BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Permainan Skyscraper *puzzle* merupakan permainan teka-teki angka, dimana setiap baris dan kolom harus diisi dengan angka-angka dengan aturan tertentu. Terdapat beberapa permainan yang sejenis dengan permainan Skyscraper *puzzle* ini seperti Sudoku dan Kakuro *puzzle*. Permainan Skyscraper *puzzle* terdiri dari kotak-kotak kecil yang membentuk baris dan kolom seperti teka-teki silang. Permainan Skyscraper *puzzle* berukuran nxn dengan beberapa petunjuk di sepanjang sisinya. Permainan ini dapat diselesaikan dengan menempatkan gedung yang direpresentasikan dengan angka pada kotak isian dengan ketinggian antara 1 sampai n. Pada satu baris dan kolom tidak diperbolehkan adanya 2 gedung yang memiliki ketinggian/nilai yang sama, selain itu jumlah gedung yang terlihat dari arah petunjuk harus sama dengan nilai petunjuk. permainan Skyscraper *puzzle* ini sudah dirancang sedemikian rupa sehingga hanya akan ditemukan tepat satu solusi.

Untuk menyelesaikan permainan ini dibutuhkan cara untuk menemukan solusi tersebut, semakin tinggi tingkat kesulitan *puzzle* maka semakin sulit pula menyelesaikannya. Untuk memudahkan manusia dalam menemukan solusi tersebut maka dibutuhkan algoritma yang mampu membantu menyelesaikan masalah tersebut. Pada dasarnya soal pada permainan ini dapat diselesaikan dengan analisis heuristik. Heuristik merupakan suatu strategi untuk melakukan proses pencarian ruang keadaan (*state space*) suatu permasalahan secara selektif, yang memandu proses pencarian yang kita lakukan di sepanjang jalur yang memiliki kemungkinan sukses paling besar, dan mengesampingkan usaha yang bodoh dan memboroskan waktu.

Ada beberapa algoritma yang sudah digunakan untuk menyelesaikan permainan Skyscraper *puzzle* ini diantaranya adalah *Brute force* dan *Backtracking*

algoritma *Brute Force* disebut juga algoritma naif algoritma ini melakukan pengecekan semua kemungkinan yang ada baik itu pada kondisi benar atau salah sedangkan algoritma *Backtracking* memiliki konsep yang sama dengan algoritma *Brute Force* tetapi algoritma *Backtracking* dapat memangkas kondisi salah sehingga tidak ada pengecekan pada kondisi salah [1]. Selain kedua algoritma diatas masih banyak algoritma yang dapat diterapkan pada game Skyscraper *puzzle* tersebut diantaranya adalah algoritma *Ant Colony Optimization* (ACO) yang merupakan salah satu algoritma metaheuristik.

Menurut Agus Leksono [2] pada penelitiannya mengenai algoritma *Ant Colony Optimization* (ACO) menjelaskan bahwa algoritma *Ant Colony Optimization* (ACO) ini sudah banyak diaplikasikan dalam berbagai permasalahan seperti TSP, QAP, JSP, pengaturan rute kendaraan, dan lain-lain. Pada penelitiannya ini Agus Leksono lebih memfokuskan algoritma ACO untuk menyelesaikan permasalahan TSP. Algoritma ACO sangat cocok untuk memecahkan permasalahan optimasi, terutama untuk masalah pencarian jalur terbaik.

Selain itu, terdapat beberapa algoritma yang sudah diterapkan pada game sejenis seperti pada penelitian yang dilakukan oleh Candi Nursadi [3] yang membandingkan algorima *Ant Colony Optimization* (ACO) *System* dengan algoritma Genetika untuk menyelesaikan permainan Sudoku *puzzle*. Sudoku *puzzle* ini merupakan permainan teka teki angka yang sama seperti Skyscraper *puzzle* tetapi memiliki peraturan yang berbeda.

Atas dasar uraian yang telah dijelaskan tersebut, penulis tertarik untuk mengambil pokok bahasan skripsi yang menerapkan algoritma *Ant Colony Optimization* (ACO) pada permainan Skyscraper *puzzle* dengan judul "Implementasi Algoritma *Ant Colony Optimization* Untuk Mencari Langkah Penyelesaian Dalam Permainan Skyscraper *Puzzle*".

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, masalah yang akan dibahas dalam penelitian ini adalah bagaimana cara menerapkan algoritma *Ant Colony Optimization* (ACO) pada permainan Skyscraper *puzzle*.

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah menerapan algoritma *Ant Colony Optimization* (ACO) untuk menyelesaikan permainan Skyscraper *puzzle*.

Sedangkan tujuan yang diharapkan akan dicapai dalam penelitian ini adalah mendapatkan solusi dalam penyelesaian masalah pada permainan Skyscraper *puzzle* dan dapat melihat kinerja dari algoritma yang diterapkan tersebut.

1.4 Batasan Masalah

Agar batasan menjadi terarah, maka permasalahan dibatasi terhadap masalah– masalah berikut :

1. Algoritma yang digunakan adalah algoritma *Ant Colony Optimization* (ACO).
2. Pemodelan dan perancangan sistem menggunakan pemodelan berbasis objek dengan tools UML (*Unified Modeling Language*).
3. Bahasa Pemrograman yang digunakan adalah C*# (C Sharp)*.
4. Ukuran *puzzle* yang digunakan adalah ukuran *puzzle* 4x4.
5. Kemungkinan ketinggian 1 dan 4 sudah ditetapkan dan dijadikan acuan untuk menjalankan algoritma *Ant Colony Optimization* (ACO) tersebut.

1.5 Metodologi Penelitian

Metode penelitian yang digunakan adalah metode deskriptif. Metode deskriptif adalah suatu metode dalam penelitian untuk membuat gambaran dengan cara mengumpulkan data, menganalisa data, membuat suatu pemecahan masalah, dan

kemudian disusun untuk menarik kesimpulan mengenai masalah tesebut. Teknik yang digunakan pada saat pengumpulan data dan pembangunan aplikasi adalah sebagai berikut:

1.5.1 Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini adalah menggunakan studi literatur. Studi literatur adalah teknik pengumpulan data dengan cara mengumpulkan literatur, jurnal, buku online dan bacaan-bacaan lainnya serta memanfaatkan media internet yang berkaitan dengan aplikasi yang akan dibangun.

1.5.2 Metode Pembangunan Perangkat Lunak

Metode yang digunakan dalam pembuatan perangkat lunak ini menggunakan model *waterfall. Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software* [4]. Motode *waterfall* melakukan pendekatan secara sistematis dan terurut, dimana tahap demi tahap yang akan dilalui harus menunggu tahap sebelumnya selesai. Tahap dari model *waterfall* adalah sebagai berikut :

1. Analisis Kebutuhan

   Merupakan tahap menganalisa hal-hal apa saja yang diperlukan dalam membangun sistem. Analisis yang dilakukan anatara lain adalah analisis game, analisis algoritma yang akan diimplementasikan pada sistem, analisis non-fungsional yang meliputi kebutuhan perangkat keras dan perangkat lunak, serta analisis fungsional.

2. Desain Sistem

   Proses perancangan data hasil analisis kebutuhan dan studi literatur dijadikan gambaran secara umum yang akan dibangun secara lengkap dalam bentuk diagram dan diimplementasikan pada tahap pengkodean.

3. Pengkodean Kode Program

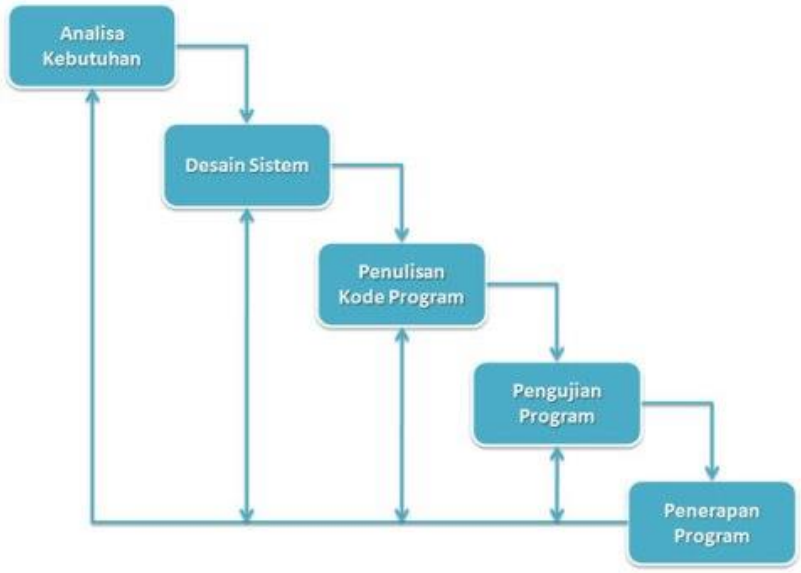
   Tahap ini merupakan tahap pengkodean yang sesuai dengan perancangan pada tahap desain sistem, pada tahap ini dilakukan implementasi algoritma *Ant Colony Optimization* pada sistem yang dibangun akan diterapkan.

4. Pengujian Program

   Tahap ini merupakan tahap setelah tahap pengkodean selesai. Tahap ini akan dilakukan pemeriksaan terhadap sistem yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

5. Penerapan Program

   Pada tahap penerapan program ini permainan Skyscraper *puzzle* sudah diserahkan kepada pengguna dan dilakukan evaluasi terhadap sistem untuk mengetahui apakah sistem telah memenuhi tujuan yang ingin dicapai.



Gambar 1.1 *Waterfall* Model [4]

1.6 Sistematika Penulisan

Sistematika penulisan laporan akhir penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini berisi penjelasan mengenai latar belakang masalah, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian serta sistematika penulisan.

BAB 2 LANDASAN TEORI

Membahas berbagai konsep dasar dan teori – teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisi permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang sudah pernah dilakukan sebelumnya termasuk sintesisnya.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Menganalisis masalah dari model penelitian, menggambarkan identifikasi masalah, analisis kebutuhan data, *hardware*, *software*, *brainware*, diagram pembangunan sistem dengan UML(*Unified Modeling Language*).

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Merupakan tahapan yang dilakukan dalam penelitian dalam menjelaskan implementasi, tampilan antarmuka, menu yang tersedia pada sistem dan pengujian terhadap sistem.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi mengenai hal-hal penting yang telah dibahas dan kemudian dibuat kesimpulan. Bab ini juga berisi saran-saran yang diberikan untuk pengembangan selanjutnya.