BAB I
PENDAHULUAN

I.1 Latar Belakang Masalah

Rubik's Cube adalah sebuah permainan *puzzle* mekanik tiga dimensi yang ditemukan pada tahun 1974 oleh seorang pemahat dan profesor arsitektur dari Hungaria bernama Erno Rubik. *Puzzle* ini disebut sebagai mainan yang paling banyak terjual di dunia dan sampai tahun 2010, telah lebih dari 350 juta buah Rubik's Cube yang terjual di seluruh dunia [1]. Rubik's Cube berbentuk kubus yang terdiri dari 26 kubus kecil yang dapat berputar pada porosnya. Setiap sisi dari kubus ini memiliki 9 permukaan yang terdiri dari enam warna yang berbeda dan diselesaikan dengan melakukan serangkaian langkah memutar kolom dan baris untuk menyeragamkan warna pada setiap sisinya setelah diacak.

Puzzle Rubik's Cube merupakan satu dari sekian persoalan kombinatorial yang cukup dikenal karena kerumitannya, dimana terdapat 43×10^{18} konfigurasi berbeda yang mungkin dihasilkan dengan mengacaknya [2]. Penyelesaian *puzzle* Rubik's Cube dengan mengandalkan intuisi maupun dengan langkah-langkah tertentu yang dihafalkan manusia biasanya memerlukan lebih dari 40 langkah [3]. Hal ini mendorong dilakukannya berbagai penelitian untuk mencari solusi optimal yaitu, serangkaian langkah terpendek diantara berbagai kemungkinan langkah alternatif yang ada untuk menyelesaikan suatu konfigurasi acak Rubik's Cube dengan mensimulasikannya pada program komputer dan menerapkan kecerdasan buatan untuk menyelesaikannya.

Beberapa algoritma telah diajukan untuk menemukan solusi optimal dalam menyelesaikan *puzzle* ini antara lain, algoritma Thistlethwaite oleh Morwen Thistlethwaite pada tahun 1981 [4], algoritma Two-Phase oleh Herbert Kociemba pada tahun 1992 [5], algoritma IDA* dengan Pattern Database oleh Richard E. Korf pada tahun 1997 [6], serta pengembangan algoritma Two-phase oleh Kociemba dan Thomas Rokicky pada tahun 2010 [3].

Sebagian besar penelitian yang ada untuk mencari solusi optimal pada *puzzle* Rubik's Cube menggunakan metode *brute force* dengan agen tunggal sehingga kurang praktis karena membutuhkan waktu lama dan memori yang besar untuk menyimpan simpul-simpul pencarian yang dibangkitkan [5][3]. Oleh karena itu diperlukan suatu pendekatan lain untuk mencari solusi optimal pada masalah *puzzle* Rubik's Cube. Salah satunya adalah pencarian dengan multi agen menggunakan metode *Swarm Intelligence,* dengan asumsi bahwa semakin banyak agen yang melakukan pencarian maka solusi dapat lebih cepat ditemukan.

Salah satu metode *Swarm Intelligence* adalah algoritma *Bee Colony Optimization* yang mensimulasikan perilaku mencari makan pada koloni lebah madu. Algoritma ini telah berhasil dikembangkan dan diimplementasikan untuk mencari solusi optimal dalam berbagai persoalan numerik dan kombinatorial, seperti *travelling salesman problem* (TSP) [7] [8], *puzzle* Sudoku [9] dan pencarian jalur terpendek [10].

Algoritma *Bee Colony Otimization* belum pernah diterapkan untuk menyelesaikan masalah pencarian langkah solusi terpendek pada Rubik's Cube, sehingga performansinya dalam menyelesaikan masalah tersebut masih belum diketahui. Cara kerja algoritma *Bee Colony Otimization* yang bersifat konstruktif dengan memperluas solusi-solusi parsial berupa solusi optimum lokal yang diperoleh dalam setiap tahapnya sampai menemukan suatu solusi yang optimal [11], dipandang sesuai untuk diterapkan dalam pencarian solusi optimal pada Rubik's Cube yang memiliki ruang pencarian yang sangat besar. Oleh karena itu penulis tertarik untuk mengeksplorasi kemungkinan menerapkan algoritma *Bee Colony Optimization* pada masalah Rubik's Cube dan mengambil pokok bahasan skripsi bagaimana menerapkan algoritma tersebut pada perangkat lunak untuk mencari solusi optimal pada *puzzle* Rubik's Cube dengan judul "Implementasi algoritma *Bee Colony Optimization* dalam pencarian langkah solusi terpendek pada *puzzle* Rubik's Cube**".**

I.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan pada sub bab I.1, maka yang menjadi pokok permasalahan adalah sebagai berikut :

1. Bagaimana mengimplementasikan algoritma *Bee Colony Optimization* pada suatu simulasi Rubik's Cube.
2. Bagaimana performansi algoritma *Bee Colony Optimization* dalam menyelesaikan masalah *puzzle* Rubik's Cube.

I.3 Maksud dan Tujuan

Maksud dari penelitian skripsi ini adalah menerapkan algoritma *Bee Colony Optimization* pada suatu simulasi Rubik's Cube.

Tujuan yang hendak dicapai dari penelitian skipsi ini adalah :

1. Mendapatkan langkah solusi terpendek dalam menyelesaikan masalah *puzzle* Rubik's Cube.
2. Mengetahui performansi algoritma *Bee Colony Optimization* dalam menyelesaikan masalah *puzzle* Rubik's Cube.

I.4 Batasan Masalah

Batasan Masalah yang ditentukan dalam penelitian ini adalah sebagai berikut :

1. Semua sisi yang tersusun atas kubus-kubus kecil penyusun Rubik's Cube akan menjadi simpul yang akan menjadi titik penelusuran.
2. Algoritma yang digunakan untuk mencari solusi adalah *Bee Colony Optimization* (*BCO*).
3. Parameter pengujian adalah jumlah langkah acak, jumlah lebah, jumlah iterasi, jumlah langkah solusi dan waktu pencarian.
4. Metode analisis perancangan yang digunakan adalah analisis perancangan dan pengembangan perangkat lunak berorientasi obyek, dengan menggunakan pemodelan *Unified Modeling Language* (*UML*).

5. Implementasi algoritma dalam penelitian ini menggunakan menggunakan bahasa pemrograman Java.
6. Visualisasi Rubik's Cube diadaptasi dari Rubik's Cube Java Applet versi 1.3.1. oleh Michael Schubart.

I.5 Metodologi Penelitian

Metodologi penelitian yang digunakan adalah metodologi Analisis deskriptif. Analisis deskriptif adalah metode analisis dengan mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya tanpa membuat kesimpulan yang berlaku untuk umum. Metodologi ini terbagi menjadi dua metode yaitu metode pengumpulan data dan metode pembangunan perangkat lunak.

1. Metode pengumpulan data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Literatur.

Studi Literatur adalah metode pengumpulan data dengan mengumpulkan teori-teori pendukung yang berhubungan dengan judul yang diambil melalui buku, jurnal, *paper* dan juga mengumpulkan data – data melalui website yang membahas tentang masalah *puzzle* Rubik's Cube dan algoritma *Bee Colony Optimization*.

b. Observasi.

Observasi adalah metode pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

c. Wawancara.

Wawancara adalah metode pengumpulan data dengan mengadakan tanya jawab dengan orang yang *expert* tentang masalah Rubik's Cube, kecerdasan buatan dan pemrograman java.

2. Metode pembuatan perangkat lunak.

Perangkat lunak dikembangkan dan dibangun dengan menggunakan metode waterfall [12], yang meliputi beberapa tahap yaitu:

a. *Requirements analysis and definition*

Tahap *Requirements analysis and definition* merupakan tahap pengumpulan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

b. *System and software design*

Tahap *System and software design* merupakan tahapmendesain perangkat lunak yang dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

c. *Implementation and unit testing*

Tahap *Implementation and unit testing* merupakan tahap hasil desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

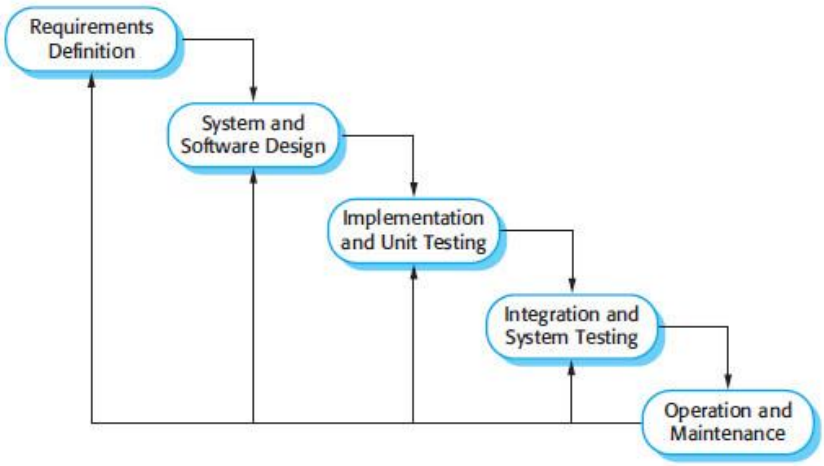
d. *Integration and system testing*

Tahap *Integration and system testing* merupakan tahap penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).

e. *Operation and maintenance*

Tahap *Operation and maintenance* merupakan tahap mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

Model proses metode pembangunan perangkat lunak *waterfall* dapat dilihat pada Gambar I.1



Gambar I.1 Model Proses Waterfall [12]

I.6 Sistematika Penulisan

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

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang permasalahan yang meliputi Rubik's Cube dan mekanismenya, masalah dalam hal mekanisme, solusi dan penelitian yang telah ada beserta masalahnya dan solusi yang ditawarkan beserta harapannya, mencoba merumuskan inti permasalahan yang dihadapi, menentukan maksud dan tujuan penelitian, metodologi penelitian, yang kemudian diikuti dengan pembatasan masalah serta sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Bab ini membahas subyek permasalahan yang diangkat yaitu Rubik's Cube beserta varian dan aturan permainannya, teori mengenai kecerdasan buatan, agen cerdas dan *Multi Agent System*, kemudian lebih spesifik membahas mengenai

swarm intelligence dan algoritma *Bee Colony Optimization,* teori graf dan representasinya, permasalahan optimasi dan pencarian jalur terpendek, yang kemudian diikuti oleh pembahasan mengenai teknik pemodelan dengan *UML* serta teknologi Java sebagai bahasa pemrograman yang digunakan.

BAB III. ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis masalah *puzzle* Rubik's Cube, analisis algoritma *Bee Colony Optimization,* analisis simulasi yang digunakan, analisis penyelesaian masalah dengan algoritma *Bee Colony Optimization,* analisis struktur data yang akan digunakan serta perancangan aplikasi yang akan dibangun.

BAB IV. IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan implementasi dari hasil analisis dan perancangan yang telah dibuat ke dalam bentuk aplikasi pemrograman, serta pengujian *white box* dan *black box* yang meliputi pengujian parameter algoritma yang diterapkan pada aplikasi yang telah dibangun dan perbandingan dengan algoritma lain.

BAB V. KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan akhir yang diperoleh dari hasil penulisan skripsi, serta saran untuk pengembangan lebih lanjut.