

PERANCANGAN PENGOLAHAN CITRA SISTEM PERHITUNGAN TOTAL OBJEK DI KASIR

Lola

*Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta,
lola.rezak@gmail.com*

Abstrak

Perkembangan teknologi yang pesat, banyak mempengaruhi aspek kehidupan sehari-hari, termasuk dengan berkembangnya tata cara orang berbelanja di supermarket yang mana banyak menerapkan *self checkout*. Terdapat jenis supermarket yang menjual seluruh barang dengan harga yang sama, contohnya seperti *dollar tree*. Perhitungan total harga barang belanja dapat dengan mudah dilakukan dengan menghitung total barang yang dibeli lalu dikalikan dengan suatu harga. Dengan memanfaatkan metode pengolahan citra, dibuatlah suatu sistem perhitungan kasir yang mudah untuk toko satu harga dengan mengimplementasikan program mendeteksi jumlah objek dari suatu citra.

Kata kunci : Jumlah Objek, Pengolahan Citra, Kasir.

1. PENDAHULUAN

Masyarakat perlu memenuhi kebutuhan sehari-harinya, mulai dari yang bersifat primer hingga sekedar untuk memenuhi keinginan. Walaupun sekarang sudah ramai belanja melalui daring, masih banyak masyarakat yang lebih memilih meluangkan waktu berbelanja di toko fisik (*offline*). Menurut laporan Capgemini's Smart Stores, menghabiskan waktu dalam antrian panjang ketika waktu membayar adalah keluhan utama konsumen di toko. Oleh sebab itulah pembeli semakin menyukai pembayaran mandiri. Dengan adanya *self checkout*, pelanggan mendapatkan pengalaman di dalam toko yang lebih baik dan proses *checkout* yang lebih cepat tanpa antrian. Selain permasalahan tersebut, pembayaran mandiri juga meningkatkan kenyamanan pelanggan dengan memberikan pembeli lebih banyak pilihan dan kontrol selama berbelanja.

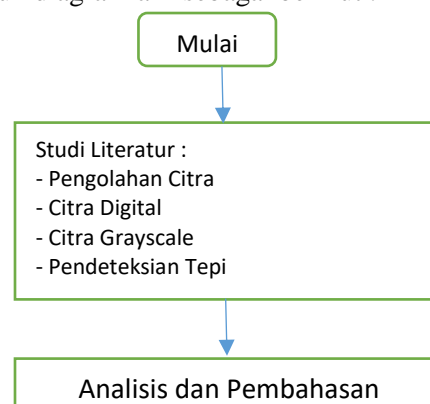
Dengan adanya pembayaran mandiri, supermarket juga dapat mengalokasikan sumber daya dan area yang digunakan kasir untuk keperluan lainnya. Dikarenakan banyaknya keuntungan yang didapatkan oleh kedua belah pihak yaitu pelanggan dan pemilik supermarket, maka pembayaran mandiri sebaiknya diimplementasikan. Saat ini kebanyakan pembayaran mandiri dilakukan dengan cara pembeli melakukan *scan* untuk setiap barang yang dibeli. Cara *checkout*

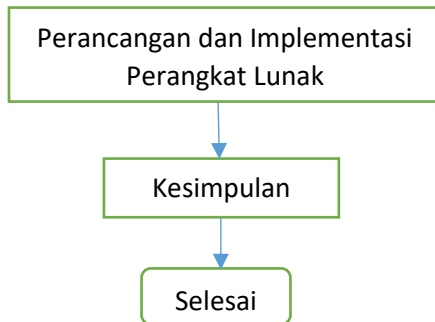
tersebut tergolong masih memakan waktu, oleh karena itu akan lebih mudah dan cepat jika *checkout* dilakukan hanya dengan pelanggan menaruh barang-barang yang dibeli diatas meja. Sistem kemudian akan menghitung jumlah barang yang dibeli dan menghitung total harga yang perlu dibayarkan.

Makalah ini fokus pada metode pengolahan citra yang digunakan untuk membangun sistem perhitungan jumlah objek. Sistem akan mengambil sebuah citra barang belanjaan setelah hasil proses menghasilkan perhitungan total biaya belanjaan.

2. METODOLOGI

Metodologi penelitian digambarkan dalam bentuk diagram alir sebagai berikut :





Gambar 1. Diagram Alir Metodologi Penelitian

Sumber :

<https://www.researchgate.net/publication/33823569>
 5 Metode-
 Metode Penelitian Dalam Penulisan Jurnal Ilmiah
 h Elektronik

3. HASIL DAN PEMBAHASAN

3.1 HASIL

Untuk contoh penyelesaian masalah, citra masukan yang digunakan untuk diproses oleh program yang dibangun adalah sebagai berikut:



Gambar 2. Citra Masukan
 Sumber : Munir, R, 2022.

Agar program yang dibangun dapat menghitung jumlah objek pada citra, citra dikonversi dahulu menjadi citra *grayscale* agar lebih mudah untuk diproses. Pada program ini, digunakan fungsi konversi citra berwarna ke citra *grayscale* dari Matlab. Berikut kode yang digunakan dan hasil citra *grayscale*.

```
I = rgb2gray(imread("gambarinput.jpg"))
```



Gambar 3. Hasil Citra Grayscale
 Sumber : Munir, R, 2022.

Selanjutnya, untuk mengetahui bagian mana saja yang merupakan objek dan bukan *background*, dilakukan pendeteksian tepi terhadap citra grayscale. Untuk mendeteksi tepi digunakan fungsi bawaan Matlab. Terdapat beberapa pilihan operator yang dapat digunakan seperti Canny, Log, Prewitt, Sobel, dan Roberts. Berikut kode yang digunakan dan hasil pendeteksian tepi.

```
edge_image = edge(I, 'Canny');
```

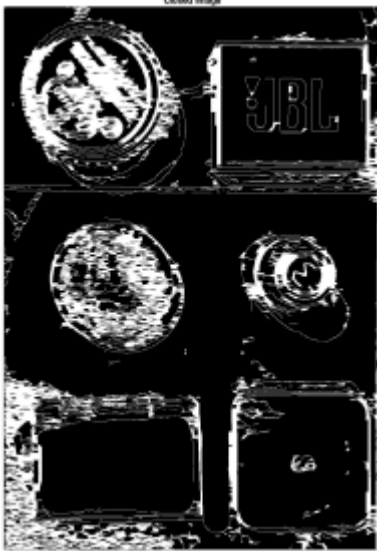


Gambar 4. Hasil Pendeteksian Tepi
 Sumber : Munir, R, 2022.

Agar lokasi objek lebih dapat dipastikan dan dipisahkan dari background, dilakukan beberapa aksi tambahan yaitu menyambungkan tepi-tepi yang tidak terhubung. Proses ini dilakukan menggunakan fungsi *imclose* dari

Matlab. Fungsi tersebut melakukan penutupan morfologis pada gambar skala abu-abu atau biner menggunakan suatu elemen struktur. Terdapat beberapa jenis elemen struktur yang dapat digunakan seperti diamond, disk, octagon, line, rectangle, dan square. Pada program ini ditutup tepi yang tidak terhubung menggunakan elemen garis (*line*) dengan panjang 10. Berikut kode yang digunakan dan hasil penutupan tepi.

```
se = strel('line',10,0); opened_image =
imclose(edge_image, se);
```



Gambar 5. Hasil Penutupan Tepi
Sumber : Munir, R, 2022.

Setelah tepi dari setiap objek tersambung dengan jelas, untuk membedakan objek dan latar belakang dilakukan pengisian dari tepi objek menggunakan fungsi *imfill* dari Matlab. Berikut kode yang digunakan dan hasil pengisian tepi.

```
filled_image = imfill(closed_image,'holes');
```



Gambar 6. Hasil Pengisian Tepi
Sumber : Munir, R, 2022.

Perbedaan antara objek dan background sudah terlihat cukup jelas, namun masih terdapat beberapa area kecil yang merupakan hasil deteksi tepi dari derau pada citra. Agar area kecil tersebut tidak mengganggu hasil perhitungan jumlah objek, dilakukan penghapusan area dengan erosi. Hasil citranya akan berupa nilai terkecil dari pixel yang berada disekitarnya. Proses ini menggunakan fungsi *imopen* dari Matlab dan menggunakan elemen struktur *square*. Berikut kode yang digunakan dan hasil penghapusan objek kecil.

```
se = strel('square',50);
opened_image = imopen(filled_image, se);
```



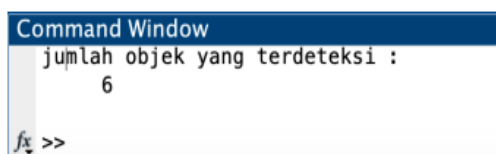
Gambar 7. Hasil Penghapusan Objek Kecil
Sumber : Munir, R, 2022.

Setelah seluruh objek terpisah secara baik dengan background, maka dapat dilakukan perhitungan jumlah objek berdasarkan jumlah area terpisah yang ada pada citra. Perhitungan menggunakan fungsi berlabel dari Matlab. Berikut kode yang digunakan dan hasil perhitungan jumlah objek.

```
[L, num]=bwlabel(opened_image, 4);
```

3.2 PEMBAHASAN

Pada hasil pengujian pada sub bab 3.1 diatas, sudah didapatkan hasil yang diharapkan. Hasil output program sebagai berikut :



```
Command Window
jumlah objek yang terdeteksi :
    6
f_x >>
```

Gambar 8. Hasil Output Program
Sumber : Munir, R, 2022.

4. KESIMPULAN

Banyak hal yang dapat diterapkan pada kehidupan sehari-hari dari pembelajaran pengolahan citra, salah satunya adalah pengaplikasian pengolahan citra (*image processing*) dan pendeteksian tepi (*edge detection*) untuk menyelesaikan permasalahan untuk menghitung total objek pada kasir *self checkout*. Berdasarkan hasil pembangunan program dan percobaan dengan citra masukan diatas, program berhasil menghitung jumlah objek yang ada pada citra. Hasil ini kemudian dapat dikalikan dengan suatu harga untuk menjadi hasil perhitungan yang nanti ditampilkan pada kasir.

Dari percobaan diatas juga ditemukan beberapa kelemahan dari program, seperti terdeteksinya bayangan dari objek. Dimana jika bayangan dari suatu objek mengenai objek lain, dapat menyebabkan program menganggap kedua objek tersebut merupakan satu kesatuan yang nantinya dapat berdampak pada hasil perhitungan. Program juga masih belum bisa menghasilkan segmentasi objek sesuai dengan

bentuk aslinya, dikarenakan efek dari *background* yang terdapat derau.

DAFTAR PUSTAKA

1. Meghanathan Natarajan, (2021). Pseudo Code for Breadth First Search (BFS). https://www.researchgate.net/figure/Pseudo-Code-for-Breadth-First-Search-BFS_fig11_266008323.
2. Munir, R, (2022). Pendeteksian Tepi. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2019-2020/13-Pendeteksian-Tepi.pdf>