

RANCANG BANGUN BARCODE PRINTER SERVER DENGAN MENGGUNAKAN ACTIVEMQ CLASSIC DI RASPBERRY PI 3

Sadr Lufti Mufreni^{*)}

Jurusan Teknologi Informasi, Universitas 'Aisyiyah Yogyakarta
Jalan Siliwangi (Ring Road Barat) 63, Mlangi, Nogotirto, Gamping, Sleman,
D.I. Yogyakarta, 55292, Indonesia

^{*)}E-mail: sadr@unisayogya.ac.id

Abstrak

Sistem antrian memiliki empat komponen yaitu modul monitor untuk menampilkan informasi antrian, modul printer barcode untuk mencetak nomor antrian, modul pengguna untuk menambah atau memproses data antrian, dan modul server untuk mengelola antrian. Ada kebutuhan untuk menampilkan daftar antrian di layar yang lebih besar dan ditempatkan di posisi yang tinggi. Jika ini dilakukan maka pendaftar akan kesulitan untuk mendapatkan nomor antrian karena modul printer barcode harus berada di dekat modul monitor. Hal ini disebabkan karena sistem tidak mendukung pemisahan modul. Penelitian ini membuat sistem untuk memenuhi kebutuhan tersebut dengan cara mengubah alur penambahan data antrian. Alur dipecah menjadi dua modul yaitu modul messaging queue untuk menyimpan permintaan dari sistem antrian yang ada dan modul printer barcode untuk mencetak nomor antrian berdasarkan permintaan di messaging queue. ActiveMQ Classic dipilih sebagai implementasi dari messaging queue. Penelitian ini membuktikan bahwa ActiveMQ Classic dapat digunakan sebagai printer server dan dapat dijalankan di Raspberry Pi 3. Pemilihan Raspberry Pi 3 dikarenakan sistem antrian yang sudah ada menggunakannya. Uji coba dilakukan untuk membuktikan bahwa penelitian berhasil menggunakan ActiveMQ Class sebagai printer server.

Kata kunci: sistem antrian, messaging queue, barcode printer server, barcode printer

Abstract

The queue system has four components namely a monitor module to display queue information, a barcode printer module to print queue numbers, a user module to add or process queue data, and a server module to manage queues. There is a need to display queue list on larger screen and higher place. If this is done, then the registrar will have a hard time getting queue number because printer module must be near monitor module. This is because the system does not support module partitioning. This research makes a system to meet those needs by changing the flow of data queue addition. The flow is broken down into two modules namely the messaging queue module to store requests from the existing queue system and the barcode printer module to print queue numbers based on requests in the messaging queue. ActiveMQ Classic was chosen as an implementation of the messaging queue. This research proves that ActiveMQ Classic can be used as a printer server running on Raspberry Pi 3. The Raspberry Pi 3 selection is due to the existing queue system. Trials are carried out to prove that the research successfully used ActiveMQ Class as a printer server.

Keywords: queue system, messaging queue, barcode printer server, barcode printer

1. Pendahuluan

Sistem antrian memiliki empat komponen yaitu modul monitor untuk menampilkan informasi antrian, modul printer barcode untuk mencetak nomor antrian, modul pengguna untuk menambah atau memproses data antrian, dan modul server untuk mengelola antrian. Ada kebutuhan untuk menampilkan daftar antrian di layar yang lebih besar dan ditempatkan di posisi yang tinggi. Jika ini dilakukan maka pendaftar akan kesulitan untuk mendapatkan nomor antrian karena modul printer barcode harus berada di dekat modul monitor. Hal ini disebabkan karena sistem tidak mendukung pemisahan modul.

Penelitian ini merupakan pengembangan dari penelitian sebelumnya [1]. Penelitian ini berfokus pada sistem printer barcode server saja. Penelitian menggunakan messaging queue untuk membuat printer server belum ditemukan oleh peneliti. Penelitian yang sudah ada [2][3][4][5][6] berfokus kepada internet of things bukan sebagai printer server.

Sistem penelitian [1] terdiri dari empat komponen utama yaitu modul monitor untuk menampilkan informasi antrian, modul printer barcode untuk mencetak nomor antrian, modul pengguna untuk menambah atau memproses nomor antrian, dan modul server yang mengelola nomor antrian. Permasalahan terjadi ketika

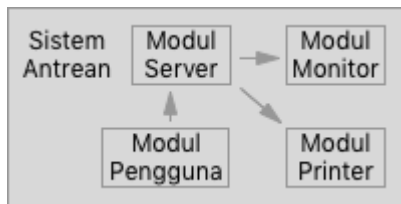
modul monitor dan modul server akan dipisahkan dengan modul pengguna dan modul printer barcode. Sistem yang ada belum mendukung kebutuhan tersebut. Penelitian ini dilakukan untuk memenuhi kebutuhan tersebut. Tujuan lain yang ingin dicapai adalah nomor antrean harus tercetak secara urut dan sistem yang baru dapat dijalankan di Raspberry Pi 3. Penggunaan Raspberry Pi dilakukan untuk menjaga tujuan pembuatan sistem antrean pada penelitian [1] yaitu menggunakan perangkat keras yang murah.

2. Metode

Penelitian dalam menganalisis sistem menggunakan Unified Modelling Language (UML). Diagram yang digunakan adalah sequence diagram [7]. Hal ini dipilih karena sequence diagram tepat untuk menggambarkan interaksi modul yang ada [8].

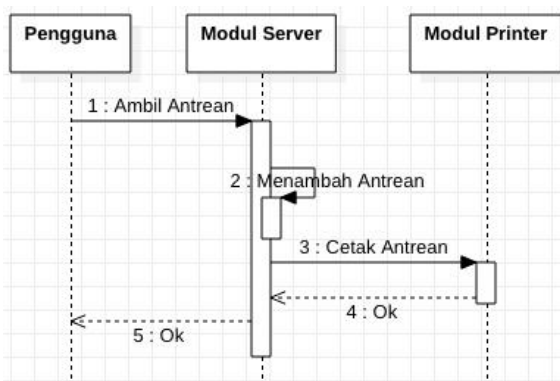
2.1. Analisis Sistem Awal

Sistem awal mempunyai dua fungsi utama yaitu menambah nomor antrean dan memproses nomor antrean. Untuk memenuhi fungsi utama tersebut, sistem dibagi menjadi empat modul yaitu modul monitor, modul printer, modul pengguna, dan modul server (Gambar 1).



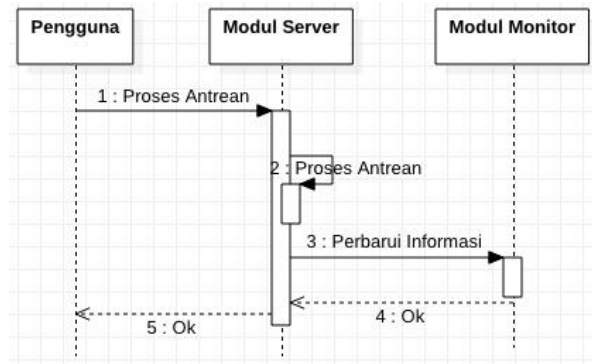
Gambar 1. Arsitektur Sistem Awal

Alur dari menambah nomor antrean adalah pengguna meminta nomor antrean dengan mengeklik modul pengguna, modul pengguna akan mengirimkan permintaan ke modul server, modul server akan mengolah permintaan dan mengirimkan data ke modul printer barcode. Modul printer barcode akan mencetak nomor antrean yang akan dibawa oleh pengguna (Gambar 2).



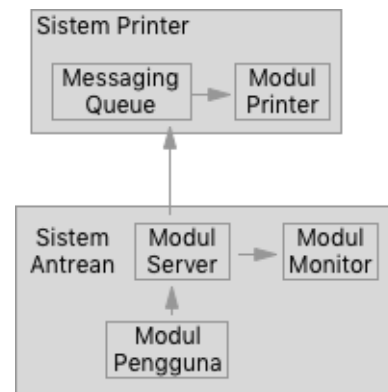
Gambar 2. Sequence Diagram Menambah Nomor Antrean

Alur dari memproses nomor antrean adalah pengguna mengeklik permintaan proses antrean di modul pengguna, modul pengguna mengirimkan permintaan tersebut ke modul server, modul server memproses permintaan tersebut dan menampilkan perubahan nomor antrean di modul monitor (Gambar 3).



Gambar 3. Sequence Diagram Memproses Nomor Antrean

2.2. Analisis Sistem Printer Barcode



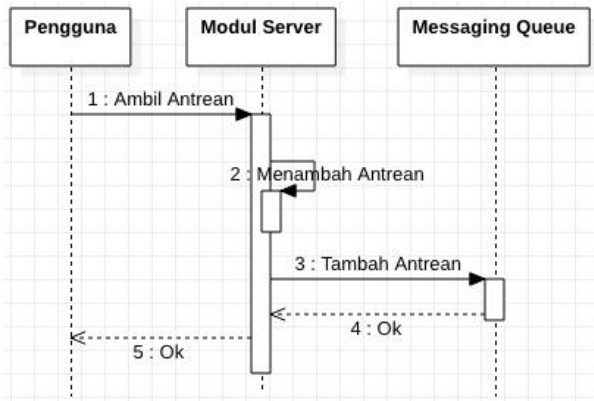
Gambar 1. Arsitektur Sistem Terbaru

Sistem printer barcode dikembangkan untuk memenuhi kebutuhan yang baru. Perubahan yang dilakukan dengan memisahkan modul printer barcode menjadi sistem tersendiri yaitu sistem printer barcode. Perubahan sistem memerlukan modifikasi untuk alur menambah nomor antrean. Sistem printer barcode terdiri dari 2 modul yaitu modul messaging queue dan modul printer (Gambar 4). Modul messaging queue bertanggung jawab untuk menerima permintaan cetak nomor antrean. Sistem Printer Barcode memiliki fitur sebagai berikut:

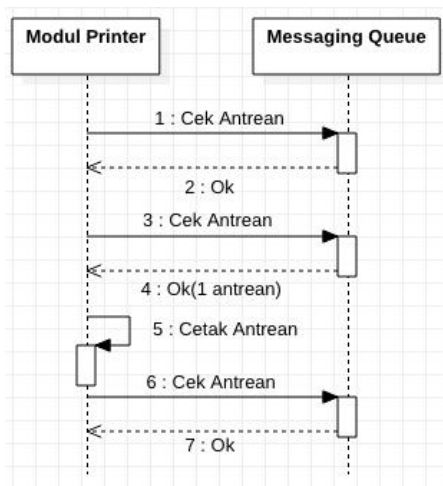
1. Bersifat asynchronous sehingga meminimalisir jeda ketika meng-klik.
2. Messaging Queue menjamin urutan percetakan nomor terpenuhi.

Alur menambah antrean dipecah menjadi dua yaitu menambah antrean di messaging queue (Gambar 5) dan mencetak antrean (Gambar 6). Menambah antrean di

messaging queue dilakukan oleh modul server ketika menerima klik dari modul pengguna. Proses percetakan nomor tidak dilakukan pada saat itu juga, ini perbedaan dengan sistem awal.



Gambar 5. Sequence Diagram Menambah Antrean di Messaging Queue



Gambar 6. Sequence Diagram Proses Antrean di Modul Printer

2.3. Implementasi Sistem

2.3.1. Java

Penelitian [1] menggunakan bahasa pemrograman Java. Kelebihannya yaitu aplikasi dapat berjalan di mayoritas sistem operasi selama terinstall Java Virtual Machine [9]. Penelitian ini menggunakan bahasa pemrograman yang sama.

2.3.2. ActiveMQ Classic (Messaging Queue Server)

Salah satu library messaging queue adalah ActiveMQ. ActiveMQ mempunyai dua versi yaitu Classic dan Artemis. ActiveMQ Classic digunakan karena peneliti sudah sering menggunakan library tersebut. Library ini

dipilih karena bisa ditanam dan dijalankan dari sistem berbasis Java [10].

2.3.3. Messaging Queue Telemetry Transport (MQTT)

ActiveMQ mendukung beberapa protokol untuk komunikasi. Salah satunya adalah MQTT. MQTT dipilih karena dapat mengirimkan data besar dengan overhead yang kecil [11][12][13].

2.3.4. JSON

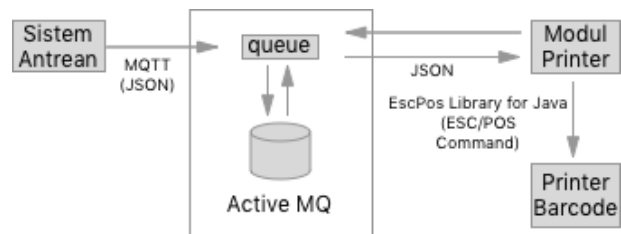
Komunikasi antara sistem antrean dan sistem printer dalam alur permintaan menambah antrean menggunakan JavaScript Object Notation (JSON) data format. Format ini merupakan standar de-facto di dunia maya [14].

2.3.5. ESC/POS (EscPos Library for Java)

ESC/POS merupakan bahasa standar yang digunakan oleh receipt printer. Bahasa ini dibuat oleh perusahaan EPSON. Library yang digunakan adalah EscPos Library for Java.

2.3.6. Raspberry Pi 3

Komputer mini Raspberry Pi 3 digunakan untuk menjalankan sistem antrean. Raspberry Pi digunakan dengan mempertimbangkan penggunaan daya listrik yang relatif rendah dibandingkan komputer pribadi. [15]. Sistem operasi yang digunakan adalah Rasbian, sistem operasi berbasis Debian untuk Raspberry Pi.



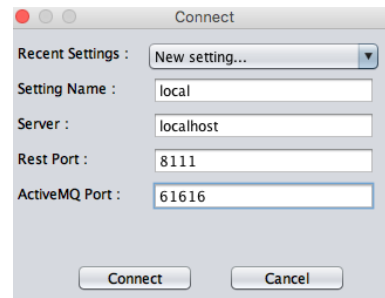
Gambar 7. Arsitektur dan Interaksi Sistem Printer Barcode

Sistem Antrean terdiri dari 3 komponen sistem terdahulu yaitu modul monitor, modul server, dan modul pengguna. Sistem antrean akan memberikan nomor antrean baru yang akan disimpan di queue oleh ActiveMQ. Modul Printer memeriksa apakah ada antrean baru yang harus dicetak di queue. Jika ada maka Modul Printer memberikan perintah kepada printer Barcode untuk mencetak nomor antrean (Gambar 7).

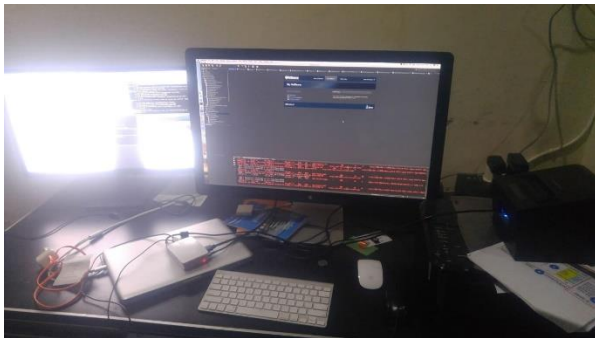
3. Hasil dan Analisis

Penambahan fitur berhasil dilakukan, dibuktikan dengan percobaan dengan menggunakan satu Macbook Pro dan satu Raspberry Pi (Gambar 8). Macbook Pro digunakan untuk menjalankan sistem antrean dan Raspberry Pi

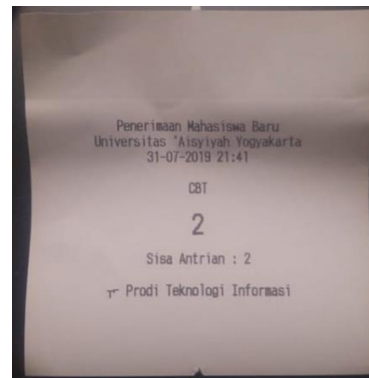
menjalankan sistem printer barcode. Raspberry Pi 3 berhasil menjalankan sistem printer barcode server dengan menggunakan ActiveMQ Classic (Gambar 9). Modul printer mencek antrean yang baru dari ActiveMQ, bila ditemukan antrean baru maka akan dicetak di barcode (Gambar 10). Konfigurasi port untuk ActiveMQ dilakukan pada saat sistem antrean pertama kali dijalankan (Gambar 11). Hasil cetak yang diperoleh setelah menggunakan sistem antrean yang baru (Gambar 12).



Gambar 11. Connect Dialog Terbaru Dengan ActiveMQ Port



Gambar 8. Alat pengetesan sistem antrean dan sistem printer barcode



Gambar 12. Hasil Cetak Menggunakan Sistem Printer Barcode

Penelitian menfokuskan pada penggunaan ActiveMQ Classic sebagai printer server dan berhasil dibuktikan.

```
Read configurations done
INFO | Using Persistence Adapter: KahaDBPersistenceAdapter[/Volumes/Data/ASDEV/MyCounter/MyCounterServer/activemq-data/localhost/KahaDB]
INFO | JMX consoles can connect to service:jmx:rmi:///jndi:rmi://localhost:1099/jmxrmi
INFO | KahaDB is version 6
INFO | PLISTore: [/Volumes/Data/ASDEV/MyCounter/MyCounterServer/activemq-data/localhost/tmp_storage] started
INFO | Apache ActiveMQ 5.15.5 (localhost, ID:Sadrs-MacBook-Pro.local-54343-1564498737148-0:1) is starting
INFO | Listening for connections at: tcp://Sadrs-MacBook-Pro.local:61616
INFO | Connector tcp://Sadrs-MacBook-Pro.local:61616 started
INFO | Apache ActiveMQ 5.15.5 (localhost, ID:Sadrs-MacBook-Pro.local-54343-1564498737148-0:1) started
INFO | For help or more information please see: http://activemq.apache.org
WARN | Store limit is 102400 mb (current store usage is 1 mb). The data directory: /Volumes/Data/ASDEV/MyCounter/MyCounterServer/activemq-data/lo
ximum available disk space: 4839 mb
WARN | Temporary Store limit is 51200 mb (current store usage is 0 mb). The data directory: /Volumes/Data/ASDEV/MyCounter/MyCounterServer/activemq
maximum available disk space: 4837 mb
```

Gambar 9. ActiveMQ Berjalan Dengan Baik Di Sistem Printer Barcode

```
Printer Locket 1 run
Console Locket 1 message : {"total_count":5,"current_count":0,"counter_name":"BEASISWA","type":0}
Console Locket 1 message : {"total_count":2,"current_count":0,"counter_name":"ANVULLEN & PROFESI","type":0}
Console Locket 1 message : {"total_count":4,"current_count":0,"counter_name":"CBT","type":0}
Console Locket 1 message : {"total_count":3,"current_count":0,"counter_name":"INFO PMB","type":0}
Console Locket 1 message : {"total_count":5,"current_count":0,"counter_name":"CBT","type":0}
Console Locket 1 message : {"total_count":1,"current_count":0,"counter_name":"S2 KEBIDANAN","type":0}
Console Locket 1 message : {"total_count":2,"current_count":0,"counter_name":"PMDK & PMBU","type":0}
Console Locket 1 message : {"total_count":4,"current_count":0,"counter_name":"INFO PMB","type":0}
```

Gambar 10. Penambahan dan Pemrosesan Nomor Antrean

4. Kesimpulan

Uji coba membuktikan bahwa sistem sudah memenuhi tujuan dari penelitian, yaitu: modul printer dipisahkan dari aplikasi, tidak ada jeda waktu (hang) ketika meng-klik tombol antrean, nomor dicetak secara terurut dan dapat dijalankan di Raspberry Pi 3. Pengembangan dapat dilakukan dalam 2 aspek yaitu reliabilitas ActiveMQ sebagai printer server dan pengembangan lebih lanjut sistem antrean dengan memisahkan modul monitor dengan aplikasi. Pemisahan modul monitor memungkinkan informasi antrean dapat dipasang di ruangan yang berbeda, ini memudahkan bagi pengunjung untuk mengetahui informasi antrean terkini.

Referensi

- [1]. S. L. Mufreni, "Rancang Bangun Sistem Antrean Multi Fungsi Yang Restful Dengan Menggunakan Credit Card-Sized Computer," *J. Transm.*, vol. 20, no. April, pp. 79–84, 2018.
- [2]. A. P. Segara, R. Primananda, and S. R. Akbar, "Implementasi MQTT (Message Queuing Telemetry Transport) pada Sistem Monitoring Jaringan berbasis SNMP (Simple Network Management Protocol)," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 2, pp. 695–702, 2018.
- [3]. H. A. Rochman, R. Primananda, and H. Nurwasito, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 6, pp. 445–455, 2017.
- [4]. R. P. Pratama, P. K. Malang, R. P. Pratama, and S. Grid, "Aplikasi Wireless Sensor Esp8266 Untuk Smart Home," *Semin. Nas. Teknol. dan Rekayasa*, vol. IV, pp. 1–10, 2017.
- [5]. S. Katsikeas et al., "Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol," *Proc. - IEEE Symp. Comput. Commun.*, no. October, pp. 1193–1200, 2017.
- [6]. C. Hasiholan, R. Primananda, and K. Amron, "Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT," vol. 2, no. 12, pp. 6128–6135, 2018.
- [7]. A. Dennis, B. H. Wixom, and D. Tegarden, *Systems Analysis and Design : An Object-Oriented Approach with UML*, 5th ed. Wiley Publishing, Inc, 2015.
- [8]. B. Rumpe, *Modeling with UML - Language, concept, methods*. 2016.
- [9]. P. Deitel and H. Deitel, *Java How to Program*, 10th Edition, Late Objects Version. Pearson Education, Inc., 2015.
- [10]. B. Snyder, D. Bosanac, and R. Davies, *ActiveMQ In Action*, 1st ed. Stamford,: Manning Publications Co, 2011.
- [11]. G. C. Hillar, *MQTT Essentials - A Lightweight IoT Protocol*, 1st ed. Packt Publishing, 2017.
- [12]. D. Soni and A. Makwana, "A survey on mqtt: a protocol of internet of things(IoT)," *Int. Conf. Telecommun. Power Anal. Comput. Tech. (Ictpact - 2017)*, no. April, pp. 0–5, 2017.
- [13]. M. V. Masdani and D. Darlis, "A comprehensive study on MQTT as a low power protocol for internet of things application," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 434, no. 1, 2018.
- [14]. B. Smith, *Beginning JSON*. Apress, 2015.
- [15]. Future Publishing, "The Ultimate Raspberry Pi Handbook," pp. 1–1037, 2016.