

Pendeteksian Rintangan pada Permainan Chrome Dino

Aria Bachrul Ulum Berlian 13519115 (*Author*)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519115@std.stei.itb.ac.id

Abstract—Permainan Chrome Dino merupakan permainan dari peramban google chrome. Permainan ini mengharuskan pemain memandu dino untuk melewati rintangan agar mendapatkan skor setinggi-tingginya. Salah satu cara untuk mendapatkan poin setinggi-tingginya adalah dengan program bot. Salah satu algoritma bot tersebut adalah dengan terlebih dahulu mendeteksi rintangan yang ada. Program pendeteksian rintangan pada permainan dino dapat dibuat dengan metode pengolahan citra. Program pendetksi ini dapat mendeteksi semua rintangan yang berada di kanan dino hingga ujung kanan permainan, di bawah skor hingga diatas tanah.

Keywords—*citra, dino, bot, deteksi, rintangan*

I. PENDAHULUAN

Permainan chrome dino merupakan permainan yang dimiliki oleh *browser* google chrome yang biasa dimainkan saat tidak memiliki koneksi internet. Permainan ini dikembangkan pada tahun 2014 oleh anggota Chrome UX [1]. Pada saat ini, permainan dino dapat diakses dengan mengetikkan 'chrome://dino/' pada alamat URL meskipun tetap terhubung dengan koneksi internet. Terdapat pula beberapa web yang menyediakan replica permainan ini agar dapat dinikmati dengan menggunakan *browser* lain ataupun platform lain selain *desktop*.



Fig. 1. Permainan Chrome Dino

Pada permainan dino, kita akan memandu Tyranosaurus Rex. Permainan ini mengharuskan kita untuk memandu dino melewati rintangan yang ada tanpa menyentuhnya. Kita dapat melompat maupun menunduk. Seiring jauhnya dino berjalan, kita akan mendapatkan poin. Inti dari permainan ini adalah

mendapatkan poin sebanyak-banyaknya dengan berlari sejauh-jauhnya. Jika dino menyentuh rintangan, maka permainan berakhir. Chrome dino saat ini dapat berganti ke mode malam dan mode siang seiring poin bertambah. Pada mode malam, dino dan rintangan menjadi berwarna putih, sedangkan latarnya menjadi berwarna hitam. Pada mode siang, dapat kita lihat pada fig 1, latarnya berwarna putih, sedangkan dino dan rintangannya berwarna hitam.

Meskipun ini merupakan permainan luring, para pemain tetap menginginkan poin sebanyak-banyaknya. Para pemain melakukan segala cara untuk mendapatkan poin tertinggi dari permainan ini. Salah satu caranya adalah dengan membuat bot. Bot adalah sebuah robot atau program untuk melakukan tugas tertentu. Dalam permainan dino, botlah yang akan memandu dino agar tidak menyentuh rintangan yang ada. Terdapat beberapa jenis bot yang dibuat untuk memenangkan permainan ini. Diantaranya adalah bot dengan menggunakan sensor lalu diproses dengan sebuah microcontroller untuk menggerakkan sebuah actuator berupa motor servo. Motor servo ini akan menekan tombol spasi yang berfungsi untuk membuat dino melompat. Dengan cara ini, dino dapat melewati rintangan. Jenis bot yang kedua adalah dengan mengenali citra dari rintangan tersebut. Bot jenis ini dapat mengenali posisi rintangan agar dapat melompat pada waktu yang tepat. Untuk itu, diperlukan suatu cara untuk mengenali rintangan tersebut.

Dino dan rintangan pada mode siang berwarna hitam, sedangkan latarnya berwarna putih. Pada mode malam berlaku sebaliknya. Dapat kita lihat bahwa citra dari permainan ini dapat dibuat menjadi citra biner, yaitu citra hitam dan putih. Pada makalah ini akan dibahas cara mengenali rintangan pada citra permainan dino menggunakan prinsip citra biner dan pendeteksian tepi.

II. LANDASAN TEORI

A. Citra

Citra merupakan sinyal dwimatra yang dapat diamati oleh system visual manusia maupun oleh system digital yang bersifat terus-menerus (continue). Citra dapat pula diartikan sebagai gambar baik dalam bentuk fisik pada bidang dua dimensi maupun dalam bentuk digital. Citra yang ditangkap system digital didiskritisasi dengan cara pencuplikan dan direpresentasikan dalam bentuk matematis menjadi fungsi yang

menyatakan intensitas cahaya pada suatu titik dalam bidang dua dimensi sebagai berikut.

$$f(x,y)$$

x, y : titik koordinat pada bidang dua dimensi

$f(x, y)$: intensitas cahaya (brigtness) pada titik (x,y)

Pada citra berwarna, citra tersebut direpresentasikan dalam tiga kanal. Kanal tersebut adalah kanal R(red), kanal G(green), dan kanal B(blue). Masing-masing kanal memiliki fungsi intensitasnya masing-masing. Kombinasi dari tiga warna tersebut dapat merepresentasikan warna apapun.

Terdapat dua jenis citra, yaitu citra diam dan citra bergerak. Citra diam adalah sebuah citra tunggal. Citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan seperti gambar yang bergerak. Setiap citra diam dalam citra bergerak disebut frame. Citra permainan video seperti chrome dino adalah contoh citra bergerak.

B. Citra Grayscale

Citra grayscale merupakan citra yang hanya memiliki satu kanal pada setiap pixelnya. Citra ini hanya menggunakan satu tingkat keabuan. Pada citra jenis ini, ketiga kanal RGB memiliki intensitas yang sama pada masing-masing pixelnya. Oleh karena itu, citra jenis ini lebih mudah untuk diproses karena hanya perlu memproses satu kanal.

Untuk membuat sebuah citra grayscale dari citra berwarna, dapat dilakukan dengan cara berikut. Cara yang pertama dan paling sederhana adalah dengan mencari rata-rata dari nilai intensitas masing-masing kanal RGB.

$$Y = (R + G + B) / 3$$

Namun, sayangnya hasil dari persamaan ini kurang tepat. Citra grayscale yang dihasilkan kurang bagus.

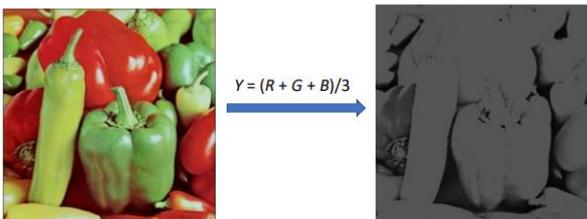


Fig. 2. Konversi grayscale dengan merata-ratakan

Oleh karena itu, citra berwarna dikonversi menjadi citra grayscale dengan rasio atau perbandingan tertentu pada masing-masing kanal. Rasio yang cukup bagus adalah sebagai berikut.

$$Y = 0.299R + 0.587G + 0.144B$$

Berikut merupakan citra yang dihasilkan dengan menggunakan rasio tersebut.

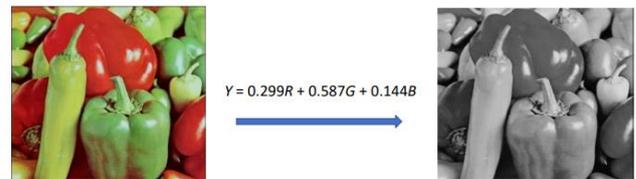


Fig. 3. Konversi grayscale dengan rasio 0.299R, 0.587G, 0.144B

Nilai intensitas setiap pixel pada citra grayscale berada diantara 0 hingga 255. Nilai 0 merepresentasikan warna hitam, sedangkan nilai 255 merepresentasikan warna putih. Namun, nilai ini tidak selalu 0 hingga 255. Nilai ini bergantung pada kedalaman pixel citra grayscale. Misalnya citra grayscale yang memiliki kedalaman 1 bit hanya memiliki skala 0 – 1. Citra kedalaman 2 bit memiliki skala 0 – 3. Citra kedalaman 4 bit memiliki skala 0 – 15 dan seterusnya. Citra dengan kedalaman 1 bit disebut juga sebagai citra biner.

C. Citra Biner

Citra Biner merupakan citra grayscale yang hanya memiliki dua nilai, yaitu 0 dan 1. Nilai 0 merepresentasikan warna hitam, sedangkan 1 merepresentasikan warna putih. Mungkin sebaliknya tergantung dari konvensi yang digunakan.



Fig. 4. Citra Biner

Citra jenis ini memiliki beberapa keunggulan. Keunggulan tersebut diantaranya adalah sebagai berikut.

1. Kebutuhan memori untuk setiap pixel lebih sedikit karena hanya membutuhkan 1bit untuk setiap pixel
2. Dapat menggunakan operasi logika (AND, OR, NOT) sehingga waktu komputasinya lebih kecil
3. Dapat digunakan untuk merepresentasikan citra hasil pendeteksian tepi (*edge images*)
4. Dapat memisahkan (segmentasi) objek dari gambar latar belakangnya
5. Dapat digunakan untuk lebih memfokuskan pada analisis bentuk morfologi karena nilai pixel tidak terlalu penting dibanding bentuknya.
6. Digunakan untuk menampilkan citra pada perangkat luaran yang hanya memiliki resolusi satu bit, misalnya printer hitam putih.

Konversi dari citra grayscale menjadi citra hitam putih dilakukan dengan operasi pengambangan (thresholding). Operasi ini akan mengelompokkan nilai derajat keabuan setiap pixel ke dalam dua kelas, yaitu hitam dan putih. Jika nilai derajat keabuan lebih besar dari pada threshold, nilai tersebut akan masuk kelompok warna putih. Selain itu, akan masuk ke dalam kelompok warna hitam.

Terdapat 3 macam operasi pengambangan. Operasi tersebut adalah pengambangan secara global, pengambangan secara local dan pengambangan secara adaptif. Pengambangan secara global hanya menggunakan nilai T yang digunakan untuk seluruh bagian didalam citra. Nilai T yang dipilih harus membuat galat sekecil mungkin. Pengambangan secara local dilakukan terhadap bagian-bagian tertentu saja dalam citra. Untuk itu, citra dipecah menjadi bagian-bagian kecil, kemudian proses pengambangan dilakukan secara local. Pada setiap bagian, dapat memiliki nilai T yang berbeda satu sama lain. Pada pengambangan adaptif, nilai ambang akan berubah secara dinamis pada citra. Pengambangan ini dapat mengakomodasi perubahan kondisi cahaya pada gambar, misalnya perubahan akbat gradien iluminasi yang kuat atau terdapat bayangan.

D. Pengolahan Citra

Pengolahan citra adalah proses yang bertujuan untuk memperbaiki kualitas citra agar mudah untuk diinterpretasi oleh manusia maupun oleh mesin (komputer). Teknik pengolahan citra mentransformasikan citra masukan menjadi citra lain sebagai keluarannya.

Operasi-operasi di dalam pengolahan citra digital adalah sebagai berikut.

1. Perbaikan kualitas citra (*image enhancement*).

Image enhancements merupakan teknik pengolahan citra untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Contoh operasi perbaikan citra adalah perbaikan kontras gelap/terang, perbaikan tepian objek (*edge enhancement*), penajaman citra (*sharpening*), *noise filtering*, *blurring*, serta koreksi geometric.

2. Pemampatan citra (*image compression*).

Pemampatan atau kompresi citra bertujuan untuk menghilangkan redundansi pada citra. Hasil dari citra kompresi adalah citra dengan ukuran memori yang lebih ringan. Terdapat dua jenis kompresi pada citra digital, yaitu lossless dan lossy. Pada kompresi lossless, data pixel dapat direkonstruksi menjadi data pixel yang sama persis dengan data sebelum dikompresi. Contoh kompresi jenis ini adalah GIF dan PNG. Pada kompresi lossy, data pixel tidak sama persis setelah proses kompresi. Terdapat informasi yang hilang. Contoh kompresi jenis ini adalah JPEG.

3. Pengorakan citra (*image analysis*).

Image analysis bertujuan untuk menghitung ukuran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik ini mengekstraksi ciri-ciri tertentu yang

membantu dalam identifikasi objek. Contoh pengorakan citra adalah Pendeteksian Tepi objek (*edge detection*), ekstraksi batas (*boundary*) dan representasi daerah (*region*).

4. Rekonstruksi citra (*image reconstruction*).
5. Restorasi Citra (*image restoration*).

Restorasi citra bertujuan untuk menghilangkan atau meminimumkan cacat pada citra. Tujuannya hampir sama dengan image enhancement. Namun, pada image restoration, penyebab degradasi gambar diketahui.

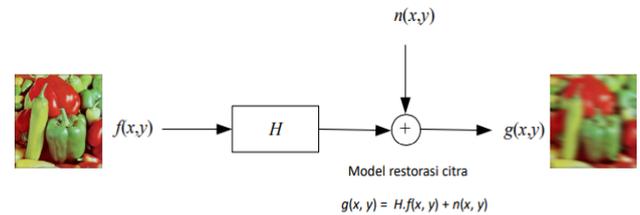


Fig. 5. Model restorasi Citra

E. Edge Detection

Pendeteksian tepi adalah Teknik untuk mengenali tepi dari sebuah gambar. Tepi adalah perubahan nilai keabuan yang mendadak dalam jarak yang singkat. Tepi juga memiliki arah tergantung pada perubahan intensitas. Terdapat empat macam tepi. Macam tepi tersebut adalah tepi curam, tepi landai, tepi garis, dan tepi atap.

Pendeteksian tepi bertujuan untuk meningkatkan penampakan garis batas suatu objek didalam citra. Teknik ini akan mengekstraksi representasi gambar garis-garis di dalam citra. Teknik pendeteksian ini juga berguna untuk mengenali objek di dalam citra. Teknik pendeteksian tepi adalah bagian dari analisis citra. Tujuannya adalah mengidentifikasi parameter-parameter yang diasosiasikan dengan ciri (feature) dari objek di dalam citra agar selanjutnya dapat digunakan untuk menginterpretasikan sebuah citra.

Pendeteksian tepi dapat dilakukan dengan beberapa operator. Operator tersebut diantaranya adalah operator gradien, operator laplace, operator laplace of gaussian, operator sobel, operator canny, operator prewitt, dan operator Roberts.

Operator gradien adalah operator yang menggunakan pendekatan kalkulus diferensial. Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan besar. Sehingga fungsi tersebut adalah sebagai berikut.

$$\frac{\partial F}{\partial x} = \lim_{h \rightarrow 0} \frac{F(x+h, y) - F(x, y)}{h}$$

Turunan parsial dari fungsi dwimatra pada citra dapat dipandang sebagai dua buah mask konvolusi, yaitu

$$G_x = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

Keputusan apakah suatu pixel merupakan tepi dilakukan dengan operasi pengambangan pada nilai kekuatan tepi pada citra. Jika nilai kekuatan tepi lebih besar dari nilai ambang, maka pixel tersebut merupakan sebuah tepi dan berlaku sebaliknya. Nilai kekuatan tepi didapat dengan rumus

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2}$$

Operator Laplace atau operator turunan kedua mendeteksi lokasi tepi lebih akurat, khususnya pada tepi yang curam. Turunan kedua pada tepi curam memiliki persilangan nol (*zero crossing*), yaitu titik yang terdapat pergantian tanda nilai turunan kedua dari positif ke negatif atau sebaliknya. Contoh mask konvolusi operator laplace adalah sebagai berikut.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Fig. 6. Mask konvolusi operator laplace

Operator laplace of gaussian adalah operator pendeteksi tepi dengan mengurangi kemunculan tepi palsu. Citra ditapis dulu dengan fungsi gaussian sebelum dilakukan operator laplace. Atau, dilakukan penurunan kedua pada fungsi gauss hingga dihasilkan fungsi topi Mexico (*Mexican Hat*). Contoh penapis LoG adalah sebagai berikut.

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Fig. 7. Mask konvolusi operator LoG

Operator Sobel menghitung magnitudo dari gradien turunan parsial yang dihitung dengan rumus sebagai berikut.

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

Operator sobel dalam bentuk mask, s_x dan s_y dinyatakan sebagai berikut.

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

Fig. 8. Mask konvolusi operator sobel

Magnitudo digunakan untuk menentukan apakah suatu pixel merupakan tepi atau bukan dengan membandingkannya dengan nilai ambang.

Operator Prewitt menggunakan persamaan gradien seperti operator Sobel. Namun, mask konvolusi yang digunakan adalah sebagai berikut.

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Fig. 9. Mask konvolusi operator Prewitt

Operator Roberts adalah operator silang. Operator Roberts memiliki mask konvolusi sebagai berikut.

$$R_+ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{dan} \quad R_- = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Fig. 10. Mask konvolusi operator Prewitt

Kekuatan tepi operator Roberts dihitung dengan menjumlahkan nilai absolut dari R_+ dan R_- .

Operator Canny adalah operator yang memiliki hasil tepi dengan ketebalan satu pixel. Operator canny menghaluskan citra dengan penapis gaussian. Setelah itu, gradien dan arah gradien setiap pixel dihitung dengan operator pendeteksi tepi lain. Jika nilai magnitude gradien suatu pixel melebihi nilai ambang, maka pixel tersebut termasuk pixel tepi.

III. IMPLEMENTASI

Pada bagian ini dijelaskan proses pengembangan pendeteksian rintangan pada permainan dino. Pendeteksian ini menggunakan metode pengolahan citra, tepatnya dengan pengolahan citra biner dan pendeteksian tepi. Namun, pendeteksian rintangan terbatas pada rintangan di depan karakter dino. Langkah-langkah untuk mendapatkan posisi rintangan dari permainan dino adalah sebagai berikut.

1. Muat Gambar

Pertama, citra dari permainan dino perlu ditangkap. Caranya adalah dengan melakukan penangkapan layer dengan tombol *print screen* atau dengan menggunakan *snipping tools*. Citra kemudian disimpan.

Pada program, muat gambar dengan menggunakan potongan program dibawah ini.

```
clear;clc;

[filename, pathname] = uigetfile('*.');
drawnow
if ~isequal(filename, 0)
```

```

    image = imread(fullfile(pathname,
filename));
    imshow(image);
end

```

Potongan program ini juga sekaligus menampilkan citra masukan ke layer.

2. Ubah citra menjadi citra grayscale

Citra hasil tangkapan layer berupa citra berwarna dengan 3 kanal, yaitu Red, Green, dan Blue. Namun, pada pemrosesan citra ini, kita hanya perlu memproses satu kanal. Sehingga, citra masukan diubah menjadi grayscale dengan satu kanal dengan menggunakan potongan program dibawah ini.

```

sizeImage = size(image);

%convert image to 1 channel
if(sizeImage(3)) > 1
    imageResult = rgb2gray(image);
end

```

3. Ubah citra menjadi citra biner

Citra pada dino hanya memiliki 3 buah nilai intensitas keabuan. Nilai keabuan tersebut digunakan untuk citra dino, skor, dan rintangan, untuk citra latar belakang serta untuk citra awan-awan pada latar. Kita tidak memerlukan awan-awan latar. Sehingga, kita ubah citra menjadi citra biner agar menyisakan dino, skor, rintangan pada satu nilai keabuan dan latar pada satu nilai keabuan lagi. Hal ini dilakukan dengan melakukan operasi pengambangan dengan nilai ambang 85.

```

% convert image to binary to remove
background
imageResult = imageResult > 85;
imshow(imageResult);

```

4. Hapus citra yang tidak menjadi region of interest

Dengan metode heuristic, kita dapat mengetahui bahwa posisi dino terhadap horizontal tetap. Posisi tanah terhadap vertikal pun tetap. Posisi skor terhadap horizontal dan vertikal tetap. Daerah yang memungkinkan terdapat rintangan pun tetap, yaitu dibawah skor, diatas tanah, dan selain pada dino. Pada makalah ini, kita hanya membahas rintangan yang berada didepan dino. Akhirnya, kita dapat merumuskan wilayah yang menjadi perhatian, yaitu dibawah skor hingga atas tanah, di kanan dino, hingga ujung kanan citra. Program untuk memangkas citra menjadi RoI adalah sebagai berikut.

```

% crop non obstacle area
x = round(15/100 * sizeImage(2));
y = round(35/100 * sizeImage(1));
w = sizeImage(2)-x;
h = (sizeImage(1)-y) - round(26/100 *
sizeImage(1));

```

```

imageResult = imcrop(imageResult, [x y w
h]);
imshow(imageResult);

```

5. Deteksi tepi rintangan pada citra

Setelah mengetahui *region of interest*, tepatnya wilayah terdapatnya rintangan, kita lakukan deteksi tepi. Oleh karena di wilayah ini hanya ada rintangan, tepi yang terdeteksi pun dapat dipastikan adalah sebuah rintangan. Pendeteksian tepi dilakukan dengan operator canny sehingga dihasilkan tepi berlebar satu pixel. Tepi ini cukup untuk menandai lokasi terdapatnya rintangan. Nilai ambang yang digunakan adalah 20/255.

```

% edge detect
threshold = 20/255;
imedge = edge(imageResult, 'canny',
threshold);
imshow(imedge, 'ColorMap', [1 1 1; 1 0 0]);

```

IV. HASIL IMPLEMENTASI

Berikut hasil implementasi dari program pendeteksi rintangan dino.



Fig. 11. Citra masukan 1

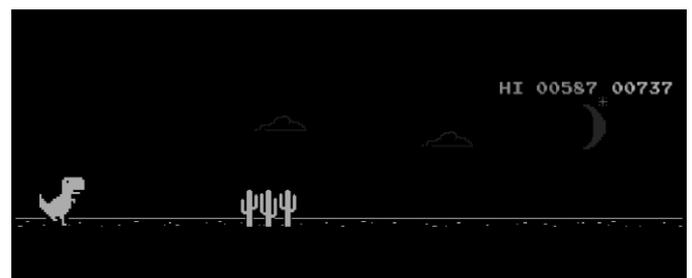


Fig. 12. Citra masukan 2

Berikut hasil keluaran setelah pemrosesan menjadi citra biner.



Fig. 13. Citra keluaran biner 1

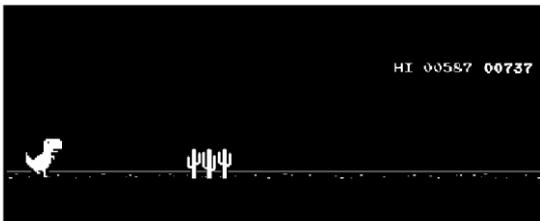


Fig. 14. Citra keluaran biner 2

Berikut hasil keluaran setelah penghapusan bagian yang bukan daerah rintangan.



Fig. 15. Citra keluaran RoI 1



Fig. 16. Citra keluaran RoI 2

Berikut hasil pendeteksian tepinya.



Fig. 17. Citra keluaran rintangan 1



Fig. 18. Citra keluaran rintangan 2

V. KESIMPULAN DAN SARAN

Pengolahan citra merupakan teknik yang dapat dimanfaatkan secara luas. Salah satu contoh pemanfaatannya adalah untuk mendeteksi rintangan yang terdapat pada permainan dino chrome. Program ini berjalan dengan baik meskipun terdapat beberapa Batasan dalam implementasinya. Program ini selanjutnya dapat digunakan untuk membangun sebuah program bot untuk mendapatkan skor setinggi-tingginya. Setelah mengetahui posisi rintangan, bot hanya perlu memerintahkan dino untuk melompat atau menunduk pada waktu yang tepat.

ACKNOWLEDGEMENT

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas rahmatnya, penulis dapat menyelesaikan makalah ini yang berjudul "Pendeteksian Rintangan pada Permainan Dino". Penulis juga mengucapkan terima kasih kepada orang tua yang senantiasa memberi dukungan. Penulis juga mengucapkan terima kasih kepada Bapak Dr.Ir.Rinaldi Munir, selaku dosen pengampu mata kuliah IF4073 Interpretasi dan Pengolahan Citra. Atas pembelajaran dari beliau penulis dapat menyusun makalah ini

REFERENCES

- [1] <https://dino-chrome.com/> (diakses pada pukul 10.12, 19 Desember 2022)
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/01-Pengantar-Pengolahan-Citra-Bag1-2022.pdf>
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/02-Pengantar-Pengolahan-Citra-Bag2-2022.pdf>
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/18-Pendeteksian-Tepi-Bagian1-2022.pdf>
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/19-Pendeteksian-Tepi-Bagian2-2022.pdf>
- [6] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/20-Citra-Biner-2021.pdf>

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, 19 Desember 2022

Aria Bachrul Ulum Berlian
13519115

