Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12003
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Development of Web-based Single Channel Multi Steps Online Queuing System with Model View Controller

Yuvi Darmayunata[1)*], Mariza Devega [2)], Yuhelmi [3)]
[1)2)3)] Universita Lancang Kuning, Indonesia
[1)] yuvidarmayunata@unilak.ac.id [2)]marizadevega@unilak.ac.id[3)] yuhelmi@unilak.ac.id

Submitted : Dec 14, 2022 | **Accepted** : Dec 28, 2022 | **Published** : Jan 4, 2023

Abstract: Abstract: Improvement in service quality will be directly proportional to customer satisfaction. One form of improving the quality of services provided by a Government Institution engaged in the Health sector, namely the Health Center is an effective and efficient queue. This research is a follow-up research where previously an analysis and design of an online queuing system has been carried out. This study aims to make it easier for patients to take queue numbers in advance through the queue website, and patients can come based on a predetermined estimated time. This of course can minimize patient waiting time and also prevent the accumulation of patients in the waiting room of the puskesmas. Single Channel Multi Steps is one of four types of queues and is the type of queue currently running at the Siak Hulu I Kampar-Riau Health Center, while the Model View Controller (MVC) was chosen because it is a feature that makes it easy to create a Web-based application.

Keywords: Queuing, Online Queuing, Single Channel Multi Steps, MVC, Web

INTRODUCTION

Improving service quality will be directly proportional to perceived customer satisfaction, both in a government agency or a Company business entity (Mukarrama et al., 2017) . In government agencies, especially in the field of health services, there are many aspects that can encourage an agency to be able to provide optimal service so as to meet customer satisfaction in interactions. One form of improving the quality of services provided by a Government Institution engaged in the Health sector, namely the Health Center is an effective and efficient queue.

Queueing analysis is frequently used for congestion analysis and resource planning for situations like scheduling patients in hospital clinics, bed allocation policies in hospital wards, and etc (Purwanda & Indarti, 2019). Basically a queue does not immediately solve the problem in the queue itself, but with queue analysis it can provide important information for management (Mital, 2018) in the form of service duration, number of patients per one unit of time, length of action and etc.

Puskesmas Siak Hulu I Kampar-Riau is one of the health centers in Kampar Regency which provides 24-hour services, specifically for Emergency Unit (ER) patients. Meanwhile, outpatient services start from Monday to Saturday with service hours: Monday-Thursday 07.30-13.00, Friday 07.30-10.30 and Saturday 08.30-12.00.

Based on the results of direct interviews that have been conducted in previous studies, the average service provided per day is 80 patients. Based on previous information, there are also queues that are considered ineffective, where there is a buildup of patients in the waiting room at the Puskesmas. This is in line with research (Burungale et al., 2018) which considers the Queue Management System to have an influence on the number of patients. The absence of waiting time predictions can result in wastage of patient time. This weakness can be overcome by using an effective Queue management system (Maulana et al., 2017). This research is a pilot project. An analysis of the running queuing system has been carried out, resulting in the information that the average patient arrival per hour is 11 patients, where in every 5.16 minutes there is one patient arrival. This means that in six working hours there are 70 patients (Zamzami et al., 2019).

To improve the quality of service and to avoid accumulation of patients in the waiting room of the Puskesmas and minimize patient waiting time, a Web-based Sigle Channel Multi-Steps Online Queuing System with Model View Controller (MVC) was created.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12003
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Single Channel Multi Steps is one of four types of queues. While the Model View Controller (MVC) was chosen because it is a feature that makes it easy to create a Web-based application (Lee & Wang, 2019). MVC itself is a software design pattern used to build web applications. The MVC framework has been widely used and is a benchmark in software development (Jailia et al., 2017).

Sigle Channael Multi Steps was chosen because it is a running queue at Puskesmas Siak Hulu I Kabupaten Kampar. The choice of MVC as a framework for code igniter is considered to be in accordance with a multi-step single channel queue because of the systematic and flexible workflow in web development, besides that MVC is a pattern that is often used in interactive software system architectures. Where the Model describes the records in the database, the View shows the results of the data contained in the model and the Controller connects and handles the processes in the Model and View.

Before implementing the system, the system design is carried out first. The resulting design consists of three stages, namely: conceptual modeling, database design and interface design (Devega et al., 2022).

Furthermore, to realize and continue the planned stages, a Web-based Sigle Channel Multi-Steps Online Queuing System with Model View Controller (MVC) was created.

METHOD

This research builds an online queuing system at Puskesmas Siak Hulu I-Kampar Regency. The types of data used are:

1. Primary Data, data taken directly from the source, obtained based on observations and calculations, in this case the patient's waiting time data, starting from the first arrival to completion.
2. Secondary Data, data obtained from journal references, books and documentation from related agencies.

The tools used are:

1. Programming Language: Hypertext Processor (PHP)
2. Databases: MySQL
3. Framework: Code Igniter

1. Single Channel Multi Steps

A queuing system consists of a set of clients, servers, and various rules. Queues also manage customer arrivals and services. According to (Joseph, 2020) queuing is a branch of applied mathematics that is used to predict the behavior of lines (also known as queues), such as line length over time, and the average time someone spends waiting in a queue. Single lane queuing system with several stages or existare two or more services running sequentially (Pramudhita, 2018).

The following is a picture of a single channel multi-step queue :

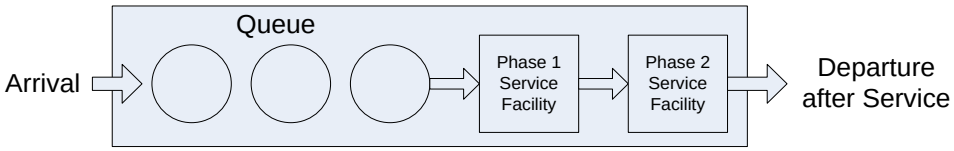

Fig.1 Single channel – multi steps queueing model (Pramudhita, 2018)

2. Model View Controller (MVC)

MVC has been recognized as an architectural model that has been widely used in creating various kinds of web applications (Dissanayake & Dias, G,K, 2017). In line with the reaserched of (Sunardi & Suharjito, 2019) states that the MVC pattern design is a pattern that is often used in interactive software system architectures. MVC is the underlying architectural pattern for most PHP frameworks (Adam & Andolo, 2019). MVC is a method to build a web by separating the model, control, and view to make it easier and faster (Mufid et al., 2019).

This following picture is MVC Architechture:

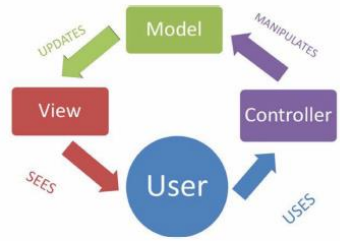

Fig.2 MVC Architecture (Jailia et al., 2017)

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12003
e-ISSN : 2541-2019
p-ISSN : 2541-044X

RESULT

Web development of an online queuing system carried out at the Puskesmas Siak Hulu I in Kampar Regency uses MVC as a framework.The following will show the script of each model, view and controller.

Here is the query for Model,view and controller section:

```php
public function getNoAntrian(){
    $noAntrian=1;
    $nowDate = date('Y-m-d');

    $this->db->limit('1');
    $this->db->where('tgl_antrian',$nowDate);
    $this->db->order_by('no_antrian','DESC');
    $antrian = $this->db->get('antrian')->row();

    if($antrian){
        $no= $antrian->no_antrian;

    }else{
        $no = 0;
    }

    $no = $no + 1;
    $this->db->set('no_antrian',$no);
    $this->db->set('tgl_antrian', $nowDate);
    $getRow = $this->db->insert('antrian');

    $hasil = array("no"=> $no);
    echo json_encode($hasil);
}
```

Fig.3 Model's Script

```php
<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>Puskesmas Siak Hulu I Kabupaten Kampar - Riau</title>

  <!-- Custom fonts for this theme -->
  <link href="<?php echo base_url('assets/user')?>/vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="
  text/css">
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">
  <link href="https://fonts.googleapis.com/css?family=Lato:400,700,400italic,700italic" rel="stylesheet" type="text/
  css">

  <!-- Theme CSS -->
  <link href="<?php echo base_url('assets/user')?>/css/freelancer.min.css" rel="stylesheet">
  <link href="<?php echo base_url('assets/user')?>/lib/noty.css" rel="stylesheet">
  <link href="<?php echo base_url('assets/user')?>/lib/themes/metroui.css" rel="stylesheet">

</head>
<style type="text/css">
  .btncostume{
    background: #2c3e50;
    color: white;
  }
  sup{
    color: red;
```

Fig.4 View scripts

```php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Antrian extends CI_Controller {


    function __construct(){
        parent::__construct();
        // $this->load->library('fpdf');
    }

    public function index()
    {
        $nowDate = date('Y-m-d H:i:s');

        $this->db->limit('1');
        $this->db->where('tgl_antrian',$nowDate);
        $this->db->order_by('no_antrian','DESC');
        $antrian = $this->db->get('antrian')->row();
        if($antrian){
            $data['no_antrian'] = $antrian->no_antrian;

        }else{
            $data['no_antrian'] = 0;
        }
        $this->load->view ('user/antrian',$data);
    }

    public function getNoAntrian(){
        $noAntrian=1;
        $nowDate = date('Y-m-d');

        $this->db->limit('1');
        $this->db->where('tgl_antrian',$nowDate);
        $this->db->order_by('no_antrian','DESC');
```

Fig.5 script Controller

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12003
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The following will show the results of the interfaces of online queuing system that has been built through screenshots from the web that can be accessed:

Fig.6 web homepage queue

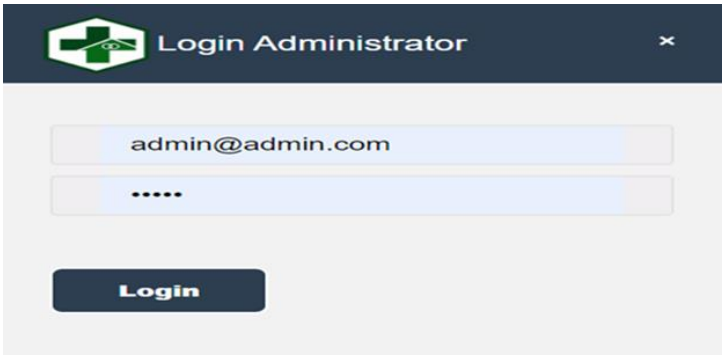

Fig.7 Administrator login form

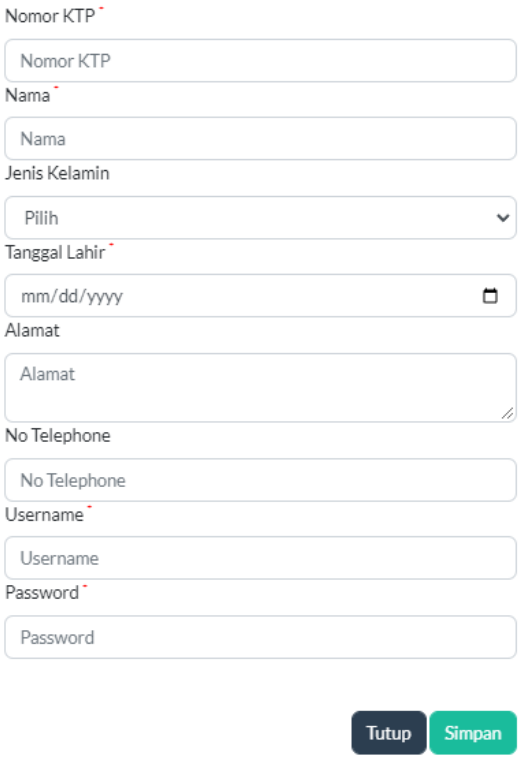

Fig.8 Registration Form

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12003
e-ISSN : 2541-2019
p-ISSN : 2541-044X

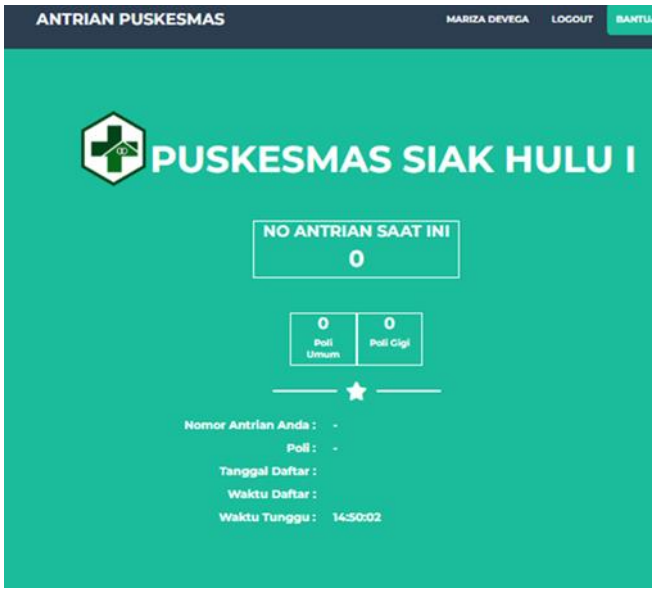

Fig.9 Queue main page

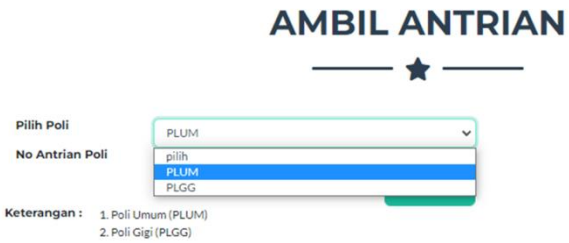

Figure 10. Retrieve Queue Numbers

Fig.11 Queue number

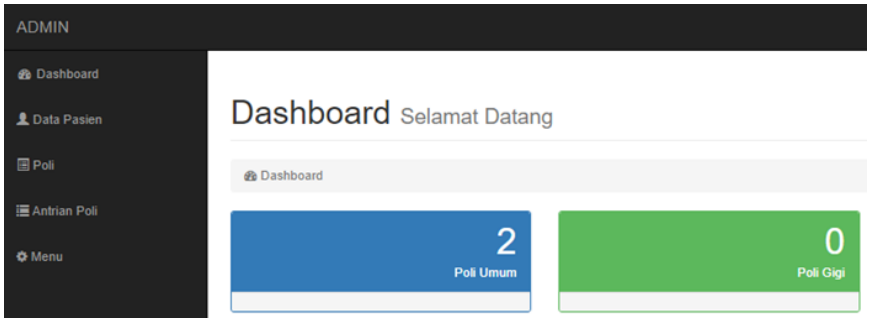

Fig.12 Admin Dashboard

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12003
e-ISSN : 2541-2019
p-ISSN : 2541-044X

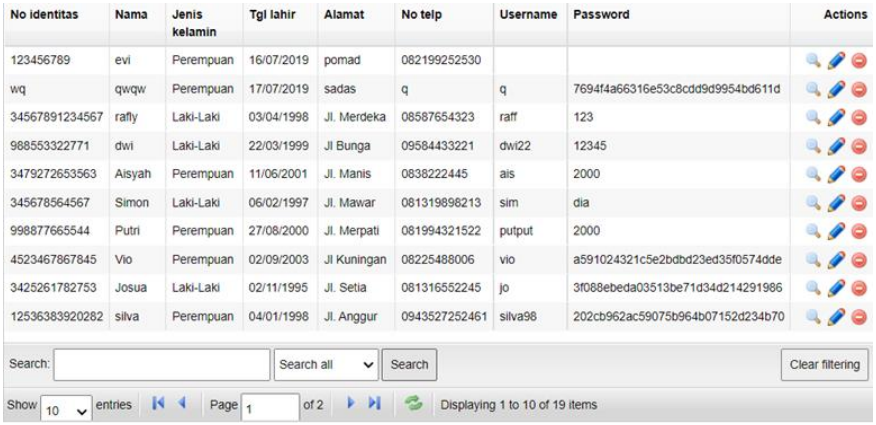

Fig.13 Patient's Data

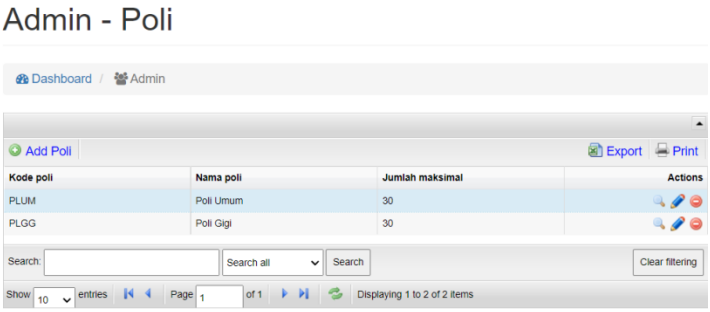

Fig. 14 Poly

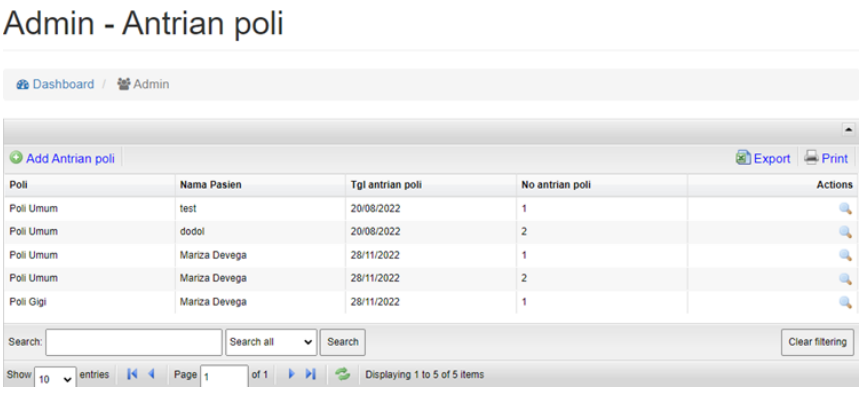

Fig.15 Poly Queue

Fig.16 Queue Info

*name of corresponding author

DISCUSSIONS

Based on the results in the previous sub-chapter, scripts from each view controller model are given, as well as screen shots and interface displays from the online queuing system that has been built, starting from the homepage, registration, taking the queue number to the admin dashboard, patient data and so on.

Figure 3 until figure 5 are the script for each model, view and controller. The model contains application data, logic definitions, function specifications, and business rule engagement. A Model in Figure 3 allows a single object or a combination of several objects. Model allows communication within the database. Meanwhile, View in Figure 4 shows the results of the data contained in the model. The view is in charge of displaying all model data. In general, this view is the one that presents the display of information for the user. Next, Controller in figure 5, The controller connects and handles the processes in the Model and View. In addition, the controller also manages the data flow after receiving commands from the view to the model sent by the user. Soure code in Figure.5 is the controller structure in the online queuing system.

1. defined('BASEPATH') OR exit('No direct script access allowed'); its function is to ensure that there is no direct access to the script.
2. class Queue extends CI_Controller {; is a class declaration named “Queue” that extends the Codeigniter Core class. The rules for creating classes/controllers in CodeIgniter are that the class name must be the same as the controller file name and start with an uppercase letter.
3. public function index(), is a function declaration within the class.

Figure 6 is the initial appearance when the user enters the queue website page. There are 2 menu tabs in the upper right corner, namely login and help. Login is intended for users who have registered and who have been recorded in the system, as shown in Figure.7. Meanwhile, if you do not have a username and password, you are required to register first. The registration form can be seen in Figure 8.

After finished registering, patients will go to the main page. On Figure 9, patients will see the current queue number and queues in the two available polyclinics at the Puskesmas Siak Hulu I in Kampar Regency, namely the General Polyclinic and the Dental Polyclinic. Then the patient takes the queue number to which it is intended as shown in figure 10.

Next, After selecting the desired poly, the poly queue number will be filled automatically according to the number of queues on that day. For example, if the patient chooses a general polyclinic, then a general poly queue will appear, date of registration, registration time and estimated waiting time for the patient, as shown in figure 11. An estimation of patient waiting time, aims so that patients do not wait too long at the Puskesmas, and also so that there is number accumulation of patients in the waiting room at the Puskesmas.

The admin dashboard page can be seen in Figure 12. In the admin dashboard, there are 2 queues at the general poly where the patient has previously entered the queue. On the left there is a menu of patient data, poly, queue poly and menus.

Figure 13 is Patient’s data. Patient Data contains patient data obtained from patient registration, and can also be entered manually by the admin.

Furthermore, the poly data contains the poly data contained in Puskesmas Siak Hulu I , Kampar Regency. This poly can later be edited by changing and updating, deleting and adding as shown in Figure 14. Next, Figure 15 is poly queue, Figure 15 show a queue report at the poly that has been carried out by the patient.

The last one, is Figure 16. Figure 16 is Queue Info. Admin can see queue info that has been entered by the patient, such as name, poly, poly queue number, queue date and queue time.

CONCLUSION

The conclusions from this study are: first, The online queuing system that was built is intended for and adapted to the existing queuing model at the Siak Hulu I Health Center in Kampar-Riau Regency. Second, The software design pattern used uses the Model View Controller (MVC). Third, Patients can not only queue online, but also get an estimated time of arrival. This implementation of course contributes to the puskesmas, that regularize the queue process, so that there is buildup occured in the waiting room. while for patients it provides convenience in the form of time efficiency.

REFERENCES

.Adam, S. I., & Andolo, S. (2019). A New PHP Web Application Development Framework Based on MVC Architectural Pattern and Ajax Technology. 2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019, 1(August), 45–50. https://doi.org/10.1109/ICORIS.2019.8874912.

Burungale, S., Kurane, K., Mhatre, S., & Vora, P. D. (2018). Patient Queue Management System. International Journal of Engineering Science Invention (IJESI), 7(2), 39–41. www.ijesi.org%7C%7CVolumewww.ijesi.org

*name of corresponding author

Devega, M., Zamzami, & Darmayunata, Y. (2022). Web-Based Online Queue Design at Puskesmas Siak Hulu I Kabupaten Kampar-Riau. Sinkron, 7(1), 120–127. https://doi.org/10.33395/sinkron.v7i1.11247.

Dissanayake, N. R., & Dias, G,K, A. (2017). Analysis of the impact of file formats for open data analytics efficiency: a case study with R. GSTF Journal on Computing (JOC), Volume 5(3), 40–44. https://doi.org/10.5176/2251-3043.

Jailia, M., Kumar, A., Agarwal, M., & Sinha, I. (2017). Behavior of MVC (Model View Controller) based Web Application developed in PHP and.NET framework. Proceedings of 2016 International Conference on ICT in Business, Industry, and Government, ICTBIG 2016. https://doi.org/10.1109/ICTBIG.2016.7892651.

Joseph, J. W. (2020). Queuing Theory and Modeling Emergency Department Resource Utilization. Emergency Medicine Clinics of North America, 38(3), 563–572. https://doi.org/10.1016/j.emc.2020.04.006.

Lee, H. Y., & Wang, N. J. (2019). Cloud-based enterprise resource planning with elastic model–view–controller architecture for Internet realization. Computer Standards and Interfaces, 64, 11–23. https://doi.org/10.1016/j.csi.2018.11.005.

Maulana, F. A., Nugroho, H. P. B. A., & Santoso, H. (2017). RUKITON - Smart queue reservation system for medical center. 2017 IEEE 8th Control and System Graduate Research Colloquium, ICSGRC 2017 - Proceedings, August, 159–163. https://doi.org/10.1109/ICSGRC.2017.8070587.

Mital, K. M. (2018). Queuing analysis for outpatient and inpatient services : a case study. 48(3), 419–439. https://doi.org/10.1108/00251741011037783.

Mufid, M. R., Basofi, A., Al Rasyid, M. U. H., Rochimansyah, I. F., & Rokhim, A. (2019). Design an MVC Model using Python for Flask Framework Development. IES 2019 - International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy, Proceedings, Mvc, 214–219. https://doi.org/10.1109/ELECSYM.2019.8901656.

Mukarrama, F. A., Nur'Eni, N., & Fadjryani, F. (2017). Sistem Antrian Single Channel - Multiple Phase dalam Meningkatkan Pelayanan Pembayaran Pajak Kendaraan Bermotor di Kantor Sistem Administrasi Manunggal Satu Atap (SAMSAT) Kota Palu. Natural Science: Journal of Science and Technology, 6(2), 175–186. https://doi.org/10.22487/25411969.2017.v6.i2.8666.

Pramudhita, A. N. (2018). Queue model in health facilities information system. Proceeding - 2017 5th International Conference on Electrical, Electronics and Information Engineering: Smart Innovations for Bridging Future Technologies, ICEEIE 2017, 2018-Janua, 152–157. https://doi.org/10.1109/ICEEIE.2017.8328780.

Purwanda, S. anas, & Indarti, S. (2019). Analysis of Single Channel-Multi Phase Queue Model. Journal of Research in Management, 2(1), 1–7. https://doi.org/10.32424/jorim.v2i1.59.

Sunardi, A., & Suharjito. (2019). MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based. Procedia Computer Science, 157, 134–141. https://doi.org/10.1016/j.procs.2019.08.150.

.

Zamzami, Devega, M., & Darmayunata, Y. (2019). Analisis antrian single channel multi steps pada puskesmas siak hulu i kabupaten kampar-riau. 04, 46–53.

*name of corresponding author