

## APLIKASI PENDETEKSIAN KEMATANGAN TOMAT

**Moch Azhar Al Ghifari, Agung Panji Sasmito, Dedy Rudhistiar**

Program Studi Teknik Informatika S1, Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia  
*paijoghifari@gmail.com*

### ABSTRAK

Tomat merupakan tumbuhan yang pertama kali ditemukan di Amerika Selatan, masih berkerabat dengan terung, kentang dan paprika. Tomat merupakan salah satu komoditas andalan Kota Batu, sebelumnya ada 18 wilayah di Kota Batu yang telah tersertifikasi sebagai pertanian organik. Pada tahun 2018, tomat yang dihasilkan sebesar 130.298Kw dan total petani yang bekerja sebanyak 278, sedangkan untuk tahun 2019 total tomat yang dihasilkan sebesar 153.622Kw dan jumlah total Petani sebanyak 342. Salah satu hal yang sering dialami petani tomat adalah jumlah buah. saat panen, petani harus memastikan buahnya bagus sampai sampai ke konsumen. Kualitas tomat dapat dilihat dari warna kulit tomat. Tomat yang masih segar dan matang biasanya berwarna merah cerah, tomat yang belum matang akan berwarna hijau ke kuningan, dan tomat mentah berwarna hijau. Namun petani hanya melihat tomat tersebut dari segi warnanya saja, padahal tidak semua tomat itu isinya sesuai dengan warna kulit luarnya. Berdasarkan hasil pengujian, diketahui bahwa sistem dapat menampilkan keterangan tomat berdasarkan nilai RGBnya dengan perhitungan thresholding dan diklasifikasikan menggunakan algoritma KNN sehingga sistem dapat mengetahui tingkat kematangan tomat yang diuji.

**Kata kunci:** Kematangan, Tomat, KNN, Thresholding, RGB

### 1. PENDAHULUAN

Tomat merupakan tumbuhan yang pertama kali ditemukan di Amerika Selatan, masih berkerabat dengan terung, kentang dan paprika [1]. Tomat merupakan salah satu komoditas andalan Kota Batu, sebelumnya ada 18 wilayah di Kota Batu yang telah tersertifikasi sebagai pertanian organik. Untuk Kecamatan Junrejo meliputi Desa Beji, Dadaprejo, Junrejo, Mojorejo, Pendem, Tlekung, dan Desa Torongrejo. Kecamatan Bumiaji meliputi Desa Sumberbrantas, Tulungrejo, Sumbergondo, Bulukerto, Gunungsari, Pandanrejo, dan Giripurno. Adapun dari lahan pertanian organik tersebut, produk yang dihasilkan meliputi padi, kale, sawi, kangkung, jambu kristal, kopi, tomat, terong, kubis, kentang, paprika, dan seledri [2]. Hal ini ditunjukkan dengan data produksi buah tomat di tahun 2018-2019 dan banyaknya petani tomat di Kota Batu.

Pada tahun 2018, untuk kecamatan Batu menghasilkan 8714Kw tomat dari 26 Petani, untuk Junrejo sebesar 21.536Kw dari 20 Petani, untuk Bumiaji 34,899Kw dari 93 Petani dan di Kota Batu sendiri sebesar 65.149Kw dari 139 Petani. Sedangkan untuk tahun 2019, untuk kecamatan Batu menghasilkan 10,115Kw tomat dari 26 Petani, untuk Junrejo sebesar 34,390Kw dari 26 Petani, untuk Bumiaji 32,306Kw dari 100 Petani dan di Kota Batu sendiri sebesar 79,811Kw dari 171 Petani. Total tomat yang dihasilkan di tahun 2018 sebesar 130.298Kw dan total petani yang bekerja sebanyak 278, sedangkan untuk tahun 2019 total tomat yang dihasilkan sebesar 153.622Kw dan jumlah total Petani sebanyak 342 [13].

Salah satu hal yang paling sering dialami oleh petani tomat adalah banyaknya buah pada masa panen, Petani harus memastikan buahnya baik hingga ke tangan konsumen. Kualitas tomat dapat dilihat dari warna kulit tomat dan ukuran buah tomat. Tomat yang masih segar dan matang biasanya berwarna merah cerah dan tentunya dengan ukuran yang besar. Namun, petani hanya menyortir tomat berdasarkan warna dan seringkali mengabaikan ukuran tomat itu sendiri. Padahal ukuran tomat yang besar dan kecil juga sangat berpengaruh terhadap penjualan di pasaran.

Saat ini teknologi informasi telah berkembang hingga perkembangan teknologi pengolahan citra telah banyak diteliti oleh berbagai peneliti Penerapan teknik tersebut pada berbagai bidang telah memungkinkan kita untuk dapat menilai, menyimpulkan dan mendapatkan suatu informasi pada suatu citra digital. Dengan menggunakan teknik pengolah citra, pengukuran berbagai parameter benang yang membutuhkan penilaian secara visual dapat dikuantifikasikan [3]. Salah satu penerapan teknologi pengolahan citra adalah *image processing*. *Image processing* adalah suatu bentuk pengolahan atau pemrosesan sinyal dengan *input* berupa gambar (*image*) dan ditransformasikan menjadi gambar lain sebagai hasil pengeluaran dengan teknik tertentu. *Image processing* dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar [4]. Teknik ini

berguna untuk berbagai bidang seperti, komputer, kedokteran, industri, dan lain-lain [5]. Contoh hasil yang diterapkan dalam kehidupan seperti penelitian membuat aplikasi *sorting* buah tomat.

Aplikasi pengolahan citra mempermudah penggunaanya dalam pengenalan pola yang berperan dalam memisahkan objek dari latar belakang secara otomatis. Selanjutnya, objek akan diproses oleh pengklasifikasi objek. Selain itu pengolahan citra berperan untuk mengenali bentuk-bentuk khusus yang dilihat oleh mesin sehingga mempermudah dalam mengenali suatu objek. Perangkat lunak untuk mengolah citra digital juga sangat populer dalam perkembangannya, digunakan oleh pengguna untuk mendeteksi, mengolah foto atau untuk berbagai keperluan lain sebagai contoh, *Adobe Photoshop* dan *GIMP (GNU Image Manipulation Program)* yang menyajikan berbagai fitur dalam memanipulasi citra digital.

Berdasarkan paparan tersebut diatas, penulis tergagas untuk membuat aplikasi pengolahan citra untuk menentukan kematangan tomat dengan metode *image processing*. Aplikasi ini dikembangkan berbasis Matlab dan mempunyai beberapa fitur yaitu, menampilkan citra gambar asli, menampilkan gambar citra biner, menampilkan nilai ciri RGB pada table dan menampilkan keterangan hasil klasifikasi. Dengan adanya aplikasi ini diharapkan dapat membantu para petani dalam memilih buah tomat yang segar dan sehat sehingga bisa dikonsumsi oleh masyarakat sekitar.

## 2. TINJAUAN PUSTAKA

### 2.1. Tanaman Tomat

Tomat termasuk dalam kategori tanaman Solanacea atau terong-terongan yang dicirikan yaitu batang dan daunnya yang berbulu halus atau kasar. Umumnya tomat berdaun lebar dan ada juga yang terdapat pada tangkai bersama (majemuk). Tanaman berakar tunggang dengan banyak akar samping. Sayuran solanaceae meliputi beberapa genus yang penting seperti tomat, kentang, terong, cabai dan ranti (*leunca*) serta tekokak[6]. Buah tomat yang sudah matang dapat dipanen dan dikonsumsi, baik diolah maupun dimakan langsung.

### 2.2. Kota Batu

Kota Batu terletak di ketinggian rata-rata 862 MDPL sehingga kebanyakan wilayah Kota Batu terletak di perbukitan atau lereng-lereng gunung. Peta wilayah Kota Batu yang sebagian besar berupa pegunungan dan perbukitan membuat Kota Batu terkenal sebagai daerah yang sejuk. Hasil perkebunan andalan yang menjadi komoditas maju Kota Batu adalah buah apel [7].

### 2.3. Digital Image Processing

Pengolahan Citra Digital atau dalam bahasa Inggrisnya adalah *Digital Image Processing* yang merupakan teknik pengambilan citra dari dunia nyata untuk diolah (lihat, ubah, simpan) menggunakan komputer. Bentuk *Digital Image* tersebut bisa berupa *file* dengan ekstensi \*.jpg, \*.gif, \*.png dan sejenisnya [11].

### 2.4. Matlab

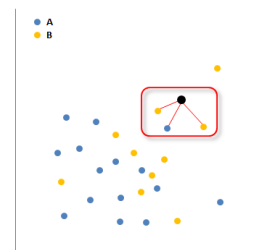
MATLAB merupakan singkatan dari *MATrix LABoratory* karena setiap data dalam MATLAB menggunakan basis matriks. MATLAB adalah bahasa pemrograman yang canggih, tertutup, dan peka huruf besar/kecil dalam lingkungan komputasi numerik yang dikembangkan oleh *MathWorks* [8].

### 2.5. Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran atau sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan [12].

### 2.6. Algoritma KNN (K-Nearest-Neighbour)

*K-Nearest Neighbour* merupakan salah satu metode untuk mengambil keputusan menggunakan pembelajaran terawasi di mana hasil dari data masukan yang baru diklasifikasi berdasarkan terdekat dalam data nilai [9].



Gambar 1. *K-Nearest Neighbour*

Gambar diatas menunjukkan adanya sejumlah titik data yang terbagi menjadi dua kelas, yaitu A (biru) dan B (kuning). Misalnya, ada data baru (hitam) yang akan kami prediksi kelasnya menggunakan algoritma KNN. Dari contoh di atas, nilai K yang digunakan adalah 3. Setelah menghitung jarak antara titik hitam satu dengan titik data lainnya, didapatkan 3 titik terdekat yang terdiri dari 2 titik kuning dan satu titik biru seperti yang digambarkan pada kotak merah, kemudian kelas untuk data baru (titik hitam) adalah B (kuning).

Kedekatan didefinisikan dalam jarak metrik, seperti jarak *Euclidean*. Jarak *Euclidean* dapat dicari dengan menggunakan persamaan 1 berikut ini:

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Keterangan:

D: Jarak kedekatan

x: Data training

y: Data testing

n: jumlah atribut individu antara 1 s.d. n

i: Atribut individu antara 1 sampai dengan n

Langkah-langkah untuk menghitung metode *KNearest Neighbor*:

1. Menentukan parameter *K* (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak Euclid (queriinstance) masing-masing objek terhadap data sampel yang diberikan menggunakan persamaan 1.
3. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak Euclid terkecil
4. Mengumpulkan kategori *Y* (Klasifikasi Nearest Neighbor)
5. Dengan menggunakan kategori Nearest Neighbor yang paling mayoritas maka dapat diprediksi nilai query instance yang telah dihitung

## 2.7. Teknik Thresholding

*Thresholding* adalah suatu metode segmentasi citra yang memisahkan objek dari latar belakang pada suatu citra berdasarkan perbedaan kecerahan atau kegelapan. Wilayah citra yang cenderung gelap akan dibuat lebih gelap (hitam sempurna dengan nilai intensitas 0), sedangkan wilayah citra yang cenderung lebih terang akan dibuat lebih terang (putih sempurna dengan nilai intensitas 1). Oleh karena itu, keluaran dari proses segmentasi menggunakan metode *thresholding* adalah citra biner dengan nilai intensitas piksel 0 atau 1. Setelah citra tersegmentasi atau objek telah berhasil dipisahkan dari latar belakang, citra biner yang diperoleh dapat digunakan sebagai topeng untuk proses *cropping* sehingga didapatkan tampilan gambar asli tanpa *background* atau dengan *background* yang bisa diubah [12].

Untuk melakukan melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus:

$$x = \frac{w}{b}$$

Penjelasan:

w = nilai derajat keabuan sebelum *thresholding*

b = jumlah derajat keabuan yang diinginkan

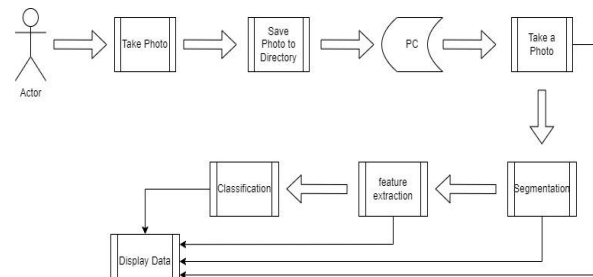
x = nilai derajat keabuan setelah *thresholding*

## 3. METODE PENELITIAN

### 3.1. Diagram Blok Sistem

Pada penelitian ini terdapat proses yang berisi System Block Diagram. Berikut ini dapat

menjelaskan proses Block Diagram pada Gambar 2 di bawah ini:

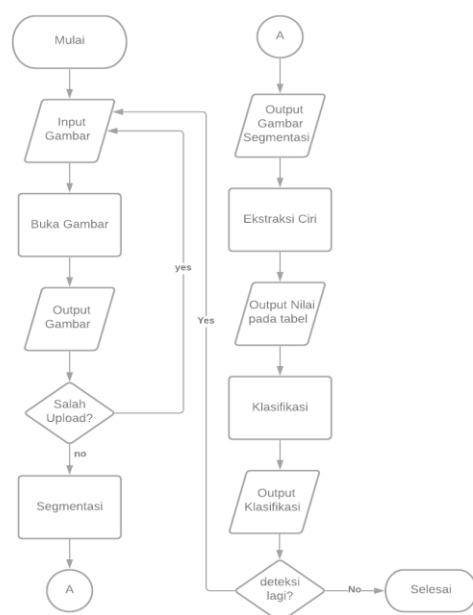


Gambar 2. Blok Diagram Sistem

Pada gambar 2, ditampilkan blok diagram sistem yang akan dikembangkan. Pertama, *user* akan mengambil foto objek penelitian, kemudian akan disimpan pada *directory* pada *PC*, setelah itu foto tersebut akan digunakan pada aplikasi lalu foto tersebut akan dilakukan proses segmentasi, yang mana pada proses segmentasi citra akan merubah citra dari warna *RGB* menjadi *grayscale* dan akan diproses kembali menjadi citra biner. Pada proses ekstraksi ciri, citra biner dan citra asli akan dihitung untuk mendapatkan nilai ciri *RGB* dari tiap citra. Pada proses klasifikasi, aplikasi akan mendeteksi tingkat kematangan citra dari target yang sudah didapatkan sebelumnya. Pada *display* data, data yang ditampilkan merupakan data citra asli, citra biner, nilai ciri *RGB*, dan keterangan hasil klasifikasi.

### 3.2. Flowchart Program

Dalam penelitian ini terdapat proses yang berisi flowchart dari program yang dibangun. Berikut ini menjelaskan proses flowchart program pada Gambar 3

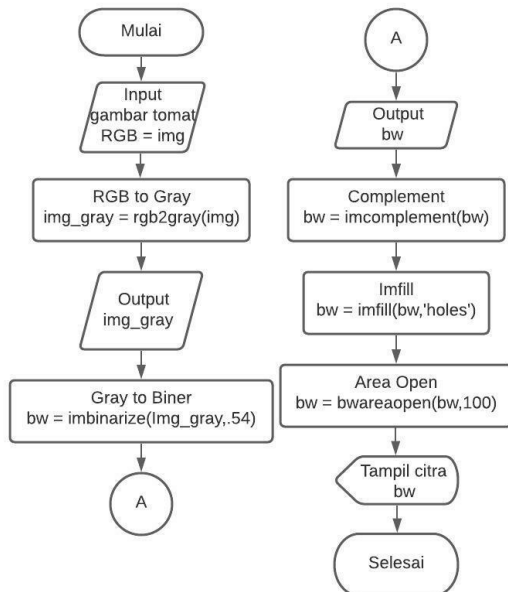


Gambar 3. flowchart program

Pada gambar 3, ditampilkan *flowchart* program yang akan dikembangkan. Pertama program akan mulai, kemudian *user* akan *input* data berupa gambar yang akan diproses menjadi buka gambar kemudian *output* yang dihasilkan berupa gambar yang sebelumnya di *input*kan. Kemudian terdapat kondisi atau *decision* yaitu apakah salah *upload*? Jika benar maka akan kembali ke bagian *input* gambar, jika tidak maka akan masuk ke proses segmentasi. Setelah dilakukan proses segmentasi, *output* yang dihasilkan berupa citra biner. Pada proses selanjutnya yaitu ekstraksi ciri, akan diambil data dari *file* citra biner dan citra asli yang akan dihitung untuk mendapatkan nilai ciri yang akan ditampilkan pada *table*. Pada proses klasifikasi, program akan memberikan keterangan kematangan tomat berupa matang, belum matang dan mentah. Kemudian terdapat kondisi lagi, apakah ingin mendeteksi kembali, jika iya maka akan kembali ke bagian *input* gambar, jika tidak maka program selesai.

### 3.3. Flowchart Thresholding

Dalam penelitian ini, terdapat proses yang berisi *flowchart Thresholding* yang digunakan. Berikut dijelaskan proses *flowchart Thresholding* pada gambar 4



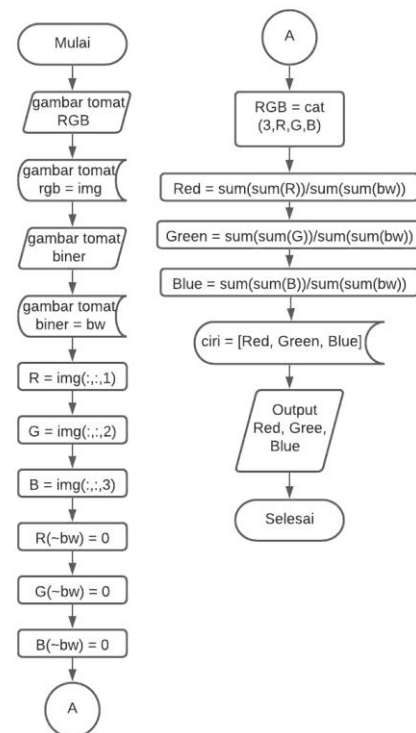
Gambar 4. *flowchart thresholding*

Pada gambar 4, ditampilkan *flowchart thresholding* yang akan digunakan. Pertama pada proses *thresholding* akan mulai. Kemudian *input* gambar tomat *RGB* dengan nama *img*. Gambar tomat tersebut diproses menjadi citra *grayscale* dengan perintah *img\_gray = rgb2gray(img)*. Setelah dilakukan proses *RGB to Gray*, hasil *output*nya berupa citra *grayscale* dengan nama

*img\_gray*. Citra *grayscale* tadi akan diproses kembali menjadi citra biner dengan menggunakan perintah *bw = imbinarize(img\_gray,.54)*. setelah dilakukan proses *Gray to Biner*, *output* yang didapat berupa citra biner. Kemudian pada proses *complement*, citra biner akan dikomplemen menggunakan perintah *bw = imcomplement(bw)*. Setelah dilakukan proses komplement, citra biner akan dilakukan *imfill* dengan perintah *bw = imfill(bw,'holes')*. Citra hasil *imfill* tadi akan diproses kembali pada proses *area open* untuk menghilangkan *noise* pada gambar dengan menggunakan perintah *bw = areaopen(bw,100)*. Setelah dilakukan proses *area open*, citra biner tadi akan di tampilkan dengan nama *bw* dan proses *thresholding* selesai.

### 3.4. Flowchart Ekstraksi Ciri

Dalam penelitian ini, terdapat proses yang berisi *flowchart Thresholding* yang digunakan. Berikut dijelaskan proses *flowchart Thresholding* pada gambar 5.



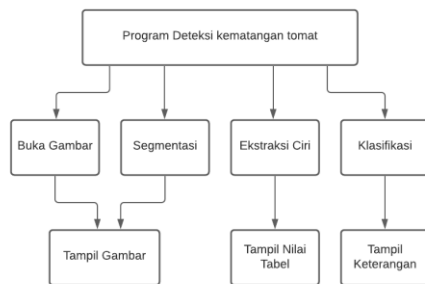
Gambar 5. *flowchart ekstraksi ciri*

Pada gambar 5, ditampilkan *flowchart thresholding* yang akan digunakan. Pertama mulai proses, kemudian program mengambil gambar tomat *RGB*, lalu gambar tomat tersebut akan disimpan pada *directory* sistem dengan nama *img*. Program juga mengambil gambar tomat biner dan disimpan pada *directory* dengan nama *bw*. Setelah itu, gambar *img* tadi akan dihitung untuk menemukan nilai  $R = \text{img}(:,:,1)$ ,  $G = \text{img}(:,:,2)$  dan  $B$

*img(:,:,3)*. Proses yang sama juga dilakukan pada citra bw, dengan perhitungan R, G dan B menggunakan negasi dan tepian = 0. Setelah diproses satu-satu dari citra RGB, proses selanjutnya yaitu *concatenate* RGB dengan perintah *cat(3,R,G,B)*. Setelah dilakukan proses *concatenate*, nilai *Red*, *Green* dan *Blue* dari citra RGB dan Biner akan ditambah sesuai perintah R, G dan B kemudian akan dibagi dan hasilnya akan disimpan *directory* dengan nama Ciri dan isi berupa *Red*, *Green* dan *Blue*. Nilai ciri tadi akan ditampilkan pada program dan selesai.

### 3.5. Struktur Menu

Dalam aplikasi yang akan dikembangkan ini, ada beberapa menu yang ditampilkan untuk dapat digunakan sesuai dengan kebutuhan pengguna. Dibawah ini adalah struktur menu yang akan dibuat



Gambar 6. Struktur Menu

Seperti pada gambar 6 Pada program deteksi kematangan tomat terdapat 4 Menu, yaitu:

1. Menu Buka Gambar, untuk membuka dan menampilkan gambar tomat yang diujikan
2. Menu Segmentasi, untuk memproses gambar tomat asli(RGB) menjadi *grayscale* dan biner dan ditampilkan
3. Menu Ekstraksi Ciri, untuk menampilkan nilai RGB pada tabel
4. Dan menu Klasifikasi, untuk menampilkan keterangan kematangan tomat.

## 4. HASIL DAN PEMBAHASAN

### 4.1. Implementasi Perangkat Lunak

Semua metode dari perancangan perangkat lunak yang telah dirancang kemudian dilakukan implementasi dengan menggunakan MATLAB R2017b.

### 4.2. Pengujian Fitur

Pengujian fitur pada penelitian ini memiliki beberapa fitur didalamnya

Tabel 1. Fitur program

No.	Pengujian	Hasil
1.	Program dapat membuka objek citra	Sesuai
2.	Program dapat menghasilkan citra biner	Sesuai
3.	Program dapat menampilkan hasil ekstraksi ciri	Sesuai

No.	Pengujian	Hasil
4.	Program dapat melakukan segmentasi citra	Sesuai
5.	Program dapat menampilkan hasil klasifikasi	Sesuai

### 4.3. Pengujian Fungsionalitas

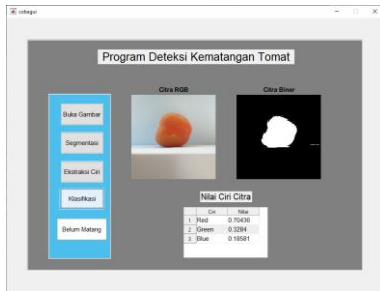
Pengujian fungsionalitas digunakan untuk membuktikan apakah sistem yang di implementasikan dapat memenuhi persyaratan dari fungsi operasional yang telah dirancang dan direncanakan sebelumnya

Tabel 2. Fungsionalitas program

No	Uji Fungsionalitas	Hasil Yang Didapat	Output Software	Keterangan
1	Software membuka Citra Gambar setelah menekan tombol "Buka Gambar"	Software membuka direktori untuk mencari citra gambarnya	Software membuka Citra Gambar setelah menekan tombol "Buka Gambar"	[√] Diterima [ ] Tidak Diterima
2	Software menampilkan citra gambar setelah user memilih gambar dari directory	Software menyimpan gambar didatabase software	Software menampilkan gambar dari directory	[√] Diterima [ ] Tidak Diterima
3	Software mengkonversi citra menjadi citra Biner setelah menekan tombol "segmentasi"	Software melakukan proses segmentasi menggunakan citra RGB menjadi grayscale dan citra biner	Software menampilkan citra biner	[√] Diterima [ ] Tidak Diterima
4	Software mengekstraksi ciri citra RGB setelah menekan tombol "ekstraksi ciri"	Software melakukan proses ekstraksi ciri dari nilai RGB pada gambar	Software menampilkan nilai dari hasil ekstraksi ciri ke table	[√] Diterima [ ] Tidak Diterima
5	Software mengklasifikasi gambar setelah menekan tombol "klasifikasi"	Software memanggil perintah pada module dan membandingkan ciri dan target yang sudah ditentukan	Software mengklasifikasi gambar setelah menekan tombol "klasifikasi"	[√] Diterima [ ] Tidak Diterima
6	Software menampilkan hasil perbandingan setelah menekan tombol "klasifikasi"	Software menyimpan data perbandingan dalam database software	Software menampilkan hasil perbandingan pada edit text	[√] Diterima [ ] Tidak Diterima

#### 4.4. Pengujian Sistem

Pengujian *system* dilakukan dengan cara memberi perbandingan terhadap citra yang ingin diuji.



Gambar 7. Hasil Uji Program

Pengujian *system* pada penelitian ini menggunakan perbandingan dari *Red*, *Green*, dan *Blue* dari data latih dan data uji.

Tabel 3. Hasil Pengujian Sistem pada data latih

No	Nama File	Red	Green	Blue	Kategori
1.	Matang 1	0,559669	0,200624	0,171535	Matang
2.	Matang 2	0,569114	0,211874	0,180302	Matang
3.	Matang 3	0,553215	0,192823	0,167937	Matang
4.	Matang 4	0,558539	0,198785	0,172351	Matang
5.	Matang 5	0,504282	0,198072	0,179743	Matang
6.	Belum Matang 1	0,499443	0,367338	0,313895	Belum Matang
7.	Belum Matang 2	0,508841	0,241478	0,164751	Belum Matang
8.	Belum Matang 3	0,536947	0,231995	0,175142	Belum Matang
9.	Belum Matang 4	0,524221	0,248237	0,181426	Belum Matang
10.	Belum Matang 5	0,535136	0,312671	0,258717	Belum Matang
11.	Mentah 1	0,382659	0,37033	0,266504	Mentah
12.	Mentah 2	0,426864	0,4059	0,320037	Mentah
13.	Mentah 3	0,400983	0,395375	0,310657	Mentah
14.	Mentah 4	0,347282	0,266973	0,174664	Mentah
15.	Mentah 5	0,381807	0,274599	0,171872	Mentah
Prosentase hasil = jumlah benar/jumlah data * 100 = 15/15 * 100 = 100					

Dari penjelasan *table* 3 diatas, setelah dilakukan pengujian dengan menggunakan foto yang didapatkan secara langsung maka, didapatkan bahwa buah yang matang memiliki nilai ciri citra terkecil sebesar *Red* = 0.50428, *Green* = 0.19282

dan *Blue* = 0.16794 dan nilai tertinggi sebesar *Red* = 0.56911, *Green* = 0.21187 dan *Blue* = 0.1803.

Untuk buah yang belum matang memiliki nilai ciri citra minimal sebesar *Red* = 0.49944, *Green* = 0.232 dan *Blue* = 0.16475 dan nilai tertinggi sebesar *Red* = 0.53695, *Green* = 0.36734 dan *Blue* = 0.3139.

Dan buah yang Mentah memiliki nilai ciri citra minimal sebesar *Red* = 0.34728, *Green* = 0.26697 dan *Blue* = 0.17187 dan nilai tertinggi sebesar *Red* = 0.42686, *Green* = 0.4059 dan *Blue* = 0.32004.

Pada *table*, untuk kolom yang berwarna merah menandakan nilai terkecil, sedangkan warna kuning menunjukkan nilai terbesar.

Tabel 4. Hasil Pengujian Sistem pada data uji

No	Nama File	R	G	B	K.S	K.P	K
1.	Matang 6	0,541595	0,19295	0,189697	Mata ng	Mata ng	Sesuai
2.	Matang 7	0,497194	0,21858	0,196283	Mata ng	Mata ng	Sesuai
3.	Matang 8	0,495783	0,200523	0,181277	Mata ng	Mata ng	Sesuai
4.	Matang 9	0,473435	0,22152	0,194637	Mata ng	Mata ng	Sesuai
5.	Matang 10	0,510542	0,254293	0,23151	Belum Mata ng	Mata ng	Tidak Sesuai
6.	Belum Matang 6	0,70436	0,328402	0,185814	Belum Mata ng	Belum Mata ng	Sesuai
7.	Belum Matang 7	0,520566	0,327679	0,271859	Belum Mata ng	Belum Mata ng	Sesuai
8.	Belum Matang 8	0,53435	0,249619	0,170275	Mata ng	Belum Mata ng	Tidak Sesuai
9.	Belum Matang 9	0,537432	0,331962	0,240851	Belum Mata ng	Belum Mata ng	Sesuai
10.	Belum Matang 10	0,536431	0,315455	0,237612	Belum Mata ng	Belum Mata ng	Sesuai
11.	Mentah 6	0,364513	0,334404	0,201823	Ment ah	Ment ah	Sesuai
12.	Mentah 7	0,333066	0,264529	0,165212	Ment ah	Ment ah	Sesuai
13.	Mentah 8	0,368671	0,322837	0,202236	Ment ah	Ment ah	Sesuai
14.	Mentah 9	0,338854	0,30386	0,170459	Ment ah	Ment ah	Sesuai
15.	Mentah 10	0,414883	0,347529	0,251428	Belum Mata ng	Ment ah	Tidak Sesuai
Prosentase hasil = jumlah benar/jumlah data * 100 = 12/15 * 100 = 80							

Dari *table 4* dapat dilihat, setelah dilakukan pengujian dengan menggunakan foto yang didapatkan secara langsung maka didapatkan bahwa buah yang matang memiliki nilai ciri citra terkecil sebesar  $Red = 0.47344$ ,  $Green = 0.19295$  dan  $Blue = 0.18128$ , dan nilai maksimal sebesar  $Red = 0.5416$ ,  $Green = 0.25429$  dan  $Blue = 0.23151$ .

Untuk buah yang belum matang memiliki nilai ciri citra minimal sebesar  $Red = 0.52057$ ,  $Green = 0.24962$  dan  $Blue = 0.17028$ , dan nilai maksimal sebesar  $Red = 0.70436$ ,  $Green = 0.33196$  dan  $Blue = 0.27186$ .

Dan untuk buah yang mentah memiliki nilai ciri citra minimal sebesar  $Red = 0.33307$ ,  $Green = 0.26453$  dan  $Blue = 0.16521$ , dan nilai maksimal sebesar  $Red = 0.414883$ ,  $Green = 0.347529$  dan  $Blue = 0.251428$ .

Dari nilai ciri citra tersebut didapatkan ada 2 data yang tidak sesuai dengan keterangan dari petani.

Pada *table*, untuk kolom yang berwarna merah menandakan nilai terkecil, sedangkan warna kuning menunjukkan nilai terbesar.

Keterangan:

$Red(R)$  : Nilai warna  $Red$  dari proses  $RGB$  pada file tomat.

$Green(G)$  : Nilai warna  $Green$  dari proses  $RGB$  pada file tomat.

$Blue(B)$  : Nilai warna  $Blue$  dari proses  $RGB$  pada file tomat

K.S : hasil perhitungan program menentukan Matang dan Belum Matang

K.P : keterangan hasil dari petani langsung

K : Membandingkan perhitungan *system* dengan petani

## 5. KESIMPULAN

Hasil dari pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan, diantaranya, Hasil pengujian sistem, didapatkan persentase tingkat kematangan dari sistem sebesar 80%, hasil ini adalah perbandingan data latih dengan data uji, nilai yang menjadi acuan yaitu nilai RGB. Hasil perbandingan dapat dilihat dari patokan nilai RGB dari tiap-tiap format tomat, nilai R pada file tomat matang sebesar 0,541595 - 0,473435, nilai G sebesar 0,19295 - 0,254293 dan B sebesar 0,181277 - 0,23151. Untuk file tomat belum matang nilai R sebesar 0,520566 - 0,70436, untuk G sebesar 0,249619 - 0,331962 dan B sebesar 0,170275 - 0,271859. Dari hasil perhitungan tersebut, jika nilai dari tiap format file tomat tersebut melebihi atau kurang dari patokan nilai format yang didapatkan maka hasilnya akan berupa keterangan tomat tersebut termasuk kategori dari format file yang lain.

## DAFTAR PUSTAKA

- [1] M. Yogi, "APLIKASI DETEKSI KEMATANGAN BUAH SEMANGKA BERBASIS NILAI RGB MENGGUNAKAN METODE THRESHOLDING," *Jurnal Riset Komputer*, 2016
- [2] H. Supriyatno, "Targetkan Kawasan Pertanian Organik Kota Batu Bertambah Jadi 20 Desa," 26 Mei 2019. [Online]. Available: <https://www.harianbhirawa.co.id/targetkan-kawasan-pertanian-organik-kota-batu-bertambah-jadi-20-desa/>
- [3] A. Wijayono and V. G. V. Putra, "PENERAPAN TEKNOLOGI PENGOLAHAN CITRA DIGITAL DAN KOMPUTASI PENGUKURAN DAN PENGUJIAN BERBAGAI PARAMETER BENANG," *Paper*
- [4] A. D. Harisna, "Image Processing," 2 5 2009. [Online]. Available: <https://ndoware.com/image-processing.html>
- [5] Anonymous, "Apa Itu Image Processing," 03 12 2019. [Online]. Available: <https://www.immersa-lab.com/apa-itu-image-processing.htm>
- [6] Anonymous, "Tomat," 2018. [Online]. Available: <https://b-pikiran.cekkembali.com/tomat/>
- [7] "Profil Kota Batu," [Online]. Available: <http://perkotaan.bpiw.pu.go.id/v2/kota-sedang/31>.
- [8] Advernesia, "Pengertian Matlab dan Kegunaannya," 21 September 2017. [Online]. Available: <https://www.advernesia.com/blog/matlab/apa-itu-matlab/>.
- [9] A. Lutfia, "ilmudataPy," [Online]. Available: <https://ilmudatapy.com/algorithm-k-nearest-neighbor-knn-untuk-klasifikasi/>. [Accessed 10 January 2022].
- [10] "Detail artikel," [Online]. Available: <https://www.dinkes.jogjapro.go.id/berita/detail/tomat-buah-sehat-kaya-manfaat-tomat-buah-sehat-kaya-manfaat>.
- [11] Anonymous, "konsep digital image processing," 4 Maret 2011. [Online]. Available: <http://dendieisme.blogspot.com/2011/03/konsep-digital-image-processing.html>.
- [12] L. I. M. Multimedia, "Modul Praktikum Pengolahan Citra Digital," 2019
- [13] "Badan Pusat Statistik," [Online]. Available: <https://batukota.bps.go.id/statistable/2019/11/28/536/luas-panen-tanaman-sayuran-menurut-kecamatan-dan-jenis-tanaman-di-kota-batu-ha-2017-dan-2018.html>