

APLIKASI PENDETEKSIAN KEMATANGAN BUAH PEPAYA MENGUNAKAN TEKNIK SEGMENTASI WARNA

Heru Wahyu Herwanto, Triyanna Widiyaningtyas, Akhmad Rosyidin

Abstrak: Segmentasi warna adalah teknik untuk membedakan atau memisahkan objek-objek yang ada dalam suatu citra, sehingga bentuk, luas, model dan ketepatan hitungan menjadi sesuatu yang tidak dapat dihindari. Untuk mencapai hal itu manusia berusaha merekayasa seperangkat elektronik hardware atau software yang otomatis agar kinerjanya praktis dan efisien. Dalam penelitian ini, telah dibuat suatu perangkat lunak aplikasi untuk mendeteksi kematangan buah pepaya. Aplikasi ini dibuat dengan menggunakan microsoft visual basic 6.0. Pendeteksian kematangan buah pepaya dilakukan dengan metode cropping, metode grayscale dan metode biner. Hasil pengujian terhadap 10 kondisi objek buah pepaya menghasilkan error rata-rata 12,14%. Pada pengujian paling tinggi nilai prosentase kematangannya adalah 87%. Sedangkan nilai paling rendah pada nilai prosentase kematangannya 1%. Berdasarkan hasil pengujian ini, dapat disimpulkan bahwa cahaya mempengaruhi dalam aplikasi pendeteksian kematangan buah. Disarankan agar dilakukan penelitian lebih lanjut dengan menggunakan metode-metode yang lain dalam pengolahan citra. Dengan tujuan supaya memiliki nilai error yang lebih kecil pada pendeteksian kematangan buah.

Kata kunci: segmentasi warna, *grayscale*, *biner*.

Salah satu tujuan perguruan tinggi adalah mengembangkan ilmu pengetahuan dan teknologi untuk meningkatkan taraf kehidupan masyarakat. Maka daripada itu kegiatan pengembangan kreativitas serta inovasi menjadi kata kunci untuk mengantarkan tercapainya tujuan tersebut. Dalam menumbuhkan dan mengembangkan kreativitas dan inovasi tidaklah cukup hanya dengan teoritis, akan tetapi harus dilatih untuk mengimplementasikan ide/gagasan dalam wacana teoritis tersebut ke dalam bentuk hasil karya.

Pengolahan citra adalah suatu teknik yang digunakan untuk memproses citra atau gambar dengan cara memanipulasinya dengan data gambar yang diinginkan untuk mendapatkan informasi tertentu dari gambar yang diamati. Banyak data yang diperoleh apabila data yang diamati menampilkan suatu keadaan yang sesuai dengan data awal. Maka dari itu hal-hal yang berhubungan dengan bentuk, warna, ukuran dan parameter-parameter lainnya yang akan diketahui semua ini karena

pada setiap element gambar (*pixel*) mempunyai kurang lebih dua buah informasi mengenai letak dan warna cat. Semakin banyak informasi yang didapatkan dari gambar atau objek maka informasi mengenai bentuk, luas dan model dari suatu benda dan lainnya. Begitu juga dapat dipergunakan untuk mendapatkan informasi dari objek atau gambar yang bergerak.

Proses segmentasi untuk membedakan atau memisahkan objek-objek yang ada dalam suatu citra, seperti memisahkan objek dengan latar belakang berdasarkan sifat-sifat tertentu dan kriteria keserupaan yang tertentu antara tingkat keabuan suatu *pixel* dengan tingkat keabuan *pixel-pixel* tetangganya. Proses segmentasi lebih banyak merupakan suatu proses pra pengolahan pada sistem pengenalan objek dalam citra, yang termasuk bidang *computer vision*.

Pada konsep sebenarnya ketepatan hitungan menjadi sesuatu yang tidak dapat dihindari dalam pendeteksian kematangan buah ini karena berhubungan dengan kon-

disi buah yang akan dipilih. Dasar dari dibuatnya pendeteksian kematangan buah ini untuk memberi kemudahan pada pemilihan buah yang akan dipilih, sehingga dalam pendeteksian kematangan buah ini lebih mengetahui kondisi atau keadaan buah yang akan dipilih.

Perlu diketahui bahwa dalam penelitian ini digunakan perangkat keras untuk pengambilan gambar yaitu *webcam*. Untuk pengolahan citra menggunakan *Microsoft visual basic 6.0* sehingga didapatkan informasi yang diinginkan.

METODE

Perancangan Sistem

Pada bagian ini, akan dilakukan perancangan dan implementasi pendeteksian kematangan buah pepaya menggunakan teknik segmentasi warna, yang meliputi: (1) Perancangan perangkat keras, dan (2) Perancangan perangkat lunak.

Perancangan perangkat keras

Tahap paling awal yang harus dilakukan pada penelitian ini adalah persiapan peralatan-peralatan (*hardware*) yang dibutuhkan dalam sistem pendeteksian kematangan buah pepaya, yaitu PC (*Personal Computer*) atau Laptop

Komputer adalah seperangkat piranti elektronik *digital* yang tiap-tiap bagiannya dapat dikelompokkan sebagai peralatan *input*, peralatan *output* dan *central processing unit*. Pada penelitian ini, spesifikasi dari laptop yang digunakan adalah sebagai berikut :

- OS : Microsoft Windows XP Professional SP 3
- System Model : Acer Aspire 4730Z
- Processor : Intel® Pentium® Dual Core T3200 @2.00GHz
- Memory : 1 Gb RAM

Unit webcam

Webcam digunakan sebagai sarana masukan untuk memberikan data obyek gambar ke komputer. Dalam penelitian ini, *webcam* yang digunakan adalah *logitech* dengan tipe C200, dengan resolusi yang digunakan berukuran 640 x 480 seperti pada gambar 1.



Gambar 1. Logitech webcam C200

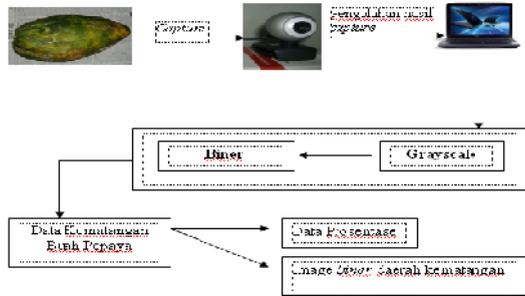
Spesifikasi dari *webcam* yang digunakan pada penelitian ini adalah sebagai berikut : (1) *VGA sensor (640 X 480 pixel)*, (2) *Video capture: up to 640 x 480 pixel*, (3) *Photos: up to 1.3 megapixels (software enhanced)*, (4) *Video capture up to 30 frames per second (with recommended systems)*, (5) *Built-in microphone with Logi-tech® RightSound™ technology*, (6) *Hi-Speed USB 2.0 certified*, (7) *Universal clip fits notebooks, LCD or CRT monitors*, (8) *Snapshot button for capturing photos*, dan (9) *Manual focus*

Perancangan perangkat lunak

Perangkat lunak yang dibutuhkan dalam membuat sistem pendeteksian kematangan buah pepaya pada penelitian ini adalah *Microsoft visual basic 6.0* dengan tambahan *library* untuk mengkoneksikan kamera dengan pemrograman *microsoft visual basic 6.0* dan *library camera active avicap32.dll* untuk pemanggilan kamera dengan menggunakan *microsoft visual basic 6.0*.

Diagram blok sistem

Secara umum diagram blok sistem yang dilakukan pada penelitian ini sesuai dengan gambar 2.



Gambar 2. Diagram Blok Sistem Kematangan Buah

Webcam menangkap (*capture*) buah pepaya kemudian komputer akan melakukan proses pengolahan citra dengan metode *cropping* selanjutnya dengan metode *gray-scale*, setelah metode *grayscale* selesai, *image* diolah kembali dengan metode *biner* sehingga dapat diperoleh data kematangan buah pepaya yang berupa *image biner* daerah kematangan dan data persentase

Rancangan GUI (Graphic User Interface)

Pada gambar 3 adalah rancangan GUI yang akan dibuat pada penelitian ini, begitu pula tentang penjelasan tiap bagian-bagiannya.



Gambar 3. Tampilan GUI aplikasi kematangan buah

Keterangan dari gambar 3 tampilan GUI aplikasi kematangan buah pepaya adalah sebagai berikut :

- 1) Tampilan Video *webcam*: berfungsi untuk menampilkan gambar langsung menggunakan *webcam*.
- 2) Kotak *Cropping* : berfungsi untuk memotong gambar.
- 3) *Tools capture* : tombol yang berfungsi untuk *capture image*.
- 4) *Tools pengolahan*: berfungsi untuk mengolah *image* hasil *capture* ke metode *grayscale* dan metode *biner*.
- 5) *Frame* hasil *Capture* : berfungsi untuk menampilkan *image* hasil *cropping*.
- 6) *Frame grayscale* : berfungsi untuk menampilkan *image* hasil pengolahan *grayscale*.
- 7) *Frame biner* objek : berfungsi untuk menampilkan *image* hasil pengolahan *biner* objek.
- 8) *Frame output* kematangan : berfungsi untuk menampilkan *image* daerah kematangan.
- 9) Persentase kematangan buah : *output* data prosentase kematangan buah.
- 10) Menu : terdapat menu keluar dan *about*.

Perancangan aplikasi

Pada perancangan sistem pendeteksian kematangan buah pepaya ini dilakukan secara *real time*. Untuk mendapatkan data kematangan dilakukan tiga metode untuk yang pertama metode *cropping* yang kedua metode *grayscale* sedangkan yang tiga metode *biner* dan menghasilkan data persentase kematangan dari objek buah pepaya.

Metode *cropping*

Cropping adalah pengolahan citra dengan kegiatan memotong satu bagian dari citra. Rumus yang digunakan :

$$x' = x - xL \text{ untuk } x = xL \text{ sampai } xR$$

$$y' = y - yT \text{ untuk } y = yT \text{ sampai } yB$$

(xL,yT) dan (xR,yB) adalah koordinat titik pojok kiri atas dan pojok kanan bawah citra yang akan di-*crop*. Ukuran citra menjadi :

$$w' = xR - xL$$

$$h' = yB - YT$$

Bentuk tampilan dari metode *cropping* seperti gambar 4.



Gambar 4. Tampilan hasil metode *cropping*

Metode *grayscale*

Metode ini merupakan proses mengubah citra berwarna menjadi citra *grayscale* yang digunakan untuk menyederhanakan model citra. Dalam citra ini tidak lagi warna, yang ada adalah derajat keabuan. Adapaun sebagian *code* untuk *grayscale* adalah sebagai berikut :

```
Private Sub CmdAbu_Click()
Dim X As Integer, Y As Integer
Dim p As Long
Dim R As Integer, G As Integer, B As Integer
Dim Gray As Integer
For Y = 0 To PicHasil.ScaleHeight - 1
For X = 0 To PicHasil.ScaleWidth - 1
    p = GetPixel(PicHasil.hdc, X, Y)
    R = p And &HFF
    G = (p \ &H100) And &HFF
    B = (p \ &H10000) And &HFF
    Gray = (0.299 * R + 0.587 * G + 0.114 * B)
    If Gray > 255 Then Gray = 255
    SetPixel PicOlah.hdc, X, Y, RGB(Gray, Gray, Gray)
Next
Next
SavePicture PicOlah.Image, App.Path +
"\hasil\gray.jpg"
CmdOk.Enabled = True
MsgBox "Grayscale Selesai"
End Sub
```

Hasil dari merubah citra warna ke citra *grayscale* (derajat keabuan) seperti gambar 5.



Gambar 5. Tampilan hasil metode *grayscale*

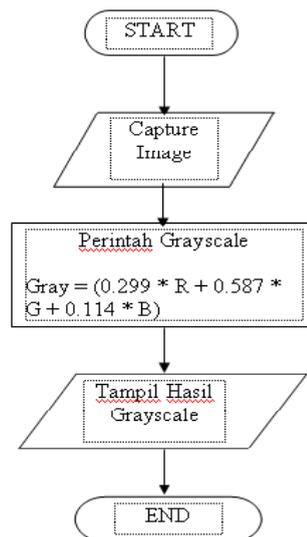
Ketika proses *grayscale* sudah selesai maka akan di akhiri dengan *message box* seperti gambar 6.



Gambar 6. Tampilan *message box grayscale* selesai

Disini, proses *grayscale* yang dilakukan dengan rumus $Gray = (0.299 * R + 0.587 * G + 0.114 * B)$ dimana If $Gray > 255$ Then $Gray = 255$ jika $Gray$ lebih besar 255 maka $Gray = 255$.

Flowchart hasil pengolahan *grayscale* seperti gambar 7.



Gambar 7. *Flowchart* metode *grayscale* metode *biner*

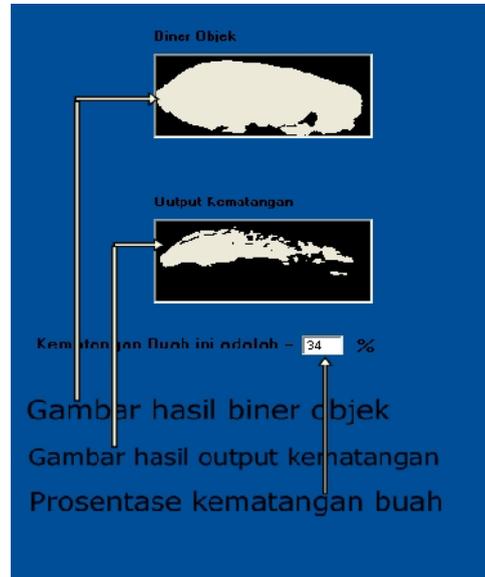
Pada citra hitam putih (*biner*) terdapat 256 level, artinya mempu-nyai skala “0” sampai “255” atau (0,255). Dalam hal ini nilai intensitas 0 menyatakan hitam,dan nilai intensitas 255 menyatakan putih. Adapun code untuk biner ini sebagai berikut :

```
Private Sub CmdBiner_Click()
Dim x As Integer, y As Integer
Dim p As Long
Dim R As Integer, G As Integer, B As Integer
Dim gray As Integer
Dim Objek As Integer
Dim count1 As Integer
Dim count2 As Integer
Dim n As Integer
For y = 0 To PicHasil.ScaleHeight - 1
  For x = 0 To PicHasil.ScaleWidth - 1
    p = GetPixel(PicHasil.hdc, x, y)
    R = p And &HFF
    G = (p \ &H100) And &HFF
    B = (p \ &H10000) And &HFF
    gray = (R + G + B) / 3
    Objek = gray

    If Objek > 37 Then
      Objek = 255
      count2 = count2 + 1
    Else
      Objek = 0
      SetPixel PicBiner.hdc, x, y, RGB(Objek, Objek, Objek)
    End If
  Next
Next
For y = 0 To PicHasil.ScaleHeight - 1
  For x = 0 To PicHasil.ScaleWidth - 1
    p = GetPixel(PicHasil.hdc, x, y)
    R = p And &HFF
    G = (p \ &H100) And &HFF
    B = (p \ &H10000) And &HFF
    gray = (R + G + B) / 3
    If gray > 120 Then
      gray = 255
      count1 = count1 + 1
    Else
      gray = 0
      SetPixel matang.hdc, x, y, RGB(gray, gray, gray)
    End If
    n = count1 / count2 * 100
  Next
Next
Form1.mtg = n
SavePicture PicBiner.Image, App.Path +
"\hasil\biner.bmp"
```

```
CmdOk.Enabled = True
End Sub
```

Pada proses *biner* ini dilakukan setelah hasil *grayscale* selesai, diproses secara *biner* untuk mendapatkan citra hitam dan putih secara jelas seperti gambar 8.



Gambar 8. Tampilan hasil metode *biner*

Disini, proses *biner* yang dilakukan ada 2 kali, yaitu Biner objek dan biner output kematangan.

Biner objek

Tampilan *biner* untuk seluruh objek. Untuk dapat menghasilkan tampilan *biner* objek *code* yang di gunakan sebagai berikut :

```
If Objek > 37 Then
  Objek = 255
  count2 = count2 + 1
Else
  Objek = 0
  SetPixel PicBiner.hdc, x, y, RGB(Objek, Objek, Objek)
End If
```

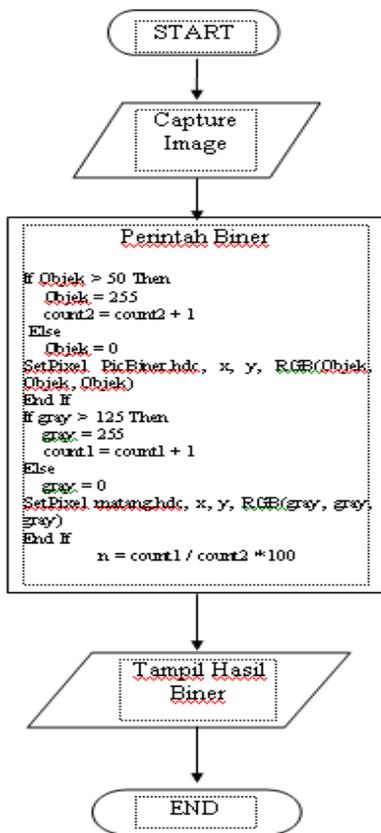
Biner Output kematangan

Tampilan *biner output* kematangan hanya menampilkan daerah kematangan dari seluruh objek. Untuk dapat menghasilkan tampilan *biner output* kematangan *code* yang digunakan sebagai berikut :

```

If gray > 120 Then
    gray = 255
    count1 = count1 + 1
Else
    gray = 0
    SetPixel matang.hdc, x, y, RGB(gray,
    gray, gray)
End If
    
```

Flowchart dari pengolahan metode biner seperti gambar 9.



Gambar 9. *Flowchart* metode biner

Persentase kematangan di peroleh dari luas daerah kematangan di bagi luas daerah objek dikalikan 100. *Output* data kematangan berupa angka desimal. Dalam perumusan *code* sebagai berikut :

$$n = \text{count1} / \text{count2} * 100$$

HASIL

Hasil dan Pengujian Metode *Cropping*

Hasil pengujian metode *cropping* dapat dilihat pada gambar 10.



(a) frame video



(b) frame hasil crop

Gambar 10. Hasil Pengujian Metode *Cropping*

Dari hasil pengujian, kotak *crop* harus bisa menangkap semua objek buah pepaya. Dengan presisi maka hasil yang diperoleh bisa terfokus pada objek buah pepaya. Pada gambar 10 (a) *frame* terlihat semuanya, untuk gambar 10 (b) merupakan *frame* hasil *cropping*. Resolusi *webcam* yang digunakan adalah 640 x 480.

Hasil dan Pengujian Metode *Grayscale*

Pada pengujian metode *gray-scale* ini *code* yang digunakan seperti gambar 11.

```

Private Sub CmdBtu_Click()
    Dim x As Integer, y As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    Dim gray As Integer
    For y = 0 To PicHasil.ScaleHeight - 1
        For x = 0 To PicHasil.ScaleWidth - 1
            p = GetPixel(FPicHasil.hdc, x, y)
            R = p And &HFF
            G = (p \ &H100) And &HFF
            B = (p \ &H1000) And &HFF
            gray = 0.299 * R + 0.587 * G + 0.114 * B
            If gray > 255 Then gray = 255
            SetPixel PicG1ah.hdc, x, y, RGB(gray, gray, gray)
        Next
    Next
    SavePicture PicG1ah.Image, App.Path + "\hasil\gray.bmp"
    CmdOK.Enabled = True

    MsgBox "Grayscale Selesai"
End Sub
    
```

Gambar 11. Code Pengujian Metode Grayscale

Hasil dari metode grayscale dapat dilihat pada gambar 12.



(a) image RGB hasil cropping



(b) image hasil metode grayscale

Gambar 12. Hasil Pengujian Metode Grayscale

Setelah pengujian metode *cropping* berhasil dijalankan, maka selanjutnya melakukan pengujian metode *grayscale*. Pengujian metode *grayscale* ini dengan merubah gambar RGB ke *gray level*. Ketika hasil dari metode *grayscale* seperti gambar 12 (b) dapat di analisa bahwa metode *grayscale* dapat berkerja dengan baik.

Hasil dan Pengujian Metode Biner

Code yang digunakan dalam metode *biner* seperti pada gambar 13.

```
Private Sub CmdBiner_Click()
Dim x As Integer, y As Integer
Dim p As Long
Dim R As Integer, G As Integer, B As Integer
Dim gray As Integer
Dim Objek As Integer
Dim count1 As Integer
Dim count2 As Integer
Dim n As Integer
For y = 0 To PicHasil.ScaleHeight - 1
For x = 0 To PicHasil.ScaleWidth - 1
p = GetPixel(PicHasil.hdc, x, y)
R = p And &HFF
G = (p \ &H100) And &HFF
B = (p \ &H10000) And &HFF
gray = (R + G + B) / 3
Objek = gray
If Objek > 37 Then
Objek = 255
count2 = count2 + 1
Else
Objek = 0
SetPixel matang.hdc, x, y, RGB(Objek, Objek, Objek)
End If
Next
Next
For y = 0 To PicHasil.ScaleHeight - 1
For x = 0 To PicHasil.ScaleWidth - 1
p = GetPixel(PicHasil.hdc, x, y)
R = p And &HFF
G = (p \ &H100) And &HFF
B = (p \ &H10000) And &HFF
gray = (R + G + B) / 3
If gray > 120 Then
gray = 255
count1 = count1 + 1
Else
gray = 0
SetPixel matang.hdc, x, y, RGB(gray, gray, gray)
End If
n = count1 / count2 * 100
Next
Next
Form1.mtg = n
SavePicture PicBiner.Image, App.Path + "\hasil\objek.bmp"
CmdCk.Enabled = True
End Sub
```

Gambar 13. Code Pengujian Metode Biner

Dalam pengujian *biner* ini mempunyai hasil 3 *output* yaitu: *output* objek *biner*, *output* daerah kematangan, *output* data persentase hasilnya seperti gambar 14.



(a) output objek biner



(b) output daerah kematangan

Kematangan Buah ini adalah = %

(c) output data prosentase

Untuk *output* daerah kematangan perhitungan diperoleh dari daerah matang per luas objek di kali 100. Dari hasil tersebut dapat di analisa bahwa pengujian biner adalah penentu dari sistem aplikasi kematangan buah karena persentase kematangan diperoleh dengan rumus $N = \text{Count1} / \text{Count2} * 100$ dan ketika hasil sudah seperti gambar 14. maka metode *biner* bekerja dengan baik.

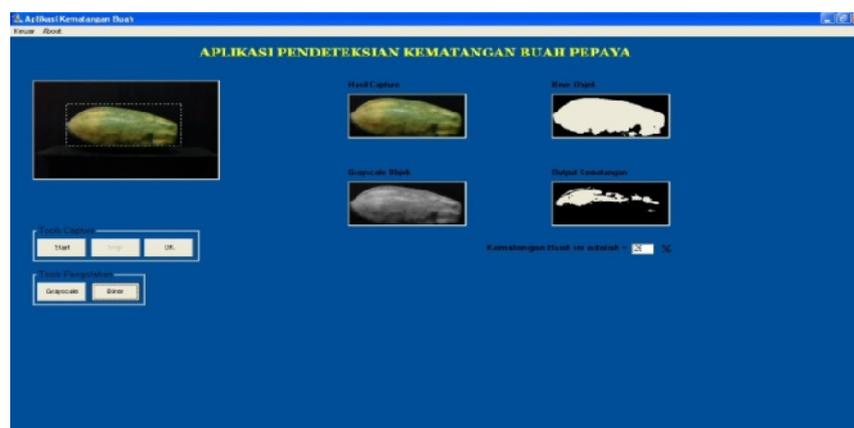
Hasil dan Pengujian Keseluruhan Sistem Aplikasi Kematangan Buah Pepaya

Pengujian secara keseluruhan adalah pengujian dengan berurutan jalannya

Gambar 14. Hasil Pengujian Metode Biner

aplikasi kematangan buah pepaya dari metode *cropping*, metode *grayscale*, metode *biner*. Untuk pengujian aplikasi kematangan buah pepaya secara keseluruhan seperti gambar 15.

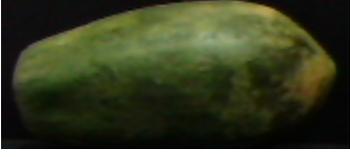
Aplikasi ini berjalan secara berurutan dari mulai metode *cropping*, metode *grayscale*, metode *biner*. Ketika ketiga metode itu berjalan secara berurutan maka aplikasi bekerja dengan baik. Pada pengujian keseluruhan ini menggunakan pepaya yang setengah matang dan memiliki *output* data prosentase sebesar 26%. Adapun hasil pengujian keseluruhan sistem dengan menggunakan berbagai kondisi objek buah pepaya seperti pada tabel 1.



Gambar 15. Hasil Pengujian Keseluruhan

Tabel 1. Data hasil pengujian software

No	Gambar	Hitung Manual	Hasil Software	Error
1.		0,88%	1%	0,12%
2.		9,37%	7%	-2,37%

3.		3%	10%	7%
4.		24,19%	46%	21,81%
5.		4,64%	7%	2,36%
6.		50,65%	68%	17,35%
7.		55,78%	87%	31,22%
8.		15,29%	17%	1,71%
9.		48,26%	72%	23,74%
10.		49,47%	68%	18,53%
<i>Error rata-rata</i>				12,14%

Dari pengujian keseluruhan sistem menggunakan 10 kondisi objek buah pepaya diperoleh *error* rata-rata sebesar 12,14%. Perolehan *error* rata-rata dilakukan dengan penghitungan jumlah total *error* dibagi jumlah total percobaan.

PEMBAHASAN

Berdasarkan pengujian terhadap 10 kondisi objek buah pepaya menunjukkan perhitungan data *manual* diperoleh dengan cara memperbesar *image* hasil metode *cropping* dengan 3,84 kali. Ukuran

yang dihasilkan 1,01mm X 1,01mm tiap *pixel* dengan ukuran *image* hasil *cropping* 173 X 77 *pixel*. Data perbandingan dilakukan dengan cara perhitungan data *manual*. Pada perhitungan data *manual* prosentase paling rendah pada 0,88% dengan kondisi buah pepaya mentah sedangkan hasil paling tinggi pada 55,78% pada percobaan ke-7 dengan kondisi buah pepaya matang. Dari hasil keseluruhan pengujian aplikasi pendeteksian kematangan buah pepaya menggunakan teknik segmentasi warna masih memiliki tingkat *error* data tinggi.

Hasil pengujian menunjukkan bahwa paling rendah pada percobaan ke-1 dengan hasil 1% oleh aplikasi pendeteksian kematangan buah dengan kondisi objek buah pepaya mentah. Hasil percobaan yang paling tinggi pada percobaan ke-7 dengan hasil 87% oleh aplikasi pendeteksian kematangan buah dengan kondisi objek buah pepaya matang.

KESIMPULAN

Dari hasil pengujian keseluruhan sistem, dapat diambil kesimpulan bahwa :

- 1) Sistem aplikasi pendeteksian kematangan buah pepaya dengan teknik segmentasi ini menggunakan tiga metode yaitu metode *cropping*, metode *gray-scale*, dan metode *biner*. Pada saat metode *cropping* akan dijalankan posisi *webcam* dan objek buah pepaya harus presisi serta proses pengolahan harus berurutan.
- 2) Pemisahan warna matang buah pepaya *image* hasil *cropping* menggunakan pengolahan citra metode *biner* yang memiliki *output image* hitam dan putih. Hitam memiliki nilai intensitas adalah 0 dan untuk putih memiliki nilai intensitas adalah 255.
- 3) Cara komunikasi antara *webcam* dan *microsoft visual basic 6.0* memerlukan *library* kamera aktif yaitu *avi-*

cap32.dll sehingga *webcam on* ketika aplikasi kematangan buah pepaya dijalankan.

- 4) Pencahayaan sangat mempengaruhi hasil untuk pembacaan objek buah pepaya, semakin terang cahaya yang masuk pada objek buah pepaya maka akan kelihatan *blur* dan jika terlalu gelap maka akan kelihatan hitam.
- 5) Pada saat pengujian keseluruhan sistem dengan percobaan 10 kondisi buah pepaya menghasilkan *error* rata-rata 12,14%.

DAFTAR RUJUKAN

- Arymurthy, Aniati M dan Suryana Setiawan. 1992. *Pengantar Pengolahan Citra*. Jakarta: ELEX MEDIA KOMPUTINDO.
- Basuki, achmad, Jozua F. Palandi dan Fatchurrochman. 2005. *Pengolahan Citra Digital Menggunakan Visual Basic*. Yogyakarta: GRAHA ILMU.
- Daryanto. 2004. *Belajar Komputer Visual Basic*. Bandung: YRAMA WIDYA.
- Fadlisyah. 2007. *Computer Vision dan Pengolahan Citra*. Yogyakarta: ANDI OFFSET.
- Fajri, achmad sulthon ainul. 2009. *Pembacaan Plat Nomer Kendaraan Menggunakan Metode Coding Berbasis Image Processing*. Proyek akhir tidak diterbitkan. Surabaya: Jurusan Teknik Telekomunikasi PENS-ITS.
- Gonzalez, Rafael C and Richard E. Woods. 2001. *Digital Image Processing*. 2nd Edition. Prentice Hall.
- Ichwannudin, mohammad. 2007. *Pencatatan Meter PLN Menggunakan Image Processing*. Proyek akhir tidak diterbitkan. Surabaya: Jurusan Teknik Tele-komunikasi PENS-ITS.
- Jahne, Bernd. 2005. *Digital Image Processing*. Netherlands: Springer.
- Rijal, yusron. 2008. *Deteksi Wajah Berbasis Segmentasi Model Warna*

- Menggunakan Template Matching Pada Objek Bergerak*. Makalah disajikan dalam Seminar Nasional Aplikasi Teknologi Informasi, Universitas Gajah Mada, Yogyakarta, 21 Juni.
- Sigit, riyanto. 2005. *Step by Step Pengolahan Citra Digital*. Yog-yakarta: ANDI OFFSET.
- Thiang dan Leonardus Indrotanoto. 2008. *Otomasi Pemisah Buah Tomat Berdasarkan Ukuran dan Warna Menggunakan Webcam Sebagai Sensor*. Makalah disajikan dalam Seminar Nasional Ilmu Komputer dan Aplikasinya, Universitas Kristen Petra, Surabaya, 27 Nopember.
- Wijaya, Marvin Ch dan Agus Priyono. 2007. *Pengolahan Citra Digital Menggunakan MATLAB*. Bandung: INFORMATIKA.
- www.ristek.go.id/pertanian/pepaya.pdf, diakses 15 desember 2009
- Young, Ian T , Jan J. Gerbrands and Lucas J. Van Vliet. 1998. *Fundamentals of Image Pro-cessing*. Netherlands: Delft Uni-versity of Technology.