

---

# GREEDY REDUCTION EDUCATION GAME BASED ON ANDROID PLATFORM

**Bagus Satrio Waluyo Poetro, Nurul Hidayah, Suryani Alifah**

Informatics Engineering, Industrial Technology, Sultan Agung Islamic University

St. Raya Kaligawe KM.4, Central Java, Indonesia

## Abstract

*The greedy algorithm is one of the simplest algorithms to solve the optimization problem. The greedy algorithm is able to solve the problem quickly. Greedy Reduction game is a game of mathematical reduction that uses the theory of greedy algorithm in the process of determining the player's victory in playing the game. Players who play this game have to answer the problem by solving the reduction operation with subtracting the number by selecting available numbers on the game, until it reaches the specified number of questions. Players will either win the game or produce a serial number if the numbers the player chooses to subtract fewer numbers from the numbers used by greedy. Tests conducted on this research is to use black box testing method. Based on the tests performed, greedy algorithm can be applied in solving the existing problems in Greedy Reduction game.*

**Keywords:** greedy, reduction, optimization

## 1. Introduction

The world of education requires children to dominate basic knowledge and skills such as reading, writing, and also counting for mathematical lessons. However, in a very important basic lesson, the child is experiencing learning difficulties. Children's learning difficulties are mostly found in mathematics lessons. Educational game is one of alternative that can help children in learning until children can learn while playing and they do not get bored [1]. Greedy algorithm is one algorithm that has a simple and fast process in solving a problem. The average greedy algorithm takes time in the process of problem solving in the program is less than 1 millisecond [2].

Previous research says there are some researchers who use greedy algorithms in problem solving on games. Among them is a study entitled Greedy Algorithm App and Dynamic Programming on Greedy Spider Game. This game requires player to save the insects trapped in the spider net where the insect will be eaten by the spider if not unfolded. The greedy algorithm in this game is functional to know the movement of spider [3]. Then, there is also a study titled Analysis of the Use of Greedy Algorithm on Chess Game. In this game, the greedy algorithm is used for an artificial intelligent (AI) or computer-generated intelligence by setting priorities for checking 6 chess pieces [4].

Research conducted by the author of The Greedy Algorithm in the Greedy Reduction Education Game Based on Android Platform is in addition to applying greedy algorithm that is used for comparison of answers that will determine the victory or defeat of players in playing the game, where the algorithm itself is an algorithm that has a fast process and an algorithm which is simple but still has a weakness that does not always produce the optimal solution in solving the problem, but in the author's research, the weakness of greedy algorithm is not harmful at all because it can be used for players to win against greedy because the choice of players can be more optimal.

## 2. Research Method

The following is the research method used in this research:

### a. Method of collecting data

Data collection methods used are secondary data collection and primary data collection. Primary data is data which taken by direct researchers from research sources (this study used interview method). Primary data is data obtained by researchers indirectly from outside the environment from research sources. Secondary data can be obtained from reliable journals, books, literature or other references, such as literature studies [5].

---

### b. System Development Process Model

Model of system development process used in this research is Modified Waterfall model. The Modified Waterfall model in the figure illustrates "reverse flow" pathways representing the revisions and corrections required in software development that can be caused by changing requirements or errors in software [6].

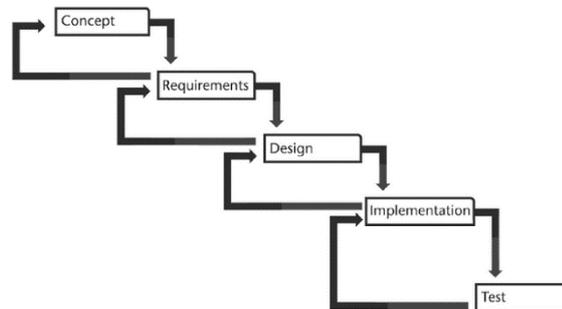


Figure 1 Modified Waterfall Model

Based on Modified Waterfall Model, the stages of research are as follows:

1. Concept, is the stage determines the concept of game or game like what will be made.
2. Requirement, is the process of doing analysis and collecting of various materials needed in the manufacture of games, for example data knowledge such as books, tutorials and so forth. The method of analysis used in this case is OOA (Object Oriented Analysis) that is analyzing things that the system needs by looking at the point of view of objects and classes around them [7].
3. Design, which is to design the game system. In this process, both the game architecture, the interface representation, and the details of the algorithm to be used (the greedy algorithm), are calculated and determined in this stage. Then, for the design method used in this research is Object Oriented Design (OOD).
4. Implementation, carried out the process of realizing the design of both algorithms, architecture and interface into the game through coding at this stage. The application used for the coding process is Android Studio in the making of Greedy Reduction game. Android Studio uses Java programming language as the base language. The process of applying the greedy algorithm into the game is done at this stage.
5. Testing, testing on the game is conducted by using black box testing method, that is external testing conducted without having to know how the system runs, for example to test the suitability of the function, the suitability of the interface and so forth.

## 3. Results and Analysis

### 3.1. Literature Review

#### a. Understanding the Game

Game comes from English. Playing a game for players requires the ability to take power, analysis and agility to solve the problems or mysteries contained in the game. There are activities, regulations also require results or achievements in a game where the game itself is actually a complex activity. Then the game can also mean a trick that has rules in playing it and has a plot or scenario that directs players to follow the scenario to play the game [8].

#### b. Understanding Education Games

Educational games or edugame are games that are more specified for educational purposes. Educational game makers must take into account this game can really educate and also increase the knowledge for the players [9].

The following are features of the educational game [8]:

1. Can be played in various purposes and benefits or can be used in various ways.

2. Targeted primarily to children with a view to developing motor and intelligence aspects of the child.
3. Actively involve the child
4. Be constructive
5. It is very concerned about the security of both forms and the use of which should not offend tribe, religion, race and intergroup, pornography or violence.

**c. Greedy Algorithm**

Greedy algorithm is one algorithm that can solve problems related to optimization, such as how to determine a solution to minimize and maximize. The principle of greedy itself is reflected in this algorithm in forming a solution by taking a step by step decision. In every step taken, there are choices that need to be well researched, in order to produce the best steps for each step taken. The greedy algorithm has the principle of, "Take what you can get now!" Following the explanation [10]:

- a. First, determine the options for the optimal local part.
- b. By taking the optimal local choice, it is then expected to produce optimal solutions globally.

Local optimum is the best choice according to greedy, which can be taken at that time. While global optimum is the best solution obtained to solve the problem as a whole [11].

The elements of the greedy algorithm are as follows [12]:

- 1) The set of candidates, is the elements that make up the solution.
- 2) The set of solutions, is the set derived from the set of candidates that have been selected as the solution, to then be used to solve the problem.
- 3) Selection function, is determining the candidate closest to the best solution or the most likely candidate to solve the problem.
- 4) Feasibility Function (feasible), is the process of checking the candidate has been selected whether to produce a viable solution.
- 5) Objective function, that is minimize or maximize value of solution which have been taken.

There are two things that cause greedy algorithm cannot produce an optimal solution [10]:

1. The greedy algorithm does not check all the overall solution alternatives.
2. There are several alternative selection functions. The greedy algorithm can produce a fairly optimal solution, then the accuracy in selecting the selection function is certainly required.

There is an example of a problem in a previous study that applies greedy algorithms to addition games, where there is a number 17 that is raised as a problem, then there are four number choices that can be used to add up the numbers, namely 5, 4, 3 and 1. Then [13]:

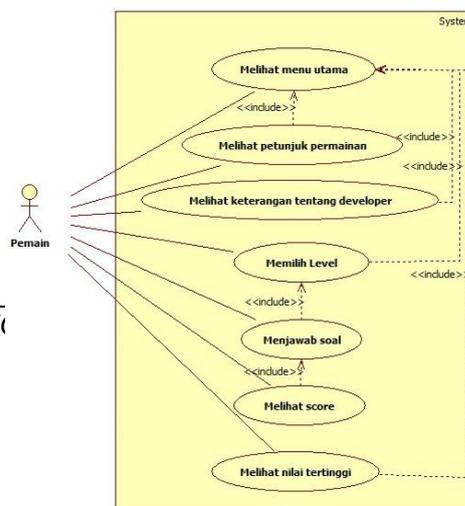
- The solution offered by the greedy algorithm is 5 + 5 + 5 + 1+ 1 (5 numbers)
- A more optimal solution is: 3 + 4 + 5 + 5 (4 numbers)

From the results above shows that the greedy algorithm does not produce optimal choices because the greedy algorithm takes the largest number or the closest to the question number before taking another number.

**3.2. Designing**

The design conducted in the making of the game is to apply the concept of OOP (Object Oriented Programming), among which there are use cases and class diagrams as diagrams that present the

game design.



Greedy Algorithm in 'i

ed on Android Platform  
s Satrio Waluyo Poetro)

*Figure 1 Use Case Diagram*

Based on the use case diagram in Figure 2, the following is the exposure of each use case function:

a. View the Main Menu

Use case Viewing Main menu to show that players can see the Main Menu in the game, Main Menu consists of several options that can be selected players in running the game.

b. Viewing the Game Instructions

This use case presents that players can see game hints to see how to play the game Greedy Reduction.

c. View Description about Developer

Use case indicates that the player can see information about system developers or developers that appear on a page, such as short bio data and so forth.

d. Selecting Category

This use case presents that players can see categories of games including easy, normal and difficult categories. Each category has a different level of difficulty so players can choose a category according to their own ability or desire.

e. Answering the Question

This use case suggests that players can certainly play the game by solving the problems that appear in the game by selecting the appropriate numbers to subtract the specific numbers that appear as a matter. The numbers selected players must be as little as possible to beat greedy.

f. Viewing Value

This use case displays the value that can be in the player when finished playing the game. This use case presents that players can see the scores or scores earned that appear after the player completes the game.

g. Viewing Top Rated

This use case presents that players can see the highest score that players get at each level.

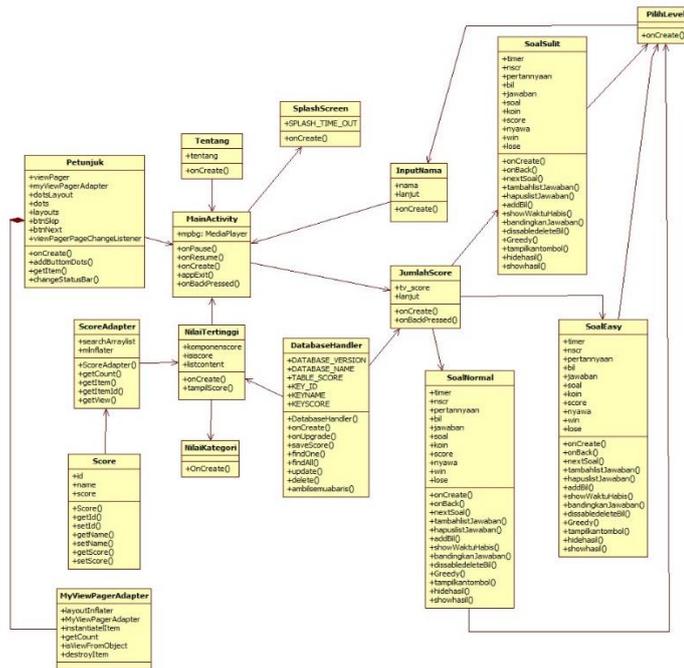


Figure 2 Use Case Greedy Reduction Game

Based on the Figure Class Diagram, it can be seen that the Greedy Reduction game has interconnected classes including SplashScreen class, MainActivity class, InputName class, Hint class, Class Categories class, HighText class, DatabaseHandler class SoalEasy class, SoalNormal class, Troubleshooting class , Class selectLevel and class Score, ScoreAdapter, and the NumberScore class.

**3.3. Output Result**

Software used in the development/manufacture of Greedy Reduction game is Android Studio 3.0 Canary 7. Android Studio uses the Java language as a programming language in the development/manufacture of an application and/or game.

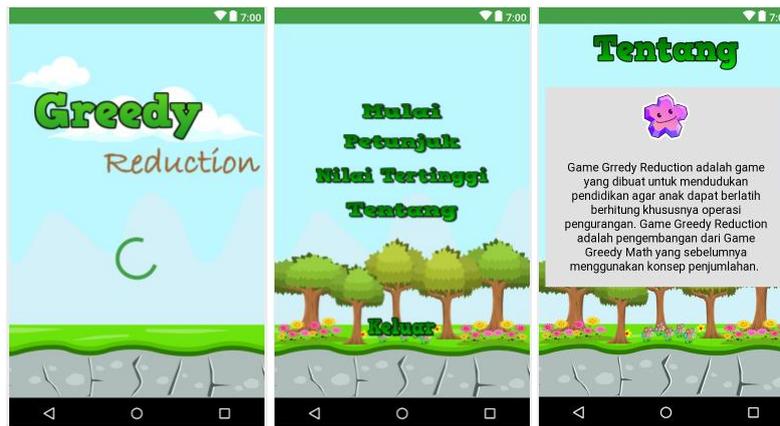


Figure 3 Greedy Reduction Game Display (part. 1)

Figure 4 (a) is a splash screen view of the Greedy Reduction game that is display on the first run. Figure 4 (b) is the main menu display and Figure 4 (c) is the page view of the developer.

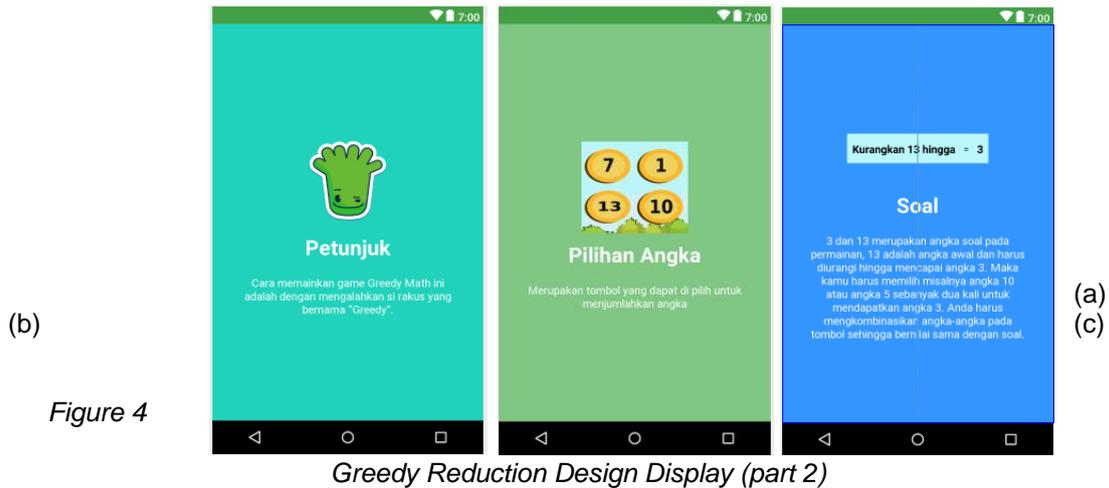


Figure 5 are 1, 2 and 3 slides show of greedy Reduction game hints as the player presses the “Petunjuk” button.

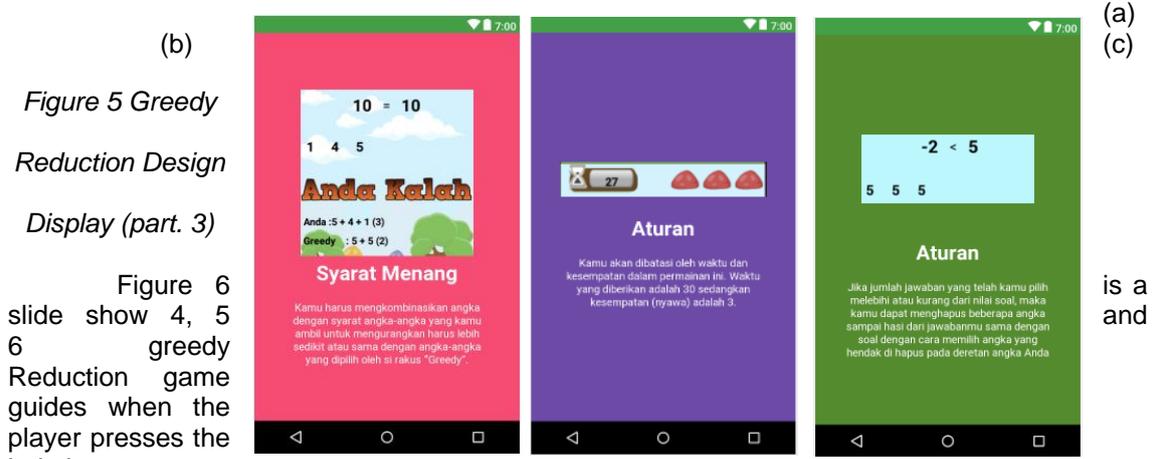


Figure 6 slide show 4, 5 6 greedy Reduction game guides when the player presses the help button.

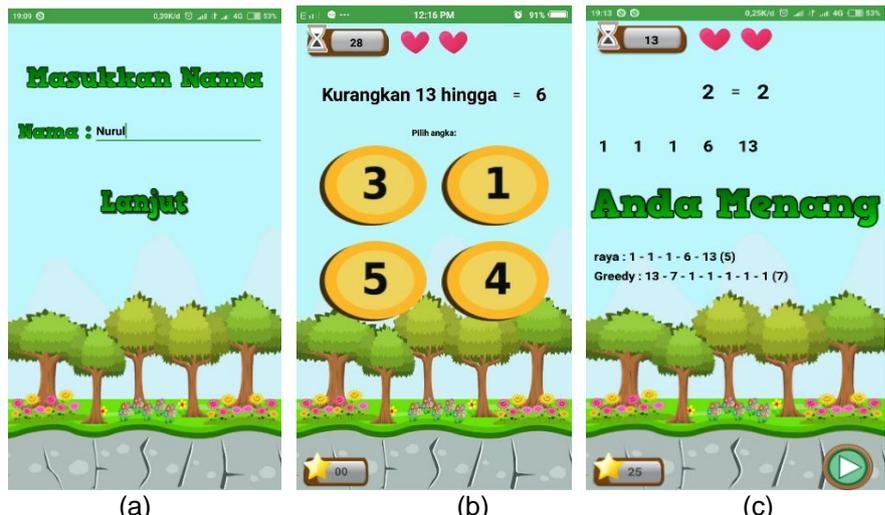


Figure 7 (a) is the display of the player input page name, Figure 7 (b) represents the page showing the question of the game page of Greedy Reduction game. 7 (c) is the game page showing the winning notification on the screen.



Figure 8 (a) is shows a lose notification on the screen. Figure 8 (b) is the page that displays the final score when the player has solved the problem at a certain level.

Greedy algorithm encoding in the Greedy Reduction game that uses Android Studio is shown below:

```
jumlah_pilihan = bilangan[0]+bilangan[1]+bilangan[2]+bilangan[3];
tv_jawaban.setText("Kurangkan " + String.valueOf(jumlah_pilihan) + " hingga");

btn1 = (Button) findViewById(R.id.btn1);
btn1.setOnClickListener((v) -> {
    //suara button klik
    mp.start();
    //bil = 1 krn bil pada tombol adalah 1
    bil = 1;
    jumlah_pilihan = jumlah_pilihan - bilangan[0];
    addBil();
});
```

Figure 9 Reduction section on the Greedy Reduction Game

Based on Figure 9, the variable  $bil = 1$  represents the number that will appear in the problem is the number 1 (one). Then there is also a state in which the variable  $jumlah\_pilihan = jumlah\_pilihan - bilangan$ . This variable  $jumlah\_pilihan$  represents the number of all available numbers on the button which will later become the number of the questions in the game. So, when the player chooses a button, the variable value of the  $jumlah\_pilihan$  will be reduced by the number chosen by the player, in this case the number is the number 1 which is filled into the number (0).

```

ArrayList<Integer> list = new ArrayList<>();
jumlah_pilihan = bilangan[0]+bilangan[1]+bilangan[2]+bilangan[3];

for(int a = 3; a >= 0; a--){
    while (jumlah_pilihan-soal >= bilangan[a] ) {
        jumlah_pilihan = jumlah_pilihan - bilangan[a];
        koin = koin + 1;
        list.add(bilangan[a]);
    }
}

Log.i( tag: "Koin", String.valueOf(koin));
Log.i( tag: "soal", String.valueOf(soal));
Log.i( tag: "bilangan[3]", String.valueOf(bilangan[3]));

//ArrayList<Integer> a = new ArrayList<Integer>();
int g =0;
for (Integer i : list) {
    g++;
    if(g == list.size() ){
        tv_jumlah_bilangan1.append(Integer.toString(i) + " (" +String.valueOf(koin)+")");
    }else{
        tv_jumlah_bilangan1.append(Integer.toString(i) + " - ");
    }
}

```

Figure 10 Greedy Algorithm Encoding in Greedy Reduction Game

Descriptions of variables include:

- The “a” variable, as index variables to populate ArrayList which will be accommodated on “bilangan []” variable
- The “bilangan []” variable, as container variable of numbers provided by system in question page.
- The “jumlah\_pilihan” variable is a variable that will accommodate and add up the number of options whilst the player chooses.
- The “koin” variable, is a variable to calculate how many options a number selected players to add numbers to a question.
- “Soal” variable, is a variable that serves as a container of question numbers.
- list.add (bilangan [a]) is an object that is created in order to access the variable bilangan [a] in ArrayList, which will re-display the player’s answer.

In Figure 10, first, an object named “list” is created to be used as an object that will access “ArrayList”. After that, there is a process of addition of the selected player number selection to be accommodated in “jumlah\_pilihan” variable. Then, there is a “for” loop function, starting with “a” variable value a which is described as integer type and has a value equal to 3. The “a” variable serves as an index to populate the “ArrayList” which is then stored in the “bilangan[]” variable. When “for” looping executable, as long as the value of “a” is still greater or equal to 0 then the program will execute the “while” command which is a loop, as long as the “jumlah\_pilihan” variable value minus the value of the “soal” variable is greater than or equal to the value contained in “bilangan [a]” array, then the value of “jumlah\_pilihan” will be reduced by the value in the bilangan [a]. Then, the value of the “koin” variable then increases one digit. The “koin” variable is a variable to calculate the number of options a player chooses to reduce the number in the question. “list.add (bilangan [a])” is for the “bilangan [a]” to be added to “ListArray” list. Each time the “for” function has finished executing, the value of “a” variable will decrease by one number.

### 3.4. Testing

Here is a test of greedy algorithm with black box testing in game Greedy Reduction:

#### 1. Algorithm Testing Scenarios

Table 1 Scenarios of Algorithm Testing

No	Functionality tested	Test Case	Expected Results
1	Selection of the	- Player Reply: Input is a	Series Notification, Win or

	number keys in the Problem Page.	number option - Greedy's answer (Comparison of player answers with greedy Algorithm answers)	Lose Score will increment by 10 numbers if win or series, and will only increase 5 numbers if lost.
--	----------------------------------	--	--

## 2. Algorithm Test Cases

Table 2 Algorithm Test Case

No	Functionality tested	Test Case	Expected Results	Actual Result
1	Selection of the number keys on the Problem Page. Arrays {13, 7, 6, 1}. Question = 2.	- Player Reply: Input number selection: 1 + 1 + 1 + 6 + 13 (5 digits) - Greedy's answer: 13 + 7 + 1 + 1 + 1 + 1 + 1 (7 digits)	Win notification, score increases 10 points.	Displays Winning Notification and a score of 10 points
2	Selection of the number keys on the Problem Page. Array {1, 3, 4, 5} Question = 5	- Player Reply: Input number selection: 3 + 5 (2 digits) - Greedy's answer: 5 + 3 (2 digits)	Series Notification, score increases 10 points only.	Showing Series Notifications and the score increases by 10 points.
3	Selection of the number keys on the Problem Page Array {1, 5, 7, 10} Question = 3	- Player Reply: Input number selection: 1 + 1 + 1 + 7 (4 digits) - Greedy's answer: 10 + 10 (2 digits)	Win notification, score increases 10 points.	Displays Winning Notification and a score of 10 points.

## 4. Conclusion

The conclusion gained from the above discussions is that Greedy algorithm or greedy algorithm is very suitable to be applied in this Greedy Reduction game, because although it has weaknesses that make greedy algorithm does not always show optimal result, this weakness can actually be advantage for game player to win in game. The greedy algorithm also has a simple and fast process in solving optimization problems.

Suggestions the author wants to convey is the need to improve the development of games that provide positive things, especially educational games that are good for children, which can improve the academic ability of children and do not feel boredom in learning, especially math-related subjects.

## References

- [1] F. Yeni, Y. HAsan, and Tarmansyah, "E-JUPEKhu Efektifitas Game Edukasi Untuk Meningkatkan Kemampuan Penjumlahan Bagi Anak Kesulitan Belajar," vol. 2, no. September, pp. 501–513, 2013.
- [2] O. Ekwardo, "Studi dan Implementasi Algoritma Optimasi Pemotongan Bar Steel," Bandung, 2008.
- [3] I. Zulhidayati and Y. Kartika, "Aplikasi Algoritma Greedy dan Program Dinamis (Dynamic Programming) pada Permainan Greedy Spiders," no. 1, pp. 1–14.
- [4] Ginonggom, Marlindawati, and A. Muzakir, "Analisis Penggunaan Algoritma Greedy pada Permainan Catur," pp. 1–14, 2013.
- [5] A. P. Utomo, "Analisa dan Perancangan Sistem Informasi Parkir di Universitas Muria Kudus," *SIMETRIS*, vol. 3, no. 1, pp. 17–24, 2013.
- [6] D. A. Vogel, *Medical Device Software Verification, Validation and Compliance*. Boston: Artech House, 2011.
- [7] A. R. Riswaya, "Sistem Penjualan Tunai dan Kredit Property," vol. 7, no. 2, pp. 106–116, 2013.

- 
- [8] E. J. Setiawan, *Pembuatan Aplikasi Game 3 Dimensi "THE TIMI."* Yogyakarta, 2014.
- [9] L. Fitriani, "Perancangan dan Pembuatan Game Edukasi Tajwid Mania Berbasis Android," Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM, Yogyakarta, 2014.
- [10] A. Juniar, "Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi," vol. 16, no. 2, pp. 175–184, 2015.
- [11] Ardiansyah, F. S. Efendi, M. Pinto, Pujiyanto, and H. S. Temapke, "Implementasi Algoritma Greedy untuk Melakukan Graph Coloring: Studi Kasus Peta," vol. 4, no. 2, pp. 440–448, 2010.
- [12] A. Nurgroho, *Algoritma dan Struktur Data dengan C#*. Yogyakarta: Andi Publisher, 2008.
- [13] N. Hidayah, S. Alifah, and B. S. W. Poetro, "Implementasi Algoritma greedy pada game edukasi mobile 'greedy math' berbasis android," Universitas Islam Sultan Agung, 2017.