



## DAFTAR PUSTAKA

- Basak, J. K., Paudel, B., Kim, N. E., Deb, N. C., Kaushalya Madhavi, B. G., & Kim, H. T. (2022). Non-Destructive Estimation of Fruit Weight of Strawberry Using Machine Learning Models. *Agronomy*, 12(10). <https://doi.org/10.3390/agronomy12102487>
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Dalam *Journal of Machine and Computing* (Vol. 4, Nomor 1). <https://doi.org/10.53759/7669/jmc202404020>
- Bradski, G., & Kaehler, A. (2009). Learning OpenCV---Computer Vision with the OpenCV Library. Dalam *IEEE Robotics & Automation Magazine* (Vol. 16, Nomor 3). <https://doi.org/10.1109/mra.2009.933612>
- Breiman, L. (2001). Random Forest. 2001 *Kluwer Academic Publishers. Manufactured in The Netherlands.*, 12343 LNCS, 5–32. [https://doi.org/10.1007/978-3-030-62008-0\\_35](https://doi.org/10.1007/978-3-030-62008-0_35)
- C. Gonzalez, R., & E. Woods, R. (2018). Digital Image Processing. Dalam *A Guide for Machine Vision in Quality Control*. <https://doi.org/10.1201/9781003002826-2>
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>
- Corley, R. H. V. (2018). Studies of bunch analysis 2 - Bunch sampling to estimate oil yield. *Journal of Oil Palm Research*, 30(2), 206–218. <https://doi.org/10.21894/jopr.2018.0019>
- DJI. (2018). *Mavic Air - Informasi Produk*. <https://www.dji.com/id/support/product/mavic-air>



- DJI. (2020). *DJI Hardware Introduction*. [https://developer.dji.com/mobile-sdk/documentation/introduction/product\\_introduction.html](https://developer.dji.com/mobile-sdk/documentation/introduction/product_introduction.html)
- Efron, B., & Tibshirani, R. J. (1993). An Introduction to the Bootstrap. Dalam *CHAPMAN & HALL/CRC* (Vol. 57, Nomor 1989).
- Fackrurrozi, ., Junaedi, A., & Derajat Matra, D. (2019). Manajemen Pemanenan Kelapa Sawit (*Elaeis guineensis* Jacq.) di Kebun Rambutan, Serdang Bedagai, Sumatera Utara. *Buletin Agrohorti*, 7(3), 319–328. <https://doi.org/10.29244/agrob.v7i3.30259>
- Fitzgibbon, A. W., Pilu, M., & Fisher, R. B. (1996). Direct least squares fitting of ellipses. *Proceedings - International Conference on Pattern Recognition*, 1(September 1996), 253–257. <https://doi.org/10.1109/ICPR.1996.546029>
- Galagher, J. (2023). *Generate Image Augmentations with Roboflow*. <https://blog.roboflow.com/image-augmentation/>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Dalam *Journal of Physics A: Mathematical and Theoretical*.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Dalam *Springer Series in Statistics* (Vol. 26, Nomor 2).
- Henson, I. E. (2012). Ripening, Harvesting, and Transport of Oil Palm Bunches. *Palm Oil: Production, Processing, Characterization, and Uses*, 137–162. <https://doi.org/10.1016/B978-0-9818936-9-3.50008-3>
- Joshua Goldman. (2018). *DJI Mavic Air review: A folding 4K mini drone that's close to perfect*. CNET. <https://www.cnet.com/reviews/dji-mavic-air-review/>
- Kalantar, A., Edan, Y., Gur, A., & Klapp, I. (2020). A deep learning system for single and overall weight estimation of melons using unmanned aerial vehicle images. *Computers and Electronics in Agriculture*, 178(August), 105748. <https://doi.org/10.1016/j.compag.2020.105748>

- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147(February), 70–90. <https://doi.org/10.1016/j.compag.2018.02.016>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Louppe, G. (2014). *Understanding Random Forests: From Theory to Practice*. July.
- Oushabi, A., Sair, S., Abboud, Y., Tanane, O., & El Bouari, A. (2015). Natural thermal-insulation materials composed of renewable resources: Characterization of local date palm fibers (LDPF). *Journal of Materials and Environmental Science*, 6(12), 3395–3402.
- Perros, N., Kalivas, D., & Giovos, R. (2021). Spatial analysis of agronomic data and uav imagery for rice yield estimation. *Agriculture (Switzerland)*, 11(9). <https://doi.org/10.3390/agriculture11090809>
- Pilario, K. E. S., Cao, Y., & Shafiee, M. (2021). A Kernel Design Approach to Improve Kernel Subspace Identification. *IEEE Transactions on Industrial Electronics*, 68(7), 6171–6180. <https://doi.org/10.1109/TIE.2020.2996142>
- Probolinggo, D. L. H. K. (2024). *Kelapa Sawit*. <https://dlh.probolinggokab.go.id/kelapa-sawit/>
- Rath, S. (2023). *Train YOLOv8 Instance Segmentation on Custom Data*. LearnOpenCV by BigVision. <https://learnopencv.com/train-yolov8-instance-segmentation/>
- Ratnaningsih, Deden, H., & Azizah. (2022). Kajian Pemetaan Komoditas Kelapa Sawit (*Elaeis guineensis jacq.*) Pada Skripsi, Tesis dan Disertasi IPB sampai tahun 2022. *Jurnal Pustakawan Indonesia*, 21(2), 124–239. <https://doi.org/10.29244/jpi.21.2.124-239>



- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Rist, L., Feintrenie, L., & Levang, P. (2010). The livelihood impacts of oil palm: Smallholders in Indonesia. *Biodiversity and Conservation, 19*(4), 1009–1024. <https://doi.org/10.1007/s10531-010-9815-z>
- Saha, S. (2018). *A Guide to Convolutional Neural Networks - the ELI5 way*. SaturnCloud. <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data, 6*(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Sidekick, C. (2024). *Equations of Ellipses*. <https://www.coursesidekick.com/mathematics/study-guides/collegealgebracoreq/equations-of-ellipses>
- Silitonga, Y. R., Heryanto, R., Taufik, N., Indrayana, K., Nas, M., & Kusriani, N. (2020). *Budidaya Kelapa Sawit & Varietas Kelapa Sawit*.
- Suparwito, H. (2019). Factors Influencing the Difficulty Level of the Subject: Machine Learning Technique Approaches. *International Journal of Applied Sciences and Smart Technologies, 1*(1), 65–82. <https://doi.org/10.24071/ijasst.v1i1.1869>
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Dalam *Springer*. <https://doi.org/10.1081/E-EOE2-120009509>
- Ultralytics. (2023a). *Instance Segmentation*. <https://docs.ultralytics.com/tasks/segment/>
- Ultralytics. (2023). *Ultralytics YOLO Docs*. <https://docs.ultralytics.com/>



Varoslavskiy, L. P. (1977). Digital Image Processing. Dalam *Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika)* (Vol. 31–32, Nomor 11).

Villarino, M. B. (2005). *Ramanujan's Perimeter of an Ellipse*. 3, 1–12.













Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). *Deep Learning for Computer Vision : A Brief Review*. 2018.

WWF. (2022). *What is Palm Oil? Facts About the Palm Oil Industry*.  
<https://www.worldwildlife.org/industries/palm-oil>















Lampiran 1 Dataset primer











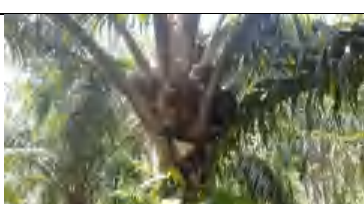

Tandan Kelapa Sawit Tidak Tertutup Pelepah

Sampel	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
1		10	7		19
2		10	8		11
3		21	9		22,5
4		26,5	10		18
5		14	11		19,5
6		41	12		10














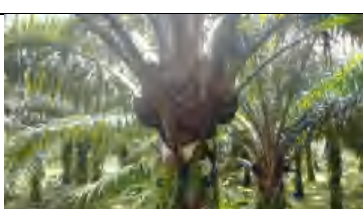
№	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
13		16	20		26
14		36,5	21		29
15		14	22		27,5
16		35,5	23		38
17		23	24		23,5
18		31	25		31















no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
19		18	26		32
27		26,5	34		35
28		27,5	35		22
29		15,5	36		19,5
30		16	37		24
31		24	38		24










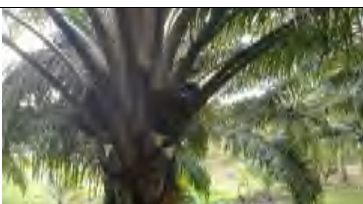

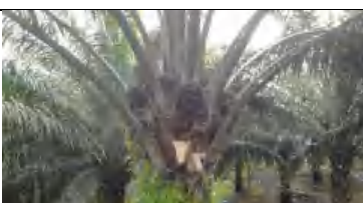




№	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
32		16	39		14,5
33		15,5	40		21
41		18	48		9
42		17,5	49		22
43		13	50		16,5
44		11,5	51		17















Sl	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
45		33,5	52		16,5
46		12	53		21,5
47		19	54		12
55		19	62		33
56		17,5	63		13
57		15	64		11








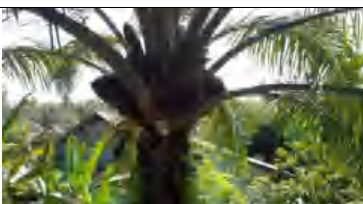






№	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
58		18	65		21
59		18	66		19
60		15,5	67		19,5
61		29	68		24,5
69		23	76		19
70		18	77		32







no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
71		23,5	78		32
72		30	79		20
73		32,5	80		37
74		25	81		48
75		39	82		42
83		37,5	90		24,5











no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
84		30	91		26,5
85		50,5	92		30
86		39	93		23
87		25	94		26
88		29	95		31,5
89		29	96		23,5















	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
97		22	99		32
98		36	100		26,5










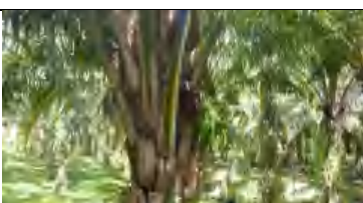

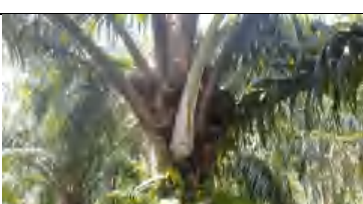
#### Tandan Kelapa Sawit Tertutup Pelepah

Sampel	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
1		10	5		14
2		10	6		10
3		21	7		41
4		26,5	8		19










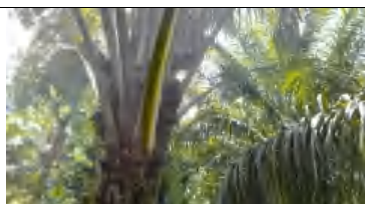



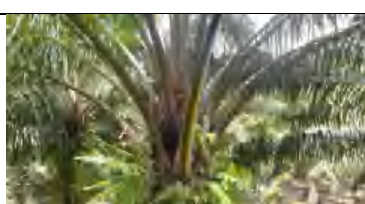
Sl	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
9		11	15		36,5
10		22,5	16		14
11		18	17		35,5
12		19,5	18		23
13		10	19		31
14		16	20		18










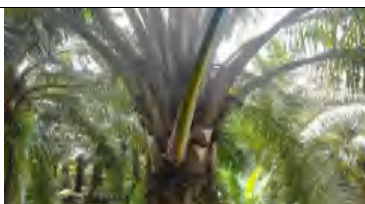




no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
21		26	27		32
22		29	28		26,5
23		27,5	29		27,5
24		38	30		15,5
25		23,5	31		16
26		31	32		24







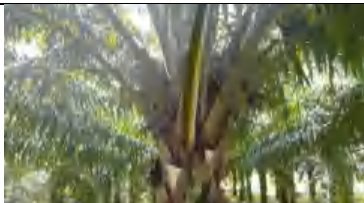









no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
33		16	39		24
34		15,5	40		14,5
35		35	41		21
36		22	42		18
37		19,5	43		17,5
38		24	44		13















№	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
45		11,5	51		16,5
46		33,5	52		17
47		12	53		16,5
48		19	54		21,5
49		9	55		12
50		22	56		19








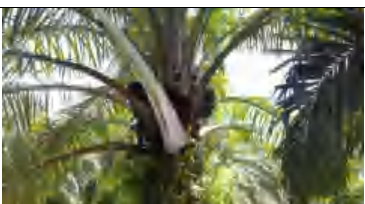






no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
57		17,5	63		33
58		15	64		13
59		18	65		11
60		18	66		21
61		15,5	67		19
62		29	68		19,5











no	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
69		24,5	75		39
70		23	76		19
71		23,5	77		32
72		30	78		32
73		32,5	79		20
74		25	80		37



№	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
81		48	87		25
82		42	88		29
83		37,5	89		29
84		30	90		24,5
85		50,5	91		26,5
86		39	92		30



No	Gambar	Massa (kg)	Sampel	Gambar	Massa (kg)
93		23	97		22
94		26	98		36
95		23,5	99		32
96		31,5	100		26,5

Dataset citra setelah *labeling* dapat dilihat pada link berikut:

<https://s.unhas.ac.id/AIMPDatasetSawit>





Lampiran 2 Hasil ekstraksi fitur dari *YOLOv8-Segmentation*

Dataset ekstraksi fitur dapat dilihat pada link berikut:

<https://s.unhas.ac.id/DatasetFiturTBSSawit>



1. TBS Kelapa Sawit Tertutup Pelepah

a. YOLOv8Nano-Segmentation

**Nano-Training**

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri -sitas	Massa
1	192.8775	171.1181	25921.95	572.274	0.461413	9
2	202.8857	185.7961	29605.88	610.8351	0.401709	10
3	217.5558	87.0263	14870	500.6616	0.916507	11
4	263.7142	241.0947	49935.7	793.3501	0.405202	42
5	245.3932	198.2835	38215.45	698.8915	0.58915	48
...	...	...	...	...	...	...
13235	280.3008	225.1272	49561.22	796.2912	0.59576	50.5

**Nano-Testing**

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri -sitas	Massa
1	193.7717	121.5642	18500.6	501.8429	0.778731	23
2	218.7333	148.152	25451.44	581.6468	0.735691	23.5
3	275.5677	235.6521	51002.27	804.2465	0.518378	24
4	194.2891	40.19053	6132.846	409.2252	0.978371	31.5
5	306.4747	177.2818	42672.56	773.493	0.815714	36
...	...	...	...	...	...	...
1617	334.234	262.432	68890.17	940.637	0.619275	39

b. YOLOv8Small-Segmentation

**Small-Training**

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri -sitas	Massa
1	191.7956	162.149	24425.46	556.9505	0.534092	9
2	205.571	159.9667	25827.44	576.4217	0.628068	10
3	212.9101	84.36022	14106.65	489.0424	0.918154	11
4	264.561	245.1447	50937.58	800.9344	0.376026	42
5	243.0324	199.7995	38137.16	697.2571	0.569328	48
...	...	...	...	...	...	...
13235	281.7863	223.6349	49493.63	796.5434	0.608397	50.5



### Small-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	201.5209	122.58	19401.24	516.6753	0.793727	23
2	221.323	201.0286	34944.13	663.8113	0.41831	23,5
3	278.8444	229.6513	50294.53	800.6132	0.567197	24
4	245.2707	223.1088	42978.6	736.1407	0.41539	31,5
5	311.0888	166.0604	40573.31	766.916	0.845608	36
...	...	...	...	...	...	...
1515	304.5993	239.445	57282.84	857.6497	0.618101	39

### c. YOLOv8Medium-Segmentation

#### Medium-Training

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	192.4704	168.253	25434.11	567.2616	0.485609	9
2	202.3829	158.6579	25218.86	569.203	0.620825	10
3	217.2962	83.17146	14194.38	495.7853	0.923849	11
4	265.438	244.6597	51005.32	801.5921	0.387854	42
5	243.8836	196.3987	37619.33	693.6064	0.592871	48
...	...	...	...	...	...	...
13235	282.0731	222.4226	49275.44	795.2321	0.614998	50.5

#### Medium-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	195.7298	123.7554	19024.4	508.234	0.774742	23
2	231.503	165.2994	30055.07	627.6409	0.700118	23.5
3	286.4147	235.9898	53085.81	822.5036	0.566671	24
4	237.7271	219.0614	40901.05	717.8213	0.388419	31.5
5	311.2557	149.3477	36509.49	746.039	0.877365	36
...	...	...	...	...	...	...
1326	299.7211	240.7283	56667.56	851.4667	0.595744	39

### d. YOLOv8Large-Segmentation

#### Large-Training

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
--------	-------------	-------------	------	----------	----------------	-------





1	192.0311	162.3904	24491.85	557.6978	0.533743	9
2	202.7598	161.4137	25704.68	573.8873	0.605187	10
3	214.9862	85.19881	14385.8	493.8293	0.918122	11
4	262.6601	241.905	49903.25	792.9043	0.389607	42
5	243.1189	200.7412	38330.55	698.8035	0.564122	48
...	...	...	...	...	...	...
13235	280.1534	221.1359	48656.97	790.1543	0.613958	50.5

#### Large-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	195.7434	133.6249	20543.03	521.9815	0.730742	23
2	248.6175	173.3294	33844.95	668.0786	0.716903	23.5
3	303.6955	231.7632	55280.58	844.8955	0.646229	24
4	240.8584	222.0907	42012.84	727.4975	0.386999	31.5
5	298.7409	157.3463	36918.26	733.7386	0.850052	36
...	...	...	...	...	...	...
1296	311.0169	156.3423	38190.06	754.368	0.864472	39

#### e. YOLOv8XtraLarge-Segmentation

##### XtraLarge-Training

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	189.6025	165.2682	24610.65	558.0849	0.490118	9
2	205.5957	161.361	26055.68	578.51	0.619691	10
3	215.6797	82.56676	13986.35	492.1129	0.923822	11
4	263.5559	242.5979	50216.88	795.4054	0.39079	42
5	243.6769	198.4956	37988.76	696.3771	0.580044	48
...	...	...	...	...	...	...
13235	280.8928	227.3647	50159.53	800.5843	0.587208	50.5

##### XtraLarge-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	198.4513	125.0312	19487.78	514.6903	0.776567	23
2	240.9984	158.0852	29922.31	633.6619	0.754796	23.5
3	328.4212	269.929	69625.87	942.1331	0.569634	24
4	244.1005	224.6505	43069.11	736.6293	0.391167	31.5





16937	323.3437	233.382	59268.18	880.2206	0.692126	50.5
-------	----------	---------	----------	----------	----------	------

#### Small-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri -sitas	Massa
1	233.5906	215.7739	39586.23	706.1375	0.383051	23
2	248.7524	227.1179	44371.95	747.8816	0.407897	23.5
3	293.0603	236.412	54414.71	834.0749	0.59096	24
4	272.8669	206.6222	44281.03	756.778	0.653152	31.5
5	316.8376	251.1374	62493.96	895.1601	0.609693	36
...	...	...	...	...	...	...
3210	315.1707	245.8401	60853.89	884.6015	0.625752	39

#### c. YOLOv8Medium-Segmentation

#### Medium-Training

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri -sitas	Massa
1	198.7686	192.1149	29991.52	614.0427	0.256571	9
2	207.2467	189.239	30802.66	623.1195	0.407714	10
3	224.8615	178.4653	31518.01	635.6419	0.608352	11
4	302.4423	246.0795	58453.14	863.8918	0.581367	42
5	277.9487	198.7896	43395.84	754.0292	0.698917	48
...	...	...	...	...	...	...
16674	322.27	229.23	58020.46	872.4691	0.702891	50.5

#### Medium-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri -sitas	Massa
1	233.6248	218.909	40167.27	711.0263	0.3493	23
2	245.9794	225.5037	43565.45	740.9531	0.399442	23.5
3	292.0413	236.7271	54297.79	832.8614	0.585608	24
4	262.1762	215.3952	44352.61	751.968	0.570112	31.5
5	318.4977	247.266	61852.98	892.225	0.630301	36
...	...	...	...	...	...	...
3210	319.2855	243.0378	60945.67	887.36	0.648526	39



## d. YOLOv8Large-Segmentation

## Large-Training

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	197.9329	191.0107	29693.79	610.9996	0.262148	9
2	207.9135	191.8262	31324.23	628.1639	0.385698	10
3	217.8842	179.6005	30734.28	625.8163	0.566163	11
4	311.2319	244.9917	59886.02	876.8146	0.616739	42
5	283.5972	199.6351	44466.06	764.7992	0.710261	48
...	...	...	...	...	...	...
16937	322.5447	235.5416	59668.82	881.9744	0.683169	50.5

## Large-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	227.6802	219.2207	39200.95	702.0531	0.270056	23
2	247.6892	219.7522	42749.43	734.9112	0.461367	23.5
3	290.8285	236.5859	54040.05	830.6528	0.58158	24
4	265.8011	211.9769	44252.24	752.8751	0.603316	31.5
5	317.982	251.0156	62689.27	896.8771	0.613877	36
...	...	...	...	...	...	...
3210	323.2835	244.8164	62160.43	896.6305	0.653089	39

## e. YOLOv8XtraLarge-Segmentation

## XtraLarge-Training

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	198.1201	189.8234	29537.13	609.45	0.286357	9
2	205.4524	190.1393	30681.25	621.6268	0.378829	10
3	220.106	177.5632	30695.5	626.4459	0.590938	11
4	309.6462	250.42	60901.04	882.2112	0.588181	42
5	282.6401	198.5055	44065.24	761.5702	0.711855	48
...	...	...	...	...	...	...
16937	321.2222	229.756	57964.52	871.4476	0.698863	50.5

## XtraLarge-Testing

Sampel	Sumbu mayor	Sumbu minor	Luas	Keliling	Eksentri-sitas	Massa
1	225.9736	212.4614	37707.49	688.8556	0.340611	23



2	246.4116	232.8128	45056.56	752.9156	0.327611	23.5
3	291.433	233.4709	53439.38	827.0325	0.598512	24
4	264.2661	213.4765	44307.91	752.5581	0.589444	31.5
5	320.7882	250.8287	63195.39	901.2592	0.623387	36
...	...	...	...	...	...	...
3210	313.8651	242.7934	59850.79	877.9643	0.633723	39



### Lampiran 3 *Source code training dan testing* model segmentasi

```

from IPython.display import Image

!git clone https://github.com/MuhammadMoin97/ultralalytics.git
!pip install -e .'[dev]'

"""##Dataset 70:15:15"""
import os
os.makedirs("segment_701515", exist_ok=True)

!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="PnxYRGVMF1knrx7kOsJs")
project = rf.workspace("fruittesting").project("tbs-segmentation-
pvlvt")
version = project.version(6)
dataset = version.download("yolov8")
HOME = "/content/ultralalytics/ultralalytics/yolo/v8/segment"
dataset.location

"""##Training"""
!python train.py model=yolov8-seg.pt
data={dataset.location}/data.yaml epochs=50
!ls /content/ultralalytics/runs/segment/train
!python val.py
model='/content/ultralalytics/runs/segment/train/weights/best.pt'
data={dataset.location}/data.yaml
Image("/content/ultralalytics/runs/segment/train/val_batch0_pred.jpg
")
Image("/content/ultralalytics/runs/segment/train/confusion_matrix.pn
g")
Image("/content/ultralalytics/runs/segment/train/results.png")

"""##Testing"""
!python predict.py
model='/content/ultralalytics/runs/segment/train/weights/best.pt'
source="/content/ultralalytics/ultralalytics/yolo/v8/segment/segment_7
02010/TBS-Segmentation-2/test/images"

```



## Lampiran 4 Source code training dan testing model estimasi massa

### Training Random Forest Regression:

```

from sklearn.metrics import mean_squared_error,
mean_absolute_percentage_error, mean_absolute_error
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor

data_training = pd.read_csv("Data_training.csv", header=0)
data_testing = pd.read_csv("Data_testing.csv", header=0)

# Menghapus kolom yang tidak diperlukan (a dan b)
data_training = data_training[['Luas_area', 'Minor_axis',
'Keliling', 'Massa']]
data_testing = data_testing[['Luas_area', 'Minor_axis',
'Keliling', 'Massa']]

data_testing.head()
# Asumsi variasi jarak dalam rentang ±0.2 meter
estimated_distance = 2.0
variation_range = 0.2
lower_bound = estimated_distance - variation_range
upper_bound = estimated_distance + variation_range

# Augmentasi data dengan faktor skala dalam rentang variasi
augmented_features = data_training.copy()
np.random.seed(42) # Untuk hasil yang bisa direproduksi
for i in range(len(data_training)):
    random_scale_factor = np.random.uniform(lower_bound /
estimated_distance, upper_bound / estimated_distance)
    augmented_features.iloc[i, :4] *= random_scale_factor # Tidak
menerapkan pada Elips dan Massa

augmented_features['Massa'] = data_training['Massa'] # Menyalin
kolom Massa asli
augmented_features.head()

# Memisahkan fitur dan target
X_train = augmented_features.drop(columns=['Massa'])
y_train = augmented_features['Massa']
X_test = data_testing.drop(columns=['Massa'])
y_test = data_testing['Massa']

# Melatih model Random Forest Regression
model = RandomForestRegressor(bootstrap=True, max_depth=5,
max_features='sqrt', n_estimators=20, random_state=42)
model.fit(X_train, y_train)

# Melakukan prediksi
y_pred = model.predict(X_test)

# Evaluasi model
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mape = mean_absolute_percentage_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse, mape, mae

```



### Testing Random Forest Regression:

```

import cv2
import joblib
from ultralytics import YOLO
import pandas as pd
import math
import numpy as np

# Load the YOLOv8 model and the regression model
model = YOLO("YOLOv8-Large Augmentasi.pt")
xgb =
"C:\\Users\\ijlal\\Downloads\\newDataset\\XGBoostRegression.joblib"
regresi_model = "regresi_large_normalisasi.joblib"
regression_model = joblib.load(regresi_model)
# regression_model = joblib.load(xgb)

# Set up the video capture
rtmp = "rtmp://192.168.1.2:1935"
videoPath = "D:\\cache\\Dataset Citra Kelapa Sawit\\ALLOFTHEM_PART2
(MEI 2024)\\SPLIT\\testing\\Pohon 86_36kg.MP4"
# videoPath2 = "D:\\cache\\DJI_0707.MP4"
cap = cv2.VideoCapture('video/Pohon 83_Tandan 159_30kg.MP4')
cap = cv2.VideoCapture(videoPath)

# Initialize variables
W = 23 # cm, actual size
f = 1600 # focal length
frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
frame_rate = int(cap.get(cv2.CAP_PROP_FPS))

# Initialize VideoWriter
output_filename = 'trial2.mp4'
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_filename, fourcc, frame_rate,
(frame_width, frame_height))

# Initialize buffers
weight_buffer = []
frame_count = 0
predicted_weight_display = None

# Process video frames
while cap.isOpened():
    success, frame = cap.read()
    if success:
        frame_count += 1
        results = model(frame)
        largest_area = 0
        largest_detection = None
        for result in results:
            boxes = result.boxes
            for mask, box in zip(result.masks.xy, result.boxes):
                x1, y1, x2, y2 = map(int, box.xyxy[0])
                area = (x2 - x1) * (y2 - y1)
                if area > largest_area:
                    largest_area = area
                    largest_detection = box

```





```

        points = np.int32(mask)

    if largest_detection is not None:
        x1, y1, x2, y2 = map(int, largest_detection.xyxy[0])
        w = x2 - x1
        h = y2 - y1
        Major_axis = max(w, h) / 2
        Minor_axis = min(w, h) / 2
        luas = math.pi * Major_axis * Minor_axis
        keliling = 2 * math.pi * math.sqrt((Major_axis**2 +
Minor_axis**2) / 2)
        d = (W * f) / w

        overlay = frame.copy()
        cv2.fillPoly(overlay, [points], (0, 255, 0))

        # hitung luas segmentasi (px)
        mask_image = np.all(overlay == (0, 255, 0), axis=-1)
        luas_area = np.sum(mask_image)

        lower_color = np.array([0, 180, 0])
        upper_color = np.array([100, 255, 100])
        mask = cv2.inRange(overlay, lower_color, upper_color)
        # Temukan kontur pada mask
        contours, _ = cv2.findContours(mask,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

        # Hitung keliling kontur terbesar
        if contours:
            largest_contour = max(contours,
key=cv2.contourArea)
            perimeter = cv2.arcLength(largest_contour, True)
            print(f"Keliling area: {perimeter} pixels")

            if len(largest_contour) >= 5: # fitEllipse
setidaknya butuh 5 points
                ellipse = cv2.fitEllipse(largest_contour)
                (center, axes, orientation) = ellipse
                Major_axis = max(axes)
                Minor_axis = min(axes)
                a = Major_axis / 2
                b = Minor_axis / 2
                luas = math.pi * a * b
                keliling = math.pi * (3 * (a + b) -
math.sqrt((3 * a + b) * (a + 3 * b)))

                # Hitung eksentrisitas
                eccentricity = np.sqrt(1 - (b ** 2 / a ** 2))
                print(f"Eccentricity: {eccentricity}")

            input_features = pd.DataFrame([[luas_area, Minor_axis,
keliling]], columns=['Luas_area', 'Minor_axis', 'Keliling'])
            predicted_weight =
regression_model.predict(input_features)[0]
            weight_buffer.append(predicted_weight)

            if frame_count % 20 == 0: # Update every 20 frames or
at the first frame
                avg_weight = sum(weight_buffer) /

```



```

len(weight_buffer)
    predicted_weight_display = avg_weight
    weight_buffer = [] # Reset buffer after averaging

    if predicted_weight_display is not None:
        label2 = f"{predicted_weight_display:.2f} kg"
        overlay = frame.copy()
        cv2.fillPoly(overlay, [points], (36, 255, 12))
        alpha = 0.3
        frame = cv2.addWeighted(overlay, alpha, frame, 1 -
alpha, 0)
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255,
0), 4)
        cv2.putText(frame, label2, (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255, 12), 2)

    out.write(frame)
    frame_display = cv2.resize(frame, (1280, 720))
    cv2.imshow("Frame", frame_display)
    if predicted_weight_display is not None:
        print(x1, x2, y1, y2)
        print(f'Luas Area: {luas_area}')
        print(f'Luas: {luas:.2f}, Keliling: {keliling:.2f}')
        print(f'w: {w}, h: {h}, major: {Major_axis}, minor:
{Minor_axis}')
        print(f"prediksi massa:
{predicted_weight_display:.2f}")
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

# Release resources
cap.release()
out.release()
cv2.destroyAllWindows()

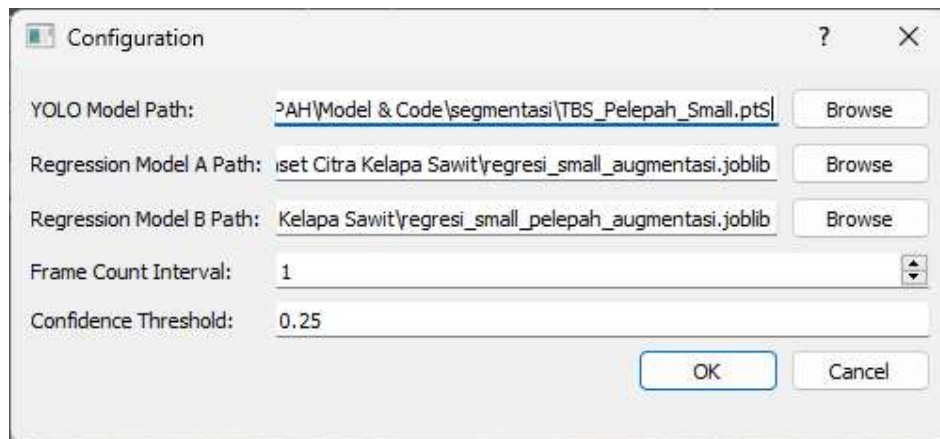
```



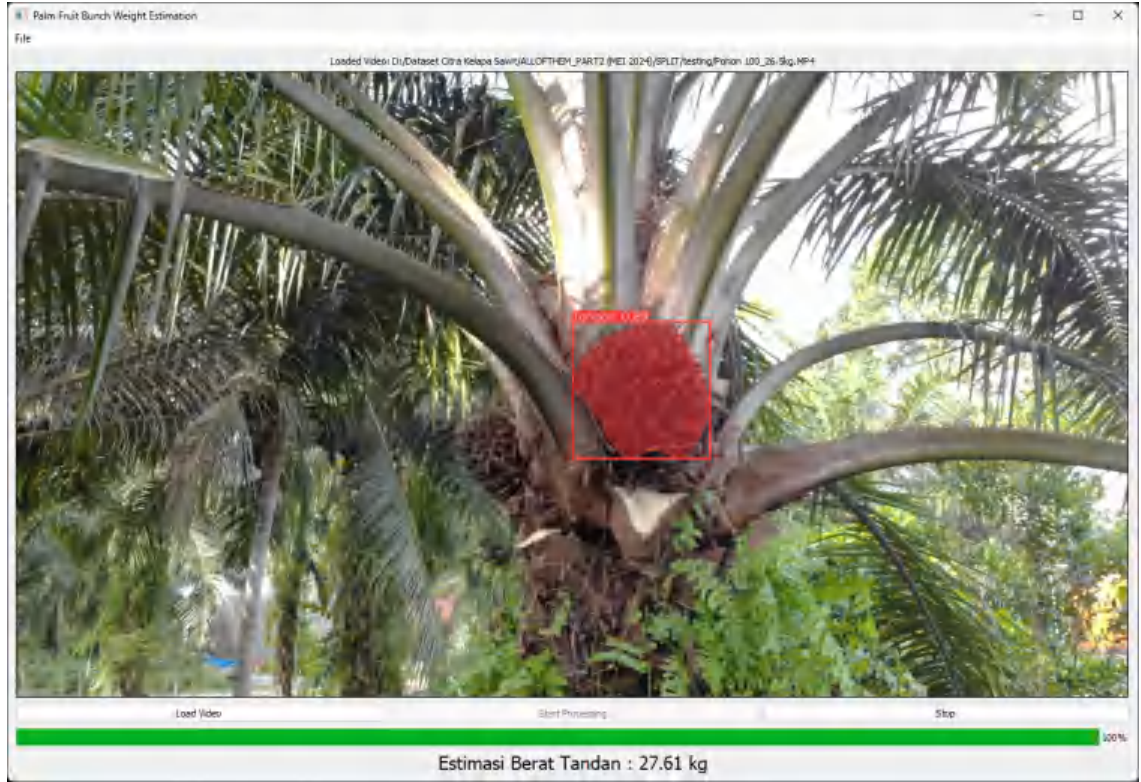
Lampiran 5 Tampilan program sistem estimasi massa TBS kelapa sawit



*Screenshot halaman awal program*



*Screenshot konfigurasi model segmentasi dan regresi*



Screenshot hasil estimasi massa TBS kelapa sawit



Lampiran 6 Dokumentasi akuisisi dataset





## LEMBAR PERBAIKAN SKRIPSI

### “SISTEM ESTIMASI MASSA TANDAN BUAH SEGAR KELAPA SAWIT MENGGUNAKAN CITRA *UNMANNED AERIAL VEHICLE* DENGAN MODEL *RANDOM FOREST REGRESSION*”

OLEH

**MUHAMMAD IJLAL NURHADI**  
**D121191068**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 31 Juli 2024.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Prof. Dr. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN.Eng	
Sekretaris	Elly Warni, S.T., M.T.	
Anggota	Ir. Christoforus Yohannes, M.T.	
	Prof. Dr. Ir. Syafruddin Syarif, MT	

Persetujuan perbaikan oleh pembimbing

Pembimbing	Nama	Tanda Tangan
I	Prof. Dr. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN.Eng	
II	Elly Warni, S.T., M.T.	