

# Web-based Queue Machine Development using Raspberry and ESP8266

Anggit Damaz Istoko

Department of Informatics

Faculty of Science and Technology, UIN Sunan Kalijaga

Yogyakarta, Indonesia

anggit.damaz@gmail.com

Aulia Faqih Rifa'i, M.Kom.

Department of Informatics

Faculty of Science and Technology, UIN Sunan Kalijaga

Yogyakarta, Indonesia

aulia.faqih@uin-suka.ac.id

**Abstract-** At a recent time, a computer-based queue machine, which is using the computer as both a client and a server, is rated to be less practical and inefficient. In this case, the queue machine will need a number of the computer as many as the lock and the network configuration. Given these points, the aim of this research is to build a practice and applicable queue machine. In the development of this system, the writer adopted the prototyping method. Acquiring Arduino Uno to convert the analog signal become a digital which will be shown in the LED P10, utilizing NodeMCU ESP8266 as a WiFi module, and adapting Raspberry Pi 3 as a server, this queue system expected to solve the problem before. In addition, to build the web, this queue system is using javascript and node.js as the software. This research derives a practice, easy to use, and portable queue machine because it uses wifi for their connection.

**Keywords-** component; Queue machine; ESP8266 Microcontroller; Arduino Uno Microcontroller; Raspberry Pi 3.

## I. INTRODUCTION

At the moment, there are still some institution that has not been used to provide technological facilities for the users, although they know it might form the system much easier. Queues often become some obstacles for the customer service in providing services to users. The queuing system is one approach to retain user service and increase user satisfaction services so that the institution has the capability to provide the best service [1].

The queue system used previously uses a computer as a server and as a client. This system still has many disadvantages like the number of computer needed, that are same as the lockets. Moreover, the configuration is rather complicated.

The expected queue system is a system that uses Arduino Uno (Arduino), NodeMCU ESP8266 (WiFi module) and Raspberry Pi 3 (Raspberry). Raspberry itself (which runs the Raspbian operating system) functions as a server that can receive data requests from NodeMCU ESP8266 that are available on the console or display via a wifi network [2]. To display the queue at the P10 LED display counter, Arduino Uno as a microcontroller receives data from NodeMCU ESP8266 [3]. All devices and software used in this system are open sources. Also things that do not raise legality issues and reduce overall costs in the design system.

## II. PURPOSE

The aim of this research is to develop the advanced queue system which have superiorities: (1) make it easier for operators to call the queue, (2) reduce the need to cover internet connection costs, (3) do not use cables for device installation reviews, and (4) users are able to monitor the queue number that is being served so that they can predict the time so that they do not need to wait too long.

## III. METHODOLOGY

### A. Material

In this study, several tools and materials were grouped into two parts, namely hardware (hardware) and software (software).

The tools needed in this research are laptops,

- Laptop: as the main tool in developing software, install the operating system, control and configure Raspberry Pi.
- Raspberry Pi 3: as the main device to be configured and utilized by the operating system.
- Micro SD: as a storage for operating systems, applications and some other data used by Raspberry Pi.
- SD Card adapter: as a connector between Micro SD Card memory with a laptop SD Card port

- Ethernet Cable: as connecting medium between Raspberry Pi with a laptop when controlling and configuring Raspberry.
- Micro Usb Cable: used to drain power to the Raspberry Pi device.
- NodeMCU ESP8266 Microcontroller: an used device for requesting data for Raspberry Pi which will be forwarded to Arduino Uno.
- Arduino Uno Microcontroller: receive data from the NodeMCU ESP8266 and display it on the P10 LED screen.
- P10 : used to display queue number data.
- Console : acted to send data to Raspberry Pi and request data from Raspberry Pi.
- Soldering
- Tenol
- Jumper Cables
- Raspberry Pi 3 Casings
- Raspberry Casing Fan
- Raspberry Heatsink

In addition, some software is also needed, namely:

- Raspbian Jessie: operation system.
- Win32 Disk Imager: a software to install the system operation
- Putty: SSH and Telnet Client
- Arduino IDE: the code editor of microcontroller NodeMCU ESP8266 and Arduino Uno
- Fritzing: to arrange the scheme of microcontroller device with the other devices
- Draw.io : to form the diagram

### B. Literature Study

Literature study has been done by the author so that relevant information and data were carried out. The data and information were collected from books, journals, theses, internet, and so on. According to the Koentjaraningrat library technique is a way of data containing various kinds of material in the literature, such as newspapers, books, manuscripts, documents and so on that are relevant to research.

### C. System Arrangement

According to Jogiyanto HM [4], the system arrangement can be defined as drawing, planning and making sketches or configurations of several elements that are separated into a whole and working whole. In designing this system, the authors carry out functional requirements analysis, non-functional needs analysis, block diagram design, system modeling, and designing *user interface*.

### D. System Making

This stage it is the core of all steps [5]. The systems are created carefully. The system is made based on a system that



has been made previously. It is made gradually, started from the hardware forthwith the software.

*E. System Testing*

System testing is carried out after the making of system has been completed, and is carried out periodically. System testing is done with the aim to find out whether the system that has been developed is in accordance with or not [6].

System testing that will be executed Raspberry hardware testing, console devices testing, network testing, API testing, and system testing.

IV. RESULTS AND DISCUSSION

A. System Development

The development consists of two parts, which are the hardware development and software development. Hardware development includes the development of console devices and LED displays which being used to display queue numbers using P10 LEDs that will convey data with Raspberry Pi. Meanwhile, software development includes system preparation, network development, and web application development.

1) Hardware Development

In developing the hardware, author uses the NodeMCU ESP8266 microcontroller, Arduino Uno microcontroller and Arduino microcontroller (Fig 1 & Fig 2).

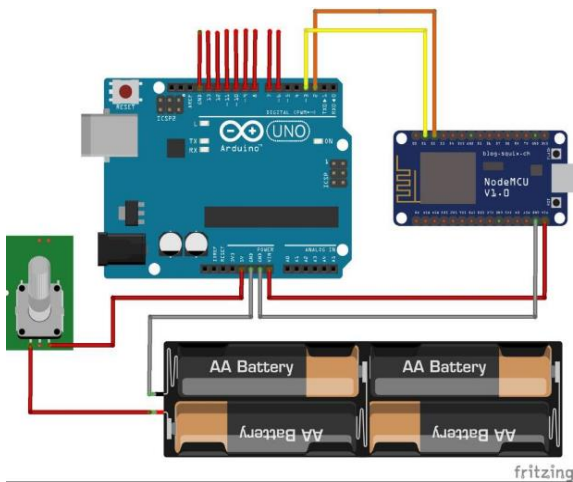


Figure 1. Output circuit to P10LED display



Figure 2. LED P10

2) Software Development

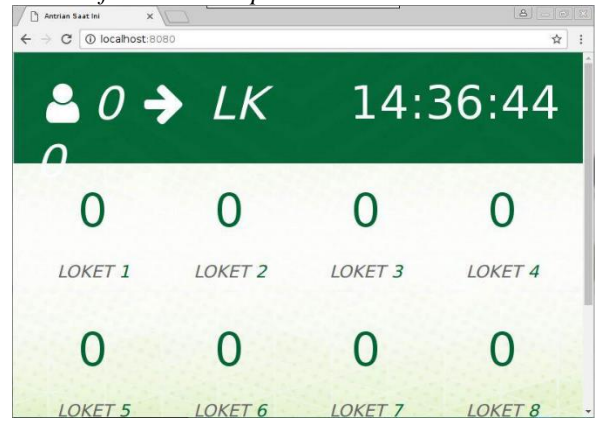


Figure 3. The main menu of the queue

There are several steps that can be done in developing the software in this study, include:

- a) *Preparing the system:* Arrange Raspberry Pi boards to be used in the development system.
- b) *Network Development:* Manage Wireless Networks (Wifi). The wireless network will be used as a link between the console and the P10 display. Therefore the network is set as a hotspot or access point.
- c) *Database development:* The database used is MongoDB. The author chooses to use this database because the data from the queue that will be stored is flexible and does not depend on complicated schemes.
- d) *Application Programming Interface (API) Development:* Application Programming Interface (API) using server-based javascript programming language, Node.js. API is built to connect data communication between Raspberry Pi, console and display.
- e) *NodeMCU ESP8266 and Arduino Uno Development:* In the system development on ESP and on Arduino Uno, the author uses C++ programming language through Arduino IDE.
- f) *Web-Based Application Development:* The author uses a web-based application in its development which aims to maintain the queue numbers displayed. On this web-based application, the author uses the Javascript programming language and is supported by HTML and CSS using semantic-UI for how it looks. The result can be displayed in on Fig 3.

B. System Implementation

System implementation is a mechanism to form and implement new information into an operation.

On this research, the queue engine system implementation is divided into two parts. The first part is the implementation of hardware consisting of the Arduino Uno microcontroller and the NodeMCU ESP8266 on the P10 LED Screen as the device output, and NodeMCU ESP8266 on the console device as an input device. Correspondingly, the other part is the software implementation consisting of microcontroller software and web-based application software.



TABLE I. SYSTEM TESTING

No	Testing Item	Detailed Testing	Results
1	Console	Console is able to requesting the next queue numbers to specific lockets Console is able to requesting a recall on the queue numbers on specific locket	Match
2	Display	Display has the ability to display the numbers and counters	Match
3	Sound	Sound can play audio file	Match
4	System	System is able to set the queue process and display detailed queue information	Match

TABLE I. shows that the testing of queue engine systems is divided into two parts. The first part is hardware testing which includes the Arduino Uno microcontroller, the NodeMCU ESP8266 microcontroller, the console device, and the output device. Together with this, the second part is software testing that includes microcontroller software testing and web-based queue system testing. It aims to find out the software and hardware that has been made that is suitable for the purpose of the software and hardware development.

## V. CONCLUSIONS

Based on experimentation that have been accomplished during The Development of Web-Based Queue Machines Using Raspberry and ESP8266, it can be obtained as follows: (1) Raspberry can be utilized as a server properly, (2) the connected endpoint devices (NodeMCU ESP8266) are able to send, receive and respond well, (3) Arduino Uno microcontroller can receive and complete data properly, (4) the developed software that can be used to monitor the queue data, (5) Chromium Web Browser is less accurate to play queue sound sustainably.

## ACKNOWLEDGMENTS

I am grateful to Allah for the good health and wellbeing that were necessary to complete this research. I wish to express my sincere thanks to Sunan Kalijaga State Islamic University Admission Office for providing me with all the necessary facilities for the research. I would also take this opportunity to express gratitude to my family, lecturers, and friends for their help and support.

- [1] N. Fridatama, Rancang Bangun Sistem Layanan Antrian Rumah Sakit Berbasis Java, Surabaya: Institut Teknologi Sepuluh Nopember, 2011.
- [2] Saptadi, A. H., Model Sistem Antrian Loket menggunakan Aplikasi Processing dengan Sistem Mikropengendali Arduino dan Raspberry PI, Yogyakarta: Institut Sains & Teknologi AKPRIND Yogyakarta, 2014.
- [3] Putro, A. E, Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi Edisi 2, Yogyakarta: Gava Media, 2006.
- [4] Jogyanto, Analisis & Desain Sistem Informasi : Pendekatan terstruktur teori dan praktek aplikasi bisnis, Yogyakarta: Andi, 2001.
- [5] Heizer, J., & Render, B., Operations Management. Tenth Edition, New Jersey, USA: Pearson, 2011.
- [6] M. Iqbal, Analisis Kinerja Sistem Pendekatan Teori dan Praktek, Depok: Penerbit Gunadarma, 2011.

