

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Permainan puzzle merupakan salah satu permainan yang memerlukan logika dan nalar dalam penyelesaiannya. Salah satunya adalah puzzle angka yang bernama Sudoku. Sudoku merupakan puzzle logika di mana pemain harus mengisi sebuah kotak dengan angka-angka antara 1 sampai 9. Adanya tantangan dari menyelesaikan masalah ini menjadikannya populer di kalangan masyarakat dunia selama beberapa dekade terakhir. Sudoku adalah singkatan Bahasa Jepang dari “*Suuji wa dokushin ni kagiru*”, artinya “angka-angkanya harus tetap tunggal” (Yusuf dan Hendra, 2013).

Puzzle Sudoku yang paling umum adalah kisi-kisi yang terdiri dari 81 buah sel yang dibentuk menjadi 9 kotak yang di dalamnya terdapat 3 baris dan 3 kolom untuk diisi bilangan bulat dari 1 sampai 9. Teka-teki dimulai dengan sejumlah bilangan bulat yang diberikan pada beberapa sel sebagai petunjuk untuk menyelesaikan permainan (Harrysson dan Laestander, 2014). Mattias (dalam Santos-Gracia, 2007) mengemukakan bahwa tujuan dari Puzzle Sudoku adalah untuk mengisi seluruh kotak  $9 \times 9$  sedemikian rupa sehingga batasan-batasan berikut terpenuhi:

- Setiap baris dalam sel hanya boleh berisi bilangan bulat dari satu sampai sembilan hanya tepat satu.
- Setiap kolom dalam sel hanya boleh berisi bilangan bulat dari satu sampai sembilan hanya tepat satu.
- Setiap sub-kisi  $3 \times 3$  juga hanya diperbolehkan berisi bilangan bulat dari satu sampai sembilan hanya tepat satu.

Tingkat kesulitan dari permainan Sudoku didasarkan pada banyaknya angka petunjuk yang muncul ketika permainan dimulai. Semakin sedikit angka petunjuk yang ditunjukkan, maka semakin sulit Permainan Sudoku ini untuk diselesaikan. Puzzle pada Gambar 1.1 adalah puzzle dengan tingkat kesulitan mudah karena dapat diselesaikan dengan logika dan tidak memerlukan tebakan sedangkan solusinya digambarkan pada Gambar 1.2.

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 |   |   |   |   |   |   |   | 2 |
|   |   | 8 |   |   | 9 |   | 3 | 7 |
| 7 |   |   | 5 | 3 |   |   |   | 8 |
|   | 8 |   |   | 7 | 3 |   | 5 | 4 |
|   |   | 6 | 4 |   | 2 | 7 |   |   |
| 9 | 7 |   | 8 | 5 |   |   |   | 1 |
|   | 1 |   |   | 8 | 7 |   |   | 9 |
| 3 | 4 |   | 6 |   |   | 8 |   |   |
| 8 |   |   |   |   |   |   |   | 1 |

Gambar 1.1 Puzzle Sudoku

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 3 | 7 | 6 | 8 | 9 | 4 | 2 |
| 4 | 6 | 8 | 1 | 2 | 9 | 5 | 3 | 7 |
| 7 | 2 | 9 | 5 | 3 | 4 | 1 | 8 | 6 |
| 2 | 8 | 1 | 9 | 7 | 3 | 6 | 5 | 4 |
| 5 | 3 | 6 | 4 | 1 | 2 | 7 | 9 | 8 |
| 9 | 7 | 4 | 8 | 5 | 6 | 2 | 1 | 3 |
| 6 | 1 | 5 | 3 | 8 | 7 | 4 | 2 | 9 |
| 3 | 4 | 2 | 6 | 9 | 1 | 8 | 7 | 5 |
| 8 | 9 | 7 | 2 | 4 | 5 | 3 | 6 | 1 |

Gambar 1.2 Solusi Puzzle Sudoku pada Gambar 1.1

Berdasarkan uraian di atas, disebutkan bahwa tingkat kesulitan akan meningkat ketika angka petunjuk yang muncul sedikit. Hal ini menyebabkan pemain tidak bisa sepenuhnya menggunakan logika dalam menyelesaikan puzzle karena akan banyak menebak beberapa kemungkinan angka untuk mengisi sel yang kosong hingga menemukan solusi yang benar. Pengisian secara manual dapat menghabiskan waktu lebih lama untuk menyelesaikan permainan, terlebih jika tebakan kurang tepat. Oleh karena itu, dilakukan penelitian untuk menyelesaikan Puzzle Sudoku yang memiliki tingkat kompleksitas tinggi dengan efektif.

Sari (2011) pada penelitiannya mengatakan bahwa permasalahan Puzzle Sudoku sulit untuk dipecahkan karena termasuk permasalahan *NP-complete*, yaitu masalah kombinatorial yang memerlukan waktu komputasi yang lama untuk diselesaikan dengan metode eksak. Cara paling mudah untuk menyelesaikan Permainan Sudoku adalah dengan menggunakan Algoritma *Brute Force*. Algoritma ini mengenumerasi semua kemungkinan isi sel dengan angka 1 sampai 9. Namun, cara ini memiliki kelemahan, yaitu akan memunculkan banyak kemungkinan. Oleh karena itu, digunakan Algoritma *Backtracking* yang merupakan perbaikan dari Algoritma *Brute Force*. Solusi hasil dari Algoritma *Backtracking* dapat ditemukan dengan langkah pencarian yang lebih sedikit dan mampu mencari solusi permasalahan secara lebih efektif karena pencariannya hanya mengarah ke solusi yang perlu dipertimbangkan saja.

Selain peneliti di atas, terdapat peneliti lain yang menggunakan algoritma serupa untuk dibandingkan dengan algoritma metaheuristik *Simulated Annealing* (SA) dan *Genetic Algorithms* (GA). Kamal dkk. (2015) mengusulkan deteksi

digital dan deskripsi Puzzle Sudoku menggunakan teknik berbasis penglihatan dan pemecahan puzzle selanjutnya dengan tiga Algoritma *Backtracking*, *Simulated Annealing*, dan *Genetic Algorithms*. Hasil dari analisis komparatif ketiga algoritma ini menunjukkan bahwa *Backtracking* memiliki kinerja terbaik di antara ketiga algoritma sedangkan *Genetic Algorithms* memiliki kinerja terburuk dan tidak dapat memecahkan Puzzle Sudoku standar dengan kompleksitas yang lebih tinggi dalam batas waktu yang telah ditetapkan. Selain itu, kinerja *Simulated Annealing* lebih baik dibanding *Genetic Algorithms* meskipun secara eksponensial menurun ketika kompleksitas puzzle melebihi ukuran 27. Rahayu dkk. (2017) dalam penelitiannya fokus pada Algoritma *Brute Force*, *Backtracking*, dan *Simulated Annealing* untuk diimplementasikan dan dianalisa waktu komputasinya ketika menyelesaikan Puzzle Sudoku. Hasilnya Algoritma *Backtracking* memiliki waktu komputasi lebih cepat dibandingkan dengan Algoritma *Brute Force*. Selanjutnya, ketika melakukan perbandingan waktu komputasi Algoritma *Backtracking* dan *Simulated Annealing* pada 4 tingkat kesulitan Puzzle Sudoku, diperoleh bahwa *Simulated Annealing* memiliki rata-rata waktu komputasi lebih kecil di level sulit. Hal ini menunjukkan bahwa *Simulated Annealing* lebih cepat dalam menyelesaikan Puzzle Sudoku pada tingkat kesulitan tinggi, meskipun keefektivitasan langkahnya bergantung pada solusi yang diperoleh sebelumnya.

Dari kelebihan dan kekurangan yang diperoleh peneliti-peneliti sebelumnya, penulis akan menggabungkan Algoritma *Backtracking* dan *Simulated Annealing* untuk menyelesaikan Puzzle Sudoku. *Simulated Annealing* adalah salah satu metode metaheuristik yang merupakan metode optimisasi stokastik yang dimodelkan pada perilaku materi yang terkondensasi pada suhu rendah. Algoritma *Backtracking* adalah sebuah algoritma urutan sistematis untuk mencari solusi yang dikemukakan oleh seorang matematikawan Lehmer pada tahun 1950. Dikarenakan Algoritma *Backtracking* memiliki waktu penyelesaian yang lebih lama dibanding *Simulated Annealing* sedangkan keefektivitasan *Simulated Annealing* tergantung pada inisialisasi yang diambil, maka penulis termotivasi untuk menggabungkan Algoritma *Backtracking* dan *Simulated Annealing* untuk menyelesaikan Puzzle Sudoku. Sejauh ini, belum ada penelitian terkait penyelesaian Sudoku dengan menggunakan gabungan kedua algoritma tersebut. Ide penggabungan algoritma

yang akan dilakukan penulis adalah dengan menggunakan Algoritma *Backtracking* untuk membangkitkan solusi awal bagi *Simulated Annealing* agar proses komputasi lebih cepat dan efektif. Selanjutnya, simulasi dan perhitungan waktu komputasi dari penggabungan kedua algoritma ini akan memanfaatkan bahasa pemrograman *Python*. Hasil dari penelitian ini diharapkan dapat menjadi alternatif penyelesaian masalah Puzzle Sudoku atau masalah lain yang memungkinkan untuk diselesaikan dengan penggabungan kedua algoritma tersebut.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, masalah yang dibahas penelitian ini adalah sebagai berikut :

1. Bagaimana menggabungkan Algoritma *Backtracking* dan *Simulated Annealing* untuk menyelesaikan Puzzle Sudoku?
2. Bagaimana kinerja gabungan Algoritma *Backtracking* dan *Simulated Annealing* dalam menyelesaikan Puzzle Sudoku?

## 1.3 Tujuan

Tujuan penelitian ini adalah untuk menggabungkan Algoritma *Backtracking* dan *Simulated Annealing* untuk menyelesaikan Puzzle Sudoku dan menganalisis kinerja algoritma gabungan tersebut.

## 1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Menambah wawasan mengenai implementasi gabungan algoritma *Backtracking* dan *Simulated Annealing* untuk menyelesaikan Puzzle Sudoku.
2. Menjadi salah satu alternatif penyelesaian Puzzle Sudoku.