

# Intelligent Tutoring System Design Using Markov Decision Process

Burhan Aji S\*, Dwijoko Purbohadi, Slamet Riyadi

*Universitas Muhammadiyah Yogyakarta, Jln.Brawijaya, Tamantirto, Kasihan,  
Bantul, Yogyakarta 55183, Indonesia*

*\*Corresponding author: burhan.aji.2016@ft.umy.ac.id*

## **Abstract**

*Class learning is a teaching and learning activity involving both teachers and students. Students in the class have different levels of intelligence, divided into three categories: lowest, average, and most intelligent. Teachers usually pay less attention to less intelligent students. Consider a student who is not intelligent enough to remain in school. However, if a smart student is required to repeat content he already comprehends, he will become bored. Therefore, this study developed an intelligent tutoring system (ITS)-based learning application with a Markov decision process (MDP). The combination of ITS and MDP enabled intelligence to determine the most appropriate subsequent step by delivering tutorials and examination questions that guide students to efficiently attain intelligence levels.*

**Keywords:** *CourseLab, Markov Decision Process (MDP), Application, Intelligent Tutoring System (ITS)*

## **1. Introduction**

Learning is an essential component of education and significantly impacts its overall quality. Diverse strategies have been devised to enhance the quality of education by leveraging its supporters' resources. A significant concern in elementary schools is the high number of tutorials used to increase the quality of learning through diverse media, such as videos, books, and interaction with the environment. If the learning process is the standard, students must be brought to a wise and learning-objective-compliant state. The intriguing challenge that emerges is how these students reach optimum learning objectives while still being able to participate and truly comprehend the materials. Ideal learning entails forming a learning process that can bring all students to the same level of intelligence.

Ideal learning can be attained if it is conducted between a single teacher and a single student (private learning), with sufficient time, equipment, and procedures [1]. Under such conditions, it is feasible for the student to achieve mastery because the teacher has a firm grasp on what to do when the student encounter obstacles. In addition to being able to learn, a teacher can also construct a learning process. However, in modern elementary schools, a teacher manages many students. The teacher has the discretion to teach as effectively as possible, but the time to monitor and assist each student with their learning issues is limited. In addition, each student's learning rate, learning style, and difficulty level vary greatly. These situations increase the likelihood that students will attain a low level of intelligence since they are further from the three learning characteristics: smart, moderately intelligent, and falling behind in materials.

The learning process in a technology can be grouped into 2 namely learning in the network and learning outside the network. Outside learning the network can be said where a teacher meets face to face with his students like in class and private lessons. While learning in the network can see videos on YouTube or other media that use

the internet network. Model learning in the network (online) is divided into 2 again, namely: deep learning network (online) synchronously, and learning in the network (online) asynchronously. Synchronous Online Learning is learning that uses internet network by utilizing an application where teachers and students are siwi communicate at the same time without pause [2].

Currently created intelligent tutorial in other words as Intelligent Tutoring System (ITS). ITS is a purposeful computer system to provide direct and specific instructions or feedback to students or learners, usually without requiring intervention from the teacher. In ITS a tutorial can be interpreted as State. To move from one state to another, we will try to use Markov Decision Process (MDP). The Markov process is a model mathematics useful in research of complex systems the basic concepts of processes Markov is from the "State" of a system and the "transition" state. Therefore it will be on try implementing the markov into an ITS [3].

Alan Turing (1950) described a hypothetical test for assessing machine intelligence which became known as the Turing test. Basically, test it will ask someone to communicate with two other agents, human and computer asking questions to both recipients. The Turing test at its core has been used for more than two decades as the current ITS development model. The ultimate goal for an ITS system is to communicate effectively [4].

Implementing a pedagogical strategy, which decides the next action the tutor should take at every step of the way student learning process among a set of alternative ways. Use strategy pedagogically, ITS is able to interact adaptively with students by taking different actions in various short-term situations to improve performance student learning in the long term. However, encouraging pedagogical strategies at ITS indeed challenging [5].

In 2013 Jonathan Paul made a plan narrative-centered tutorial with Markov Decision Process (MDP) [6]. On guidance one-on-one, face-to-face teaching has been considered the gold standard for teaching effective because of its significant pedagogical benefits compared to classroom lectures traditional. Work with a smart guidance system that belongs to the business community managing effective tutoring such as personalized feedback, scaffolding, and problem sets for various educational subjects. According to him most intelligent tutoring systems have no built-in mechanisms to reason interactive narrative plotts; learning scenarios in the guidance system intelligent usually abstracts narrative context to focus on educational subject matter. Embed intelligent tutoring systems in a learning-centered environment narrative requires integration with capable intelligent narrative technologies and managing interactive stories [7].

The representation of the state deviates from the presumption from Markov. It emphasizes plot states rather than plot trajectories. While country representation was chosen to avoid data dispersion problems over the model training, may have limited ability to identify story trajectories it suggests a need for a learning process.

Then in 2014 create an E-Learning Design for a learning model with feed control come back. In the current condition, a lecturer handles a large number of students so there are limitations to giving proper attention and treatment [8]. As a result, the possibility of students achieving small mastery. Various learning methods have developed with the hope of having an effect size close to ideal learning, one of them uses the concept of Master Learning (ML). Adjusted against the background of existing problems, the ITS model is designed for models tutorial drill, meaning that it is only applied to instructional learning so that intelligent characteristics built

with markov algorithm [9]. From the results they make is to show how the data education can be modeled as MDP, to show that policy the resulting decisions successfully guide the system to a goal state in the simulation, and demonstrate that the MDP approach can be used to send useful suggestions. easy to follow for tutors and students [10].

## 2. Method

The following are the steps involved in developing an application's design.



Figure 1 Application Design

### 2.1. Requirement Analysis

The analysis of requirements for the design and implementation of this application was collected through supervisor interviews. The following are the requirements analysis results through interviews for the application.

- a. The materials distributed were fractional. The data were then split into three difficulty levels: common, mixed, and decimal fractions.
- b. There was a question test page to determine the initial students' condition. The test consisted of ten questions.
- c. There was a page with information on common, mixed, and decimal fractions. The common fractions were grouped into addition, subtraction, and division. Mixed fractions were divided into addition, subtraction, and division. Meanwhile, decimal fractions were subdivided into addition, subtraction, division, and multiplication.
- d. There was a question practice section to assess students' level of comprehension. There was one question used for practice purposes. However, while repeating activities with the right answers, students were asked different questions each time.

- e. There was a test page to measure the degree of comprehension once information and practice questions had been presented. There was a maximum of ten questions in the test.
- f. There was a comparison between the practice and the examination scores on the score page.
- g. The system utilized the Markov decision process (MDP) to make decisions based on different learning sequences. Those who were falling behind and reasonably intelligent attained their respective levels of intelligence rapidly.

## 2.2. Application Design

Creating a design is the most crucial aspect of application development since it helps programmers construct each level's sequence of learning topics.

## 2.3. Application Development

Several tools and materials were required to enable the design and implementation of the ITS application utilizing the MDP algorithm throughout the application development phase.

### Tools

This application development utilized both hardware and software. The hardware was a personal computer (PC) with the following specifications.

- a. AMD A10 Processor
- b. 8GB RAM
- c. Windows 10 Education
- d. 1TB Harddisk

Meanwhile, the software used included the following.

- a. App.diagrams.net

It refers to software used for various diagrams, ranging from flowcharts to entity relationship diagrams (ERD).

- b. Balsamiq Mockups

It is an application program to develop an application's user interface display. Typically, this program generates an application's initial display or design before being implemented into the application.

- c. CourseLab 3.0

It is an interactive learning media for students to study independently or traditionally. This work intends to construct an MDP algorithm to develop a learning application.

- d. Notepad++

It is a text and code editor that supports many programming languages, including JavaScript, utilized in CourseLab.

### Materials

Interviews with the supervising lecturer and teachers at SD Negeri Ngebel yielded the data required to develop this application. The acquired materials served as the basis for designing and implementing ITS utilizing MDP. The MDP formula was given by the following equation.

$$V^{\pi^*}(s) = EIR_{max} * (1 - \frac{step}{20}) + \max_{a \in A} \sum_{s' \in P_{S\pi(s)}(s')} V^*(s') \quad (2-7)$$

## Application Development

The application development and algorithm implementation were described in detail at this stage.

### a. Establishing a New Project in CourseLab

It began with opening the CourseLab 3.0 application, followed by filling in the name and the folder. The location could be adjusted. Subsequently, the module name was filled, continued with selecting simple and clicking finish.

### b. Algorithm Implementation

It proceeded by opening the project file created earlier, opening folder 1, right-clicking the mouse on the module file, and selecting edit with Notepad++. The following is the implementation of the MDP formula, converted into a code.

```
Learning_State=0;
NextState=[];
V=0;
V_phi=0;
Optimum_point = 0;
while (V <= Reward_max-0.1)
{V=V+0.1;
for (element = 1; element < Tuple.length; element++)
{V_phi = EIR + 0.65*V*Tuple[element];
Difference_e = V - V_phi;
Difference_e = Math.sqrt(Difference_e*Difference_e);
if (Optimum_point<Difference_e)
{Optimum_point = Difference_e;
Learning_State = Tuple[element];
NextState [0] = Learning_State;
NextState [1] = V_phi;
else {};
```

### c. Creating Materials

One example of the material for smart students was added at this stage, executing by filling the text with sample questions and adding navigation buttons from the object of selecting slides for questions.

#### d. Constructing Exercise Questions

At this stage, a new slide was created to add a question to test the algorithm. Making questions was performed by adding a test from the object and then clicking the object, followed by right-clicking the object, selecting an action, filling in on success, and adding the following JavaScript content.

```
"CLV.oGlobal['skor11']=1;" skor11 adalah nama object dari test arti 1
```

adalah ketika dia menjawab benar akan dapat 1 untuk logika disistem.

Kemudian isi pada on failure tambahkan javascript

```
"CLV.oGlobal['skor11']=0;" arti 0 adalah ketika dia menjawab salah akan
```

mendapatkan 0 untuk logika disistem. Tambahkan navigation button

klik kanan button pilih action, isi pada on click tambahkan javascript

```
isi "var Test = CLV.oGlobal['skor11']; var Score =
```

```
CLV.oGlobal['Score']; var Step = CLV.oGlobal['Step'];
```

```
CLV.oGlobal['Step']=Step+1;GetOptimumStep(Score,Step,Test);"di
```

button ini lah berfungsi untuk perhitungan algoritma tersebut.

## 2.4. Application Testing

Two types of testing, white box and black box, were performed. On the one hand, white box testing was utilized for system testing to guarantee that all commands and conditions were executed at a minimum. On the other hand, black box testing was applied to test the specific features of a software application. The following table exhibits the plan for examining test cases and scenarios, along with the test types.

Table 1 Test Plan

No	Test Case	Test Scenario	Test Type
1	MDP algorithm	Testing the learning application algorithm	White Box
2	Exercises	Answering practice questions	Black Box
		Start button	Black Box
3	Theory	Continue button	Black Box
4	Exercises	Answering practice questions	Black Box
		Continue button	Black Box
5	Examination	Completing the examination	Black Box
		Continue button	Black Box

## 3. Analysis Results

### 3.1. Application Display

The application was designed using CourseLab and JavaScript for the algorithm. The result was then implemented in the application development.

### 3.2. Exercises



Figure 2 Exercise

Figure 2 depicts a practice question page displayed the first time the application was launched. This page acted as the problem's introduction and measured the student's comprehension degree while working on the problem. This page exhibited ten questions that should be completed in ten minutes. Students must click the start button to begin the exercise.

### 3.3. Smart-level topics

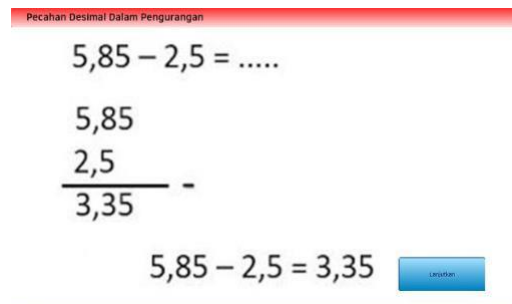


Figure 3 Smart-level topic

Figure 3 illustrates the eighth topic for students with high intelligence. This section contained example questions solely. Students were expected to comprehend this topic after completing tasks on the previous topic. Once the students comprehended the material, they should click the continue button to go to the question page.

### 3.4. Moderate-level topics

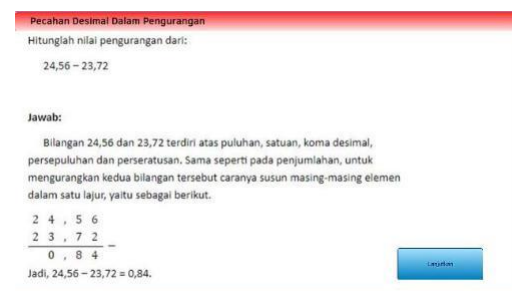


Figure 4 Moderate-level topic

Moderately skilled students were provided with a topic presentation, as displayed in Figure 4. There were some sample questions and answers provided on this page. The students comprehending the material should proceed to the question page by clicking the continue button.

### 3.5. Lagging-level topics



Figure 5 Lagging-level topic

Another method for catching up with behind-the-curve students is exhibited in Figure 5. Questions, answers, and videos were all displayed on the page. A window with further details emerged when selecting the explain button. Subsequently, selecting the sample question tab brought up a pop-up with several examples. There was a continue option for students to click if they were confident in their knowledge and ready to go to the question page.

### 3.6. Practice questions on each topic and type of exercises

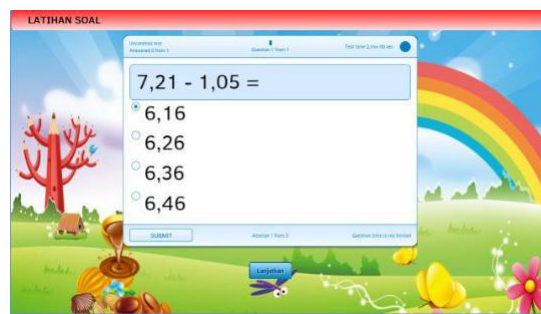


Figure 6 Practice questions on each topic and type of exercises

Figure 6 displays practice questions dependent on student intelligence after studying the topic. This page served to determine the next steps for the students. A student's answers to the questions determined whether or not he moved on to the next topic. After completing the practice questions, the students should click the submit button to save their answers and then click the continue button to learn the objective of the following topic.

### 3.7. Examination

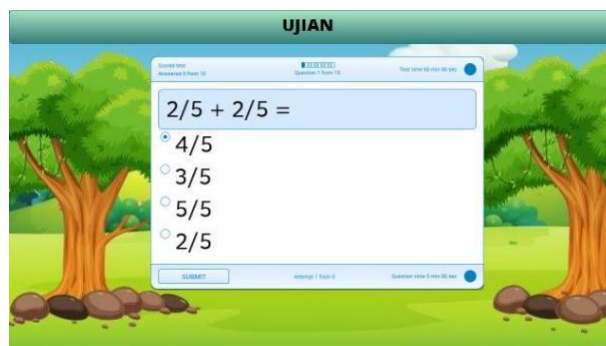


Figure 7 Examination

Figure 7 exhibits an examination page used to reassess the students' understanding. The ten questions on that page should be answered in 20 minutes. If students chose to take



the examination, they would see results beginning with the first topic, be given time to study, and then be taken to the examination immediately after. They were given time to work on practice questions where answers could be identified, beginning with either the first three topics, the fourth, or the seventh. The browser automatically redirected to this examination page when the allotted time expired. If there were more than 20 steps (i.e., if students had to answer the questions more than 20 times), they were directed to the examination page. Moreover, if students obtained a perfect score of 100 on the practice questions, they would not bother studying that material before the examination because they had mastered it.

### 3.8. Score



Figure 8 Score

Figure 8 displays two scores: the exercise and the examination results. The exercise score was acquired after a student completed the first question on the first page before learning. In contrast, the examination score was the student's final score after learning various topics. This score was determined by the student's performance on the examination.

### 3.9. Testing

This stage displayed the testing results for the scenarios listed in Table 3.1.

Table 2 Testing

No	Test Scenario	Result
1	Testing the learning application algorithm	Succeed
2	Answering practice questions	Succeed
3	Start button	Succeed
4	Continue button	Succeed
5	Answering practice questions	Succeed
6	Continue button	Succeed
7	Completing the examination	Succeed
8	Continue button	Succeed

### 3.10. Discussion

This section discussed the decision-making procedure when the developed formula was applied to the MDP algorithm. A snapshot of the debate was presented. The reward was calculated using the following formula, considering the student's degree of comprehension.

$$V^{\pi^*}(s) = EIR_{\max} * (1 - \frac{step}{20}) + \max_{a \in A_Y} \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi^*}(s')$$

Figure 9 Discussion formula

Students who performed well on practice questions with a score below 40 received a maximum reward of 10 points, those who performed well with a score between 40 and 80 acquired a maximum reward of 8 points, and those who did well with a score above 80 obtained a maximum reward of 5 points.

$1 - \frac{step}{20}$  merupakan perhitungan pengaruh dari step atau keadaan saat siswa siswi dalam pembelajaran di aplikasi.

$\max_{a \in A_Y}$  menurut (Whiteley, 2008) angka terbaik dalam rumus ini adalah 0,65.

$P_{s\pi(s)}$  merupakan angka setiap pembobotan soal berdasarkan keadaan siswa siswi pada saat pembelajaran.

## 4. Conclusion

Specifically, MDP with module features could be employed to develop an ITS-based learning application. Learning consisted of reading the materials and answering the questions. During question completion, time could be limited. The problem-solving process was restricted to a maximum of 20 stages. MDP was characterized by how the subsequent step was determined depending on the answers, the number of steps, the total score, and the answer score. The learning order varied from student to student based on their abilities.

## References

- [1] Bloom, B. S. (1968). Learning for Mastery. University of California.
- [2] Dewi, Y. F. (2020, August 18). "Meaningful Learning From Home through the DaSi AsSi Learning Model". (In Indonesian) Retrieved from LPMP JAMBI: <https://lpmpjambi.id>
- [3] Howard, R. (1960). Markov Processes. In Dynamic Programming and Markov Processes (p.3). Cambridge: The Technology Press of The Massachusetts Institute of Technology.
- [4] SHUTE, V. J., & JOSEPH, P. (1994). Intelligent tutoring systems: Past, present, and future(Interim Report, Apr. 1993- Apr. 1994).
- [5] Shen, S., & Mostafavi, B. (2018). Exploring Induced Pedagogical Strategies Through a Markov Decision Process Framework: Lessons Learned. Journal of Educational Data Mining, Volume 10, No 3.
- [6] Rowe, J. (2013). Narrative-Centered Tutorial Planning with Concurrent Markov Decision Processes.
- [7] Vanlehn, K. (2006). International Journal of Artificial Intelligence in Education. 227-265.
- [8] Mott, B. W., & Lester, J. C. (2006). ITS'06: Proceedings of the 8th international conference on Intelligent Tutoring Systems. Taipei City Government: Springer-Verlag.
- [9] Purbohadi, D., Nugroho, L., Santosa, I., & Kumara, A. (2014). Design of Intelligent Tutoring System for Mastery Learning. Yogyakarta: American Scientific Publishers.
- [10] King, M. Y., & D'Inverno, M. (2016). Stimulating collaborative activity in online social learning.