BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan kecerdasan buatan yang pesat, membuat kecerdasan buatan menjadi opsi yang menarik untuk diterapkan pada sebuah game. Hal tersebut dibuktikan dari banyaknya algoritma kecerdasan buatan yang digunakan dalam berbagai game, seperti game kartu (Ricardo & Fernandes, 2016), game ular (Sebastianelli dkk., 2021a), dan bahkan game online Valorant yang sedang popular saat ini (Fernanda dkk., 2022). Kecerdasan buatan dapat meningkatkan *gameplay*, seperti pengambilan keputusan yang lebih baik, dan mengoptimalkan strategi pada beberapa game. Salah satu program yang sangat terkenal, yaitu AlphaGo yang berhasil mencapai tingkat kemenangan 99,8% melawan program Go lainnya dan mengalahkan juara Go Eropa manusia pada 5 pertandingan (Zhao dkk., 2018).

Permainan ular menjadi salah satu game yang banyak digunakan sebagai objek dari penerapan kecerdasan buatan (Hau Hor dkk., 2022). Banyak algoritma yang digunakan, seperti algoritma genetika (Białas, 2019), Proximal Policy Optimization (Zhang & Cai, 2020), Deep-Q-Network (Wei dkk., 2018), dan Best First Search (Kong & Mayans, 2021). Dari banyaknya algoritma yang telah diterapkan, algoritma genetika menjadi salah satu algoritma optimasi yang paling cocok dalam mencari solusi terbaik (Hau Hor dkk., 2022). Algoritma genetika berasal dari sebuah bidang studi yang dikenal sebagai komputasi evolusioner, digunakan untuk menyalin proses reproduksi dan memilih solusi yang paling cocok. Fungsi ini

memungkinkan algoritma genetika untuk menemukan solusi atas masalah yang tidak dapat diambil oleh metode lain karena kurangnya fitur (Carr, 2014).

Neural networks merupakan algoritma yang akan dikombinasikan dengan algoritma genetika. *Neural networks* memiliki sistem yang terdiri dari koneksi antara unit-unit pemrosesan informasi yang disebut neuron, yang meniru cara kerja otak manusia dalam membawa dan memproses sinyal. *Neural networks* mencoba meniru struktur dan fungsi neuron alami dengan menggunakan input, output, dan fungsi aktivasi. Perceptron adalah bentuk paling sederhana dari *neural networks*, terdiri dari satu neuron dengan dua input dan satu output (Shiruru, 2016). Dengan demikian, *neural networks* merupakan tambahan yang berguna untuk berbagai algoritma yang digunakan dalam pemecahan masalah. Berbeda dengan algoritma genetika, *neural networks* bukan tipikal algoritma yang dapat berdiri sendiri untuk menyelesaikan masalah yang cukup kompleks (Hau Hor dkk., 2022). *Neural networks* biasanya dikombinasikan dengan algoritma genetika menjadi *evolutionary neural networks*, seperti pada game ular (Białas, 2019), Game 2048 (Boris & Goran, 2017), Flappy bird (Mishra dkk., 2019), dan Slither.io (Miller dkk., 2019).

Pengujian parameter merupakan tahapan eksperimen yang dilakukan secara berulang dengan sistem *trial and error* untuk menemukan konfigurasi terbaik dari sebuah algoritma (Y. Ma, 2024). Konfigurasi parameter yang tepat dapat meningkatkan performa dari algoritma genetika sebagai *evolution operator* dan *neural networks* sebagai *vision* atau arah gerak dari ular pada permainan ular klasik. Algoritma genetika dapat terjebak dalam solusi lokal jika parameternya tidak di-*set*

dengan benar. Pengujian parameter dapat membantu dalam mengidentifikasi nilai parameter optimal yang dapat mengarahkan pada solusi global (Uthansakul dkk., 2020). Selain itu, pemilihan parameter yang tepat dapat meningkatkan kecepatan dari konvergensi yang berpengaruh pada runtime dari programnya (Rahul Ramesh Patil, 2023).

Beberapa penelitian sudah dilakukan dengan menerapkan algoritma genetika dan neural network pada permainan ular. Penelitian pertama oleh (Chi Yuen dkk., 2021) melakukan empat eksperimen dan menemukan bahwa hasil terbaik didapat pada *mutation* rate 0.05, 20000 generasi, satu hidden layer, dan 16 neuron, namun penelitian ini tidak menyediakan GUI untuk permainan ular. Penelitian kedua oleh (Hau Hor dkk., 2022) juga berfokus pada pengujian parameter tanpa GUI, menemukan bahwa ular memiliki kinerja terbaik dengan nilai parameter menengah untuk panjang blok, persentase mutasi, dan intensitas mutasi, serta rasio kinerja terbaik/buruk 18:2. Penelitian ketiga oleh (Białas, 2019) menggunakan satu hidden layer dan 6 neuron, dengan hasil terbaik pada populasi 1000, namun juga tanpa GUI. Penelitian keempat oleh (B. HalmosiC. Sik-Lányi, 2019) menyertakan GUI dan menggunakan *feed-forward neural networks* dengan satu hidden layer, namun memiliki batasan generasi maksimal 500 yang membatasi objektivitas hasil.

Dari empat penelitian yang sudah dilakukan sebelumnya, beberapa masih memiliki kekurangan yang sama, yaitu tidak menampilkan GUI permainan ular. Kehadiran GUI tentu sangat penting, karena digunakan sebagai bahan evaluasi, seperti bagaimana ular menghindari bahaya yang ada dan bagaimana ular mendapatkan makanannya. Selain itu, pengujian parameter yang dilakukan dengan

membandingkan generasi kecil ke terbesar merupakan pengujian yang kurang optimal, karena generasi paling besar sudah pasti mendapatkan *score* yang lebih baik. Berdasarkan kekurangan yang masih ada pada penelitian sebelumnya, program ini akan dibangun dengan GUI dari permainan ular dan dilakukan dalam 4 pengujian yang terdiri dari 60 kali eksperimen. Pengujian ini dilakukan untuk mendapatkan performa yang maksimal dan efisiensi waktu komputasi yang didasarkan pada pemilihan parameter yang tepat dari algoritma genetika dan *neural networks*. Selain dari menghasilkan konfigurasi parameter yang baik, mengetahui interaksi antara parameter pada kedua algoritma juga merupakan salah satu alasan dilakukannya pengujian, seperti parameter apa yang mempengaruhi *score* yang didapat, dan parameter apa yang berpengaruh besar terhadap waktu komputasi program. Permainan ular hanya digunakan sebagai model ekperimen AI-nya. Permainan ular memiliki metrik kinerja yang jelas, seperti panjang ular atau skor permainan. Metrik ini memudahkan evaluasi efektivitas konfigurasi parameter yang diuji.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, dapat dirumuskan masalah pada penelitian ini, yaitu:

a. Bagaimana implementasi dari algoritma genetika dalam melakukan regenerasi matriks bobot yang optimal untuk meningkatkan kinerja dari neural networks yang menentukan arah gerak dari ular?
b. Bagaimana pemilihan parameter yang tepat pada algoritma genetika dan *neural networks*

1.3 Tujuan Penelitian

Dari uraian masalah yang sudah dirumuskan, didapat tujuan penelitian sebagai berikut:

a. Mengimplementasikan algoritma genetika dalam proses regenerasi matriks bobot yang optimal dengan tujuan untuk meningkatkan kinerja neural networks dalam menentukan arah gerak dari ular.
b. Mengukur penggunaan jumlah generasi, jumlah populasi, dan *mutation chance* pada algoritma genetika, dan jumlah neurons untuk menemukan konfigurasi parameter yang tepat, dengan input berupa matriks bobot pada *neural networks*.

1.4 Manfaat Penelitian

Adapun beberapa manfaat yang didapatkan dari penelitian ini, sebagai berikut:

a. Penelitian ini berkontribusi untuk menemukan konfigurasi parameter algoritma genetika dan *neural networks* yang tepat berdasarkan hasil eksperimen yang dilakukan.
b. Penelitian ini membantu meningkatkan pemahaman tentang interaksi parameter, seperti jumlah generasi, populasi, *mutation chance*, dan jumlah neuron yang mempengaruhi besar atau kecilnya *score* yang didapat dan lama atau tidaknya waktu komputasi atau *runtime* pada program.
c. Penelitian ini memiliki output berupa model dari kombinasi algoritma genetika dan neural *networks* yang di mana dapat diimplementasikan pada berbagai permainan yang memiliki *board game*, seperti catur, go, tic-tac toe, dan yang lainnya.

1.5 Batasan Masalah

Berikut merupakan batasan masalah yang terdapat pada penelitian ini:

a. Permainan ular yang dibuat hanya berupa program sederhana yang dibuat dengan bahasa python dan library pygame.
b. Permainan ular tidak memiliki fitur apapun, hanya terdapat ular, makanan dan tembok pada masing-masing sudut.
c. Jumlah generasi dan populasi yang digunakan pada tahapan eksperimen kurang dari 2000
d. *Mutation chance* hanya berkisar antara 0.1% sampai 0.5% pada tahapan eksperimen
e. Jumlah neuron yang diuji hanya berjumlah 16 neuron dan 24 neuron.