Vol. 13 No. 3S1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v13i3S1.8092

A WEB BASED 8-PUZZLE EDUCATIONAL GAME FOR RECOGNIZING SULAWESI'S ENDEMIC ANIMALS USING THE BFS ALGORITHM

Mutmainnah Muchtar[1*], Karmila Alamsyah Wellem[2]

[1*]Department of Computer Science, Universitas Sembilanbelas November Kolaka
[2]Department of Management, Sekolah Tinggi Ilmu Ekonomi Enam Enam Kendari

Keywords:
8 Puzzle;
Breadth First Search;
Educational game;
Sulawesi's endemic animals.

Corespondent Email:
muchtarmutmainnah@gmail.com

Abstract. *The advancement of AI (artificial intelligence) has provided new opportunities to enhance learning through interactive digital media. This study presents the development of a web based 8-Puzzle educational game designed to introduce Sulawesi's endemic animals using the BFS (Breadth First Search) algorithm. The system was developed using HTML, CSS, and JavaScript, and tested in a web browser environment on Windows 11. The waterfall method was applied during system development. Black-box testing ensured that all functions, such as shuffle, reset, navigation, and information display, operated correctly. Performance evaluation was carried out on 12 puzzle configurations, measuring solution steps, nodes expanded, and execution time. The results show that the BFS algorithm effectively solved all puzzles with optimal accuracy, producing solution times ranging from 1 ms to 140 ms depending on puzzle complexity. This research demonstrates how AI algorithms can be integrated into educational games to promote both problem solving skills and environmental awareness.*

1. INTRODUCTION

Advances in information technology have opened up significant opportunities for the development of interactive and educational digital learning media[1], [2]. One effective approach is game-based learning, which utilizes digital games as a learning medium that can increase user motivation and understanding of a topic[3]. This model has proven effective in combining entertainment and education, making the learning process more enjoyable and meaningful.

Artificial intelligence (AI) plays an increasingly vital role in education by enabling personalized, adaptive, and interactive learning experiences that enhance students' understanding and engagement[4]. The 8-Puzzle game is a classic example often used to introduce the concepts of search algorithms and state-space representation[5]. This game challenges the player or intelligent system to move eight numbered tiles in a 3×3 grid until they reach the correct configuration. Solving the 8-Puzzle game requires systematic exploration of the solution steps, making it suitable for learning algorithms such as Breadth First Search (BFS), Depth First Search (DFS), A-Star (A*) and Hill Climbing. Among these algorithms, BFS is known as a search method that guarantees the shortest solution because it sequentially searches each node in order of depth, although it requires more memory usage [6].

Several previous studies have implemented the BFS algorithm in the 8-Puzzle game to analyse its performance and computational complexity. A study by Balogun et al. [7] compared the performance of BFS, DFS and A* in an 8-puzzle game and showed that BFS is capable of producing optimal solutions despite requiring longer execution times. Ivanochko et al [8] evaluate the performance of two fundamental search methods: BFS and DFS, in solving the classic 15 Puzzle problem. Their research found that BFS effectively identifies the shortest solution path but demands greater memory resources. In contrast, DFS explores deeper paths more rapidly with lower memory usage, although it does not always guarantee the optimal solution. The study concludes that BFS is more suitable when solution optimality is required, while DFS is advantageous in contexts prioritizing speed and reduced computational cost. Diah, et al [9] conducted research that uses two search algorithms, BFS and DFS, to solve Sudoku puzzles. The comparison revealed that when both algorithms and humans arrived at similar solutions, it indicated careful construction of the Sudoku puzzle.

On the other hand, game-based learning has also been increasingly employed to promote local cultural and environmental awareness in Indonesia. Yahya et al.[10] developed an educational board game to raise awareness among children about endangered flora and fauna in West Bali National Park. Beta testing indicates that the game effectively increases children's understanding and awareness of biodiversity conservation in West Bali National Park. Saputra et al [11] develop an engaging learning medium to introduce Indonesian cultural heritage to students by designing an educational game called Marbel Budaya Nusantara in the form of a jigsaw puzzle to make learning more interactive and enjoyable. Based on beta testing, the game achieved a usability score of "satisfactory" category.

Sulawesi Island is known as one of the regions with the highest levels of biodiversity in Indonesia, and even in the world. Some of Sulawesi's characteristic species include the anoa (Bubalus depressicornis), tarsier (Tarsius spectrumgurskyae), maleo (Macrocephalon maleo), babirusa (Babyrousa celebensis), and bear cuscus (Ailurops ursinus). Unfortunately, public awareness, especially among the younger generation, of these endemic fauna remains relatively low [12]. This is partly due to the lack of engaging, contextual, and easily accessible learning media to introduce the region's rich fauna.

Based on this, this study proposes the development of a Web Based 8-Puzzle Educational Game for Recognizing Endemic Animals of Sulawesi Using the Breadth-First Search (BFS) Algorithm. This game not only functions as a means of entertainment and digital interaction, but also as a learning medium to introduce the basic concepts of search algorithms in artificial intelligence and increase user awareness of the richness of Sulawesi's local fauna. Thus, this study contributes to combining the fields of artificial intelligence, educational technology, and biodiversity conservation through an innovative and contextual game-based learning approach.

2. LITERATURE REVIEW

2.1. 8-Puzzle

8-Puzzle is a puzzle game consisting of numbered or illustrated squares that must be arranged in the correct order. The game consists of 8 squares and 1 empty space that can be moved up, down, right, or left. The puzzle is 3x3, so the squares can only move within that limit. Once shuffled, players must find a way to reassemble the puzzle [13]. It also serves as an excellent model for studying search algorithms and problem solving strategies [6].

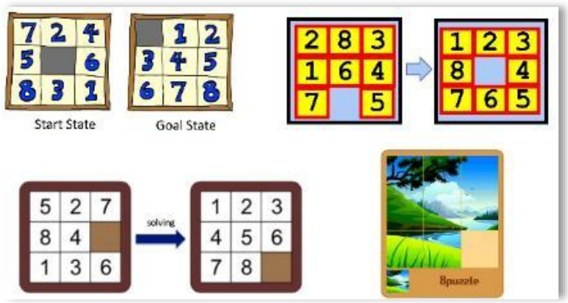

Figure 1. Different types of 8-puzzle games

2.2. *Sulawesi's Endemic Animals and Environmental Education*

Sulawesi is one of the most ecologically unique islands in Indonesia, located within the Wallacea region, which serves as a transition zone between Asian and Australian fauna. The island is home to numerous endemic species such as the Anoa (Bubalus depressicornis and Bubalus quarlesi), Tarsius (Tarsius spectrumgurskyae), Babirusa (Babyrousa celebensis), Maleo (Macrocephalon maleo), and Bear Cuscus (Ailurops ursinus), etc. [14]. However, public awareness and understanding of these species remain low, especially among younger generations. Environmental education through interactive media, such as digital educational games, offers an innovative approach to increase knowledge and appreciation of biodiversity. According to [15], digital learning platforms can play a crucial role in promoting environmental awareness and conservation values among students.

2.3. *BFS (Breadth-First Search) Algorithm*

The BFS algorithm is a fundamental search technique used in artificial intelligence to explore all nodes at a given depth before moving to the next level. It guarantees finding the shortest path to the goal if one exists [6]. BFS is widely used in pathfinding, puzzle solving, and state-space search problems, such as the 8-Puzzle or maze navigation[16]. Each node in the search tree represents a possible state of the problem, while edges represent transitions between states. The major advantage of BFS is its completeness, it will always find a solution if one exists, and optimality, meaning it finds the solution with the minimal number of steps[17]. In 8-puzzle game, BFS systematically explores all possible tile configurations until the goal state is reached, ensuring an optimal solution but often at the cost of higher memory usage. Implementing BFS in an educational game not only helps demonstrate AI problem-solving processes but also teaches learners about the logic behind search algorithms and optimal decision paths [18]. The following is the pseudocode of BFS algorithm:

```
Algorithm BFS(initial_state, goal_state):
Create an empty queue Q
Create an empty set VISITED
Enqueue (initial_state, []) into Q
// [] is the path to reach this state
  Add initial_state to VISITED
  while Q is not empty do:
  (current_state, path) ← Dequeue(Q)
    if current_state == goal_state:
     return path + [current_state]
     // solution found
    for each neighbor in
successors(current_state):
    if neighbor not in VISITED:
     Add neighbor to VISITED
Enqueue (neighbor, path + [current_state])
into Q

   return "No solution found"
```

3. RESEARCH METHODS

This research adopts an experimental approach through the design and development of an educational game based on the 8-Puzzle problem. The main objective is to demonstrate the application of Artificial Intelligence search algorithms, specifically BFS, in solving puzzle configurations, while simultaneously serving as a digital learning medium to introduce Sulawesi's endemic animals. The system was implemented using web-based technologies such as HTML, CSS, and JavaScript, enabling cross-platform accessibility on desktop and mobile browsers.

3.1. *Data Collection*

The dataset includes 12 endemic animal images from Sulawesi, such as Anoa Dataran Rendah, Tarsius, Maleo, Babirusa, Kuskus Beruang, Monyet Hitam Sulawesi, Julang Sulawesi, Serak Sulawesi, Cabai Sulawesi, Mandar Talaud, Anoa Pegunungan, and Musang Sulawesi. Each image is accompanied by a short descriptive text explaining its habitat, behavior, and conservation status. The images are used as fixed puzzle backgrounds divided into 3×3 grid pieces. All images were obtained from freely available websites. Table 1 shows the 12 endemic animal image datasets used in this study.

Table 1. Endemic Image Dataset

No	Animal's name	Image
1	Anoa Dataran Rendah	

2	Kuskus Beruang	
3	Burung Maleo Senkawor	
4	Babirusa Sulawesi Utara	
5	Tarsius	
6	Monyet Hitam Sulawesi	
7	Julang Sulawesi	
8	Serak Sulawesi	
9	Cabai Sulawesi	
10	Mandar Talaud	
11	Anoa Pegunungan	
12	Musang Sulawesi	

3.2. *System Architecture*

The system consists of three main components:

a. User Interface Layer: provides interactive puzzle gameplay, navigation buttons, and visual representation of the puzzle tiles.
b. Logic Layer: implements the BFS algorithm for automated puzzle solving and manages puzzle state transitions.
c. Content Layer: stores the fixed dataset of twelve endemic animal images and their corresponding descriptive information displayed upon puzzle completion.

The 8-puzzle board is represented as a 3×3 grid, where each tile corresponds to a numbered position (1–8) and one blank tile (0) for movement. The system supports user-initiated actions such as shuffling, manual solving, and automatic solving through BFS.

3.3. *BFS Implementation*

The BFS algorithm is implemented to find the optimal sequence of moves leading from an initial scrambled puzzle state to the goal configuration. The algorithm expands nodes level by level, exploring all possible moves (up, down, left, right) from the current blank tile position. Each puzzle configuration is stored as a state in a queue structure, ensuring the shallowest node (shortest path) is always explored first.

3.4. *Testing and Performance Measure*

The testing phase consisted of two main scenarios: black-box testing [19] and algorithm performance testing. Black-box testing was carried out to verify the functional correctness of the 8-Puzzle game, focusing on user interaction and interface behaviour. Performance testing measured the efficiency of the BFS algorithm using twelve puzzle configurations, each representing a different endemic animal. Three parameters were measured: the number of solution steps, the number of nodes expanded, and the execution time in milliseconds.

4. RESULT AND DISCUSSION

The development of the web based 8-Puzzle educational game was carried out using HTML, CSS, and JavaScript, designed and edited with Notepad++ as the primary text editor and executed through a standard web browser (Google Chrome or Mozilla Firefox). The system runs on a Windows 11 (64-bit) operating system, equipped with 16 GB of RAM and an AMD Ryzen 7000 series processor, ensuring stable and efficient performance during testing and development. The system architecture was built following the Waterfall development model, consisting of sequential stages: requirements analysis, design, implementation, testing, and evaluation[20], [21]. This model was selected because it provides a structured approach suitable for small to medium educational software projects.

The game integrates Artificial Intelligence principles through the implementation of the Breadth First Search algorithm for automated puzzle solving. The primary interface allows users to interactively play the 8-Puzzle game using twelve fixed images representing Sulawesi’s endemic animals such as Anoa, Tarsius, Maleo, Babirusa, etc. When the puzzle is successfully completed, an information pop-up displays descriptive details about the respective animal, combining both computational learning and environmental education. Figure 2 shows the User Interface of the 8-puzzle game of endemic animals of Sulawesi which has been run in the browser.

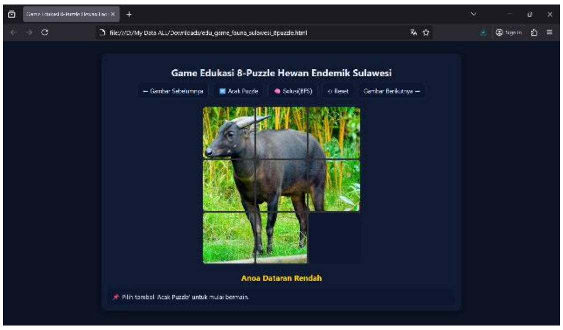

Figure 2. Interface of the game

Meanwhile, Figure 3 shows the interface when the puzzle is scrambled, while Figure 4 shows the information display of the endemic animals that appear when the puzzle is successfully assembled correctly.

Figure 3. Puzzled image

The `shuffle()` function in this system is responsible for randomizing the initial configuration of the 8-puzzle. Below is a code snippet to randomize the puzzle. It works by repeatedly swapping the blank tile (represented by 0) with one of its valid neighboring tiles, selected randomly in each iteration. The process runs 30 times to produce a sufficiently mixed yet still solvable puzzle state.

```
function shuffle(){
  for(let i=0;i<30;i++){
    const blank=state.indexOf(0);
    const n=neighbors(blank);
    const
pick=n[Math.floor(Math.random()*n.le
ngth)];

[state[blank],state[pick]]=[state[pi
ck],state[blank]];
  }
  buildBoard();
  log("Puzzle diacak");
}
```

Figure 4. The pop-up information

The `solveBFS()` function implements the Breadth-First Search (BFS) algorithm to automatically solve the 8-puzzle, as shown below. It begins by initializing a queue q that stores the current puzzle state, the position of the blank tile, and the path of moves taken. The algorithm explores all possible next states level by level using the First In First Out (FIFO) approach by removing the first element in the queue with `q.shift()`.When the goal state is found, the function calls `animate(path)` to visualize the sequence of moves leading to the solution. This BFS approach ensures that the shortest possible solution path is always found.

```
// BFS Solver
function solveBFS(){
  log('BFS mulai untuk ${animalImages[currentAnimal].name}...');
  const q=[[state.slice(),state.indexOf(0),[]]];
  const seen=new Set([state.toString()]);
  while(q.length){
    const [s,b,path]=q.shift();
    if(s.toString()===GOAL){ animate(path); return; }
    for(const nb of neighbors(b)){
      const ns=s.slice();
      [ns[b],ns[nb]]=[ns[nb],ns[b]];
      const key=ns.toString();
      if(!seen.has(key)){
        seen.add(key);
        q.push([ns,nb,[...path,nb]]);
      }
    }
  }
  log("BFS gagal menemukan solusi");
}
```

Two testing scenarios were conducted to evaluate the system's functionality and performance. The first scenario, black-box testing, focused on verifying whether all user interface components and system functions operated as intended, such as image display, shuffle, reset, and BFS solving features. The tests were performed by executing each function from the user interface and observing the output response. Table 2 presents the detailed black-box testing scenarios, procedures, and expected outcomes.

Table 2. Blackbox testing

No	Testing Scenario	Test Steps	Expected Result
1	Displaying endemic animal images	Run the application and select an animal from the list	The animal image appears correctly in the puzzle pieces.
2	Shuffling the puzzle	Press the Shuffle button	The puzzle pieces change positions randomly but remain solvable
3	Using the BFS algorithm for automatic solving	Press the Solve BFS button	The puzzle automatically rearranges itself into the correct order
4	Displaying animal information after puzzle completion	Complete the puzzle manually or automatically	An information box appears showing the description of the endemic animal
5	Function of Reset and navigation buttons	Press the Reset / Next / Previous buttons	The image changes or returns to its initial position correctly without errors

All test cases were executed successfully with outputs matching the expected results. The system correctly displayed all animal images, generated solvable random puzzles, executed the BFS solving algorithm without errors, and displayed informative descriptions upon completion. The Reset and navigation buttons also functioned reliably, confirming that the web-based game met all functional requirements and provided a stable and interactive user experience suitable for educational use

The second scenario, algorithm performance testing, measured the computational performance of the BFS algorithm using twelve puzzle configurations. The parameters recorded in this testing included the number of solution steps, the number of expanded nodes, and the execution time in milliseconds. The goal was to analyse the computational efficiency and scalability of the BFS algorithm in solving varying levels of puzzle complexity. Table 3 presents the summarized results of the performance testing.

Table 3. BFS performance measurement

Puzzled Images	BFS Performance Result
 Test 1	Number of solution steps: 16 Nodes expanded: 7325 Execution time: 20.00 ms
 Test 2	Number of solution steps: 8 Nodes expanded: 201 Execution time: 2.00 ms
 Test 3	Number of solution steps: 14 Nodes expanded: 4470 Execution time: 15.00 ms
 Test 4	Number of solution steps: 10 Nodes expanded: 897 Execution time: 4.00 ms
 Test 5	Number of solution steps: 8 Nodes expanded: 291 Execution time: 2.00 ms
 Test 6	Number of solution steps: 10 Nodes expanded: 446 Execution time: 1.00 ms
 Test 7	Number of solution steps: 16 Nodes expanded: 9366 Execution time: 33.00 ms
 Test 8	Number of solution steps: 8 Nodes expanded: 261 Execution time: 1.00 ms
 Test 9	Number of solution steps: 20 Nodes expanded: 38182 Execution time: 140.00 ms
 Test 10	Number of solution steps: 8 Nodes expanded: 311 Execution time: 2.00 ms
 Test 11	Number of solution steps: 10 Nodes expanded: 436 Execution time: 1.00 ms
 Test 12	Number of solution steps: 18 Nodes expanded: 22397 Execution time: 81.00 ms

The experimental results show that the BFS algorithm successfully solved all twelve 8-Puzzle configurations of Sulawesi's endemic animal images. The algorithm demonstrated consistent completeness, meaning it was able to find the correct solution path for every test case. However, its performance varied depending on the complexity of the initial puzzle state. The number of solution steps ranged from 8 to 20 moves, reflecting the differences in puzzle difficulty. Correspondingly, the number of expanded nodes ranged between 201 and 38182, showing that BFS explores a large number of possible states before reaching the solution, especially in more scrambled configurations. Execution time also increased with complexity, from as low as 1.00 ms for simple puzzles to as high as 140.00 ms for complex ones. These findings confirm that while BFS guarantees an optimal solution, it requires significant computational resources as the state space grows. BFS performs efficiently for small problem spaces like the 8-Puzzle but may become computationally expensive for larger or more complex search problems.

From an educational perspective, the integration of the BFS algorithm into a visual and interactive game successfully bridges AI learning and environmental education. Players not only understand the concept of systematic search but also learn about Sulawesi's endemic animals including Anoa, Tarsius, Maleo, and Babirusa, etc. The automatic solving feature also provides visual feedback that helps learners comprehend how BFS explores and reconstructs the state space step by step.

In summary, the testing results demonstrate that the developed system operates correctly, performs efficiently under realistic conditions, and fulfils its technical objectives. The combination of AI based problem solving and biodiversity learning serves as a model for future interactive educational tools integrating computer science concepts with local environmental content.

5. CONCLUSION

a. The web based 8-Puzzle educational game was successfully developed using the BFS (Breadth First Search) algorithm to automatically solve puzzle configurations while introducing Sulawesi's endemic animals as an interactive learning medium. The system runs smoothly on web browsers and provides an engaging way to combine artificial intelligence with environmental education.
b. Testing results demonstrate that the BFS algorithm consistently produces optimal solutions across all 12 puzzle scenarios, with measurable parameters including the number of solution steps, expanded nodes, and execution time. These findings confirm that BFS is effective for small scale search problems such as the 8-puzzle.
c. Despite its accuracy, BFS has limitations in memory efficiency when handling larger puzzles. Future work may include implementing alternative algorithms such as A*, or Iterative Deepening Search, and enhancing the system with features like gamification, user scoring, and mobile compatibility to improve educational impact and performance.

DAFTAR PUSTAKA

[1] T. E. Coleman and A. G. Money, "Student-centred digital game–based learning: a conceptual framework and survey of the state of the art," *High Educ (Dordr)*, vol. 79, no. 3, pp. 415–457, Mar. 2020, doi: 10.1007/s10734-019-00417-0.

[2] J. Y. Sari, Y. P. Pasrun, M. Muchtar, R. Karim, and R. A. Saputra, "Development of Multimedia-Based Online Learning Media for Class VI Students of SDN 105 Kendari," *MEKONGGA: Jurnal Pengabdian Masyarakat*, vol. 1, no. 1, pp. 1–6, 2024, doi: 10.69616/mekongga.v1i1.170.

[3] M. H. Hussein, S. H. Ow, L. S. Cheong, M. K. Thong, and N. Ale Ebrahim, "Effects of Digital Game-Based Learning on Elementary Science Learning: A Systematic Review," *IEEE Access*, vol. 7, pp. 62465–62478, 2019, doi: 10.1109/ACCESS.2019.2916324.

[4] S. Rifky *et al.*, *Artificial Intelligence: Teori dan Penerapan AI di Berbagai Bidang*. PT. Sonpedia Publishing Indonesia, 2024.

[5] H. Mustafidah and B. Nurdiyansah, "Rancang Bangun Aplikasi Penyelesaian Puzzle 8 Angka Menggunakan Metode Hill Climbing," vol. 16, no. 1, 2019.

[6] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. United Kingdom: Pearson Education, 2021. [Online]. Available: www.pearsonglobaleditions.com

[7] G. B. Balogun, D. Ibisagba, A. Bajeh, T. O. Debo, A. Muyideen, and O. J. Peter, "Comparative analysis of AI-based search algorithms in solving 8 puzzle problems," *Bull Natl Res Cent*, vol. 48, no. 1, Nov. 2024, doi: 10.1186/s42269-024-01274-3.

[8] I. Ivanochko, M. GregušMl, and S. Treiová, "Optimizing the 15 Puzzle with AI: Comparative analysis of breadth-first and depth-first search algorithms," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 57–64. doi: 10.1016/j.procs.2025.07.156.

[9] N. M. Diah, S. Riza, S. Ahmad, N. Musa, and S. Hashim, "Sudoku solutions: a comparative analysis of breadth-first search, depth-first search, and human approaches," *Journal of Education and Learning*, vol. 19, no. 1, pp. 561–569, Feb. 2025, doi: 10.11591/edulearn.v19i1.21214.

[10] S. Yahya, A. Alifah, and A. A. Kusumasari, "Pengembangan Serius Game Pada Pengenalan Fauna di Taman Nasional Bali Barat," *Journal of Art, Design, Art Education & Culture Studies*, vol. 9, no. 1, 2024.

[11] A. Agung Saputra, F. Nonggala Putra, and R. Darma Rusdian Yusron, "Pembuatan Game Edukasi Pengenalan Kebudayaan Indonesia Menggunakan Metode Game Development Life Cycle (GDLC) Berbasis Android," 2022.

[12] M. Christita *et al.*, "Pendidikan Konservasi Satwa Endemik Sulawesi Anoa (Bubalus spp.) Melalui Anoa School Outreach di Sulawesi Utara," in *Proceedings of the Seminar Nasional Biologi Inovasi Penelitian dan Pembelajaran Biologi II (IP2B II)*, Surabaya, Feb. 2018, pp. 69–74.

[13] A. M. Tianti and R. C. P. Fanani, "Implementasi Algoritma A* untuk Stimulasi Kognitif Anak Usia 5-6 Tahun Melalui Permainan 8 Puzzle," *Melek IT Information Technology Journal*, vol. 10, no. 1, pp. 82–88, 2024.

[14] Saroyo, P. Siahaan, M. L. D. Langoy, and R. Koneri, "Pendidikan Konservasi Satwa Endemik Sulawesi bagi Siswa Sekolah Dasar di Kelurahan Batuputih Bawah, Kecamatan Ranowulu, Kota Bitung, Sulawesi Utara," *Vivabio Jurnal Pengabdian Multidisiplin*, vol. 1, no. 3, pp. 26–32, Dec. 2019.

[15] F. Firmansyah, M. Rahayu, and G. Ginanjar, "Peningkatan Kemampuan Literasi Konservasi melalui E-Modul Keanekaragaman Hayati di Cagar Alam Rawa Danau Berbasis Discovery Learning pada Siswa SMA," *Biosfer: Jurnal Biologi dan Pendidikan Biologi*, vol. 10, no. 1, pp. 58–64, Jun. 2025.

[16] S. L. Pardede, F. R. Athallah, Y. N. Huda, and F. D. Zain, "A Review of Pathfinding in Game Development," *[CEPAT] Journal of Computer Engineering: Progress, Application and Technology*, vol. 1, no. 01, p. 47, May 2022, doi: 10.25124/cepat.v1i01.4863.

[17] A. Suswiantara, N. Dzakhwan, A. Fahruddin, M. N. Dzaky, A. Auliana, and T. Setiadi, "Penerapan Algoritma Best First Search (BFS) untuk Penentuan Slot Parkir," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 4, pp. 1606–1613, Oct. 2024, doi: 10.57152/malcom.v4i4.1541.

[18] R. Rahim *et al.*, "Congklak, a traditional game solution approach with breadth first search," in *MATEC Web of Conferences*, EDP Sciences, Sep. 2018. doi: 10.1051/matecconf/201819703007.

[19] N. Ningsi, S. Sunyanti, and S. Sarimuddin, "Perancangan e-Marketplace Maswira (Masyarakat Pesisir berwirausaha) pada Dinas Pertanian dan Perikanan Kab. Bombana," *Jurnal Ilmiah Edutic : Pendidikan dan Informatika*, vol. 8, no. 1, pp. 51–58, 2021, doi: 10.21107/edutic.v8i1.12064.

[20] E. Bayu Kristanto, S. Andrayana, U. Nasional, J. Sawo Manila, K. Jakarta Selatan, and D. Khusus, "Application of Waterfall SDLC Method in Designing Student's Web Blog Information System at the National University," 2020. [Online]. Available: https://iocscience.org/ejournal/index.php/mantik

[21] I. Asriani, M. Muchtar, R. R. Ismail, and A. Paliling, "Expert System for Determining Diseases and Pests in Seaweed Using Forward Chaining (Case Study : Watorumbe Village , Mawasangka Tengah)," vol. 1, no. 1, 2024, doi: 10.69616/mcs.v1i1.175.