Pendeteksi Tingkat Kematangan Tomat dengan Menggunakan Naive Bayes dan K-Nearest Neighbor

Vijjasena 13517084
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
Vijjasena99@gmail.com

***Abstrak*—Tomat merupakan sayuran yang sering sekali dimakan dalam kehidupan sehari-hari, baik menjadi bagian makanan lain maupun dimakan secara langsung. Dalam membudidayakan tomat, pabrik makanan membutuhkan hasil panen yang banyak dan juga cara yang cepat untuk memanennya. Dengan teknologi yang semakin maju dan machine learning menjadi salah satu topik yang sering diperbincangkan, hal tersebut dapat membatu kecepatan proses panen di bidang pertanian. Pada makalah ini akan dibahas bagaimana model Naive Baayes dan K-Nearest Neighbor digunakan untuk memilah mana dari tomat yang sudah dipanen dapat dibedakan dari yang mentah dan yang matang.**

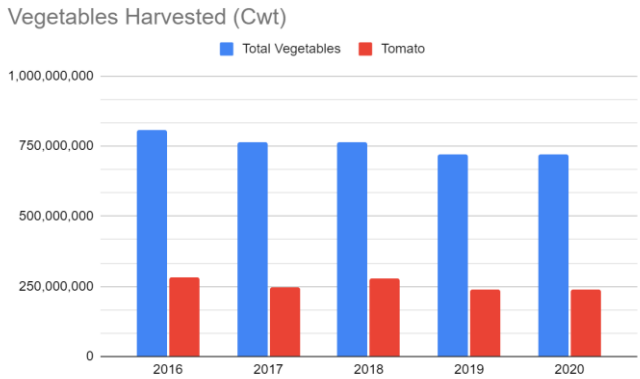
Kata Kunci—Klasifikasi; Naive Bayes; K-Nearest Neighbor; Tomat; Machine Learning

I. PENDAHULUAN

Tomat – mengidentifikasi tomat – peran *computer vision* – fitur utama yang digunakan dalam CV untuk identifikasi tomat (warna) – susunan warna – jenis jenis color spaces.

Tomat yang sering digolongkan sebagai sayuran memiliki nama latin *Lycopersicon esculentum.* Terdapat berbagai cara untuk menikmati tomat, seperti direbus bersamaan dengan sup atau dijadikan bahan utama sup, digoreng, ditumis sebagai penambah rasa makanan yang sedang dimasak tetapi tomat juga tidak kalah lezat ketika dimakan secara langsung. Referensi [9] menyatakan kalau awalnya tomat dibudidayakan di sekitar Chile, Colombia, Ecuador, Bolivia, dan Peru dengan jenis tomat yang dibudidayakan masih berupa tomat liar. Setelah tomat dibudidayakan hingga jenisnya hampir serupa dengan yang ada sekarang (lebih maju) barulah tomat masuk ke daerah Eropa. Sedangkan berdarkan [4], menyebutkan kalau tomat berasal dari daerah tersebut karena air pada daerah-daerah tersebut tidak telalu dingin dan juga tidak terlalu panas sehingga sesuai untuk membudidayakan tomat.

Produksi buah tomat pada 25 tahun belakangan ini berantung pada negara-negara seperti Amerika, Rusia, Turki, China, Mesir, dan Itali. Berdasarkan data yang disajikan oleh United States Department of Agriculture pada situsnya National Agricultural Statistics Service (USDA-NASS) produksi buah tomat sedniri cukup mendominasi total produksi sayur-sayuran yang ada di Amerika seperti yang dapat dilihat pada Tabel I.1 dibawah ini. Produksi buah tomat pada tahun 2016 mencapai 35% dari keseluruhan, tahun 2017 emncapai 32% dari keseluruhan, tahun 2018 merupakan produksi tertinggi dengna total 36% dari keseluruhan produksi sayuran di US, tahun 2019 dan 2020 mengalami penurunan menjadi 33% dari total keseluruhan sayur yang di produksi di US. Hal tersebut membuktikan kalau tomat sendiri merupakan tanaman yang mudah untuk diproduksi



Gambar I.1. Grafik Produksi Tomat Terhadap Produksi Sayuran

Tentu, untuk mendpatkan hasil panen yang baik dibutuhkan cara untuk memilah mana buah tomat yang baik dan mana yang belum matang maupun terdapat kecacatan pada buah tomat itu sendiri. Untuk mengatahaui hal tersebut pertama dapat dilihat dari bentuk dan ukuran dari buah tomat itu sendiri. Menurut [4] buah tomat yang diharapkan oleh konsumen memiliki bentuk yang bulat dengan lebih mementingkan lebar dari buah dibandingkan dengna panjang dari buah. Buah tomat yang diharapkan memiliki berat di sekitar 75 gram dengan jumlah lokul yang dimilikinya berjumlah setidaknya 3. Gambar I.2. berikut ini merupakan gambar perbandingan lokul yang dimiliki oleh buah tomat.

Gambar I.2. Bilokular (Kiri) dan Multilokular (Kanan)

Selain dari bentuk dan ukurannya, cara yang paling mudah untuk membedakan tomat adalah dengan fitur warna. Seperti yang telah diketahui, tomat memiliki warna merah ketika sudah

matang dan hijau ketika belum matang. Walapun terdapat beberapa jenis tomat yang memiliki warna jinga sampai kekuningan ketika sudah matang.

Berdasarkan data dari USDA-NASS pada Gambar I.1. mengenai total produksi tomat terhadap keseluruhan sayuran, insdustri makanan memerlukan cara memilah tomat yang sudah matang dalam cara yang cepat dan cukup akurat sehingga produksi parbik dapat memenuhi permintaan dari konsumen. Untuk menyelesaikan hal tersebut banyak jalur produksi pada pabrik telah mengimplementasikan *computer vision* untuk membantu memilah mana tomat yang sudah matang dan mana yang belum. Teknologi computer vision sendiri menggunakan warna sebagai fitur utama yang membedakan.

Warna, dalam penggunaannya terdapat beberapa jenis yang dapat digunakan. Penggunaan *colour spaces* bergantung dengan jenis penggunaannya, seperti contoh, *colour space cyan*, *yellow*, *magenta*, *black* (CYMK) cocok untuk digunakan dalam *hardware processing* seperti mencetak gambar, menyimpan gambar, dan lain sebagainya. Menurut [13] ada 3 jenis dari *colour spaces*, yaitu:

1. *Hardware-oriented spaces*

   Colour space dengan jenis hardware-oriented, seperti namanya, diperuntukan untuk hardware processing, seperti contoh penyimpanan gambar, pengambilan gambar, dan penampilan gambar. *Colour space* jenis ini dapat memperlihatkan perubahan warna yang sangat kecil sehingga cocok untuk mengvaluasi bagaimana warna dari makanan berubah pada saat produksi. Contoh dari *colour space* jenis ini adalah CYMK (*cyan*, magenta, *yellow*, *black*), RGB (*red*, *green*, *blue*), dan YIQ (*luminance*, *in-phase*, *quadrature*). Dari ketiga jenis tersebut, RGB merupakan *colour space* yang baik digunakan untuk menangkap pemandangan natural dan memperlihatkan bagaimana cara dari fosfor bekerja. *Colour space* RGB ditentukan berdasarkan titik pada tiga sumbu yaitu sumbu merah, hijau, dan biru seperti yang diperlihatkan pada Gambar I.3. mengenai representasi coulour space RGB Untuk YIQ dan CYMK lebih baik digunakan untuk menampilkan gambar pada televisi, percetakan. Oleh karena itu, YIQ dan CYMK tidak digunakan untuk mengevaluasi bagaimana perubahan warna pada makanan di saat produksi.

2. *Human-oriented spaces*

   Jenis *colour space* ini mengacu pada konsep hue, shade, dan tone. Ketiga elemen penyusun tersebut berasal dari katakteristik warna yang intuitif. Untuk itu seluruh jenis dari *colour space* ini berdasarkan *hue* dan *saturation* (HS). Contoh dari *colour space* ini adalah HSI (*hue, saturation, intensity*), HSV (*hue*, *saturation, value*), HSB (*hue, saturation, brightness*). Pada *colour space* ini, elemen *hue* merupakan atribut dari sensasi visual mengenai bagaimana sebuah area terlihat oleh mata mengacu pada warna. Warna yang terlihat dapat berupa merhah, kuning, hijau, biru, atau kombinasi dari warna-warna tersebut. *Saturation* merupakan elemen yang menunjukan seberapa hidup warna tersebut mengacu kepada pencahayaan yang dimilikinya. Sementara *brightness, value, dan intensity* merupakan seberapa warna tersebut tercampur dengan warna putih atau dengan kata lain seberapa warna tersebut dapat memantulkan cahaya. Untuk dapat lebih baik memahami *colour space* ini dapat melihat ilustrasi yang ditampilkan pada Gambar I.4 mengenai representasi *colour space* HSV. Sesuai dengan namanya, *human-oriented spaces*, *colour spaces* ini memiliki performa yang lebih baik untuk mengenali warna pada makanan karena representasi warnanya berdasarkan bagaimana manusia melihat warna pada dunia nyata dan bukan pada layar (dunia maya). Namun, *colour spaces* ini memiliki kelemahan yaitu kurangnya sensitifitas terhadap perubahan warna yang kecil karena mengacu kepada penglihatan manusia.

3. *Instrumental spaces*

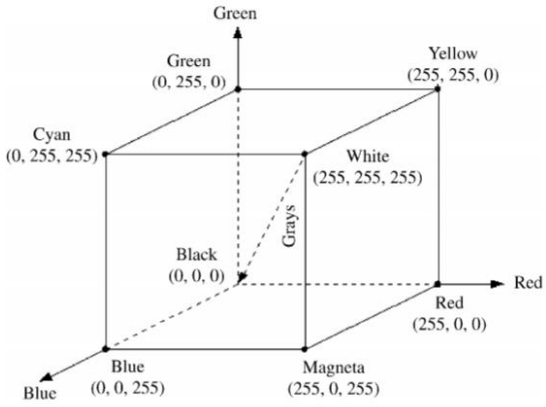
   Seperti namanya, *instrumental spaces* digunakan untuk menidentifikasi warna pada instrumen. Sebagian besar dari *colour spaces* dari jenis ini distandarisasi oleh Commission Internationale d'Eclairage (CIE) yang merupakan sebuah organisasi technical, scientific, dan cultural bersifat non-profit untuk melakukan hal-hal seperti standardisasi, prosedur, pentunjuk, dan hal lainnya. Contoh dari jenis *colour space* ini adalah CIE XYZ dan CIELAB. CIE XYZ adalah *colour space* yang dirancang berdasarkan persepsi psikologi terhadap cahaya. Kalkulasi warna pada *colour space* ini adalah menggunakan Standard Observer yang berhubungan terhadap sel kerucut untuk merespon warna merah, hijau, dan biru pada mata. Perancangan *colour space* ini bertujuan untuk menyelesaikan masalah hanya merangsang salah satu sel kerucut pada mata. XYZ adalah perwakilan dari lightness (Y) dan komponen vitural yang menyerupai kurva kerucut dari warna merah dan biru (X dan Z). Selain dari *colour space* XYZ terdapat CIELAB yang memiliki elemen L*, a*, b*. Atribut L* merupakan tingkan penerangan pada warna yang dievaluasi, sedangkan a8 dan b* merupakan representasi warna merah, hijau, biru, dan kuning. CIELAB sering digunakan untuk mengevaluasi makanan karena memiliki distribusi wanra yang lebih merata (uniform).

Setelah mengetahui definisi dan colour spaces yang dapat digunakan untuk membantu memilah tomat pada computer vision, diperlukan representasi yang lebih definitif agar mesin dapat mengenali bahwa tomat yang sedang dievaluasi merupakan tomat berwarna merah dan sudah matang. Pada Tabel I.1. dapat dilihat rentang nilai untuk dapat merepresentasikan warna pada beberapa *colour spaces* yang baik dalam mengevaluasi makanan.

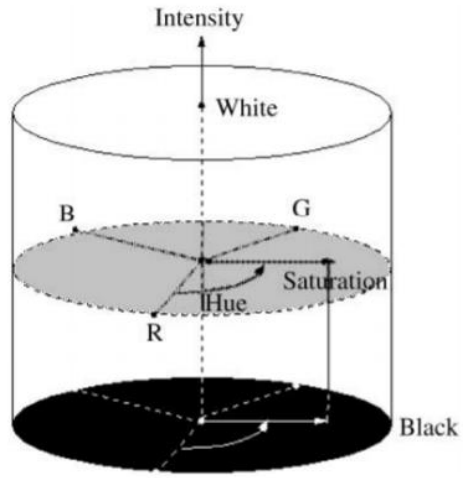
Tabel I.1. Rentang Nilai dari Parameter *Colour Spaces*

Jenis	Parameter	Deskripsi	Nilai
RGB	R	Nilai warna merah	[0,255]
	G	Nilai warna hijau	[0,255]

	B	Nilai warna biru	[0,255]
HSV	H	Nilai *hue*	[0,360]
	S	Nilai *Saturation*	[0,100]
	V	Nilai *value*	[0,100]
L*a*b*	L*	Nilai penerangan	[0,100]
	a*	Nilai warna merah/hijau	[-128,127]
	b*	Nilai warna biru/kuning	[-128,127]



Gambar I.3. Representasi RGB *Colour Space* **(Cheng, Jiang, Sun, & Wang, 2001)**



Gambar I.4. Representasi HSI *Colour Space* **(Cheng, Jiang, Sun, & Wang, 2001)**

Computer vision yang digunakan dibantu dengan model yang dihasilkan machine learning. Machine learning kemudian menerapkan metode pattern recognition pada modelnya. Pattern recognition sendiri terdapat dua jenis, pertama adalah clustering dan kedua adalah klasifikasi. Kedua metode tersebut dapat diterapkan untuk membantu mengenali tomat yang matang dan yang mentah berdasarkan warnanya. Namun dari kedua metode tersebut, klasifikasi merupakan metode yang lebih populer untuk digunakan dalam permasalahan ini. Metode yang sering digunakan untuk memilah tomat adalah dengan klasifikasi atau yang lebih baik disbut dengan supervised learning. Terdapat beberapa jenis metode dalam supervised learning, sebagai contoh adalah, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), Naive Bayes (NB), dan lainnya.

Penggunaan computer vision sudah pernah diterapkan untuk mengevaluasi berbagaimacam buah atupun sayuran, Beredasarkan [1] Tabel I.2. menampilkan hasil akurasi yang didapatkan dengan menggunakan metode tertentu untuk melakukan klasifikasi.

Tabel I.2. Nilai Akurasi Penilaian Objek dengan Metode Tertentu

Objek	Colour Space	Metode	Akurasi
Apel	HSI	SVM	95
	L*a*b*	MDA	100
Aplukat	RGB	K-Means	82.22
Blueberry	RGB	KNN, SK-Means	85-98
Jeruk Limau	RGB	ANN	100
Kelapa Sawit	L*a*b*	ANN	91.67
Kesemak	RGB dan L*a*b*	QDA	90.24
Kurma	RGB	K-Means	99.6
Mangga	RGB	SVM	96
	L*a*b*	MDA	90
	L*a*b* dan HSB	LS-SVM	88
Pepper	HSV	SVM	93.89
Pisang	L*a*b*	LDA	98
	RGB	AN	96
Semangka	YCbCr	ANN	86.51
Tomat	HSV	SVM	90.8
	RGB	DT	94.29
	RGB	LDA	81
	L*a*b*	ANN	96

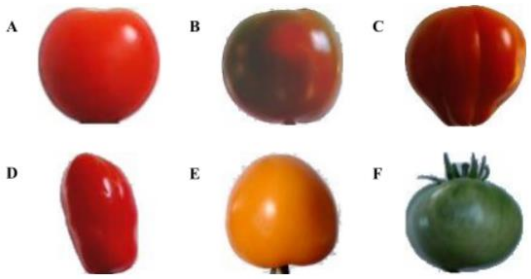
Berdasarkan tabel tersebut, makalah ini akan mencoba untuk mengevaluasi tomat dengan menggunakan colour space HSV dan juga RGB dengan metode KNN dan juga NB agar data dari tabel tersebut dapat ditambahkan dan dijadikan referensi penulisan selanjutnya.

II. Bahan dan Metode Rancangan Solusi

A. Sampel Tomat

Sampel tomat yang akan digunakan untuk menyusun solusi diambil dari gambar digital, tidak diambil secara langsung melalui kamera. Untuk *dataset* dari gambar tomat baik matang maupun mentah diambil dari salah satu halaman pada situs kaggle (https://www.kaggle.com/moltean/fruits). Pada *dataset* terebut disediakan beberapa jenis tomat dengan beberapa warna, seperti jingga atau merah pada umumnya untuk tomat

yang sudah matang. Gambar II.1 merupakan ilustrasi warna dari beberapa jenis tomat yang akan digunakan.



Gambar II.1. Sampel Gambar Perancangan Model, Tomat Matang (A-E), Tomat Mentah (F).

B. Metodologi

Berdasarkan [1] proses untuk merancang solusi dapat dibagi menjadi 3 tahapan utama, yaitu, ekstraksi fitur dari sampel, perancangan model, dan yang terakhir adalah evaluasi model. Berikut ini merupakan deskripsi proses pengembangan secara detail.

1) Ekstraksi Fitur dari Sampel

Pada tahapan ini sampel tomat akan diambil informasi mengenai warna yang dimilikinya. Namun, sebelum mengambil warna dari gambar perlu dilakukan beberapa langkah *image processing* terlebih dahulu agar nilai warna dari gambar tidak bias. Berikut ini tahapan dari pengambilan warna tomat dari gambar.

- *Image enhancement.* Sampel gambar yang akan digunakan untuk melatih model dari *machine learning* akan disesuaikan terlebih dahulu untuk memperkecil gangguan dari pencahayaan ketika gambar tersebut diambil. Untuk proses ini digunakan fungsi yang sudah ada pada kakas Matlab, yaitu *imadjust*. Fungsi tersebut digunakan untuk menyesuaikan nilai intensitas dari gambar masukan berdasarkan nilai yang dimiliki gambar tersebut pada jenis *grayscale*. Hasil perubahan tersebut dapat dilihat pada Gambar II.2(A).
- *Segmentation*. Setelah sampel gambar disesuaikan nilai intensitas dengan menggunakan *imadjust*, kemudian gambar akan disegmentasi agar ketika sampling, warna dari latar belakang tidak mengganggu dan menyebabkan bias. Untuk itu tahapan yang perlu diterapkan pada sampel adalah pertama gambar akan diubah menjadi jenis *grayscale* untuk dilakukan proses *thresholding*. Untuk menemukan threshold dari gambar, digunakan metode otsu yang diperlihatkan pada persamaan (1)

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad (1)$$

$$\sigma_B^2(k) = \frac{(m_G P_1(k) - m(k))^2}{P_1(k)\,(1 - P_1(k))} \quad (2)$$

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 \cdot P_i \quad (3)$$

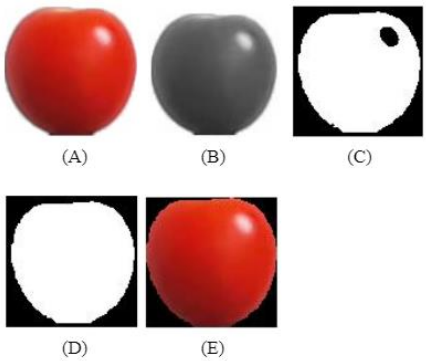
Dimana

$\eta(k)$ = *nilai threshold*

$\sigma_B^2(k)$ = *variansi di antara kelas*

σ_G^2 = *variansi global*

Hasil dari operasi *thresholding* tersebut dapat dilihat pada Gambar II.2(B). Hasil dari *thresholding* tersebut kemudian akan diolah lebih lanjut dengan menutupi bagian dari tomat yang tidak terambil menggunakan gambar bulatan kecil. Proses tersebut menggunakan fungsi yang disediakan oleh Matlab, yaitu, *imfill*. Hasil gambar tersebut kemudian dijadikan filter untuk memilih hanya tomat dari gambar tersebut seperti yang ditunjukkan pada Gambar II.2(C).



Gambar II.2 Hasil Image Enhancement dan Segmentation

- *Colour extraction*. Proses ini menggunakan fungsi yang sudah disediakan oleh Matlab untuk mengubah *colour space* dari gambar sampel yang ingin diolah. Pada mulanya, gambar masukan memiliki *colour space* RGB dan digunakan fungsi *rgb2hsv* untuk mengubah gambar menjadi *colour space* HSV. Untuk mengakses nilai-nilai dari RGB dan juga HSV dapat digunakan operasi matriks pada Matlab.

2) Perancangan Model

Untuk merancang model pemilah tomat, digunakan dataset yang berisi berbagai macam jenis tomat seperti yang dapat dilihat pada Gambar II.1. mengenai jenis gambar tomat yang digunakan untuk melatih model. Dalam *dataset* tersebut terdapat 3.945 gambar untuk tomat yang matang dan 475 gambar untuk tomat yang tidak matang. Untuk melakukan pengujian terhadap model telah disiapkan 1935 gambar untuk tomat matang dan 156 untuk tomat yang tidak matang.

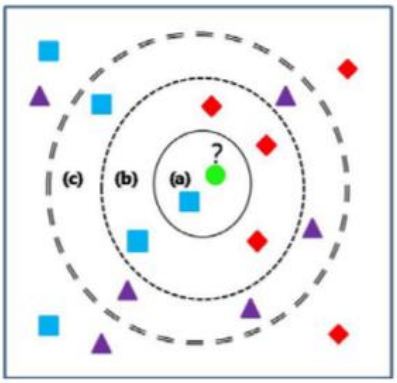
Model nantinya akan dibuat berdasarkan metode *supervised learning*. Mata metode ini akan dibuat data dari gambar untuk melatih model yang berisikan data-data dari *colour space* berserta label dari data tersebut. Untuk data tomat yang matang diberikan label 'R' dan yang belum matang akan digunakan label 'U'. Setelah gambar-gambar untuk melatih model disiapkan maka akan dibuat model berdasarkan metode-metode *supervised learning*.

a) K-Nearest Neighbor (KNN)

KNN merupakan salah satu metode klasifikasi yang mengklasifikasikan sebuah *instance* dari *dataset* yang ingin diklasifikasikan kepada data yang berada di dekatnya (*nearest neighbor*) [3]. Untuk mengetahui jarak dari satu *instance* ke *instance* lainnya dapat digunakan berbagai cara, salah satunya adalah *euclidean distance* [5], dengan persamaannya sebagai berikut.

$$d(x,y) = \left(\sum_{i=1}^{m}((x_i - y_i)^2)\right)^{1/2} \quad (4)$$

Meskipun *euclidean distance* merupakan perhitungan jarak yang paling umum tidak menutup kemungkinan untuk menggunakan perhitungan jarak yang lain seperti *manhattan distance*. Untuk memahami metode ini lebih baik, Gambar II.3 menampilkan ilustrasi untuk algoritma KNN.



Gambar II.3 K-Nearest Neighbor

Dalam penerapannya untuk membangun model digunakan Matlab Machine Learning Toolkit. Fungsi yang digunakan untuk membangun model adalah *fitcknn* dengan nilai k yang ditentukan adalah 2.

b) Naive Bayes (NB)

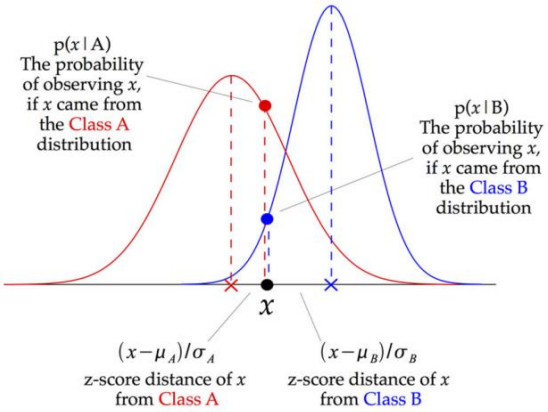
Menurut (Rish, 2001, August), Naive Bayes adalah sebuah *classifier* yang akan memberikan kelas yang paling cocol sesuai dengan deskripsi yang diberikan oleh vektor fitur. Persamaan berikut merupakan asumsi yang digunakan, yang mana fitur yang ada bersifat independen diberikan kelasnya.

$$P(X|C) = \prod_{i=1}^{n} P(X_i|C) \quad (5)$$

Diberikan X = (X_1 , ... , X_n) merupakan vektor dari fitur dan C adalah kelas. Sementara untuk teorema *bayes* sendiri adalah sebagai berikut [7].

$$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)} \quad (6)$$

Dengan ketentuan P(h) adalah peluang terjadinya kejadian h. P(h) biasa disebut juga dengan peluang awal dari h. Begitu juga dengan P(D) yang merupakan peluang awal dari kejadian D. Terdapat beberapa jenis metode *naive bayes*, yang akan digunakan pada perancangan model nantinya adalah metode *gaussian naive* bayes yang sudah disediakan oleh kakas Matlab. Fungsi yang digunakan adalah *fitcnb* dengan mode normal yaitu gaussian naive bayes. Gambar II.4 menampilkan bagaimana cara kerja dai algoritma gaussian naive bayes.



Gambar II.4 Cara Kerja Gaussian Naive Bayes (Raizada & Lee, 2013)

Dalam penerapannya model diberikan matriks yang berisi data mengenai warna dari sampel tomat dalam *colour space* RGB dan juga HSV, bersamaan dengan jumlah kelas yang ada.

3) Evaluasi Model

Setelah model berhasil dirancang, maka akan dilakukan prediksi terhadap *dataset* untuk melakukan pengujian. Untuk melakukan evaluasi terhadap hasil prediksi yang dilakukan oleh model, digunakan *confusion matrix*. Matriks ini memiliki dimensi *true class* dan *predicted class*. *True class* merupakan kelas yang seharusnya dimiliki oleh tomat tersebut dan dalam *confussion matrix* diwakili dengan baris dari matriks tersebut. *Predicted class* merupakan kelas tomat dari hasil prediksi yang dilakukan model. Kelas tersebut diwakili oleh kolom dari *confussion matrix*. Tabel II.1 berikut merupakan ilustrasi dari confussion matrix dengan beberapa istilahnya.

Tabel II.1. Confussion Matrix

	Predicted Positive	*Predicted Negative*
Actual Positive	A	B
Actual Negative	C	D

Dari Tabel II.1 tersebut, A merupakan jumlah prediksi data yang benar dan dari kelas *positive*. B merupakan jumlah prediksi data yang salah dan dari kelas *negative*. C merupakan jumlah prediksi data yang salah dan dari kelas *positive*. D merupakan jumlah prediksi data yang salah dan dari kelas

negative. Terdapat beberapa pengukuran yang dapat membantu dalam mengevaluasi hasil prediksi dari model yang sudah dirancang, di antaranya adalah *accuracy*, *recall*, dan juga *f-measure*. Pengukuran ini dapat dihitung dengan informasi yang ditampilkan oleh *confussion matrix* karena di dalamnya terdapat istilah *true positive* (TP), *true negative* (TN), *false positive* (FP), dan juga *false negative* (FN). Agar dapat lebih memahami pengukuran yang akan dilakukan terhadap model diperlukan pengertian dari keempat istilah ini, berikut ini penjelasannya [11].

- *True Positive*. Merupakan hasil dari prediksi model ketika model berhasil untuk memprediksi kelas positif. Peluang terjadinya disebut *true positive rate* dengan cara perhitungannya berdasarkan persamaan berikut.

$$TP = \frac{A}{A+B} \quad (7)$$

- *True Negative*. Merupakan hasil dari prediksi model ketika model berhasil memprediksi kelas negatif. Peluang terjadinya TN disebut dengan *true negative rate*. Cara perhitungannya berdasarkan persamaan berikut.

$$TN = \frac{D}{C+D} \quad (8)$$

- *False Positive*. Merupakan hasil dari prediksi ketika model gagal untuk memprediksi data dari kelas positif. Peluang terjadinya disebut dengan *false positive rate* dengan perhitungannya berdasarkan persamaan berikut.

$$FP = \frac{C}{C+D} \quad (9)$$

- *False Negative*. Merupakan hasil prediksi ketika model gagal untuk memprediksi data dari kelas negatif. Peluang terjadinya disebut dengan *false negative rate* dengan perhitungannya berdasarkan persamaan berikut.

$$FN = \frac{B}{A+B} \quad (10)$$

Setelah mengetahui istilah yang akan digunakan untuk penyusunan dari metode pengukuran, maka berikut ini merupakan penjelasan lebih detail untuk masing-masing pengukuran beserta persamaan yang digunakan untuk menghitung pengukuran tersebut.

- *Accuracy*. Akurasi merupakan nilai yang menunjukkan total dari jumlah prediksi yang benar. Berikut ini merupakan persamaan yang digunakan untuk menghitung akurasi dari model berdasarkan *confussion matrix*.

$$Accuracy = \frac{A+D}{A+B+C+D} \quad (11)$$

Dapat dilihat dari persamaan (11) dan juga *confussion matrix*, akurasi merupakan nilai dimana model hasil positif untuk kelas yang positif dijumlah dengan hasil negatif untuk kelas yang negatif untuk kemudian dibagi oleh kelseluruhan data yang ada

- *Recall*. Merupakan nilai yang dihasilkan oleh model ketika memprediksi kelas positif dari data dan mendapatkan nilai yang benar. Untuk persamaan yang digunakan untuk menghitung nilai recall adalah sama dengan persamaan (7) yang digunakan untuk menghitung *true positive rate*.

- *Precision*. Merupakan jumlah dari prediksi positif dari total keseluruhan kelas positif yang diprediksikan oleh model. Berikut ini merupakan persamaan yang digunakan untuk mengkalkukasi *precision*.

$$Precision = \frac{D}{B+D} \quad (12)$$

- *F-measure*. Merupakan nilai yang menggambarkan keseimbangan antara *precision* dan juga *recall*. F-measure juga sering disebut dengan F1 *score*, berikut ini merupakan persamaan yang digunakan untuk menghitung nilai *f-measure* [6].

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (13)$$

Persamaan (13) dapat dibaca sebagai harmonic mean dari prescirion dan juga recal. Persamaan (13) juga merupakan penyederhanaan dan dapat digeneraliasi menjadi persamaan (14) sebagai berikut.

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} \quad (14)$$

Nilai dari *f-measure* merupakan pengukuran yang paling signifikan karena dapat menarik kesipulan dari nilai yang didapatkan pada *precision* dan juga *recall*.

III. Aplikasi dan Hasil Prediksi Model

Setelah ditancangnya model untuk memilah tomat dengan menggunakan metode Gaussian Naive Bayes dan juga K-Nearest Neighbor. Dilakukan perancangan aplikasi agar pengguna dapat melakukan klasifikasi gambar tomat dengan lebih mudah.

1) Aplikasi

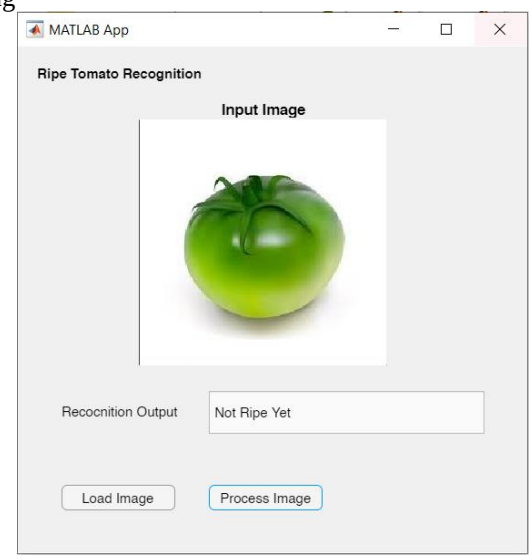
Perancangan aplikasi ini menggunakan Matlab App Designer. Kakas tersebut sudah menyedikana beberapa fungsi awal untuk dapat menjalankan sebuah layar kosong tetapi terdapat beberapa jenis pilihan untuk memulai *project*. Dengan menggunakan kakas tersebut tentu bahasa yang digunakan merupakan bahasa Matlab dan juga terdapat beberapa kode yang menggunakan bahasa C atau C++. Berikut ini merupakan tampilan awal dari aplikasi yang sudah dirancang.

Gambar III.1. Tampilan Hasil Aplikasi

Aplikasi memiliki tombol kontrol yang sederhana. Terdapat dua tombol yang dapat diguakan oleh pengguna yaitu "*Load image*" yang digunakan untuk memilih gambar tomat yang ingin diprediksi oleh aplikasi dan yang kedua adalah tombol "*Process Image*" yang digunakan untuk memulai proses prediksi oleh model yang telah dirancang.

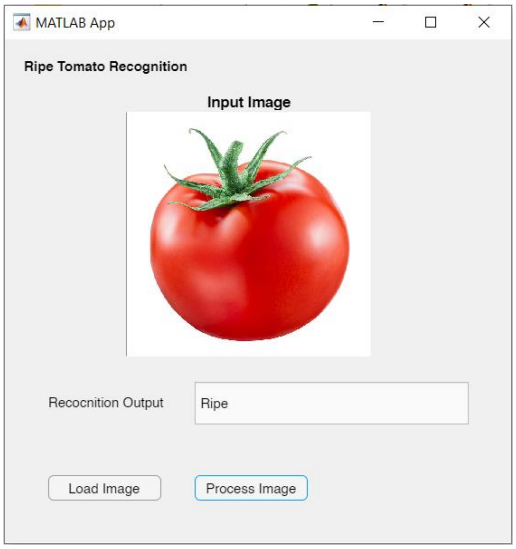
2) Hasil Prediksi dan Statistik

Berikut ini merupakah hasil prediksi dari aplikasi, masing-masing untuk tomat yang sudah matang dan yang belum matang



Gambar III.2. Hasil Prediksi Tomat Belum Matang

Dapat dilihat pada Gambar III.2 untuk hasil prediksi pada tomat yang belum matang. Pada aplikasi terdapat hasil "Not Ripe Yet" di kolom recognition output. Pada Gambar III.3 ini merupakan hasil prediksi pada tomat yang sudah matang. Pada kolom recognition output terdapat kata "Ripe" untuk tomat yang sudah matang.



Gambar III.3. Hasil Prediksi Tomat Matang

Untuk hasil statistik dari model yang sudah dirancang tidak ditampilkan pada aplikasi sendiri melainkan menggunakan *script* tersendiri untuk dapat mengaksesnya. Hal ini dilakukan agar tidak membingungkan pengguna ketika ingin menggunakan aplikasi. Berikut ini merupakan hasil *cofussion matrix* untuk model *Gaussian Naive Bayes* (G-NB) dan juga KNN yang dilakukan pada *dataset* uji.

Tabel III.1. Confussion Matrix dari Model Gaussian Naive Bayes

G-NB	R	U
R	1342	54
U	0	158

Tabel III.2. Confussion Matrix dari Model K-Nearest Neighbor

KNN	R	U
R	1343	53
U	5	153

Dari *confussion matrix* tersebut kelas R merupakan kelas positif yang telah ditentukan. Berdasarkan data yang sudah ada dapat dihitung nilai-nilai untuk kedua model. Tabel III.3 merupakan hasil perhitungan untuk model G-NB dan KNN.

Tabel III.3 Hasil Statisti untuk Model Gussian Naive Bayes dan KNN

Gaussian Naive Bayes	
Pengukuran	Nilai
TP	0,961
FP	0
TN	1
FN	0,039
Accuracy	0,965
Precision	0,745
Recall	0,961
F1 Score	0,840

K-Nearest Neighbor	
Pegukuran	Nilai
TP	0,962
FP	0,032
TN	0,968
FN	0,038
Accuracy	0,963
Precision	0,743
Recall	0,962
F1 Score	0,838

IV. KESIMPULAN

Setelah model dan aplikasi selesai dirancang dapat dilihat hasil yang diberikan oleh model cukup memuaskan dengan nilai akurasi sebesar 0,965 dan nilai f1 *score* 0,840. Nilai f1 score tersebut mengartikan kalau model yang dirancang sudah cukup baik. Untuk itu dapat dikatakan kalau model KNN dan juga Naive Bayes dapat dikembangkan lebih lanjut untuk dapat menjadi model pembeda tomat yang lebih kompleks. Untuk selanjutnya dapat ditambahkan variasi warna dari sampel yang digunakan untuk dapat mewakili pertumbuhan tomat yang sebenarnya sehingga alat untuk memilah tomat bekerja lebih baik.

VIDEO LINK AT YOUTUBE

Berikut ini merupakan *link* video presentasi aplikasi yang telah dirancang.

http://bit.ly/PresentasiAplikasiMakalah

REFERENCES

[1] Castro, W., Oblitas, J., De-La-Torre, M., Cotrina, C., Bazán, K., & Avila-George, H. (2019). Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces. *IEEE Access*, 7, 27389-27400.

[2] Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern recognition*, 34(12), 2259-2281.

[3] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21-27.

[4] Hobson, G., & Grierson, D. (1993). *Tomato. In Biochemistry of fruit ripening (pp. 405-442).* Springer, Dordrecht.

[5] Kim¹, J. I., Kim, B. S., & Savarese, S. (2012). Comparing image classification methods: K-nearest-neighbor and support-vector-machines. *In Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics (Vol. 1001, pp. 48109-2122)*.

[6] Maratea, A., Petrosino, A., & Manzo, M. (2014). Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, 257, 331-341.

[7] Mitchell, T. M. (1997). *Machine learning.*

[8] Raizada, R. D., & Lee, Y. S. (2013). Smoothness without smoothing: why Gaussian naive Bayes is not naive for multi-subject searchlight studies. *PloS one*, 8(7), e69566.

[9] Rick, C. M. (1978). The Tomato. *Scientific American*, 239(2), 76-89.

[10] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. *In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46)*.

[11] Santra, A. K., & Christy, C. J. (2012). Genetic algorithm and confusion matrix for document clustering. *International Journal of Computer Science Issues (IJCSI)*, 9(1), 322.

[12] Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion Matrix-based Feature Selection. *MAICS*, 710, 120-127.

[13] Wu, D., & Sun, D. W. (2013). Colour measurements by computer vision for food quality control–A review. *Trends in Food Science & Technology*, 29(1), 5-20.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Mei 2021

Vijjasena / 13517084