

Duplicator for Find the Real Object Game

Louis Riemenn / 13519016
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519016@std.stei.itb.ac.id

Abstract—Perkembangan teknologi yang pesat banyak mempengaruhi kehidupan sehari-hari manusia. Salah satu perkembangan yang sedang terjadi adalah perkembangan dalam industri hiburan, salah satunya adalah permainan. Pada saat ini ada banyak permainan yang diimplementasikan dengan bantuan automatic system sehingga dapat memudahkan pembuat permainan dalam membuat sebuah permainan.

Kata Kunci – Objek, Duplikasi Objek, Permainan

I. PENDAHULUAN

Seiring dengan perkembangan jaman. Perilaku, Tindakan, bahkan kebiasaan orang pada umumnya akan ikut berubah. Pada era serba digital seperti sekarang ini, ada banyak hal yang dapat dilakukan dengan teknologi, khususnya dalam hal *image editing*. Misalnya saja, dalam kehidupan bersosial media, banyak orang melakukan *image editing* untuk memberikan gambar yang membuat orang lain memiliki ekspresi atau emosi yang diinginkan oleh *editor* setelah mereka melihat hasil dari gambar yang telah diubah (*edit*). Salah satu perubahan gambar yang cukup disukai adalah pencampuran tubuh-tubuh hewan seperti gambar di bawah ini.



Gambar 1.1 Hewan Campuran

Pada gambar di atas diperlihatkan gambar seekor bebek dengan kepala mirip babi dan seekor kuda dengan kepala menyerupai hiu. Gambar ini merupakan gambar yang diperoleh melalui hasil *editing* yang secara sengaja merubah beberapa bagian dari gambar.

Selain pada gambar, *editing* juga kerap dilakukan pada video, seperti halnya dalam pembuatan film yang banyak dilakukan saat ini.



Gambar 1.2 Real Life Vs Film

Sebagian besar film-film dengan genre sci-fi dibuat dengan teknologi *editing* dengan memanfaatkan *green screen*. Proses *editing* dari kedua gambar sebelumnya memiliki satu kesamaan umum, yaitu keduanya memanfaatkan suatu objek dari gambar yang berbeda untuk memberikan kesan baru pada gambar yang berbeda. Dalam melakukan *image editing* seperti pada gambar gambar di atas, diperlukan banyak Langkah. Namun dari banyak langkah yang ada, salah satu cara sederhana yang diperlukan adalah menduplikasi objek dari sebuah gambar. Objek yang telah diduplikat kemudian dapat diproses baik sebagian maupun seluruhnya, untuk bisa ditambahkan ke dalam sebuah gambar yang berbeda.

Biasanya, pengambilan, penduplikasian, dan pengolahan objek ini dilakukan untuk berbagai macam hal khususnya dalam bidang seni, seperti contoh-contoh di atas. Namun duplikasi objek juga bisa dilakukan untuk sebuah permainan seperti contohnya permainan untuk menemukan objek yang asli dari sekumpulan objek.

Makalah ini akan berfokus pada metode pengolahan citra yang dapat digunakan untuk bisa melakukan duplikasi objek yang ada pada sebuah gambar (citra) dan akan dimanfaatkan untuk membuat permainan "Find the Real Object" yang mana permainan ini berfokus pada menemukan objek yang asli (bukan hasil penambahan) pada sebuah citra..

II. LANDASAN TEORI

A. Pengolahan Citra Digital

Pengolahan citra adalah proses mengolah sebuah citra menjadi citra lain dengan berbagai cara dan dengan tujuan tertentu. Menurut Efford (2000), pengolahan citra digital adalah istilah yang umum digunakan untuk berbagai Teknik

yang keberadaannya adalah untuk memanipulasi dan memodifikasi citra dengan menggunakan berbagai cara. Pengolahan citra dilakukan dengan beberapa tujuan yaitu.

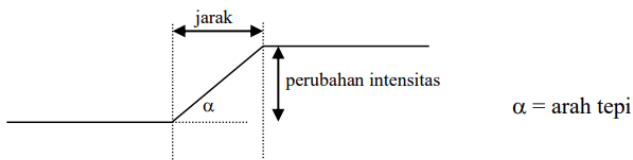
- Memperbaiki kualitas gambar seperti peningkatan kontras, restorasi citra, transformasi warna, rotasi, dan sebagainya.
- Melakukan proses pengambilan informasi, deskripsi objek, atau pengenalan objek yang terkandung pada citra
- Melakukan ekstraksi ciri pada citra (*feature extraction*) yang optimal untuk melakukan analisis
- Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data

B. Citra Digital

Citra digital adalah representasi citra kontinu melalui pencuplikan (*sampling*) secara ruang dan waktu. Citra digital tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat (x,y) dan amplitudo f(x,y). Koordinat (x,y) menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitudo f(x,y) menunjukkan nilai intensitas warna citra. The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

C. Pendeteksian Tepi Sobel

Tepi (edge) adalah perubahan nilai intensitas nilai keabuan yang mendadak (besar) dalam jarak yang singkat. Tepi memiliki arah, dan arah ini berbeda-beda pada bergantung pada perubahan intensitas.



Gambar 2.1 Tepi pada citra

Dalam pendeteksian tepi, ada beberapa operator yang dapat digunakan untuk mencari tepi dan salah satunya adalah operator Sobel. Operator sobel adalah magnitudo dari gradien yang dihitung dengan rumus

$$M = \sqrt{S_x^2 + S_y^2}$$

Yang dalam hal ini, turunan parsial dihitung dengan

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$S_y = (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4)$$

Sx dan Sy bisa juga dipandang sebagai *mask* (penapis) seperti pada gambar di bawah ini

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

III. IMPLEMENTASI SOLUSI

Secara umum, untuk bisa menduplikat atau membuat salinan dari sebuah objek yang ada pada sebuah citra, maka perlu dilakukan pengenalan objek terlebih dahulu. Setelah dilakukan pengenalan objek, barulah objek diduplikat dengan cara mengambil atau memotong bagian tertentu dari citra yang mengandung objek yang telah dikenali. Objek yang telah dikenali, kemudian dapat diolah sedemikian rupa untuk dapat dipakai dalam kebutuhan *editing*, seperti diletakkan pada citra yang berbeda untuk memberikan kesan lingkungan yang baru, atau bisa menduplikasi dengan arah yang sudah diubah untuk memberikan kesan sekumpulan objek pada sebuah citra.

Untuk bisa mengenail sebuah objek ada beberapa langkah yang perlu dilakukan. Namun secara umum, langkah yang dilakukan adalah memperjelas bagian objek dan latar belakang.

Langkah pertama adalah untuk mengubah citra ke dalam bentuk grayscale. Hal ini bertujuan untuk memudahkan proses pengenalan objek. Untuk mengubah citra ke bentuk grayscale kita bisa melakukannya dengan memanfaatkan fungsi bawaan yang telah disediakan oleh matlab seperti gambar di bawah ini.

```
imGray = im2gray(img);
```



Gambar 3.1 Citra asli



Gambar 3.2 Citra dalam bentuk grayscale

Setelah citra dikonversi ke bentuk grayscale dilakukan sedikit penajaman citra. Hal ini ditujukan untuk bisa lebih memudahkan bagian objek dan latar belakang. Hal ini bisa dilakukan dengan fungsi bawaan matlab seperti gambar di bawah ini.

```
imGray = imsharpen(imGray);
```



Gambar 3.3 Citra grayscale yang telah ditajamkan

Setelah citra ditajamkan, barulah dilakukan pendeteksian tepi pada citra. Hal ini dilakukan dengan tujuan menemukan tepi objek yang ada pada citra. Untuk pendeteksian tepi ini, digunakan fungsi pendeteksian tepi dengan metode sobel. Fungsi sobel diimplementasikan seperti pada gambar di bawah ini.

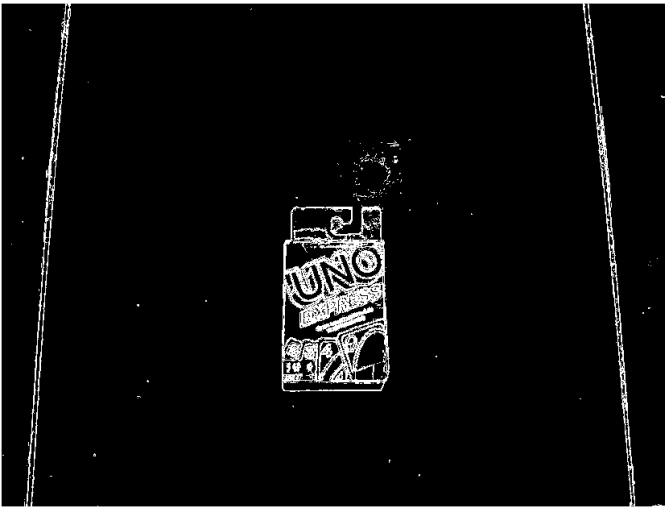
```
function edge_image = Sobel(img)
    Sx = [-1 0 1; -2 0 2; -1 0 1];
    Sy = [1 2 1; 0 0 0; -1 -2 -1];
    Jx = conv2(double(img), double(Sx), 'same');
    Jy = conv2(double(img), double(Sy), 'same');
    Jedge = sqrt(Jx.^2 + Jy.^2);
    edge_image = uint8(Jedge);
end
```



Gambar 3.4 Citra hasil pendeteksian tepi

Setelah melakukan pendeteksian tepi, citra hasil pendeteksian tepi kemudian diambangkan dengan nilai ambang T. Hal ini bertujuan untuk menghilangkan bagian-bagian yang dirasa bukan tepi dengan cara memberikan nilai 0 pada pixel yang nilainya dibawah ambang nilai T dan 255 untuk pixel yang nilainya di atas ambang nilai T. Pengambangan ini dapat diimplementasikan seperti gambar di bawah ini (T=60).

```
[p,1] = size(imEdge);
for i=1:p
    for j=1:1
        if(imEdge(i,j)<60)
            imEdge(i,j)=0;
        else
            imEdge(i,j)=255;
        end
    end
end
imBin = imbinarize(imEdge);
```



Gambar 3.5 Citra hasil pengambangan

Selanjutnya dilakukan penyambungan garis yang terputus. Hal ini dilakukan untuk menyambungkan garis tepi objek yang mungkin terputus sehingga dapat membuat objek utuh. Bagian dapat diimplementasikan dengan fungsi bawaan matlab seperti pada gambar di bawah ini.

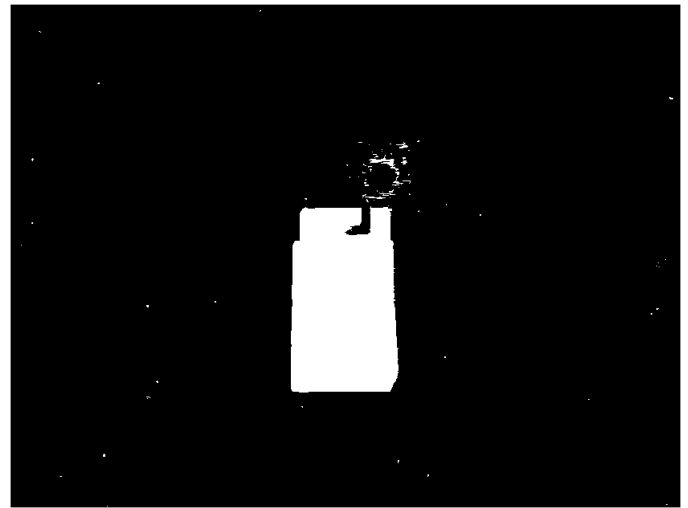
```
se = strel('line',10,0);
imClosed = imclose(imBin,se);
```



Gambar 3.6 Citra tepi yang telah ditutup

Setelah ditutup, citra hasil kemudian diisi untuk setiap bagian yang ada di dalam sebuah objek. Hal ini dilakukan dengan mengisi bagian kosong (hitam) yang sekitarnya ditutupi oleh bagian yang berisi (putih). Sebelum dilakukan, perlu juga diterapkan penghapusan border karena setelah percobaan ditemukan bahwa pada beberapa image, masih ada border yang ada di pinggir image. Bagian ini dapat diimplementasikan dengan menggunakan fungsi bawaan dari matlab seperti pada gambar di bawah ini.

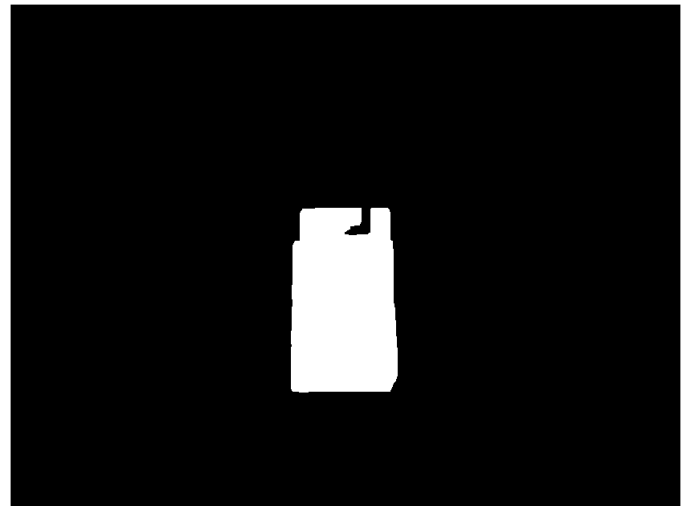
```
imFilled = imfill(imclearborder(imClosed),'holes');
```



Gambar 3.7 Citra yang telah diisi

Setelah itu, Langkah selanjutnya adalah menghilangkan bagian-bagian kecil yang bukan merupakan bagian dari object. Untuk melakukan ini, kita bisa menggunakan fungsi bawaan yang sudah ada seperti pada gambar di bawah ini.

```
se = strel('square',20);
opnImg = imopen(imFilled, se);
```



Gambar 3.8 Citra hasil fungsi 'imopen'

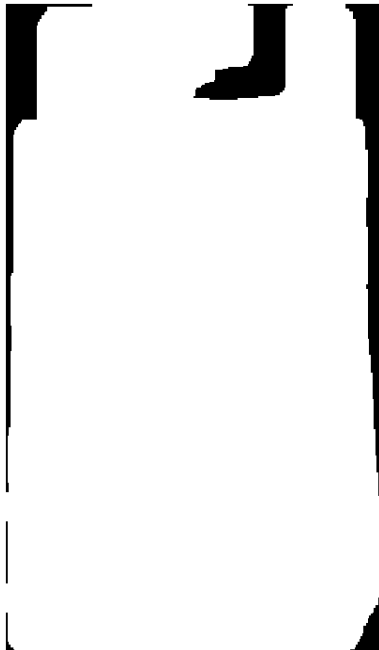
Setelah dilakukan pembersihan, maka bagian terakhir adalah mengekstrak objek dari citra. Cara ini dilakukan dengan mencari bagian citra yang mengandung sebuah objek (bagian putih) yang tentunya mencakup sebuah wilayah (tidak hanya sekedar titik). Untuk mengimplementasikan langkah ini, bisa digunakan fungsi bawaan yang telah disediakan oleh matlab seperti pada gambar di bawah ini.

```

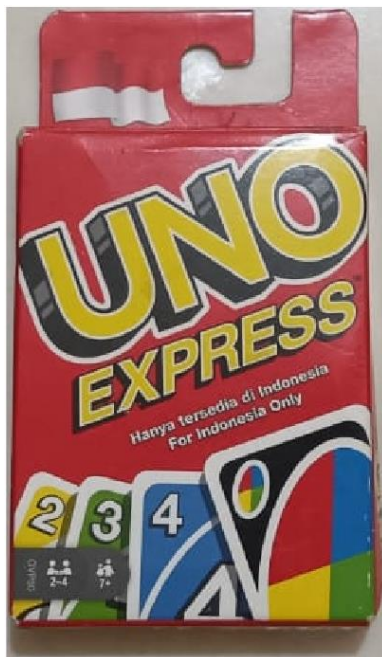
regions = regionprops(opnImg, "Area", "BoundingBox", "Image");
area = regions.Area;
count = numel(regions);
maxa = area;
boundingBox = regions.BoundingBox;
objGray = regions.Image;
for i = 1:count
    if (maxa < regions(i).Area)
        maxa = regions(i).Area;
        boundingBox = regions(i).BoundingBox;
        objGray = regions.Image;
    end
end
objRgb = imcrop(img,boundingBox);

```

Gambar 3.10 Citra berwarna dari objek



Gambar 3.9 Citra grayscale dari objek



IV. HASIL

Setelah mengenali objek, objek yang dikenali kemudian diolah untuk menciptakan permainan 'Find the Real Object'. Oleh karena itu, kita harus menduplikas objek yang ada sehingga citra hasil akan memiliki lebih dari 1 objek yang mana nantinya harus dipilih oleh pengguna. Untuk melakukan hal ini, kita hanya perlu memperbanyak jumlah objek dalam suatu citra. Selain itu, posisi objek juga bisa diatur atau dipilih random sedemikian rupa sehingga dapat membuat citra hasilnya tampak lebih natural. Selain itu, kita juga bisa mengubah kemiringan objek dengan cara memutar objek sebelum menempatkannya untuk mendapat kesan natural dan untuk meningkatkan tingkat kesulitan dari permainan ini.

Citra Asli	Citra Hasil
	 Menambahkan 2 objek
	 Citra dengan menerapkan kemiringan
	 Menambahkan 2 objek

kasih kepada bapak Rinaldi Munir sebagai dosen yang telah membimbing penulis. Tidak lupa juga penulis mengucapkan terima kasih kepada orang tua dan teman-teman yang terus menyemangati penulis selama pengerjaan makalah ini

REFERENCES

[1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/citra22-23.htm> Diakses pada 15 Desember 2022

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 16 Desember 2022



Louis Riemenn
13519016



Menerapkan kemiringan pada objek yang ditambahkan

V. KESIMPULAN

Ilmu pengolahan citra dapat diterapkan ke dalam banyak hal dalam kehidupan sehari-hari. Salah satu contoh penerapannya adalah menggunakan *image processing* dan pedeteksian tepi untuk menciptakan gambar baru untuk dapat digunakan dalam permainan yang berfokus pada mencari benda yang asli.

Pada penerapan ini, ditemukan kelemahan yaitu berupa bayangan dari objek yang masih tertangkap. Bayangan objek ini menyebabkan gambar yang dihasilkan kurang natural dan mempermudah tingkat kesulitan permainan.

VI. PENUTUP

Pada makalah yang sederhana ini, penulis ingin menunjukkan ide yang sederhana untuk menerapkan materi pengolahan citra yang telah dipelajari. Penulis juga ingin mengucapkan terima