The Tuition Payment Queuing System Uses Android-Based First in First Out (FIFO) Algorithm

Marlince NK Nababan¹, Windania Purba¹, Evta Indra¹
¹Faculty of Technology and Computer Science, Universitas Prima Indonesia, Indonesia
²Department of Information System, Faculty of Technology and Computer Sciences,
* marlince@unprimdn.ac.id

Abstract. Queuing is an important component of a process in service. Queues are formed in a company or institution due to customers. Disorderly queues are a big problem for all concerned. Where, disorderly queues occur at University Prima Indonesia at the time of tuition payment so that the campus has to carry out a crowded queue process and students have to stand waiting for the next call resulting in less effective and efficient even waste of energy for students and employees. This research designed a system of tuition payment queues using the First In First Out (FIFO) algorithm which aims to reduce the time students are queuing at the checkout at the time of tuition payment so as not to wait too long so that the time of students and campus employees is not wasted too much and reduce the crowded queue time. The result of this study is an android-based tuition payment system, which is useful for reducing student build up when paying tuition. Thus it can be said that the calculation result of P0 queue has a result of 0.25%, and the number of customers 3 people, the average customer in the queue has a value of 2.25, the total customer time is 20 minutes, and the average customer time in the row is 15 minutes, the server busy time is 0.75, and the server's empty time is 0.25%.

1. Introduction
The advancement of technology is growing, making many creative and innovative people more advanced to facilitate all things field especially in terms of work. Application creation of exam scheduling system using a genetic algorithm [1]. The tuition payment monitoring system has been designed only by knowing the movement according to the number of students who can help employee performance, and previous research designed the Android-based Billing Application [2-3]. Tuition payments are student obligations made by each semester in an educational institution. Currently at University Prima Indonesia in making tuition payments still by visiting the financial space and waiting to be called by the finance department when some students who are working so that students pay tuition is not on time. Analysis Product Inventory Control by EOQ Method Using Genetic Algorithms for Efficiency of Inventory Costs [4].

Researchers will design a technological innovation of an Android-based tuition payment queue system with the concept of FIFO algorithms. The use of this application students can register in advance and see the payment time made by accessing anywhere without having to go to campus waiting to be called in the financial room. This system has a notification feature so that when the number is called, students get an alert in the form of a notification on android. The problem with this study is how to design a tuition payment queue system using FIFO algorithm.

The purpose of this research is to reduce the time students who are queuing in the financial department at the time of tuition payment so as not to wait too long so that the time students and campus employees do not waste too much and reduce the crowded queue time and how to create an application of the tuition payment queue system by implementing the Android-based FIFO algorithm.
2. Methodology

2.1. Queuing System
The advancement of queue systems has been used for many things in telecommunications, traffic engineering, computing engineering, modelling and system engineering [4-5], and some industrial businesses such as shops, offices, and hospitals. The researchers argue that the queuing system is one of the options used in business decision making to meet certain service systems. The theory of the queuing system began with Agner Krarup Erlang's research which created a telephone model of information exchange in Copenhagen.

2.2. FIFO Algorithms
The FIFO (First in first Out) transfer algorithm associates the time when a page is brought into memory with each page. At the time the page must be moved, the older page is selected. The data that is first written to the queue path is the first data to be read.

Unlike queue paths, FIFO is not a temporary object, it is an entity in the file system and can be created with the FIFO command. Free processes use FIFO as long as they have access to it.

The way FIFO is opened is slightly different from the queue path. A queue path (its two file data structures, its VFS in ode, and shared data page) is created at once while FIFO already exists and is opened and closed by its users. Additionally, FIFO is handled in almost the same way as queue lines and FIFO uses the same data structure and operations. Comparison of three queue mechanisms, Priority Queuing (PQ), Custom Queuing (CQ), Weighted Fair Queueing (WFQ), Class-Based Weighted Fair Queuing (CBWFQ), Low-Latency Queueing (LLQ) [5-6]. Mustafa's research made three comparisons: First in first Out, queue priority, and weighted queues. From the results of the study, any algorithm used against the CPU will occur router delay [7]. FIFO method principle: FIFO has a First In First Out system that says that "First components must be sent or removed first and reduce errors". This system will help track OEM records and a simple method [8-9-10].

The FIFO algorithm is the simplest. The principle of this algorithm is that the principle of queues (nonpriority queues), patients who come first will be invaded first well. A queue of linear data structures consisting of a set of elements. Both ends of the queue can be accessed to perform operations on elements commonly called heads and tails. Elements can be inserted at any time, but only the longest elements in the queue can be deleted.

Two principal operations can be performed in the queue; queue and dequeue. Elements are inserted from the back queue and dequeue. This algorithm uses a data stack structure. If there is no blank frame when a page fault occurs, then the selected victim is the frame that is in the bottom stack, which is the page that is the last in memory as in figure 1.

![Figure 1. Algorithm FIFO](image)

FIFO (First in first Out) is a non-preemptive, non-priority scheduling algorithm. Each process is given an execution schedule based on the order of time of arrival. As soon as the process gets the execution allotment then the process will be executed until it is completed, [13-14-15].

2.3. Android
The device has only one foreground screen. Normally when the User Interface (UI) will accumulate above the previous screen (home). When you look at the help, then help will overwrite the previous UI, so on. All of the above processes are recorded in the application stack by the system activity manager. Each user interface is represented by the Activity manager class. Each Activity has a cycle as shown below.

Activity cycles during this cycle run, they can have more than 2 statuses. We can't control every status because everything is handled by the system. However, we will get a message when a status change occurs through the on change method.

The built system certainly requires a flow diagram of the entire system, for the input mechanism, process, and results needed by the purpose of the research. The first stage of the process is the preparation of data, which is the process of retrieving data from students. Data taken from students can be the name, parent number of the student, semester, major, and so on. The stage will involve admins acting as media to activate the queue number.
Furthermore, the process of creating and sending queue numbers through the android app to each student. The process is needed so that when the student makes the payment can be processed by the admin because the student data is already connected to the database. The mechanism for carrying out the process requires student data, which is the data that has been compiled so that the accuracy of the resulting system is increased.

The next stage is the implementation stage where the system that has been built will be tested directly. The queue process will be done by the way students log into the system and open the queue booking menu and then choose the administration to be paid, then the system will create a queue schedule in the form of the queue number and the date and time of the queue.

The next process is that the system will insert the event data into the database. In general, the process of the android-based tuition payment system is seen in Figure 2.

![Flow Chart of Tuition Payment Queue System](image)

**Figure 2. Flow Chart of Tuition Payment Queue System**

### 2.4 Use Case Diagram

After the design of the flow chart of the system then next is the design of the use case diagram. Where the system will involve two actors namely admin and student. Admins are tasked with checking queue tokens from students, as well as processing payment transactions. Students have six use cases, namely to enter the application, must go through the login first, the next process, the system will display the Home menu, in which there are four use cases namely Booking Queue, Queue History, Payment Info and Application Usage Guide. The Use case of the built-in system can be seen in Figure 3.

![Use Case Diagram System](image)

**Figure 3. Use Case Diagram System**
3. Results and Discussions

3.1. First in first out (FIFO) Method Analysis

In the case of completion with the First in first Out method to be done is to determine the payment queue of tuition. First in first Out to be done is to determine the tuition payment queue. The data objects as many as 20 data that in this case represent the person who will pay tuition for more details, the data is based on research. The following data table of tuition payment queue time can be seen in table 1 below:

| No | Entry Time | Old Student | Time out | Waiting Time | long time in system | Server |
|----|------------|-------------|----------|--------------|---------------------|--------|
| 1  | 08:02      | 15          | 08:17    | 0            | 15                  | 1      |
| 2  | 08:05      | 20          | 08:25    | 0            | 20                  | 2      |
| 3  | 08:07      | 20          | 08:37    | 10           | 30                  | 1      |
| 4  | 08:10      | 25          | 08:50    | 15           | 40                  | 2      |
| 5  | 08:11      | 15          | 08:52    | 26           | 41                  | 1      |

Suppose, 1 enters with the type of event in, the student in queue 1, the student in system 1 with the busy queue status and server time 0.

Table 2 above can be concluded that the queue to pay tuition from 08.02 to 12.20 is a time during the queue of 5 hours 18 minutes with the number of students 20 people and each person time is 15 minutes, 20 minutes, 26 minutes, and 30 minutes.

Based on the table of time of the tuition payment queue above, analysis can be done in the queue analysis table. Here's table 2 below:

| No | Student Time | student | Event Type | Students In line | Students In The System | Empty Time |
|----|--------------|---------|------------|------------------|------------------------|------------|
| 1  | 08:00        | 0       | 0          | 0                | 0                      | 02         |
| 2  | 08:02        | 1       | In         | 0                | 0                      | 03         |
| 3  | 08:05        | 2       | In         | 1                | 2                      | 0          |
| 4  | 08:07        | 3       | In         | 2                | 3                      | -          |
| 5  | 08:10        | 4       | In         | 3                | 4                      | 0          |
| 6  | 08:11        | 5       | In         | 4                | 5                      | -          |
| 7  | 08:12        | 6       | In         | 5                | 6                      | 0          |
| 8  | 08:13        | 7       | In         | 6                | 7                      | -          |
| 9  | 08:14        | 8       | In         | 7                | 8                      | 0          |
| 10 | 08:15        | 9       | In         | 8                | 9                      | -          |
| 11 | 08:16        | 10      | In         | 9                | 10                     | 0          |
| 12 | 08:17        | 1       | Out        | 8                | 9                      | 0          |

The analysis conducted by the author on the activities of the tuition payment queue at UNPRI is by conducting an observation where there are 20 customers present to pay tuition with different arrival hours with a period of 5 hours 18 minutes, each has 15 to 30 minutes per person. Specify: P0, L, Lq, W, Wq, Wp, and Wk. Then Probability of absence of queues \( 0 = 12\pi = 1 - 1520 = 1 - 0.75 = 0.25 = 0.25\% \), Number of customers in the queue system \( - = 1520 - 15 = 3 \) students, Average customer in queue line \( Lq = \lambda 2 \pi (\pi - \lambda) = 15220(5) = 225100 = 2.25 \), total customer time in queue system \( W1 \pi - \lambda = 120 - 15 = 0.2*100 = 20 \)
minutes, Average customer time in queue row \( W_q = \lambda \pi (\pi - \lambda) = 1520(20 - 15) = 0.15 \times 100 = 15 \) minutes.

Thus it can be said that the calculation result of P0 queue has a result of 0.25\%, and the number of customers 3 people, the average customer in the queue has a value of 2.25, the total customer time is 20 minutes, and the average customer time in the row is 15 minutes, the server busy time is 0.75, and the server's empty time is 0.25\%.

4. Conclusion

Implementation of the tuition payment queue system is done using the Android-based FIFO algorithm. On Android, use the Android Studio IDE for app development, while algorithm FIFO is used as a tuition payment queue trouble shooter.

References

[1] Murat DENER, M. Hanefi CALP, 2018, Solving The Exam Scheduling Problems In Central Exams With Genetic Algorithms, Mugla Journal of Science and Technology, p 103

[2] Asep Saefullah, 2015, Perancangan Sistem Informasi Monitoring Antrian Pembayaran Kuliah Pada Lkm Perguruan Tinggi Raharja, Vol.9 No.1, ISSN: 1978 -8282.

[3] C. E. Chinwuko, E. C. Daniel, O. P. Ugochukwu and O. O. J., "Analysis of a queueing system in an organization (a case study of First Bank PLC, Nigeria)," American Journal of Engineering Research, vol. 03, no. 02, pp. 63-72, 2014.

[4] Indroprasto, Erma Suryani. (2012). Analysis Product Inventory Control by EOQ Method Using Genetic Algorithms for Efficiency of Inventory Costs. Its Engineering Journal. A305-A309.

[5] Eissa A, Amgad Saeed Dabwan dan Anwar Ahmad, 2018, Valuation of Inventory Methods in the Industrial Companies in Yemen, International Journal for Scientific Research & Development, p 591

[6] S. Aminatunnisa et al., “Penerapan Metode Monte Carlo Untuk Simulasi Sistem Antrian Service Sepeda Motor Berbasis Web,” J. Sist. Inf. dan Ilmu Komput. Prima (JUSIKOM PRIMA), vol. 2, no. 2, pp. 77–84, 2019.

[7] S. K. Dhar and T. Rahman, "Case Study for Bank ATM Queueing Model," IOSR Journal of Mathematics, vol. 7, no. 1, pp. 01-05, 2013.

[8] Mustafa Elgili Mustafa, 2015, The Effect of Queuing Mechanisms First in First out (FIFO), Priority Queuing (PQ) and Weighted Fair Queuing (WFQ) on Network's Routers and Applications, (International Journal of Engineering Innovation & Research), p 188

[9] Manohar HM and S Appaida, 2017, Stabilization of FIFO system and Inventory Management, International Research Journal of Engineering and Technology (IRJET), p 5632

[10] Onoja Emmanuel and Yahya Uthman Abdullahi, 2015, Inventory Valuation Practices and Reporting: Nigerian Textile Industry Experience, Mediterranean Journal of Social Sciences, p 74

[11] Robert Houmes and Inga Chira, 2015, The valuation effect of LIFO’s repeal on high pricing power firms, emeraldinsight, p 306

[12] Christiano dkk,Sistem Operasi, Book, 2003.

[13] Arif Akbarul Huda, Pintar Pemograman, Book, 2016.

[14] Eka Budhy Prasetya, Pembuatan Aplikasi Car Storage Dengan Menggunakan, e-ISSN: 2550- 0678, ISSN: 1979-5564.
[15] Jemmy Edwin dan Wenselaus, 2012 "Model Antrian Fifo (First-In First-Out) Pada Pelayanan Mahasiswa Fakultas Teknik Universitas Janabadra Berbasis Multimedia", ISSN 2088 – 3676.