

BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

a. Dataset Training

Dataset training berjumlah 30 buah apel fuji yang diukur tingginya dengan cara manual menggunakan penggaris, lalu diukur juga lebar buah apel fuji menggunakan penggaris. Kemudian ditimbang dengan menggunakan timbangan manual. Berikut kumpulan ukuran tinggi, lebar dan berat buah apel fuji tabel 4.1 *dataset training*.

Table 4.1 **Dataset Training**

| No | Tinggi | Lebar | Berat |
|-----|--------|-------|-------|
| 1. | 7 | 6,4 | 181 |
| 2. | 6,2 | 5,8 | 164 |
| 3. | 6,2 | 6,6 | 156 |
| 4. | 7 | 6,3 | 168 |
| 5. | 6,9 | 6,8 | 169 |
| 6. | 5,8 | 6,3 | 152 |
| 7. | 4,3 | 6,5 | 123 |
| 8. | 6,8 | 6,2 | 170 |
| 9. | 6,6 | 6,2 | 161 |
| 10. | 6,1 | 6,8 | 153 |
| 11. | 7 | 6,4 | 181 |
| 12. | 6,2 | 5,8 | 164 |

Table 4.2 Lajutan Dataset Training

| No | Tinggi | Lebar | Berat |
|-----|--------|-------|-------|
| 13. | 6,2 | 6,6 | 156 |
| 14. | 7 | 6,3 | 168 |
| 15. | 6,9 | 6,8 | 169 |
| 16. | 5,8 | 6,3 | 152 |
| 17. | 4,3 | 6,5 | 123 |
| 18. | 6,8 | 6,2 | 170 |
| 19. | 6,6 | 6,2 | 161 |
| 20. | 6,1 | 6,8 | 153 |
| 21. | 6,3 | 7,4 | 164 |
| 22. | 6,7 | 6,7 | 166 |
| 23. | 6,1 | 6,3 | 152 |
| 24. | 5,1 | 6 | 120 |
| 25. | 6,5 | 6,6 | 171 |
| 26. | 5,4 | 6,9 | 160 |
| 27. | 6,4 | 7 | 172 |
| 28. | 6,7 | 6,9 | 160 |
| 29. | 6,5 | 6,5 | 174 |
| 30. | 6,5 | 7,4 | 172 |

b. Data Testing

Dataset testing adalah buah apel fuji yang diletakkan dalam posisi berdiri dengan *background* berwarna putih, kemudian ditentukan sebanyak 10 citra buah apel fuji dengan masing-masing pengambilan foto sebanyak 1 kali dengan tinggi stiker 15 cm dan dengan *resolusi* 3024 x 4032 *pixel*. Contoh *dataset* yang digunakan pada penelitian ini ditunjukkan pada tabel 4.2




Table 4.3 Contoh Dataset

| No | Nama Citra | Keterangan | Citra |
|----|------------|---|--|
| 1 | Ap1.jpg | Tinggi : 5,8 cm Lebar : 7 cm Berat : 154 gram |  |
| 2 | Ap2.jpg | Tinggi : 6,2 cm Lebar : 6,6 cm Berat : 154 gram |  |



Table 4.2 Lanjutan Contoh Dataset

| No | Nama Citra | Keterangan | Citra |
|----|------------|---|--|
| 3 | Ap3.jpg | Tinggi : 6 cm Lebar : 7,3 cm Berat : 147 gram |  |
| 4 | Ap4.jpg | Tinggi : 5,5 cm Lebar : 8 cm Berat : 169 gram |  |
| 5 | Ap5.jpg | Tinggi : 6,6 cm Lebar : 8 cm Berat : 177 gram |  |

Table 4.2 Lanjutan Contoh Dataset

| No | Nama Citra | Keterangan | Citra |
|----|------------|---|--|
| 6 | Ap6.jpg | Tinggi : 6 cm Lebar : 7,1 cm Berat : 151 |  |
| 7 | Ap7.jpg | Tinggi : 5,4 cm Lebar : 6 cm Berat : 147 gram |  |
| 8 | Ap8.jpg | Tinggi : 5,3 cm Lebar : 6,2 cm Berat : 152 gram |  |

Tabel 4.2. Lanjutan Contoh Dataset

| No | Nama Citra | Keterangan | Citra |
|----|------------|---|---|
| 9 | Ap9.jpg | Tinggi : 5,8 cm Lebar : 5,6 cm Berat : 150 gram |  |
| 10 | Ap10.jpg | Tinggi : 5,7 cm Lebar : 6,3 cm Berat : 174 gram |  |

4.2. Hasil Analisis

Pada sub bab ini, ditunjukkan hasil uji coba sistem yang telah dilakukan dimulai dari *pre-processing*, segmentasi citra menggunakan metode *thresholding*, implementasi metode regresi linear, hitung tinggi dan lebar untuk menemukan berat buah apel fuji, hingga uji coba.

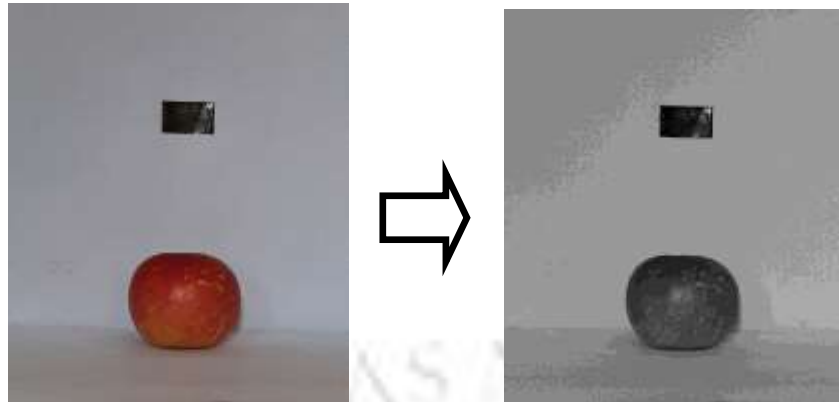


Gambar 4.1 Diagram Alir Sistem Berat Buah Apel Fuji

Cara kerja dari sistem ini yaitu mengambil gambar menggunakan kamera *smartphone* dan hasil dari pengambilan gambar tersebut menjadi masukan untuk sistem. Kemudian akan diproses sesuai tahap tahapnya.

4.1.1. Transformasi RGB to Grayscale

Pada bagian ini data yang diperoleh dari pengambilan *dataset* kemudian terlebih dahulu diolah untuk memudahkan proses selanjutnya. Di Tahap *pre-processing* langkah yang dilakukan adalah mentransformasi citra asli menjadi *grayscale*. Data yang berjumlah 10 citra diubah menjadi gambar keabuan. Adapun proses *pre-processing* di tunjukkan pada gambar 4.2. yang menampilkan citra asli dengan citra yang telah di transformasi.



Gambar 4.2 Transformasi Citra Asli Menjadi Grayscale

Kode :

```
Mat citra = Imgcodecs.imread(temp.getAbsolutePath());  
Mat apell = citra.clone();  
Imgproc.cvtColor(citra, citra, Imgproc.COLOR_BGR2GRAY);
```

Segmen Program 4.1 Merubah Citra Asli Menjadi Grayscale

4.1.2. Segmentasi *Thresholding*

Pemrosesan citra selanjutnya setelah tersegmentasi dengan jelas antara objek dengan *background*. Setelah itu citra diubah menjadi citra biner. Pada gambar 4.3. menunjukkan contoh hasil proses segmentasi.



Gambar 4.3 Hasil Segmentasi

Kode :

```
Imgproc.threshold(citra, citra, 0, 255, Imgproc.THRESH_OTSU);

BufferedImage biner = new BufferedImage(citra.width(),
    citra.height(), BufferedImage.TYPE_BYTE_GRAY);

// Get the BufferedImage's backing array and copy the pixels directly into it
byte[] dataz = ((DataBufferByte) biner.getRaster().getDataBuffer()).getData();
System.out.println(dataz);
citra.get(0, 0, dataz);

//String a = "hasilGray.jpg";
ImageIcon imgz = new ImageIcon(biner);
Image gbrz = imgz.getImage();
Image newimgz = gbrz.getScaledInstance(250, 430,
    java.awt.Image.SCALE_SMOOTH);
ImageIcon rezisez = new ImageIcon(newimgz);
jLabel4.setIcon(rezisez);

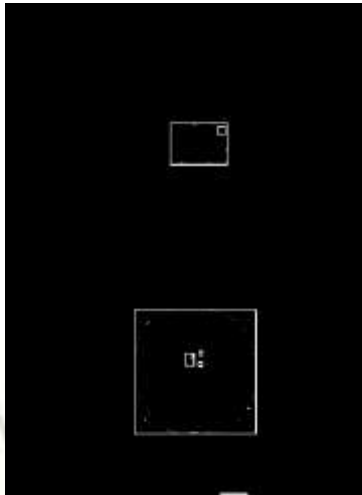
//Imgproc.threshold(citra, citra1, 148, 255, Imgproc.THRESH_BINARY);
Imgproc.medianBlur(citra, citra, 21);
Imgproc.Canny(citra, citra, 0, 255);

Mat hierarchy = new Mat();
List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
```

Segmen Program 4.2 Segmentasi Thresholding

4.1.3. Implementasi Deteksi Kontur

Input dari deteksi tepi objek adalah hasil dari segmentasi yang kemudian di deteksi tepi objeknya menggunakan metode kontur untuk menghasilkan garis kotak pada tepi objek. Pada gambar 4.4 merupakan salah satu contoh hasil deteksi tepi objek menggunakan metode kontur.



Gambar 4.4 Hasil Deteksi Kontur Citra

Kode :

```

Imgproc.findContours(citra, contours, hierarchy,
    Imgproc.RETR_TREE, Imgproc.CHAIN_APPROX_SIMPLE);

Mat drawing = Mat.zeros(citra.size(), CvType.CV_8UC1);

Rect rect;

double BK, c;
int berat_apel = 0, paling_tinggi = 0, paling_lebar = 0,
    kordinat_y = 0, kordinat_x = 0,
    paling_atas = citra.rows(), tinggi_palingAtas = 0;
double y2 = 0;
double x2 = 0;
String br;
Scalar color = new Scalar(255);
for (int i = 0; i < contours.size(); i++) {
    Imgproc.drawContours(drawing, contours, i, color);

    rect = Imgproc.boundingRect(contours.get(i));

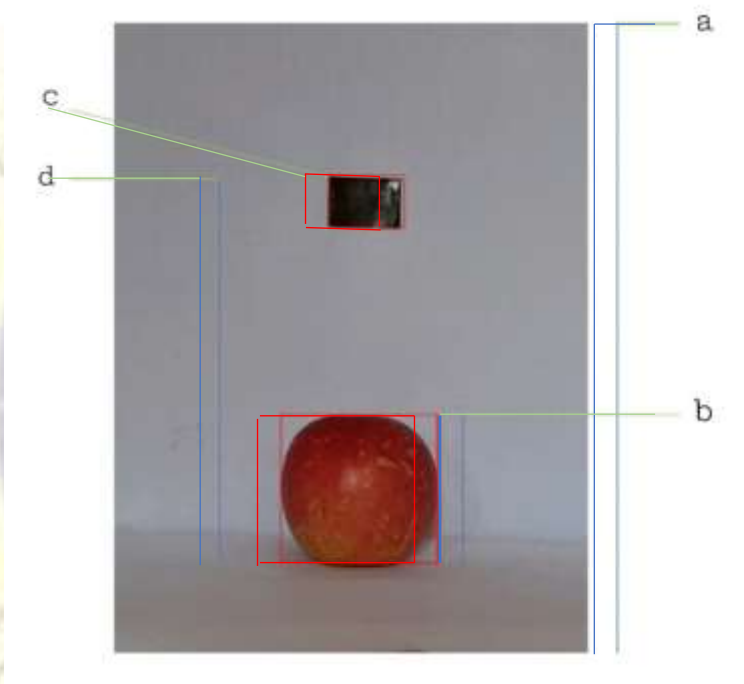
    Imgproc.rectangle(citra, rect.tl(), rect.br(), new Scalar(255), 9);
    int tinggi = rect.height;
    int lebar = rect.width;
}

```

Segmen Program 4.3 Implementasi Deteksi Kontur

4.1.4. Kalkulasi Rasio, Tinggi Dan Lebar

Pada penghitungan rasio objek yang dilakukan untuk mendapatkan hasil rasio tinggi dan lebar citra buah apel fuji, tinggi citra stiker, tinggi citra keseluruhan, dan tinggi pada objek yang sebenarnya yang nantinya dibutuhkan dalam proses konversi satuan piksel ke satuan *centimeter*. Berikut contoh penentuan rasio yang terdapat pada gambar 4.5.



Gambar 4.5 Hasil Pencarian Rasio

Kode :

```
if (paling_lebar < rect.width) {
    paling_lebar = rect.width;
    kordinat_x = rect.x;
}
if (paling_tinggi < rect.height) {
    paling_tinggi = rect.height;
    kordinat_y = rect.y;
    y2 = rect.br().y;
}

if (paling_atas > rect.y) {
    paling_atas = rect.y;
    tinggi_palingAtas = rect.height;
}
}

Imgcodecs.imwrite("hasil.jpg", citra);
```

Segmen Program 4.4 Mencari Rasio

Penjelasan gambar diatas sebagai berikut :

- Tinggi citra (satuan piksel)
- Tinggi citra manusia (satuan piksel)
- Tinggi stiker yang sebenarnya (satuan *centimeter*)
- Tinggi citra stiker (satuan piksel)

Masukan dari proses penghitungan tinggi dan lebar buah apel fuji ialah hasil dari pencarian rasio objek pada citra yang kemudian dilakukan penghitungan dengan rumus sesuai dengan tahap yang ada pada sub bab sebelumnya, yaitu pada rumus. Contoh penghitungan tinggi badan dari salah satu *dataset* citra sebagai berikut :

- tps = Jumlah piksel citra stiker (px)
- lpa = Lebar piksel buah apel fuji (px)
- tpa = Tinggi piksel buah apel fuji (px)
- ts = Tinggi objek stiker sebenarnya(cm)

- l_a = Lebar buah apel fuji (cm)
- t_a = Tinggi buah apel fuji (cm)
- s = Jumlah 1 piksel per cm (cm)

Dari contoh citra hasil kalkulasi rasio objek di atas, ada beberapa yang dapat diketahui antara lain, Tinggi Citra Stiker (tps), Tinggi Citra Buah Apel Fuji (tpa), Lebar Citra Buah Apel Fuji (lpa), dan Tinggi Stiker Yang Sebenarnya (ts) yang ditentukan tingginya 15 cm.

a. Rumus untuk menghitung Jumlah Piksel Per Centimeter (Cm)

- $1 \text{ px} = a \text{ (cm) ?}$

$$1 \text{ px} = \frac{\text{tps}}{\text{ts}}$$

ts

$$1 \text{ px} = s$$

b. Rumus untuk menghitung tinggi buah apel fuji :

- $t_a \text{ (cm) ?}$

$$t_a = \text{tpa} \times s$$

$$= t_a \text{ (cm)}$$

c. Rumus untuk menghitung tinggi buah apel fuji :

- $l_a \text{ (cm) ?}$

$$l_a = \text{lpa} \times s$$

$$= l_a \text{ (cm)}$$

Berikut ini contoh penghitungan berdasarkan salah satu *dataset* apel 1 :

Diketahui : tps = 2161 px

$$\text{tpa} = 834 \text{ px}$$

$$\text{lpa} = 905 \text{ px}$$

$$\text{ts} = 15 \text{ cm}$$

a. Menghitung Jumlah Pikel Per *Centimeter* (cm)

Sebelum melanjutkan ke penghitungan tinggi apel fuji (t_a) dan lebar buah apel fuji (l_a), terlebih dahulu menghitung jumlah dari 1 Pikel Per *Centimeter* (s), adapun contoh penghitungannya sebagai berikut:

• 1 px = s (cm) ?

$$s = \frac{\text{ts (cm)}}{\text{tps (px)}}$$

$$s = \frac{15 \text{ cm}}{2161 \text{ px}}$$

$$s = 0,0069412309 \text{ cm}$$

b. Menghitung Tinggi Buah Apel Fuji

Setelah didapat hasil dari jumlah Pikel Per *Centimeter* (s), barulah dilakukan penghitungan Tinggi Buah Apel Fuji (t_a) sesuai rumus sebagai berikut:

• t_a (cm) ?

$$t_a = 843 \text{ px} \times 0,0069412309 \text{ cm}$$

$$t_a = 5,85 \text{ cm}$$

c. Menghitung Lebar Buah Apel Fuji

Setelah didapat hasil dari jumlah Piksel Per *Centimeter* (s), barulah dilakukan penghitungan Lebar Buah Apel Fuji (ta) sesuai rumus sebagai berikut:

- la (cm) ?

$$la = 905 \text{ px} \times 0,0069412309 \text{ cm}$$

$$la = 6,28 \text{ cm}$$

Dari contoh penghitungan tinggi dan lebar buah apel fuji tersebut, telah didapatkan hasil tinggi buah apel fuji yaitu sekitar **5,85 cm** dan lebar buah apel fuji yaitu sekitar **6,28 cm**.

4.1.5. Hasil Berat Buah Apel Fuji

Berikut ini adalah hasil dari tahap uji coba sistem deteksi berat badan berdasarkan tinggi dan lebar dengan menggunakan metode regresi linear yang menghasilkan berat buah apel fuji yang ditunjukkan pada gambar 4.6.



Gambar 4.6 Hasil Berat Dari Program

Berdasarkan Gambar 4.6, adalah gambar hasil program aplikasi deteksi berat

buah apel fuji berdasarkan tinggi dan lebar. Untuk gambar pertama adalah gambar citra apel fuji asli, selanjutnya gambar kedua yaitu citra *grayscale*, yang ketiga gambar hasil segmentasi *thresholding* dan untuk gambar keempat yaitu implementasi deteksi kontur dengan *drawcontours*. Sedangkan untuk hasil berat buah apel fuji 152.072 gram adalah hasil akhir berat buah apel fuji berdasarkan tinggi dan lebar.

Dibawah ini rumus untuk menghitung rata – rata selisih perhitungan manual dan hasil sistem. Kemudian juga menghitung rata – rata akurasi.

$$\text{Rata-rata} = \frac{\text{Jumlah Nilai Yang Sesuai}}{\text{Jumlah Total Gambar}}$$

a. Menghitung rata-rata Akurasi gram

$$\begin{aligned} \text{Rata – rata akurasi} &= 1,93 \text{ gram} + 0,43 \text{ gram} + 2,33 \text{ gram} + 17,35 \text{ gram} + \\ & 24,77 \text{ gram} + 0,08 \text{ gram} + 4,27 \text{ gram} + 7,7 \text{ gram} + \\ & 0,27 \text{ gram} + 20,83 \text{ gram} \\ & \underline{\hspace{10em}} \\ & \hspace{10em} 10 \\ & = \frac{87,96}{10} = 8,79 \text{ cm} \end{aligned}$$

b. Menghitung rata-rata Akurasi %

$$\begin{aligned} \text{Rata – rata akurasi} &= 98,74 \% + 99,62 \% + 98,41 \% + 89,73 \% + 86,00 \% + \\ & 94,64 \% + 97,09 \% + 96,25 \% + 99,82 \% + 88,02 \% \\ & \underline{\hspace{10em}} \\ & \hspace{10em} 10 \end{aligned}$$

$$= \frac{948,23}{10} \% = 94,83 \%$$

10

Table 4. 4 Nilai Akurasi Berat Buah Apel Fuji

| <i>DataSet</i> | Berat Buah Apel Fuji (gram) | | Selisih (gram) | Akurasi (%) |
|----------------|--|----------------|---------------------------|--------------------|
| | Asli | Sistem | | |
| Apel 1 | 154 gram | 152,07 gram | 1,93 gram | 98,74 % |
| Apel 2 | 154 gram | 153,43 gram | 0,43 gram | 99,62 % |
| Apel 3 | 147 gram | 144,67 gram | 2,33 gram | 98,41 % |
| Apel 4 | 169 gram | 151,65 gram | 17,35 gram | 89,73 % |
| Apel 5 | 177 gram | 152,23 gram | 24,77 gram | 86,00 % |
| Apel 6 | 151 gram | 142,92 gram | 8,08 gram | 94,64 % |
| Apel 7 | 147 gram | 142,73 gram | 4,27 gram | 97,09 % |
| Apel 8 | 152 gram | 146,30 gram | 7,7 gram | 96,25 % |
| Apel 9 | 150 gram | 149,73 gram | 0,27 gram | 99,82 % |
| Apel 10 | 174 gram | 153,17 gram | 20,83 gram | 88,02 % |

Berdasarkan Tabel 4.2, dari salah satu data uji yang terdiri dari buah apel fuji dengan jarak stiker 15 cm diperoleh rata-rata selisih berat buah apel fuji antara pengukuran manual dengan hasil sistem yaitu 8,79 gram dan rata-rata *Approximate Value* (ApV) 94,83 %.

Berikut perbandingan antara jarak stiker 10 cm, 13 cm, 15 cm, 18 cm, dan 20 cm.

Table 5.4 Perbandingan Berat Buah Apel Fuji Berdasarkan Tinggi Stiker

| DataSet | Berat Buah Apel Fuji (gram) | | Selisih (gram) | Akurasi (%) |
|------------------------|--------------------------------|-------------|-------------------|----------------|
| | Asli | Sistem | | |
| Tinggi Stiker 10 cm | 159 gram | 227 gram | 68 gram | 142,76 % |
| Tinggi Stiker 13 cm | 159 gram | 141 gram | 18 gram | 88,67 % |
| Tinggi Stiker 15 cm | 159 gram | 166 gram | 7 gram | 95,78% |
| Tinggi Stiker 18 cm | 159 gram | 135 gram | 24 gram | 84,90% |
| Tinggi Stiker 20 cm | 159 gram | 121 gram | 38 gram | 76,10% |

Berdasarkan Tabel 4.2, dari 10 data uji yang terdiri dari buah apel fuji dengan jarak stiker 10 cm, 13 cm, 15 cm, 18 cm, dan 20 cm diperoleh rata-rata selisih berat buah apel fuji antara pengukuran manual dengan hasil yang paling mendekati dengan hasil sistem yaitu dengan tinggi stiker 15 cm.