

## INTEGRASI SISTEM TRANSAKSI PADA VENDING MACHINE MENGGUNAKAN CLOUD MQTT

Yohanes Dewa Bayu Adyawadhana <sup>1)</sup> Harianto <sup>2)</sup> Ira Puspasari <sup>3)</sup>

Program Studi/Jurusan Teknik Komputer  
Universitas Dinamika.

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1) [dewabayu27@gmail.com](mailto:dewabayu27@gmail.com), 2) [hari@dinamika.ac.id](mailto:hari@dinamika.ac.id), 3) [ira@dinamika.ac.id](mailto:ira@dinamika.ac.id)

**Abstrak:** *Cashless* adalah salah satu metode yang digunakan untuk melakukan pekerjaan transaksi jual beli. pekerjaan tersebut sangat dipermudah dengan adanya teknologi *cashless*. Selain *cashless* ada pula teknologi yang mempermudah pekerjaan transaksi jual beli yaitu menggunakan sebuah mesin bernama *vending machine*. *Vending Machine* diciptakan untuk menjual barang-barang mudah seperti makanan dan minuman tanpa diperlukannya seorang penjual untuk menjaga barang dagangan. *Vending Machine* memiliki fungsi yang praktis, namun di Indonesia mesin ini kurang diminati karena banyak dari *vending machine* di Indonesia masih menggunakan koin dan uang kertas sebagai sarana transaksinya, sehingga perkembangannya tidak seanggih mesin yang ada di negara-negara Asia lainnya. Melihat kurangnya peminat pada *vending machine* di Indonesia, maka pada penelitian ini dibuat sistem pengiriman data *vending machine* menggunakan protokol MQTT sebagai sarana yang membuat data transaksi *vending machine* menjadi terhubung ke internet dan kemudian ditampilkan ke aplikasi android. Aplikasi android sendiri digunakan untuk memantau data yang telah di dihubungkan ke database melalui protokol MQTT, sehingga riwayat transaksi dapat dipantau. Riwayat transaksi yang dapat dipantau sangat membantu pihak penjual dalam perhitungan produk. Dengan alat ini pihak pembeli tidak perlu membawa uang tunai karena *vending machine* dilengkapi dengan metode *cashless* menggunakan RFID. Pembeli cukup membawa E-KTP sebagai sarana transaksi, karena E-KTP sudah mengandung RFID yang mampu dibaca oleh *vending machine*. Hasil dari pengujian yang telah dilakukan adalah MQTT mampu mengirimkan data berupa data *character*, dari Modul Wifi data dapat diterima oleh Broker MQTT dengan baik, kemudian data pada Broker MQTT dapat terkirim dengan baik ke Database, sehingga data dapat tersimpan sebagai data riwayat transaksi. Hasil pengiriman ke masing-masing poin dapat terkirim dengan baik dengan prosentase keberhasilan pengiriman sebesar 100%. Database yang digunakan adalah Database lokal, sehingga dibutuhkan koneksi IP *Public* untuk membuat Database menjadi terkoneksi ke Internet.

**Kata kunci:** Broker MQTT, Cashless, IOT, Vending Machine

### PENDAHULUAN

Perkembangan teknologi di Indonesia saat ini telah memberikan banyak manfaat, salah satunya untuk mempermudah pekerjaan masyarakat. Mulai dari pekerjaan yang ringan hingga pekerjaan berat, salah satu pekerjaan yang dipermudah dengan adanya teknologi adalah proses jual beli barang, saat ini pekerjaan tersebut sangat dipermudah dengan adanya teknologi yang bernama *cashless*. Selain *cashless* yakni menggunakan sebuah mesin. Mesin tersebut diciptakan untuk menjual barang-barang mudah

seperti makanan dan minuman tanpa diperlukannya seorang penjual untuk menjaga barang dagangan tersebut, mesin itu disebut *vending machine*. *Vending machine* di Indonesia tidak terlalu diminati, sehingga perkembangannya tidak seanggih negara-negara Asia lainnya. Pada penelitian sebelumnya (Adiputra, 2015) telah dibuat sebuah mesin penjual *softdrink* otomatis atau *vending machine* yang menjual produk *softdrink*. Penelitian ini masih menggunakan uang koin dan uang kertas sebagai metode transaksinya. Sedangkan pada penelitian (Nurrochman, 2019), telah dibuat sebuah *vending machine* minuman

berkarbonasi yang mana inputan dari penelitian ini menggunakan E-KTP sebagai data identitas pembelinya. Namun pada penelitian ini proses transaksinya juga masih menggunakan uang koin.

Menurut (Kompas, 2017) Indonesia sangat tertinggal dalam hal penggunaan *vending machine* jika dibandingkan dengan Jepang dan Singapura. Di negara Jepang penggunaan 1 unit *vending machine* digunakan untuk 23 orang, dengan jumlah total mencapai 5 juta unit. Begitu juga di Singapura penggunaan 1 unit *vending machine* digunakan untuk 1000 orang, dengan perkiraan sekitar 15 ribu unit untuk melayani 15 juta orang, sedangkan di Indonesia hanya ada total sekitar 4 ribu unit yang digunakan untuk lebih dari 250 juta penduduk, dengan jumlah yang terbilang sedikit pun masyarakat Indonesia masih banyak yang tidak tertarik dengan penggunaan *vending machine*. Hal itu dapat terjadi karena teknologi *vending machine* di Indonesia masih susah untuk digunakan, contohnya masih menggunakan uang koin maupun uang kertas untuk pembayarannya, sehingga masyarakat lebih memilih berbelanja di toko biasa.

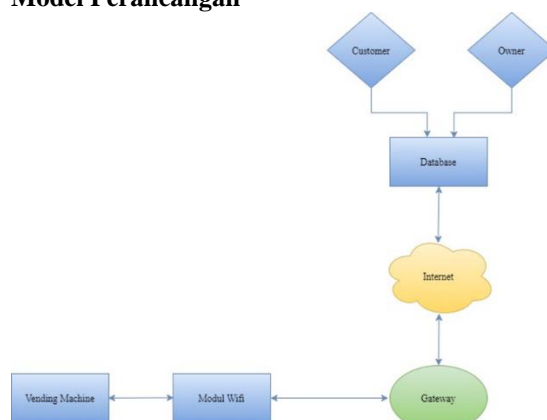
Melihat kurangnya minat pengguna *vending machine* di Indonesia, maka dibutuhkan suatu inovasi yang dapat menarik peminat, selain untuk menarik peminat dari pengguna *vending machine* tentu saja hal utamanya adalah untuk menjamin proses transaksi serta dapat menyimpan riwayat transaksi agar proses transaksi dapat dikelola dengan lebih mudah lagi.

Pada penelitian ini dibuat sistem pengiriman data *vending machine* menggunakan protokol MQTT sebagai sarana komunikasi ke jaringan internet, dimana protokol MQTT sendiri dirancang untuk komunikasi antar perangkat berdaya rendah, selain itu pada penelitian (Safii, 2019), protokol MQTT juga mampu mengirimkan data lebih cepat melampaui kecepatan protokol HTTP dengan persentase 5-10%. Hal ini membuat pengiriman data transaksi *vending machine* semakin baik karena pengiriman data bersifat *realtime*, sehingga semakin cepat pengiriman datanya, maka semakin baik pula proses transaksinya. Protokol MQTT mengirimkan data ke Database, database juga terhubung ke internet yang kemudian ditampilkan ke aplikasi android. Aplikasi android sendiri digunakan untuk memantau data yang telah di dihubungkan ke database melalui protokol MQTT tersebut, sehingga riwayat transaksi dapat dipantau yang mana sangat membantu pihak penjual dalam perhitungan produk. Dengan alat ini pihak pembeli juga tidak perlu membawa uang tunai karena *vending machine* ini dilengkapi dengan metode

cashless menggunakan RFID. Pembeli cukup membawa E-KTP sebagai identitasnya, karena E-KTP sendiri sudah mengandung RFID yang mana mampu dibaca oleh *vending machine* ini. RFID *reader* pada *vending machine* membaca UID (*User Identity*) dari E-KTP yang kemudian disimpan di database menggunakan server MQTT. Data yang disimpan di database bukan hanya UID saja, melainkan data pembeli yang meliputi: UID, saldo pembeli, serta data riwayat pembelian.

## METODE PENELITIAN

### Model Perancangan



Gambar 1. Blok diagram model perancangan

Gambar 1 merupakan model perancangan dari sistem pengiriman data *vending machine* yang dibagi menjadi beberapa bagian, berikut adalah penjelasan setiap bagian dari diagram di atas.

#### a. *Vending Machine*

Proses ini merupakan proses pengaplikasian dari alat *vending machine* yang berjalan jika pembeli sudah menginputkan suatu ID yang telah terdaftar pada alat, jika pembeli sudah menginputkan ID yang telah terdaftar pada alat, maka alat dapat melanjutkan proses transaksi. Transaksi yang dimaksud adalah mengeluarkan minuman ketika mendapatkan aksi dari sebuah tombol yang terdapat pada alat. Selain untuk mengeluarkan makanan, proses ini juga mengirimkan data berupa nilai dari ID pembeli serta aksi yang telah dilakukan oleh alat, yang kemudian dikirim ke proses selanjutnya yaitu proses Modul Wifi.

#### b. *Modul Wifi*

Pada tahap ini, data pada alat vending machine dikategorikan ke beberapa topik berbeda dengan tujuan sebagai pembeda antar nilai. Selain membagi data ke beberapa topik berbeda, Modul Wifi juga bertugas sebagai

jembatan penghubung antara alat vending machine dengan Broker MQTT yang mana Modul Wifi mengirimkan data yang telah dibagi menjadi beberapa topik kepada Broker MQTT.

**c. Gateway**

Dalam penelitian ini Gateway memiliki 2 peran yaitu sebagai Broker MQTT dan juga sekaligus sebagai Subscriber. Peran sebagai Broker MQTT, Gateway menerima data dari Modul Wifi, data yang diterima berupa topik beserta *message* yaitu isi dari bahasan topik tersebut yang ditampilkan sesuai dengan topik yang telah dibagi pada tahap Modul Wifi. Peran kedua sebagai Subscriber, Gateway mengambil data dari Broker MQTT yang kemudian data dikirimkan ke Database melalui jaringan internet.

**d. Internet**

Pada sistem penelitian ini internet hanya digunakan sebagai sarana penghubung antar Gateway yang berperan sebagai Subscriber dengan Database. Dengan adanya koneksi internet, data dapat dipantau secara online yang mana nantinya penjual maupun pembeli dapat mengakses dari mana saja.

**e. Database**

Tahap Database yaitu tahap penyimpanan dari penerimaan maupun pengiriman data. Jadi seluruh akses data masuk kedalam proses Database ini yang mana membuat proses ini sebagai proses riwayat data. Pada tahap ini Database menerima data dari Gateway yang berperan sebagai Subscriber.

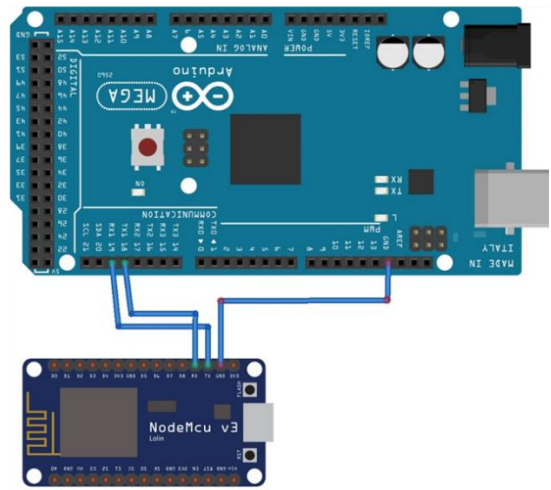
**f. Customer**

Tahap Customer adalah tahap yang dilakukan oleh seorang pembeli, yang mana pembeli dapat melakukan proses registrasi dan histori melalui aplikasi android. Pada proses registrasi, pembeli melakukan pengisian data yang bertujuan sebagai data pengenalan bagi UID dari E-KTP yang digunakan, sedangkan pada proses histori, pembeli dapat melakukan pemeriksaan dari riwayat transaksi yang telah dilakukan, mulai dari barang yang dibeli, hingga sisa saldo.

**g. Owner**

Tahap Owner adalah tahap yang dilakukan oleh seorang pemilik atau penjual, yang mana penjual juga mampu melakukan 2 proses yaitu proses periksa error dan histori. Pada proses periksa error, penjual menerima peringatan jika terjadi masalah dengan *vending machine*, sehingga penjual dapat melakukan tindakan perbaikan secepatnya, sedangkan pada proses

histori, penjual juga dapat melakukan pemeriksaan dari riwayat transaksi yang sudah dilakukan seperti waktu transaksi, barang, hingga akumulasi dari saldo pembeli.

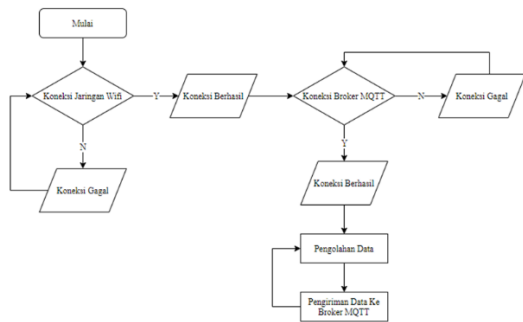


Gambar 2. Rangkaian Skematik Modul Wifi

Rangkaian pada gambar 2 merupakan rangkaian yang digunakan pada penelitian ini. Arduino Mega adalah mikrokontroler yang digunakan pada alat *vending machine*, dan kemudian mengirimkan data *character* kepada mikrokontroler NodeMcu. NodeMcu disini berfungsi sebagai modul wifi yang melanjutkan data kepada Broker MQTT. Koneksi port yang dibutuhkan adalah menyambungkan port TX pada Arduino Mega menuju port RX pada NodeMcu, hal ini bertujuan agar Arduino Mega dapat mengirim data ke NodeMcu, sedangkan NodeMcu dapat menerima data dari Arduino Mega. Koneksi setelah itu adalah menyambungkan port RX pada Arduino Mega menuju port TX pada NodeMcu, hal ini bertujuan sama seperti koneksi port sebelumnya, namun bersifat kebalikkannya yaitu agar Arduino Mega dapat menerima data yang dikirim oleh NodeMcu. Koneksi yang ketiga adalah menyambungkan port GND pada kedua mikrokontroler.

**Perancangan Perangkat Lunak**

**a. Algoritma Modul Wifi**



Gambar 3. Algoritma Modul Wifi

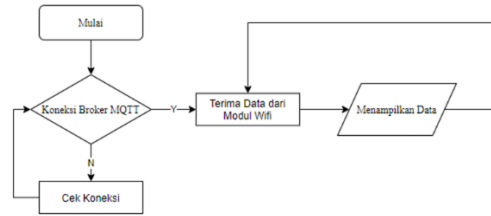
Sistem ini dimulai dari koneksi antara modul wifi dan jaringan wifi dengan mendeklarasikan SSID dan PASSWORD dari jaringan wifi yang saat ini digunakan, hal ini dilakukan agar perangkat modul wifi dapat terhubung ke jaringan internet. Apabila koneksi berhasil, maka sistem menampilkan teks bahwa koneksi telah berhasil, namun jika koneksi gagal sistem terus menerus menampilkan teks yang menyatakan bahwa sistem sedang menghubungkan ke jaringan wifi.

Setelah koneksi modul wifi ke jaringan wifi berhasil, maka masuk pada proses koneksi ke broker MQTT. Pada proses koneksi ke broker MQTT diperlukan beberapa data yaitu alamat server, alamat port, user, dan password yang berada pada broker MQTT. Jika ke 4 data tersebut telah sesuai dengan data yang berada pada broker MQTT, maka koneksi pasti berhasil.

Setelah koneksi modul wifi ke broker MQTT berhasil, maka masuk pada proses pengolahan data. Pada proses ini dilakukan pembagian data berdasarkan topik, topik yang digunakan ada 2 topik yaitu topik UID dan saldo, kedua topik inilah yang digunakan pada algoritma aplikasi android.

Setelah pembagian data selesai, maka masuk proses pengiriman data ke broker MQTT dimana data yang dikirim berbentuk data character. Setelah mengirim data ke broker MQTT, maka proses selanjutnya adalah kembali ke proses pengolahan data yang mana membuat data sebelumnya terganti oleh data yang baru, sehingga data yang terkirim ke broker MQTT terus terupdate dengan tujuan agar data berjalan secara realtime.

**b. Algoritma Broker MQTT**



Gambar 4. Algoritma Broker MQTT

Sistem ini dimulai dengan proses awal yaitu periksa koneksi antar modul wifi dan broker MQTT, jika koneksi gagal, maka broker MQTT tidak menerima data apapun, sehingga data yang ditampilkan adalah kosong, sedangkan jika koneksi berhasil, maka broker MQTT menerima data dari modul wifi. Penerimaan data dari modul wifi berupa data *character* seperti yang telah dijelaskan pada algoritma modul wifi, setelah menerima data tersebut broker MQTT menampilkan data secara berurutan dari data yang lama hingga data terbaru.

**c. Algoritma Database**

Dalam beberapa penjelasan sebelumnya telah dijelaskan bahwa fungsi dari database yaitu sebagai sarana penyimpanan data, oleh sebab itu dalam sistem database dibuat 2 tabel yang membedakan jenis data. Berikut adalah 2 tabel yang terdapat pada sistem database:

1. Tabel Daftar Pembeli

Pada tabel ini disimpan data dari pembeli yang telah mendaftar. Fungsi dari tabel ini adalah untuk memastikan kode UID dari E-KTP telah terdaftar, sehingga pembeli dapat melanjutkan proses transaksi. Berikut adalah contoh dari tabel daftar pembeli:

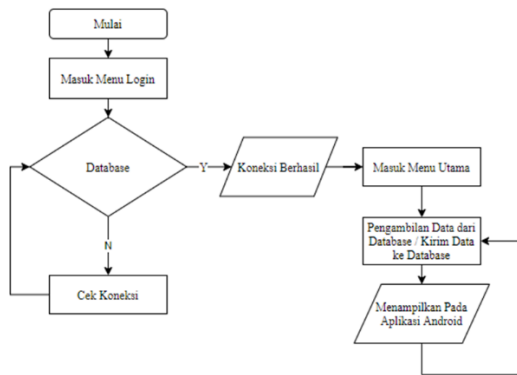
UID	NAMA	SALDO

2. Tabel Riwayat Transaksi

Tabel terakhir adalah tabel riwayat transaksi yang mana berfungsi untuk memantau proses transaksi yang telah dilakukan pembeli. Berikut adalah contoh dari tabel riwayat transaksi:

WAKTU	UID	NAMA	MAKANAN	SALDO

#### d. Algoritma Aplikasi Android



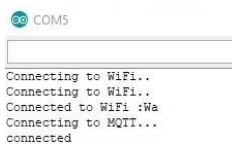
Gambar 5. Algoritma Aplikasi Android

Pada sistem ini dibagi menjadi 2 menu yaitu menu login dan menu utama, proses pertama dimulai dengan menu login. Pada menu login terdapat taskbar untuk mengisi UID beserta data koneksi menuju Database yang bertujuan untuk koneksi ke Database dan memastikan apakah UID telah terdaftar. Jika UID belum terdaftar, maka muncul tampilan untuk mendaftar, namun jika sudah, maka sistem mencoba koneksi Database dan masuk ke menu utama.

Pada menu utama dilakukan proses pengambilan data di broker Database, pengambilan data dilakukan dengan cara memanggil UID yang digunakan. Setelah pengambilan data, proses selanjutnya adalah mengolah data tersebut dan kemudian data yang telah diolah ditampilkan pada aplikasi android.

## HASIL PENGUJIAN

### Pengujian Pengiriman Data UID dari Modul Wifi ke MQTT



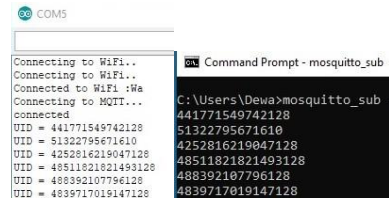
Gambar 6. Koneksi NodeMCU dan MQTT



Gambar 7. Broker MQTT Sebagai subscriber

Dapat dilihat pada gambar 6 proses yang dilakukan adalah mengkoneksikan Broker MQTT menggunakan satu jaringan dengan host MQTT. Lalu pada gambar 7 Broker MQTT dibuka sebagai subscriber yang menerima data dari NodeMCU sesuai dengan topik yang digunakan yaitu UID.

Setelah data terkirim dari NodeMCU yang berperan sebagai Modul Wifi, memastikan data juga diterima di Broker MQTT yang dapat dilihat pada gambar 8. Jika data yang diterima sama dengan data yang dikirim, maka dapat dimastikan pengiriman data berhasil.



Gambar 8. Pengiriman Data UID dari NodeMCU ke Broker MQTT

Pada tabel 1 terdapat hasil uji pengiriman UID, dari beberapa pengiriman data menghasilkan status pengiriman, status pengiriman didapat dari persamaan antara UID yang dikirim oleh NodeMCU dan Broker MQTT. Jika status pengiriman Berhasil, maka dapat dinyatakan data terkirim dengan baik, namun jika status pengiriman Tidak Berhasil, maka data tidak terkirim ke Broker MQTT. Berikut hasil pengujian beberapa UID dari E-KTP.

Tabel 1. Hasil Uji Pengiriman data UID dari NodeMCU ke Broker MQTT

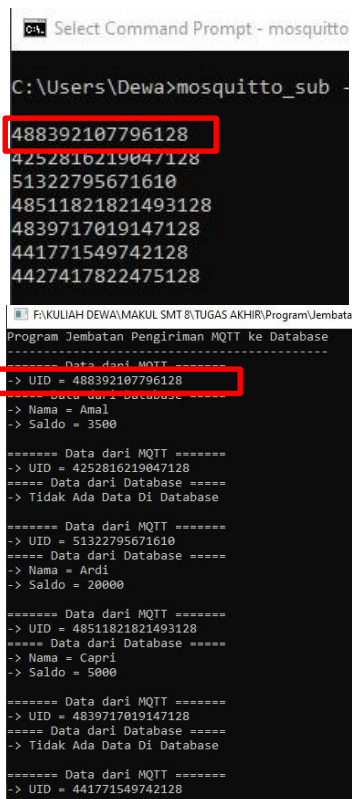
No	UID Modul Wifi	UID Broker	Status Pengiriman
1	441771549742128	441771549742128	Berhasil
2	51322795671610	51322795671610	Berhasil
3	4252816219047120	4252816219047120	Berhasil
4	48511821821493100	48511821821493100	Berhasil
5	488392107796128	488392107796128	Berhasil
6	4839717019147120	4839717019147120	Berhasil
7	4427417822475120	4427417822475120	Berhasil
8	4839717019147120	4839717019147120	Berhasil
9	4252816219047120	4252816219047120	Berhasil
10	488392107796128	488392107796128	Berhasil
11	48511821821493100	48511821821493100	Berhasil
12	4427417822475120	4427417822475120	Berhasil
13	441771549742128	441771549742128	Berhasil
14	51322795671610	51322795671610	Berhasil
15	48511821821493100	48511821821493100	Berhasil
16	488392107796128	488392107796128	Berhasil
17	4839717019147120	4839717019147120	Berhasil
18	4427417822475120	4427417822475120	Berhasil
19	4252816219047120	4252816219047120	Berhasil
20	51322795671610	51322795671610	Berhasil
21	48511821821493100	48511821821493100	Berhasil
22	488392107796128	488392107796128	Berhasil
23	51322795671610	51322795671610	Berhasil
24	4839717019147120	4839717019147120	Berhasil
25	51322795671610	51322795671610	Berhasil
26	441771549742128	441771549742128	Berhasil
27	48511821821493100	48511821821493100	Berhasil
28	4427417822475120	4427417822475120	Berhasil
29	4427417822475120	4427417822475120	Berhasil
30	4252816219047120	4252816219047120	Berhasil

Dilihat dari hasil pengujian pengiriman data pada tabel 1, terdapat 30 kali pengujian pengiriman data dari NodeMCU ke Broker MQTT. Dari 30

pengujian tersebut, data yang berhasil dikirim sebanyak 30 data, maka dapat disimpulkan bahwa persentase keberhasilan pengiriman data dari NodeMCU ke Broker MQTT adalah 100%.

### Pengujian Pengiriman Data Broker MQTT ke Database

Sebelum memulai pengiriman data ke Database, memastikan Modul Wifi telah terkoneksi dengan jaringan yang sama dengan host Broker MQTT seperti pada pengujian 1. Setelah data dari NodeMCU terkirim ke Broker MQTT seperti pada pengujian 1, maka dapat dilakukan pengiriman dari Broker MQTT ke Database melalui aplikasi Console C# sebagai perantara untuk mengirim ke Database. Membuka Aplikasi Console C# dan lakukan pengiriman data.



Gambar 9. Data di Broker MQTT dan Data Pada Console C#

Show all | Number of rows: 25

Options	UID	Nama	Saldo
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	51322795671610	Ardi	20000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	488392107796128	Amal	3500
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	48511821821493128	Capri	5000

Gambar 10. Data pada Database

Pada gambar 9 dapat disimpulkan Bahwa data dari Broker MQTT telah berhasil diterima oleh Aplikasi Console C#, yang mana Broker MQTT mengirimkan data UID yang telah diterima dari NodeMCU pada pengujian 10 ke Aplikasi Console C#. Kemudian data UID yang telah diterima oleh Console C#, dibandingkan ke data yang ada Database dengan tujuan untuk menampilkan Nama dan Saldo dari data UID yang dibandingkan. Pengambilan data pada Database dilakukan menggunakan query SELECT.

```

===== Data dari MQTT =====
-> UID = 488392107796128
===== Data dari Database =====
-> Nama = Amal
-> Saldo = 3500
    
```

UID	Nama	Saldo
51322795671610	Ardi	20000
488392107796128	Amal	3500
48511821821493128	Capri	5000

Gambar 11. Perbandingan data yang terdaftar di Database

Jika UID yang dibandingkan terdaftar di Database, maka Aplikasi Console C# mengambil data Nama dan Saldo yang sebaris dengan UID tersebut. Contohnya ada pada gambar 11, yang mana Aplikasi Console C# mendapatkan Nama Amal, dan Saldo 3500, sedangkan jika UID yang dibandingkan tidak terdaftar di Database, maka data yang didapat oleh Aplikasi Console C# pun tidak ada, seperti yang dicontohkan pada gambar 12.

```

===== Data dari MQTT =====
-> UID = 4252816219047128
===== Data dari Database =====
-> Tidak Ada Data Di Database
    
```

UID	Nama	Saldo
51322795671610	Ardi	20000
488392107796128	Amal	3500
48511821821493128	Capri	5000

Gambar 12. Perbandingan data yang tidak terdaftar di Database

Tabel 2. Data yang dikirim dari MQTT ke Database

No	MQTT UID	UID	Database Nama	Saldo	Status
1	51322795671610	51322795671610	Ardi	20000	Terdaftar
2	48511821821493100	48511821821493100	Capri	5000	Terdaftar
3	441771549742128	441771549742128			Tidak Terdaftar
4	4839717019147120	4839717019147120			Tidak Terdaftar
5	4252816219047120	4252816219047120			Tidak Terdaftar
6	4427417822475120	4427417822475120			Tidak Terdaftar
7	488392107796128	488392107796128	Amal	3500	Terdaftar
8	441771549742128	441771549742128			Tidak Terdaftar
9	488392107796128	488392107796128	Amal	3500	Terdaftar

No	MQTT UID	UID	Database Nama	Saldo	Status
10	48511821821493100	48511821821493100	Capri	5000	Terdaftar
11	4252816219047120	4252816219047120			Tidak
12	51322795671610	51322795671610	Ardi	20000	Terdaftar
13	4839717019147120	4839717019147120			Tidak
14	4427417822475120	4427417822475120			Terdaftar
15	51322795671610	51322795671610	Ardi	20000	Terdaftar
16	4839717019147120	4839717019147120			Tidak
17	48511821821493100	48511821821493100	Capri	5000	Terdaftar
18	441771549742128	441771549742128			Tidak
19	4839717019147120	4839717019147120			Terdaftar
20	488392107796128	488392107796128	Amal	3500	Terdaftar
21	4427417822475120	4427417822475120			Tidak
22	51322795671610	51322795671610	Ardi	20000	Terdaftar
23	4839717019147120	4839717019147120			Tidak
24	488392107796128	488392107796128	Amal	3500	Terdaftar
25	441771549742128	441771549742128			Tidak
26	4252816219047120	4252816219047120			Terdaftar
27	48511821821493100	48511821821493100	Capri	5000	Terdaftar
28	488392107796128	488392107796128	Amal	3500	Terdaftar
29	51322795671610	51322795671610	Ardi	20000	Terdaftar
30	441771549742128	441771549742128			Tidak

Tabel 3. Data yang diterima oleh Database

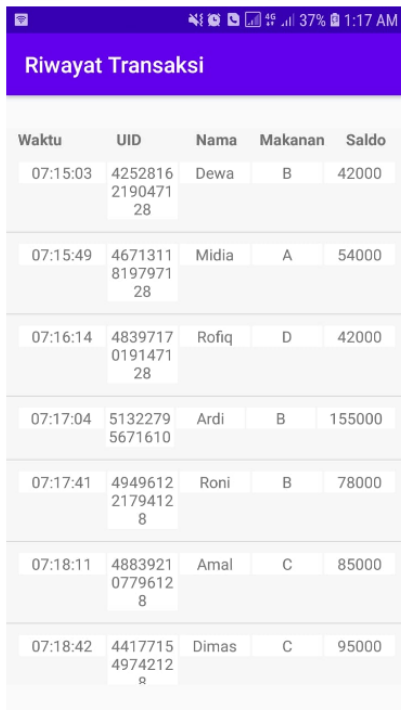
No	Database			Status Pengiriman		
	UID	Nama	Saldo	Status	MQTT-C#	Database
1	51322795671610	Ardi	20000	Terdaftar	Berhasil	Berhasil
2	48511821821493100	Capri	5000	Terdaftar	Berhasil	Berhasil
3				Tidak	Berhasil	Berhasil
4				Terdaftar	Berhasil	Berhasil
5				Tidak	Berhasil	Berhasil
6				Terdaftar	Berhasil	Berhasil
7	488392107796128	Amal	3500	Terdaftar	Berhasil	Berhasil
8				Tidak	Berhasil	Berhasil
9	488392107796128	Amal	3500	Terdaftar	Berhasil	Berhasil
10	48511821821493100	Capri	5000	Terdaftar	Berhasil	Berhasil
11				Tidak	Berhasil	Berhasil
12	51322795671610	Ardi	20000	Terdaftar	Berhasil	Berhasil
13				Tidak	Berhasil	Berhasil
14				Terdaftar	Berhasil	Berhasil
15	51322795671610	Ardi	20000	Terdaftar	Berhasil	Berhasil
16				Tidak	Berhasil	Berhasil
17	48511821821493100	Capri	5000	Terdaftar	Berhasil	Berhasil
18				Tidak	Berhasil	Berhasil
19				Terdaftar	Berhasil	Berhasil
20	488392107796128	Amal	3500	Terdaftar	Berhasil	Berhasil
21				Tidak	Berhasil	Berhasil
22	51322795671610	Ardi	20000	Terdaftar	Berhasil	Berhasil
23				Tidak	Berhasil	Berhasil
24	488392107796128	Amal	3500	Terdaftar	Berhasil	Berhasil
25				Tidak	Berhasil	Berhasil
26				Terdaftar	Berhasil	Berhasil
27	48511821821493100	Capri	5000	Terdaftar	Berhasil	Berhasil
28	488392107796128	Amal	3500	Terdaftar	Berhasil	Berhasil
29	51322795671610	Ardi	20000	Terdaftar	Berhasil	Berhasil
30				Tidak	Berhasil	Berhasil

Tabel 2 dan 3 merupakan hasil uji pengiriman data dari Broker MQTT ke Database, pengujian dilakukan sebanyak 30 kali, sehingga terdapat 30 kali proses pengiriman data dari Broker MQTT ke Database. Dari pengujian sebanyak 30 data yang dapat dilihat pada tabel 2 dan 3, pengiriman yang berhasil diterima oleh Database adalah 30 data, sehingga dapat ditarik kesimpulan bahwa pengiriman dari Broker MQTT ke Database memiliki persentase keberhasilan 100%.

## Pengujian Pengiriman Data dari Database ke Android

Sebelum melakukan pengujian, memastikan telah melakukan pengiriman data seperti pada pengujian 1 dan 2 hingga data terkirim ke Database.

Waktu	UID	Nama	Makanan	Saldo
07:15:03	4252816219047128	Dewa	B	42000
07:15:49	4671311819797128	Midia	A	54000
07:16:14	4839717019147128	Rofiq	D	42000
07:17:04	51322795671610	Ardi	B	155000
07:17:41	494961221794128	Roni	B	78000
07:18:11	488392107796128	Amal	C	85000
07:18:42	441771549742128	Dimas	C	95000
07:19:19	4427417822475128	Ivan	A	37000
07:19:47	4839717019147128	Rofiq	B	37000
07:20:20	51322795671610	Ardi	C	150000
07:20:46	441771549742128	Dimas	D	90000
07:21:13	4252816219047128	Dewa	B	37000
07:21:43	488392107796128	Amal	C	80000
07:22:08	494961221794128	Roni	D	73000
07:22:40	4671311819797128	Midia	A	49000
07:23:13	4427417822475128	Ivan	A	32000
07:23:41	51322795671610	Ardi	C	145000
07:24:11	4671311819797128	Midia	B	44000
07:24:40	4427417822475128	Ivan	D	27000
07:25:05	488392107796128	Amal	D	75000
07:25:32	4252816219047128	Dewa	B	32000
07:26:06	4839717019147128	Rofiq	C	32000
07:26:43	441771549742128	Dimas	A	85000
07:27:15	494961221794128	Roni	C	68000



Gambar 13. Data pada Database dan Aplikasi Android

Pada gambar 13 terdapat 30 data yang ada di Database dan diambil oleh Aplikasi Android. Data

pada Database diterima oleh Aplikasi Android dengan menggunakan perintah SELECT pada Aplikasi Android, sehingga memanggil tabel yang dipilih oleh query SELECT. Pada gambar 4.8 data yang diambil adalah seluruh data dari tabel riwayat\_transaksi. Yaitu Waktu, UID, Nama, Makanan, dan Saldo.

Melakukan perbandingan antara data yang terdapat di Database dan Aplikasi Android agar dapat mengetahui keberhasilan dari pengiriman data. Beberapa contoh perbandingan antar data terdapat pada gambar 14, jika data yang ada di Database sesuai dengan data yang diterima oleh Aplikasi Android, maka pengiriman data dinyatakan berhasil.

Waktu	UID	Nama	Makanan	Saldo
07:15:03	4252816219047128	Dewa	B	42000
07:15:49	4671311819797128	Midia	A	54000
07:16:14	4839717019147128	Rofiq	D	42000

Gambar 14. Perbandingan Data dari Database dan Aplikasi Android

Pada tabel 3 dan 4 terdapat 30 data uji yang telah dikirim dari Database ke Aplikasi Android. Dari 30 data tersebut, data yang dapat diterima oleh Aplikasi Android adalah sebanyak 30 data, sehingga dapat ditarik kesimpulan bahwa pengiriman data dari Database ke Aplikasi Android memiliki persentase keberhasilan sebanyak 100%.

Tabel 4. Data pada Database

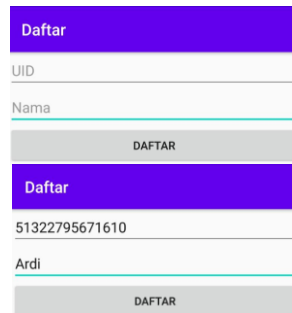
No	Waktu	UID	Database Nama	Makanan	Saldo
1	07:15:03	4252816219047128	Dewa	B	42000
2	07:15:49	4671311819797128	Midia	A	54000
3	07:16:14	4839717019147128	Rofiq	D	42000
4	07:17:04	51322795671610	Ardi	B	155000
5	07:17:41	494961221794128	Roni	B	78000
6	07:18:11	488392107796128	Amal	C	85000
7	07:18:42	441771549742128	Dimas	C	95000
8	07:19:19	4427417822475128	Ivan	A	37000
9	07:19:47	4839717019147128	Rofiq	B	37000
10	07:20:20	51322795671610	Ardi	C	150000
11	07:20:46	441771549742128	Dimas	D	90000
12	07:21:13	4252816219047128	Dewa	B	37000
13	07:21:43	488392107796128	Amal	C	80000
14	07:22:08	494961221794128	Roni	D	73000
15	07:22:40	4671311819797128	Midia	A	49000
16	07:23:13	4427417822475128	Ivan	A	32000
17	07:23:41	51322795671610	Ardi	C	145000
18	07:24:11	4671311819797128	Midia	B	44000
19	07:24:40	4427417822475128	Ivan	D	27000
20	07:25:05	488392107796128	Amal	D	75000
21	07:25:32	4252816219047128	Dewa	B	32000
22	07:26:06	4839717019147128	Rofiq	C	32000
23	07:26:43	441771549742128	Dimas	A	85000
24	07:27:15	494961221794128	Roni	C	68000

No	Waktu	UID	Database Nama	Makanan	Saldo
25	07:45:00	4252816219047128	Dewa	D	22000
26	07:45:32	51322795671610	Ardi	C	140000
27	07:45:58	488392107796128	Amal	B	70000
28	07:46:32	4671311819797128	Midia	A	39000
29	07:46:59	4839717019147128	Rofiq	B	27000
30	07:47:31	494961221794128	Roni	C	63000

Tabel 5. Data pada Aplikasi Android

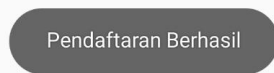
No	Waktu	UID	Nama	Makanan	Saldo	Status Pengiriman
1	07:15:03	4252816219047128	Dewa	B	42000	Berhasil
2	07:15:49	4671311819797128	Midia	A	54000	Berhasil
3	07:16:14	4839717019147128	Rofiq	D	42000	Berhasil
4	07:17:04	51322795671610	Ardi	B	155000	Berhasil
5	07:17:41	494961221794128	Roni	B	78000	Berhasil
6	07:18:11	488392107796128	Amal	C	85000	Berhasil
7	07:18:42	441771549742128	Dimas	C	95000	Berhasil
8	07:19:19	4427417822475128	Ivan	A	37000	Berhasil
9	07:19:47	4839717019147128	Rofiq	B	37000	Berhasil
10	07:20:20	51322795671610	Ardi	C	150000	Berhasil
11	07:20:46	441771549742128	Dimas	D	90000	Berhasil
12	07:21:13	4252816219047128	Dewa	B	37000	Berhasil
13	07:21:43	488392107796128	Amal	C	80000	Berhasil
14	07:22:08	494961221794128	Roni	D	73000	Berhasil
15	07:22:40	4671311819797128	Midia	A	49000	Berhasil
16	07:23:13	4427417822475128	Ivan	A	32000	Berhasil
17	07:23:41	51322795671610	Ardi	C	145000	Berhasil
18	07:24:11	4671311819797128	Midia	B	44000	Berhasil
19	07:24:40	4427417822475128	Ivan	D	27000	Berhasil
20	07:25:05	488392107796128	Amal	D	75000	Berhasil
21	07:25:32	4252816219047128	Dewa	B	32000	Berhasil
22	07:26:06	4839717019147128	Rofiq	C	32000	Berhasil
23	07:26:43	441771549742128	Dimas	A	85000	Berhasil
24	07:27:15	494961221794128	Roni	C	68000	Berhasil
25	07:45:00	4252816219047128	Dewa	D	22000	Berhasil
26	07:45:32	51322795671610	Ardi	C	140000	Berhasil
27	07:45:58	488392107796128	Amal	B	70000	Berhasil
28	07:46:32	4671311819797128	Midia	A	39000	Berhasil
29	07:46:59	4839717019147128	Rofiq	B	27000	Berhasil
30	07:47:31	494961221794128	Roni	C	63000	Berhasil

### Pengujian Pengiriman Data dari Android ke Database



Gambar 15. Aplikasi Daftar

Gambar 15 merupakan tampilan dari pendaftaran UID E-KTP agar pembeli dapat melakukan transaksi. Lakukan pengisian data UID dan Nama, UID setiap E-KTP dapat dilihat pada layer LCD setelah melakukan tapping pada RFID vending machine. Tekan tombol Daftar untuk melakukan pendaftaran.



Gambar 16. Tampilan pendaftaran berhasil

Setelah menekan tombol Daftar, maka muncul pesan bahwa pendaftaran selesai dilakukan, pesan seperti pada gambar 16.



UID	Nama	Saldo
51322795671610	Ardi	0

Gambar 17. Data terdaftar di Database

Setelah itu data masuk kedalam tabel *daftar\_pembeli* yang ada di Database *vending\_machine*. Memastikan data yang di daftarkan sama dengan yang terkirim ke tabel *daftar\_pembeli* seperti gambar 17, jika data berbeda, maka dapat dikatakan bahwa pengiriman gagal. Gambar 18 merupakan contoh dari pendaftaran 8 E-KTP berbeda yang telah dilakukan.

UID	Nama	Saldo
51322795671610	Ardi	0
441771549742128	Dimas	0
488392107796128	Amal	0
494961221794128	Roni	0
4252816219047128	Dewa	0
4427417822475128	Ivan	0
4671311819797128	Midia	0
4839717019147128	Rofiq	0

Gambar 18. Data Pada Tabel *daftar\_pembelian*

Saldo awal setelah pendaftaran adalah 0, karena pada penelitian ini tidak menyediakan fasilitas untuk pengisian saldo, sehingga saldo diisi secara manual oleh penjual atau owner.

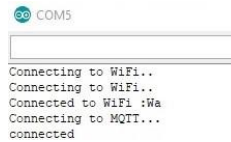
Tabel 5. Pendaftaran E-KTP

No	Android		Database			Status Pengiriman
	UID	Nama	UID	Nama	Saldo	
1	51322795671610	Ardi	51322795671610	Ardi	0	Berhasil
2	441771549742128	Dimas	441771549742128	Dimas	0	Berhasil
3	488392107796128	Amal	488392107796128	Amal	0	Berhasil
4	494961221794128	Roni	494961221794128	Roni	0	Berhasil
5	4252816219047128	Dewa	4252816219047128	Dewa	0	Berhasil
6	4427417822475128	Ivan	4427417822475128	Ivan	0	Berhasil
7	4671311819797128	Midia	4671311819797128	Midia	0	Berhasil
8	4839717019147128	Rofiq	4839717019147128	Rofiq	0	Berhasil
9	441771549742128	Dimas	441771549742128	Dimas	0	Berhasil
10	494961221794128	Roni	494961221794128	Roni	0	Berhasil
11	488392107796128	Amal	488392107796128	Amal	0	Berhasil
12	4427417822475128	Ivan	4427417822475128	Ivan	0	Berhasil
13	51322795671610	Ardi	51322795671610	Ardi	0	Berhasil
14	4671311819797128	Midia	4671311819797128	Midia	0	Berhasil
15	4839717019147128	Rofiq	4839717019147128	Rofiq	0	Berhasil
16	4252816219047128	Dewa	4252816219047128	Dewa	0	Berhasil
17	488392107796128	Amal	488392107796128	Amal	0	Berhasil
18	51322795671610	Ardi	51322795671610	Ardi	0	Berhasil
19	4427417822475128	Ivan	4427417822475128	Ivan	0	Berhasil
20	494961221794128	Roni	494961221794128	Roni	0	Berhasil
21	4671311819797128	Midia	4671311819797128	Midia	0	Berhasil
22	4252816219047128	Dewa	4252816219047128	Dewa	0	Berhasil
23	4839717019147128	Rofiq	4839717019147128	Rofiq	0	Berhasil
24	441771549742128	Dimas	441771549742128	Dimas	0	Berhasil
25	441771549742128	Dimas	441771549742128	Dimas	0	Berhasil
26	4671311819797128	Midia	4671311819797128	Midia	0	Berhasil
27	51322795671610	Ardi	51322795671610	Ardi	0	Berhasil
28	488392107796128	Amal	488392107796128	Amal	0	Berhasil
29	4252816219047128	Dewa	4252816219047128	Dewa	0	Berhasil
30	494961221794128	Roni	494961221794128	Roni	0	Berhasil

Berdasarkan data yang diperoleh pada tabel 5, sebanyak 30 data. Seluruh pengiriman dapat dikatakan berhasil tanpa ada kendala, karena data yang diterima oleh Database adalah 30 data,

sehingga persentase keberhasilan pengiriman data dari Aplikasi Android ke Database adalah 100%.

## Pengujian Pengiriman Data dari Android ke Database



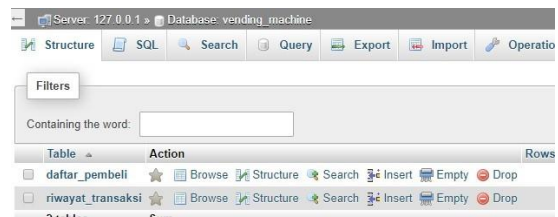
Gambar 19. Koneksi Modul Wifi dengan Wifi dan MQTT

Memastikan Modul Wifi terkoneksi dengan jaringan yang sama dengan jaringan MQTT dan Laptop, gambar 19 adalah tampilan serial monitor dari Modul Wifi yaitu NodeMCU.



Gambar 20. Aplikasi Console C#

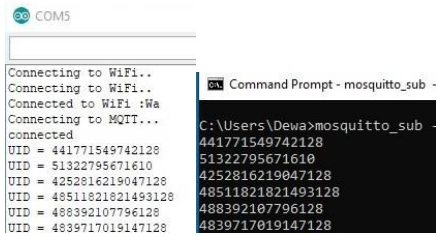
Mem buka aplikasi Console C# yang berfungsi sebagai perantara antara Broker dengan Database.



Gambar 21. Database

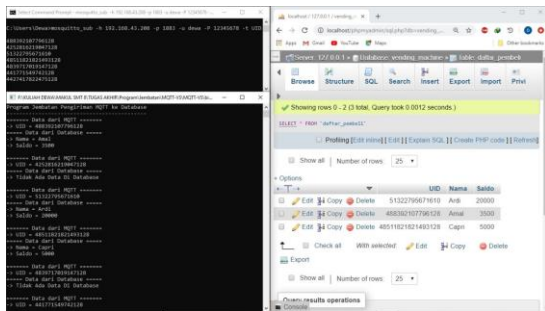
Memastikan modul apache dan mysql dari XAMPP telah aktif, kemudian buka database di Localhost menggunakan browser pada laptop. Database digunakan untuk menyimpan sekaligus memantau setiap transaksi yang berlangsung. Setelah melakukan ketiga hal diatas, proses transaksi baru dapat dimulai.

Proses transaksi dimulai dari menanyakan apakah UID sudah terdaftar atau tidak dengan cara melakukan tapping E-KTP ke RFID yang berada di *vending machine*. Data UID dikirim ke Modul Wifi, setelah Modul Wifi menerima data UID dari *vending machine*, data dikirim ke Broker MQTT. Contoh pengiriman data UID dari Modul Wifi ke Broker MQTT ada pada gambar 22.



Gambar 22. Pengiriman Data dari Modul Wifi ke Broker MQTT

Setelah data terkirim ke Broker MQTT, data secara otomatis terkirim ke aplikasi console c# yang mana console c# berperan sebagai subscriber, sehingga datanya terkirim seperti contoh pada gambar 23. Setelah data diterima oleh console c# data kemudian dikirim ke Database yang bertujuan untuk menanyakan apakah UID sudah terdaftar atau tidak. Jika tidak terdaftar, maka muncul pesan pada LCD yang ada di *vending machine* yang meminta pelanggan untuk mendaftarkan E-KTP terlebih dahulu.



Gambar 23. Pengiriman Broker MQTT hingga ke Database

Langkah selanjutnya dibagi menjadi 2 kasus, yaitu Ketika E-KTP terdaftar dan E-KTP tidak terdaftar, berikut adalah penjelasannya:

**a. E-KTP Terdaftar:**

Lanjut ke proses transaksi yang mana pelanggan menekan tombol pada *vending machine*, lalu Modul Wifi menerima data saldo terbaru dari pelanggan. Setelah mendapatkan nilai saldo Modul Wifi melakukan update saldo pada Database dengan menggunakan komunikasi pengiriman yang sama seperti pemeriksaan UID sebelumnya. Dari gambar 24 dapat disimpulkan bahwa setelah melakukan transaksi kedua, saldo menjadi berkurang.

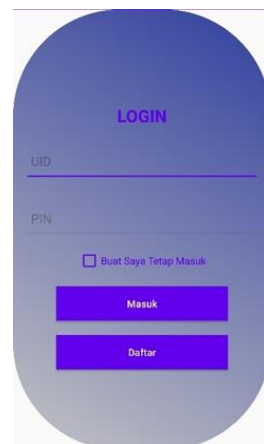
Waktu	UID	Nama	Makanan	Saldo
07:15:03	4252816219047128	Dewa	B	42000
07:15:49	48173116191797128	Midla	A	34000
07:16:14	4839717019147128	Rofiq	D	42000
07:17:04	51322795671610	Ardi	B	155000
07:17:41	494961221794128	Roni	B	78000
07:18:11	488392107796128	Amal	C	85000
07:18:42	441771549742128	Dimas	C	95000
07:19:19	4427417822475128	Ivan	A	37000
07:19:47	4839717019147128	Rofiq	B	37000
07:20:20	51322795671610	Ardi	C	150000
07:20:46	441771549742128	Dimas	D	90000
07:21:13	4252816219047128	Dewa	B	37000

Gambar 24. Tabel riwayat transaksi pada Database

**b. E-KTP Tidak Terdaftar:**

Ketika E-KTP belum Terdaftar, pelanggan diminta untuk mendaftarkan E-KTP dengan cara membuka aplikasi Daftar dan melakukan pendaftaran sesuai dengan penjelasan pada pengujian 4.4.

Selanjutnya data transaksi dapat dilihat dengan cara membuka aplikasi android, tampilan awal aplikasi android adalah menu login yang dapat dilihat pada gambar 25, *user* dianjurkan untuk melakukan proses login terlebih dahulu jika ingin mengakses data transaksi. Selanjutnya ada 2 menu yang disediakan, yaitu menu login sebagai *owner* dan menu login sebagai *customer*. Pada menu login sebagai *customer*, *user* disajikan tampilan riwayat transaksi dari pemilik UID E-KTP yang telah terdaftar, tampilan Riwayat transaksi dapat dilihat pada gambar 26.

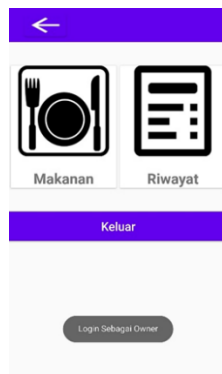


Gambar 25. Menu Login



Gambar 26. Menu Login Customer

Selanjutnya pada menu login sebagai *owner* tampil 2 pilihan menu yaitu menu riwayat transaksi, dan menu makanan. Dapat dilihat pada gambar 27, yaitu tampilan menu login sebagai *owner*.



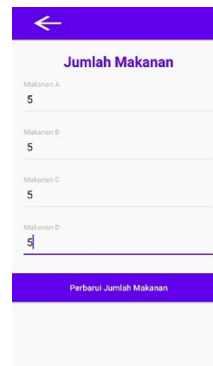
Gambar 27. Menu Login Owner



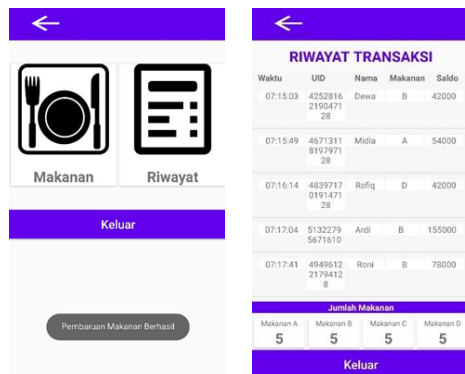
Gambar 28. Riwayat transaksi

Menu riwayat transaksi yang dapat dilihat pada gambar 28 menampilkan data seluruh transaksi yang telah dilakukan oleh pembeli, selain itu menu riwayat transaksi juga menampilkan

jumlah makanan yang tersedia pada mesin, jumlah makanan tersebut berkurang setiap terjadi transaksi pembelian terhadap makanan tersebut, jadi *owner* mampu memantau jumlah makanan yang ada di mesin. Lalu tampilan menu makanan seperti yang terlihat pada gambar 29 memiliki fitur pembaruan jumlah makanan, fungsinya agar *owner* dapat memasukkan jumlah makanan sesuai dengan jumlah yang ada di mesin. Setelah melakukan pembaruan jumlah makanan, maka jumlah tersebut otomatis terbaru pada menu riwayat transaksi seperti pada gambar 30.



Gambar 29. Tampilan menu makanan



Gambar 30. Pembaruan jumlah makanan

Pengujian seluruh sistem terdiri dari gabungan antara pengujian sebelumnya yaitu pengujian 1, pengujian 2, pengujian 3, dan pengujian 4, sehingga hasil pengujian seluruh sistem bergantung pada keberhasilan pengiriman data pada pengujian sebelumnya. Dari seluruh pengujian pengiriman data utama yang telah dijelaskan pada pengujian 1, pengujian 2, pengujian 3, dan pengujian 4, memiliki hasil persentase keberhasilan dalam mengirim data sebesar 100%, sehingga dalam pengujian keseluruhan sistem dapat dikatakan berjalan dengan baik karena pengiriman datanya telah terkirim dengan baik.

## KESIMPULAN

Dari beberapa hasil pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan yaitu:

1. Modul Wifi NodeMCU dapat mengirimkan data ke Broker MQTT tanpa ada halangan dengan persentase keberhasilan dari pengiriman data sebesar 100%, data yang diterima pun akurat, atau sama seperti data yang dikirimkan.
2. Pengiriman dari Broker MQTT ke Database tidak dilakukan secara langsung, sehingga membutuhkan aplikasi tambahan sebagai perantara pengirimannya. Penelitian ini menggunakan C# Console Visual Studio sebagai jembatan pengiriman data ke Database. Berdasarkan data uji yang telah didapatkan, proses pengiriman berjalan dengan baik dengan persentase keberhasilan mencapai 100%. Untuk pengiriman dari Database ke Aplikasi Android juga dapat dikatakan baik dengan persentase keberhasilan pengiriman datanya mencapai 100%. Namun kekurangan dari pengiriman Database ini ialah Database yang digunakan merupakan Database lokal yang dibuat menjadi terhubung internet dengan menggunakan IP *Public* milik instansi lain, sehingga koneksi dari Database ini bergantung pada koneksi server pemilik IP *Public* tersebut.
3. Sama halnya seperti pengiriman data dari Database ke android, pengiriman data yang dilakukan pada android ke Database dapat dikatakan terkirim dengan baik karena memiliki persentase keberhasilan mencapai 100%. Namun karena Database yang digunakan sama, maka koneksinya bergantung pada IP *Public* milik instansi lain, jika server tersebut mati, maka koneksi ke Database tidak bisa dilakukan secara online menggunakan internet, namun hanya bisa dilakukan secara lokal.

## Saran

Adapun beberapa saran yang diberikan untuk melakukan penelitian mengenai pengiriman data menggunakan MQTT secara lebih baik lagi. Beberapa saran tersebut antara lain:

1. Menggunakan Broker MQTT yang langsung terhubung ke internet dan dapat diubah logika pemrogramannya, sehingga pengiriman melalui internet dapat dilakukan dengan lebih mudah dan lebih fleksibel lagi karena dapat diprogram sesuai kebutuhan.
2. Menggunakan Broker MQTT yang bisa langsung berkomunikasi dengan database,

sehingga tidak memerlukan program tambahan untuk menyimpan data yang telah disubscribe.

## DAFTAR PUSTAKA

- Adiputra, D. S. (2015). Mesin Penjual Softdrink Otomatis Berbasis ATMEGA8535. *ELEMENTER*, Vol. 1, No. 2.
- Iskandar, H. R. (2019). Eksperimental Uji Kekeruhan Air Berbasis Internet of Things Menggunakan Sensor DFRobot SEN0189 dan MQTT Cloud Server. *TE*, 002.
- Kompas. (2017, 12 8). *Riwayat dan Asa "Vending Machine" di Indonesia*. Diambil kembali dari biz.kompas.com: <https://biz.kompas.com/read/2017/12/08/114616128/riwayat-dan-asa-vending-machine-di-indonesia>
- Nurrochman, A. R. (2019, Juli 22). *Rancang Bangun Vending Machine Minuman Berkarbonasi Berbasis Arduino Mega 2560*. Diambil kembali dari repository.mercubuana.ac.id: <https://repository.mercubuana.ac.id/49986/>
- Rochman, H. P. (2017). Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 6, p. 445-455.
- Safii, M. F. (2019). Analisis Quality Of Service Protokol Mqtt Dan Http Pada Penerapan Sistem Monitoring Suhu Berbasis Nodemcu (Studi Kasus Ruang Server Kampus 3 Ist Akprind Yogyakarta). *Jurnal JARKOM*, Vol. 7 No. 1.