

Sistem Computer Vision Berbasis Metode Position Averaging Point untuk Pemilah Ikan Lele

Aditya Kurniawan, Kholilatul Wardani

Politeknik Kota Malang

Kompleks Pendidikan Internasional Tlogowaru No 3, telp/fax (0341) 754088

e-mail: aditya@poltekom.ac.id

Abstrak – Pada penelitian ini dibuat aplikasi machine vision yang menggunakan algoritma averaging point untuk mengatasi kesalahan pengukuran panjang ikan lele secara visual dikarenakan posisi tubuh (lengkungan) objek ukur yang bervariasi. Algoritma ini bekerja dengan cara menentukan titik tengah, batas kiri dan kanan objek, serta menarik sebuah garis lurus yang melewati titik tengah untuk mengukur panjang objek.

Rata-rata akurasi pengukuran dengan menggunakan algoritma ini adalah 96.5%. Simpangan / error terbesar adalah 2.5 mm pada lengkungan dengan penyusutan dari ujung ke ujung sebesar 30%. Berdasarkan perhitungan Pearson Correlation, disimpulkan bahwa lengkungan memiliki hubungan sebab akibat yang kecil terhadap error dalam pengukuran panjang ikan dengan konstanta Pearson sebesar 0.26.

Kata kunci: machine vision, averaging point, ikan lele, Pearson

1. Pendahuluan

Machine Vision merupakan perkembangan dari aplikasi pengolahan citra (*image processing*) yang memiliki output berupa aksi dari aktuator. Image yang diidentifikasi oleh sensor / kamera akan dikirimkan ke komputer (yang dalam hal ini sebagai otak) yang kemudian diproses sedemikian rupa, selanjutnya komputer memberikan perintah kepada aktuator untuk melakukan sebuah perintah / aksi. Dengan machine vision, maka akan mempermudah proses produksi untuk menghasilkan benda/barang yang cepat dan reliabel. Dalam hal pemilah ikan lele yang memanfaatkan sistem *computer vision* untuk pemilahan ikan nya, didapatkan beberapa permasalahan diantaranya yaitu; (1) posisi ikan lele yang sedang diukur menggunakan algoritma image harus dalam posisi lurus. (2) posisi ikan lele yang melengkung akan mengurangi akurasi ukuran panjang ikan pada pengukuran berbasis image. Karena hal itulah maka perlu adanya pendekatan baru dalam sistem pemilah ikan berbasis *computer vision* yang dapat mengatasi permasalahan tersebut. Penelitian ini bertujuan untuk melihat bagaimana metode averaging point dapat meminimalisir kesalahan ukur panjang ikan yang berbentuk melengkung.

Algoritma yang digunakan pada penelitian ini adalah algoritma dasar feature *extraction*, yang mana pemisahan antara komponen warna (merah, hijau, dan biru) dipisahkan melalui algoritma sebagai berikut :

$$(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) = \left(\left(\langle P_{(x,y)} \text{ MOD}(2^{16}) \rangle \text{ MOD}(2^8) \right), \left(\text{Int} \left\langle \frac{\langle P_{(x,y)} \text{ MOD}(2^{16}) \rangle}{(2^8)} \right\rangle \right), \left(\text{Int} \left\langle \frac{P_{(x,y)}}{2^{16}} \right\rangle \right) \right) \quad (1)$$

Dengan:

$P(x,y)$: adalah nilai mentah pada pixel kolom y baris x yang bernilai 0 - 16581375.

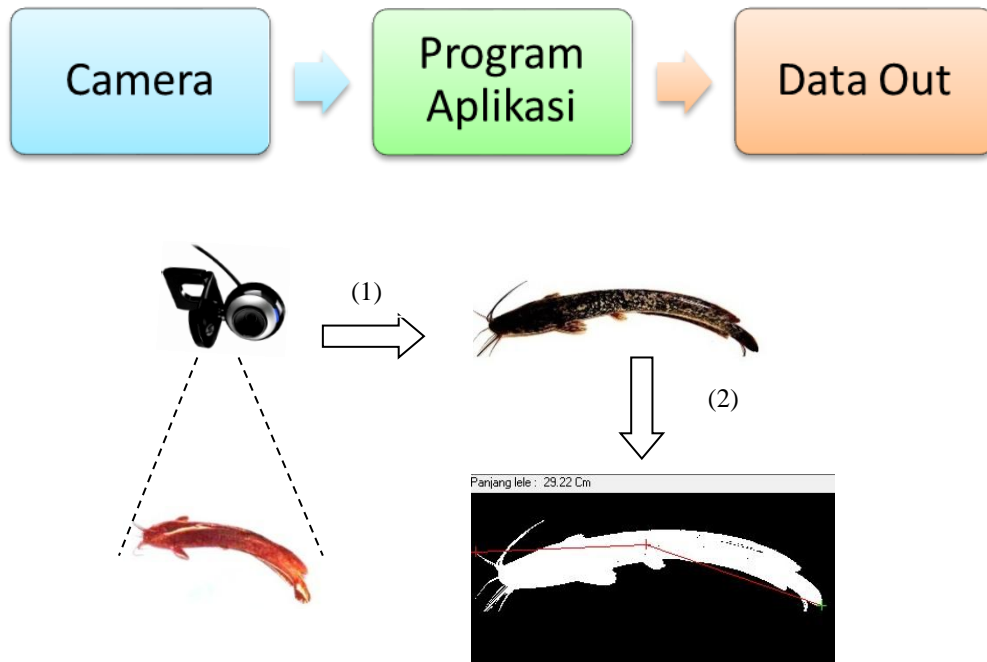
$R(x,y)$: adalah nilai red (merah) pada pixel kolom y baris x yang bernilai 0-255.

$G(x,y)$: adalah nilai green (hijau) pada pixel kolom y baris x yang bernilai 0-255.

$B(x,y)$: adalah nilai blue (biru) pada pixel kolom y baris x yang bernilai 0-255.

2. Metodologi Penelitian

Metode penelitian yang digunakan adalah dengan melakukan capture pada citra ikan lele dengan variasi lengkungan yang beragam, dan membandingkannya dengan ukuran lele yang sebenarnya (dalam mm). *Capture* dengan menggunakan metode *averaging ponit* dilakukan sebanyak 10 kali capture pada panjang dan lengkungan ikan yang sama. Perulangan dilakukan dengan variasi lengkungan sebanyak 9 variasi. Setup penelitian dapat digambarkan seperti pada gambar Gambar1.



Gambar 1. Setup penelitian

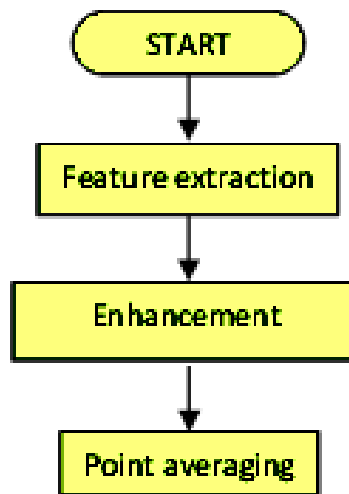
Tabel 1. Variabel penelitian

Variabel independen	Variabel dependen
Panjang lele	Pixel

Tabel 2. Variabel penelitian

No	Panjang lele	Lengkungan
1	40mm	90%
2	40mm	80%
3	40mm	70%
4	40mm	60%
5	40mm	50%
6	40mm	40%
7	40mm	30%
8	40mm	20%
9	40mm	10%

3. Hasil dan Pembahasan



Gambar 2. flowchart sistem

Pada penelitian ini digunakan proses algoritma yang memiliki 3 langkah yaitu *feature extraction*, *image enhancement*, dan *point averaging*. Dalam proses *feature extraction*, data mentah dalam pixel x,y di ekstraksi menjadi 3 komponen warna yaitu R(x,y);G(x,y) dan B(x,y) dengan rentang nilai antara 0-255. Algoritma yang digunakan adalah sebagai berikut:

$$(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) = \left(\left(\langle P_{(x,y)} \text{ MOD}(2^{16}) \rangle \text{ MOD}(2^8) \right), \left(\text{Int} \left\langle \frac{\langle P_{(x,y)} \text{ MOD}(2^{16}) \rangle}{(2^8)} \right\rangle \right), \left(\text{Int} \left\langle \frac{P_{(x,y)}}{2^{16}} \right\rangle \right) \right) \quad (2)$$

Listing program yang digunakan adalah dengan membuat *function* sebagai berikut :

```

Function BacaDanEkstrakAll(indexSimpanDataOutput As Integer, pictureboxYangAkanDibaca As PictureBox) As String
  For i = 1 To pictureboxYangAkanDibaca.ScaleWidth
    For j = 1 To pictureboxYangAkanDibaca.ScaleHeight
      Pixel(indexSimpanDataOutput, i, j) = pictureboxYangAkanDibaca.Point(i, j)
      If Pixel(indexSimpanDataOutput, i, j) < 0 Then Pixel(indexSimpanData, i, j) = 0
    Next j
  Next i
  For i = 1 To pictureboxYangAkanDibaca.ScaleWidth
    For j = 1 To pictureboxYangAkanDibaca.ScaleHeight
      RED(indexSimpanDataOutput, i, j) = Int((Pixel(indexSimpanDataOutput, i, j) Mod (2 ^ (2 * 8))) Mod (2 ^ 8))
      GREEN(indexSimpanDataOutput, i, j) = Int((Pixel(indexSimpanDataOutput, i, j) Mod (2 ^ (2 * 8))) / (2 ^ 8))
      BLUE(indexSimpanDataOutput, i, j) = Int((Pixel(indexSimpanDataOutput, i, j)) / (2 ^ (2 * 8)))
    Next j
  Next i
End Function
  
```

Setelah itu, proses enhancement yaitu brightness dan contrast yang digunakan untuk memisahkan objek (ikan) dengan background warna dengan menggunakan algoritma berikut:

$$R_{bright(x,y)}, G_{bright(x,y)}, B_{bright(x,y)} = R_{(x,y)} + Cb, G_{(x,y)} + Cb, B_{(x,y)} + Cb \quad (3)$$

Dengan :

$R_{bright(x,y)}$, $G_{bright(x,y)}$, $B_{bright(x,y)}$ = adalah nilai komponen warna pada pixel kolom y baris x setelah operasi pencerahan / brightness

Setelah itu, proses *contrast* sebagai proses *enhancement* final yang bertujuan untuk memisahkan objek dan background dengan menggunakan algoritma berikut :

$$R_{cont(x,y)}, G_{cont(x,y)}, B_{cont(x,y)} = (R_{(x,y)} + Cb) \times Cc, (G_{(x,y)} + Cb) \times Cc, (B_{(x,y)} + Cb) \times Cc \quad (4)$$

Dengan :

$R_{cont(x,y)}$, $G_{cont(x,y)}$, $B_{cont(x,y)}$ = adalah nilai komponen warna pada pixel kolom y baris x setelah operasi pencerahan / brightness dan contrast

Cc = adalah contanta contrast yang bernilai 0.1 – 1.9

Listing program yang digunakan adalah dengan membuat *function* sebagai berikut :

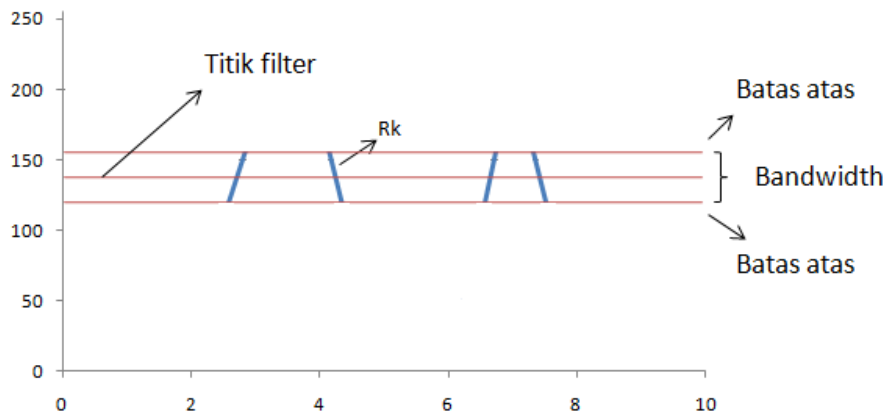
Listing program brightness

```
Function PointProcessingBrightnessAll(IndexSimpanDataInput As Integer, indexSimpanDataOutput As Integer,
pictureboxYangAkanDiproses As PictureBox, ContsBrightness As Integer)
    For i = 1 To pictureboxYangAkanDiproses.ScaleWidth
        For j = 1 To pictureboxYangAkanDiproses.ScaleHeight
            RED(indexSimpanDataOutput, i, j) = RED(IndexSimpanDataInput, i, j) + ContsBrightness
            GREEN(indexSimpanDataOutput, i, j) = GREEN(IndexSimpanDataInput, i, j) + ContsBrightness
            BLUE(indexSimpanDataOutput, i, j) = BLUE(IndexSimpanDataInput, i, j) + ContsBrightness
        Next j
    Next i
End Function
```

Listing program contrast

```
Function PointProcessingContrastAll(IndexSimpanDataInput As Integer, indexSimpanDataOutput As Integer,
pictureboxYangAkanDiproses As PictureBox, ContsContrast10sd190 As Double)
    Dim ContsContrast As Double
    ContsContrast = ContsContrast10sd190 / 100
    For i = 1 To pictureboxYangAkanDiproses.ScaleWidth
        For j = 1 To pictureboxYangAkanDiproses.ScaleHeight
            RED(indexSimpanDataOutput, i, j) = RED(IndexSimpanDataInput, i, j) * ContsContrast
            GREEN(indexSimpanDataOutput, i, j) = GREEN(IndexSimpanDataInput, i, j) * ContsContrast
            BLUE(indexSimpanDataOutput, i, j) = BLUE(IndexSimpanDataInput, i, j) * ContsContrast
        Next j
    Next i
End Function
```

Setelah melewati proses *enhancement*, maka proses selanjutnya adalah colour filtering untuk memisahkan background dengan objek. Perhitungan ini dilakukan dengan teknik sebagai berikut:



Gambar 3. grafik colour filtering

Listing program yang digunakan adalah dengan membuat function sebagai berikut:

Function FilterWarnaMonotoneThresholdingHasilPutihResiduHitam(IndexSimpanDataInput As Integer, indexSimpanDataOutput As Integer, pictureboxYangAkanDiproses As PictureBox, Rthres As Integer, Gthres As Integer, Bthres As Integer, Colourband As Integer)

```

For i = 1 To pictureboxYangAkanDiproses.ScaleWidth
  For j = 1 To pictureboxYangAkanDiproses.ScaleHeight
    If RED(IndexSimpanDataInput, i, j) > (Rthres - Colourband / 2) And RED(IndexSimpanDataInput, i, j) <
(Rthres + Colourband / 2) And GREEN(IndexSimpanDataInput, i, j) > (Gthres - Colourband / 2) And
GREEN(IndexSimpanDataInput, i, j) < (Gthres + Colourband / 2) And BLUE(IndexSimpanDataInput, i, j) > (Bthres -
Colourband / 2) And BLUE(IndexSimpanDataInput, i, j) < (Bthres + Colourband / 2) Then
      RED(indexSimpanDataOutput, i, j) = 255
      GREEN(indexSimpanDataOutput, i, j) = 255
      BLUE(indexSimpanDataOutput, i, j) = 255
    Else
      RED(indexSimpanDataOutput, i, j) = 0
      GREEN(indexSimpanDataOutput, i, j) = 0
      BLUE(indexSimpanDataOutput, i, j) = 0
    End If
  Next j
Next i

```

End Function

Point averaging adalah sebuah proses yang dapat memberikan koordinat titik tengah objek dengan melakukan perhitungan *averaging* (rata rata) pada pixel yang berkaitan dengan objek untuk menentukan koordinat tengah yang tepat. Perhitungan ini dilakukan dengan algoritma awal sebagai berikut:

Untuk $x = 0$
 If $R_{(0,y)} \text{ AND } G_{(0,y)} \text{ AND } B_{(0,y)} = 0$ then $X(0) = X(0) + 1$

Untuk $y = 0$

If $R_{(x,0)}$ AND $G_{(x,0)}$ AND $B_{(x,0)} = 0$ then $Y(0) = Y(0) + 1$

Maka dari itu akan didapatkan $X(0) - X(pW)$ dan $Y(0) - Y(pH)$

Dengan :

pW = Picture width

pH = Picture height

Untuk mendapatkan titik tengah untuk masing masing koordinat, maka digunakan algoritma :

$$X = \frac{\sum_{i=0}^{pW} X(i)}{Tpo}, Y = \frac{\sum_{i=0}^{pH} Y(i)}{Tpo} \quad (5)$$

Dengan :

Tpo = Total pixel objek

Listing program yang digunakan adalah dengan membuat function sebagai berikut :

```
Function HitungCentreObject(IndexSimpanDataInput As Integer, pictureboxYangAkanDiproses As PictureBox)
    Dim HitungPixelPutihH, HitungPixelPutihV As Double
    HitungPixelPutihH = 0
    HitungPixelPutihV = 0
    For i = 1 To pictureboxYangAkanDiproses.ScaleWidth - 1
        For j = 1 To pictureboxYangAkanDiproses.ScaleHeight - 1
            If RED(IndexSimpanDataInput, i, j) = 255 Then HitungPixelPutihH = HitungPixelPutihH + i
        Next j
    Next i
    For j = 1 To pictureboxYangAkanDiproses.ScaleWidth - 1
        For i = 1 To pictureboxYangAkanDiproses.ScaleHeight - 1
            If RED(IndexSimpanDataInput, i, j) = 255 Then HitungPixelPutihV = HitungPixelPutihV + j
        Next i
    Next j
    HitungCentreObjectHc = Int(HitungPixelPutihH / (HitungPixelPutih(IndexSimpanDataInput,
pictureboxYangAkanDiproses)))
    HitungCentreObjectVc = Int(HitungPixelPutihV / (HitungPixelPutih(IndexSimpanDataInput,
pictureboxYangAkanDiproses)))
End Function
```

Penelitian ini menghasilkan data sebagai berikut :

Tabel 3. Variabel penelitian

No	Panjang ikan (sebenarnya)	Lengkungan	Pixel	Panjang ikan (terukur)	error (dalam %)
1	40 mm	90%	234	39 mm	2.5
2	40 mm	80%	226	37.6 mm	6
3	40 mm	70%	237	39.4 mm	1.5
4	40 mm	60%	227	37.8 mm	5.5
5	40 mm	50%	237	39.5 mm	1.25
6	40 mm	40%	234	38.9 mm	2.75
7	40 mm	30%	225	37.5 mm	6.25
8	40 mm	20%	235	39.1 mm	2.25
9	40 mm	10%	239	39.8 mm	0.5

Maksimum panjang terukur : 39.8 mm
 Minimum panjang terukur : 37.5 mm
 Maksimum error : 6.25 %
 Minimum error : 0.5 %
 Rata rata panjang terukur : 38.7 mm
 Rata rata error : 3.1 %

Dengan menggunakan *pearson correlation coefficient* dengan membandingkan variabel lengkungan ikan dengan error pengukuran maka didapatkan :

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

Dengan
 variable X = lengkungan ikan
 variable Y = error pengukuran
 maka didapatkan
 r = 0.26

Tabel 4. pearson coefficient

Correlation	Negative	Positive
None	-0.09 to 0.0	0.0 to 0.09
Small	-0.3 to -0.1	0.1 to 0.3
Medium	-0.5 to -0.3	0.3 to 0.5
Large	-1.0 to -0.5	0.5 to 1.0

4. Kesimpulan

Algoritma *averaging point* dapat mendeteksi ukuran panjang ikan lele dengan akurasi rata rata sebesar 96.9 %. Simpangan / *error* terbesar adalah 2.5 mm pada lengkungan dengan

penyusutan dari ujung ke ujung sebesar 30%. Berdasarkan perhitungan *Pearson Correlation*, dapat disimpulkan bahwa lengkungan memiliki hubungan sebab akibat yang kecil terhadap error dalam pengukuran panjang ikan (konstanta = 0.26). hal ini mengindikasikan algoritma averaging point relatif baik dalam melakukan pengukuran terhadap ikan dengan posisi yang bervariasi dibandingkan dengan tanpa menggunakan algoritma ini.

Penelitian ini secara keseluruhan belum mendalami tentang variasi iluminasi yang memiliki tingkat kemungkinan tinggi dalam kemunculan error pengukuran panjang ikan. Pada penelitian selanjutnya diharapkan dapat mendalami hubungan tingkat iluminasi dengan tingkat error pengukuran.

Daftar Pustaka

- [1] Lim, Kelvin K. P. *Clarias batu*, a New Species of Catfish (Teleostei: Claridae) from Pulau Tioman, Peninsular Malaysia. *The Raffles Bulletin of Zoology*. Malaysia. 1999; 6: 157–167
- [2] Misimi, E., Mathiassen., Erikson, U. Computer Vision – Based Sorting of Atlantic Salmon (*Salmo salar*) Fillets According to Their Color Level. *Journal of Food Science*. -. 2007; 72:1
- [3] Ng, Heok Hee. *Clarias insolitus*, a new species of clariid catfish (Teleostei: Siluriformes) from southern Borneo. *Zootaxa*. -. 2003; 284:1-8
- [4] Sudarto, Teugels, Guy G, Pouyaud, Laurent. Description of a New Clariid Catfish, *Clarias pseudonieuhofii* from West Borneo (Siluriformes: Clariidae). *Zoological Studies*. -. 2004; 43:8-19
- [5] White, D.J. Automated Measurement of Species and Length of Fish by Computer Vision. University of Aberdeen Journal. United Kingdom. 2006;
- [6] Kenneth R, Castleman. *Digital Image Processing*. 1st. - : Prentice Hall. 1996: 245.