LOGIC : Jurnal Ilmu Komputer dan Pendidikan
Volume 2, No. 2, Tahun 2024
ISSN 2985-4172 (media online)
Hal 380-385

Implementasi Algoritma Bee Colony Optimization Dalam Mencari Langkah Solusi Tercepat Pada Puzzle Rubik's Cube

Vallian Dwi Sayoga[1]*, Rengga Herdiansyah[2]

[1,2]Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
Email: [1*]valiantdwi22@gmail.com, [2]dosen01101@unpam.ac.id
(* : coressponding author)

Abstrak − Puzzle Rubik's Cube adalah permainan teka-teki yang menantang, yang melibatkan pengaturan ulang blok-blok warna untuk mencapai konfigurasi yang diinginkan. Dalam penelitian ini, kami mengusulkan implementasi algoritma Bee Colony Optimization (BCO) untuk mencari langkah solusi tercepat dalam menyelesaikan Puzzle Rubik's Cube. Algoritma BCO terinspirasi oleh perilaku koloni lebah dalam mencari sumber makanan dan telah terbukti efektif dalam mencari solusi optimal dalam berbagai masalah optimisasi. Kami menerapkan pendekatan ini dengan memodelkan konfigurasi Rubik's Cube sebagai ruang pencarian dan menggunakan lebah sebagai agen pencarian yang bergerak di sekitar ruang pencarian untuk mencari langkah-langkah solusi yang meminimalkan jumlah langkah yang diperlukan untuk menyelesaikan teka-teki. Kami melakukan eksperimen dan perbandingan dengan pendekatan lain, seperti algoritma genetika dan algoritma pencarian heuristik, untuk mengevaluasi keefektifan algoritma BCO dalam menyelesaikan Rubik's Cube. Hasil eksperimen menunjukkan bahwa algoritma BCO mampu menemukan solusi tercepat dengan jumlah langkah yang lebih sedikit dibandingkan dengan pendekatan lain yang kami uji. Implikasinya, implementasi algoritma BCO dalam mencari langkah solusi tercepat pada Puzzle Rubik's Cube dapat membantu pemain dan penggemar Rubik's Cube dalam menyelesaikan teka-teki dengan lebih efisien. Penelitian ini berkontribusi pada pengembangan metode optimisasi dalam konteks pemecahan teka-teki dan dapat menjadi landasan untuk penelitian lebih lanjut dalam pengaplikasian algoritma BCO pada permasalahan optimisasi lainnya.

Kata Kunci: Rubik; Bee Colony Optimasi ; Puzzle

Abstract *− The Rubik's Cube puzzle is a challenging game that involves rearranging color blocks to achieve the desired configuration. In this study, we propose the implementation of the Bee Colony Optimization (BCO) algorithm to find the fastest solution steps in solving the Rubik's Cube puzzle. The BCO algorithm is inspired by the foraging behavior of bee colonies and has been proven effective in finding optimal solutions to various optimization problems. We apply this approach by modeling the Rubik's Cube configuration as a search space and using bees as search agents that move around the search space to find solution steps that minimize the number of moves required to solve the puzzle. We conduct experiments and comparisons with other approaches, such as genetic algorithms and heuristic search algorithms, to evaluate the effectiveness of the BCO algorithm in solving the Rubik's Cube. The experimental results show that the BCO algorithm is able to find the fastest solution with fewer moves compared to the other approaches we tested. The implications are that implementing the BCO algorithm to find the fastest solution steps in the Rubik's Cube puzzle can assist players and Rubik's Cube enthusiasts in solving puzzles more efficiently. This research contributes to the development of optimization methods in the context of puzzle solving and can serve as a basis for further research in the application of the BCO algorithm to other optimization problems.*

Keywords*: Rubic; Bee Colony Optimization; Puzzle*

1. PENDAHULUAN

Puzzle Rubik's Cube merupakan satu dari sekian persoalan kombinatorial yang cukup di kenal karena kerumitannya, dimana terdapat 43x10 konfigurasi berbeda yang mungkin dihasilkan dengan mengacaknya(Liwandouw, Vania Beatrice, 2017).

Beberapa algoritma telah diajukan untuk menyelesaikan solusi optimal dalam menyelesaikan puzzle ini antara lain, algoritma Thistlethwaite oleh Morweb Thistlethwaite pada tahun 1981, algoritma Two-Phase oleh Herbert Kociemba pada tahun 1992 , algoritma IDA* dengan Pattern Database oleh Richard E. Korf pada tahun 1997, serta pengembangan algoritma two-phase oleh Kociemba dan Thomas Rokicky pada tahun 2010(Tomas Rokicki, 2012).

Sebagian besar penelitian yang ada untuk mencari soolusi optimal pasa puzzle Rubik's Cube menggunakan metode brute force dengan agen tunggal sehingga kurang praktis karena

LOGIC : Jurnal Ilmu Komputer dan Pendidikan
Volume 2, No. 2, Tahun 2024
ISSN 2985-4172 (media online)
Hal 380-385

membutuhkan waktu lama dan memori yang besar untuk menyimpan simpul-simpul pencarian yang dibangkitkan(Tomas Rokicki, 2012).

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

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

a. Studi Literatur

Dalam penelitian ini 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. Wawancara.

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

2.2. Metode Pembuatan Perangkat Lunak

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

a. *Requirements analysis and definition*

Tahap ini 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 tahab mendisain perangkat lunak yang dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

c. *Implementasi and unit testting*

Tahap ini merupakan tahap hasil desain program diterjemahkan kedalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan.program yang dibangun langsung diuji baik secara unit.

d. *Integration and system testing*

Tahap ini merupakn tahap penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).

e. *Operation and maintenance*

Tahap ini merupakan tahap mengoprasikan program dilingkungan dan melakuakan pemeliharaan, seperti penyesuain atau perubahan karena adaptasi dengan situasi sebenarnya.

3. ANALISA DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai analisis dan perancangan pada sistem yang dibangun yaitu implementasi algoritma Bee Colony Optimization (BCO) untuk mencari langkah solusi tercepat pada puzzle Rubik's Cube.

3.1 Deskripsi Masalah

Solusi optimasi untuk menyelesaikan puzzle Rubik's cube adalah serangkakin langkah solusi terpendek untuk menghasilkan solusi tercepat dari suatu konfigurasi yang acak sampai mencapai konfigurasi yang terselesaikan yaitu warna permukaan cubies pada setiap sisi Rubik's Cube menjadi seragam.Solusi tersebut dapat direpresentasikan sebagai jalur terpendek diantara berbagai alternatif

LOGIC : Jurnal Ilmu Komputer dan Pendidikan
Volume 2, No. 2, Tahun 2024
ISSN 2985-4172 (media online)
Hal 380-385

jalur yang mungkin dilalui dari suatu keadaan Rubik's Cube yang acak menuju ke keadaan yang terselesaikan pada suatu graf berarah(Fabiana Meijon Fadul, 2019)

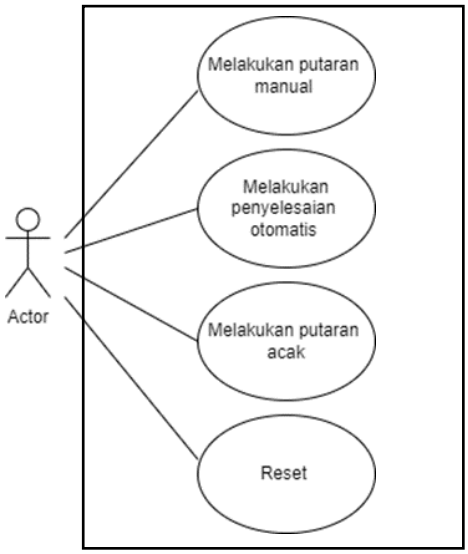
Algoritma koloni lebah yang digunakan adalah *Bee Colony Optimization* (BCO) dan akan diimplementasikan dalam pencarian jalur terpendek pada suatu simulasi Rubik's Cube.

3.2 Analisis Kebutuhan Fungsional

Pemodelan aplikasi yang akan dibangun dilakukan dengan metode analisis perancangan dan pengembangan perangkat lunak berorientasi objek dan menggunakan pemodelan Unified Modeling Language(UML).

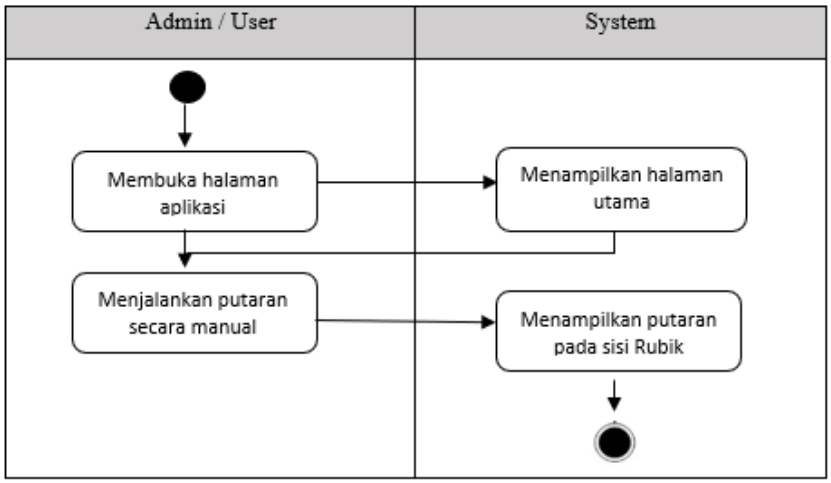
Karena hanya implementasi aplikasi maka pemodelan hanya mencangkup use case diagram, dan activity diagram. Proses yang dirancang diuraikan menjadi beberapa bagian yang dapat membentuk sistem tersebut menjadi satu kesatuan komponen.

a. Use Case Diagram



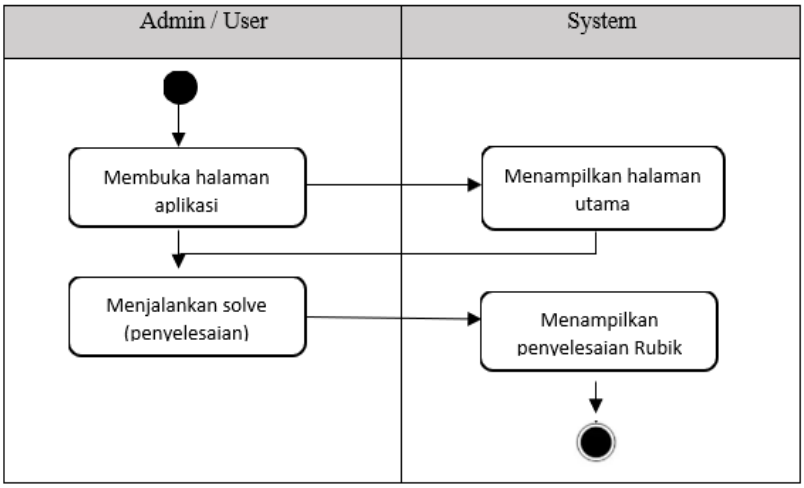
Gambar 1. *Use Case Diagram*

b. *Activity Diagram*

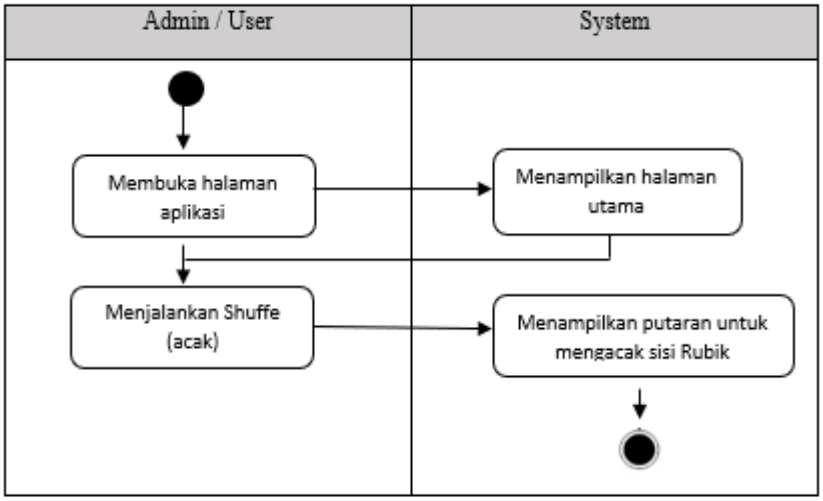


Gambar 2. *Activity Diagram* Putaran Manual

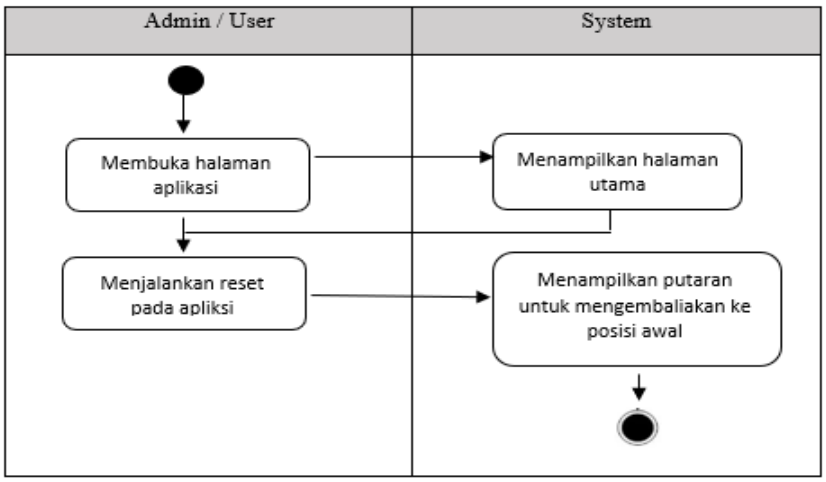
LOGIC : Jurnal Ilmu Komputer dan Pendidikan
Volume 2, No. 2, Tahun 2024
ISSN 2985-4172 (media online)
Hal 380-385



Gambar 3. *Activity Diagram* Putaran Otomatis



Gambar 4. *Activity Diagram* Putaran Acak

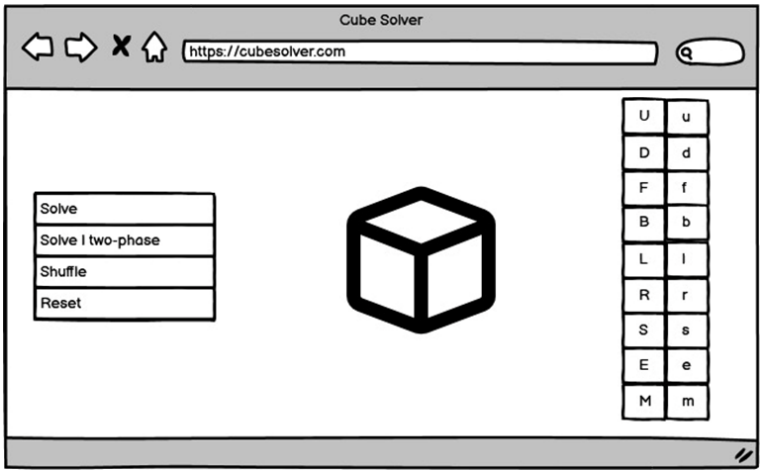


Gambar 5. *Activity Diagram* Untuk mereset

LOGIC : Jurnal Ilmu Komputer dan Pendidikan
Volume 2, No. 2, Tahun 2024
ISSN 2985-4172 (media online)
Hal 380-385

3.3 Perancangan Antarmuka

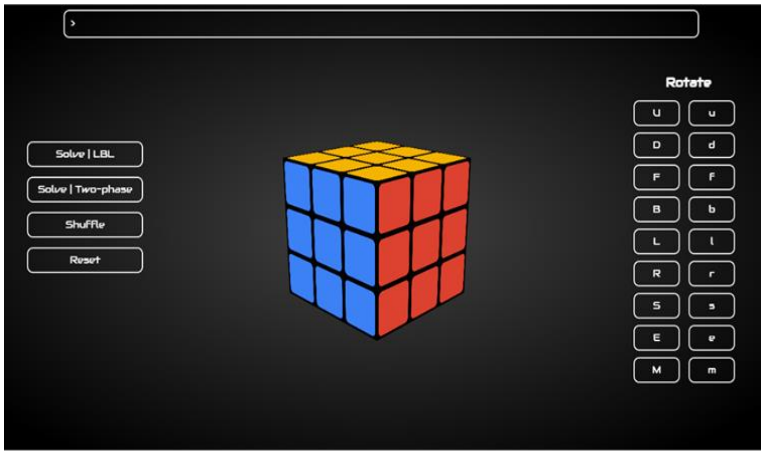
Perancangan interface atau perancangan antar muka merupakan bagian penting dari pengembangan suatu system,perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun sehingga akan mempermudah saat mengimplementasikan dan membangun aplikasi. Perancangan antar muka untuk aplikasi Rubik's Cube dengan BCO(Pambudianto Farizky, 2019). Berikut adalah rancangan antar muka yang akan dibuat pada system di jabarkan pada gambar 6. di bawah ini.



Gambar 6. Interface Halaman Utama Aplikasi

3.4 Implementasi Antarmuka

Berikut adalah implementasi antar muka yang akan dibuat pada system di jabarkan pada gambar 7.



Gambar 7. Tampilan Halaman Utama Pada Aplikasi

4. KESIMPULAN

Berdasarkan hasil yang didapat dalam penelitian dan penyusunan skripsi ini serta disesuikan dengan tujuan penelitian maka diperoleh kesimpulan sebagai berikut :

a. Algoritma BCO yang diterapkan dapat menghasilkan langkah solusi terpendek untuk menghasilkan solusi tercepat pada masalah puzzle Rubik's Cube.

LOGIC : Jurnal Ilmu Komputer dan Pendidikan
Volume 2, No. 2, Tahun 2024
ISSN 2985-4172 (media online)
Hal 380-385

b. Performa waktu algoritma BCO dalam menyelesaikan masalah puzzle Rubik's Cube pada kasus terburuk lebih lama dibandingkan dengan algoritma lain, karena terdapat kemungkinan terjebak dalam local optimum.Namun dari segi ruang algoritma BCO lebih efisien bila dibandingkan dengan algoritma lain karena jumlah maksimum simpul yang ditelusuri pada setiap kedalaman bersifat linear tergantung pada jumlah lebah.

REFERENCES

Liwandouw, Vania Beatrice, D. (2017). *Desain Algoritma Berbasis Kubus Rubik dalam Desain Algoritma Berbasis Kubus Rubik dalam Perancangan Kriptografi Simetris*. *April 2015*.

Tomas Rokicki, H. K. (2012). *God's Number is 20*. Cube20.Org. https://www.cube20.org/

Fabiana Meijon Fadul. (2019). *済無No Title No Title No Title*.

Pambudianto Farizky. (2019). *Perancangan interface*.

Gunawan, C. R., Ihsan, A., & Munawir, M. (2018). Optimasi Penyelesaian Permainan Rubik's Cube Menggunakan Algoritma IDA* dan Brute Force. *Jurnal Infomedia*, *3*(1), 37–42. https://doi.org/10.30811/jim.v3i1.627

Korf, R. E. (1997). Finding optimal solutions to Rubik's cube using pattern databases. *Proceedings of the National Conference on Artificial Intelligence*, 700–705.

Munawar. (2005). *Pemodelan Visual dengan UML*. 175–188.

Wajhillah, R., Bahri, S., Informatika, B. S., & Buatan, K. (2020). *PENGGUNAAN KECERDASAN BUATAN UNTUK PENYELESAIAN TEKA- TEKI KUBUS MENGGUNAKAN OPEN SOURCE COMPUTER VISION*. *8*(2), 177–181.