

Aplikasi Pendeteksian Kematangan Tomat Menggunakan *Thresholding*

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 adalah tanaman yang pertama kali ditemukan di Amerika Selatan, masih berkerabat dengan terong, kentang dan paprika. Tomat merupakan salah satu komoditas andalan Kota Batu, sebelumnya ada terdapat 18 wilayah di Kota Batu yang telah tersertifikasi pertanian organik. Pada tahun 2018 total produksi tomat tahun 2017 sebanyak 130.298Kw dan jumlah petani yang bekerja sebanyak 278, sedangkan pada tahun 2019 total produksi tomat sebanyak 153.622Kw dan jumlah 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 dan ukuran buah tomat. Tomat yang masih segar dan matang biasanya berwarna merah cerah dan tentunya dengan ukuran yang besar. Namun, petani hanya memilah tomat berdasarkan warna dan seringkali mengabaikan ukuran tomat itu sendiri. Padahal besar kecilnya ukuran tomat juga sangat berpengaruh dalam penjualan di pasaran. Berdasarkan hasil perancangan dan implementasi aplikasi pencarian tempat wisata menggunakan metode *location based service* di kabupaten Timor Tengah Selatan berbasis *android*, dari pengujian fungsional yang dilakukan pada 3 sistem operasi yang berbeda yaitu Nougat, *pia*, oreo, disimpulkan bahwa fungsi layanan berbasis lokasi berjalan dengan baik.

Kata kunci : Kematangan, Kota Batu, Tomat, Matlab

1. PENDAHULUAN

Tomat merupakan tumbuhan yang pertama kali ditemukan di Amerika Selatan, masih berkerabat dengan terong, 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 [3].

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 pengolahan citra, pengukuran berbagai parameter benang yang membutuhkan penilaian secara visual dapat dikuantifikasikan [4]. 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 [5]. Teknik ini berguna untuk berbagai bidang seperti, komputer, kedokteran, industri, dan lain-lain [6]. Contoh hasil yang diterapkan dalam kehidupan seperti penelitian membuat aplikasi *sorting* buah tomat.

Berdasarkan paparan tersebut diatas, penulis tergas 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 Penelitian Terkait

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 [1].

Tomat (*Lycopersicon esculentum Mill*) merupakan sayuran berbentuk buah yang banyak diproduksi di daerah tropis dan subtropis. Budidaya tomat terus berkembang seiring dengan meningkatnya permintaan tomat. Tomat mudah rusak, penanganan tomat yang tidak tepat sebelum, selama dan setelah panen dapat mempercepat proses kerusakan yang mengakibatkan penurunan kualitas yang pada akhirnya mempengaruhi nilai gizi dan nilai ekonomi. Untuk menghindari kerusakan pada buah tomat maka perlu dilakukan penanganan yang baik dan benar terhadap buah tomat.

2.2 Tanaman Tomat

Tomat termasuk dalam kategori tanaman Solanaceae 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[7]. Buah tomat yang sudah

matang dapat dipanen dan dikonsumsi, baik diolah maupun dimakan langsung.

2.3 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 [8].

2.4 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.

2.5 Matlab

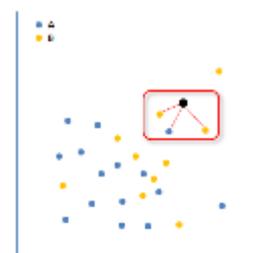
MATLAB merupakan singkatan dari *MATrix LABoratory* karena setiap data dalam MATLAB menggunakan basis matriks. MATLAB adalah bahasa pemrograman yang canggih, tertutup, dan peka huruf besar/kecil dalam lingkungan komputasi numerik yang dikembangkan oleh *MathWorks*.

2.6 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 [11].

2.7 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 [20].



gambar 2.3 Algoritma KNN

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) [21].

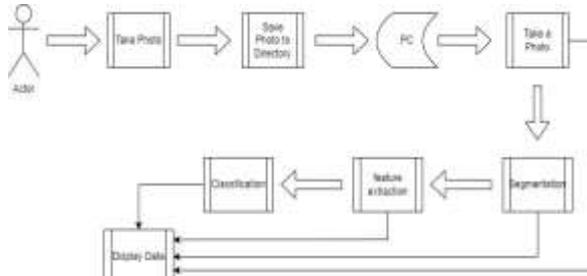
2.8 Teknik Thresholding

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 [11].

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 3.1 di bawah ini:



Gambar 3.1 Diagram Blok System

Pada gambar 3.1, 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 3.2

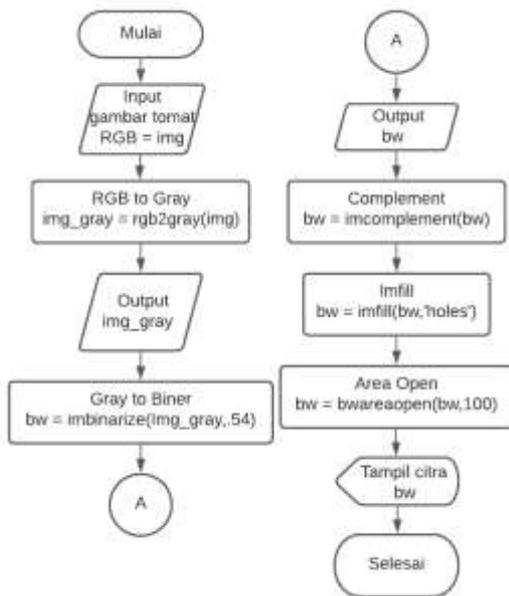


Gambar 3.2 flowchart program

Pada gambar 3.2, 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 inputkan. 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 3.3

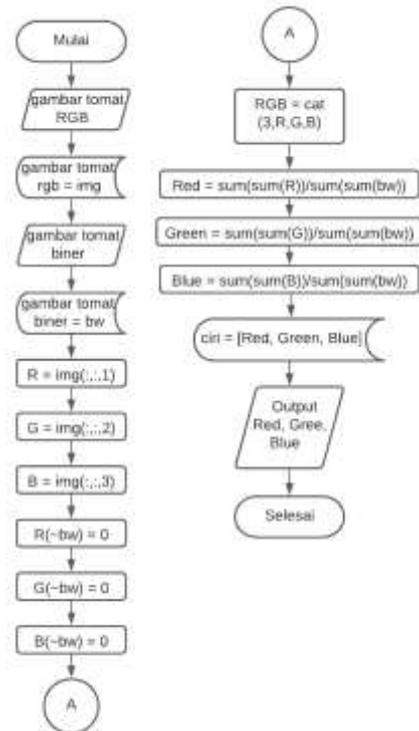


Gambar 3.3 flowchart thresholding

Pada gambar 3.3, 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 *outputnya* 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 3.4



Gambar 3.4 flowchart ekstraksi ciri

Pada gambar 3.4, 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 $B = \text{img}(:, :3)$. 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 3.5 Struktur Menu

Seperti pada gambar 3.5 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 4.1 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
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 4.2 Fungsionalitas program

No	Uji Fungsionalitas	Hasil Yang Didapat	Output Software	Keterangan
1	Software membuka Citra	Software membuka direktori	Software membuka Citra	[√] Diterima [] Tidak

No	Uji Fungsionalitas	Hasil Yang Didapat	Output Software	Keterangan
	Gambar setelah menekan tombol "Buka Gambar"	untuk mencari citra gambarnya	Gambar setelah menekan tombol "Buka Gambar"	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 mengesktraksi cirikan gambar RGB setelah menekan tombol "ekstraksi ciri"	software melakukan proses esktraksi 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 4.1 Hasil Uji Program

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

Tabel 4.3 Hasil Pengujian Sistem pada data latih

No.	Nama File	Red	Green	Blue	Kategori
1.	Matang 1	0,5596 69	0,2006 24	0,17153 5	Matang
2.	Matang 2	0,5691 14	0,2118 74	0,18030 2	Matang
3.	Matang 3	0,5532 15	0,1928 23	0,16793 7	Matang
4.	Matang 4	0,5585 39	0,1987 85	0,17235 1	Matang
5.	Matang 5	0,5042 82	0,1980 72	0,17974 3	Matang
6.	Belum Matang 1	0,4994 43	0,3673 38	0,31389 5	Belum Matang
7.	Belum Matang 2	0,5088 41	0,2414 78	0,16475 1	Belum Matang
8.	Belum Matang 3	0,5369 47	0,2319 95	0,17514 2	Belum Matang
9.	Belum Matang 4	0,5242 21	0,2482 37	0,18142 6	Belum Matang
10.	Belum Matang 5	0,5351 36	0,3126 71	0,25871 7	Belum Matang
11.	Mentah 1	0,3826 59	0,3703 3	0,26650 4	Mentah
12.	Mentah 2	0,4268 64	0,4059 7	0,32003 7	Mentah
13.	Mentah 3	0,4009 83	0,3953 75	0,31065 7	Mentah
14.	Mentah 4	0,3472 82	0,2669 73	0,17466 4	Mentah
15.	Mentah 5	0,3818 07	0,2745 99	0,17187 2	Mentah
Prosentase hasil = jumlah benar/jumlah data * 100 = 15/15 * 100 = 100					

Dari penjelasan *table* 4.3 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 4.4 Hasil Pengujian Sistem pada data uji

No	Nama File	R	G	B	K.S	K.P	K
1.	Matang 6	0,54 1595	0,19 295	0,18 9697	Matang	Matang	Sesuai
2.	Matang 7	0,49 7194	0,21 8538	0,19 6283	Matang	Matang	Sesuai
3.	Matang 8	0,49 5783	0,20 0523	0,18 1277	Matang	Matang	Sesuai
4.	Matang 9	0,47 3435	0,22 152	0,19 4637	Matang	Matang	Sesuai
5.	Matang 10	0,51 0542	0,25 4293	0,23 151	Belum Matang	Matang	Tidak Sesuai
6.	Belum Matang 6	0,70 436	0,32 8402	0,18 5814	Belum Matang	Belum Matang	Sesuai
7.	Belum Matang 7	0,52 0566	0,32 7679	0,27 1859	Belum Matang	Belum Matang	Sesuai
8.	Belum Matang 8	0,53 435	0,24 9619	0,17 0275	Matang	Belum Matang	Tidak Sesuai
9.	Belum Matang 9	0,53 7432	0,33 1962	0,24 0851	Belum Matang	Belum Matang	Sesuai
10.	Belum Matang 10	0,53 6431	0,31 5455	0,23 7612	Belum Matang	Belum Matang	Sesuai
11.	Mentah 6	0,36 4513	0,33 4404	0,20 1823	Mentah	Mentah	Sesuai
12.	Mentah 7	0,33 3066	0,26 4529	0,16 5212	Mentah	Mentah	Sesuai
13.	Mentah 8	0,36 8671	0,32 2837	0,20 2236	Mentah	Mentah	Sesuai
14.	Mentah 9	0,33 8854	0,30 386	0,17 0459	Mentah	Mentah	Sesuai
15.	Mentah 10	0,41 4883	0,34 7529	0,25 1428	Belum Matang	Mentah	Tidak Sesuai
Prosentase hasil = jumlah benar/jumlah data * 100 = 12/15 * 100 = 80							

Penjelasan *table* 4.4

Dari penjelasan *table* 4.4 diatas, setelah dilakukan pengujian dengan menggunakan foto yang didapatkan secara langsung maka didapatkan bahwa buah yang matang memiliki nilai ciri citra terkecil sebesar *Red* = 0.47344, *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 kesesuaian kematangan tomat antara hasil keterangan sistem dan keterangan petani.

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

Keterangan:

1. $Red(R)$: Nilai warna Red dari proses RGB pada file tomat.
2. $Green(G)$: Nilai warna $Green$ dari proses RGB pada file tomat.
3. $Blue(B)$: Nilai warna $Blue$ dari proses RGB pada file tomat
4. K.S: hasil perhitungan program menentukan Matang dan Belum Matang
5. K.P: keterangan hasil dari petani langsung
6. K: Membandingkan perhitungan *system* dengan petani

5. KESIMPULAN

Hasil dari pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan, diantaranya:

Berdasarkan hasil pengujian diatas, dapat disimpulkan bahwa aplikasi yang dibuat dapat mendeteksi tingkat kematangan tomat dengan menggunakan nilai ekstraksi ciri dari data latih dan data uji yang dihasilkan dari proses thresholding yang dapat merubah citra rgb menjadi citra keabuan dan citra biner. Nilai rata-rata yang didapatkan dari data latih dan data uji akan di klasifikasikan menggunakan algoritma Knn untuk membedakan tomat belum matang dan tomat mentah.

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] "Badan Pusat Statistik," [Online]. Available: <https://batukota.bps.go.id/statictable/2019/11/28/536/luas-panen-tanaman-sayuran-menurut-kecamatan-dan-jenis-tanaman-di-kota-batu-ha-2017-dan-2018.html>.
- [4] A. Wijayono and V. G. V. Putra, "PENERAPAN TEKNOLOGI PENGOLAHAN CITRA DIGITAL DAN KOMPUTASI PENGUKURAN DAN PENGUJIAN BERBAGAI PARAMETER BENANG," *Paper*.
- [5] A. D. Harisna, "Image Processing," 2 5 2009. [Online]. Available: <https://ndoware.com/image-processing.html>.
- [6] Anonymous, "Apa Itu Image Processing," 03 12 2019. [Online]. Available: <https://www.immersa-lab.com/apa-itu-image-processing.htm>.
- [7] Anonymous, "Tomat," 2018. [Online]. Available: <https://b-pikiran.cekkembali.com/tomat/>.
- [8] "Profil Kota Batu," [Online]. Available: <http://perkotaan.bpiw.pu.go.id/v2/kota-sedang/31>.
- [9] Anonymous, "konsep digital image processing," 4 Maret 2011. [Online]. Available: <http://dendieisme.blogspot.com/2011/03/konsep-digital-image-processing.html>.
- [10] Advernesia, "Pengertian Matlab dan Kegunaannya," 21 September 2017. [Online]. Available: <https://www.advernesia.com/blog/matlab/apa-itu-matlab/>.
- [11] L. I. M. Multimedia, "Modul Praktikum Pengolahan Citra Digital," 2019.
- [12] "Detail artikel," [Online]. Available: <https://www.dinkes.jogjaprovo.go.id/berita/detail/tomat-buah-sehat-kaya-manfaat-tomat-buah-sehat-kaya-manfaat>.
- [13] "Mendeteksi skema warna dalam gambar," [Online]. Available: <https://docs.microsoft.com/id-id/azure/cognitive-services/computer-vision/concept-detecting-color-schemes>.
- [14] I. N. P. Mukhti, Suwandi and H. Bethaningtyas, *SISTEM OTOMASI DALAM PENYORTIRAN TOMAT DENGAN IMAGE*.
- [15] H. Prabowo, "DETEKSI KONDISI KEMATANGAN BUAH JERUK BERDASARKAN," 2017.
- [16] B. M. Sari, "Identifikasi Tingkat Kematangan Buah Strawberry Berdasarkan Warna," pp. 225-230, 2020.
- [17] I. W. A. Heryanto, Artama, M. W. S. Kurniawan and I. G. A. Gunadi, "SEGMENTASI WARNA DENGAN METODE," 2020.
- [18] M. H. Fauzi and H. Tjandrasa, "IMPLEMENTASI THRESHOLDING CITRA MENGGUNAKAN".
- [19] M. Rifqi, M. Akbar and Y. Fitriasia, "Aplikasi Pendeteksian Kematangan Tandan Buah Segar (TBS) Kelapa Sawit Berdasarkan Komposisi Warna Menggunakan Algoritma K-NN," pp. 99-107, 2020.
- [20] A. J. T. D. Yanosma and K. Anggriani, "IMPLEMENTASI METODE K-NEAREST NEIGHBOR (KNN) DAN SIMPLE ADDITIVE WEIGHTING (SAW) DALAM PENGAMBILAN KEPUTUSAN SELEKSI PENERIMAAN ANGGOTA PASKIBRAKA," *Pseudocode*, vol. III, no. 2, pp. 98-112, 2016.
- [21] A. Lutfia, "ilmudataPy," [Online]. Available: <https://ilmudatapy.com/algoritma-k-nearest-neighbor-knn-untuk-klasifikasi/>. [Accessed 10 January 2022].