Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 1, No. 2, Juli 2005

ISSN 0216 - 0544

DETEKSI CITRA BERWARNA MENGGUNAKAN KOMBINASI MATRIK MASK DELAPAN ARAH DAN THRESHOLD

Arif Muntasa, S.Si, MT

Jurusan Teknik Informatika
Fakultas Teknik Universitas Trunojoyo

Email : arifmuntasa@trunojoyo.ac.id

ABSTRAK

Deteksi tepi objek dari citra gray scale telah banyak diteliti. Deteksi tepi objek dari citra gray scale mempunyai komputasi yang lebih sederhana dibandingkan dengan citra berwarna. Hal ini disebabkan citra gray scale setiap pixelnya mempunyai nilai R=G=B. Deteksi tepi dengan menggunakan citra berwarna, dilakukan dengan mengonvolusi matrik mask terhadap pixel untuk nilai R, G dan B. Konvolusi dilakukan sebanyak delapan kali, hasil konvolusi dicari nilai yang paling besar. Hasil Konvolusi setiap pixel untuk masing-masing R, G dan B yang paling besar dibandingkan dengan *threshold* R, G dan B. Jika minimal dua dari tiga nilai R, G dan B lebih besar dari *threshold* R, G dan B, maka akan ditandai pixel tersebut sebagai tepi objek. Hasil deteksi objek tergantung pada matrik mask dan *threshold*. Semakin kecil *threshold* akan mengakibatkan tepi objek menjadi lebih tebal dan bisanya membentuk region kecil yang sebenarnya bukan tepi objek. Sedangkan *threshold* yang besar akan mengakibatkan tepi objek kelihatan lebih tipis, namun seringkali tepi objek terputus. Hasil deteksi objek dapat digunakan sebagai sata awal untuk menganalisa suatu objek, misalnya bentuk objek.

Kata Kunci : *Gray Scale, Threshold, Mask*, Deteksi Tepi

1. PENDAHULUAN

Citra dapat direpresentasikan dengan menggunakan matrik 2 dimensi. Nilai masing-masing matrik yang terletak pada posisi baris = x dan kolom =y, seperti tampak pada matrik berikut ini

.

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0.M-1). \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(0.M-1) \\ f(2,0) & f(2,1) & f(2,2) & \dots & f(0.M-1) \\ . & & . & & \\ f(N-1,0) & f(N-1,1) & f(N-1,2) & \dots\dots & f(N-1.M-1) \\ f(N,0) & f(N,1) & f(N,2) & \dots\dots & f(N.M-1) \end{pmatrix}$$

f(x,y) merupakan nilai pixel pada posisi (x,y). Untuk Citra *gray scale,* merupakan citra 8 bit dimana untuk setiap pixelnya f(x,y) mempunyai nilai R=G=B, nilai kisaran untuk masing-masing pixel adalah antara L_{Min} = 0 samapai $L_{Max} = (2^8) - 1 = 255$. Sedangkan pada Citra Berwarna merupakan citra 24 bit merupakan kombinasi R=8 bit, G=8 bit dan B=8 bit.

Sehingga citra Berwarna dapat memiliki jutaan warna atau 256*256*256 = 16.777.216 kombinasi warna. Untuk setiap pixelnnya mempunyai komposisi warna R = 0.299, G=587 dan B=0.114 [Ioannis Pitas, *"Digital Image Processing Algorithms"*, Prentice Hall, Singapore, 1993, hal 31]. Untuk masing-masing pixel berarti memuat nilai R, G dan B. Secara prinsip citra *gray scale* dan citra berwarna, mempunyai representasi yang sama.

Pada proses deteksi tepi citra *gray scale* ataupun citra berwarna, perlu dipahami konsep relasi antar pixel. Konsep relasi antar pixel didasarkan pada tetangga terdekatnya. Misalkan pixel P pada posisi (x,y) dinotasikan dengan f(x,y) mempunyai 4 tetangga, yang dinotasikan dengan $N_4(P]$. Masing-masing adalah F[x-1,y], F[x, y-1], F[x+1,y] dan F[x,y+1].

	F[x-1,y]	
F[x,y-1]	**f[x,y]**	F[x,y+1]
	F[x+1,y]	

Sedangkan 4 tetangga diagonal P yang dinotasikan dengan $N_D[P]$ adalah F[x-1,y-1], F[x-1, y+1], F[x+1,y-1] dan F[x+1,y+1].

F[x-1,y-1]		F[x-1,y+1]
	f[x,y]	
F[x+1,y-1]		F[x+1,y+1]

Gabungan $N_4[P]$ dan $N_D[P]$ disebut dengan 8 tetangga P yang dinotasikan dengan $N_8[P]$.

F[x-1,y-1]	F[x-1,y]	F[x-1,y+1]
F[x,y-1]	**f[x,y]**	F[x,y+1]
F[x+1,y-1]	F[x+1,y]	F[x+1,y+1]

Konsep relasi antar pixel tersebut digunakan sebagai dasar proses konvolusi terhadap citra yang akan diproses untuk menentukan tepinya. Pertama-tama, citra dikonvolusi dengan menggunakan matrik mask delapan arah. Hasil konvolusi kemudian diambil nilai maksimalnya. Misalkan hasil konvolusi adalah $G_1[f(x,y)]$, $G_2[f(x,y)]$, $G_3[f(x,y)]$, $G_4[f(x,y)]$, $G_5[f(x,y)]$, $G_6[f(x,y)]$, $G_7[f(x,y)]$ dan $G_8[f(x,y)]$, maka nilai maksimalnya adalah

$$G[f(x,y]) = \text{Max}\,(GR_i[f(x,y] \mid i=1,2,\ldots,8) \quad \ldots\ldots\ldots(1)$$

karena citra yang dikonvolusi merupakan citra berwarna, maka nilai maksimal dihitung untuk masing-masing R,G dan B. Sehingga persamaan tersebut menjadi

$$\mathbf{GR}[f(x,y]) = \text{Max}\,(\mathbf{GR}_i[f(x,y] \mid i=1,2,\ldots,8) \quad \ldots\ldots\ldots(2)$$

$$\mathbf{GG}[f(x,y]) = \text{Max}\,(\mathbf{GG}_i[f(x,y] \mid i=1,2,\ldots,8) \quad \ldots\ldots\ldots(3)$$

$$\mathbf{GB}[f(x,y]) = \text{Max}\,(\mathbf{GB}_i[f(x,y] \mid i=1,2,\ldots,8) \quad \ldots\ldots\ldots(4)$$

Jika minimal dua dari tiga persamaan memenuhi., maka nilai GR[f(x,y]), GG[f(x,y]) dan GB[f(x,y]) akan dipakai pada proses selanjutnya. Hasil konvolusi yang paling besar dibandingkan dengan nilai *threshold*. Nilai *threshold* diberikan untuk masing-masing R,G maupun B. Adapun persamaannya dapat dilihat pada persamaan 5, 6 dan 7

$$\text{fRed} = \begin{cases} 255 & \text{Jika } \mathbf{GR}[f(x,y]) > R \\ 0 & \text{Lainnya} \end{cases} \quad \ldots\ldots\ldots(5)$$

$$\text{fGrren} = \begin{cases} 255 & \text{Jika } \mathbf{GG}[f(x,y]) > G \\ 0 & \text{Lainnya} \end{cases} \quad \ldots\ldots\ldots(6)$$

$$fBlue = \begin{cases} 255 & \text{Jika } GB[f(x,y]) > B \\ 0 & \text{Lainnya} \end{cases} \quad \ldots\ldots\ldots(7)$$

Hasil perbandingan akan dianggap sebagai tepi objek, apabila memenuhi minimal dua dari tiga perbandingan yang dilakukan.

Pada penelitian ini, matrik mask yang dipakai adalah Kirsch, Prewitt dan sobel [Dwayne Phillips, *"Image Processing In C"*, Prentice Hall, United States Of America, 1994, hal 171] seperti terlihat pada tabel matrik dibawah ini :

Tabel 1. Matrik Mask Kirsch, Prewitt dan Sobel

Arah	Kirsch	Prewitt	Sobel
1	5 5 5 -3 0 -3 -3 -3 -3	1 1 1 1 -2 1 -1 -1 -1	1 2 1 0 0 0 -1 -2 -1
2	-3 5 5 -3 0 5 -3 -3 -3	1 1 1 1 -2 -1 1 -1 -1	2 1 0 1 0 -1 0 -1 -2
3	-3 -3 5 -3 0 5 -3 -3 5	1 1 -1 1 -2 -1 1 1 -1	1 0 -1 2 0 -2 1 0 -1
4	-3 -3 -3 -3 0 5 -3 5 5	1 -1 -1 1 -2 -1 1 1 1	0 -1 -2 1 0 -1 2 1 0
5	-3 -3 -3 -3 0 -3 5 5 5	-1 -1 -1 1 -2 1 1 1 1	1 2 1 0 0 0 -1 -2 -1
6	-3 -3 -3 5 0 -3 5 5 -3	-1 -1 1 -1 -2 1 1 1 1	-2 -1 0 -1 0 1 0 1 2
7	5 -3 -3 5 0 -3 5 -3 -3	-1 1 1 -1 -2 1 -1 1 1	-1 0 1 -2 0 2 -1 0 1
8	5 5 -3 5 0 -3 -3 -3 -3	1 1 1 -1 -2 1 -1 -1 1	0 1 2 -1 0 1 -2 -1 0

2. ALGORITMA

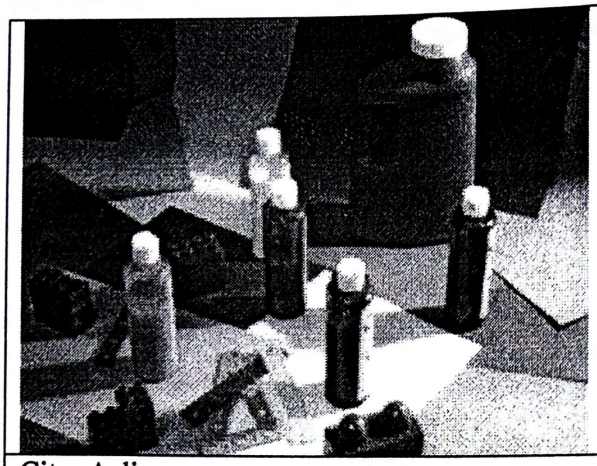
Algortima pada proses deteksi tepi ini, penulis paparkan dalam bentuk pseducode, seperti dibawah ini:
Pseudocode selengkapnya adalah sebagai berikut

```
//baca gambar mulai dari kiri atas sampai bawah
for x←0 to Width do begin
  for y←0 to Height do Begin
    //Lakukan Konvolusi Mulai Matrik 1 sampai ke 8
    for k←1 to 8 do begin
      for i←1 to 3 do begin
        for j←1 to 3 do begin
          CellsK [i,j] =M [i,j]
        end for j
      end for i
      //Konvolusi Setiap Pixel
      TotR←0; TotG←0; TotB←0
      for col←1 to 3 do begin
       for row←1 to 3 do begin
         R ← ImgInput[X+col-2, Y+row-2]
         G ← ImgInput[X+col-2, Y+row-2]
         B ← ImgInput[X+col-2, Y+row-2]
         TotR ← TotR + CellsK [col,row] * R
         TotG ← TotG + CellsK [col,row] * G
         TotB ← TotB + CellsK [col,row] * B
       end for row
      end for col
      //Cek Apabila hasil konvolusi melebihi nilai 255
      if |TotR| > 255 then
        ValueRK ← 255
      else
        ValueRK ← TotR
      end
      if |TotG| > 255 then
        ValueGK ← 255
      else
        ValueGK ← TotR
      end
      if |TotB| > 255 then
        ValueBK ← 255
      else
        ValueBK ← TotR
      end
    end for k

    GR←0; GG←0; GB←0;
    //Bandingkan Hasil Konvolusi Mulai dari Matrik 1 sampai 8
    for M←1 to 8 do begin
     if (ValueRM>GR) and (ValueGM>GG) and (ValueBM>GB) then begin
      GR←ValueRM
      GG←ValueGM
      GB←ValueBM
     end if
     if (ValueRM>GR) and (ValueGM>GG) and (ValueBM<GB) then begin
      GR←ValueRM
      GG←ValueGM
```

```
      GB←ValueB_M
    end if
    if (ValueR_M>GR) and (ValueB_M>GB) and (ValueG_M<GG) then begin
      GR←ValueR_M
      GG←ValueG_M
      GB←ValueB_M
    end if
    if (ValueG_M>GG) and (ValueB_M>GB) and (ValueR_M<GR) then begin
      GR←ValueR_M
      GG←ValueG_M
      GB←ValueB_M
    end if
  end for M

  //Bandingkan dengan nilai threshold
  if GR>ThresholdR then
      fRed ← 255
  else
     fRed ← 0
  end if
  if GG>ThresholdG then
      fGreen ← 255
  else
     fGreen ← 0
  end if
  if GB>ThresholdB then
      fBlue ← 255
  else
     fBlue ← 0
  end if

  // tandai tepi objek
  if fRed ← 255 and fGreen ← 255 and fBlue ← 255 then
      wHasil ← 255
  else if fRed ← 255 and fGreen ← 255 and fBlue ← 0 then
      wHasil ← 255
  else if fRed ← 255 and fGreen ← 0 and fBlue ← 255 then
      wHasil ← 255
  else if fRed ← 0 and fGreen ← 255 and fBlue ← 255 then
      wHasil ← 255
  else
      wHasil ← 0
  end if

  //Gambarkan hasil Objek
  ImgOutput(x,y) ← SetPixel[wHasil]
 end for y
end for x
```

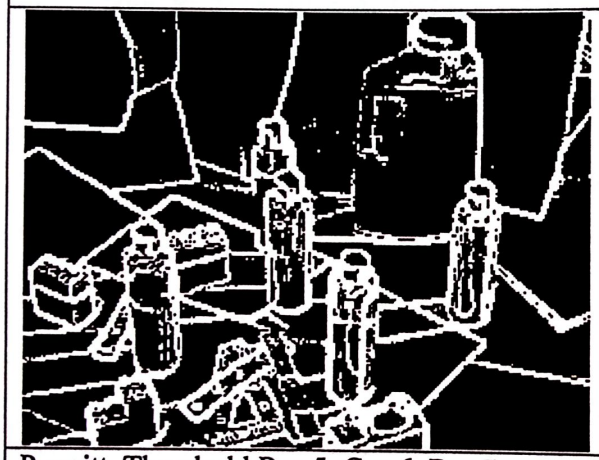
3. UJI COBA

Uji coba yang penulis lakukan adalah dengan menggunakan citra berwarna 24 bit dengan menggunakan matrik mask Kirsch, Prewitt dan Sobel dikombinasikan dengan threshold untuk masing-masing r,g dan b nilainya tidak sama. Nilai threshold dirubah dua kali. Nilai threshold kecil dan nilai threshold yang agak besar. Hasil selengkapnya dapat dilihat pada gambar berikut ini :

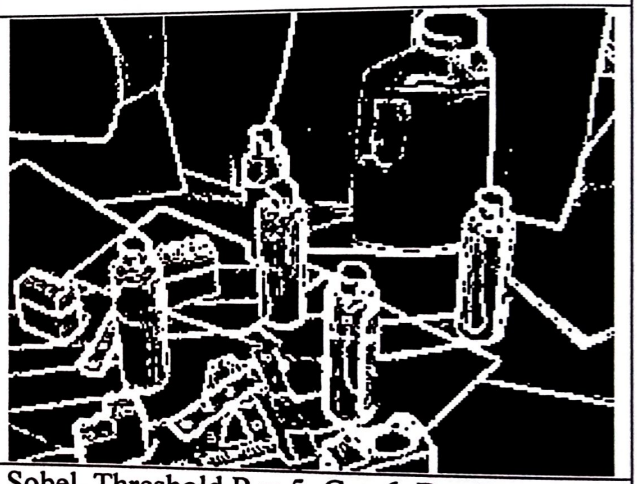


Citra Asli

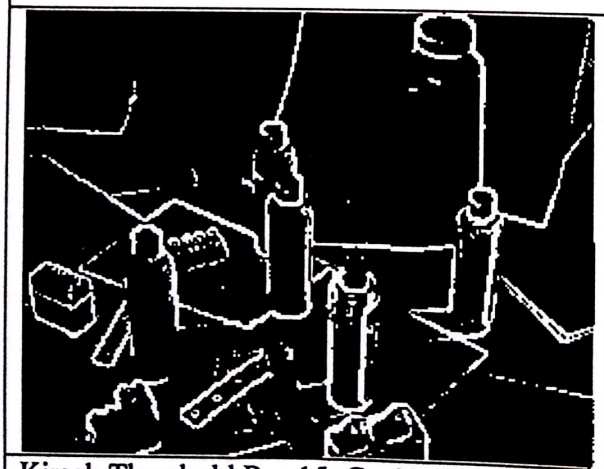
Kirsch, Threshold R = 5, G = 6, B = 7



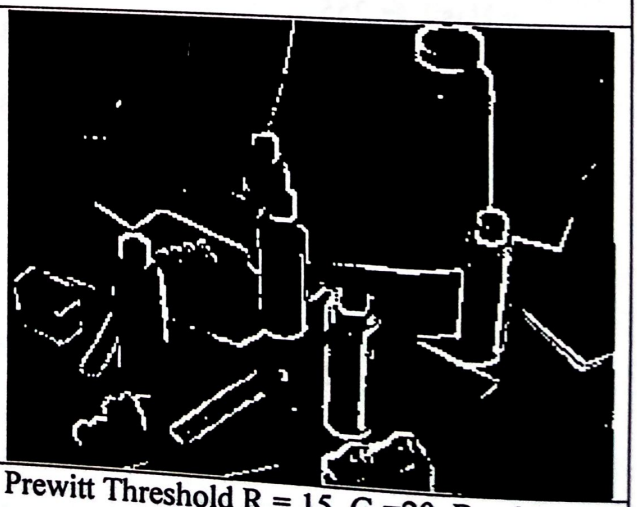
Prewitt, Threshold R = 5, G = 6, B = 7



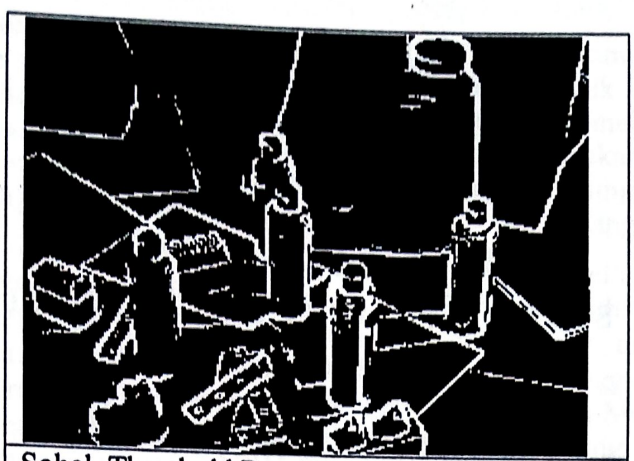
Sobel. Threshold R = 5, G = 6, B = 7



Kirsch Threshold R = 15, G =20, B = 25



Prewitt Threshold R = 15, G =20, B = 25



Sobel. Threshold R = 15, G =20, B = 25

Gambar 1. Hasil Semua Uji Coba

Dari hasil uji coba tersebut dapat dianalisis, bahwa dengan menggunakan threshold yang kecil, maka akan menghasilkan citra dengan tepi objek yang agak tebal, kecenderungan tepi objek putus menjadi sangat kecil dan menghasilkan objek-objek kecil. Hal ini disebabkan nilai hasil konvolusi lebih besar dari threshold, sehingga dianggap tepi objek namun sebenarnya bukan tepi objek. Sedangkan dengan menggunakan threshold yang agak besar akan mnyebabkan tepi objek lebih tipis, kecenderungan tepi objek putus menjadi sangat besar dan objek kecil yang dihasilkan lebih sedikit.

4. KESIMPULAN

Hasil deteksi tepi dengan menggunakan matrik mask 8 arah seperti Kirsrh, Prewitt dan Sobel juga sangat dipengaruhi oleh threshold. Nilai threshold yang terlalu kecil akan menghasilkan objek-objek kecil yang sebenarnya bukan tepi objek. Sedangkan dengan menggunakan threshold yang besar akan menyebabkan tepi objek cenderung putus.

5. SARAN

Deteksi tepi citra tersebut dapat diperbaiki lagi dengan menambahkan satu algoritma edge linking, yang berfungsi untuk menghubungkan edge yang terputus.

6. REFERENSI

1. Ioannis Pitas, *"Digital Image Processing Algorithms"*, Prentice Hall, Singapore, 1993
2. Dwayne Phillips, *"Image Processing In C"*, Prentice Hall, United States Of America, 1994
3. Andreas Koschan, *"A Comparative Study On Color Edge Detection "*, Proceedings 2nd Asian Conference on Computer Vision ACCV'95, Singapore, 5-8 December 1995, Vol. III, pp. 574-578.
4. Mark A. Ruzon Carlo Tomasi, *"Color Edge Detection with the Compass Operator"*, IEEE Conference on Computer Vision and Pattern Recognition '99, Volume 2, pages 160-166, June 1999.