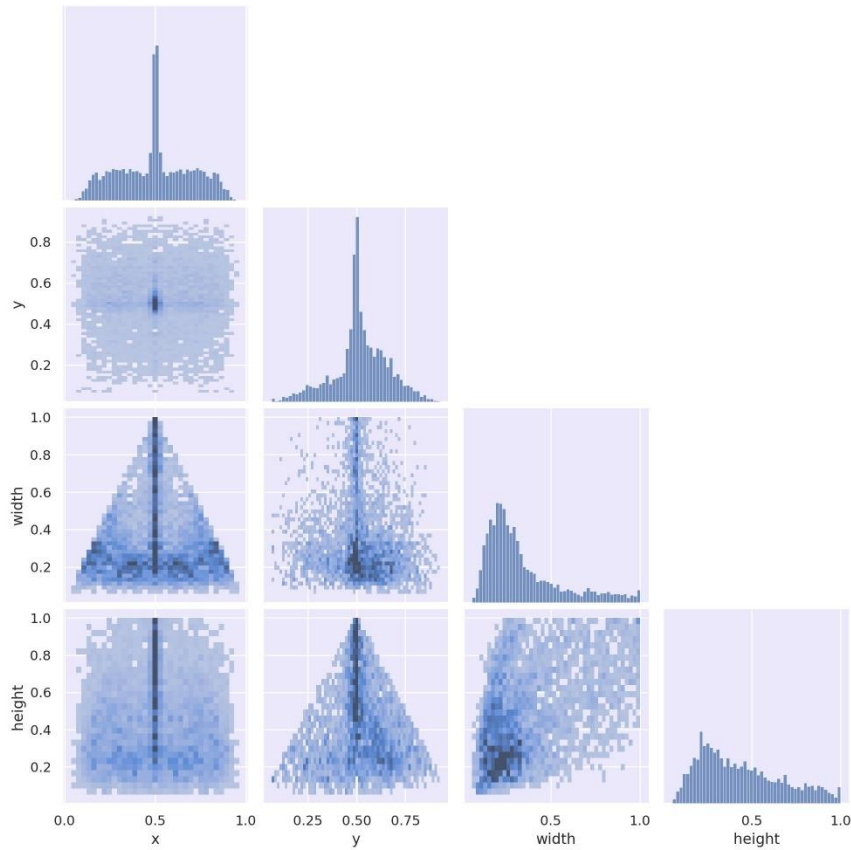
Gambar 6
Hasil training
Sumber: Penulis

Visualisasi kurva F1 dalam file "F1_curve.png" memberikan pemahaman penting tentang keseimbangan sensitivitas dan presisi selama proses pelatihan model. Ketika evaluasi model dilakukan, skor F1 adalah metrik statistic yang sangat berguna untuk mengukur akurasi model dalam kasus dataset yang tidak seimbang.



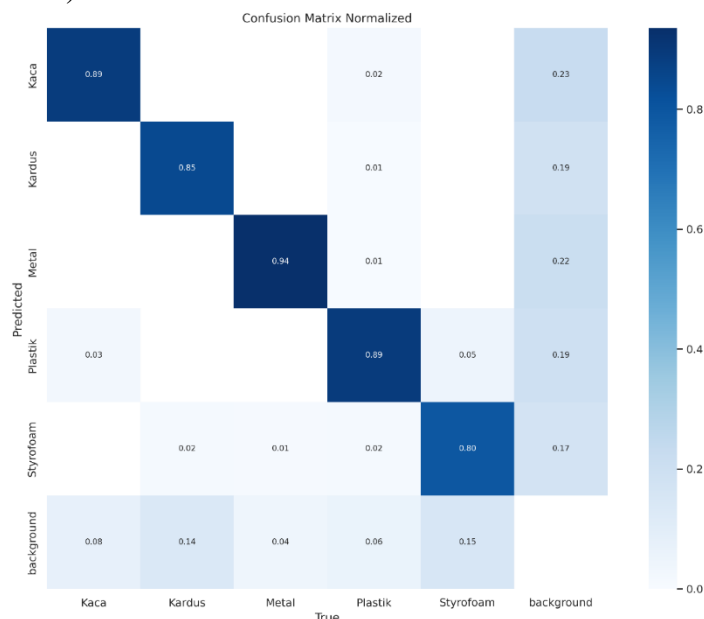
Gambar 7
Kurva F1

Correlogram label, yang menunjukkan korelasi antar kelas objek dalam dataset pelatihan, membantu memahami hubungan antara kelas label dan menemukan pola korelasi atau kesalahan label yang dapat mempengaruhi pelatihan model. File yang disebut "label_correlogram.jpg" juga menampilkan correlogram label.



Gambar 8
 Correlogram Label

Untuk menilai kinerja model terhadap kelas positif, kurva presisi-sensitivitas ("P_curve.png" dan "PR_curve.png" menunjukkan kompromi antara presisi (seberapa tepat model mengidentifikasi kelas positif) dan sensitivitas (seberapa komprehensif model mengidentifikasi kelas positif). Untuk membedakan antara kelas negatif dan positif, kualitas keseluruhan model dapat diukur dengan mengevaluasi luas area di bawah kurva (AUC).

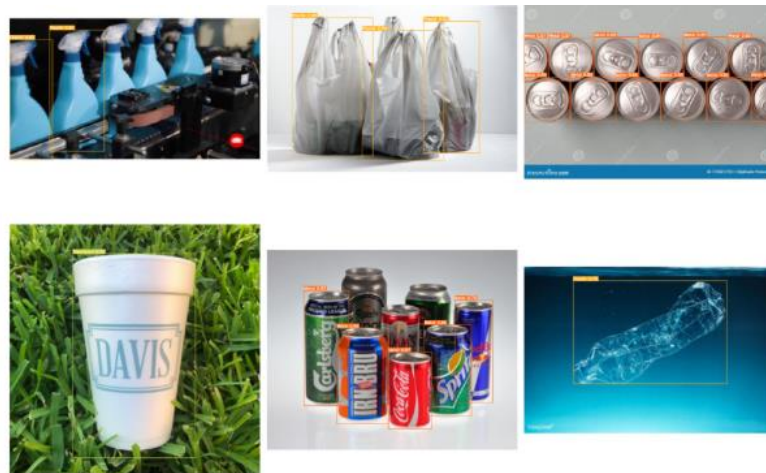


Gambar 9
 Confusion Matrix

File label seperti "batch0_labels" dan "batch0_pred" adalah visualisasi yang memungkinkan untuk membandingkan label sebenarnya (ground truth) dengan prediksi yang dibuat oleh model pada batch data pertama. Ini membantu menentukan apakah model benar-benar mendeteksi dan mengklasifikasikan objek gambar. Kesesuaian antara label dan prediksi ini memberikan gambaran awal tentang kualitas pelatihan yang telah dilakukan. File label seperti "batch0_labels" dan "batch0_pred" adalah visualisasi yang memungkinkan untuk membandingkan label sebenarnya (benar nyata) dengan prediksi yang dibuat oleh model pada batch data pertama. Ini membantu dalam menentukan apakah model mendeteksi dan mengklasifikasikan objek gambar dengan akurat.

4.1.3.4 Proses Pengujian

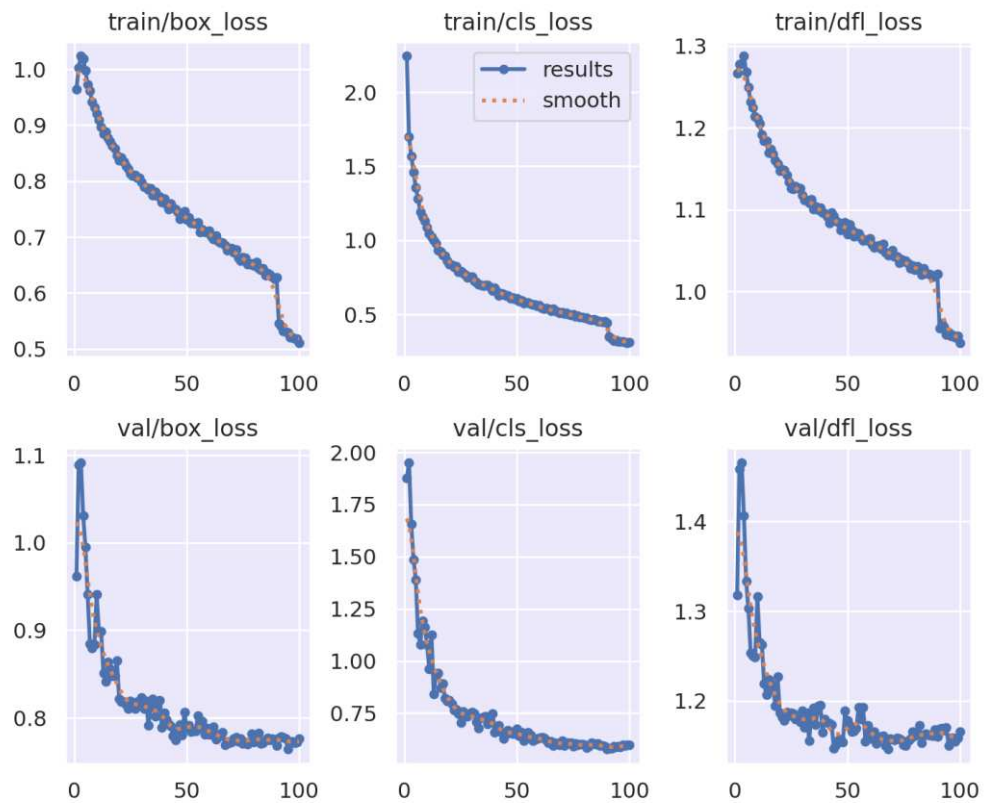
Pengujian model adalah langkah penting untuk mengevaluasi seberapa baik model deteksi objek belajar dari data pelatihan. Pada fase pengujian dengan gambar, setiap instance dalam gambar akan dianalisis untuk mengevaluasi kemampuan model untuk menemukan dan melokalisasi objek dengan benar. Untuk setiap gambar, parameter utama seperti precision, recall, dan Intesection over Union (IoU) akan dihitung. Precision menunjukkan persentase prediksi yang benar dari semua prediksi yang dibuat oleh model, sedangkan recall menunjukkan seberapa banyak objek sebenarnya yang berhasil dideteksi oleh model. Sebaliknya, IoU akan mengevaluasi bounding boxes yang dihasilkan oleh model dengan membandingkan ketidaksesuaian antara prediksi dan kebenaran nyata.



Gambar 10
Contoh Hasil Deteksi Gambar

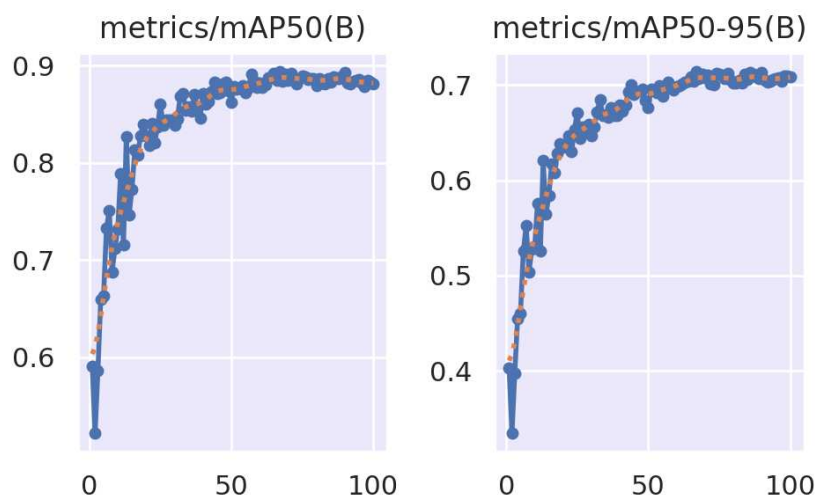
Pengujian model ini akan menunjukkan performa model saat ini dan iterasi perbaikan di masa depan. Hasil pengujian ini akan dibandingkan dengan pekerjaan serupa dalam literatur untuk menentukan peningkatan atau modifikasi yang diperlukan untuk model.

4.1.3.5 Evaluasi Model



Gambar 11
Evaluasi Bagian Nilai *loss*

Pada proses pelatihan dengan 100 iterasi, ketiga kehilangan yang terdiri dari kotak, kelas, dan distribution focal loss (dfl) memiliki nilai yang rendah, sehingga model menghasilkan prediksi yang baik. Namun, karena dataset pelatihan adalah dataset yang sudah dipelajari, tingkat *loss* tidak dapat dijadikan tolok ukur. Oleh karena itu, kita dapat melihat hasil prediksi model untuk dataset validasi. Grafik *loss* pada dataset validasi yang menggunakan 20% dari total dataset memberikan hasil yang cukup memuaskan, dan ketiga variabel *loss* dapat ditekan seiring iterasi berjalan.



Gambar 12
Evaluasi Bagian Nilai mAP

Metric mean average precision (mAP) yang dibuat oleh model YOLOv8 untuk kotak dan mask memiliki hasil yang cukup memuaskan. Perhitungan mAP didasarkan pada perhitungan IoU. Pada mAP50, hasil prediksi minimal adalah 50% bagian kotak tumpang tindih terhadap kotak target, dan pada mAP95, hasil prediksi minimal adalah 95% bagian kotak tumpang tindih terhadap kotak target, sehingga model dapat mencapai nilai yang cukup tinggi pada angka 0.88.

4.1.3.6 Pengujian Langsung

Pengujian secara langsung, juga dikenal sebagai inference, adalah menguji model terbaik yang dihasilkan dari proses pelatihan dengan data. Proses inference mirip dengan proses pelatihan, gambar dimasukkan ke dalam model, dan model memproses gambar tersebut lapisan demi lapisan. Tahapan deteksi (inference) berbeda dengan tahapan pelatihan, karena perlu menghitung *loss* untuk memperbaiki model dan tahapan validasi, selain itu perlu menghitung beberapa nilai evaluasi matriks. Pada tahap inference data yang dimasukkan tidak memiliki label atau target, sehingga perhitungan IoU untuk pengukuran tidak dapat dilakukan. Hasil deteksi model hanya diproses melalui proses NMS untuk menghilangkan beberapa prediksi yang tidak akurat dan kotak tumpang tindih.



Gambar 13
Prediksi Menggunakan Test Set

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Diperlukan Strategi terpadu untuk mengelola timbunan sampah. Ada beberapa cara untuk mengurangi sampah, seperti menghindari penggunaan plastik sekali pakai, mengelola sampah dengan cara yang baik, seperti kompos dan daur ulang, dan memilah jenis sampah, seperti sampah organik dan anorganik. Dengan menggunakan metode pembelajaran transfer yang didasarkan pada model arsitektur YOLOv8, penelitian ini berhasil mengembangkan Sistem Pendeteksian Jenis Sampah yang dapat mendeteksi sampah secara otomatis. Dengan menggunakan arsitektur YOLOv8, model yang dikembangkan oleh penulis menunjukkan kecepatan dan efisiensi tinggi dalam memprediksi gambar dan mengklasifikasi jenis sampah. Ini menunjukkan bahwa model ini dapat digunakan secara efektif dalam pengelolaan sampah berbasis teknologi. Karena sistem pendeteksian tersedia secara online, maka biaya pembuatan aplikasi ini cukup terjangkau.

5.2 Saran

Disarankan untuk penelitian lanjutan dengan membuat pelatihan model lebih akurat dan tidak mengalami bias, hal ini dapat dicapai dengan menambah *dataset* sampah ke dalam jumlah kelas dan *dataset* itu sendiri. Selain itu, untuk meningkatkan tingkat produktivitas masyarakat, dapat mengintegrasikan istem lain, seperti perhitungan berat sampah dan pelaporan pendapatan dari penjualan sampah.

DAFTAR PUSTAKA

- [1] SIPSN, "Sistem Informasi Pengelolaan Sampah Nasional." [Online]. Available: <https://sipsn.menlhk.go.id/sipsn/>
- [2] L. E. Torres-García, A., Rodea-Aragón, O., Longoria-Gandara, O., Sánchez-García, F., & González-Jiménez, "Intelligent waste separator. *Computacion y Sistemas*, 19(3).," pp. 487–500, 2015.
- [3] N. Rima Dias Ramadhani, Nur Aziz Thohari, A., Kartiko, C., Junaidi, A., Ginanjar Laksana, T., & Alim Setya Nugraha, "Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah," *RESTI (Rekayasa Sist. dan Teknol. Informasi)*, pp. 312–318, 2021.
- [4] R. Meyrena, S. D., & Amelia, "Analisis Pendayagunaan Limbah Plastik Menjadi Ecopaving Sebagai Upaya Pengurangan Sampah.," *Indones. J. Conserv.*, pp. 96–100, 2020.
- [5] B. Citra Malina, A., Muchtar, A., Penelitian dan Pengabdian pada Masyarakat, L., Hasanuddin, U., Jalan Perintis Kemerdekaan Km, M., Makassar, K., & Penelitian dan Pengembangan Daerah Kota Makassar Jalan Jenderal Ahmad Yani No, "KAJIAN LINGKUNGAN TEMPAT PEMILAHAN SAMPAH DI KOTA MAKASSAR," *J. Inov. dan Pelayanan Publik Makassar*, vol. 1, no. 1, 2017.
- [6] G. LeCun, Y., Bengio, Y., & Hinton, "Deep learning," *Nature*, pp. 436–444, 2015.
- [7] I. Jakhar, D., & Kaur, "Artificial intelligence, machine learning and deep learning: definitions and differences.," *Clin. Exp. Dermatology*, pp. 131–132, 2020.
- [8] S. Zhang, X., & Xu, "Research on Image Processing Technology of Computer Vision Algorithm.," in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2020, pp. 122–124.
- [9] M. Riziq sirfatullah Alfarizi, M., Zidan Al-farish, M., Taufiqurrahman, M., Ardiansah, G., & Elgar, "PENGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING," *Karimah Tauhid*, 2023.
- [10] M. Kuroki, "Using Python and Google Colab to teach undergraduate microeconomic theory.," *Int. Rev. Econ. Educ.*, 2021.