

DAFTAR PUSTAKA

- Abtohi, S. 2017. Skripsi. Implementasi Web Scrapping dan Klasifikasi Sentimen Menggunakan Metode Support Vector Machine. Universitas Islam Indonesia.
- Admin. 2020. Persiapan Bahan Baku Utama Produksi Saus Sambal. Internet. Diakses tgl 6 September 2021, <https://www.anekarasa.co.id/2020/10/17/persipan-bahan-baku-utama/>. PT. Bumbu Aneka Rasa.
- Awad, M., & Khanna, R. 2015. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers (p. 268). Springer nature.
- Badan Pusat Statistik. 2020. Statistik Hortikultura 2020. Katalog.
- Banyal, N. A., Surianti, S., & Dayat, A. R. 2016. Klasifikasi Citra Plasmodium Penyebab Penyakit Malaria dalam Sel Darah Merah Manusia dengan Menggunakan Metode Multi Class Support Vector Machine (SVM). ILKOM Jurnal Ilmiah, 8(2), 111-118.
- Basri, Indrabayu, and Andani Achmad. 2015. Gaussian Mixture Models Optimization for Counting The Number of Vehicle by Adjusting The Region of Interest under Heavy Traffic Condition. Universitas Hasanuddin-Makassar.
- Cambell, C., & Ying, Y. 2011. Learning with Support Vector Machines: Synthesis Lecturers on Artificial Intelllignce and Machine Learning. Morgan & Claypool.
- Deptan. 2009. Standar Prosedur Operasional (SPO) Pengolahan Cabe. Direktorat Pengolahan Hasil Pertanian. Jakarta.
- Faihah, R. T. 2010. Support Vector Machine (SVM). Makalah Data Mining. Universitas Trunojoyo.
- Falah, R. F., Nurhayati, O. D., & Martono, K. T. 2016. Aplikasi Pendeteksi Kualitas Daging Menggunakan Segmentasi Region of Interest Berbasis Mobile. Jurnal Teknologi dan Sistem Komputer, 4(2), 333-343.

- Rujukan Standar Specification for Fresh Chilies MS: 894 (1984).
<http://www.famaxchange.org/>. (Diakses tgl 30 Agustus 2021)
- Fetri. (2019). Disertasi. Pengaruh Konsentrasi dan Waktu Aplikasi Kitpsan Terhadap Pertumbuhan dan Hasil Cabai (*Capsicum annum L.*). Universitas Andalas-Padang.
- Gazali, W., Soeparno, H., & Ohliati, J. (2012). Penerapan Metode Konvolusi Dalam Pengolahan Citra Digital. *Jurnal Mat Stat*, 12(2), 103-113.
- Ginting, E. D. 2012. Skripsi. Deteksi Tepi Menggunakan Metode Canny dengan Matlab untuk Membedakan Uang asli dan Uang Palsu. Program Studi Teknik Informatika.
- Hamid and Haryanto. (2012). Untung besar dari bertanam cabai hibrida. *AgroMedia*.
- Hendrawan, Y., Rohmatulloh, B., Prakoso, I., Liana, V., Fauzy, M.R., Damayanti, R., Hermanto, M.B. and Al Riza, D.F., (2021). Classification of large green chilli maturity using deep learning. In *IOP Conference Series: Earth and Environmental Science* (Vol. 924, No. 1, p. 012009). IOP Publishing.
- Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2), 415-425.
- Ihsanuddin, 2018. 5 Industri Utama Disiapkan untuk Revolusi Industri 4.0 [WWW Document]. *KOMPAS.com*. URL. <https://ekonomi.kompas.com/read/2018/04/04/134003826/5-industri-utama-disiapkan-untuk-revolusi-industri-40>, diakses 4 September 2021.
- Ikhsanuddin, R. M. (2014). Identifikasi Citra pada Plat Nomor Kendaraan Mobil Pribadi Menggunakan Metode K-Nearest Neighbour.
- Indarto, I., & Murinto, M. (2017). Deteksi kematangan buah pisang berdasarkan fitur warna citra kulit pisang menggunakan metode transformasi ruang warna HIS. *JUITA: Jurnal Informatika*, 5(1), 15-21.
- Indrabulan, T. 2017. Perbandingan Metode Deteksi Objek Berbasis Video untuk Survei Arus Lalu Lintas. Universitas Hasanuddin-Makassar.
- Istiqomah, et.al. 2018. Analisis Pengembangan Klaster Hortikultura di Kabupaten Ngawi. *Jurnal Litbang Provinsi Jawa Tengah*, 16(1), 103-118.

- Kanan, C., & Cottrell, G. W. (2012). Color-to-grayscale: does the method matter in image recognition?. *PloS one*, 7(1), e29740.
- Kesumawati, A. 2018. Perbandingan Metode Support Vector Machine (SVM) Linear, Radial Basis Function (RBF), dan Polinomial Kernel dalam Klasifikasi Bidang Studi Lanjut Pilihan Alumni UII. Universitas Islam Indonesia-Yogyakarta.
- Khuriyati. (2020). Quality Assessment of Chilies (*Capsicum annum L.*) by Using a Smartphone Camera. In IOP Conference Series: Earth and Environmental Science (Vol. 425, No. 1, p. 012040). IOP Publishing.
- Kowalczyk, A. (2014). Linear Kernel: Why is it recommended for text classification ? .Diakses: 30 Agustus, 2021, internet <https://www.svmtutorial.com/2014/10/svm-linear-kernel-good-text-classification/>
- Liu, Z. 2011. A method of SVM with normalization in intrusion detection. *Procedia Environmental Sciences*, 11, 256-262.
- Mufti, S. K. 2018. Sistem Rekognisi Kantuk pada Pengendara Mobil Berbasis Android. Unhas-Makassar.
- Munir, R. 2006. Aplikasi Image Thresholding untuk Segmentasi Objek. In Seminar Nasional Aplikasi Teknologi Informasi (SNATI).
- Nugraha, A. S. (2019). Penerapan Metode Support Vector Machine Pada Part Of Speech Tag Bahasa Indonesia (Doctoral dissertation, Universitas Komputer Indonesia).
- Nurfalach, D. R. (2010). Di UPTD Perbibitan Tanaman Hortikultura Desa Pakopen Kecamatan Bandungan Kabupaten Semarang.
- Nurhikma, A. 2019. Prototype Sistem Klasifikasi Kematangan Stroberi Menggunakan Algoritma Support Vector Machine. Universitas Hasanuddin-Makassar.
- Octaviani, P. A., Wilandari, Y., & Ispriyanti, D. (2014). Penerapan Metode Klasifikasi Support Vector Machine (SVM) Pada Data Akreditasi Sekolah Dasar (SD) Di Kabupaten Magelang. *Jurnal Gaussian*, 3(4), 811-820.
- Patel, S. (2017). Chapter 2: SVM (Support Vector Machine)—Theory. *Dialses* 30

- Agustus, 2021, dari Machine Learning 101: <https://medium.com/machine-learning-101/chapter-2-svm-support-vectormachine-theory-f0812effc72>
- Permata, E., Purnama, K. E., & Purnomo, M. H. (2016). Klasifikasi Jenis dan Fase Parasit Malaria Plasmodium Falciparum dan Plasmodium Vivax Dalam Sel Darah Merah Menggunakan Support Vector Machine. *Setrum: Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer*, 1(2), 50-57.
- Prasetyo, E. (2011). *Pengolahan Citra Digital dan Aplikasinya menggunakan Matlab*. Yogyakarta: Andi.
- Pratt. 2007. *Digital Image Processing*, Los Altos, California, A John Wiley & Sons.
- Prianggodo, L. B., & Rohmah, R. N. (2016). *Perancangan Object Tracking Robot Berbasis Image Processing Menggunakan Raspberry Pi* (Doctoral dissertation, Universitas Muhammadiyah Surakarta).
- Pricila, J. M. 2016. *Perbandingan Beberapa Pendekatan Multiclass SVM Klasifikasi Artikel Berbahasa Indonesia*. Universitas Komputer Indonesia.
- Putra, D. (2010). *Pengolahan citra digital*. Penerbit Andi.
- Purnamawan, I. K. (2015). Support vector machine pada information retrieval. *Jurnal Pendidikan Teknologi dan Kejuruan*, 12(2), 139-146.
- Rai, P. 2011. *Kernel Methods and Nonlinear Classification*. Presentation.
- Santosa, B. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu, 978(979), 756.
- Sari, P. D. dkk. 2017. *Analisis Credit Scoring Menggunakan Regresi Logistik LASSO dan Support Vector Machine (SVM)* (Doctoral dissertation, Bogor Agricultural University (IPB)).
- Scholkopf, B., and Smola, A. J., *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- Sidehabi, S. W., Suyuti, A., Areni, I. S., & Nurtanio, I. (2018). The Development of Machine Vision System for Sorting Passion Fruit using Multi-Class Support Vector Machine. *Journal of Engineering Science & Technology Review*, 11(5).
- Subagyono, dkk. 2010. *Budidaya dan Pascapanen Cabai Merah (Capsicum annuum L.)*.
- Sugianto. 2015. *Skripsi. Pemutuan Buah Cabai Merah Besar (Capsicum annuum*

L.) Menggunakan Pengolahan Citra Digital dan Jaringan Syaraf Tiruan.
Universitas Jember-Jatim

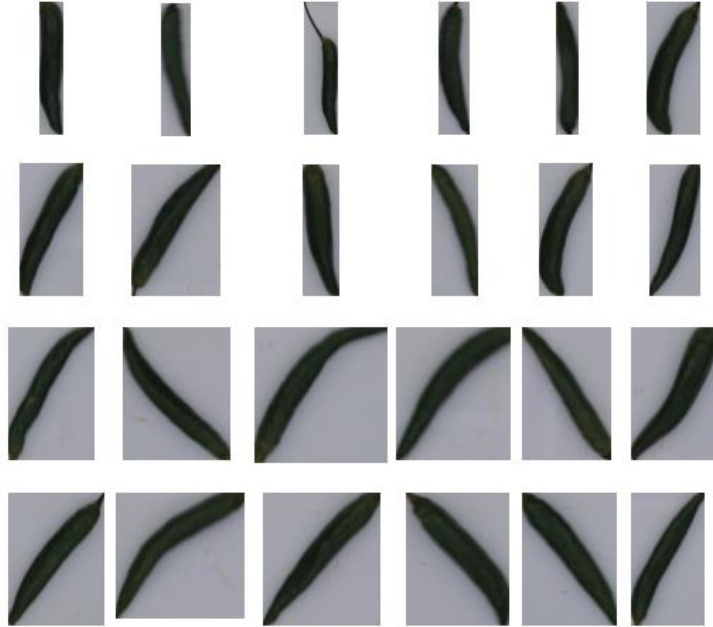
Sunandar, 2017. Perbaikan Kualitas Citra Menggunakan Metode Gaussian Filter.
MEANS (Media Informasi Analisa dan Sistem), 2(1), pp.19-22.

Susanto, A. (2019). Penerapan Operasi Morfologi Matematika Citra Digital untuk
Ekstraksi Area Plat Nomor Kendaraan Bermotor. Pseudocode, 6(1), 49-57.

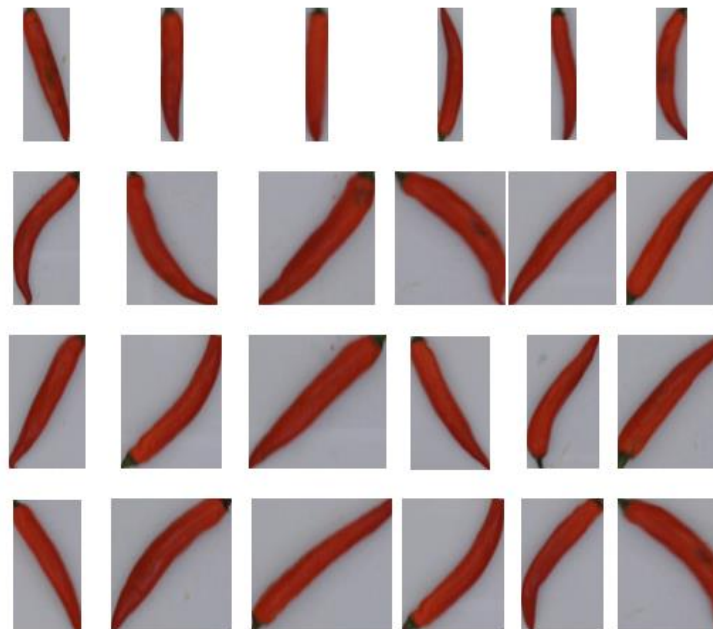
Suyanto, D. 2017. Data Mining untuk Klasifikasi dan Klasterisasi Data. Bandung:
Informatika Bandung.

LAMPIRAN

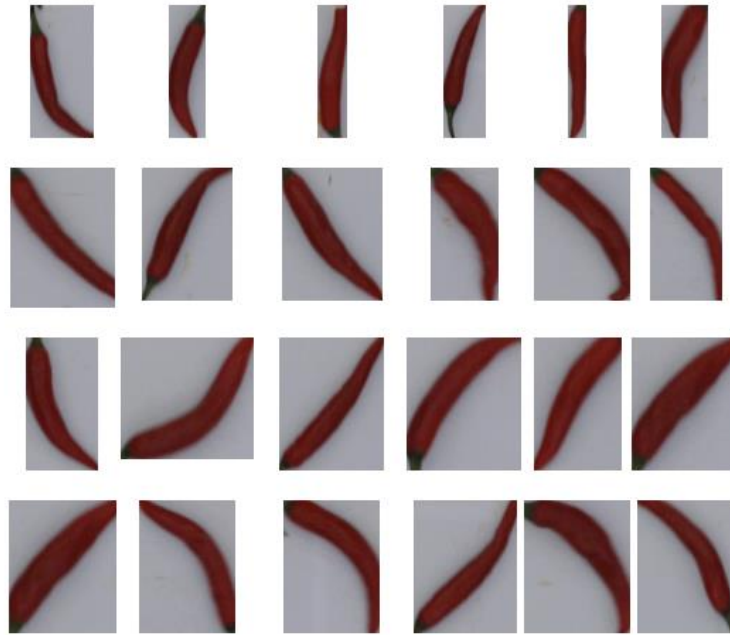
1. Contoh *frame* yang terpilih untuk proses *training*



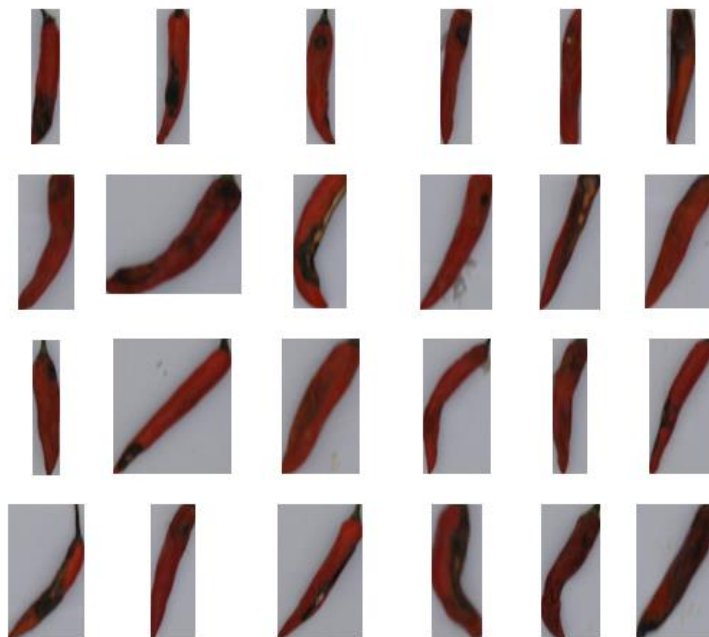
Frame cabai merah besar belum matang



Frame cabai merah besar setengah matang



Frame cabai merah besar matang



Frame cabai merah besar busuk

2. Nilai dari fitur $\bar{R}\bar{G}\bar{B}$ dan pelabelan kelas untuk data *training*

No	<i>File name</i>	mean_r	mean_g	mean_b	<i>Class</i>
1	Belum Matang1.jpg	27.487	34.922	28.746	1
2	Belum Matang2.jpg	27.161	35.029	28.906	1
3	Belum Matang3.jpg	27.28	34.962	28.944	1
4	Belum Matang4.jpg	27.503	37.779	30.453	1
5	Belum Matang5.jpg	27.334	37.646	30.248	1
6	Belum Matang6.jpg	27.57	37.664	30.284	1
7	Belum Matang7.jpg	28.287	37.968	30.982	1
8	Belum Matang8.jpg	28.17	37.648	30.395	1
9	Belum Matang9.jpg	28.414	37.555	30.493	1
10	Belum Matang10.jpg	27.41	36.152	29.13	1
11	Belum Matang11.jpg	26.847	35.819	28.583	1
12	Belum Matang12.jpg	26.831	35.633	28.411	1
13	Belum Matang13.jpg	30.11	41.127	32.142	1
14	Belum Matang14.jpg	29.784	40.971	31.917	1
15	Belum Matang15.jpg	29.635	40.86	32.015	1
16	Belum Matang16.jpg	26.47	37.107	29.691	1
17	Belum Matang17.jpg	26.692	37.02	30.112	1
18	Belum Matang18.jpg	28.078	37.188	30.223	1
19	Belum Matang19.jpg	26.702	36.349	30.251	1
20	Belum Matang20.jpg	26.46	36.039	29.534	1
21	Belum Matang21.jpg	26.383	36.064	29.563	1
22	Belum Matang22.jpg	29.218	41.489	33.458	1
23	Belum Matang23.jpg	29.056	41.256	33.229	1
24	Belum Matang24.jpg	28.372	41.088	32.941	1
25	Belum Matang25.jpg	29.907	40.014	32.674	1
26	Belum Matang26.jpg	29.849	39.729	32.014	1
27	Belum Matang27.jpg	29.252	39.204	31.663	1
28	Belum Matang28.jpg	29.057	40.633	32.803	1
29	Belum Matang29.jpg	29.456	40.756	32.79	1
30	Belum Matang30.jpg	29.489	40.631	32.702	1
31	Belum Matang31.jpg	28.404	36.188	29.247	1
32	Belum Matang32.jpg	27.954	35.712	28.542	1
33	Belum Matang33.jpg	27.177	34.967	27.915	1
34	Belum Matang34.jpg	29.152	38.906	31.217	1
35	Belum Matang35.jpg	29.053	38.899	30.952	1
36	Setengah Matang1.jpg	19.243	33.713	122.111	2
37	Setengah Matang2.jpg	19.097	33.596	121.328	2
38	Setengah Matang3.jpg	19.039	33.313	120.436	2

No	File name	mean_r	mean_g	mean_b	Class
39	Setengah Matang4.jpg	20.347	32.254	125.123	2
40	Setengah Matang5.jpg	20.56	32.356	124.402	2
41	Setengah Matang6.jpg	21.041	32.559	123.864	2
42	Setengah Matang7.jpg	21.115	35.235	133.278	2
43	Setengah Matang8.jpg	21.298	35.376	133.498	2
44	Setengah Matang9.jpg	21.173	35.235	132.928	2
45	Setengah Matang10.jpg	19.476	33.515	135.668	2
46	Setengah Matang11.jpg	19.967	33.626	135.987	2
47	Setengah Matang12.jpg	20.159	33.813	135.088	2
48	Setengah Matang13.jpg	18.159	35.675	138.362	2
49	Setengah Matang14.jpg	18.242	35.635	138.493	2
50	Setengah Matang15.jpg	18.974	35.838	138.428	2
51	Setengah Matang16.jpg	19.326	32.717	129.962	2
52	Setengah Matang17.jpg	19.014	32.554	129.173	2
53	Setengah Matang18.jpg	18.872	32.335	128.574	2
54	Setengah Matang19.jpg	21.568	32.633	123.069	2
55	Setengah Matang20.jpg	20.782	31.932	122.2	2
56	Setengah Matang21.jpg	20.193	31.614	120.91	2
57	Setengah Matang22.jpg	20.234	32.646	129.963	2
58	Setengah Matang23.jpg	19.207	32.227	130.099	2
59	Setengah Matang24.jpg	18.922	32.156	129.851	2
60	Setengah Matang25.jpg	18.616	32.121	126.766	2
61	Setengah Matang26.jpg	18.488	31.795	124.409	2
62	Setengah Matang27.jpg	18.634	31.486	123.091	2
63	Setengah Matang28.jpg	19.79	33.141	128.205	2
64	Setengah Matang29.jpg	20.088	33.213	128.264	2
65	Setengah Matang30.jpg	20.86	33.499	127.521	2
66	Setengah Matang31.jpg	21.389	34.943	130.604	2
67	Setengah Matang32.jpg	21.419	35.089	130.6	2
68	Setengah Matang33.jpg	21.747	35.513	129.423	2
69	Setengah Matang34.jpg	22.557	32.879	125.727	2
70	Setengah Matang35.jpg	22.136	32.738	125.797	2
71	Matang1.jpg	22.777	26.621	82.857	3
72	Matang2.jpg	22.23	26.009	82.386	3
73	Matang3.jpg	21.823	25.432	81.314	3
74	Matang4.jpg	22.544	25.338	86.954	3
75	Matang5.jpg	22.175	25.111	86.449	3
76	Matang6.jpg	21.862	25.078	86.143	3
77	Matang7.jpg	22.226	25.681	86.999	3
78	Matang8.jpg	21.798	25.559	87.16	3

No	File name	mean_r	mean_g	mean_b	Class
79	Matang9.jpg	22.211	25.394	86.937	3
80	Matang10.jpg	21.891	25.437	86.549	3
81	Matang11.jpg	21.505	24.895	86.292	3
82	Matang12.jpg	21.109	24.564	86.258	3
83	Matang13.jpg	22.857	25.523	81.874	3
84	Matang14.jpg	22.858	25.501	81.572	3
85	Matang15.jpg	22.897	25.533	80.853	3
86	Matang16.jpg	24.131	27.377	91.113	3
87	Matang17.jpg	23.731	27.797	91.131	3
88	Matang18.jpg	23.604	27.363	91.319	3
89	Matang19.jpg	24.663	27.248	89.59	3
90	Matang20.jpg	24.7	27.301	89.305	3
91	Matang21.jpg	24.431	27.471	89.069	3
92	Matang22.jpg	21.29	25.104	78.836	3
93	Matang23.jpg	20.502	24.06	78.01	3
94	Matang24.jpg	20.564	24.292	77.795	3
95	Matang25.jpg	22.383	25.854	83.711	3
96	Matang26.jpg	21.791	25.42	82.341	3
97	Matang27.jpg	21.991	25.392	81.848	3
98	Matang28.jpg	22.968	26.586	91.763	3
99	Matang29.jpg	22.74	26.169	91.895	3
100	Matang30.jpg	23.098	26.234	92.509	3
101	Matang31.jpg	20.476	26.166	96.415	3
102	Matang32.jpg	20.369	25.669	96.81	3
103	Matang33.jpg	19.885	25.71	96.745	3
104	Matang34.jpg	23.923	28.252	94.019	3
105	Matang35.jpg	24.231	27.965	93.584	3
106	Busuk1.jpg	19.161	26.076	76.416	4
107	Busuk2.jpg	19.567	26.208	76.194	4
108	Busuk3.jpg	20.06	26.321	75.743	4
109	Busuk4.jpg	19.742	25.575	70.372	4
110	Busuk5.jpg	19.493	25.402	69.615	4
111	Busuk6.jpg	19.639	25.419	68.802	4
112	Busuk16.jpg	20.293	30.069	90.501	4
113	Busuk17.jpg	20.359	29.627	89.585	4
114	Busuk18.jpg	20.05	29.641	88.691	4
115	Busuk22.jpg	22.649	29.968	75.677	4
116	Busuk23.jpg	22.586	28.928	83.914	4
117	Busuk24.jpg	22.548	29.485	78.066	4
118	Busuk28.jpg	20.345	29.28	86.388	4

No	File name	mean_r	mean_g	mean_b	Class
119	Busuk29.jpg	20.366	29.019	86.018	4
120	Busuk30.jpg	20.163	28.945	85.331	4
121	Busuk31.jpg	20.07	29.579	68.756	4
122	Busuk32.jpg	19.51	29.143	68.484	4
123	Busuk33.jpg	19.144	28.67	67.444	4
124	Busuk34.jpg	18.569	27.506	88.364	4
125	Busuk35.jpg	18.439	27.159	87.687	4
126	Busuk36.jpg	18.476	27.217	86.9	4
127	Busuk37.jpg	20.463	29.436	84.784	4
128	Busuk38.jpg	20.093	29.086	83.773	4
129	Busuk39.jpg	20.3	29.066	83.242	4
130	Busuk49.jpg	21.49	32.194	93.693	4
131	Busuk50.jpg	20.795	31.789	93.202	4
132	Busuk51.jpg	20.819	31.567	93.224	4
133	Busuk52.jpg	19.308	28.183	95.891	4
134	Busuk53.jpg	19.251	28.088	95.537	4
135	Busuk54.jpg	19.551	28.211	95.404	4
136	Busuk55.jpg	20.779	30.745	93.656	4
137	Busuk56.jpg	20.876	30.647	93.247	4
138	Busuk57.jpg	20.565	30.337	92.905	4
139	Busuk58.jpg	21.41	29.808	69.813	4
140	Busuk59.jpg	21.412	29.645	69.562	4

3. Contoh Source Code

```
# Menggunakan library
import os
import cv2
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline

# Membuat fungsi menghitung nilai rata2 rgb (Ekstraksi Fitur)
def get_mean_rgb(hasilcrop, mask2):
    # inisialisasi variabel
    mean_red = 0
    mean_green = 0
    mean_blue = 0

    #total pixel berwarna putih
    total = 0

    # Mengambil nilai dari gambar 3 channel dan menjumlahkannya
    for i in range(len(hasilcrop)):
        for j in range(len(hasilcrop[0])):
            if mask2[i][j] == 255:
                total = total + 1
                mean_red = mean_red + hasilcrop[i][j][0]
                mean_green = mean_green + hasilcrop[i][j][1]
                mean_blue = mean_blue + hasilcrop[i][j][2]
    # Hasil dari jumlah dibagi dengan total dari masing2 channel
    if total > 0:
        mean_red = round((mean_red / total),3)
        mean_green = round((mean_green / total),3)
        mean_blue = round((mean_blue / total),3)

    return [mean_red, mean_green, mean_blue]
#total pixel berwarna putih
total = 0
```

```
# Mengambil nilai dari gambar 3 channel dan menjumlahkannya
for i in range(len(hasilcrop)):
    for j in range(len(hasilcrop[0])):
        if mask2[i][j] == 255:
            total = total + 1
            mean_red = mean_red + hasilcrop[i][j][0]
            mean_green = mean_green + hasilcrop[i][j][1]
            mean_blue = mean_blue + hasilcrop[i][j][2]
    # Hasil dari jumlah dibagi dengan total dari masing2 channel
    if total > 0:
        mean_red = round((mean_red / total),3)
        mean_green = round((mean_green / total),3)
        mean_blue = round((mean_blue / total),3)

    return [mean_red, mean_green, mean_blue]
```

```

# Membuat fungsi untuk deteksi berdasarkan fitur warna
def deteksiwarna(frame):
    # Untuk mengaburkan gambar menggunakan GaussianBlur
    blurred = cv2.GaussianBlur(frame, (3, 3), 0)

    # Mengonversi dari gambar BGR ke HSV
    hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)

    # Membuat ambang batas warna yang akan dideteksi
    thresh1 = cv2.inRange(hsv, (0,0,0), (179, 255, 60))
    #warna merah
    thresh2 = cv2.inRange(hsv, (0,159,0), (179, 255, 255))
    # Menggabungkan kedua ambang batas
    mask = cv2.bitwise_or(thresh1, thresh2)

    return mask

```

Source Code : Ekstraksi Fitur

```

# Membuat fungsi untuk dataset
def create_dataset():
    # Mengambil data gambar dari tiap folder kelas
    folders = ['Belum Matang', 'Setengah Matang', 'Matang', 'Busuk']
    # Membuat variabel kolom untuk dataset
    names = ['filename', 'mean_r', 'mean_g', 'mean_b', 'Class']
    # Membuat dataframe berdasarkan nama kolom yang dibuat
    df = pd.DataFrame([], columns=names)
    for folder in folders:
        # Memanggil data gambar berdasarkan path
        path = 'E:/KULIAH/BISMILLAH/GOGOGO/New Dataset/Training/' +
        folder
        # Melakukan arah list berdasarkan path
        files = os.listdir(path)
        # os.makedirs(path)
        # Mengambil setiap gambar
        for file in files:
            imgpath = path + '/' + file
            # Membaca gambar
            main_img = cv2.imread(imgpath)
            # Mengubah ukuran gambar
            img1 = cv2.resize(main_img, (960,540))

            # Mengatur pencahayaan dari gambar
            alpha = 1.1 # Contrast control (1.0-3.0)
            beta = 0 # Brightness control (0-100)

            # Mengonversi skala pencahayaan dari gambar
            img1 = cv2.convertScaleAbs(img1, alpha=alpha, beta=beta)

            # Memanggil fungsi deteksi warna
            detect = deteksiwarna(img1)
            # Inisialisasi kernel
            kernel = np.array((15,15))
            # Morfologi gambar
            oke = cv2.erode(detect, kernel, iterations=4)

```

```

        areaArray = []
        # Mencari kontur
        contours, _ = cv2.findContours(oke, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

        im = np.copy(img1)
        for n, mask in enumerate(contours):
            # Kontur area
            area = cv2.contourArea(mask)
            # Menyimpan area kontur di areaArray
            areaArray.append(area)
            # Mencari area kontur terbesar
            areaLargest1 = np.argmax(areaArray)
            areaLargestMax1 = max(areaArray)
            areaLargestCnt1 = contours[areaLargest1]

            # Melakukan boundingRect
            x, y, w, h = cv2.boundingRect(areaLargestCnt1)

            # Membatasi titik y yang akan diambil
            if y > 5 and y < 300:
                # Kondisi luas terbesar diatas 1000
                if areaLargestMax1 > 1000 :
                    # Melakukan boundingbox
                    boundingbox = cv2.rectangle(im, (x - 1, y -
1), (x + w, y + h), (0, 255, 255), 1)
                    #cv2.imshow("nom_" + listfile[i], boundingbox)
            # Melakukan pemotongan dari objek yang diteksi
d disesuaikan dengan masknya
            hasilcrop = im[y: y + h, x:x + w]
            mask2 = oke[y: y + h, x:x + w]

            lower = 0
            upper = 0
            fold = 0

            # Mengubah dari nama folder menjadi angka
            if folder == 'Busuk':
                fold = 4
            elif folder == 'Matang':
                fold = 3
            elif folder == 'Setengah Matang':
                fold = 2
            else:
                fold = 1
            # Menghitung rata2 rgb dari fungsi yang telah dibuat
            mean_rgb = get_mean_rgb(hasilcrop, mask2)
            # Membuat dataset berdasarkan variabel kolom
            vector = [file] + mean_rgb + [fold]
            df_temp = pd.DataFrame([vector], columns=names)
            df = df.append(df_temp)

        return df

```

```

# Memanggil fungsi create_dataset
dataset = create_dataset()

dataset.to_csv("Training_versi2.csv", index=False,
index_label=False, mode='a')

```

Source Code: Training

```

# Library
# Untuk mengolah data
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

# Untuk mengimport SVM
from sklearn import svm

# Untuk digunakan pada SVM dengan parameter tuning
from sklearn.model_selection import GridSearchCV
# Standarisasi dengan metode StandardScaler
from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA
# Untuk memanggil metrik akurasi
from sklearn import metrics
# Untuk visualisasi data
import seaborn as sns
import matplotlib.pyplot as plt

# Menyimpan model yang akan digunakan untuk testing
from joblib import dump, load

# Menghitung nilai akurasi untuk model
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix

# Menyimpan fitur atribut ke dalam variabel X_train
X = df_train.drop(labels = ['filename','Class'],axis = 1)
# Menyimpan class (label) pada y_train
y = df_train['Class']

# Hyperparameter yang akan di tuning
parameters = [{'kernel':['rbf'],
                'gamma': [1e-5,1e-4,1e-3,0.01,0.1,0.2,0.5],
                'C':[1e-3,0.01,0.1,1,10,100]}
              ]

# Menggunakan Gridsearch dengan memanggil class SVC
param_grid = GridSearchCV(svm.SVC(),parameters,cv=5)
#melakukan training pada objek dan label
param_grid.fit(X_train, y_train)

```

```

# Hasil hyperparameter tuning dengan skor terbaik yang di dapatkan
print(f"Best parameter {param_grid.best_params_} with score
{param_grid.best_score_}")

# Menampilkan hasil dari model SVM berdasarkan Hyper Parameter
Tuning
means = param_grid.cv_results_['mean_test_score']
stds = param_grid.cv_results_['std_test_score']
for mean, std, params in zip(means, stds,
param_grid.cv_results_['params']):
    print('%0.3f (+/-%0.03f) for %r' % (mean, std * 2, params))

# Melakukan prediksi pada data testing
y_pred1 = param_grid.predict(X_test)
print('hasil prediksi train dari best_param : ',
metrics.accuracy_score(y_test, y_pred1))

# Menampilkan hasil klasifikasi
print(metrics.classification_report(y_test, y_pred1))

accuracy1 = metrics.accuracy_score(y_test, y_pred1)

conf1= confusion_matrix(y_test, y_pred1)

model1 = svm.SVC(decision_function_shape='ovr', C= 100, gamma = 0.2,
kernel = 'rbf')
model1.fit(X_train, y_train)

from sklearn.multiclass import OneVsRestClassifier

clf_2 = OneVsRestClassifier(svm.SVC(C= 100, gamma = 0.2, kernel =
'rbf'))
clf_2.fit(X_train, y_train)
# Melakukan prediksi pada data testing
y_pred3 = clf_2.predict(X_test)
metrics.accuracy_score(y_test, y_pred3)

# Menyimpan nama model yang akan digunakan
filename = 'Svm_ova_v1.sav'
dump(clf_2, open(filename, 'wb'))

```

Source Code: Testing

```

def klasifikasi(hasil, roi, x, y):
    if(hasil[0] == 1):
        label = cv2.putText(roi, "BM", (x , y - 10),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 0),2)
    elif (hasil[0] == 2):
        label = cv2.putText(roi, "SM", (x, y - 10),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 0), 2)
    elif (hasil[0] == 3):
        label = cv2.putText(roi, "M", (x, y - 10),
cv2.FONT_HERSHEY_COMPLEX, 1,(255, 255, 0), 2)
    else :
        label = cv2.putText(roi, "B", (x, y - 10),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 0),2)

```



```

from sklearn.svm import SVC
from joblib import dump, load
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

# Memuat file train
df = pd.read_csv('E:/KULIAH/TUGAS AKHIR/VERSI2/Training_versi2.csv')

# Memisahkan label dan target dari dataset
X = df.drop(['filename', 'Class'], axis=1)
y = df['Class']

# Menggunakan Fungsi StandarScaler
sc_X = StandardScaler()

# Train data label (x)
X_train = sc_X.fit_transform(X)

# Load Model
svm_model = open('E:/KULIAH/TUGAS AKHIR/OVA/Svm_ova_v1.sav', 'rb')
svm_from_joblib = load(svm_model)

#Memprediksi label training
y_pred = svm_from_joblib.predict(X_train)
accuracy=accuracy_score(y,y_pred)
print("Akurasi training:" ,round(accuracy,3))

count = 0
counts = 0
video_capture = cv2.VideoCapture('E:/KULIAH/BISMILLAH/GOGOGO/VIDEO
BARU/testCampur1.mp4')

if (video_capture.isOpened() == False):
    print("Error reading video file")

frame_width = int(video_capture.get(3)/2)
frame_height = int(video_capture.get(4)/2)

size = (frame_width, frame_height)

# result = cv2.VideoWriter('HASIL.avi',
#                             cv2.VideoWriter_fourcc(*'MJPG'),
#                             15.0, size)

while True:
    ret, frame = video_capture.read()
    if ret:
        height, weight, _ = frame.shape
        ratio = 0.5
        frame = cv2.resize(frame, (0, 0), None, ratio, ratio)
        alpha = 1.2 #Contrast control (1.0-3.0)
        beta = 0 # Brightness control (0-100)
        frame = cv2.convertScaleAbs(frame, alpha=alpha, beta=beta)

```

```

x1= 220
y1= 0

x2= 730
y2= 540

roi = frame[y1:y2, x1:x2]

detect = warnaHsv(roi)
gray =cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
dst = cv2.Canny(gray, 20, 50, None, 3)
dst = 255-dst
oke = cv2.bitwise_and(dst,detect)
kernel = np.array((20,20))
oke = cv2.erode(oke, kernel, iterations=7)
cnts= cv2.findContours(oke.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)

threshold_min_area = 1000
threshold_max_area = 13500

n = len(cnts)
hasilcrop2 = []
mask3 = []

for i in range(n):
    c = cnts[i]
    x, y, w, h = cv2.boundingRect(c)
    if w < 57.5 or h < 50 :
        continue

    if y in range(120,270):
        mask1 = np.zeros((roi.shape[0],roi.shape[1])) # buat
background hitam
        mask1.fill(0)

        #Menggambar kontur
        cv2.drawContours(mask1, [c], -1, (255, 255, 255), 1)

        masks = cv2.fillPoly(mask1, [c], [255,255,255])
        masks = np.uint8(masks)
        area = cv2.contourArea(c)
        if area > threshold_min_area and area <
threshold_max_area:
            cv2.rectangle(roi, (x ,y ), (x + w, y + h), (0,
255, 0), 1) #[1]
            hasilcrop = roi[y: y + h, x:x + w]
            mask2 = masks[y: y + h, x:x + w]

```

```

        else:
            if h > 360:
                h = int(h/2)
                cv2.rectangle(roi, (x , y ), (x + w, y + h),
(255, 0, 255), 1) #[2]
                cv2.rectangle(roi, (x, y + h + 3), (x + w ,
y + h + h), (255, 0, 255), 1) #[5]
                hasilcrop = roi[y: y + h, x:x + w]
                mask2 = masks[y: y + h, x:x + w]
                hasilcrop2 = roi[y + h: y + h + h, x: x + w]
                mask3 = masks[y + h: y + h + h, x: x + w]
            else:
                break

            mean_rgb1 = get_mean_rgb(hasilcrop2,mask3)
            y_test1 = sc_X.transform(np.array([mean_rgb1]))
            y_hasil1 = svm_from_joblib.predict(y_test1)
            klasifikasi(y_hasil1, roi, x, y+w) #[3]

            mean_rgb = get_mean_rgb(hasilcrop,mask2)
            y_test = sc_X.transform(np.array([mean_rgb]))
            y_hasil = svm_from_joblib.predict(y_test)
            klasifikasi(y_hasil, roi, x, y) #[4]

        # result.write(frame)
        cv2.imshow('Video', frame)
        cv2.imshow('ROI', roi)
        cv2.imshow('Mask', oke)

        if cv2.waitKey(20) & 0xFF == ord('q'):
            break
    else:
        break
# result.release()
video_capture.release()
cv2.destroyAllWindows()

```

LEMBAR PERBAIKAN SKRIPSI

“KLASIFIKASI MUTU CABAI MERAH BESAR (*Capsicum Annuum* L.)

BERBASIS VIDEO PROCESSING”





OLEH:

TUTI AMALIA
D42116014


Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 25 Mei 2022.

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

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Indrabayu, S.T., M.T., M.Bus.Sys.	
Sekretaris	Dr. Ir. Ingrid Nurtanio, M.T.	
Anggota	Elly Warni, S.T., M.T.	
	A. Ais Prayogi Alimuddin, S.T., M.Eng.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Indrabayu, S.T., M.T., M.Bus.Sys.	
II	Dr. Ir. Ingrid Nurtanio, M.T.	