



TUGAS AKHIR - TF 181801

PREDIKSI TINGKAT KEMATANGAN BUAH APEL BERBASIS *GLCM*, *CIE*
LAB, DAN *MACHINE LEARNING*

GABRIEL VIERI

NRP. 02311740000072

Dosen Pembimbing:

Dr.rer.nat. Ir. Aulia Muhammad Taufiq Nasution, M.Sc.

DEPARTEMEN TEKNIK FISIKA

Fakultas Teknologi Industri dan Rekayasa Sistem

Institut Teknologi Sepuluh Nopember

Surabaya 2022

Halaman ini sengaja dikosongkan



FINAL PROJECT - TF 181801

***PREDICTION OF APPLE RIPENESS BASED ON GLCM, CIE LAB, AND
MACHINE LEARNING***

GABRIEL VIERI

NRP. 02311740000072

Supervisors:

Dr.rer.nat. Ir. Aulia Muhammad Taufiq Nasution, M.Sc.

DEPARTMENT OF ENGINEERING PHYSICS

Faculty of Industrial Technology and System Engineering

Institut Teknologi Sepuluh Nopember

Surabaya 2022

Halaman ini sengaja dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini.

Nama : Gabriel Vieri
NRP : 0231174000072
Departemen / Prodi : Teknik Fisika / SI Teknik Fisika
Fakultas : Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “**Prediksi Tingkat Kematangan Buah Apel berbasis *GLCM, CIE LAB, dan Machine Learning***” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya-benarnya.

Surabaya, 14 Juli 2022

Yang membuat pernyataan,



Gabriel Vieri

NRP. 0231174000072

Halaman ini sengaja dikosongkan

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**PREDIKSI TINGKAT KEMATANGAN BUAH APEL BERBASIS GLCM, CIE
LAB, DAN MACHINE LEARNING**

Oleh:

Gabriel Vieri

NRP. 0231174000072

Surabaya,

**Menyetujui,
Pembimbing I**



Dr. rer.nat. Aulia M.T. Nasution
Ph.D., Pendidikan, Engineering, Surabaya
0815 822943, IN

Dr. rer.nat. Ir. Aulia Muhammad Taufiq Nasution, M.Sc.

NIP. 19671117 199702 1 001

**Mengetahui,
Kepala Departemen
Teknik Fisika FTIRS-ITS**



Dr. Suyanto, S.T., M.T.
NIP. 19741113 199512 1 002

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

PREDIKSI TINGKAT KEMATANGAN BUAH APEL BERBASIS GLCM, CIE LAB, DAN MACHINE LEARNING

TUGAS AKHIR





Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Institut Teknologi Sepuluh Nopember

Oleh:

GABRIEL VIERI

NRP. 0231174000072

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr.rer.nat. Ir. Aulia Muhammad, M.Sc.  (Pembimbing I)
2. Detak Yan Pratama, S.T., M.Sc.  (Ketua Penguji)
3. Iwan Cony Setiadi, S.T., M.T.  (Penguji I)
4. Dr. Katherin Inderawati, S.T., M.T.  (Penguji II)

SURABAYA

2022

Halaman ini sengaja dikosongkan

KLASIFIKASI TINGKAT KEMATANGAN BUAH APEL BERBASIS GLCM, CIE LAB, DAN MACHINE LEARNING

Nama : Gabriel Vieri
NRP : 0231174000072
Departemen : Teknik Fisika FTIRS - ITS
Dosen Pembimbing : Dr.rer.nat. Ir. Aulia Muhammad Taufiq
Nasution, M.Sc.

ABSTRAK

Buah apel sebagai salah satu komoditas pertanian yang disukai karena tekstur, gizi, rasa, dan daya tarik visual. Apel dapat diklasifikasi berdasarkan tingkat kematangannya dari tekstur dan warna kulit buah apel. Analisa tekstur dilakukan dengan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM), skala warna CIE Lab untuk menganalisa warna, dan mengkombinasikan fitur keduanya dalam algoritma *machine learning*. Penentuan tingkat kematangan buah apel secara fisiologis dilakukan dengan menghitung *starch conversion rate* yang terbentuk akibat reaksi dengan larutan *iodine*. Gambar dilakukan *preprocessing* dengan *crop* dan *resize*. Fitur yang diekstrak oleh GLCM pada jarak 1, 2, 3 dan sudut 0° , 45° , 90° , 135° adalah kontras, ASM, energi, homogeneitas, dan disimilaritas. Pelatihan model untuk klasifikasi tingkat kematangan dilakukan dengan skenario penggunaan seleksi fitur dan reduksi dimensi sebelum masuk sepuluh algoritma *machine learning*. Seleksi fitur dilakukan dengan menghitung Z-value pada *Standard Error of difference means* (SEd). Untuk reduksi dimensi fitur dilakukan analisa PCA. Hasilnya adalah dengan algoritma *machine learning* seperti *Nearest Neighbors*, *Linear SVM*, *Gaussian Process*, *Decision Tree*, *Random Forest*, *Neural Net*, *AdaBoost*, *Naive Bayes*, dan *QDA* didapatkan akurasi 100%. Sehingga klasifikasi apel berdasarkan tingkat kematangan dapat dilakukan dengan menggunakan GLCM, Z-value SEd, dan analisa PCA.

Kata Kunci: CIE Lab, GLCM, kematangan apel, machine learning

Halaman ini sengaja dikosongkan

**CLASSIFICATION OF APPLE RIPENESS BASED ON GLCM, CIE
LAB, DAN MACHINE LEARNING**

Name : Gabriel Vieri
NRP : 02311740000072
Department : Engineering Physics FTIRS - ITS
Supervisors : Dr.rer.nat. Ir. Aulia Muhammad Taufiq
Nasution, M.Sc.

ABSTRACT

Apples are one of the preferred agricultural commodities because of their texture, nutrition, taste, and visual appeal. Apples can be classified based on the level of maturity of the texture and color of the skin of the apple. Texture analysis is performed by feature extraction of Gray Level Co-occurrence Matrix (GLCM), CIE Lab color scale to analyze color, and combining both features in machine learning algorithms. Determination of the level of physiological maturity of apples is done by calculating the starch conversion rate formed by the reaction with iodine solution. The image is preprocessed by cropping and resizing. The features extracted by GLCM at distances 1, 2, 3 and angles 0°, 45°, 90°, 135° are contrast, ASM, energy, homogeneity, and dissimilarity. Model training for maturity level classification is carried out by using feature selection and dimension reduction scenarios before entering ten machine learning algorithms. Feature selection is done by calculating the Z-value on the Standard Error of difference means (SEd). For feature dimension reduction, PCA analysis was performed. The result is that machine learning algorithms such as Nearest Neighbors, Linear SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes, and QDA get 100% accuracy. Therefore, the classification of apples based on the level of ripeness can be done using GLCM, Z-value SEd, and PCA analysis.

Keywords: *apple maturity, GLCM, CIE Lab, machine learning*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang senantiasa melimpahkan rahmat dan berkat-Nya, sehingga penulisan tugas akhir ini dapat diselesaikan. Tugas akhir ini ditulis sebagai salah satu syarat untuk menyelesaikan program pendidikan Sarjana Teknik Fisika, Fakultas Teknologi Industri dan Rekayasa Sistem, Institut Teknologi Sepuluh Nopember Surabaya.

Selama penulisan tugas akhir ini, penulis menyadari dalam penyusunan proposal skripsi ini tidak akan selesai tanpa bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Dr. Suyanto, S.T., M.T. selaku Kepala Departemen Teknik Fisika, Institut Teknologi Sepuluh Nopember.
2. Dr.rer.nat. Ir. Aulia M. T. Nasution, M.Sc. selaku dosen pembimbing tugas akhir yang telah membimbing dan mengarahkan pengerjaan tugas akhir dari awal hingga akhir.
3. Lizda Johar Mawarani, S.T., M.T. selaku dosen wali atas bimbingan, saran, dan motivasi yang diberikan pada penulis.
4. Segenap dosen Departemen Teknik Fisika, Institut Teknologi Sepuluh Nopember yang telah memberikan ilmunya kepada penulis.
5. Orang tua dan keluarga atas doa, bimbingan, serta kasih sayang yang selalu tercurah selama ini.
6. Keluarga besar Institut Teknologi Sepuluh Nopember khususnya teman-teman di Departemen Teknik Fisika atas semua dukungan, semangat, serta kerjasamanya.

Serta pihak-pihak lain yang tidak dapat disebutkan satu-persatu. Semoga laporan tugas akhir ini dapat dipergunakan dengan sebaik-baiknya.

Surabaya, 14 Juli 2022

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
COVER PAGE	iii
PERNYATAAN BEBAS PLAGIASI	v
LEMBAR PENGESAHAN	vii
LEMBAR PENGESAHAN	ix
ABSTRAK.....	xi
ABSTRACT.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Relevansi Penelitian	2
1.6 Sistematika Laporan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Review Penelitian Sebelumnya.....	5
2.2 Dasar Teori.....	7
2.2.1 Apel.....	7
2.2.2 Computer Vision.....	9
2.2.3 Skala Warna.....	10
2.2.4 K-means Clustering	11
2.2.5 Starch Conversion Rate	12
2.2.6 Gray-Level Co-occurrence Matrix (GLCM)	13
2.2.7 Seleksi Fitur	15
2.2.8 Machine Learning	17
BAB III METODOLOGI PENELITIAN	21
3.1 <i>Flowchart</i> Penelitian	21
3.2 Studi Literatur	21
3.3 Pengambilan Citra.....	22
3.4 Penentuan Tingkat Kematangan	23
3.5 Preprocessing Citra	26

3.6	RGB to CIE Lab	27
3.7	RGB to Greyscale.....	27
3.8	Ekstraksi Fitur	27
3.9	Training	27
3.10	Penulisan Laporan	29
BAB IV HASIL DAN PEMBAHASAN		31
4.1	Analisa Tingkat Kematangan Apel	31
4.2	Analisa Ekstraksi Fitur GLCM dan CIE Lab	34
4.3	Analisa SED dan PCA.....	37
4.4	Analisa Machine Learning	38
BAB V KESIMPULAN DAN SARAN.....		41
5.1	Kesimpulan.....	41
5.2	Saran.....	41
DAFTAR PUSTAKA		43
LAMPIRAN.....		xliv
A.	Image Segmentation - K-means Clustering.....	xliv
B.	GLCM	xlvii
C.	PCA and Machine Learning	lii
BIODATA PENULIS		lv

DAFTAR GAMBAR

Gambar 2.1 Titik hitam adalah kebun apel berada di daerah Batu yang dikelilingi 3 Gunung (A=G. Kawi, B=G. Anjasmoro, C=G. Arjuna) (Makmur Sitompul and Sugito, 2013).....	8
Gambar 2.2 Perbedaan warna dari tingkat kematangan apel (<i>Malus domestica</i>), fase 1: belum ranum, fase 2: ranum, fase 3: tua (Cárdenas-Pérez <i>et al.</i> , 2017).....	8
Gambar 2.3 Gray level dari apel belum ranum (Kiri), apel ranum (Kanan), dan grafik gray level dibanding jumlah pixel (Duprat <i>et al.</i> , 1995).....	9
Gambar 2.4 Contoh k-means clustering (Nagesh Singh Chauhan, 2019)	12
Gambar 2.5 Proses mendapatkan SCR	12
Gambar 2.6 Starch-Iodine Index (Blanpied and Silsby, 1992).....	13
Gambar 2.7 Proses mengubah matriks ke matriks GLCM (Haralick, Shanmugam and Dinstein, 1973)	14
Gambar 2.8 Hughes Phenomenon	16
Gambar 2.9 Confusion matrix	18
Gambar 3.1 Diagram Alir Metodologi Penelitian	21
Gambar 3.2 Setup pengambilan data	22
Gambar 3.3 Gambar original yang diambil dari kamera	22
Gambar 3.4 Proses penentuan tingkat kematangan apel.....	23
Gambar 3.5 Iodine, KI, dan air	24
Gambar 3.6 Kristal iodida.....	24
Gambar 3.7 Hasil larutan iodine	24
Gambar 3.8 Memotong apel	25
Gambar 3.9 Potongan apel setelah dicelupkan ke larutan iodine	25
Gambar 3.10 Hasil clustering potongan apel.....	25
Gambar 3.11 Proses perhitungan starch conversion rate	26
Gambar 3.12 Proses preprocessing gambar	26
Gambar 3.13 Contoh hasil k-means clustering apel	27
Gambar 3.14 Flowchart training data	27
Gambar 4.1 Sampel hasil tes iodine.....	31
Gambar 4.2 Hasil pengelompokkan k-means clustering	31
Gambar 4.3 Starch conversion rate buah apel	32
Gambar 4.4 Starch conversion rate sampel uji	32

Gambar 4.5 Foto apel setelah dikelompokkan.....	33
Gambar 4.6 RGB dari foto tiap apel	34
Gambar 4.7 Hasil ekstrasi GLCM fitur kontras	35
Gambar 4.8 Hasil ekstrasi GLCM fitur disimilaritas	35
Gambar 4.9 Hasil ekstrasi GLCM fitur homogeneitas	35
Gambar 4.10 Hasil ekstrasi GLCM fitur ASM	36
Gambar 4.11 Hasil ekstrasi GLCM fitur energi.....	36
Gambar 4.12 Hasil ekstraksi CIE Lab	36
Gambar 4.13 Standard error of difference between means seluruh fitur	37
Gambar 4.14 Perbedaan ada tidaknya SEd sebelum PCA	38
Gambar 4.15 Machine Learning tanpa seleksi fitur SEd (dari kiri atas Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes, QDA)	40
Gambar 4.16 Machine Learning dengan seleksi fitur SEd (dari kiri atas Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes	40

DAFTAR TABEL

Tabel 2.1 Penelitian penunjang.....	5
Tabel 2.2 Kelompok algoritma machine learning.....	19
Tabel 3.1 Detail pengambilan data	23
Tabel 3.2 Detail pengaturan algoritma machine learning.....	29
Tabel 4.1 Pembagian sesuai kelompok dari (Blanpied and Silsby, 1992).....	32
Tabel 4.2 Hasil seleksi fitur SEd pada GLCM	38
Tabel 4.3 Hasil traning berbagai metode	39

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Buah apel merupakan salah satu komoditas pertanian yang disukai di seluruh dunia karena tekstur, gizi, rasa, dan daya tarik visual. Apel sebagai tanaman buah tahunan tumbuh pada daerah sub tropis atau daerah dengan suhu rendah seperti pegunungan dan dataran tinggi. Apel memiliki banyak jenis tetapi dapat digolongkan sebagai tanaman dari genus *Malus*. Di Indonesia, apel pertama kali didatangkan bangsa Belanda untuk dibudidayakan di dataran tinggi Bromo-Tengger-Semeru. Salah satu jenis apel yang berhasil dibudidaya hingga menjadi komoditas unggulan daerah Malang Raya adalah apel Manalagi (*Malus sylvestris*). Harga apel yang berada di pasaran sangat bergantung dari tingkat kematangan apel saat dipetik. Tingkat kematangan apel dapat dibagi menjadi belum ranum (*unripe*) dan ranum (*ripe*). Tingkat kematangan mempengaruhi kualitas rasa, tekstur, dan visual dari buah apel. Aspek visual yaitu warna dan tekstur dari buah apel berubah sesuai dengan tingkat kematangannya. Keuntungan dari mengkategorikan buah apel sebelum dipanen adalah buah apel yang belum memenuhi kriteria tidak perlu dipetik. Apel yang belum ranum dapat masak di pohon sehingga harga dari buah apel yang diproduksi meningkat. Peningkatan keuntungan dari tidak memanen apel yang belum ranum dengan harga yang rendah bermanfaat bagi industri perkebunan apel secara menyeluruh. Penentuan tingkat kematangan buah apel secara obyektif menggunakan metode yang dapat merusak (dekstruktif) seperti *puncture test* untuk mengukur kekerasan apel, *Soluble Solids Contents (SSC)* untuk mengukur kadar gula, dan *titratable acidity (TA)* untuk mengukur kadar keasaman, dan *starch pattern index (SPI)* untuk menentukan pola pati yang sudah matang (Rutkowski, Michalczuk and Konopacki, 2008). Metode dekstruktif ini tidak cocok untuk diterapkan di lapangan secara langsung. Pada praktiknya penentuan tingkat kematangan buah apel oleh petani dan penjual buah apel dilakukan secara subyektif yaitu dengan mengandalkan pengalaman dalam menganalisa warna, tekstur, dan tingkat kekerasan buah apel. Penilaian secara subyektif ini tidak dapat diandalkan oleh karena itu dilakukan penelitian untuk menganalisa tingkat kematangan apel dari tampak luarnya.

Untuk dapat mengklasifikasi tingkat kematangan apel maka diperlukan metode untuk mengekstrak fitur dari foto apel. Apel yang matang dan belum matang memiliki perbedaan tekstur dan warna. Oleh karena itu digunakan ekstraksi fitur GLCM untuk menganalisa

tekstur dan skala warna CIE Lab untuk menganalisa warna. Teknik GLCM telah banyak digunakan untuk berbagai penelitian seperti identifikasi pola batik (Ayu, 2019), mengklasifikasi kanker payudara (Julio Sarosa, Utamingrum and Bachtiar, 2019), diagnosa penyakit Alzheimer (Gao, 2021), dan diagnosa Covid-19 (Pi and Lima, 2021). Keunggulan menggunakan skala warna CIE Lab untuk analisa warna dibuktikan pada evaluasi kualitas daging ham (Valous *et al.*, 2009), evaluasi kualitas sayur kubis (Pace *et al.*, 2017), dan evaluasi kualitas radicchio (Pace *et al.*, 2015). Memasangkan machine learning dengan analisa tekstur dan warna telah berhasil mengklasifikasi tomat berdasarkan tingkat kematangan (Lavindi *et al.*, 2019) dan memprediksi kualitas apel (Zhang, Zou and Pan, 2020). Penelitian ini diharapkan menjadi dasar untuk dapat memklasifikasi tingkat kematangan apel secara *real-time*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah pada penelitian ini adalah bagaimana membangun metode prediksi kematangan buah apel secara non destruktif melalui ekstraksi fitur GLCM dari citra yang diperoleh, skala warna CIE Lab, dan *machine learning*.

1.3 Tujuan

Tujuan dilakukannya penelitian ini adalah membangun metode prediksi kematangan buah apel secara non destruktif melalui ekstraksi fitur GLCM dari citra yang diperoleh, skala warna CIE Lab, dan *machine learning*.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

- a) Buah apel yang dijadikan objek penelitian adalah apel manalagi atau apel malang (*Malus sylvestris*).
- b) Pengambilan citra menggunakan kamera 12 MP *Xiaomi Redmi 5*.
- c) Penelitian ini menggunakan bahasa pemrograman *Python* dan *library* dari *sklearn*.
- d) Ekstraksi GLCM, PCA, dan *training machine learning* dilakukan dengan menggunakan *virtual machine* dari *Google Colab*.

1.5 Relevansi Penelitian

Penelitian yang diusulkan ini diharapkan memberi manfaat sebagai berikut:

- a) Penelitian ini diharapkan mampu dijadikan awalan penelitian untuk mengotomasi proses memanen apel (*Malus sylvestris*).
- b) Penelitian ini diharapkan dapat dijadikan inspirasi pembaca dalam menggunakan GLCM, CIE Lab, dan *machine learning* untuk mengklasifikasikan obyek.

1.6 Sistematika Laporan

Sistematika laporan diawali dengan BAB I Pendahuluan. Selanjutnya BAB II Tinjauan Pustaka dan Dasar Teori terdiri dari *review* penelitian sebelumnya dan penjelasan teori yang digunakan dalam penelitian ini. BAB III Metode penelitian dibagi menjadi 10 bagian sesuai dengan *flowchart*. BAB IV Analisa Data dan Pembahasan yang terdiri dari tiga analisa yaitu analisa tingkat kematangan, analisa GLCM, dan analisa *machine learning*. Terakhir adalah pemaparan kesimpulan dan saran pada BAB V.

Halaman ini sengaja dikosongkan

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Review Penelitian Sebelumnya

Penelitian yang menggunakan metode K-means clustering, starch conversion rate, CIE Lab, GLCM, dan machine learning telah banyak dilakukan sebelumnya seperti ditampilkan pada Tabel 2.1 sebagai literatur pembandingan.

Tabel 2.1 Penelitian penunjang

No	Profil Pustaka	Metode dan Temuan
1	<p>Judul: <i>Determination of bagged ‘Fuji’ apple maturity by visible and near-infrared spectroscopy combined with a machine learning algorithm</i></p> <p>Penulis: Mengsheng Zhang, Bo Zhang, Hao Li, Maosheng Shen, Shijie Tian, Haihui Zhang, Xiaolin Ren, Libo Xing, Juan Zhao.</p> <p>Jurnal/Prosiding: Infrared Physics & Technology, 2020, vol. 111, hal. 1-10.</p>	<p>Metode: Penelitian ini menganalisa bagaimana <i>Vis-NIR spectroscopy</i> dan <i>machine learning</i> dapat menentukan kematangan apel ‘Fuji’ dari <i>starch index</i>. Dari <i>starch index</i>, apel dikategorikan menjadi 3 level, belum matang, matang panen, matang. Selain itu tingkat kematangan apel juga ditentukan menggunakan metode tradisional seperti <i>puncture test</i> untuk mengukur kekerasan apel, <i>Soluble Solids Contents/SSC</i> untuk mengukur kadar gula, dan <i>titratable acidity/TA</i> untuk mengukur kadar keasaman. Algoritma machine learning yang digunakan adalah <i>least squares support vector machine</i>, <i>probabilistic neural network</i>, <i>extreme learning machine</i>, <i>partial least squares discrimination analysis</i>, dan <i>linear discriminant analysis (LDA)</i>.</p> <p>Temuan: Dengan membandingkan hasil dari berbagai metode, didapatkan prediksi dari model RF-SPA-LSSVM berbasis 15 panjang gelombang karakteristik adalah yang terbaik dengan akurasi 89%. Hasil ini menunjukkan bahwa tingkat kematangan apel dapat secara akurat diketahui dengan <i>Vis-Nir Spectroscopy</i> dan <i>machine learning</i> (Zhang <i>et al.</i>, 2020).</p>

Tabel 2.1 (Lanjutan bagian ke-1)

No	Profil Pustaka	Metode dan Temuan
2	<p>Judul: <i>Neural Network based on GLCM, and CIE L*a*b* Color Space to Classify Tomatoes Maturity</i></p> <p>Penulis: Eri Eli Lavindi, Edi Jaya Kusuma, Guruh Fajar Shidik, Ricardus Anggi Pramunendar, Ahmad Zainul Fanani, Pujiono</p> <p>Jurnal/Prosiding: 2019 International Seminar on Application for Technology of Information and Communication (iSemantic), hal. 48-50.</p>	<p>Metode: Pada penelitian ini dilakukan klasifikasi kematangan tomat menggunakan <i>Gray Level Co-occurrence Matrix</i> (GLCM) dan skala warna CIE Lab. Fitur yang diekstrak dari GLCM adalah kontras, korelasi, energi, homogeneitas, dan entropi. Klasifikasi 5 kelas dilakukan dengan menggunakan tiga skenario <i>Backpropagation Neural Network</i> (BPNN). Tiap skenario juga menggunakan 3 jarak GLCM dan 5 <i>learning rate</i> BPNN yang berbeda.</p> <p>Temuan: Metode terbaik untuk mengklasifikasi tomat berdasarkan tingkat kematangannya adalah menggunakan GLCM dan tiga hidden layer BPNN yang terdiri dari 21, 5, dan 3 nodes. Akurasi mencapai 95,5% (Lavindi <i>et al.</i>, 2019).</p>
3	<p>Judul: <i>Evaluation of the Ripening Stages of Apple (Golden Delicious) by Means of Computer Vision System</i></p> <p>Penulis: Stefany Cardenas-Perez, Jorge Chanona-Perez, Juan V. Mendez-Mendez, Georgina Calderon-Dominguez, Ruben Lopez-Santiago, Mari'a J. Perea-Flores, Israel Arzate-Vazquez.</p> <p>Jurnal/Prosiding: Biosystems Engineering, 2017, vol. 159, hal. 46-58.</p>	<p>Metode: Penelitian ini menggunakan <i>computer vision system</i> untuk mengevaluasi tahapan kematangan apel. Analisa keasaman (<i>titratable acidity/TA</i>) dilakukan untuk mendapatkan data pembanding. Nilai yang dapat diekstrak dari tiap gambar pada skala warna CIE Lab adalah L*, a*, b*, ΔE, C* (chroma), dan h* (hue angle). Korelasi dari tiap variabel dianalisa menggunakan <i>Principal Component Analysis</i> (PCA). Validasi metode dilakukan dengan <i>Multivariate discriminant analysis</i> (MDA).</p> <p>Temuan: Hasil dari TA berkorelasi dengan parameter warna kecuali L* dengan reliabilitas 91%. Analisa dengan 4 parameter yaitu a*, b*, C* dan h* menghasilkan klasifikasi dengan akurasi 100% (Cárdenas-Pérez <i>et al.</i>, 2017).</p>

Tabel 2.1 (Lanjutan bagian ke-2)

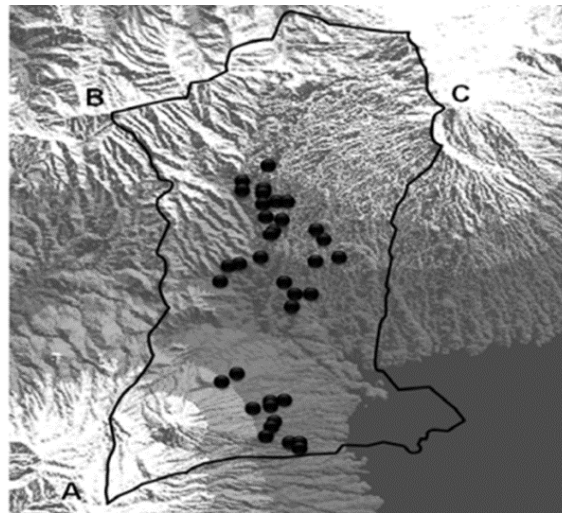
No	Profil Pustaka	Metode dan Temuan
4	<p>Judul: <i>Non-destructive and Contactless Quality Evaluation of Table Grapes by a Computer Vision System</i></p> <p>Penulis: Dario Pietro Cavallo, Maria Cefola, Bernardo Pace, Antonio Francesco Logrieco, Giovanni Attolico</p> <p>Jurnal/Prosiding: Computers and Electronics in Agriculture, 2019, vol. 156, hal. 558-556.</p>	<p>Metode: Penelitian ini mengevaluasi anggur jenis Victoria dan jenis Italia dengan menggunakan computer vision system. Anggur dikategorikan berdasarkan kualitasnya oleh 8 panelis menjadi 5 kelas, 3 kelas, dan 2 kelas. Gambar dikonversi ke skala warna CIE Lab dan ekstraksi fitur menggunakan k-means clustering. Klasifikasi dilakukan dengan menggunakan <i>random forest model</i>.</p> <p>Temuan: Hasil yang didapatkan adalah <i>cross-validation classification accuracy</i> mencapai 92% pada jenis Victoria dan mencapai 100% pada jenis Italia (Cavallo <i>et al.</i>, 2019).</p>
5	<p>Judul: <i>Uncertainty Analysis and Modelling of the Starch Index During Apple Fruit Maturation</i></p> <p>Penulis: Ann Peirs, Nico Scheerlinck, Amalia Berna Perez, Pa'1 Jancso'k, Bart M. Nicolai.</p> <p>Jurnal/Prosiding: Postharvest Biology and Technology, 2002, vol. 26, hal. 199-207.</p>	<p>Metode: Penelitian ini menganalisa ketidakpastian dari starch index dalam menentukan kematangan apel. Apel diuji dengan <i>iodine test</i> dan dievaluasi oleh inspektur secara manual sesuai standar. Persentase <i>starch index</i> yang dihitung dibandingkan dengan skor dari inspektur.</p> <p>Temuan: Hasil yang diperoleh adalah evaluasi <i>starch index</i> oleh inspektur menghasilkan error hingga 60%. Dengan menghitung <i>starch index</i> maka dapat meningkatkan prediksi dalam menentukan waktu panen (Peirs <i>et al.</i>, 2002).</p>

2.2 Dasar Teori

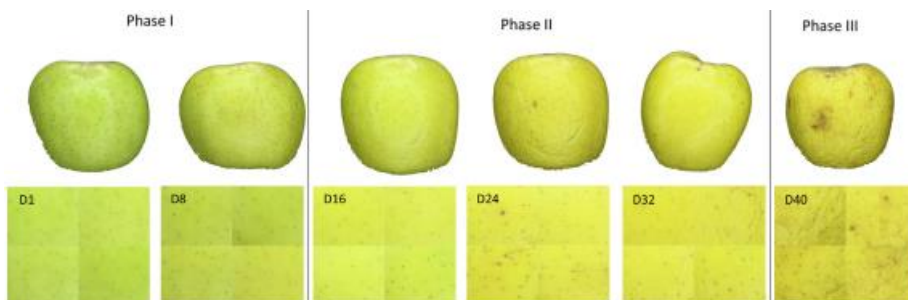
2.2.1 Apel

Banyak perdebatan bagaimana apel pertama kali dibudidayakan di Indonesia, apakah dibawa dari bangsa Belanda pada zaman penjajahan atau didatangkan setelah kemerdekaan. Berdasarkan Pak Widodo yang dikutip dari paper (Makmur Sitompul and Sugito, 2013) bibit

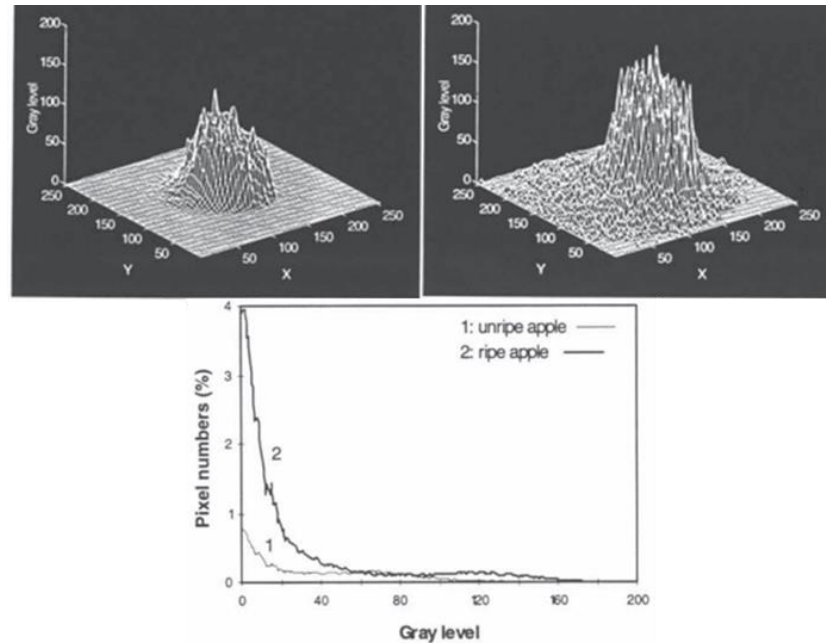
apel jenis manalagi (*Malus sylvestris*) didatangkan dari Australia pada 1950 untuk dibudidayakan di Cipanas, Jawa Barat. Kemudian pada 1956, Widodo sebagai kepala *Indonesian Center for Horticulture Research and Development* (ICHORD) di Malang mulai mencoba budidaya apel di Tlekung, Batu. Banyak tempat yang dicoba untuk menumbuhkan apel di Indonesia akan tetapi penelitian yang sukses untuk membudidayakan apel di Indonesia pertama kali dilaporkan pada 1963. Salah satu kunci sukses dalam pembungaan apel di Malang adalah *defoliation* (secara selektif memilih daun untuk dilepas dari pohon). Apel yang dapat dimakan mulai fase belum ranum memiliki warna yang berbeda dengan apel yang ranum maupun apel tua. Perbedaan warna dapat dikenali kamera yang menangkap gambar dalam RGB. Pada ruang grayscale, perbedaan warna mengakibatkan perbedaan pada *grey level* dari piksel gambar dapat dilihat dalam Gambar 2.3.



Gambar 2.1 Titik hitam adalah kebun apel berada di daerah Batu yang dikelilingi 3 Gunung (A=G. Kawi, B=G. Anjasmoro, C=G. Arjuna) (Makmur Sitompul and Sugito, 2013)



Gambar 2.2 Perbedaan warna dari tingkat kematangan apel (*Malus domestica*), fase 1: belum ranum, fase 2: ranum, fase 3: tua (Cárdenas-Pérez *et al.*, 2017)



Gambar 2.3 Gray level dari apel belum ranum (Kiri), apel ranum (Kanan), dan grafik gray level dibanding jumlah pixel (Duprat *et al.*, 1995).

2.2.2 Computer Vision

Sistem yang didesain untuk memberikan informasi secara visual untuk diolah komputer dapat dikategorikan sebagai *computer vision system* (CVS). Pada dasarnya penglihatan (*vision*) menggunakan informasi spasial dan temporal yang berasal dari pantulan cahaya dari sekitar. Fokus dari computer vision ada pada aspek komputasinya, menanyakan bagaimana informasi visual dapat ditunjukkan untuk dikenali dan dimanipulasi. Pemrosesan informasi visual memiliki dua sumber utama. Yang pertama adalah bagaimana mendesain model dari proses melihat secara biologis. Yang kedua adalah bagaimana mendesain mesin yang dapat memproduksi hasil yang diinginkan dari informasi visual (Ikeuchi, 2021).

Komponen dasar CVS meliputi kamera, *interface*, dan komputer. Kamera bertindak sebagai sensor yang mendeteksi cahaya dan mengubah muatan pada hole/elektron. Pada dasarnya terdapat 3 tipe kamera vidicons, *charge coupled devices* (CCD), dan *complementary metal oxide silicon* (CMOS). Vidicons atau *video camera tube* adalah kamera analog yang menggunakan tabung katoda. Vidicons kemudian diganti dengan teknologi digital CCD dan CMOS. Interface adalah program yang dijalankan untuk memproses gambar yang ditangkap oleh kamera (*image processing*). Proses yang dilakukan antara lain *object detection*, *object segmentation*, *feature extraction*, *classification*, dan lain-

lain. *Hardware* seperti komputer dan sejenisnya menyesuaikan dengan fungsi yang ingin dijalankan.

2.2.3 Skala Warna

Skala warna/*color space* mendeskripsikan jangkauan warna. Skala warna secara umum dapat dipisahkan menjadi dua, untuk keperluan analisa dan untuk keperluan editing. Skala warna yang menjadi pusat skala adalah CIE XYZ. CIE adalah kependekan dari the *International Commission on Illumination (Commission Internationale de L' Eclairage)* organisasi yang bertanggungjawab untuk standarisasi data iluminasi. Persamaan skala warna CIE XYZ didasarkan pada skala RGB yang juga berdasarkan LMS. Skala warna LMS adalah skala warna yang merepresentasikan tiga tipe kerucut mata manusia yang memiliki sensitivitas pada panjang gelombang long, medium, dan short.

CIEXYZ tidak mempunyai perceptual yang dapat berguna dalam keperluan praktik. Dalam usaha menambahkan perceptual behaviour ke skala warna, CIE membuat skala CIELAB dimana L^* (*lightness*), a^* (*redness–greenness*), and b^* (*yellowness–blueness*). Model seperti sangat mirip dengan bagaimana mata manusia mempersepsi warna, yaitu dengan melihat perbedaan warna dan jarak diantara warna. Komputasi dari CIEXYZ dilakukan dengan menormalkan ke titik netral (putih). Dengan menggunakan asumsi sumber cahaya D maka warna tertentu dapat dipersepsi sama dengan putih ($a^*=b^*=0$). Terdapat banyak spesifikasi sumber cahaya D namun pada umumnya digunakan D65 yang merepresentasikan pencahayaan matahari siang hari dengan temperatur warna 6500 K.

Konversi RGB ke CIE XYZ menggunakan persamaan berikut:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.1)$$

Konversi CIE XYZ ke CIE LAB menggunakan persamaan berikut:

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16 \quad (2.2)$$

$$a^* = 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad (2.3)$$

$$b^* = 200 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad (2.4)$$

dimana,

$$f(t) = t^{\frac{1}{3}} , \quad \text{untuk } t > 0.008856 \quad (2.5)$$

$$f(t) = 7.787 t + \frac{16}{116}, \quad \text{untuk } t \geq 0.008856 \quad (2.6)$$

untuk standard illuminant D65:

$$X_n = 96.4212$$

$$Y_n = 100$$

$$Z_n = 82.5188$$

Dengan menggunakan koordinat CIELAB yang berada di ruang tiga dimensi maka *Euclidean difference*, *chroma*, dan *hue* dapat dihitung sebagai berikut:

$$\Delta E_{ab}^* = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{\frac{1}{2}} \quad (2.7)$$

$$C_{ab}^* = (a^{*2} + b^{*2})^{\frac{1}{2}} \quad (2.8)$$

$$h_{ab}^* = \tan^{-1} \left(\frac{b^*}{a^*} \right) \quad (2.9)$$

2.2.4 K-means Clustering

K-means clustering merupakan salah satu metode yang digunakan untuk mensegmentasi gambar. Image segmentation adalah proses membagi gambar digital menjadi beberapa bagian berbeda yang terdiri dari kumpulan piksel dengan ciri yang serupa. Tujuan segmentasi adalah untuk mengubah representasi dari gambar menjadi sesuatu yang lebih mudah dianalisa oleh komputer. Teknik umum yang digunakan untuk segmentasi adalah dengan melihat nilai piksel yang tidak berkelanjutan atau dengan kata lain mendeteksi sudut (*edge*) yang mendefinisikan sebuah daerah. Teknik lain adalah dengan mendeteksi persamaan antar bagian dari gambar seperti *clustering* dan *thresholding*. Pada *clustering*, tujuan yang ingin dicapai belum diketahui berbeda dengan klasifikasi dimana terdapat tujuan akhir.

K-means clustering adalah *clustering* yang membagi daerah berdasarkan sentroidnya. Tujuan dari *k-means clustering* adalah mengurangi jarak sentroid dengan setiap data atau piksel. Proses yang dilakukan adalah pertama-tama menentukan jumlah kluster yang diinginkan. Kemudian meletakkan sentroid secara acak dan mengalokasi tiap piksel ke sentroid terdekat sehingga didapatkan *K-cluster*. Langkah ini diiterasi hingga mendapatkan J terbaik. Contoh *clustering* ada pada Gambar 2.4.



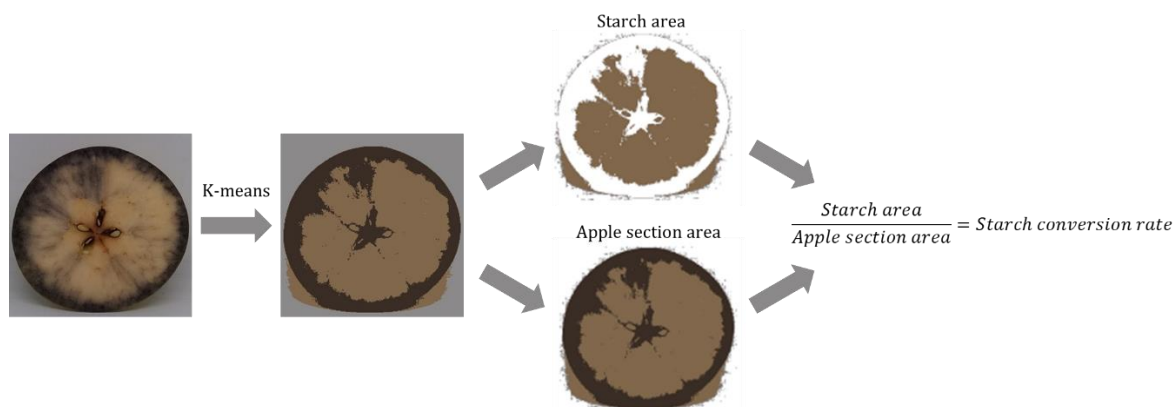
Gambar 2.4 Contoh k-means clustering (Nagesh Singh Chauhan, 2019)

Formula untuk menentukan fungsi obyektif k-means:

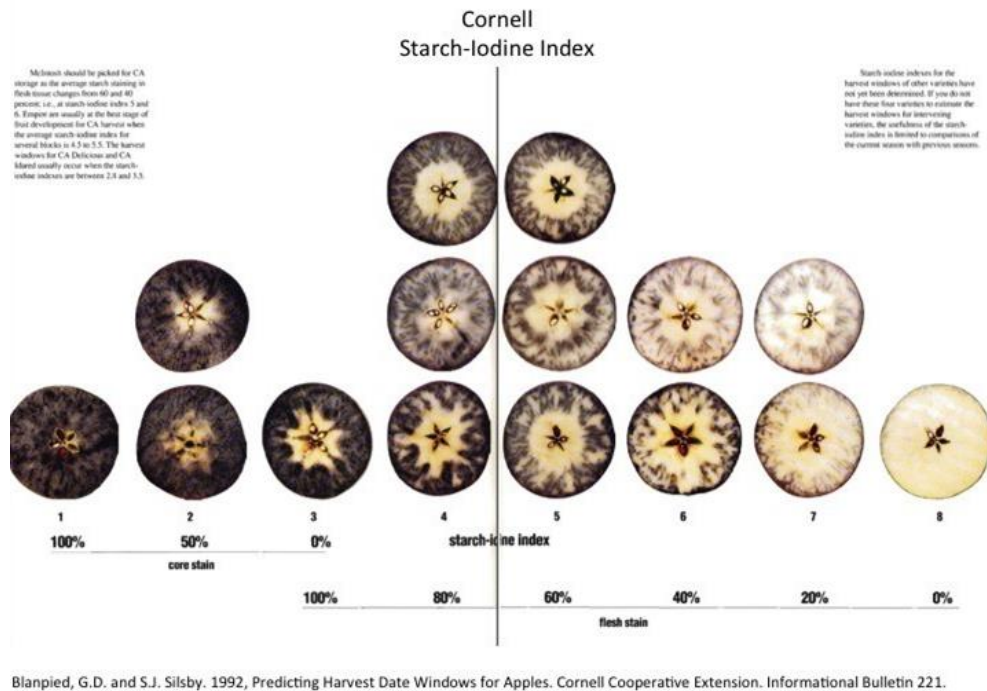
$$J = \sum_{j=1}^k \sum_{i=1}^n |x_i^{(j)} - c_j|^2 \quad (2.10)$$

2.2.5 Starch Conversion Rate

Starch Conversion Rate atau SCR adalah nilai seberapa besar daging buah apel yang sudah menjadi pati (*starch*). Semakin banyak pati maka apel menjadi semakin matang. Pada penelitian ini buah apel yang sudah dicelupkan ke larutan *iodine* akan mengalami proses perubahan kimiawi yang menyebabkan pati menjadi berwarna coklat muda dan daerah yang belum matang berwarna hitam. Proses pematangan pada buah apel terjadi secara homogen dari tengah buah apel. Hal ini dikarenakan nutrisi dari pohon apel masuk ke buah apel melalui tangkai yang tersambung dengan inti buah yang biasanya terdapat biji buah apel. SCR ini dilakukan berdasarkan indeks pada metode untuk memprediksi waktu panen apel (Blanpied and Silsby, 1992) yang terlampir pada Gambar 2.6.



Gambar 2.5 Proses mendapatkan SCR



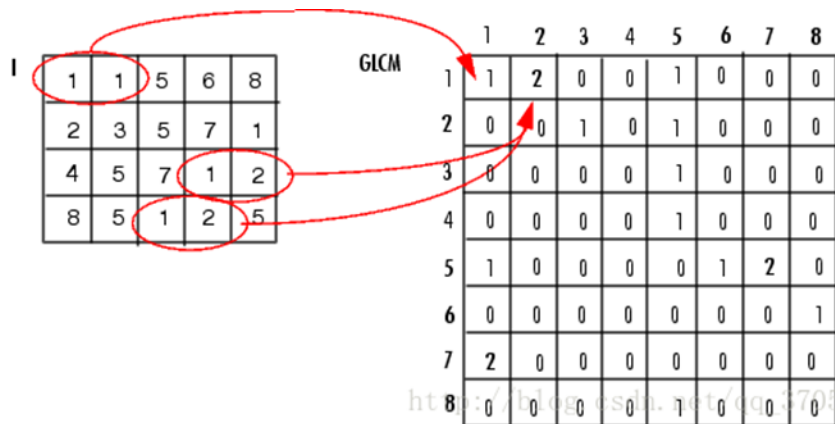
Gambar 2.6 Starch-Iodine Index (Blanpied and Silsby, 1992)

2.2.6 Gray-Level Co-occurrence Matrix (GLCM)

Grey Level Co-occurrence Matrix adalah metode ekstraksi tekstur berdasarkan statistik. GLCM didefinisikan sebagai probabilitas antar nilai piksel i berdampingan dengan nilai piksel j . Nilai probabilitas ini dapat diekspresikan dalam matriks. GLCM dipopulerkan oleh melalui penelitian untuk mengklasifikasi obyek berdasarkan tekstur (Haralick, Shanmugam and Dinstein, 1973). Digunakan dengan premis bahwa distribusi spasial dari piksel dalam gambar mengandung informasi tekstur. Haralick et. al. mengajukan 14 jenis statistik berdasarkan *gray level co-occurrence matrix*, yaitu: *energy, entropy, contrast, uniformity, correlation, variance, sum average, sum variance, sum entropy, difference variance, difference average, difference entropy, related information measurement, and maximum correlation coefficient*.

Secara sederhana GLCM dapat diselesaikan seperti Gambar 2.7. Contohnya titik (1,1) memiliki nilai 1 bila dipasangkan dengan piksel (1,2) bernilai 1 yang berdekatan secara horizontal maka kemungkinan $GLCM(1,1)$ untuk seluruh matriks ada 1. Demikian juga untuk $GLCM(1,2)$ yang memiliki dua kemungkinan pada seluruh matriks (matlab documentation). Pada contoh $GLCM(i,j)$ diatas digunakan $d=1$ dan sudut 0° maka piksel yang berdekatan adalah piksel yang bersebelahan secara horizontal. Bila menggunakan jarak

d dan sudut yang lain maka bisa didapatkan matriks GLCM yang berbeda. Untuk mengekstraksi fitur maka digunakan statistik berdasarkan matriks GLCM(i,j).



Gambar 2.7 Proses mengubah matriks ke matriks GLCM (Haralick, Shanmugam and Dinstein, 1973)

Fitur yang dapat diekstrak dari matriks GLCM ini dibagi menjadi kontras, energi, dan deskriptif. Kelompok kontras termasuk dissimilarity dan homogeneity. Pada kelompok energi terdapat energi dan ASM. Rata-rata, varians, dan korelasi termasuk kelompok deskriptif. Pada persamaan dibawah ini adalah nilai baris (*row*) dan j adalah nilai kolom pada matrix. P adalah probabilitas antara pasangan i dan j .

- Kontras

Kontras merepresentasikan perbedaan dalam nilai antar piksel. Bila variasi nilai antar piksel konstan maka nilai kontras adalah nol.

$$Contrast = \sum_i \sum_j (i - j)^2 P_{i,j} \quad (2.11)$$

- Dissimilarity

Berbeda dengan kontras yang perbedaan nilai baris dan kolom bertambah secara eksponensial, dissimilarity bertambah secara linear.

$$Dissimilarity = \sum_i \sum_j P_{i,j} |i - j| \quad (2.12)$$

- Homogeneity

Kontras dan dissimilarity menghasilkan angka yang lebih besar saat perhitungan menjauh dari diagonal. Akan tetapi nilai homogeneity berbanding terbalik dengan kontras dimana menghasilkan angka yang semakin kecil saat menjauh dari diagonal.

$$Homogeneity = \sum_i \sum_j \frac{P_{i,j}}{1 + (i - j)^2} \quad (2.13)$$

- ASM / angular second moment

Pada mekanika klasik, angular second moment atau momen inersia menghitung seberapa besar torsi yang dibutuhkan untuk merotasi percepatan sudut tertentu. Sama seperti mekanika, pada matriks seberapa besar kemungkinan tiap pasangan piksel i, j dinamakan ASM.

$$\tau = \frac{I}{\alpha} = m r^2 \quad (2.14)$$

$$ASM = \sum_i \sum_j P_{i,j}^2 \quad (2.15)$$

- Energi

Energi merepresentasikan seberapa besar kemiripan nilai piksel yang berdekatan. Semakin mirip nilai piksel maka energinya semakin besar.

$$Energy = \sqrt{ASM} \quad (2.16)$$

2.2.7 Seleksi Fitur

Ekstraksi fitur dari gambar seperti GLCM sering kali menghasilkan fitur-fitur yang pada jarak dan sudut tertentu tidak dapat membedakan tekstur yang dimiliki oleh gambar dengan kategori yang berbeda. Oleh karena itu dengan menggunakan metode seleksi fitur, dapat diketahui hasil ekstraksi mana saja yang dapat dibuang atau tidak digunakan untuk proses selanjutnya yaitu *machine learning*. Selain untuk mengurangi fitur yang dianggap tidak penting atau tidak memiliki informasi yang berguna (*redundant*) dalam proses klasifikasi, penggunaan seleksi fitur dapat mengurangi banyaknya waktu yang dibutuhkan untuk melakukan *training* data.

- *Standard Error of difference between means* (SEd)

Salah satu metode yang dapat digunakan untuk menyeleksi fitur adalah *standard error of difference between means* atau SEd. Metode ini dapat mengetahui bagaimana perbedaan antar fitur pada dua kelas yang independen dan fitur mana yang dapat berkontribusi untuk membedakan kelas yang ada. SEd dapat dites dengan menggunakan hipotesa bila nilai Z yang dihasilkan lebih besar dari *critical value* Z pada *confidence level* $1-\alpha$ tertentu. Nilai SEd dan Z dapat dihitung dengan formula dibawah.

$$SE_D = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \quad (2.17)$$

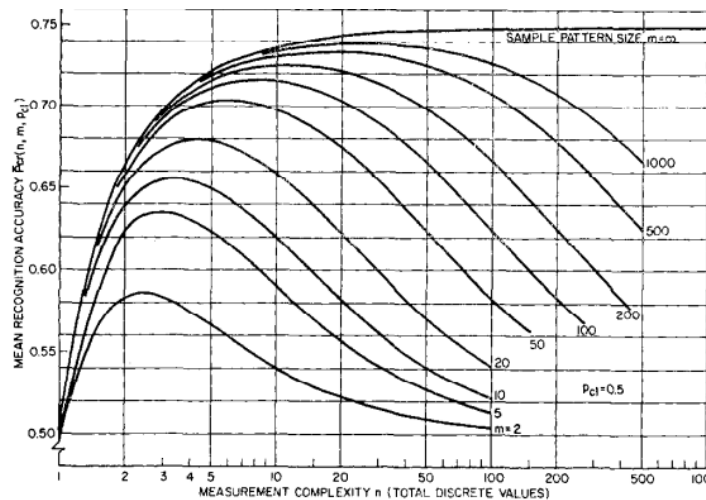
$$Z = \frac{D}{SE_D} = \frac{|M_1 - M_2|}{SE_D} \quad (2.18)$$

dengan, $\sigma = \text{standard deviation}$

M = means / rerata

- *Principal Component Analysis (PCA)*

Berbeda dengan SED, PCA merupakan salah satu algoritma yang digunakan untuk mengurangi dimensionalitas data. PCA tidak menyeleksi secara langsung akan tetapi mengubah fitur menjadi *principal components*. Menurut *Hughes phenomenon* (Hughes, 1968), bila jumlah sampel tetap dan jumlah dimensi terus bertambah maka kekuatan prediksi dari model *machine learning* akan bertambah hingga titik tertentu sebelum berkurang. Sebagai contoh, dengan probabilitas sampel (m) sebanyak 2 maka dimensi fitur (n) yang memiliki akurasi terbaik (sumbu y) adalah dua seperti pada Gambar 2.8. Dengan begitu, untuk mengurangi dimensi data maka digunakan teknik PCA. PCA dapat digunakan untuk variabel yang berkorelasi secara linear.



Gambar 2.8 Hughes Phenomenon

PCA adalah teknik yang mengkonversi fitur yang berkorelasi kedalam rangkaian fitur yang tidak berkorelasi pada dimensi yang lebih kecil dengan mentransformasi secara orthogonal. Fitur yang tidak berkorelasi ini disebut *principal components*. PCA dapat digunakan untuk seleksi fitur karena dengan mengubah fitur menjadi *principal component* dan memilih *principal component* teratas yang memiliki informasi terbesar dari data dilihat dari *variance* maka dapat mengelompokkan data tanpa *principal components* lain yang tidak dipilih (tidak memiliki banyak kontribusi untuk membedakan antar kelompok).

Principal components diberikan peringkat berdasarkan *variance* atau seberapa besar kontribusi tiap fitur untuk mengelompokkan data. Langkah-langkah yang dilakukan pada PCA dimulai dengan menstandarisasi data. Standarisasi dilakukan dengan mengurangi nilai x dengan rerata (*mean*) dan dibagi dengan standar deviasi.

$$x_{new} = \frac{x - \bar{x}}{\sigma} \quad (2.19)$$

Selanjutnya adalah menghitung kovarian (*covariance*) matriks. Rerata dari matriks setelah distandarisasi adalah 0 dan standar deviasinya 1. *Covariance* dari x dan y merupakan total dari x dikurangi dengan rerata x dikali dengan y dikurangi rerata y dan dibagi dengan jumlah sampel.

$$Cov(x, y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N} \quad (2.20)$$

Dari matriks kovarian, dihitung *eigenvalues*. Determinan dari matriks sama dengan 0 sehingga dapat dicari *eigenvalue*-nya.

$$A v - \lambda v = 0 \quad (2.21)$$

$$(A - \lambda I)v = 0 \quad (2.22)$$

$$\det(A - \lambda I) = 0 \quad (2.23)$$

dimana, A = matriks

λ = eigenvalue

v = eigenvector

I = matriks identitas

Dengan menggunakan *Cramer's rule* dari hasil determinan. Maka dapat diketahui nilai dari *eigenvectors*. Nilai *eigenvectors* dipilih sesuai dengan berapa *principal component* yang diinginkan kemudian dilakukan perkalian matriks awal dengan *eigenvectors*. Hasil akhir dari PCA adalah matriks hasil perkalian dengan *eigenvectors*.

2.2.8 Machine Learning

Machine learning adalah jenis algoritma yang memungkinkan komputer belajar tanpa perlu diprogram secara eksplisit. *Machine learning* memiliki banyak fungsi salah satunya adalah untuk memprediksi. Model *machine learning* untuk prediksi dapat dibagi menjadi dua yaitu untuk memprediksi kelas atau klasifikasi dan memprediksi angka atau

regresi. Model *machine learning* untuk prediksi kelas menghasilkan *confusion matrix*. Untuk mengetahui performa *machine learning* maka dilakukan analisa pada *confusion matrix* dengan berbagai parameter yaitu *accuracy*, *precision*, *sensitivity*, *specificity*, dan *f1-score*.

- **Confusion matrix**

Matriks yang dihasilkan dari training machine learning ini adalah matriks yang menunjukkan berapa jumlah data yang pada saat testing benar atau salah. Data yang sebenarnya termasuk kelas 0 akan tetapi pada saat prediksi termasuk kelas 1 akan termasuk *false positive* (FP). Begitu juga dengan sel-sel matriks yang lain, data yang termasuk kelas 0 dan benar diprediksi adalah *true negative* (TN).

		Predicted	
		0	1
Actual	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Gambar 2.9 Confusion matrix

- **Accuracy**

Akurasi adalah rasio yang menunjukkan banyaknya data yang benar diprediksi dibanding dengan total data.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.24)$$

- **Precision**

Presisi adalah persentase dari data yang benar diprediksi positif dibagi dengan total data yang diprediksi positif.

$$Precision = \frac{TP}{TP + FP} \quad (2.25)$$

- **Sensitivity**

Sensitivitas atau sering juga disebut *recall* adalah persentase dari data yang benar diprediksi positif dibagi dengan total data yang sebenarnya positif.

$$Sensitivity = Recall = \frac{TP}{TP + FN} \quad (2.26)$$

- **Spesificity**

Spesifisitas merupakan kebalikan dari sensitivitas. Spesifisitas adalah persentase dari data yang benar diprediksi negatif dibagi dengan total data yang sebenarnya negatif.

$$Spesificity = \frac{TN}{TN + FP} \quad (2.27)$$

- *F1-score*

Nilai f1 adalah rata-rata harmonik dari presisi dan sensitivitas. Nilai ini menunjukkan seberapa model yang dapat memprediksi benar. Keunggulan dari nilai f1 ini dibanding akurasi adalah f1 tidak dipengaruhi jumlah data tiap kelas sehingga meskipun jumlah data dari kelas 0 dan kelas 1 rasionya tidak 1:1, nilai f1 dapat merepresentasikan seberapa bagus model *machine learning* dengan lebih baik dibandingkan akurasi.

$$F1\ Score = \left(\frac{\frac{1}{Precision} + \frac{1}{Recall}}{2} \right)^{-1} = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2.28)$$

Pada dasarnya dari sekian banyak algoritma *machine learning* yang digunakan saat ini, algoritma dapat dikategorikan berdasarkan bagaimana model *machine learning* bekerja. Secara umum penggunaan algoritma *machine learning* pada bidang pertanian dapat disederhanakan seperti dikutip dari penelitian (Benos *et al.*, 2021) pada **Tabel 2.2**.

Tabel 2.2 Kelompok algoritma machine learning

Singkatan	Model	Contoh Algoritma
ANN	Artificial Neural Network	Backpropagation Neural Network, Extreme Learning Machine
BM	Bayesian Models	Gaussian Naïve Bayes, Multinomial Naïve Bayes
DL	Deep Learning	Convolutional Neural Network, Deep Belief Network
DR	Dimensionality Reduction	Quadratic Discriminant Analysis, Linear Discriminant Analysis
DT	Decision Tree	Classification Tree, Regression Tree
EL	Ensemble Learning	Random Forest, AdaBoost
IBM	Instance Based Model	K-Nearest Neighbor, Weighted Nearest Neighbors
SVM	Support Vector Machine	RBF SVM, Support Vector Regression

- *Artificial Neural Network (ANN)*

ANN terinspirasi dari cara otak manusia bekerja yang dapat melakukan fungsi kompleks seperti *pattern generation*, pembelajaran, dan pengambilan keputusan. Otak manusia terdiri dari miliaran sel saraf yang saling terkait dan berkomunikasi satu sama lain. ANN merupakan model sederhana dari struktur biologis jaringan sel saraf. Sel saraf pada ANN dibagi menjadi tiga *layer*, yang pertama adalah *input layer* (*layer* yang memasukkan

data), *hidden layer* (layer yang melakukan pembelajaran), dan *output layer* (layer yang memberi prediksi).

- *Bayesian Models (BM)*

BM adalah model yang melakukan analisa dengan metode statistik *Bayesian interference*. Model ini menyelesaikan masalah yang termasuk *supervised learning* dimana pembelajaran menggunakan data yang sudah memiliki label.

- *Deep Learning (DL)*

DL merupakan cabang dari machine learning yang memungkinkan model mengkomputasi banyak layer pada data yang kompleks. Salah satu keuntungan DL adalah ekstraksi fitur dilakukan oleh model machine learning sendiri sehingga tidak perlu menggunakan ekstraksi fitur dari yang lain.

- *Dimensionality Reduction (DR)*

DR adalah model yang bertujuan untuk mengurangi dimensi dari data. Dengan kata lain, fitur dari data dikurangi untuk menghasilkan model yang lebih baik dalam memprediksi data.

- *Decision Tree (DT)*

DT adalah model yang melakukan perhitungan dalam arsitektur yang berbentuk seperti pohon. Pada DT, data dibagi menjadi dua cabang apakah memenuhi syarat dibagi atau tidak. Pembagian ini dilakukan terus menerus hingga model pohon tercipta yang dapat memprediksi data.

- *Ensemble Learning (EL)*

EL adalah model yang bertujuan untuk meningkatkan performa prediksi dengan mengkonstruksi kombinasi dari beberapa algoritma *machine learning*. Algoritma machine learning yang biasa digunakan adalah *Decision Tree*.

- *Instance Based Model (IBM)*

IBM adalah model yang belajar dengan menyimpan data dalam memori dan membandingkan dengan data baru. Kelemahan dari model ini adalah seberapa kompleks model seiring dengan jumlah data yang digunakan.

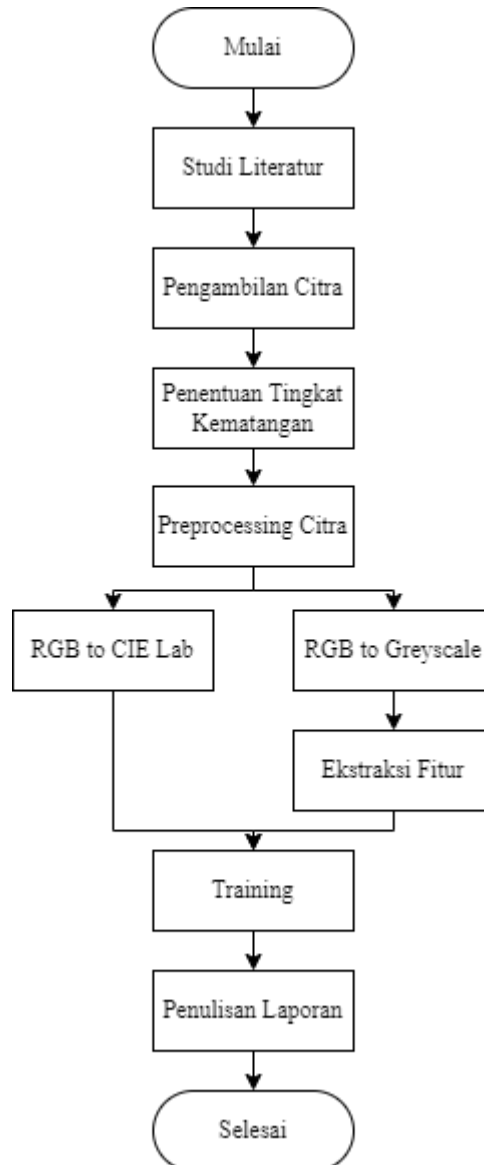
- *Support Vector Machine (SVM)*

SVM adalah model yang menggunakan hyperplane untuk mengklasifikasi data. Kemampuan klasifikasi SVM dapat ditingkatkan dengan mentransformasi data ke dimensi yang lebih tinggi dengan kernel.

BAB III METODOLOGI PENELITIAN

3.1 *Flowchart* Penelitian

Metode penelitian tugas akhir ini dilakukan berdasarkan Gambar 3.1 Diagram Alir Metodologi Penelitian



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.2 Studi Literatur

Studi literatur adalah kegiatan yang dilakukan sebagai penunjang keberhasilan penelitian yang akan dilakukan. Literatur yang dipelajari dan dijadikan bahan rujukan antara

lain buku, jurnal, artikel ilmiah, dan penelitian sebelumnya yang membahas mengenai apel, metode GLCM, skala warna CIE Lab, dan *machine learning* yang menggunakan fitur dari GLCM dan CIE Lab.

3.3 Pengambilan Citra

Citra diambil dengan memfoto tampak luar apel dari 4 sisi. Apel dirotasi dan diambil citra pada sisi depan (0°), kanan (90°), belakang (180°), dan kiri (270°). Pengambilan citra dilakukan pada malam hari dan menggunakan kotak foto untuk memberikan pencahayaan yang konstan selama pengambilan citra. Perangkat yang digunakan adalah *smartphone* Xiaomi Redmi 5 dengan spesifikasi kamera: 12 MP, $f/2.2$, $1.25 \mu\text{m}$, PDAF, dan fitur LED *flash*. Tiap citra diambil menggunakan aplikasi *Open Camera App* pada jarak 30 cm, untuk detail lain dapat dilihat pada Tabel 3.1.



Gambar 3.2 Setup pengambilan data



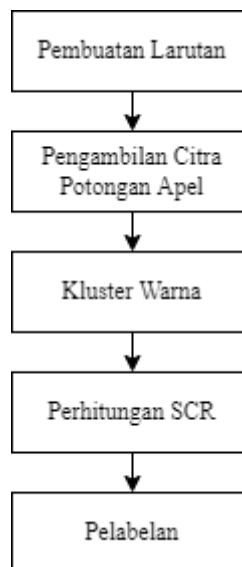
Gambar 3.3 Gambar original yang diambil dari kamera

Tabel 3.1 Detail pengambilan data

Detail	Nilai
Rasio	4:3
Dimensions	4000x3000 (12 MP)
Size	~5MB
Exposure, White Balance	Locked
Focus	Infinity
ISO	100
Zoom	No (Auto)
Flash	No
Exposure compensation	+0.0 EV
Photo mode	Standard

3.4 Penentuan Tingkat Kematangan

Penentuan tingkat kematangan apel dilakukan dengan mengevaluasi *starch conversion rate* dari tiap apel. Proses penentuan tingkat kematangan buah apel dapat dilihat pada Gambar 3.4.



Gambar 3.4 Proses penentuan tingkat kematangan apel

3.4.1 Pembuatan Larutan

Larutan dibuat berdasarkan resep dari (Blanpied and Silsby, 1992). Langkah yang dilakukan adalah sebagai berikut:

1. 8,8 gram Kalium Iodida (KI) di-*dissolve* ke 30 ml air hangat. Diaduk pelan hingga KI larut.



Gambar 3.5 Iodine, KI, dan air

2. 2,2 gram kristal Iodida ditambahkan kedalam larutan dan diaduk hingga larut.



Gambar 3.6 Kristal iodida

3. Larutan diencerkan dengan 1 liter air. Diaduk hingga larutan tercampur sempurna.



Gambar 3.7 Hasil larutan iodine

3.4.2 Pengambilan Citra Potongan Apel

Apel dipotong menjadi dua secara horizontal dari ekuator. Salah satu potongan apel dicelupkan kedalam larutan selama 1 menit. Kemudian potongan apel diambil dan ditunggu hingga kering sebelum didokumentasikan.



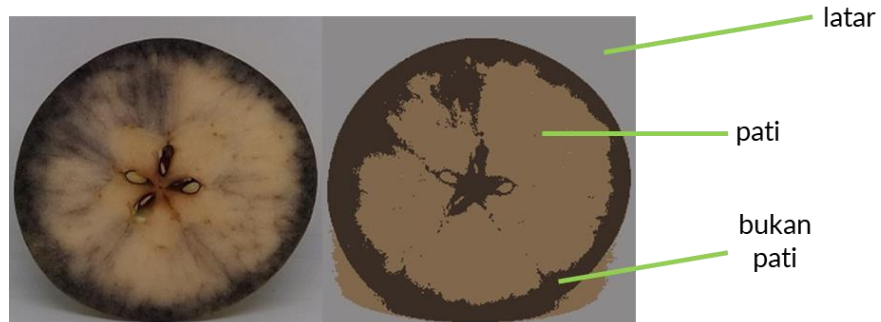
Gambar 3.8 Memotong apel



Gambar 3.9 Potongan apel setelah dicelupkan ke larutan iodine

3.4.3 Kluster Warna

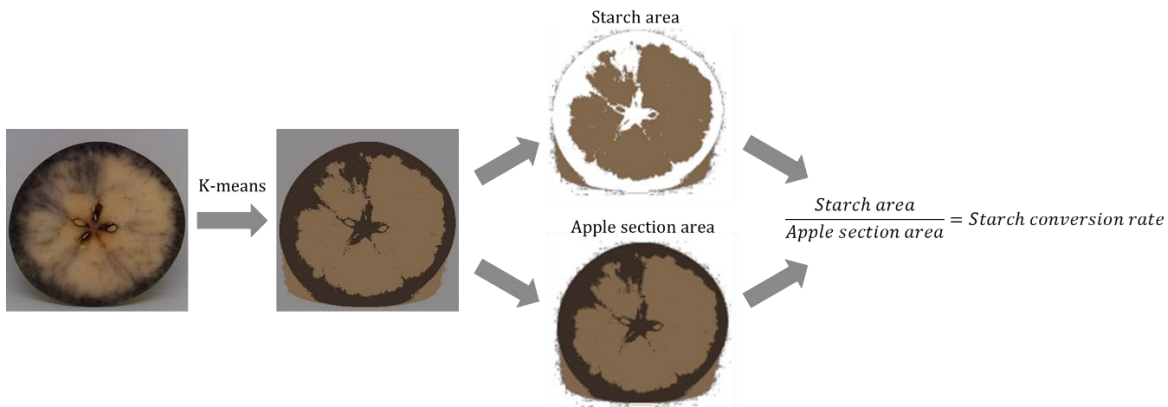
Kluster warna dilakukan untuk membagi citra menjadi 3. Tiap piksel dalam citra dikelompokkan menjadi *starch/pati*, *non-starch*, dan latar. Metode yang digunakan adalah *K-means clustering*.



Gambar 3.10 Hasil clustering potongan apel

3.4.4 Perhitungan SCR

Starch conversion rate (SCR) dihitung dengan membagi jumlah pixel dari bagian pati apel (*starch*) dengan luas total apel. Dengan menggunakan hasil dari *K-means clustering* maka dapat dikelompokkan piksel yang termasuk *starch*, *non-starch*, dan *background*.



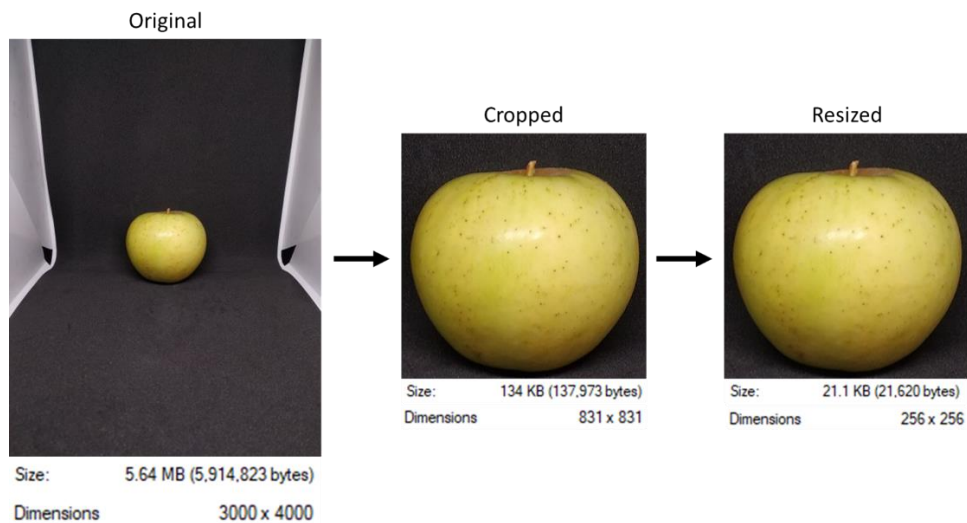
Gambar 3.11 Proses perhitungan starch conversion rate

3.4.5 Pelabelan

Apel yang telah diketahui nilai *starch conversion rate*-nya dikelompokkan menjadi dua, matang dan tidak matang. Apel yang dibawah nilai median dilabel apel tidak matang dan yang diatas nilai median dilabel apel matang.

3.5 Preprocessing Citra

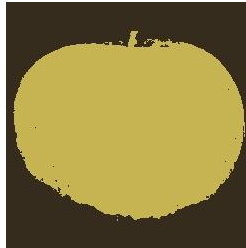
Foto apel yang berjumlah 100 gambar dipotong (*crop*) untuk meminimalkan area yang tidak digunakan. Kemudian gambar diperkecil ukurannya (*resize*) menjadi berdimensi 256 x 256. Resize dilakukan agar semua citra yang akan digunakan pada proses selanjutnya memiliki jumlah piksel atau dimensi yang seragam.



Gambar 3.12 Proses preprocessing gambar

3.6 RGB to CIE Lab

Untuk mendapat nilai RGB dari foto buah apel maka dilakukan *k-means clustering* dengan $k=2$. Sehingga piksel apel dapat dipisahkan dengan nilai piksel latar belakangnya. Selanjutnya dengan menggunakan matriks mengkonversi skala warna RGB ke CIE Lab.



Gambar 3.13 Contoh hasil *k-means clustering* apel

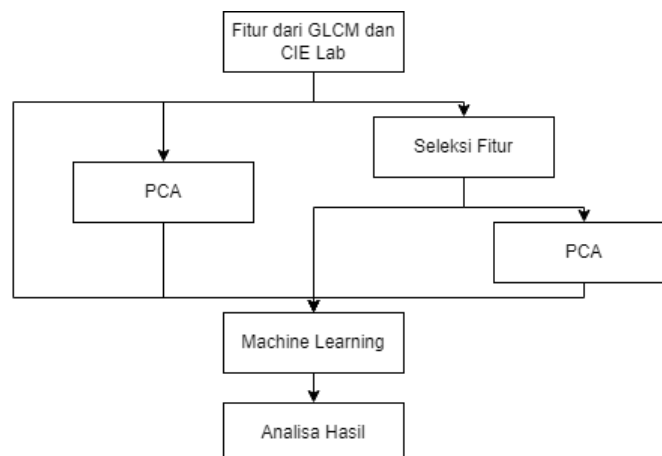
3.7 RGB to Greyscale

Untuk mengkonversi skala warna RGB ke Greyscale digunakan fungsi *imread as gray* pada *skimage python*. Tingkat warna abu-abu diatur dengan 8 tingkatan sehingga tiap piksel gambar memiliki nilai antara 0 (putih) hingga 7 (hitam).

3.8 Ekstraksi Fitur

Ekstraksi fitur tekstur gambar menggunakan *Gray Level Co-occurrence Matrix* (GLCM) dilakukan pada *python* dan menggunakan fungsi *skimage.feature* dan *greycoprops*. GLCM dilakukan dengan 3 *distance* ($d=1, 2, 3$) dan 4 *angles* ($0^\circ, 45^\circ, 90^\circ, 135^\circ$). Fitur yang diekstrak adalah *contrast*, *ASM*, *energy*, *homogeneity*, dan *dissimilarity*. Hasilnya didapatkan 60 fitur tiap gambar.

3.9 Training



Gambar 3.14 Flowchart training data

Sebelum masuk pelatihan model untuk klasifikasi tingkat kematangan apel digunakan 4 skenario berbeda berdasarkan ada tidaknya seleksi fitur dan penggunaan PCA untuk mereduksi dimensi fitur. Seleksi fitur dilakukan dengan *standard error of difference between means* (SEd). Dari SEd didapatkan nilai probabilitas *Z-value* dan dibandingkan dengan *critical Z-value* pada *confidence level* 80% dan 90%. Baik PCA dengan SEd maupun PCA tanpa SEd, keduanya mengurangi dimensi fitur menjadi 2. Akan tetapi perbedaannya adalah PCA tanpa SEd akan menggunakan seluruh fitur hasil ekstraksi sedangkan PCA dengan SEd akan menggunakan fitur-fitur mana yang telah memenuhi kriteria yang ditetapkan SEd. Skenario *machine learning* yang digunakan adalah sebagai berikut:

- Tanpa seleksi fitur SEd dan tanpa PCA
Pada skenario ini 60 fitur hasil ekstraksi GLCM dan 3 fitur nilai CIE Lab langsung masuk ke training sehingga total inputnya adalah 63 fitur.
- Tanpa seleksi fitur SEd dengan PCA
63 total fitur GLCM dan CIE Lab dilakukan analisa PCA dan diambil 2 *principal component* teratas.
- Dengan seleksi fitur SEd dan tanpa PCA
Fitur yang telah diseleksi dengan SEd jumlahnya bergantung dari hasil uji hipotesa dengan *confidence level* 80% dan 90%.
- Dengan seleksi fitur SEd dengan PCA
Fitur-fitur yang telah memenuhi uji hipotesa akan dianalisa PCA dan diambil 2 *principal component* teratas.

Terdapat 10 algoritma *machine learning* yang digunakan yaitu *Neural Net*, *Nearest Neighbors*, *Linear SVM*, *RBF SVM*, *Gaussian Process*, *Decision Tree*, *Random Forest*, *AdaBoost*, *Naive Bayes*, dan *QDA*. Konfigurasi tiap algoritma dapat dilihat pada Tabel 3.2. Pada *machine learning* digunakan rasio *training* dan *testing* yaitu 60:40 dengan menggunakan fungsi *train_test_split* pada *library sklearn*. Terdapat sepuluh algoritma *machine learning* yang digunakan dengan pengaturan yang berbeda. Pada algoritma *Neural Network* digunakan dua pengaturan yaitu default *MLPClassifier* dan dengan *hidden layer* (21,5). Pengaturan dua *hidden layer* ini mengikuti pengaturan pada penelitian (Lavindi *et al.*, 2019).

Tabel 3.2 Detail pengaturan algoritma machine learning

Algoritma	Library	Classifier
Nearest Neighbors	sklearn.neighbors	KNeighborsClassifier(3)
Linear SVM	sklearn.svm	SVC(kernel="linear", C=0.025)
RBF SVM	sklearn.svm	SVC(gamma=2, C=1)
Gaussian Process	sklearn.gaussian_process	GaussianProcessClassifier(1.0 * RBF(1.0))
Decision Tree	sklearn.tree	DecisionTreeClassifier(max_depth=5)
Random Forest	sklearn.ensemble	RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1)
Neural Network	sklearn.neural_network	MLPClassifier(alpha=1, max_iter=1000)
		MLPClassifier(hidden_layer_sizes=(21, 5), alpha = 0.0001, max_iter = 100, activation = 'logistic', learning_rate_init = 0.01, learning_rate = 'adaptive', solver = 'adam')
AdaBoost	sklearn.ensemble	AdaBoostClassifier()
Naïve-Bayes	sklearn.naive_bayes	GaussianNB()
QDA	sklearn.discriminant_analysis	QuadraticDiscriminantAnalysis()

3.10 Penulisan Laporan

Tahapan terakhir adalah penulisan laporan sesuai dengan data dan praktik yang dilakukan. Laporan disusun mulai dari pendahuluan hingga penutup sesuai dengan sistematika laporan. Laporan ditulis mengacu format laporan tugas akhir yang digunakan pada Departemen Teknik Fisika.

Halaman ini sengaja dikosongkan

BAB IV

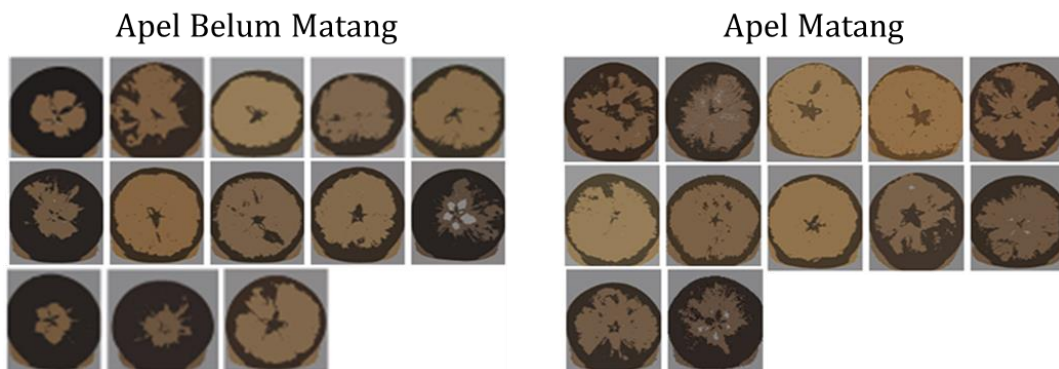
HASIL DAN PEMBAHASAN

4.1 Analisa Tingkat Kematangan Apel

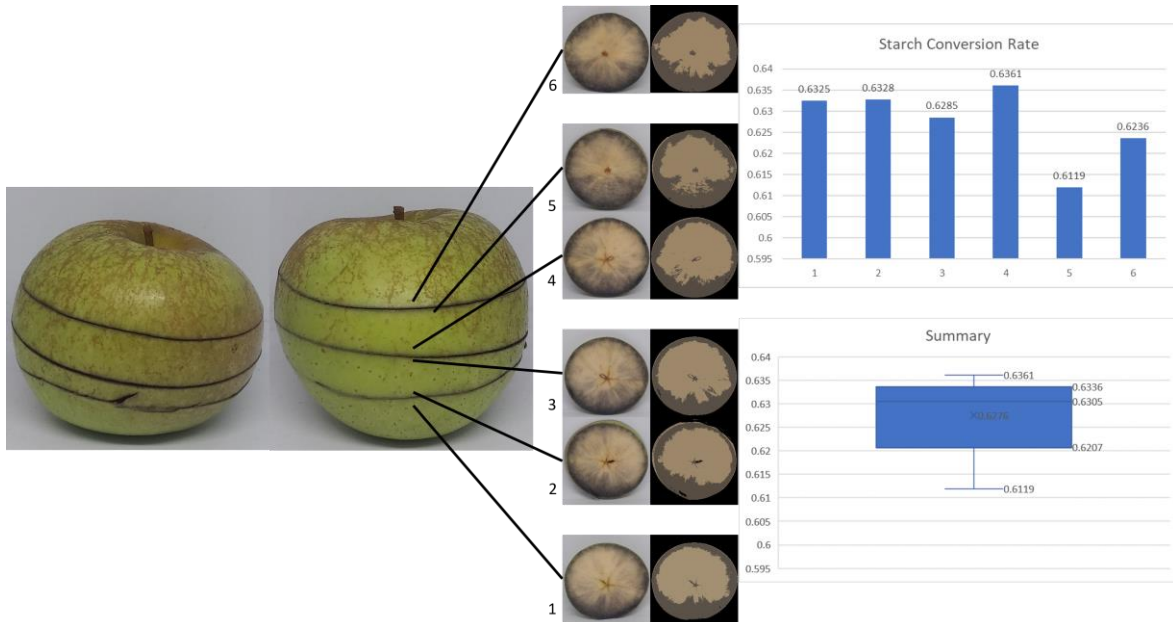
Tingkat kematangan apel dapat dilihat dari tekstur dan warna buah apel. Untuk mengetahui seberapa matang apel yang digunakan sebagai sampel maka digunakan *iodine test*. Apel yang matang akan memiliki lebih banyak area yang tidak berubah warna menjadi hitam. Dikarenakan pati (*starch*) tidak bereaksi terhadap larutan *iodine*. Apel yang *Starch Conversion Rate* $< 0,6784$ adalah apel belum matang dan apel yang *Starch Conversion Rate* $> 0,6784$ adalah apel matang. Contoh hasil tes seperti pada Gambar 4.1. Gambar yang telah didapatkan kemudian dipotong (*crop*) dan dikecilkan (*resize*) agar komputasi lebih ringan dan data seragam. Untuk mengetahui jumlah area yang terpengaruh larutan *iodine* maka dilakukan *k-means clustering*. *Starch Conversion Rate* didapat dengan membagi daerah berwarna terang (*starch*) dengan luas potongan apel. Sampel uji memiliki *starch conversion rate* seperti Gambar 4.2. Proses pematangan buah apel terjadi secara homogen artinya pada jarak berapapun pemotongan apel untuk hasil *Starch Conversion Rate*-nya sama. Contoh hasil *Starch Conversion Rate* pada Gambar 4.3 menunjukkan nilai yang mirip pada 6 luas penampang apel yang dipotong.



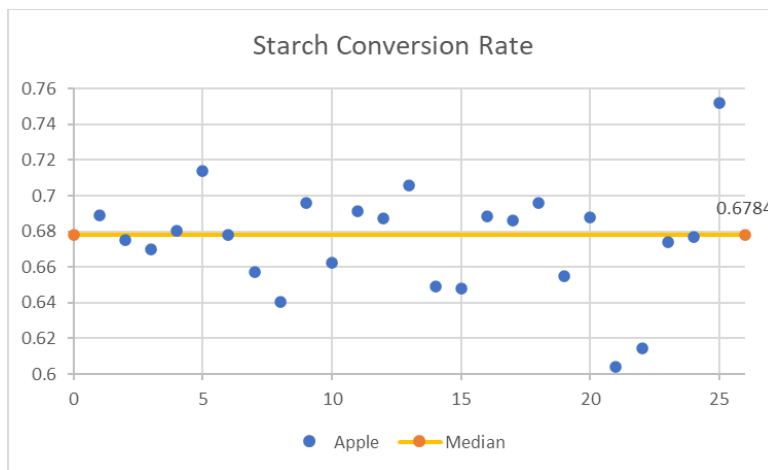
Gambar 4.1 Sampel hasil tes iodine



Gambar 4.2 Hasil pengelompokan k-means clustering



Gambar 4.3 Starch conversion rate buah apel



Gambar 4.4 Starch conversion rate sampel uji

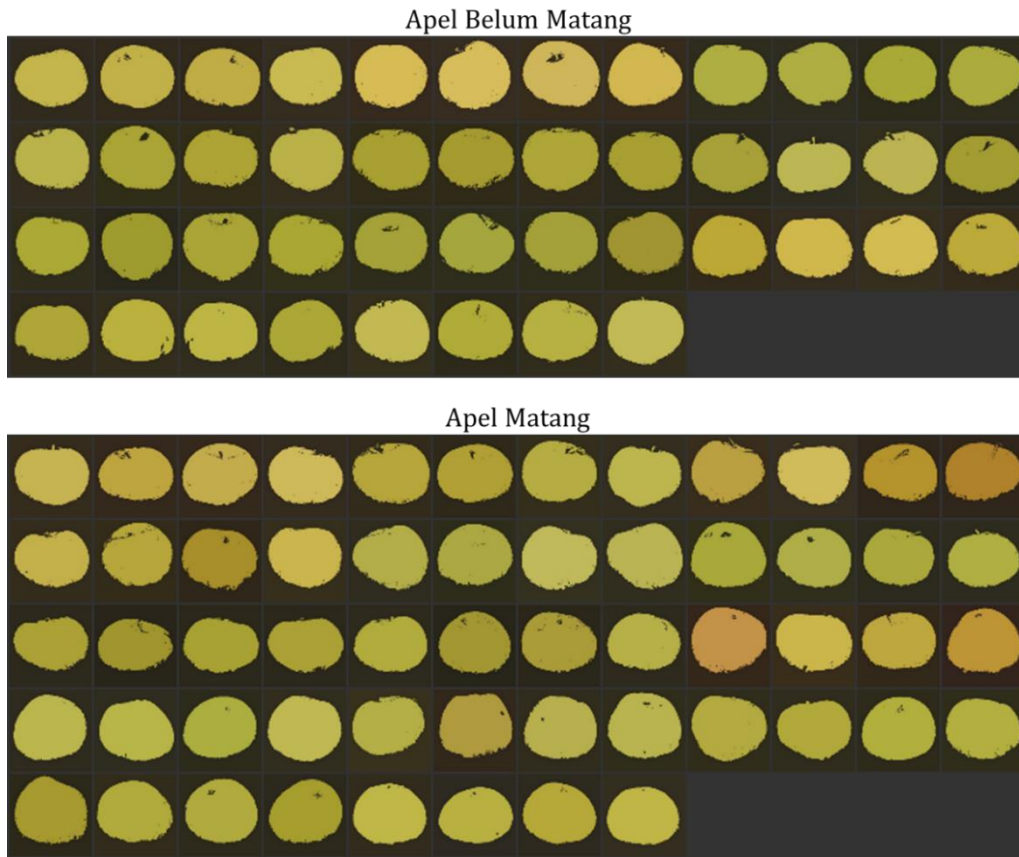
Tabel 4.1 Pembagian sesuai kelompok dari (Blanpied and Silsby, 1992)

Kelompok	Batas Bawah	Batas Atas	Jumlah Apel
1	0	0.125	0
2	0.125	0.25	0
3	0.25	0.375	0
4	0.375	0.5	0
5	0.5	0.625	2
6	0.625	0.75	22
7	0.75	0.875	1
8	0.875	1	0
Total			25

Starch conversion rate dari semua sampel berada pada rentang 0.604 hingga 0.752 dengan rerata 0.6751 dan standar deviasi 0.03. Menurut penelitian (Blanpied and Silsby, 1992), analisa tingkat kematangan buah apel dari pola *starch* dapat digolongkan menjadi 8 kelompok. Dengan mengelompokkan apel berdasarkan *starch conversion rate* menjadi 8 kelompok maka buah apel yang digunakan pada penelitian ini seperti pada Tabel 4.1. Pada penelitian ini sampel apel yang digunakan terbatas dan hamper seluruh apel termasuk kelompok 6. Bila menggunakan pengelompokkan sesuai penelitian (Blanpied and Silsby, 1992) maka rentang kelas buah apel menjadi tidak seimbang dan perbedaan sampel antar kelompok terlalu besar sehingga tidak ideal. Oleh karena itu digunakan median sebagai pemisah antar dua kelompok apel yang akan diprediksi. Hasil pembagiannya ada pada Gambar 4.4. Hasil pengelompokkan apel diaplikasikan ke foto apel untuk diberi label. Setiap apel difoto dari 4 sisi (depan, kanan, kiri, belakang). Untuk mendapatkan rerata RGB dari tiap foto apel maka digunakan *k-means clustering* dengan $k=2$. Hasilnya seperti Gambar 4.5. Secara kasat mata dapat dilihat bahwa apel matang memiliki warna yang lebih dominan merah dan kuning dibandingkan apel yang belum matang berwarna kuning dan hijau.



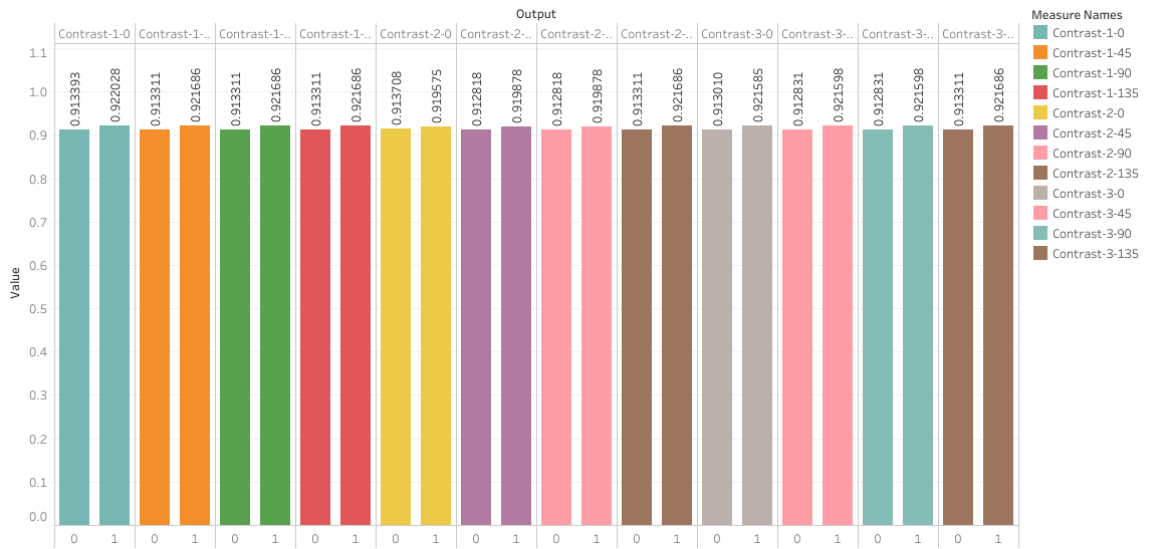
Gambar 4.5 Foto apel setelah dikelompokkan



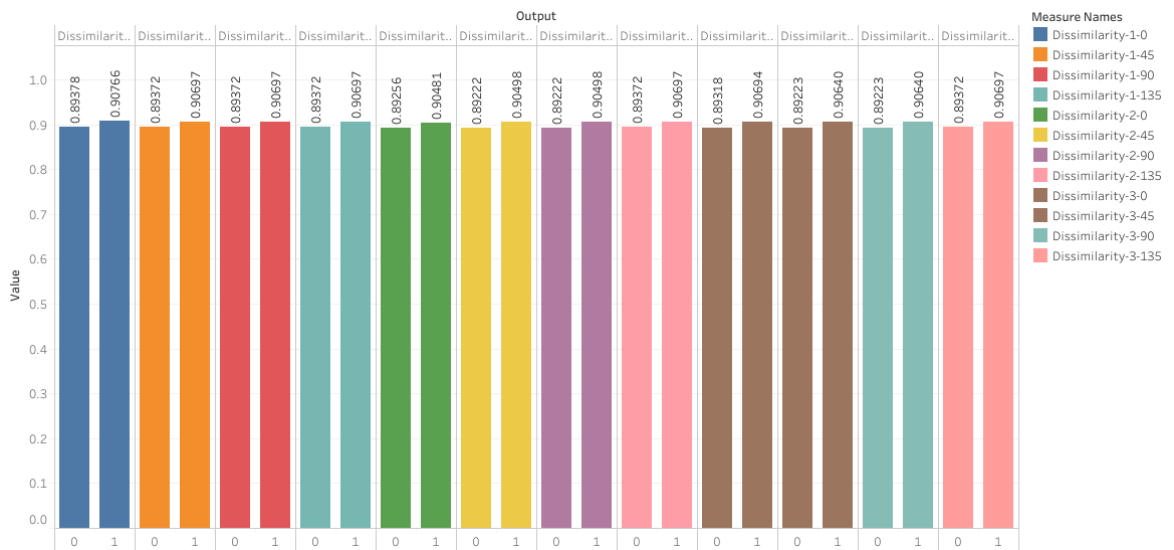
Gambar 4.6 RGB dari foto tiap apel

4.2 Analisa Ekstraksi Fitur GLCM dan CIE Lab

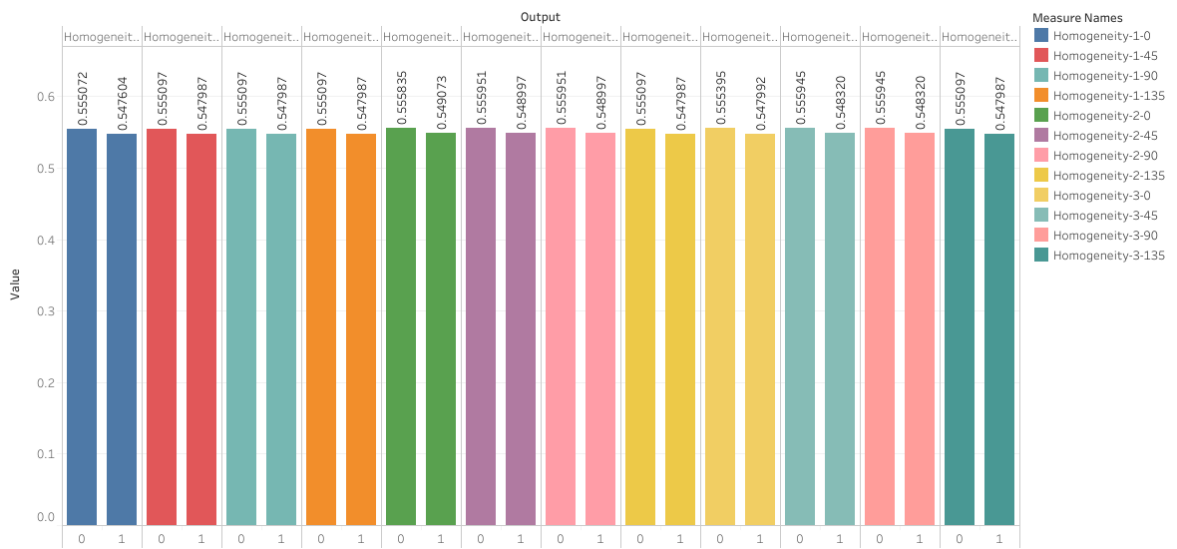
Foto dari apel yang sudah diproses dianalisa teksturnya dengan GLCM. Tiap foto dianalisa *contrast*, *ASM*, *energy*, *homogeneity*, dan *dissimilarity* untuk $d=1, 2, 3$ dan sudut= $0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}$. Fitur kontras, disimilaritas, dan homogeneitas tidak terlihat ada perbedaan signifikan pada dua kelompok apel yaitu kelompok 0 (apel yang dilabel belum matang) dan kelompok 1 (apel yang dilabel matang). Pada fitur ASM dan energi lebih terlihat perbedaan antara apel yang matang dan yang belum matang. Hasil rerata warna kulit buah apel yang telah dikonversi ke CIE Lab dapat dilihat pada Gambar 4.12.



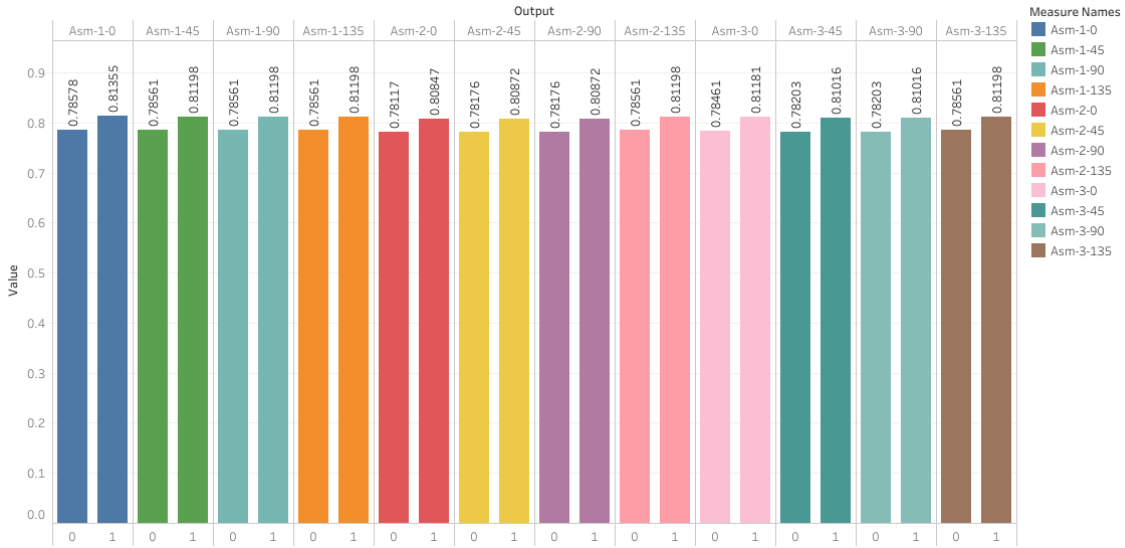
Gambar 4.7 Hasil ekstrasi GLCM fitur kontras



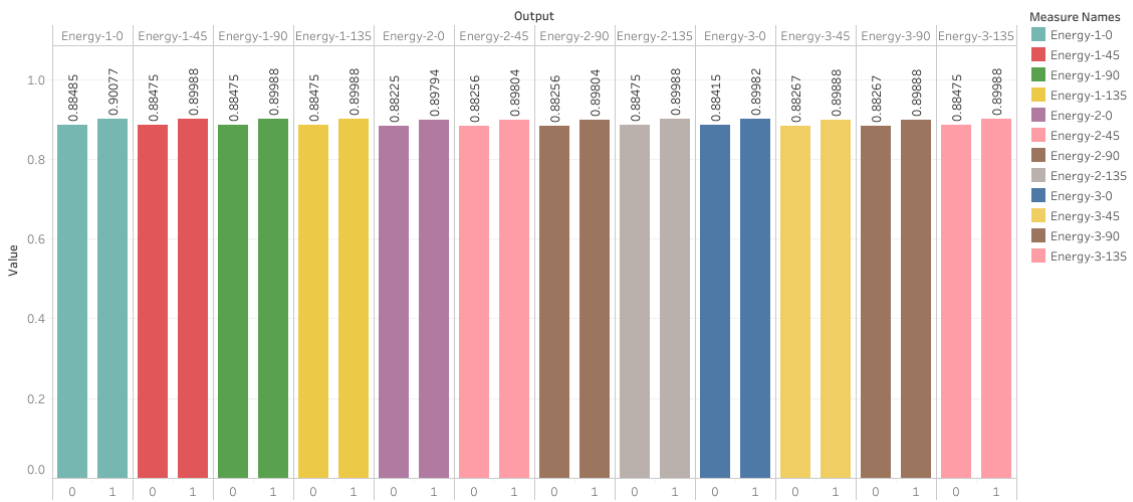
Gambar 4.8 Hasil ekstrasi GLCM fitur disimilaritas



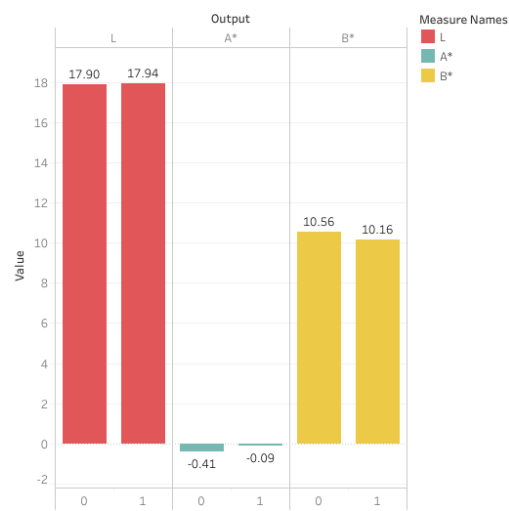
Gambar 4.9 Hasil ekstrasi GLCM fitur homogeneitas



Gambar 4.10 Hasil ekstraksi GLCM fitur ASM



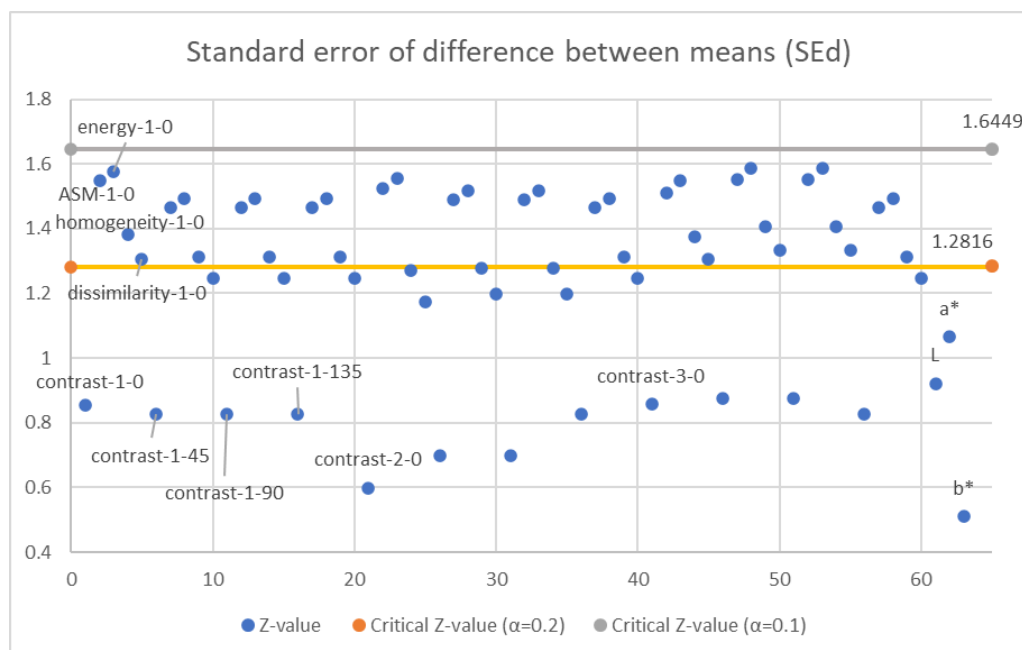
Gambar 4.11 Hasil ekstraksi GLCM fitur energi



Gambar 4.12 Hasil ekstraksi CIE Lab

4.3 Analisa SEd dan PCA

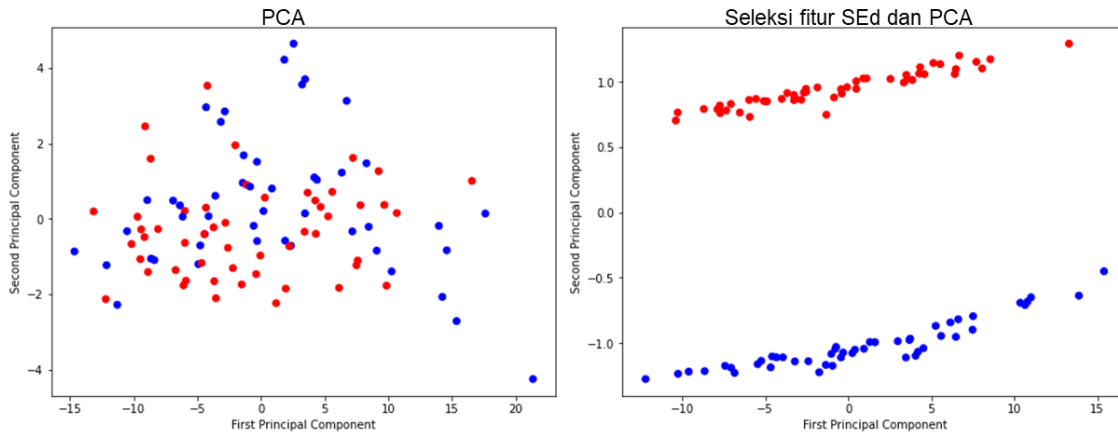
Untuk mengetahui fitur mana yang dapat membedakan antara apel belum matang dengan apel matang maka dilakukan seleksi fitur dengan SEd dan PCA. Dengan menggunakan metode *standard error of difference between means* (SEd) pada *confidence level* 90% juga tidak ada fitur yang memenuhi kriteria melebihi *critical Z-value*. Namun pada *confidence level* 80% ($\alpha=0.2$) didapatkan hasil seperti Gambar 4.12. Fitur yang tidak melebihi *critical Z-value* tidak digunakan termasuk nilai CIE L, a*, b* yang juga tidak memenuhi kriteria. Ringkasan fitur GLCM yang dipilih terdapat pada Tabel 4.2. Dengan mengkombinasikan SEd dengan PCA maka didapatkan dua *principal components* terbaik yang secara langsung dapat dilihat dua kelompok apel seperti Gambar 4.14. Dengan menggunakan SEd maka fitur-fitur yang tidak memiliki perbedaan rerata yang cukup besar untuk memenuhi kriteria *confidence level* 80% dapat tidak digunakan. Hasilnya adalah PCA dengan SEd tidak akan dipengaruhi oleh fitur-fitur yang sebenarnya tidak dibutuhkan sehingga PCA pada Gambar 4.14 (kanan) dapat menghasilkan *principal components* yang dapat membedakan kelompok apel matang (merah) dan kelompok apel belum matang (biru).



Gambar 4.13 Standard error of difference between means seluruh fitur

Tabel 4.2 Hasil seleksi fitur SED pada GLCM

	d=1				d=2				d=3				0 = not selected 1 = selected
	0	45	90	135	0	45	90	135	0	45	90	135	
contrast	0	0	0	0	0	0	0	0	0	0	0	0	0
ASM	1	1	1	1	1	1	1	1	1	1	1	1	1
energy	1	1	1	1	1	1	1	1	1	1	1	1	1
homogeneity	1	1	1	1	0	0	0	1	1	1	1	1	1
dissimilarity	1	0	0	0	0	0	0	0	1	1	1	0	0

**Gambar 4.14** Perbedaan ada tidaknya SED sebelum PCA

4.4 Analisa Machine Learning

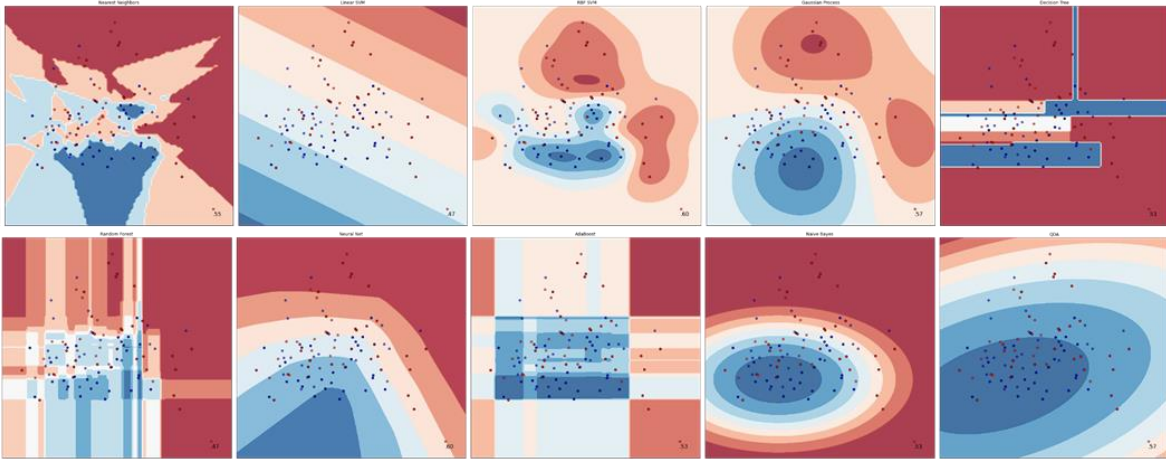
Machine learning dilakukan tanpa seleksi fitur (60 fitur GLCM dan 3 CIE Lab) dan dengan seleksi fitur SED (37 fitur GLCM). Akurasi dari tiap metode dirangkum pada Tabel 4.3. Model *machine learning* untuk prediksi tingkat kematangan buah apel yang dilihat dari seleksi fitur tekstur dengan GLCM dan fitur warna dengan CIE Lab dikorelasikan dengan *starch conversion rate* pada penelitian ini memiliki akurasi yang bervariasi. Rata-rata nilai F1 pada model yang menggunakan seleksi SED dan PCA adalah 1.0 dibandingkan dengan yang hanya seleksi SED, hanya seleksi PCA, dan tanpa keduanya berturut-turut 0.59, 0.55, dan 0.51. Hal ini menunjukkan bahwa fitur yang digunakan PCA setelah seleksi fitur SED adalah fitur yang dapat membedakan mana apel yang belum matang dan apel matang. Dapat dikatakan bahwa metode ini dapat memprediksi mana apel yang telah dilabel apel matang dari *starch conversion rate*-nya. Dengan menggunakan *dimension reduction technique* yaitu PCA didapatkan akurasi yang signifikan lebih baik setelah seleksi fitur. Dapat dilihat perbandingan dari Gambar 4.15 menjadi Gambar 4.16. Pada Gambar 4.15 algoritma *machine learning* tidak dapat mengklasifikasi dua kelompok apel dan pengelompokannya cenderung acak dan tidak beraturan yang mengakibatkan skor f1 yang didapatkan memiliki

rata-rata 0.55. Berbeda dengan Gambar 4.16, algoritma *machine learning* dapat dengan baik mengkategorikan kelompok apel matang dan kelompok belum matang. Rerata skor f1-nya adalah 1 sehingga dapat dikatakan bahwa metode ini yaitu menggunakan seleksi fitur SEd dan PCA setelah itu dengan *machine learning* dapat memprediksi buah apel dari tingkat kematangannya.

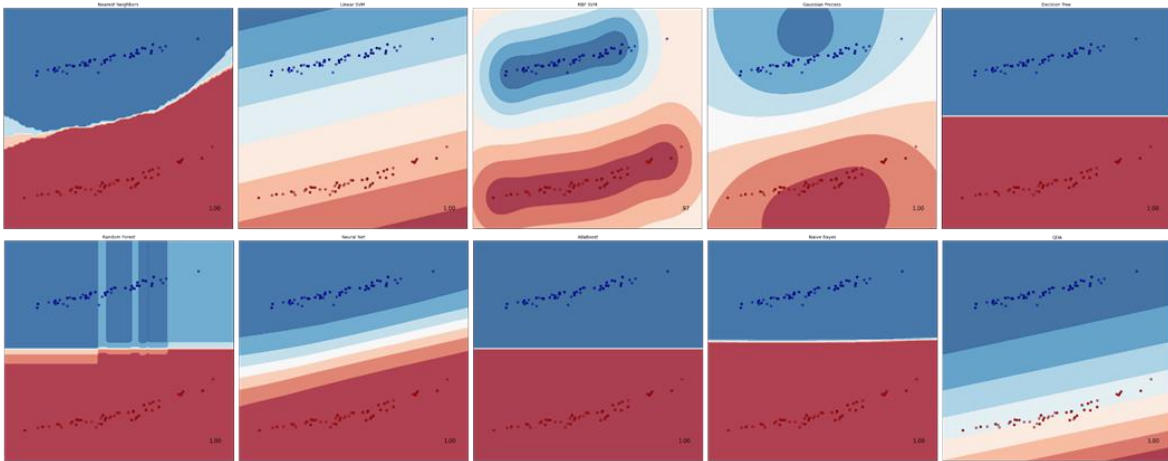
Tabel 4.3 Hasil traning berbagai metode

		TN	FN	FP	TP	accuracy	precision	sensitivity	specificity	F1	
Neural Net (63-21-5-1)		7	1	18	14	0.53	0.44	0.93	0.88	0.60	
Nearest Neighbors		9	13	9	9	0.45	0.50	0.41	0.41	0.45	
Linear SVM		9	10	9	12	0.53	0.57	0.55	0.47	0.56	
RBF SVM		2	0	16	22	0.60	0.58	1.00	1.00	0.73	
Gaussian Process		12	12	6	10	0.55	0.63	0.45	0.50	0.53	
Decision Tree		11	12	7	10	0.53	0.59	0.45	0.48	0.51	
Random Forest		9	15	9	7	0.40	0.44	0.32	0.38	0.37	
Neural Net (63-100-1)		10	15	8	7	0.43	0.47	0.32	0.40	0.38	
AdaBoost		14	14	4	8	0.55	0.67	0.36	0.50	0.47	
Naive Bayes		2	4	16	18	0.50	0.53	0.82	0.33	0.64	
QDA		14	17	4	5	0.48	0.56	0.23	0.45	0.32	
Mean						0.50	0.54	0.53	0.53	0.51	
PCA	Nearest Neighbors	12	12	6	10	0.55	0.63	0.45	0.50	0.53	
	Linear SVM	6	9	12	13	0.48	0.52	0.59	0.40	0.55	
	RBF SVM	12	10	6	12	0.60	0.67	0.55	0.55	0.60	
	Gaussian Process	10	9	8	13	0.58	0.62	0.59	0.53	0.60	
	Decision Tree	12	13	6	9	0.53	0.60	0.41	0.48	0.49	
	Random Forest	11	17	7	5	0.40	0.42	0.23	0.39	0.29	
	Neural Net (2-100-1)	11	10	7	12	0.58	0.63	0.55	0.52	0.59	
	AdaBoost	9	8	9	14	0.58	0.61	0.64	0.53	0.62	
	Naive Bayes	7	8	11	14	0.53	0.56	0.64	0.47	0.60	
Mean						0.54	0.59	0.52	0.49	0.55	
Seleksi SEd	Neural Net (37-12-5-1)	8	12	10	10	0.45	0.50	0.45	0.40	0.48	
	Nearest Neighbors	8	7	10	15	0.58	0.60	0.68	0.53	0.64	
	Linear SVM	5	6	13	16	0.53	0.55	0.73	0.45	0.63	
	RBF SVM	11	9	7	13	0.60	0.65	0.59	0.55	0.62	
	Gaussian Process	9	9	9	13	0.55	0.59	0.59	0.50	0.59	
	Decision Tree	6	9	12	13	0.48	0.52	0.59	0.40	0.55	
	Random Forest	11	11	7	11	0.55	0.61	0.50	0.50	0.55	
	Neural Net (37-100-1)	5	6	13	16	0.53	0.55	0.73	0.45	0.63	
	AdaBoost	11	11	7	11	0.55	0.61	0.50	0.50	0.55	
	Naive Bayes	3	5	15	17	0.50	0.53	0.77	0.38	0.63	
	QDA	7	8	11	14	0.53	0.56	0.64	0.47	0.60	
	Mean						0.53	0.57	0.62	0.47	0.59
	PCA	Nearest Neighbors	18	0	0	22	1.00	1.00	1.00	1.00	1.00
		Linear SVM	18	0	0	22	1.00	1.00	1.00	1.00	1.00
		RBF SVM	18	1	0	21	0.98	1.00	0.95	0.95	0.98
Gaussian Process		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
Decision Tree		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
Random Forest		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
Neural Net (2-100-1)		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
AdaBoost		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
Naive Bayes		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
QDA		18	0	0	22	1.00	1.00	1.00	1.00	1.00	
Mean						1.00	1.00	1.00	0.99	1.00	





Gambar 4.15 Machine Learning tanpa seleksi fitur SED (dari kiri atas Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes, QDA)



Gambar 4.16 Machine Learning dengan seleksi fitur SED (dari kiri atas Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisa penelitian yang telah dilakukan, maka kesimpulan yang dapat diambil adalah prediksi tingkat kematangan buah apel dapat dilakukan dengan menggunakan ekstraksi fitur GLCM (*Gray Level Co-occurrence Matrix*), seleksi fitur *Z-value* SED (*Standard Error of difference means*), dan analisa PCA (*Principal Component Analysis*). Ekstraksi Fitur GLCM seperti *contrast*, *ASM*, *energy*, *homogeneity*, dan *dissimilarity* dilakukan pada jarak 1, 2, dan 3 serta sudut 0° , 45° , 90° , dan 135° . Namun perlu dilakukan seleksi fitur *Z-value* SED untuk menentukan fitur mana yang dapat menunjukkan perbedaan apel matang dari yang belum matang. Skala warna CIE Lab tidak dapat membedakan dua kelompok apel. Dengan mengurangi dimensi fitur menjadi dua *principal component* melalui analisa PCA maka dapat dilakukan training dengan algoritma *machine learning* yang memiliki akurasi 100% yaitu *Nearest Neighbors*, *Linear SVM*, *Gaussian Process*, *Decision Tree*, *Random Forest*, *Neural Net*, *AdaBoost*, *Naive Bayes*, dan *QDA*.

5.2 Saran

Saran untuk penelitian selanjutnya sebagai berikut:

- Model yang menggunakan GLCM dan PCA dapat digunakan untuk klasifikasi dua kelas apel (matang dan belum matang). Untuk penelitian selanjutnya dapat menganalisa bagaimana performa model ini digunakan pada lebih banyak tingkat kematangan apel.
- Penelitian ini fokus pada prediksi apel dan untuk selanjutnya dapat ditingkatkan menjadi mendeteksi dan memprediksi tingkat kematangan apel yang belum dipetik dari pohon apel.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Ayu, R.J. (2019) “Batik Pattern Identification using GLCM and Artificial Neural Network Backpropagation,” *2018 Third International Conference on Informatics and Computing (ICIC)* [Preprint]. Available at: <https://doi.org/10.1109/IAC.2018.8780412>.
- Benos, L. *et al.* (2021) “Machine learning in agriculture: A comprehensive updated review,” *Sensors*. MDPI AG. Available at: <https://doi.org/10.3390/s21113758>.
- Blanpied, G.D. and Silsby, K.J. (1992) “Predicting Harvest Date Windows for Apples,” *Cornell University Information Bulletin* 221, July.
- Cárdenas-Pérez, S. *et al.* (2017) “Evaluation of the ripening stages of apple (Golden Delicious) by means of computer vision system,” *Biosystems Engineering*, 159, pp. 46–58. Available at: <https://doi.org/10.1016/j.biosystemseng.2017.04.009>.
- Cavallo, D. *et al.* (2019) “Non-destructive and contactless quality evaluation of table grapes by a computer vision system,” *Computers and Electronics in Agriculture*, 156(June 2018), pp. 558–564. Available at: <https://doi.org/10.1016/j.compag.2018.12.019>.
- Duprat, F. *et al.* (1995) “Laser Light Based Machine System for Nondestructive Ripeness Sensing of Golden Apples,” *IFAC Proceedings Volumes*, 28(6), pp. 117–123. Available at: [https://doi.org/10.1016/s1474-6670\(17\)47170-9](https://doi.org/10.1016/s1474-6670(17)47170-9).
- Gao, S. (2021) “Gray level co-occurrence matrix and extreme learning machine for Alzheimer’s disease diagnosis,” *International Journal of Cognitive Computing in Engineering*, 2, pp. 116–129. Available at: <https://doi.org/10.1016/j.ijcce.2021.08.002>.
- Haralick, R.M., Shanmugam, K. and Dinstein, I. (1973) “Textural Features of Image Classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), pp. 610–621. Available at: <https://doi.org/10.1109/TSMC.1973.4309314>.
- Hughes, G.P. (1968) “On the mean accuracy of statistical pattern recognizers,” *IEEE Transactions on Information Theory*, 14(1), pp. 55–63.
- Ikeuchi, K. (2021) *Computer Vision: A Reference Guide Second Edition*. Edited by K. Ikeuchi. Cham: Springer International Publishing. Available at: <https://doi.org/10.1007/978-3-030-63416-2>.
- Julio Sarosa, S.A., Utamingrum, F. and Bachtiar, F.A. (2019) “Breast Cancer Classification Using GLCM and BPNN,” *Int. J. Advance Soft Compu. Appl*, 11(3).
- Lavindi, E.E. *et al.* (2019) “Neural Network based on GLCM, and CIE L*a*b* Color Space to Classify Tomatoes Maturity,” *2019 International Seminar on Application for*

- Technology of Information and Communication (iSemantic)* [Preprint]. Available at: <https://doi.org/10.1109/ISEMANTIC.2019.8884307>.
- Makmur Sitompul, S. and Sugito, Y. (2013) “Spatial Productivity Analysis of Tropical Apple (*Malus sylvestris* Mill) in Relation to Temperature with PCRaster,” *Journal of Agricultural Science and Technology A*, pp. 183–192.
- Nagesh Singh Chauhan (2019) *Introduction to Image Segmentation with K-Means clustering*, <https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html>.
- Pace, B. *et al.* (2015) “Adaptive self-configuring computer vision system for quality evaluation of fresh-cut radicchio,” *Innovative Food Science and Emerging Technologies*, 32, pp. 200–207. Available at: <https://doi.org/10.1016/j.ifset.2015.10.001>.
- Pace, B. *et al.* (2017) *Automatic Identification of Relevant Colors in Non-Destructive Quality Evaluation of Fresh Salad Vegetables*, *International Journal of Food Processing Technology*.
- Peirs, A. *et al.* (2002) *Uncertainty analysis and modelling of the starch index during apple fruit maturation*, *Postharvest Biology and Technology*. Available at: www.elsevier.com/locate/postharvbio.
- Pi, P. and Lima, D. (2021) “Gray level co-occurrence matrix and extreme learning machine for Covid-19 diagnosis,” *International Journal of Cognitive Computing in Engineering*, 2, pp. 93–103. Available at: <https://doi.org/10.1016/j.ijcce.2021.05.001>.
- Rutkowski, K.P., Michalczuk, B. and Konopacki, P. (2008) “Nondestructive Determination Of ‘Golden Delicious’ Apple Quality and Harvest Maturity,” *Journal of Fruit and Ornamental Plant Research*, 16, pp. 39–52.
- Valous, N.A. *et al.* (2009) “Colour calibration of a laboratory computer vision system for quality evaluation of pre-sliced hams,” *Meat Science*, 81(1), pp. 132–141. Available at: <https://doi.org/10.1016/j.meatsci.2008.07.009>.
- Zhang, C., Zou, K. and Pan, Y. (2020) “A method of apple image segmentation based on color-texture fusion feature and machine learning,” *Agronomy*, 10(7). Available at: <https://doi.org/10.3390/agronomy10070972>.
- Zhang, M. *et al.* (2020) “Determination of bagged ‘Fuji’ apple maturity by visible and near-infrared spectroscopy combined with a machine learning algorithm,” *Infrared Physics and Technology*, 111. Available at: <https://doi.org/10.1016/j.infrared.2020.103529>.

LAMPIRAN

Semua file dapat dilihat di <https://github.com/vieri03/Prediction-of-Apple-Ripeness>

A. Image Segmentation - K-means Clustering

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import glob
import csv

from google.colab import drive
drive.mount('/content/gdrive')
%cd /content/gdrive/My Drive/Colab Notebooks

path = '/content/gdrive/My Drive/Colab Notebooks/K-means-after/*.*'

filename = '/content/gdrive/My Drive/Colab Notebooks/K-means-
result/conversion_rate_after_new.csv'

# open the file in the write mode
f = open(filename, 'w')

# create the csv writer
writer = csv.writer(f)
head = ["Number", "Image", "Cluster 0", "Cluster 1", "Cluster 2", "Sum
Cluster 0", "Sum Cluster 1", "Sum Cluster 2"]
writer.writerow(head)
#####

# K-means clustering, K=3
image_no = 1
for file in sorted(glob.glob(path)):
    original_image = cv.imread(file)
    img =cv.cvtColor(original_image,cv.COLOR_BGR2RGB)
    vectorize = img.reshape((-1,3))

    # convert to np.float32
    vectorized = np.float32(vectorize)

    # define criteria, number of clusters(K) and apply kmeans()
    criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 10, 1
.0)
    K = 3
    attempts = 10
    ret,label,center=cv.kmeans(vectorized,K, None,criteria,attempts,cv.K
MEANS_PP_CENTERS)
```

```

# Now convert back into uint8, and make original image
center = np.uint8(center)
res = center[label.flatten()]
result_image = res.reshape((img.shape))
result_image2 = cv.cvtColor(result_image,cv.COLOR_RGB2BGR)

result = '/content/gdrive/My Drive/Colab Notebooks/K-means-
result/segmented' + str(image_no) + '.jpg'
cv.imwrite(result, result_image2)

pixel_0 = np.sum(center[0])
pixel_1 = np.sum(center[1])
pixel_2 = np.sum(center[2])

row = [image_no, file, center[0], center[1], center[2], pixel_0, pi
xel_1, pixel_2]

# write a row to the csv file
writer.writerow(row)

image_no += 1

# close the file
f.close()

#####

# K-means clustering, K=2 RGB to Lab
# RGB to Lab conversion
#source:
https://gist.github.com/manojpandey/f5ece715132c572c80421febeba66ae
# Step 1: RGB to XYZ      http://www.easyrgb.com/index.php?X=MATH&H=0
2#text2
# Step 2: XYZ to Lab      http://www.easyrgb.com/index.php?X=MATH&H=0
7#text7

def rgb2lab(inputColor):
    num = 0
    RGB = [0, 0, 0]
    for value in inputColor:
        value = float(value) / 255
        if value > 0.04045:
            value = ((value + 0.055) / 1.055) ** 2.4
        else:
            value = value / 12.92
        RGB[num] = value * 100
        num = num + 1

    XYZ = [0, 0, 0, ]

```

```

X = RGB[0] * 0.4124 + RGB[1] * 0.3576 + RGB[2] * 0.1805
Y = RGB[0] * 0.2126 + RGB[1] * 0.7152 + RGB[2] * 0.0722
Z = RGB[0] * 0.0193 + RGB[1] * 0.1192 + RGB[2] * 0.9505
XYZ[0] = round(X, 4)
XYZ[1] = round(Y, 4)
XYZ[2] = round(Z, 4)

# Observer= 2°, Illuminant= D65
XYZ[0] = float(XYZ[0]) / 95.047      # ref_X = 95.047
XYZ[1] = float(XYZ[1]) / 100.0      # ref_Y = 100.000
XYZ[2] = float(XYZ[2]) / 108.883    # ref_Z = 108.883

num = 0
for value in XYZ:
    if value > 0.008856:
        value = value ** (0.3333333333333333)
    else:
        value = (7.787 * value) + (16 / 116)
    XYZ[num] = value
    num = num + 1

Lab = [0, 0, 0]
L = (116 * XYZ[1]) - 16
a = 500 * (XYZ[0] - XYZ[1])
b = 200 * (XYZ[1] - XYZ[2])

Lab[0] = round(L, 4)
Lab[1] = round(a, 4)
Lab[2] = round(b, 4)
return Lab

```

B. GLCM

```

import numpy as np
from skimage import io
from scipy import stats
from skimage.feature import greycoprops
import glob
import csv

def offset(length, angle):
    """Return the offset in pixels for a given length and angle"""
    dv = length * np.sign(-np.sin(angle)).astype(np.int32)
    dh = length * np.sign(np.cos(angle)).astype(np.int32)
    return dv, dh

def crop(img, center, win):
    """Return a square crop of img centered at center (side = 2*win +
1)"""
    row, col = center

```

```

    side = 2*win + 1
    first_row = row - win
    first_col = col - win
    last_row = first_row + side
    last_col = first_col + side
    return img[first_row: last_row, first_col: last_col]

def cooc_maps(img, center, win, d=[1], theta=[0], levels=256):
    """
    Return a set of co-
    occurrence maps for different d and theta in a square
    crop centered at center (side = 2*w + 1)
    """
    shape = (2*win + 1, 2*win + 1, len(d), len(theta))
    cooc = np.zeros(shape=shape, dtype=np.int32)
    row, col = center
    Ii = crop(img, (row, col), win)
    for d_index, length in enumerate(d):
        for a_index, angle in enumerate(theta):
            dv, dh = offset(length, angle)
            Ij = crop(img, center=(row + dv, col + dh), win=win)
            cooc[:, :, d_index, a_index] = encode_cooccurrence(Ii, Ij
, levels)
    return cooc

def encode_cooccurrence(x, y, levels=256):
    """Return the code corresponding to co-
    occurrence of intensities x and y"""
    return x*levels + y

def decode_cooccurrence(code, levels=256):
    """Return the intensities x, y corresponding to code"""
    return code//levels, np.mod(code, levels)

def compute_glcms(cooccurrence_maps, levels=256):
    """Compute the cooccurrence frequencies of the cooccurrence maps"""
    Nr, Na = cooccurrence_maps.shape[2:]
    glcms = np.zeros(shape=(levels, levels, Nr, Na), dtype=np.float64
)
    for r in range(Nr):
        for a in range(Na):
            table = stats.itemfreq(cooccurrence_maps[:, :, r, a])
            codes = table[:, 0]
            freqs = table[:, 1]/float(table[:, 1].sum())
            i, j = decode_cooccurrence(codes, levels=levels)
            glcms[i, j, r, a] = freqs
    return glcms

```

```

def compute_props(glcms, props=('contrast',)):
    """Return a feature vector corresponding to a set of GLCM"""
    Nr, Na = glcms.shape[2:]
    features = np.zeros(shape=(Nr, Na, len(props)))
    for index, prop_name in enumerate(props):
        features[:, :, index] = greycoprops(glcms, prop_name)
    return features.ravel()

def haralick_features(img, win, d, theta, levels, props):
    """Return a map of Haralick features (one feature vector per pixel)"""
    rows, cols = img.shape
    margin = win + max(d)
    arr = np.pad(img, margin, mode='reflect')
    n_features = len(d) * len(theta) * len(props)
    feature_map = np.zeros(shape=(rows, cols, n_features), dtype=np.float64)
    for m in range(rows):
        for n in range(cols):
            coocs = cooc_maps(arr, (m + margin, n + margin), win, d, theta, levels)
            glcms = compute_glcms(coocs, levels)
            feature_map[m, n, :] = compute_props(glcms, props)
    return feature_map

#####
from google.colab import drive
drive.mount('/content/gdrive')
%cd /content/gdrive/My Drive/Colab Notebooks/

#####

import os
from os import listdir
from PIL import Image

path = '/content/gdrive/My Drive/Colab Notebooks/feature/'

for filename in listdir(path):
    if filename.endswith('.jpg'):
        try:
            img = Image.open(path + filename) # open the image file
            img.verify() # verify that it is, in fact an image
        except (IOError, SyntaxError) as e:
            print('Bad file:', filename)

path = '/content/gdrive/My Drive/Colab Notebooks/feature/*.*'

```

1

```
filename = '/content/gdrive/My Drive/Colab Notebooks/result-image-
twentyfifth.csv'

# open the file in the write mode
f = open(filename, 'w')

# create the csv writer
writer = csv.writer(f)
head = ["Number", "Image", "contrast-1-0", "ASM-1-0", "energy-1-
0", "homogeneity-1-0", "dissimilarity-1-0",
        "contrast-1-45", "ASM-1-45", "energy-1-45", "homogeneity-1-
45", "dissimilarity-1-45",
        "contrast-1-90", "ASM-1-90", "energy-1-90", "homogeneity-1-
90", "dissimilarity-1-90",
        "contrast-1-135", "ASM-1-135", "energy-1-135", "homogeneity-
1-135", "dissimilarity-1-135",
        "contrast-2-0", "ASM-2-0", "energy-2-0", "homogeneity-2-
0", "dissimilarity-2-0",
        "contrast-2-45", "ASM-2-45", "energy-2-45", "homogeneity-2-
45", "dissimilarity-2-45",
        "contrast-2-90", "ASM-2-90", "energy-2-90", "homogeneity-2-
90", "dissimilarity-2-90",
        "contrast-2-135", "ASM-2-135", "energy-2-135", "homogeneity-
2-135", "dissimilarity-2-135",
        "contrast-3-0", "ASM-3-0", "energy-3-0", "homogeneity-3-
0", "dissimilarity-3-0",
        "contrast-3-45", "ASM-3-45", "energy-3-45", "homogeneity-3-
45", "dissimilarity-3-45",
        "contrast-3-90", "ASM-3-90", "energy-3-90", "homogeneity-3-
90", "dissimilarity-3-90",
        "contrast-3-135", "ASM-3-135", "energy-3-135", "homogeneity-
3-135", "dissimilarity-3-135"]
writer.writerow(head)

image_no = 1
for file in sorted(glob.glob(path)): #sorted by name
    img = io.imread(file, as_gray=True)

    d = (1, 2, 3)
    theta = (0, np.pi/4, np.pi/2, 3*np.pi/4)
    props = ('contrast', 'ASM', 'energy', 'homogeneity', 'dissimilarity
')
    levels = 8
    win = 19

    feature_map = haralick_features(img, win, d, theta, levels, props)
    feature_map[0, 0, :]
```



```

#split distance
array = np.array_split(feature_map[0, 0, :], 3)

#split angles
newarr_0 = np.array_split(array[0], 4)
newarr_1 = np.array_split(array[1], 4)
newarr_2 = np.array_split(array[2], 4)

#split properties
result_0_0 = np.array_split(newarr_0[0], 5)
result_0_1 = np.array_split(newarr_0[1], 5)
result_0_2 = np.array_split(newarr_0[2], 5)
result_0_3 = np.array_split(newarr_0[3], 5)

result_1_0 = np.array_split(newarr_1[0], 5)
result_1_1 = np.array_split(newarr_1[1], 5)
result_1_2 = np.array_split(newarr_1[2], 5)
result_1_3 = np.array_split(newarr_0[3], 5)

result_2_0 = np.array_split(newarr_2[0], 5)
result_2_1 = np.array_split(newarr_2[1], 5)
result_2_2 = np.array_split(newarr_2[2], 5)
result_2_3 = np.array_split(newarr_0[3], 5)

row = [image_no, file, result_0_0[0], result_0_0[1], result_0_0[2],
result_0_0[3], result_0_0[4],
      result_0_1[0], result_0_1[1], result_0_1[2], result_0_1[3],
result_0_1[4],
      result_0_2[0], result_0_2[1], result_0_2[2], result_0_2[3],
result_0_2[4],
      result_0_3[0], result_0_3[1], result_0_3[2], result_0_3[3],
result_0_3[4],
      result_1_0[0], result_1_0[1], result_1_0[2], result_1_0[3],
result_1_0[4],
      result_1_1[0], result_1_1[1], result_1_1[2], result_1_1[3],
result_1_1[4],
      result_1_2[0], result_1_2[1], result_1_2[2], result_1_2[3],
result_1_2[4],
      result_1_3[0], result_1_3[1], result_1_3[2], result_1_3[3],
result_1_3[4],
      result_2_0[0], result_2_0[1], result_2_0[2], result_2_0[3],
result_2_0[4],
      result_2_1[0], result_2_1[1], result_2_1[2], result_2_1[3],
result_2_1[4],
      result_2_2[0], result_2_2[1], result_2_2[2], result_2_2[3],
result_2_2[4],
      result_2_3[0], result_2_3[1], result_2_3[2], result_2_3[3],
result_2_3[4]]

```

```

writer.writerow(row)
image_no += 1

f.close()

```

C. PCA and Machine Learning

```

#source: https://scikit-learn.org/stable/auto\_examples/classification/plot\_classifier\_comparison.htm
l#sphx-glr-auto-examples-classification-plot-classifier-comparison-py
# Code source: Gaël Varoquaux, Andreas Müller
# Modified for documentation by Jaques Grobler
# License: BSD 3 clause

```

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.inspection import DecisionBoundaryDisplay

names = [
    "Nearest Neighbors", "Linear SVM", "RBF SVM", "Gaussian Process",
    "Decision Tree", "Random Forest", "Neural Net", "AdaBoost", "Naive Bayes",
    "QDA",]
classifiers = [
    KNeighborsClassifier(3),
    SVC(kernel="linear", C=0.025),
    SVC(gamma=2, C=1),
    GaussianProcessClassifier(1.0 * RBF(1.0)),
    DecisionTreeClassifier(max_depth=5),
    RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1),
    MLPClassifier(alpha=1, max_iter=1000),
    AdaBoostClassifier(),
    GaussianNB(),
    QuadraticDiscriminantAnalysis(),]

```

```

input=pd.read_csv('pca.csv')
df=pd.DataFrame(input,columns=['component 1','component 2','output'])

X = df.iloc[:, :2].values
y = df.iloc[:, -1].values

datasets = [(X,y)]

figure = plt.figure(figsize=(99, 9))
i = 1
# iterate over datasets
for ds_cnt, ds in enumerate(datasets):
    # preprocess dataset, split into training and test part
    X, y = ds
    X = StandardScaler().fit_transform(X)
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.4, random_state=42
    )

    x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5
    y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5

    # just plot the dataset first
    cm = plt.cm.RdBu
    cm_bright = ListedColormap(["#FF0000", "#0000FF"])
    ax = plt.subplot(len(datasets), len(classifiers) + 1, i)
    if ds_cnt == 0:
        ax.set_title("Input data")
    # Plot the training points
    ax.scatter(X_train[:, 0], X_train[:, 1], c=y_train, cmap=cm_bright,
edgecolors="k")
    # Plot the testing points
    ax.scatter(
        X_test[:, 0], X_test[:, 1], c=y_test, cmap=cm_bright, alpha=0.6,
edgecolors="k"
    )
    ax.set_xlim(x_min, x_max)
    ax.set_ylim(y_min, y_max)
    ax.set_xticks(())
    ax.set_yticks(())
    i += 1

# iterate over classifiers
for name, clf in zip(names, classifiers):
    ax = plt.subplot(len(datasets), len(classifiers) + 1, i)
    clf.fit(X_train, y_train)
    score = clf.score(X_test, y_test)

```

```
DecisionBoundaryDisplay.from_estimator(  
    clf, X, cmap=cm, alpha=0.8, ax=ax, eps=0.5  
)  
  
# Plot the training points  
ax.scatter(  
    X_train[:, 0], X_train[:, 1], c=y_train, cmap=cm_bright,  
    edgecolors="k"  
)  
# Plot the testing points  
ax.scatter(  
    X_test[:, 0],  
    X_test[:, 1],  
    c=y_test,  
    cmap=cm_bright,  
    edgecolors="k",  
    alpha=0.6,  
)  
  
ax.set_xlim(x_min, x_max)  
ax.set_ylim(y_min, y_max)  
ax.set_xticks(())  
ax.set_yticks(())  
if ds_cnt == 0:  
    ax.set_title(name)  
ax.text(  
    x_max - 0.3,  
    y_min + 0.3,  
    ("%2f" % score).rstrip("0"),  
    size=15,  
    horizontalalignment="right",  
)  
i += 1  
  
plt.tight_layout()  
plt.show()
```

BIODATA PENULIS



Penulis bernama lengkap Gabriel Vieri lahir di Malang, Jawa Timur pada 3 Oktober 1999. Penulis menempuh pendidikan di SDK Cor Jesu Malang pada tahun 2005-2011. Penulis melanjutkan pendidikan di SMPK Santa Maria 1 Malang hingga lulus pada 2014. Setelah lulus pendidikan SMP, penulis melanjutkan studi menengah atas di SMAK Santa Maria Malang dan lulus pada 2017. Pendidikan sarjana ditempuh di Departemen Teknik Fisika, Institut Teknologi Sepuluh Nopember Surabaya. Selama menjadi mahasiswa penulis tergabung sebagai asisten laboratorium Vibrasi dan Akustik. Penulis juga memiliki pengalaman magang di PT. Maxzer Solusi Steel dan mengerjakan proyek "Desain dan Analisa Vibrasi Struktur Mesin Pelet Apung". Penulis dapat dihubungi melalui alamat email: gabrielvieri@protonmail.com.

Halaman ini sengaja dikosongkan