

Analisis Implementasi OV2640 dan MFRC522 Sebagai Bagian dari Sistem Belanja Self-Service

Fahreza Haidar Avif¹, Raden Sumiharto*², Diyah Utami Kusumaning Putri³,
Ilona Usuman⁴

¹ Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia

^{2,3,4} Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ¹fahrezahaydar@mail.ugm.ac.id, *²r_sumiharto@ugm.ac.id, ³diyah.utami.k@ugm.ac.id, ⁴ilona@ugm.ac.id

Abstrak

Peningkatan angka konsumsi dan perubahan preferensi masyarakat pasca pandemi COVID-19 mengakibatkan minat terhadap layanan mandiri (self-service) kian meningkat. Hal ini mendorong sektor retail untuk beradaptasi dalam mengembangkan sistem belanja self-service. Penelitian yang ada sebelumnya telah melakukan eksplorasi mengenai potensi penggantian identifikasi barang berbasis barcode dengan alternatif identifikasi lain seperti Quick Response Code (QR-Code) atau Radio Frequency Identification (RFID). Oleh karena itu penelitian ini membahas mengenai pengaruh identifikasi QR-Code dan RFID dalam kecepatan transaksi pada sistem belanja self-service. Penelitian ini menggunakan OV2640 untuk identifikasi QR-Code dan MFRC522 untuk identifikasi RFID. Hasil penelitian yang dilakukan menunjukkan identifikasi RFID memiliki kecepatan respon 3-4 kali lebih cepat dibandingkan identifikasi QR-Code sehingga mampu menyelesaikan transaksi 63% lebih cepat dibandingkan identifikasi QR-Code.

Kata kunci—Layanan Mandiri, QR-Code, RFID, Web-Application, Lingkungan Cerdas

Abstract

Increase of consumption rates and preferences change of consumers in the post COVID-19 era have resulted a growing demand for self-service solutions. This has prompted the retail sector to adapt and develop self-service shopping systems. Prior research has explored the potential of replacing barcode-based item identification with alternative methods such as Quick Response Code (QR-Code) or Radio Frequency Identification (RFID). Therefore, this study discusses the impact of QR-Code and RFID identification on transactions speed in self-service shopping systems. This study using OV2640 for QR-Code identification and MFRC522 for RFID identification. The results of the study indicate that RFID identification has a response speed that is 3-4 times faster than QR-Code identification, enabling transactions to be completed 63% faster compared to QR-Code identification.

Keywords—Self-Service, QR-Code, RFID, Web-Application, Smart System

1. PENDAHULUAN

Era globalisasi membawa dampak positif dan negatif bagi masyarakat. Salah satu dampak negatifnya adalah pola konsumsi masyarakat yang cenderung menjadi lebih konsumtif. Data dari Badan Pusat Statistika (BPS) menunjukkan bahwa rata-rata pengeluaran masyarakat

untuk konsumsi per bulan pada tahun 2023 meningkat 9,35% dibandingkan tahun sebelumnya [1].

Berdasarkan statistik tersebut, sektor ritel menjadi sektor yang paling terpengaruh dengan peningkatan konsumerisme ini karena merupakan sektor tempat transaksi jual beli barang atau jasa tingkat akhir. Pengaruh berupa peningkatan volume pembeli dapat dilihat terutama pada supermarket atau swalayan yang merupakan tempat favorit masyarakat berbelanja kebutuhan. Tingginya volume pembeli di sektor ritel selain mampu memberi keuntungan lebih, namun juga memberi tantangan dan permasalahan baru terkait pelayanan transaksi yang diberikan oleh pelaku ritel. Pelayanan transaksi yang lambat dan tidak mumpuni dalam menghadapi tingginya jumlah pembeli seringkali mengakibatkan antrian panjang yang mengurangi kenyamanan pembeli dan berpotensi untuk membuat pembeli untuk mengurungkan niat untuk berbelanja atau memilih berbelanja di tempat lain [2] [3].

Permasalahan tersebut juga diperparah dengan berubahnya preferensi masyarakat pasca pandemi COVID-19. Masyarakat cenderung menghindari kerumunan di tempat umum sehingga layanan mandiri/*self-service* menjadi semakin diminati[4]. Minat masyarakat yang tinggi terhadap layanan mandiri / *self-service* yang dinilai cepat dan nyaman mendorong sektor *retail* untuk adaptif dalam menanggapi kondisi yang terjadi dengan berkembangnya sistem belanja mandiri/sistem belanja *self-service* [5].

Perkembangan sistem belanja *self-service* membawa perubahan terhadap komponen yang digunakan pada proses transaksi seperti penggunaan pembayaran non tunai/*cashless* seperti penggunaan kartu kredit/debit, transfer bank, *e-wallet* dan lainnya yang kini sangat digemari [6]. Selain perubahan pada metode pembayaran, perkembangan sistem belanja *self-service* juga mempertimbangkan untuk mengganti penggunaan identifikasi barcode dengan identifikasi lain.

Upaya perubahan metode identifikasi dilakukan pada penelitian Pallavi P et al[2] dan Saini et al [7] dengan mengembangkan sistem belanja *self-service* menggunakan identifikasi QR-Code. QR-Code, singkatan dari "*Quick Response Code*" adalah jenis barcode dua dimensi yang menyimpan data atau informasi terepresentasi dalam kumpulan kotak hitam [10]. QR-Code yang disematkan di produk akan dibaca dengan *reader* yang terdiri dari modul kamera OV2640 dan terhubung pada ESP32-CAM. *Reader* diletakkan pada setiap troli belanja agar pembeli dapat melakukan *scanning* barang yang dibeli dengan leluasa dan akan menampilkan data mengenai produk yang telah di-*scan* di *mobile-apps* yang disediakan.

Upaya lainnya juga dilakukan pada penelitian Shahroz et al.[8] dan Das et al.[9] dengan mengembangkan sistem belanja *self-service* menggunakan identifikasi RFID. *Radio-Frequency Identification* atau yang disingkat RFID merupakan sistem identifikasi tanpa kontak fisik yang memanfaatkan gelombang radio untuk melakukan pertukaran data antara *reader*/pembaca dengan *tag/transponder*. [11]. RFID tag yang disematkan ke produk akan dibaca dengan perangkat *reader* yang terdiri dari modul MFRC522 dan Arduino Uno. Sama dengan penelitian yang dijelaskan sebelumnya, *reader* akan diletakkan pada setiap troli belanja agar pembeli dapat melakukan *scanning* barang yang dibeli dengan leluasa dan akan menampilkan data mengenai produk yang telah di-*scan* di *mobile-apps* yang disediakan.

Berdasarkan penelitian yang telah dilakukan sebelumnya, penggunaan identifikasi QR-Code atau RFID dipertimbangkan untuk menggantikan identifikasi barcode yang sekarang digunakan. Meski dengan perbedaan pendekatan antara kedua identifikasi tersebut, hasil yang diinginkan adalah mengembangkan sistem belanja *self-service* dengan proses transaksi yang cepat bagi pembeli sehingga penelitian ini akan membahas mengenai pengaruh dan kecepatan kedua identifikasi tersebut dalam transaksi belanja *self-service*. Sama seperti penelitian sebelumnya, akan digunakan modul kamera OV2640 untuk identifikasi QR-Code dan modul MFRC522 untuk identifikasi RFID. Penelitian ini sekaligus akan membangun sebuah sistem belanja *self-service* berbasis *Web-Application* yang memiliki fitur otentikasi pembeli, terintegrasi dengan *payment gateway* untuk menangani pembayaran pembeli secara *cashless*, dan dapat digunakan untuk kedua identifikasi yang diujikan.

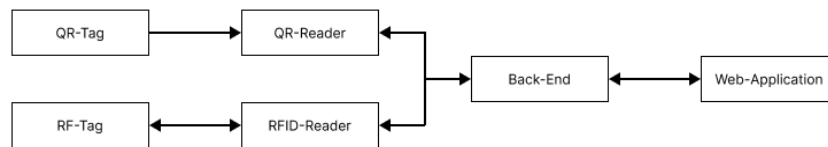
2. METODE PENELITIAN

2.1 Analisis Sistem

Penelitian ini dilakukan untuk menguji pengaruh perbedaan identifikasi QR-Code dengan OV2640 dan identifikasi RFID dengan MFRC522 terhadap transaksi belanja dalam sistem belanja *self-service*. Oleh karena itu, diperlukan sistem belanja *self-service* yang dapat digunakan dan memiliki kebutuhan sebagai berikut :

1. Sistem dapat menggunakan kedua identifikasi secara bersamaan.
2. Sistem memiliki fitur otentikasi untuk pengguna sebelum melakukan transaksi.
3. Sistem memberi informasi transaksi ke pengguna dalam sebuah antarmuka.
4. Komunikasi antara identifikasi dan antarmuka dapat berjalan secara *realtime*.
5. Mengakomodasi pembayaran non-tunai untuk penyelesaian transaksi.

Berdasarkan kebutuhan sistem diatas, sistem belanja *self-service* yang dirancang memiliki beberapa komponen yaitu *Tag*, *Reader*, *Back-End (Authentication, Database dan Payment Gateway)* dan *Web-Application*. Garis besar dari sistem belanja *self-service* yang dirancang ditunjukkan pada Gambar 2.1.

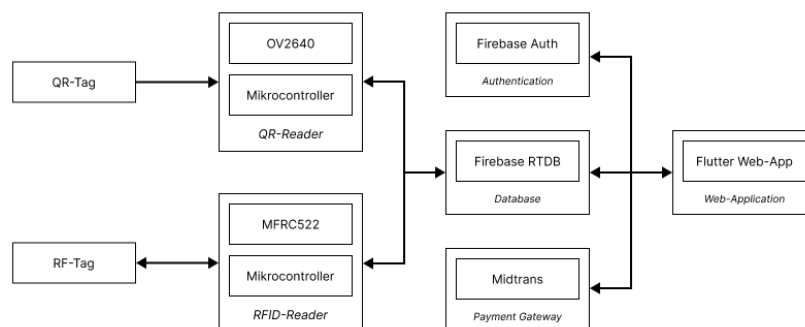


Gambar 2.1 Diagram Garis Besar Sistem Belanja Self-Service

Berdasarkan garis besar sistem diatas, sistem belanja *self-service* yang dirancang untuk penelitian ini memiliki rincian sebagai berikut :

1. *Tag* QR dan RFID sebagai pembawa data berupa kode produk.
2. *Reader* berupa OV2640 dan MFRC522 yang terhubung dengan *Microcontroller* dan tersambung dengan jaringan WiFi untuk membaca data yang dibawa *tag*.
3. *Firestore* untuk otentikasi pengguna dan komunikasi data secara *realtime*.
4. *Midtrans Payment Gateway* untuk mengakomodasi pembayaran.
5. *Web-Application* yang dibuat dengan *Flutter SDK* sebagai antarmuka dan pusat kontrol transaksi.

Sesuai dengan uraian yang telah dijelaskan, berikut pada Gambar 2.2 menunjukkan diagram arsitektur dari sistem yang dirancang.

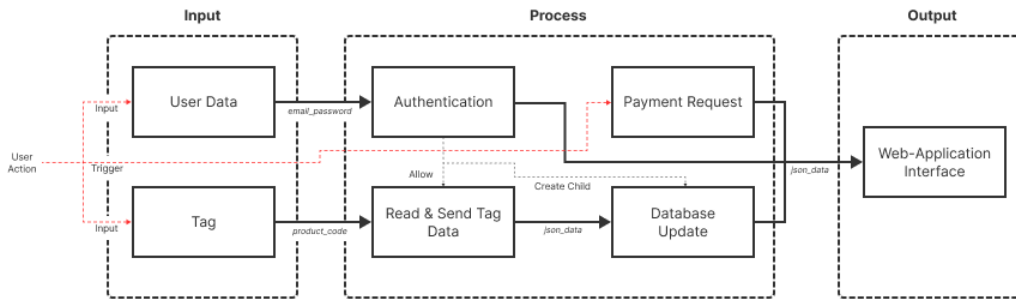


Gambar 2.2 Diagram Arsitektur Sistem

Proses transaksi dimulai pada *Web-Application* dengan pengguna melakukan otentikasi akun sebelum melakukan transaksi. Setelah otentikasi berhasil, pengguna akan dialihkan ke halaman transaksi. Sistem secara otomatis akan membuat ruang pada *database* untuk menjadi

ruang data transaksi dan memberi informasi ke *reader* bahwa terdapat proses transaksi beserta alamat ruang transaksinya. Setelah itu, pengguna dapat memulai melakukan *scanning* barang menggunakan *reader* dan data dari *tag* pada barang yang dibaca oleh *reader* akan dikirim ke *database*. *Web-Application* akan *me-listen* perubahan pada ruang transaksi di *database* dan merespon segala aksi yang dilakukan pengguna dengan menambahkan informasi mengenai barang didapat pada antarmuka. Saat pengguna selesai melakukan proses *scanning* maupun aksi lain, pengguna dapat *men-trigger* proses pembayaran pada antarmuka *Web-Application*.

Sesuai dengan alur proses yang telah dipaparkan, Gambar 2.3 berikut menunjukkan diagram input-proses-output dari sistem belanja *self-service*.



Gambar 2.3 Diagram Input-Proses-Output

2.2 Rancangan Perangkat Keras

Sesuai yang ditunjukkan pada Gambar 2.3 dan Gambar 2.1, sistem akan memiliki perangkat *hardware* berupa *reader*/pembaca untuk kedua *tag* yang digunakan untuk melakukan proses *read & send data* menuju *database*. Terdapat 4 perangkat *reader* yang akan digunakan pada penelitian ini yaitu :

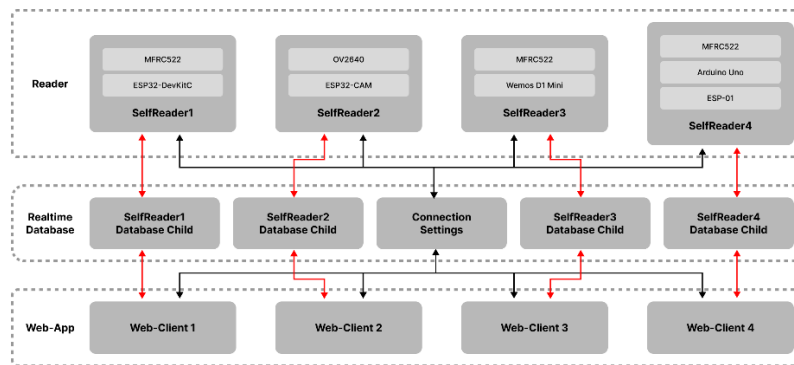
1. Reader RFID yang terdiri MFRC522 dan ESP32-DevKitC (*SelfReader1*).
2. Reader QR-Code yang terdiri dari OV2640 dan ESP32-CAM (*SelfReader2*).
3. Reader RFID yang terdiri MFRC522 dan Wemos D1 Mini (*SelfReader3*).
4. Reader RFID yang terdiri MFRC522, Arduino Uno dan ESP-01 (*SelfReader4*).

Berdasarkan daftar *reader* diatas, akan digunakan 1 *reader QR-code* dan 3 *reader RFID*. Penggunaan 3 *reader RFID* ini bertujuan untuk mengetahui pengaruh perbedaan mikrokontroler terhadap pembacaan dari MFRC522. Hal tersebut tidak dapat dilakukan pada OV2640 karena mikrokontroler lain tidak memiliki kompatibilitas dengan OV2640. Sebaliknya ESP32-CAM juga tidak mampu untuk terhubung dengan MFRC522 karena tidak memiliki jumlah pin yang cukup.

2.3 Rancangan Perangkat Lunak

2.3.1 Database

Firestore adalah *Backend-as-a-Service* (BaaS) yang dikembangkan oleh Google. Firestore Realtime Database merupakan salah satu layanan yang diberikan Firestore berupa *NoSQL- Database* yang menyimpan data dalam bentuk JSON dan memungkinkan untuk melakukan sinkronisasi secara *real-time*. Selain untuk menyimpan data transaksi, Firestore RTDB pada penelitian ini juga digunakan untuk mengatur koneksi antara *Reader* dan *Web-Client*. *Web-Client* dapat memilih *reader* yang tersedia untuk digunakan. Hal tersebut dibuat agar 4 *reader* dapat digunakan secara bergantian pada satu *Web-Client* tanpa mempengaruhi *reader* lain bahkan dapat digunakan dengan adanya beberapa *Web-Client*. Berikut pada merupakan diagram blok komunikasi sistem.



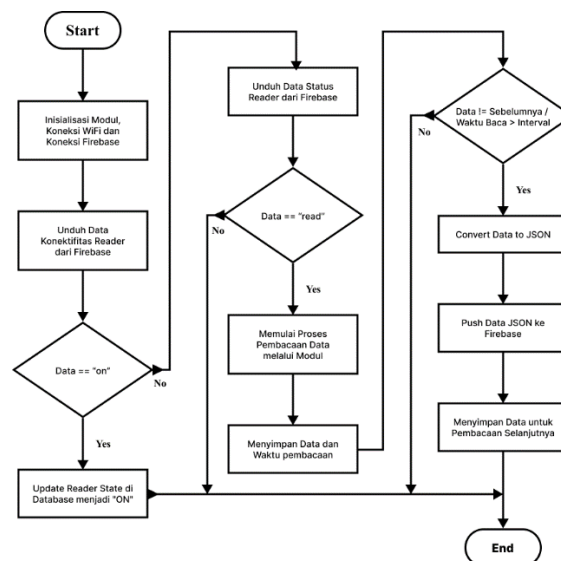
Gambar 2.4 Diagram Komunikasi Sistem

2.3.2 Algoritma Reader Pada Saat Transaksi

Reader bekerja dengan diawali melakukan inisialisasi WiFi, Firebase, variabel dan modul yang digunakan. Kemudian *reader* akan mengambil/mengunduh data dari *database* yang berisi data yang mengindikasikan bahwa *reader* dalam keadaan aktif/*on* ketika data yang didapat bukan “*on*” maka *reader* akan meng-*update* data menjadi “*on*”.

Setelah itu *reader* akan *me-listen* data dari *database* yang mengindikasikan status dari *reader*. Terdapat 2 status yang dapat dibaca disini yaitu *off* dan *read*. Status *off* berarti *reader* dalam keadaan *idle* sedangkan status *read* berarti *reader* diminta untuk melakukan proses *read & send data*. Saat terdapat *client* yang tersambung dan akan melakukan transaksi yang ditandai dengan status berubah menjadi *read*, *reader* akan menerima hasil pembacaan/*scanning data* dari MFRC522/OV2640. Setelah data didapat, akan diverifikasi terlebih dahulu apakah data yang didapat tidak sama dengan data pembacaan sebelumnya atau waktu pembacaan melebihi interval yang ditentukan yaitu 1000 ms agar data dapat melanjutkan proses pengiriman ke *database*.

Setelah lolos verifikasi, mikrokontroller akan mengambil alamat ruang transaksi, mengubah data pembacaan ke dalam bentuk JSON lalu mengirim data JSON ke ruang transaksi serta menyimpan data pembacaan untuk verifikasi selanjutnya. Proses ini akan diulang sampai status *reader* kembali menjadi *off* yang mengindikasikan transaksi telah berhenti. Berikut pada Gambar 2.5 menunjukkan diagram alir dari *reader*.

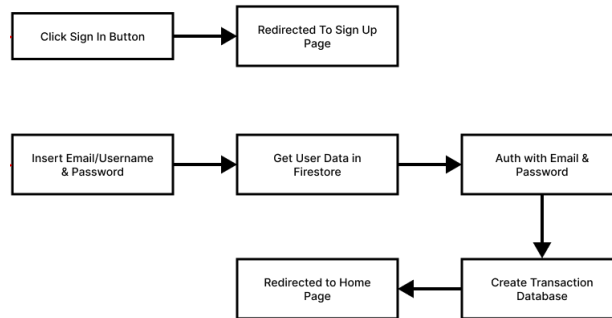


Gambar 2.5 Diagram Alir Reader

2.3.3 Web-Application

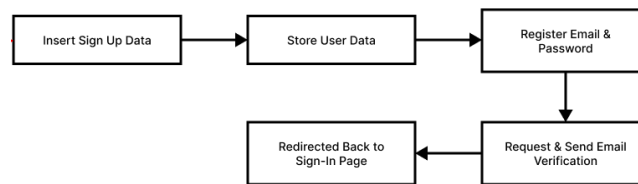
Pada penelitian ini akan dibuat Web-Application sebagai antarmuka pusat kontrol saat melakukan transaksi. Web-Application dibuat menggunakan Flutter SDK dan nantinya akan di-deploy melalui Firebase Hosting. *Web-Application* yang akan dibuat terdiri dari 3 halaman yaitu *Sign-In*, *Sign-Up*, dan Transaksi (*Home*).

Sign-in Page merupakan halaman untuk melakukan otentikasi akun sebelum melakukan transaksi. Pengguna dapat menuju halaman *Sign-Up* jika belum memiliki akun dengan mengklik tombol yang disediakan. Berikut pada Gambar 2.6 merupakan diagram proses pada *Sign-in Page*.



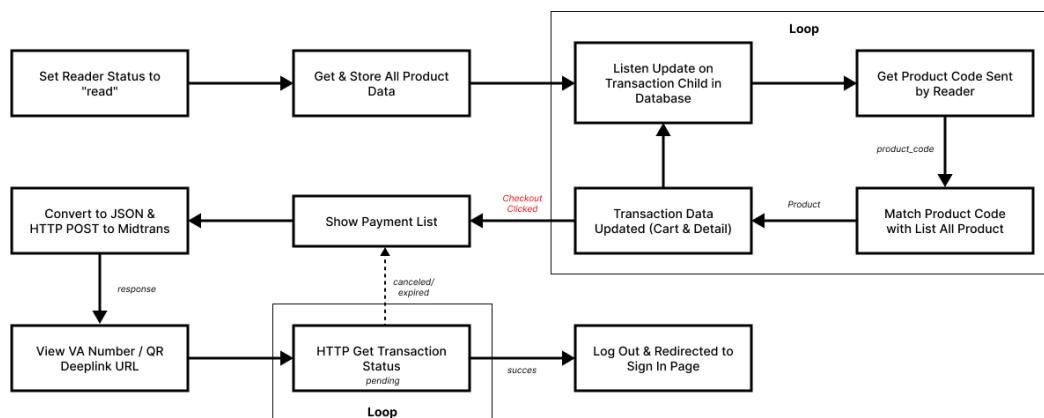
Gambar 2.6 Diagram Proses pada Sign-in Page

Sign-up Page merupakan halaman untuk melakukan pendaftaran akun apabila pengguna belum memiliki akun. Berikut Gambar 2.7 menunjukkan diagram proses pada *Sign-up Page*.



Gambar 2.7 Diagram Proses pada Sign-up Page

Home Page merupakan halaman untuk melakukan transaksi dan pembayaran, terdiri dari tabel yang menunjukkan detail dari barang yang dibeli dan detail dari transaksi seperti informasi *user*, total harga, jumlah barang, dll. Berikut pada Gambar 2.8 merupakan diagram yang menunjukkan proses yang terjadi pada *Home Page*.

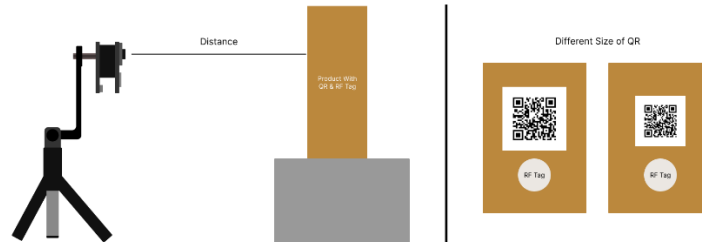


Gambar 2.8 Diagram Proses pada Home Page

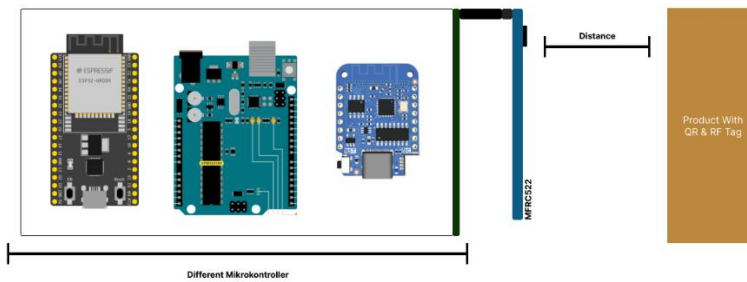
2.4 Rancangan Pengujian

2.4.1 Rancangan Pengujian Pembacaan Data Reader

Pengujian ini dilakukan bertujuan untuk mendapatkan hasil mengenai jumlah pembacaan data valid yang dapat dicapai *reader* dalam waktu 10 detik, jarak efektif, ukuran QR Code (2-4 cm) yang dapat digunakan dan pengaruh mikrokontroler terhadap pembacaan data RFID. Pengujian ini akan dilakukan sebanyak 10 kali. Berikut pada Gambar 2.9 dan Gambar 2.10 menunjukkan skema dari pengujian pembacaan data masing-masing identifikasi.

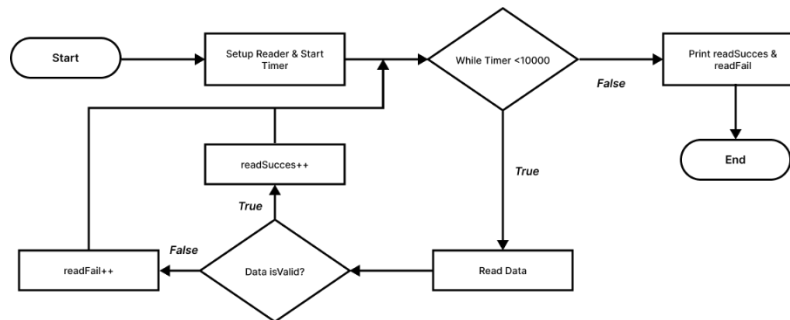


Gambar 2.9 Skema Pengujian Pembacaan QR-Code



Gambar 2.10 Skema Pengujian Pembacaan RFID

Berdasarkan penjelasan diatas, dapat diketahui bahwa *reader* memiliki alur/algorithm yang berbeda dengan alur pada subbab 2.3.2. Berikut pada Gambar 2.11 menunjukkan diagram alir dari algoritma reader pada pengujian pembacaan data ini.

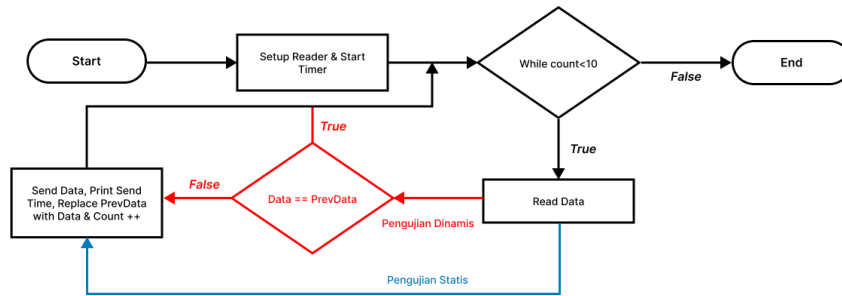


Gambar 2.11 Diagram Alir Pengujian Pembacaan Data

2.4.2 Rancangan Pengujian Pembacaan dan Pengiriman Data ke Firebase

Proses pengujian ini dibagi menjadi dua yaitu dengan membaca 10 data yang sama (statis) dan 10 data yang berbeda (dinamis) lalu mengirimkannya ke *database* dan mencatat waktu penyelesaian pengiriman. Tujuan dari pengujian ini adalah memperoleh informasi mengenai waktu respons yang dihasilkan oleh *reader* dalam menyelesaikan proses pembacaan dan pengiriman data ketika menggunakan barang yang sama maupun berbeda-beda. Pengujian ini akan dilakukan sebanyak 10 kali. Dan sama seperti pengujian sebelumnya, pada pengujian

ini *reader* memiliki alur/algorithm yang berbeda dengan alur pada subbab 2.3.2. Berikut pada menunjukkan diagram alir dari algoritma *reader* pada pengujian pembacaan data ini.



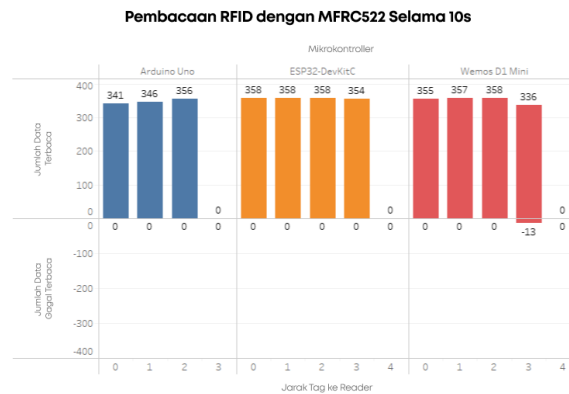
Gambar 2.12 Diagram Alir Pengujian Pembacaan dan Pengiriman Data ke Firebase

3. HASIL DAN PEMBAHASAN

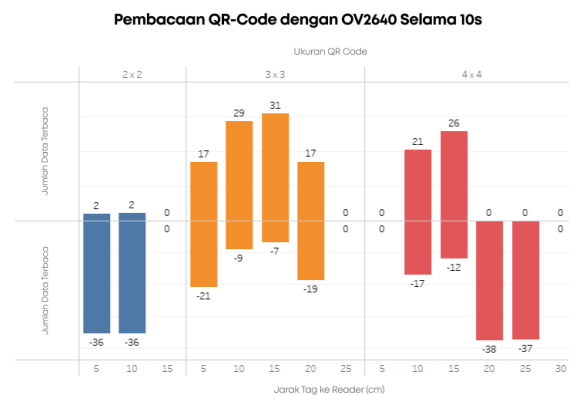
3.1 Pengujian Reader

3.1.1 Pengujian Pembacaan Data

Pengujian ini dilakukan melalui pengiriman perintah ke *reader* untuk melakukan pembacaan data secara terus-menerus selama 10 detik dengan tag dalam posisi tetap dan tidak berubah-ubah. Hasil pengujian ini dapat dilihat pada Gambar 3.1 dan Gambar 3.2.



Gambar 3.1 Hasil Pengujian Pembacaan RFID dengan MFRC522 Selama 10s



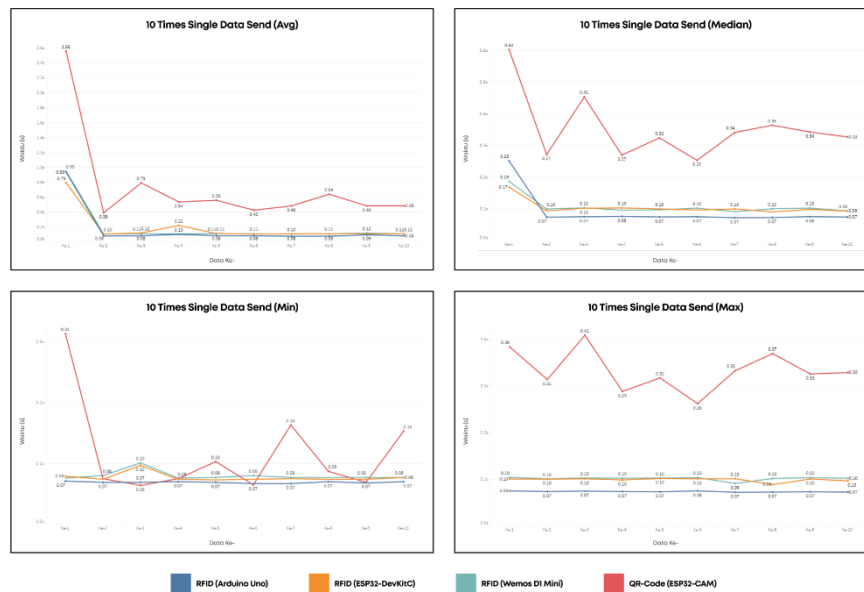
Gambar 3.2 Hasil Pengujian Pembacaan QR-Code dengan OV2640 Selama 10s

Gambar 3.1 menunjukkan bahwa pembacaan RFID dengan MFRC522 memiliki jarak efektif dalam pembacaan data yang terbatas pada rentang 0 hingga 3 cm. Dalam periode pembacaan selama 10 s tersebut, pembacaan RFID dengan MFRC522 dapat membaca data sebanyak ± 350 . Sempitnya jarak pembacaan efektif ini dikarenakan dua hal. Pertama keterbatasan MFRC522 yang secara spesifikasi memang memiliki jarak baca yang terbatas yaitu ± 5 cm. Kedua keterbatasan protokol ISO/IEC 14443 yang diatur untuk dapat membaca dengan jarak maksimal 10 cm dari antenna *reader*.

Sedangkan untuk hasil pengujian pembacaan QR Code menggunakan OV2640 pada Gambar 3.2 menunjukkan bahwa pembacaan data paling baik dilakukan menggunakan ukuran tag QR 3x3cm dengan jarak efektif yang lebih luas, yaitu antara 5 hingga 20 cm. Hasil pengujian menunjukkan rata-rata dalam 10 detik modul dapat membaca data sebanyak ± 37 kali dengan kegagalan yang cukup tinggi. Keterbatasan pembacaan QR ini disebabkan oleh OV2640 yang tidak memiliki kemampuan auto-focus dan *infrared* sehingga tidak terlalu mendukung untuk aplikasi pembacaan QR seperti ini dikarenakan tag yang terlalu besar, terlalu kecil, terlalu dekat, ataupun terlalu jauh dapat menyebabkan pembacaan oleh OV2640 terganggu..

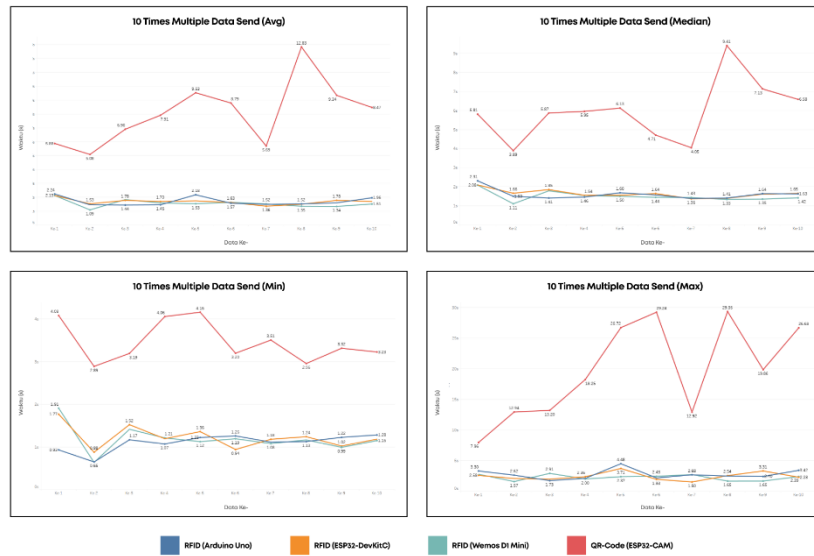
3.1.2 Pengujian Pembacaan dan Pengiriman Data ke Firebase

Proses pengujian ini dibagi menjadi dua yaitu dengan membaca 10 data yang sama (statis) dan berbeda (dinamis) lalu mengirimkannya ke *database* dan mencatat waktu penyelesaian pengiriman. Hasil pengujian ini dapat dilihat pada Gambar 3.3 dan Gambar 3.4.



Gambar 3.3 Hasil Pengujian Pembacaan dan Pengiriman Data Barang Statis ke Firebase

Hasil pengujian pada Gambar 3.3 menunjukkan bahwa dalam pembacaan statis (pembacaan barang barang yang sama) pembacaan RFID dengan MFRC522 secara rata-rata memiliki kecepatan *read & send data* 3-8 kali lebih cepat daripada pembacaan QR Code dengan OV2640 meski memiliki kecepatan tercepat yang cukup sejajar. Pembacaan RFID dengan MFRC522 pada ketiga mikrokontroller memiliki kecepatan *read & send data* yang cukup stabil mulai pada data ke-2 di kisaran 100ms dibandingkan pembacaan QR Code dengan OV2640 yang kurang stabil dengan perbedaan nilai minimum dan maksimum mencapai 700 ms pada data kedua.

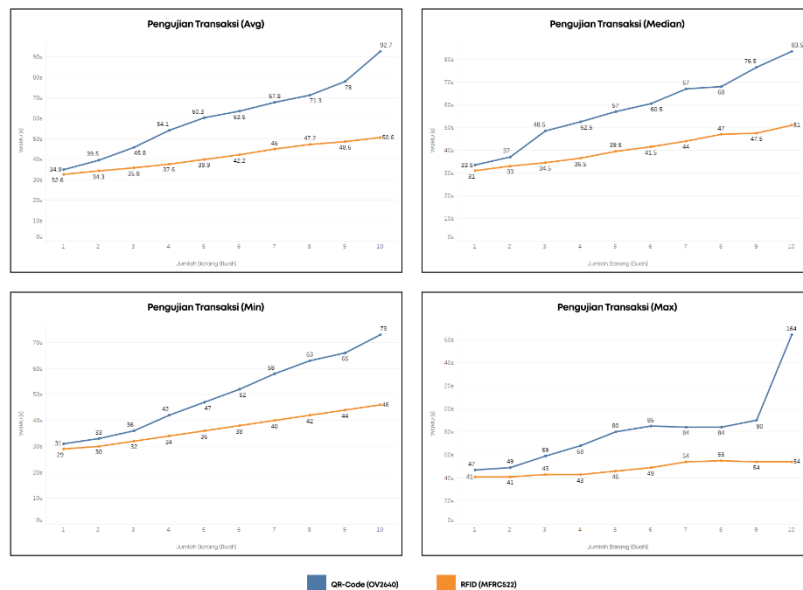


Gambar 3.4 Hasil Pengujian Pembacaan dan Pengiriman Data Barang Dinamis ke Firebase

Hasil pengujian pada Gambar 3.4 menunjukkan bahwa dalam kondisi dinamis (adanya pergantian barang yang dibaca) pembacaan RFID dengan MFRC522 memiliki waktu respon lima kali lebih cepat daripada QR Code dengan OV2640. Pembacaan RFID dengan MFRC522 pada ketiga mikrokontroller memiliki waktu *read & send data* yang cukup stabil dibandingkan pembacaan QR Code dengan OV2640. Hal ini mengindikasikan bahwa pembacaan QR-Code dengan OV2640, selain membutuhkan waktu yang lebih lama juga memiliki kekurangan dalam membaca barang/tag yang berbeda.

3.2 Pengujian Transaksi

Pengujian transaksi bertujuan untuk mencatat data terkait waktu yang dibutuhkan oleh pengguna untuk menyelesaikan transaksi dengan jumlah barang tertentu menggunakan kedua identifikasi yang diuji yaitu identifikasi QR-Code dengan OV2640 dan identifikasi RFID dengan MFRC522. Hasil dari Pengujian ditunjukkan pada Gambar 3.5 .



Gambar 3.5 Hasil Pengujian Transaksi

Hasil pengujian menunjukkan bahwa identifikasi RFID menggunakan MFRC522 memiliki kecepatan penyelesaian transaksi 7-63% lebih cepat dibandingkan identifikasi QR-Code menggunakan OV2640. Data yang didapat menunjukkan bahwa proses identifikasi barang menggunakan RFID dengan MFRC522 memiliki pengaruh yang signifikan dalam mempercepat waktu yang dibutuhkan pengguna untuk menyelesaikan transaksi dalam sistem belanja *self-service*. Hal ini disebabkan oleh identifikasi RFID dengan MFRC522 secara kemampuan memiliki kecepatan pembacaan data dan jeda antar barang yang lebih dibandingkan dengan identifikasi QR-Code dengan OV2640 seperti yang dijelaskan pada Subbab 3.1.1.

4. KESIMPULAN

Berdasarkan pengujian dan hasil yang diperoleh terkait pengaruh antara identifikasi QR-Code dengan OV2640 dan RFID dengan MFRC522 pada sistem belanja *self-service* berbasis *Web-Application* yang terintegrasi dengan *Firestore Database* dan *Midtrans Payment Gateway*, dapat diambil kesimpulan bahwa identifikasi MFRC522 memiliki kecepatan dan keberhasilan pembacaan yang tinggi dengan jarak pembacaan efektif yang lebih rendah dibandingkan dengan identifikasi OV2640 yang berpengaruh pada kecepatan penyelesaian transaksi hingga 63% lebih cepat pada transaksi 10 buah barang.

5. SARAN

Saran untuk yang dapat dilakukan adalah melakukan penelitian lebih lanjut dengan menggunakan modul/reader yang memiliki spesifikasi lebih baik untuk memaksimalkan keuntungan setiap identifikasi. Seperti menggunakan GM67 untuk pembacaan QR-Code atau RMT-01 untuk pembacaan RFID.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistika Indonesia, *Pengeluaran untuk Konsumsi Penduduk Indonesia*, vol. 27, no. 2. 2023.
- [2] P. Pallavi, B. Raviteja, A. Varun, and R.Narender, "Smart shopping cart," *Int. J. Adv. Sci. Technol.*, vol. 28, no. 17, pp. 91–97, 2019, doi: 10.15864/ajec.1104.
- [3] A. Narayan, A. Krishnan, and A. S. Ponraj, "IOT Based Comprehensive Retail Malpractice Detection and Payment System*," *Proc. CONECCT 2020 - 6th IEEE Int. Conf. Electron. Comput. Commun. Technol.*, 2020, doi: 10.1109/CONECCT50063.2020.9198519.
- [4] R. Ranjith and S. Naveen, "Smart Shopping Trolley using QR Code and ESP32Cam," *13th Int. Conf. Adv. Comput. Control. Telecommun. Technol. ACT 2022*, vol. 8, pp. 360–365, 2022.
- [5] G. Aguzman, C. H. Pangaribuan, and A. H. Sasongko, "Self-Service Checkouts: The Role Of Service Innovation In Indonesian Convenience Stores," *Psychol. Educ.*, vol. 57, no. 9, pp. 81–87, 2020, [Online]. Available: www.psychologyandeducation.net.
- [6] A. Dina Marsela, J. Nathanael, and N. Marchelyta, "Penggunaan E-Wallet sebagai Kemajuan Teknologi Digital dalam Menentukan Preferensi Masyarakat di Surabaya," *Pros. Semin. Nas. Ilmu Sos.*, pp. 784–790, 2022.
- [7] M. S. Saini, S. Ratre, S. S. Reddy, D. Venkata, V. Vyas, and E. Engineering, "AUTOMATED BILLING SYSTEM SMART TROLLEY USING ESP32 CAM MODULE WITH THE HELP OF ARDUINO UNO," vol. 9, no. 4, 2022.
- [8] M. Shahroz, M. F. Mushtaq, M. Ahmad, S. Ullah, A. Mehmood, and G. S. Choi, "IoT-Based Smart Shopping Cart Using Radio Frequency Identification," *IEEE Access*, vol. 8,

-
- pp. 68426–68438, 2020, doi: 10.1109/ACCESS.2020.2986681.
- [9] T. K. Das, A. K. Tripathy, and K. Srinivasan, “A Smart Trolley for Smart Shopping,” *2020 Int. Conf. Syst. Comput. Autom. Networking, ICSCAN 2020*, 2020, doi: 10.1109/ICSCAN49426.2020.9262350.
- [10] M. HARA, “Development and popularization of QR code,” *Synth. English Ed.*, vol. 12, no. 1, pp. 19–28, 2019, doi: 10.5571/syntheng.12.1_19.
- [11] T. Lotlikar, R. Kankapurkar, A. Parekar, and A. Mohite, “Comparative study of Barcode, QR-code and RFID System.” *International Journal of Circuit Theory and Application*, p. 5, 2013.