



UNIVERSITAS
Dinamika

**INTEGRASI SISTEM TRANSAKSI PADA *VENDING MACHINE*
MENGUNAKAN CLOUD MQTT**

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

UNIVERSITAS
Dinamika

Oleh:

Yohanes Dewa Bayu Adyawadhana

16410200015

FAKULTAS TEKNOLOGI DAN INFORMATIKA

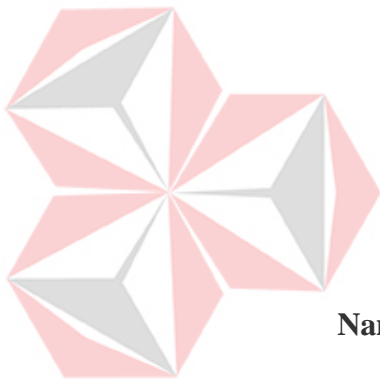
UNIVERSITAS DINAMIKA

2020

**INTEGRASI SISTEM TRANSAKSI PADA *VENDING MACHINE*
MENGUNAKAN CLOUD MQTT**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



Disusun Oleh :

Nama : Yohanes Dewa Bayu Adyawadhana

NIM : 16410200015

Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2020

TUGAS AKHIR

INTEGRASI SISTEM TRANSAKSI PADA *VENDING MACHINE* MENGUNAKAN CLOUD MQTT

Dipersiapkan dan disusun oleh
Yohanes Dewa Bayu Adyawadhana
NIM: 16410200015

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas
Pada: Selasa, 18 Agustus 2020

Susunan Dewan Pembahas

Pembimbing:

I. Harianto, S.Kom., M.Eng.

NIDN 0722087701


II. Ira Puspasari, S.Si., M.T.

NIDN 0710078601

Pembahas:

Dr. Jusak


NIDN 0708017101


Digitally signed by
Universitas Dinamika
Date: 2020.09.03
08:09:29 +07'00'


Digitally signed by
Universitas Dinamika
Date: 2020.09.02
09:54:02 +07'00'


Digitally signed by
Universitas
Dinamika
Date: 2020.09.03
10:08:03 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana


Digitally signed by
Universitas
Dinamika
Date: 2020.09.03
23:46:22 +07'00'

Dr. Jusak

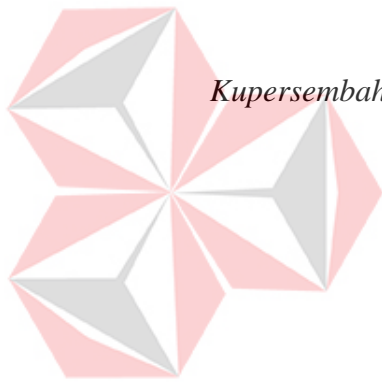
NIDN 0708017101

Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA

“Konyol Boleh.. Bodoh Jangan..”



UNIVERSITAS
Dinamika



Kupersembahkan tugas akhir ini untuk kedua orang tua dan teman-teman yang selalu mendukung dan memberikan semangat.

UNIVERSITAS
Dinamika

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Yohanes Dewa Bayu Adyawadhana
NIM : 16410200015
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **INTEGRASI SISTEM TRANSAKSI PADA VENDING
MACHINE MENGGUNAKAN CLOUD MQTT**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 18 Agustus 2020

Yang menyatakan



Yohanes Dewa Bayu Adyawadhana

Nim : 16410200015

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 secanggih mesin yang ada di negara-negara Asia lainnya. Melihat kurangnya peminat pada *vending machine* di Indonesia, maka pada tugas akhir ini dibuat sistem pengiriman data *vending machine* menggunakan protokol MQTT sebagai sarana yang akan membuat data transaksi *vending machine* menjadi terhubung ke internet dan kemudian akan 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*

KATA PENGANTAR

Puji dan syukur atas kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Integrasi Sistem Transaksi Pada *Vending Machine* Menggunakan Cloud MQTT”.

Dalam pelaksanaan tugas akhir dan penyelesaian laporan tugas akhir ini, penulis mendapatkan bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Orang tua dan keluarga besar penulis yang selalu memberikan dukungan dan motivasi.
2. Bapak Harianto, S.Kom., M.Eng., selaku dosen pembimbing yang telah memberikan banyak saran dan wawasan untuk menjadi lebih baik.
3. Ibu Ira Puspasari, S.Si., M.T., selaku dosen pembimbing yang telah memberikan dukungan penuh berupa motivasi, saran, dan wawasan bagi penulis selama pelaksanaan tugas akhir dan pembuatan laporan tugas akhir ini.
4. Bapak Dr. Jusak, selaku dosen pembahas yang banyak memberikan masukan agar tugas akhir ini menjadi lebih baik.
5. Dan teman-teman lain yang masih bertahan berada disisi penulis maupun yang pernah berada disisi penulis, dukungan yang pernah diberikan tidak akan dilupakan oleh penulis.

Semoga Tuhan Yang Maha Esa memberikan rahmat-Nya kepada seluruh pihak yang membantu penulis dalam penyelesaian tugas akhir.

Penulis menyadari di dalam laporan tugas akhir ini masih memiliki banyak kekurangan, meskipun demikian penulis tetap berharap laporan tugas akhir ini dapat bermanfaat bagi semua pihak dan dapat menjadi bahan acuan untuk penelitian selanjutnya.

Surabaya, 18 Agustus 2020

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Komunikasi Wireless	5
2.1.1 Vending Machine.....	5
2.2 Protokol	5
2.2.1 MQTT	5
2.3 Database	7
2.3.1 MySQL.....	7
2.4 Android	7
2.4.1 Android Studio	8
2.5 <i>Cashless</i>	8
2.6 Mikrokontroler	8
2.6.1 NodeMCU ESP8266	8

2.6.2 RFID.....	9
BAB III METODOLOGI PENELITIAN	11
3.1 Model Perancangan.....	11
3.2 Perancangan Perangkat Keras	13
3.2.1 Modul Wifi.....	13
3.3 Perancangan Perangkat Lunak	14
3.3.1 Algoritma Modul Wifi	14
3.3.2 Algoritma Broker MQTT.....	15
3.3.3 Algoritma Database.....	16
3.3.4 Algoritma Aplikasi Android	17
3.4 Indikator Keberhasilan	18
3.4.1 Pengujian pengiriman data pembeli antara mikrokontroler NodeMCU ESP8266 dengan Broker MQTT	18
3.4.2 Pengujian penyimpanan data pembeli ke database MySQL	18
3.4.3 Pengujian pengiriman data pembeli antara Database dengan Aplikasi Android.....	18
BAB IV HASIL DAN PEMBAHASAN.....	19
4.1 Pengujian Pengiriman Data UID dari Modul Wifi ke MQTT	19
Tujuan	19
Peralatan yang Digunakan.....	19
Cara Pengujian	19
Hasil Pengujian	20
Analisis Data	22
4.2 Pengujian Pengiriman Data Broker MQTT ke Database.....	22
Tujuan	22
Peralatan yang Digunakan.....	22

Cara Pengujian	22
Hasil Pengujian	22
Analisis Data	27
4.3 Pengujian Pengiriman Data dari Database ke Android.....	27
Tujuan	27
Peralatan yang Digunakan.....	27
Cara Pengujian	27
Hasil Pengujian	27
Analisis Data	31
4.4 Pengujian Pengiriman Data dari Android ke Database.....	31
Tujuan	31
Peralatan yang Digunakan.....	31
Cara Pengujian	31
Hasil Pengujian	32
Analisis Data	34
4.5 Pengujian Seluruh Sistem	34
Tujuan	34
Peralatan yang Digunakan.....	34
Cara Pengujian	34
Hasil Pengujian	35
Analisis Data	40
BAB V PENUTUP	41
5.1 Kesimpulan	41
5.2 Saran.....	42
DAFTAR PUSTAKA	43
BIODATA	44

LAMPIRAN.....	45
---------------	----



UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2.1 NodeMCU ESP8266	9
Gambar 3.1 Blok Diagram Model Perancangan	11
Gambar 3.2 Rangkaian Skematik Modul Wifi.....	13
Gambar 3.3 Algoritma Modul Wifi	14
Gambar 3.4 Algoritma Broker MQTT	15
Gambar 4.1 Koneksi NodeMCU dan MQTT.....	20
Gambar 4.2 Broker MQTT Sebagai subscriber	20
Gambar 4.3 Pengiriman Data UID dari NodeMCU ke Broker MQTT.....	20
Gambar 4.4 Data di Broker MQTT dan Data Pada Console C#.....	23
Gambar 4.5 Data pada Database	23
Gambar 4.6 Perbandingan data yang terdaftar di Database	24
Gambar 4.7 Perbandingan data yang tidak terdaftar di Database	24
Gambar 4.8 Data pada Database dan Aplikasi Android	28
Gambar 4.9 Perbandingan Data dari Database dan Aplikasi Android.....	29
Gambar 4.10 Aplikasi Daftar	32
Gambar 4.11 Tampilan Pendaftaran Berhasil	32
Gambar 4.12 Data Terdaftar di Database.....	32
Gambar 4.13 Data Pada Tabel daftar_pembelian	33
Gambar 4.14 Koneksi Modul Wifi dengan Wifi dan MQTT.....	35
Gambar 4.15 Aplikasi Console C#.....	35
Gambar 4.16 Database	35
Gambar 4.17 Pengiriman Data dari Modul Wifi ke Broker MQTT	36
Gambar 4.18 Pengiriman Broker MQTT hingga ke Database.....	37
Gambar 4.19 Tabel Riwayat Transaksi pada Database.....	37

Gambar 4.20 Menu Login	38
Gambar 4.21 Menu Login <i>Customer</i>	38
Gambar 4.22 Menu Login <i>Owner</i>	39
Gambar 4.23 Riwayat Transaksi	39
Gambar 4.24 Tampilan Menu Makanan	39
Gambar 4.25 Pembaruan Jumlah Makanan	40



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 3.1 Daftar Pembeli	16
Tabel 3.2 Riwayat Transaksi.....	16
Tabel 4.1 Hasil Uji Pengiriman data UID dari NodeMCU ke Broker MQTT.....	21
Tabel 4.2 Data yang dikirim dari MQTT ke Database.....	25
Tabel 4.3 Data yang diterima oleh Database	26
Tabel 4.4 Data pada Database	29
Tabel 4.5 Data pada Aplikasi Android.....	30
Tabel 4.6 Pendaftaran E-KTP	33



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Arduino Modul Wifi (NodeMCU)	45
Lampiran 2 Visual Studio (Console C#)	50
Lampiran 3 Android Studio (build.gradle (;app))	54
Lampiran 4 Android Studio (AndroidManifest.xml)	55
Lampiran 5 Android Studio (list.xml)	56
Lampiran 6 Android Studio (activity_main.xml)	58
Lampiran 7 Android Studio (MainActivity.java)	60



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di Indonesia saat ini telah memberikan banyak manfaat, salah satu nya 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 secanggih 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, Riwayat dan Asa “Vending Machine” di Indonesia, 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 tugas akhir 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 akan 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 akan disimpan di database bukan hanya UID saja, melainkan data pembeli yang meliputi: UID, saldo pembeli, serta data riwayat pembelian.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan dalam tugas akhir ini:

1. Bagaimana cara mengirim data pembeli dari Modul Wifi ke Broker MQTT melalui jaringan wifi?
2. Bagaimana cara mengirim data pembeli dari Broker MQTT ke Aplikasi Android?
3. Bagaimana cara mengirim data pembeli dari Aplikasi Android ke Database?

1.3 Batasan Masalah

Dalam perancangan sistem ini, pembahasan masalah dibatasi pada beberapa hal berikut:

1. Sistem tidak memiliki fasilitas untuk pengisian saldo pembeli.
2. Sistem menggunakan database sederhana, sehingga tidak menjamin keamanan akan database tersebut.
3. Broker yang digunakan pada sistem bersifat tetap atau tidak dapat diedit.
4. Tabel transaksi yang digunakan hanya sebatas transaksi sederhana, seperti riwayat penjualan maupun riwayat pembelian.

1.4 Tujuan Penelitian

Berdasarkan uraian latar belakang dan rumusan masalah di atas, dalam Tugas Akhir ini didapatkan tujuan pembuatan laporan sebagai berikut:

1. Mengirimkan data pembeli dari Modul Wifi menuju ke Broker MQTT melalui jaringan wifi.
2. Mengirimkan data pembeli dari Broker MQTT menuju Aplikasi Android melalui jaringan wifi.
3. Mengirimkan data pembeli dari Aplikasi Android menuju Database melalui jaringan wifi.

1.5 Manfaat Penelitian

Adapun manfaat penyusunan tugas akhir ini adalah untuk membuat suatu inovasi yang dapat menarik peminat dari penggunaan *vending machine* di Indonesia.



UNIVERSITAS
Dinamika

BAB II

TINJAUAN PUSTAKA

2.1 Komunikasi Wireless

2.1.1 Vending Machine

Vending machine adalah sebuah mesin yang menjual barang secara otomatis yang mana tidak membutuhkan tenaga operator untuk menjual barang, sehingga dapat mempermudah pekerjaan dari seorang penjual. Menurut (Kompas, Asal mula mesin penjual otomatis di Jepang, 2019) *vending machine* ditemukan pada abad pertama oleh seorang pahlawan dari Alexandria. *Vending Machine* modern pertama kali diperkenalkan di London pada tahun 1883, *vending machine* ini menjual kartu pos yang kemudian dipasang di area stasiun. Perkembangan vending machine justru berkembang pesat di Jepang, dimana hampir segala kebutuhan sehari-hari menggunakan *vending machine* sebagai mesin penjualnya.

2.2 Protokol

Protokol adalah suatu peraturan atau prosedur untuk mengirim sebuah data pada perangkat elektronik. Salah satunya ada pada sistem komputer, agar setiap komputer dapat berkomunikasi dengan komputer lainnya maka dibutuhkan sebuah protokol. Protokol yang biasa digunakan dalam sistem komputer yaitu seperti UDP, FTP, TCP, ICMP. Adapun protokol yang digunakan sebagai komunikasi IoT pada sebuah sistem komputer, salah satunya yaitu HTTP dan MQTT.

2.2.1 MQTT

MQTT merupakan salah satu protokol yang dapat digunakan untuk komunikasi pada sistem komputer. MQTT berjalan di atas TCP/IP dan dirancang khusus untuk *machine to machine* yang tidak memiliki alamat khusus seperti halnya sebuah mikrokontroler. Menurut (Iskandar, 2019) selain komunikasi berbasis web atau pengukuran menggunakan teknologi berbasis fiber optic, IoT berbasis MQTT broker ini digunakan sebagai langkah pengembangan untuk membuat sistem monitoring jarak jauh. Sistem transaksi pada tugas akhir ini juga dapat dikatakan

sebagai sistem monitoring jarak jauh, yang mana penjual dapat memantau kondisi *vending machine* dari manapun itu selama memiliki koneksi internet.

Sistem kerja MQTT adalah menerapkan sistem publish dan subscribe. Ada 3 bagian penting pada sistem MQTT yaitu publisher, subscriber, dan broker. Pada penerapannya, device akan terhubung pada sebuah broker yang mempunyai suatu topik tertentu. Berikut adalah penjelasan dari publish, subscribe, broker, dan topik:

1. Publish merupakan cara suatu device untuk mengirimkan data ke subscriber. Biasanya pada tahap publish ini device yang digunakan adalah sebuah device yang terhubung dengan sensor tertentu. Sedangkan device yang akan mengirimkan data kepada subscriber akan disebut sebagai publisher.
2. Subscribe merupakan cara suatu device untuk menerima data dari publisher. Sama seperti publisher, device yang akan menerima data dari publisher akan disebut sebagai subscriber.
3. Broker pada MQTT berfungsi sebagai jembatan pengiriman dan penerimaan data dari publisher serta subscriber, broker juga bisa diibaratkan sebagai server yang memiliki alamat IP khusus. Pada tugas akhir ini akan menggunakan *Cloud MQTT* sebagai brokernya, yang mana *Cloud MQTT* akan memberikan alamat server, port, *username*, serta *password* untuk setiap akunnya. Broker *Cloud MQTT* hanya akan menampilkan data sesuai dengan topik yang dikirimkan.
4. Topik merupakan pengelompokan data disuatu kategori tertentu. Pada sistem kerja MQTT, topik bersifat wajib hukumnya. Pada setiap pengiriman maupun penerimaan data antara Publisher dan Subscriber harus memiliki suatu topik tertentu sebagai pembeda antar data.

Pada tugas akhir ini akan menggunakan MQTT sebagai protokol pengiriman datanya, alasan penggunaan dari MQTT sendiri dikarenakan sifat protokol MQTT yang mampu mengirimkan data berukuran kecil. MQTT juga digunakan untuk pengiriman data dari mesin ke mesin yang tidak memiliki alamat khusus, selain itu pengiriman data melalui MQTT bersifat data *realtime* sehingga sangat sesuai dengan kebutuhan dari tugas akhir ini.

2.3 Database

Database merupakan kumpulan dari berbagai data dan informasi yang tersimpan dan tersusun di dalam suatu sistem komputer secara sistematis, dapat diperiksa, diolah atau dimanipulasi. Istilah database sendiri mengacu pada koleksi data-data yang saling terkait satu sama lain dimana tujuan database adalah dapat digunakan untuk mengelola data dengan lebih efektif dan efisien. Cara mengelola data dari database yaitu dengan menggunakan sebuah perangkat lunak komputer, perangkat lunak yang digunakan untuk mengelola dan memanggil database disebut dengan DBMS (*database management system*).

2.3.1 MySQL

MySQL merupakan salah satu DBMS yang memiliki fasilitas *open source* dengan lisensi GPL (*General Public License*) yang mana dapat digunakan untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. MySQL termasuk ke dalam jenis RDBMS (*Relational Database Management System*) yang mana data dari sebuah tabel database memiliki relasi dengan tabel lainnya. Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Sehingga dalam database MySQL dapat terdiri dari satu ataupun beberapa tabel.

SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada database yang terstruktur. Jadi MySQL juga dapat diartikan sebagai DBMS yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server.

2.4 Android

Menurut (Giyartono, 2015) “Android adalah *operating system* atau OS berbasis linux yang diperuntukan khusus untuk *mobile device* seperti *smartphone* atau PC table, persis seperti *sybian* yang digunakan oleh Nokia dan Blackberry OS, jelasnya seperti *Microsoft windows* yang sangat dikenal baik oleh para pengguna komputer dan laptop, jika kita analogikan, android adalah windows nya sedangkan *smartphone* atau *handphone* atau tablet adalah unit komputernya.”

Android bersifat open source atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak.

Dengan sifat open source perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi. Pada tugas akhir ini akan digunakan sebuah perangkat lunak bernama *android studio* sebagai aplikasi *design* pada *smartphone*, yang mana akan mengolah data sehingga *user* dapat memantau data yang diperlukan.

2.4.1 Android Studio

Android Studio adalah perangkat lunak yang digunakan untuk mengembangkan aplikasi android dengan berbagai fitur sesuai dengan inovasi pembuatnya. *Android Studio* merupakan Integrated Development Environment (IDE) resmi yang khusus digunakan untuk pengembangan aplikasi Android.

2.5 Cashless

Cashless merupakan sistem transaksi yang mana segala proses transaksinya tidak lagi menggunakan uang tunai melainkan melalui media elektronik seperti kartu debit dan dompet *virtual*. Saat ini sistem *cashless* populer dikalangan masyarakat, dengan sistem ini selain dapat mempermudah serta mempersingkat proses transaksi, *cashless* juga dapat menjamin keamanan dari sistem transaksi karena terdapat rekam jejak proses transaksinya.

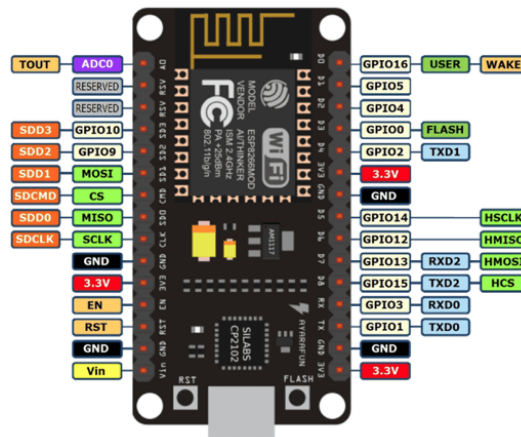
2.6 Mikrokontroler

Menurut (Haryono, 2017) mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer. Mikrokontroler juga dapat disebut sebagai sistem komputer yang mempunyai salah satu atau beberapa tugas yang sangat spesifik.

2.6.1 NodeMCU ESP8266

NodeMCU adalah sebuah platform IoT yang bersifat *open source*, terdiri dari perangkat keras berupa *System OnChip* ESP8266. NodeMCU dapat dianalogikan sebagai board arduinonya ESP8266, karena telah membuat ESP8266 menjadi satu kesatuan ke dalam sebuah board yang juga memiliki berbagai fitur layaknya

mikrokontroler ditambah kemampuan akses terhadap Wifi serta chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB. *Pinout* dari NodeMCU ESP8266 dapat dilihat pada gambar 2.1.



Gambar 2.1 NodeMCU ESP8266

2.6.2 RFID

RFID (*Radio Frequency Identification*) adalah sebuah sistem identifikasi berbasis *wireless* yang memungkinkan akan pengambilan data tanpa harus bersentuhan seperti *barcode* atau *magnetic card*. RFID menggunakan sistem identifikasi dengan gelombang radio, alat ini menggunakan sistem radiasi elektromagnetik untuk mengirimkan data, karena itu minimal dibutuhkan dua buah perangkat agar alat ini dapat berfungsi, perangkat yang dibutuhkan yaitu *Tag* dan *Reader*. Berikut adalah penjelasan dari RFID *Tag* dan RFID *Reader*:

2.6.2.1 RFID Tag

Alat yang melekat pada objek yang akan diidentifikasi oleh RFID Reader. Terdapat 2 jenis RFID *Tag* yaitu *Tag* pasif dan *Tag* aktif. *Tag* pasif adalah *Tag* tanpa adanya penggunaan baterai sebagai dayanya, sedangkan *Tag* aktif menggunakan baterai untuk dapat berfungsi. RFID *Tag* berupa perangkat *read-only* yang artinya hanya dapat dibaca saja ataupun perangkat *read-write* yang artinya dapat dibaca dan ditulis ulang.

2.6.2.2 RFID Reader

RFID *Reader* merupakan alat pembaca dari RFID *Tag*. Ada dua macam RFID *Reader* yaitu *Reader* Pasif dan *Reader* Aktif. *Reader* Pasif memiliki

sistem pembaca pasif yang hanya dapat menerima sinyal radio dari *Tag* Aktif. Jangkauan penerima alat ini dapat mencapai sampai dengan jarak 600 meter. Reader Aktif memiliki sistem pembaca aktif yang dapat memancarkan sinyal interogator ke *Tag* dan menerima balasan autentikasi dari *Tag*. Sinyal interogator ini juga menginduksi *Tag* dan akhirnya menjadi sinyal DC sehingga dapat menjadi sumber daya *Tag* Pasif.

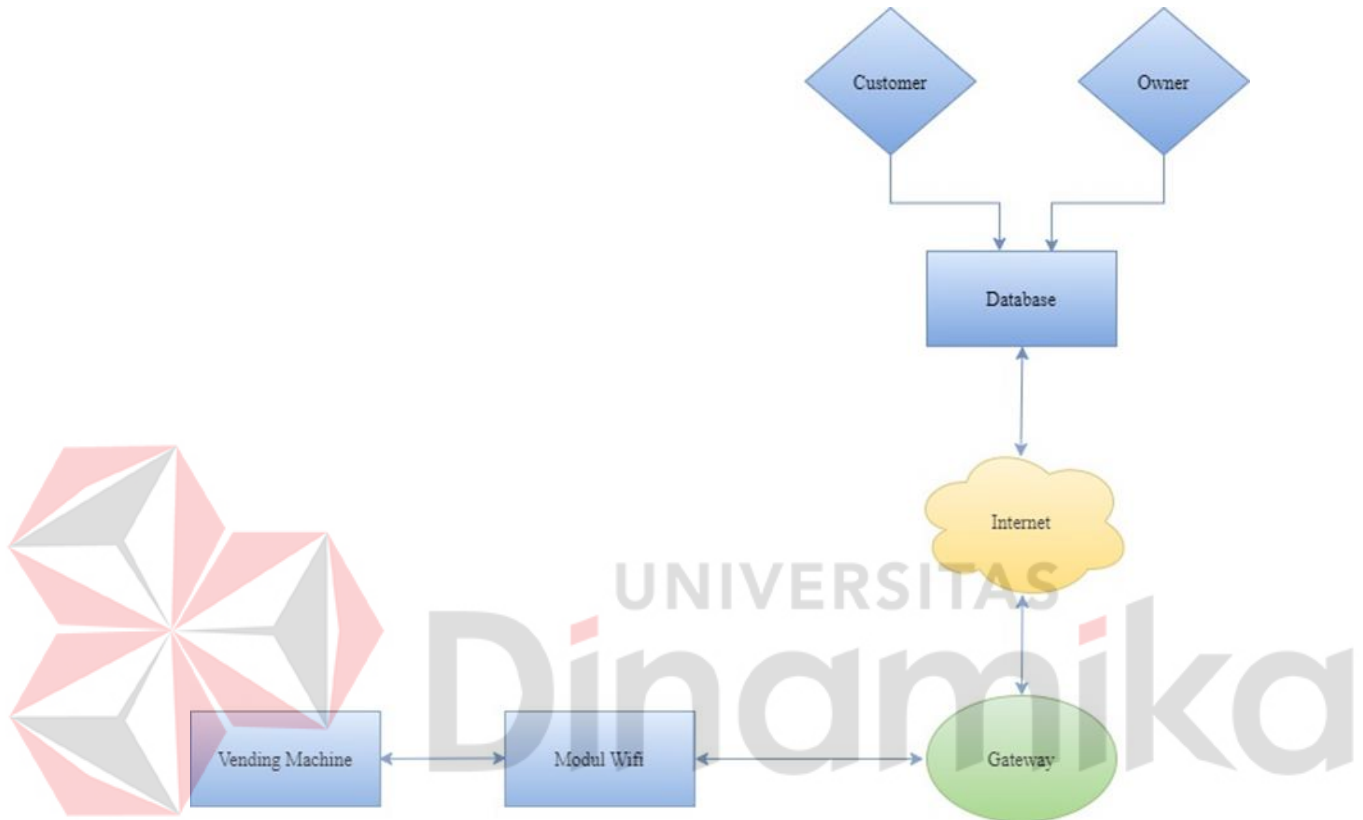


UNIVERSITAS
Dinamika

BAB III

METODOLOGI PENELITIAN

3.1 Model Perancangan



Gambar 3.1 Blok Diagram Model Perancangan

Gambar 3.1 merupakan model perancangan dari sistem pengiriman data *vending machine* yang dibagi menjadi beberapa bagian, berikut adalah penjelasan setiap bagian dari diagram di atas.

1. *Vending Machine*

Proses ini merupakan proses pengaplikasian dari alat *vending machine* yang akan 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 akan mengirimkan data berupa nilai dari ID pembeli serta aksi yang telah dilakukan oleh alat, yang kemudian akan di kirim ke proses selanjutnya yaitu proses Modul Wifi.

2. Modul Wifi

Pada tahap ini, data pada alat *vending machine* akan 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 akan mengirimkan data yang telah dibagi menjadi beberapa topik kepada Broker MQTT.

3. Gateway

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

4. Internet

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

5. Database

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

6. Customer

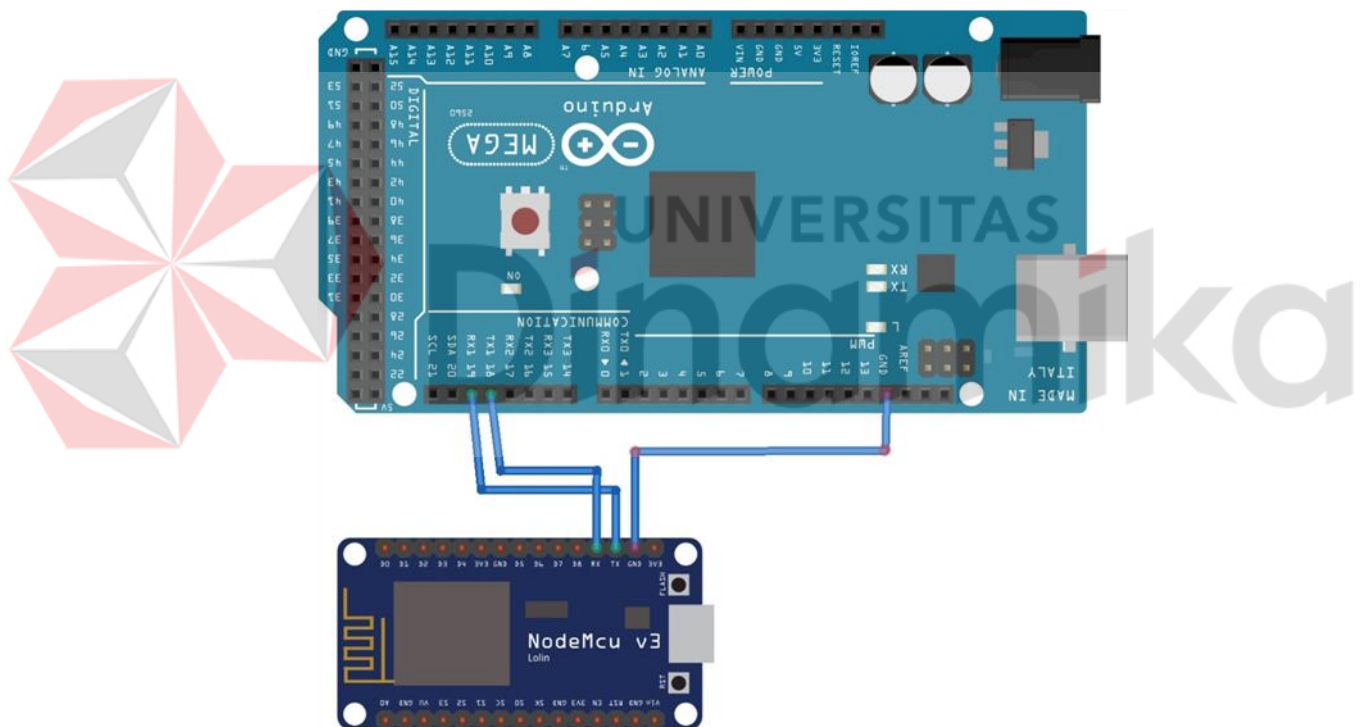
Tahap *Customer* adalah tahap yang akan dilakukan oleh seorang pembeli, yang mana pembeli dapat melakukan proses registrasi dan histori melalui aplikasi android. Pada proses registrasi, pembeli akan melakukan pengisian data yang bertujuan sebagai data pengenalan bagi UID dari E-KTP yang digunakan. Sedangkan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

7. Owner

Tahap *Owner* adalah tahap yang akan dilakukan oleh seorang pemilik atau penjual, yang mana penjual juga mampu melakukan 2 proses yaitu proses update makanan dan histori. Pada proses update makanan, penjual dapat mengubah jumlah makanan sesuai dengan jumlah yang ada di mesin. Sedangkan pada proses histori, penjual dapat melakukan pemeriksaan riwayat transaksi yang sudah dilakukan seperti waktu transaksi, makanan, hingga akumulasi dari saldo pembeli.

3.2 Perancangan Perangkat Keras

3.2.1 Modul Wifi



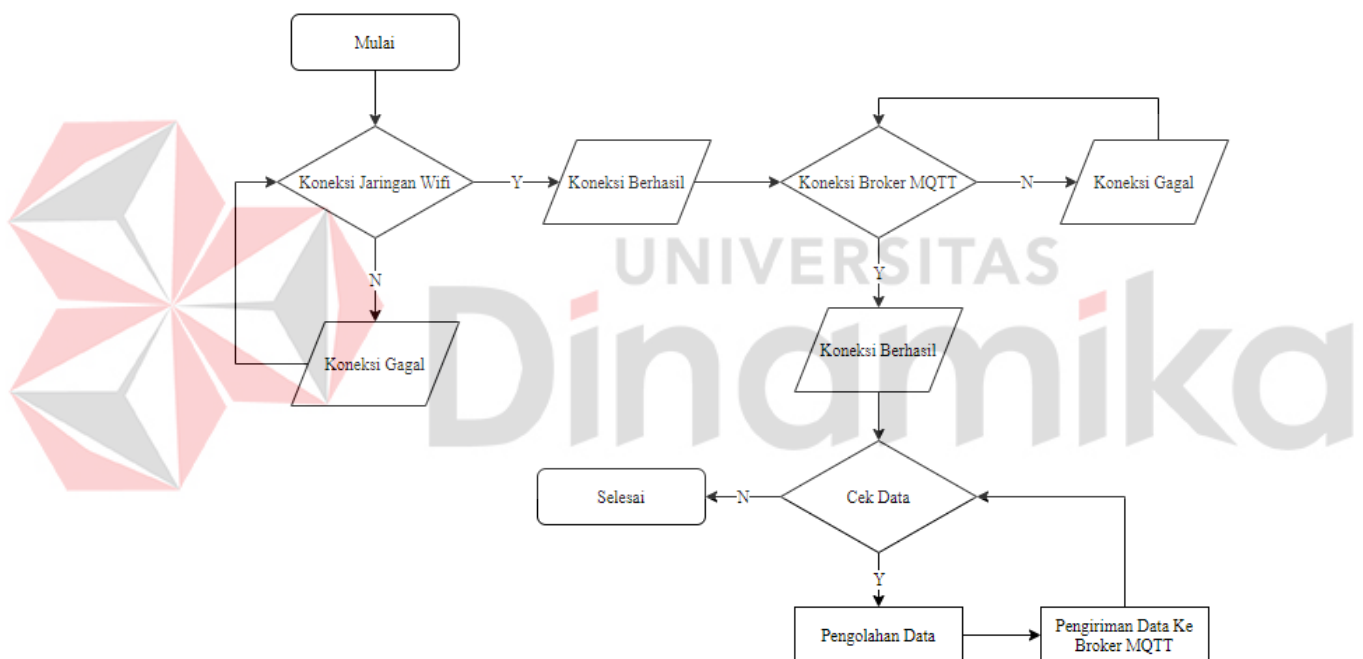
Gambar 3.2 Rangkaian Skematik Modul Wifi

Rangkaian pada gambar 3.2 merupakan rangkaian yang akan digunakan pada tugas akhir ini. Arduino Mega adalah mikrokontroler yang akan digunakan oleh alat *vending machine*, dan kemudian akan mengirimkan data *character* kepada mikrokontroler NodeMcu. NodeMcu disini berfungsi sebagai modul wifi yang akan 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.

3.3 Perancangan Perangkat Lunak

3.3.1 Algoritma Modul Wifi



Gambar 3.3 Algoritma Modul Wifi

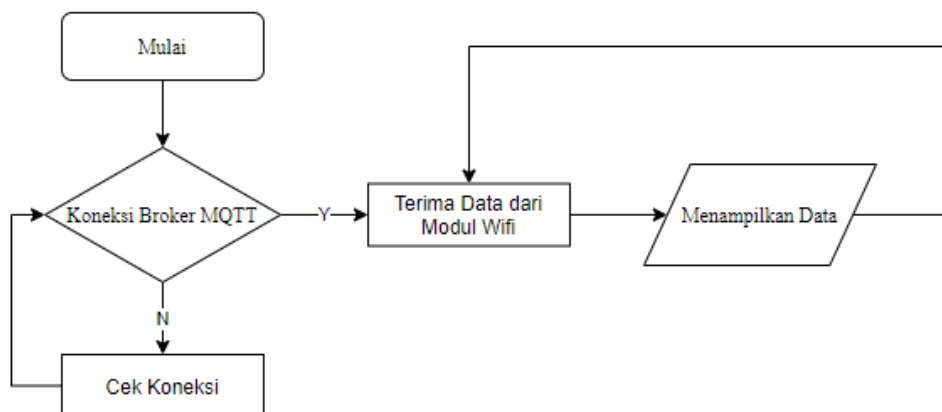
Gambar 3.3 merupakan gambar algoritma modul wifi, sistem ini dimulai dari koneksi antara modul wifi dan jaringan wifi dengan mendeklarasikan SSID dan PASSWORD dari jaringan wifi yang saat ini digunakan, hal ini dilakukan agar perangkat modul wifi dapat terhubung ke jaringan internet. Apabila koneksi berhasil, maka sistem akan menampilkan teks bahwa koneksi telah berhasil, namun jika koneksi gagal sistem akan terus menerus menampilkan teks yang menyatakan bahwa sistem sedang menghubungkan ke jaringan wifi.

Setelah koneksi modul wifi ke jaringan wifi berhasil maka akan 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 akan berhasil.

Setelah koneksi modul wifi ke broker MQTT berhasil maka akan masuk pada proses pengolahan data. Proses ini diawali dengan memeriksa kondisi apakah data telah tersedia atau tidak, jika tidak ada data yang diterima maka proses akan selesai, namun jika ada data yang diterima, maka akan masuk ke proses pengolahan data. Pada proses pengolahan data akan dilakukan pembagian data berdasarkan topik, topik yang akan digunakan ada 2 topik yaitu topik UID dan saldo, kedua topik inilah yang akan digunakan pada algoritma aplikasi android.

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

3.3.2 Algoritma Broker MQTT



Gambar 3.4 Algoritma Broker MQTT

Sistem ini dimulai dengan proses awal yaitu memeriksa koneksi antara modul wifi dan broker MQTT seperti gambar 3.4, jika koneksi gagal maka broker MQTT tidak akan menerima data apapun sehingga data yang ditampilkan adalah kosong,

sedangkan jika koneksi berhasil maka broker MQTT akan 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 akan menampilkan data secara berurutan dari data yang lama hingga data terbaru.

3.3.3 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 akan 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. Tabel 3.1 adalah contoh dari tabel daftar pembeli:

Tabel 3.1 Daftar Pembeli

UID	NAMA	SALDO

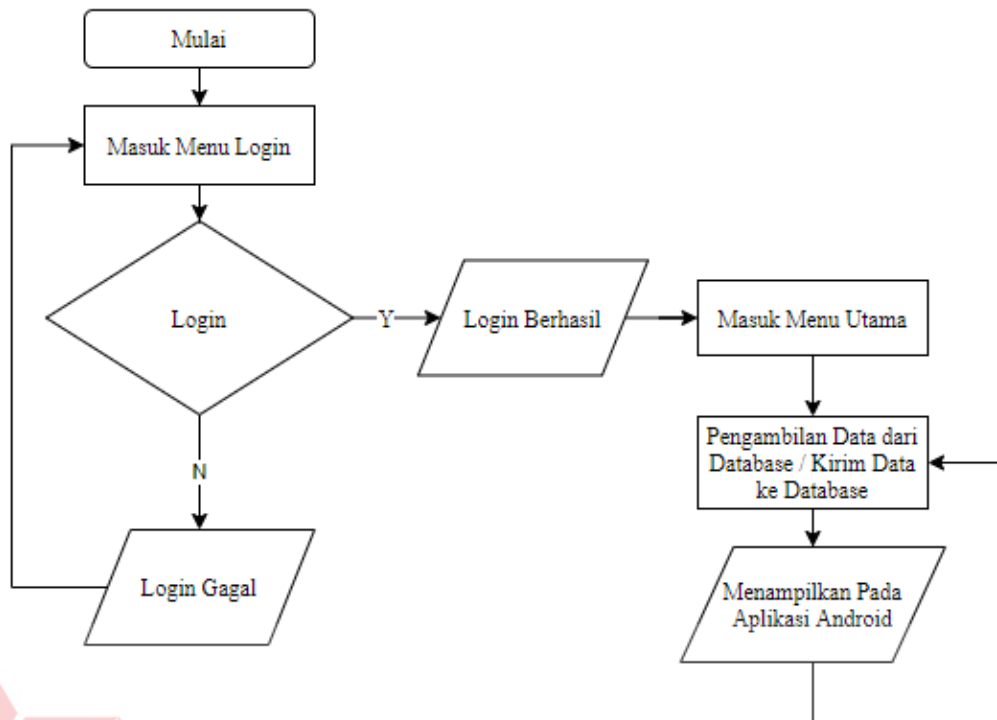
2. Tabel Riwayat Transaksi

Tabel terakhir adalah tabel riwayat transaksi yang mana berfungsi untuk memantau proses transaksi yang telah dilakukan pembeli. Tabel 3.2 adalah contoh dari tabel riwayat transaksi:

Tabel 3.2 Riwayat Transaksi

WAKTU	UID	NAMA	MAKANAN	SALDO

3.3.4 Algoritma Aplikasi Android



Gambar 3.6 Algoritma Aplikasi Android

Gambar 3.6 merupakan algoritma aplikasi android, sistem ini akan dibagi menjadi 2 menu yaitu menu login dan menu utama, proses pertama dimulai dengan menu login. Pada menu login terdapat teksbar untuk mengisi UID beserta data lainnya yang bertujuan untuk koneksi ke Database dan memastikan apakah UID telah terdaftar. Jika UID belum terdaftar maka akan muncul tampilan untuk mendaftar, namun jika sudah maka sistem akan mencoba koneksi Database dan masuk ke menu utama. Pada menu utama akan dilakukan proses pengambilan data di broker Database, pengambilan data dilakukan dengan cara memanggil UID yang akan digunakan. Setelah pengambilan data, proses selanjutnya adalah mengolah data tersebut dan kemudian data yang telah diolah akan ditampilkan pada aplikasi android.

3.4 Indikator Keberhasilan

3.4.1 Pengujian pengiriman data pembeli antara mikrokontroler NodeMCU ESP8266 dengan Broker MQTT

Pengujian ini dilakukan menggunakan mikrokontroler NodeMCU ESP8266 yang akan dikoneksikan ke broker MQTT melalui jaringan wifi. Indikator keberhasilan dari pengujian ini adalah mengirimkan data pembeli ke broker MQTT, apabila data pembeli dapat terkirim ke broker MQTT dengan baik, maka dapat disimpulkan bahwa pengujian telah berhasil.

3.4.2 Pengujian penyimpanan data pembeli ke database MySQL

Pada pengujian ini akan dilakukan pemeriksaan data pembeli yang terkirim ke database MySQL, seperti pada penjelasan sebelumnya bahwa data akan dibagi menjadi 2 tabel, jadi pada pengujian ini indikator keberhasilannya akan ditentukan dari berhasil atau tidaknya data yang terkirim ke 2 tabel tersebut.

3.4.3 Pengujian pengiriman data pembeli antara Database dengan Aplikasi Android

Pada pengujian ini akan dilakukan pemeriksaan pengiriman data pembeli antara Database dengan Aplikasi Android atau sebaliknya. Indikator keberhasilan dari pengujian ini adalah mengirimkan data pembeli dari Database menuju ke Aplikasi Android atau sebaliknya, apabila data pembeli dapat terkirim ke Aplikasi Android atau sebaliknya dengan baik, maka dapat disimpulkan bahwa pengujian telah berhasil.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Pengiriman Data UID dari Modul Wifi ke MQTT

Tujuan

Tujuan dari pengujian pengiriman data UID dari Modul Wifi ke MQTT adalah untuk memastikan UID yang dibaca oleh *RFID reader RC-522* pada *vending machine* dapat terkirim dengan baik.

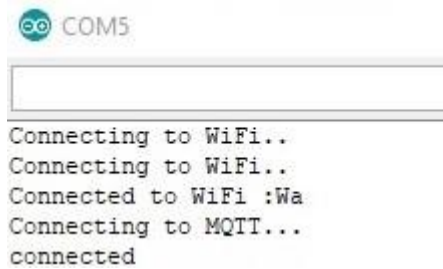
Peralatan yang Digunakan

1. Laptop untuk menjalankan Arduino IDE dan Broker MQTT
2. NodeMCU yang berfungsi sebagai Modul Wifi

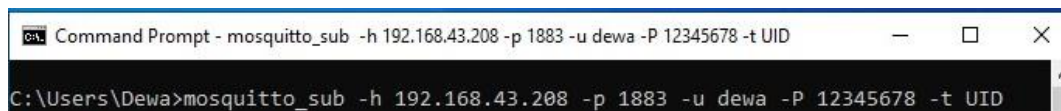
Cara Pengujian

1. Buka aplikasi Arduino IDE
2. Pastikan NodeMCU terhubung dengan jaringan yang sama dengan yang digunakan oleh Laptop
3. Upload program pada NodeMCU
4. Kirimkan data dari Arduino Mega ke NodeMCU dan pastikan datanya terkirim
5. Pastikan topik pada Broker MQTT sama dengan topik yang akan dikirim oleh NodeMCU
6. Buka Broker MQTT sebagai subscribe sehingga dapat menerima data dari NodeMCU
7. Amati data yang dikirim NodeMCU dan bandingkan dengan data yang diterima oleh Broker MQTT

Hasil Pengujian



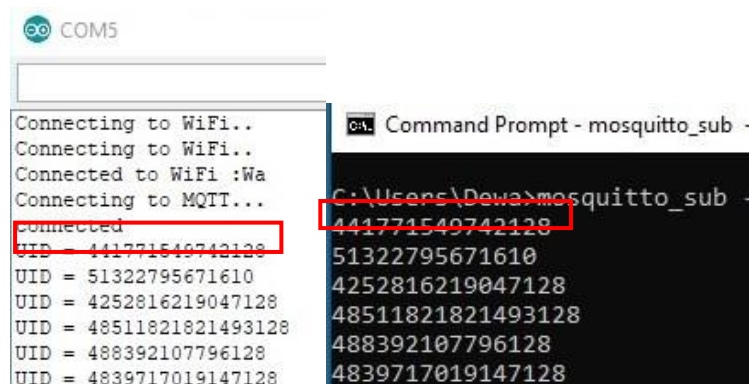
Gambar 4.1 Koneksi NodeMCU dan MQTT



Gambar 4.2 Broker MQTT Sebagai subscriber

Dapat dilihat pada Gambar 4.1 proses yang dilakukan adalah mengkoneksikan Broker MQTT menggunakan satu jaringan dengan host MQTT. Lalu pada Gambar 4.2 Broker MQTT dibuka sebagai subscriber yang akan menerima data dari NodeMCU sesuai dengan topik yang digunakan yaitu UID.

Setelah data terkirim dari NodeMCU yang berperan sebagai Modul Wifi, pastikan data juga diterima di Broker MQTT yang dapat dilihat pada Gambar 4.3. Jika data yang diterima sama dengan data yang dikirim, maka dapat dipastikan pengiriman data berhasil.



Gambar 4.3 Pengiriman Data UID dari NodeMCU ke Broker MQTT

Pada Tabel 4.1 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 4.1 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

Analisis Data

Dilihat dari hasil pengujian pengiriman data pada Tabel 4.1, 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%.

4.2 Pengujian Pengiriman Data Broker MQTT ke Database

Tujuan

Tujuan dari pengujian ini adalah memastikan data dari Broker MQTT dapat terkirim ke database, serta data dari database dapat di kirimkan Kembali ke Broker MQTT.

Peralatan yang Digunakan

1. Laptop untuk membuka Broker MQTT, Aplikasi Console C#, dan Database

Cara Pengujian

1. Pastikan pengiriman data pada pengujian 4.1 berhasil
2. Buka Aplikasi Console C# dan jalankan.
3. Buka Xampp, aktifkan modul apache dan mysql
4. Buka localhost/phpMyAdmin dan masuk ke database “vending_machine” tabel riwayat_transaksi
5. Lakukan pengiriman data UID seperti pada pengujian 4.1
6. Amati data yang ada di Broker MQTT dan bandingkan dengan data yang masuk pada Database

Hasil Pengujian

Sebelum memulai pengiriman data ke Database, pastikan Modul Wifi telah terkoneksi dengan jaringan yang sama dengan host Broker MQTT seperti pada pengujian 4.1. Setelah data dari NodeMCU terkirim ke Broker MQTT seperti pada pengujian 4.1, maka dapat dilakukan pengiriman dari Broker MQTT ke Database melalui aplikasi Console C# sebagai perantara untuk mengirim ke Database. Buka Aplikasi Console C# dan lakukan pengiriman data.

```

F:\KULIAH DEWA\MAKUL SMT 8\TUGAS AKHIR\Program\Jembatai
Program Jembatan Pengiriman MQTT ke Database
===== Data dari MQTT =====
-> UID = 488392107796128
===== Data dari Database =====
-> Nama = Amal
-> Saldo = 3500

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

===== Data dari MQTT =====
-> UID = 51322795671610
===== Data dari Database =====
-> Nama = Ardi
-> Saldo = 20000

===== Data dari MQTT =====
-> UID = 48511821821493128
===== Data dari Database =====
-> Nama = Capri
-> Saldo = 5000

===== Data dari MQTT =====
-> UID = 4839717019147128
===== Data dari Database =====
-> Tidak Ada Data Di Database

===== Data dari MQTT =====
-> UID = 441771549742128
===== Data dari Database =====
-> Tidak Ada Data Di Database

C:\Users\Dewa>mosquitto_sub -t #
488392107796128
4252816219047128
51322795671610
48511821821493128
4839717019147128
441771549742128
4427417822475128

```

Gambar 4.4 Data di Broker MQTT dan Data Pada Console C#

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

Gambar 4.5 Data pada Database

Pada Gambar 4.4 dapat disimpulkan Bahwa data dari Broker MQTT telah berhasil diterima oleh Aplikasi Console C#, yang mana Broker MQTT akan mengirimkan data UID yang telah diterima dari NodeMCU pada pengujian 4.1 ke Aplikasi Console C#. Kemudian data UID yang telah diterima oleh Console C#, dibandingkan ke data yang ada Database seperti pada gambar 4.5, 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 4.6 Perbandingan data yang terdaftar di Database

Jika UID yang dibandingkan terdaftar di Database, maka Aplikasi Console C# akan mengambil data Nama dan Saldo yang sebaris dengan UID tersebut. Contohnya ada pada gambar 4.6, yang mana Aplikasi Console C# akan 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 4.7.

```

===== 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 4.7 Perbandingan data yang tidak terdaftar di Database

Tabel 4.2 Data yang dikirim dari MQTT ke Database

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

Tabel 4.3 Data yang diterima oleh Database

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

Tabel 4.2 dan 4.3 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.

Analisis Data

Dari pengujian sebanyak 30 data yang dapat dilihat pada tabel 4.2 dan 4.3, pengiriman yang berhasil diterima oleh Database adalah 30 data. Sehingga dapat ditarik kesimpulan bahwa pengiriman dari Broker MQTT ke Database memiliki persentase keberhasilan 100%.

4.3 Pengujian Pengiriman Data dari Database ke Android

Tujuan

Tujuan dari pengujian ini adalah memastikan komunikasi dari Database menuju ke Android dapat berjalan dengan baik.

Peralatan yang Digunakan

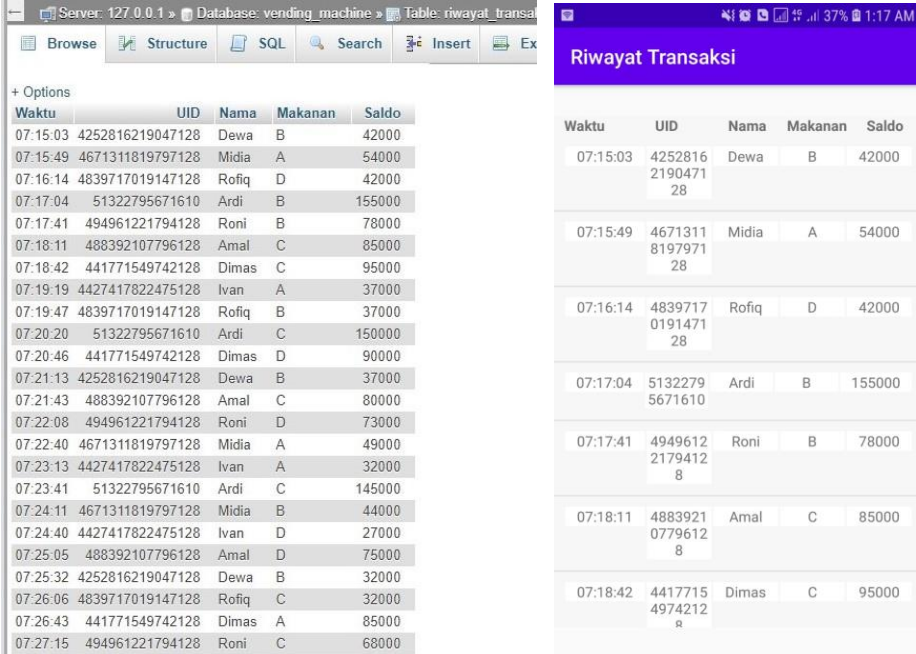
1. Laptop untuk membuka Aplikasi Console C# dan Database
2. Android untuk membuka Aplikasi Riwayat Transaksi

Cara Pengujian

1. Jalankan Aplikasi Console C# dan Database
2. Install Aplikasi Riwayat Transaksi pada Android
3. Lakukan Pengiriman Data seperti pada pengujian 4.1
4. Buka Aplikasi Riwayat Transaksi yang telah terinstall di Android
5. Amati data yang ada di Database, dan bandingkan dengan data yang diterima di Aplikasi Android.

Hasil Pengujian

Sebelum melakukan pengujian, pastikan telah melakukan pengiriman data seperti pada pengujian 4.1 dan 4.2 hingga data terkirim ke Database.



The image shows two side-by-side screenshots. The left screenshot is from a database management tool (likely phpMyAdmin) showing the 'riwayat_transaksi' table. The right screenshot is from an Android application titled 'Riwayat Transaksi', displaying the same data in a mobile-friendly format.

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 4.8 Data pada Database dan Aplikasi Android

Pada gambar 4.8 terdapat kumpulan data yang ada di Database dan diambil oleh Aplikasi Android. Data pada Database akan 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.

Lakukan 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 4.9, 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

Riwayat Transaksi

Waktu	UID	Nama	Makanan	Saldo
07:15:03	4252816 2190471 28	Dewa	B	42000
07:15:49	4671311 8197971 28	Midia	A	54000
07:16:14	4839717 0191471 28	Rofiq	D	42000

Gambar 4.9 Perbandingan Data dari Database dan Aplikasi Android

Hasil uji pengiriman data dari Database ke Aplikasi Android ditampilkan pada tabel 4.4 dan 4.5, pengujian ini dilakukan sebanyak 30 kali.

Tabel 4.4 Data pada Database

No	Database				
	Waktu	UID	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

No	Waktu	Database			
		UID	Nama	Makanan	Saldo
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
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 4.5 Data pada Aplikasi Android

No	Waktu	Android				Status
		UID	Nama	Makanan	Saldo	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

No	Waktu	Android			Saldo	Status Pengiriman
		UID	Nama	Makanan		
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

Analisis Data

Pada tabel 4.4 dan 4.5 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%.

4.4 Pengujian Pengiriman Data dari Android ke Database

Tujuan

Pengujian ini bertujuan untuk memastikan komunikasi antara Android menuju ke Database berjalan dengan baik, agar dapat melakukan pendaftaran UID yang belum terdaftar pada Database, sehingga pembeli dapat melakukan transaksi.

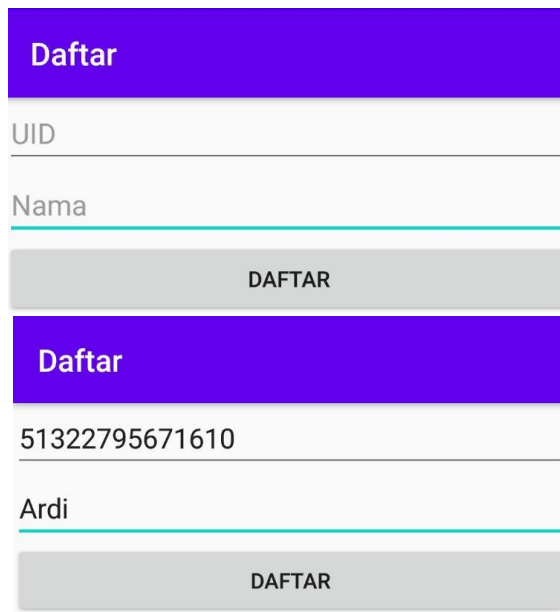
Peralatan yang Digunakan

1. Android untuk membuka Aplikasi Daftar
2. Laptop untuk membuka Database

Cara Pengujian

1. Buka Aplikasi Daftar pada android
2. Isi UID sesuai dengan yang ditampilkan di LCD *vending machine* serta isi Nama
3. Tekan Daftar
4. Amati hasil pendaftaran pada tabel daftar_pembeli pada Database

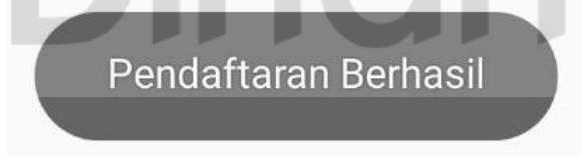
Hasil Pengujian



The image shows two screenshots of a registration application. The top screenshot shows the initial state with input fields for 'UID' and 'Nama' (Name), and a 'DAFTAR' (Register) button. The bottom screenshot shows the state after registration, with the UID field filled with '51322795671610' and the Nama field filled with 'Ardi', and the 'DAFTAR' button still present.

Gambar 4.10 Aplikasi Daftar

Gambar 4.10 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 4.11 Tampilan Pendaftaran Berhasil

Setelah menekan tombol Daftar maka akan muncul pesan bahwa pendaftaran selesai dilakukan, pesan seperti pada gambar 4.11.

+ Options			
	UID	Nama	Saldo
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	51322795671610	Ardi	0

Gambar 4.12 Data Terdaftar di Database

Setelah itu data akan masuk kedalam tabel `daftar_pembeli` yang ada di Database `vending_machine`. Pastikan data yang di daftarkan sama dengan yang terkirim ke tabel `daftar_pembeli` seperti gambar 4.12, jika data berbeda maka dapat

dikatakan bahwa pengiriman gagal. Gambar 4.13 merupakan contoh dari pendaftaran 8 E-KTP berbeda yang telah dilakukan.

+ Options				UID	Nama	Saldo
<input type="checkbox"/>	Edit	Copy	Delete	51322795671610	Ardi	0
<input type="checkbox"/>	Edit	Copy	Delete	441771549742128	Dimas	0
<input type="checkbox"/>	Edit	Copy	Delete	488392107796128	Amal	0
<input type="checkbox"/>	Edit	Copy	Delete	494961221794128	Roni	0
<input type="checkbox"/>	Edit	Copy	Delete	4252816219047128	Dewa	0
<input type="checkbox"/>	Edit	Copy	Delete	4427417822475128	Ivan	0
<input type="checkbox"/>	Edit	Copy	Delete	4671311819797128	Midia	0
<input type="checkbox"/>	Edit	Copy	Delete	4839717019147128	Rofiq	0

Gambar 4.13 Data Pada Tabel daftar_pembelian

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

Tabel 4.6 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

No	Android		Database			Status Pengiriman
	UID	Nama	UID	Nama	Saldo	
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

Analisis Data

Berdasarkan data yang diperoleh pada tabel 4.6, 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%.

4.5 Pengujian Seluruh Sistem

Tujuan

Pengujian ini bertujuan untuk memastikan seluruh sistem berkerja dengan baik dengan melakukan pengujian 4.1 sampai pengujian 4.4

Peralatan yang Digunakan

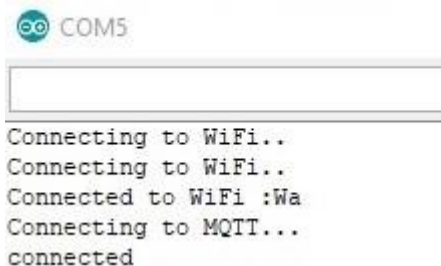
1. NodeMCU
2. Laptop
3. Android

Cara Pengujian

1. *Upload* program pada NodeMCU dan pastikan jaringan yang digunakan sama dengan jaringan laptop dan Broker MQTT
2. Buka XAMPP lalu aktifkan modul apache dan mysql
3. Buka Localhost, lalu buka database vending_machine di browser pada laptop

4. Buka Aplikasi Riwayat Transaksi pada android
5. Amati proses pengiriman data yang dilakukan dari Modul Wifi menuju ke Broker MQTT, lalu ke Database, kemudian ke Android. Atau proses pengiriman data sebaliknya dari Android ke Database, lalu ke Broker MQTT, dan kemudian diterima oleh Modul Wifi

Hasil Pengujian



```

COM5
Connecting to WiFi..
Connecting to WiFi..
Connected to WiFi :Wa
Connecting to MQTT...
connected

```

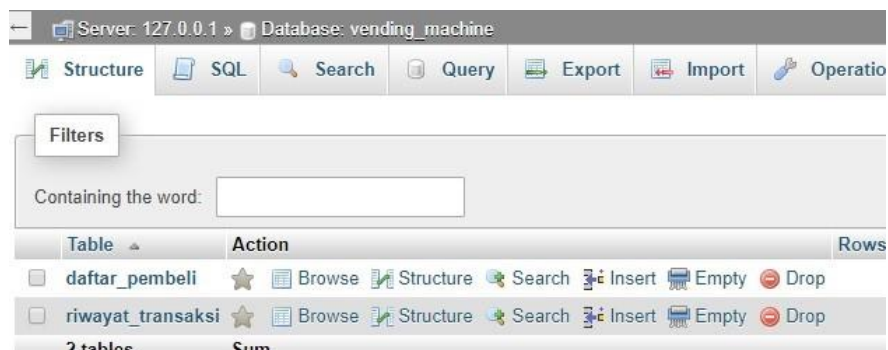
Gambar 4.14 Koneksi Modul Wifi dengan Wifi dan MQTT

Pastikan Modul Wifi terkoneksi dengan jaringan yang sama dengan jaringan MQTT dan Laptop, gambar 4.14 adalah tampilan serial monitor dari Modul Wifi yaitu NodeMCU.



Gambar 4.15 Aplikasi Console C#

Buka aplikasi Console C# seperti gambar 4.15 yang berfungsi sebagai perantara antara Broker dengan Database vending_machine seperti gambar 4.16.

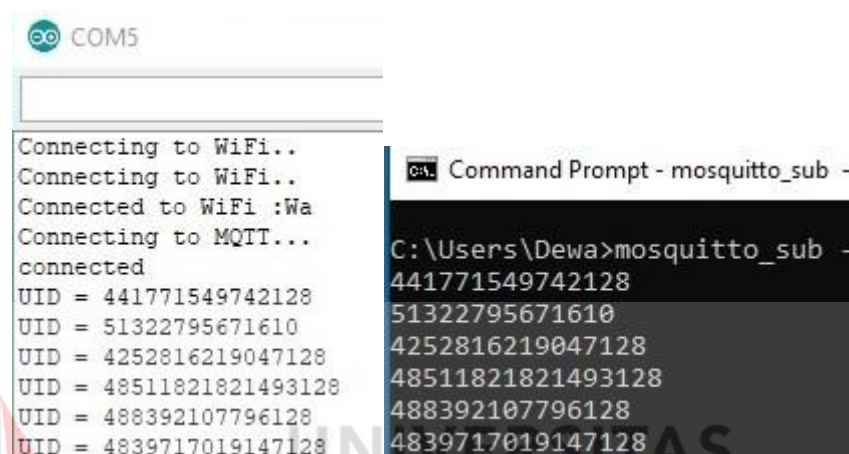


Gambar 4.16 Database

Pastikan 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 akan dikirim ke Modul Wifi, setelah Modul Wifi menerima data UID dari *vending machine*, data akan dikirim ke Broker MQTT. Contoh pengiriman data UID dari Modul Wifi ke Broker MQTT ada pada gambar 4.17.



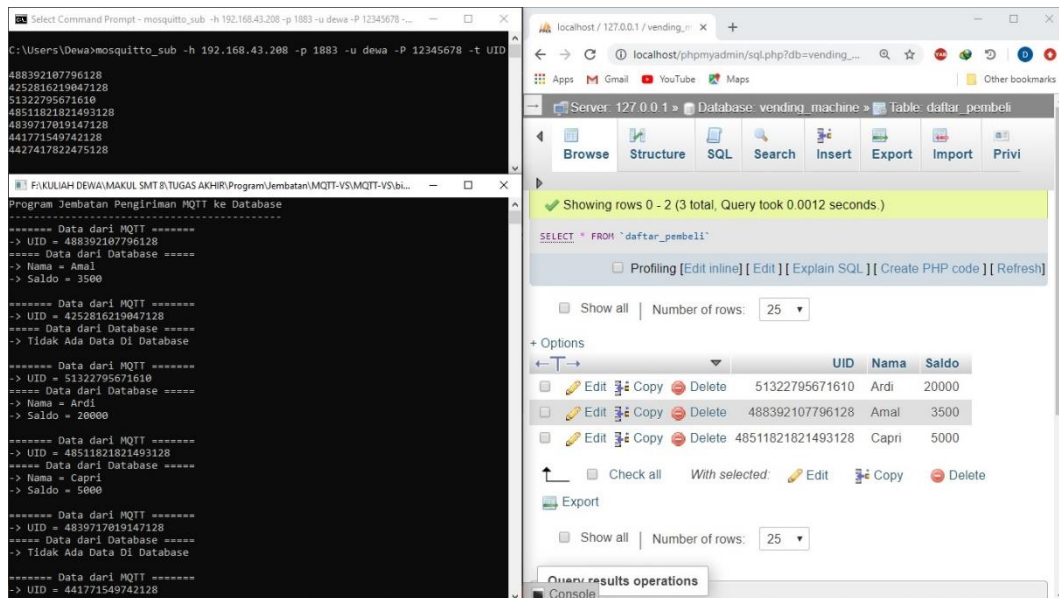
```

COM5
Connecting to WiFi..
Connecting to WiFi..
Connected to WiFi :Wa
Connecting to MQTT...
connected
UID = 441771549742128
UID = 51322795671610
UID = 4252816219047128
UID = 48511821821493128
UID = 488392107796128
UID = 4839717019147128

C:\Users\Dewa>mosquitto_sub -
441771549742128
51322795671610
4252816219047128
48511821821493128
488392107796128
4839717019147128
  
```

Gambar 4.17 Pengiriman Data dari Modul Wifi ke Broker MQTT

Setelah data terkirim ke Broker MQTT, data akan secara otomatis terkirim ke aplikasi console c# yang mana console c# berperan sebagai subscriber sehingga datanya akan terkirim seperti contoh pada gambar 4.18. Setelah data diterima oleh console c# data kemudian akan dikirim ke Database yang bertujuan untuk menanyakan apakah UID sudah terdaftar atau tidak. Jika tidak terdaftar, maka akan muncul pesan pada LCD yang ada di *vending machine* yang meminta pelanggan untuk mendaftarkan E-KTP terlebih dahulu.



Gambar 4.18 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:

1. E-KTP Terdaftar:

Lanjut ke proses transaksi yang mana pelanggan akan menekan tombol pada *vending machine*, lalu Modul Wifi akan menerima data saldo terbaru dari pelanggan. Setelah mendapatkan nilai saldo Modul Wifi akan melakukan update saldo pada Database dengan menggunakan komunikasi pengiriman yang sama seperti pemeriksaan UID sebelumnya. Dari gambar 4.19 dapat disimpulkan bahwa setelah melakukan transaksi kedua, saldo menjadi berkurang.

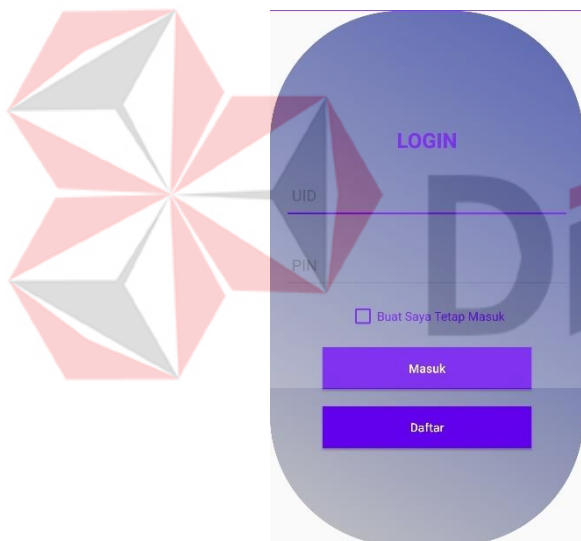
Waktu	UID	Nama	Makanan	Saldo
07:15:03	4252816219047128	Dewa	B	42000
07:15:49	4671311819797128	Midia	A	54000
07:16:14	4839717019147128	Rofiq	D	42000
07:17:04	51322795671610	Ardi	B	155000
07:17:41	494961221794128	Roni	B	78000
07:18:11	488392107796128	Amal	C	85000
07:18:42	441771549742128	Dimas	C	95000
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

Gambar 4.19 Tabel Riwayat Transaksi pada Database

2. E-KTP Tidak Terdaftar:

Ketika E-KTP belum Terdaftar, pelanggan akan 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 4.20, *user* dianjurkan untuk melakukan proses login terlebih dahulu jika ingin mengakses data transaksi. Selanjutnya akan ada 2 menu yang disediakan, yaitu menu login sebagai *owner* dan menu login sebagai *customer*. Pada menu login sebagai *customer*, *user* akan disajikan tampilan riwayat transaksi dari pemilik UID E-KTP yang telah terdaftar, tampilan Riwayat transaksi dapat dilihat pada gambar 4.21.

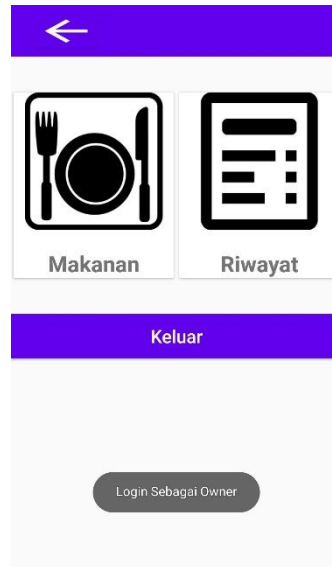


Gambar 4.20 Menu Login



Gambar 4.21 Menu Login *Customer*

Selanjutnya pada menu login sebagai *owner* akan tampil 2 pilihan menu yaitu menu riwayat transaksi, dan menu makanan. Dapat dilihat pada gambar 4.22, yaitu tampilan menu login sebagai *owner*.

Gambar 4.22 Menu Login *Owner*

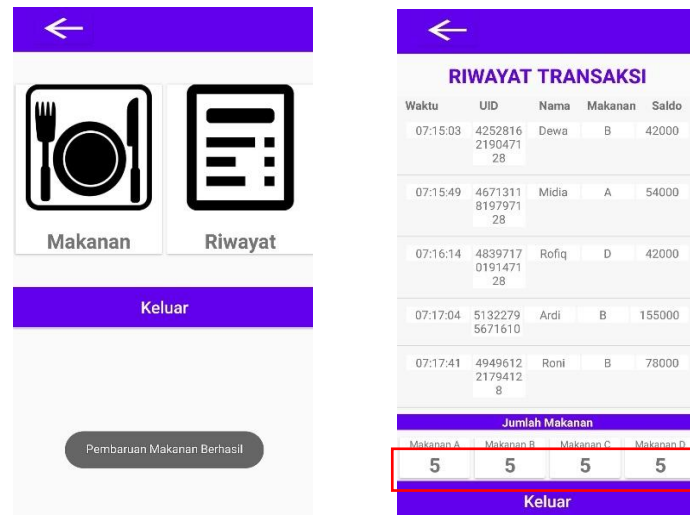
Waktu	UID	Nama	Makanan	Saldo
07:15:03	4252816 2190471 28	Dewa	B	42000
07:15:49	4671311 8197971 28	Midia	A	54000
07:16:14	4839717 0191471 28	Rofiq	D	42000
07:17:04	5132279 5671610	Ardi	B	155000
07:17:41	4949612 2179412 8	Roni	B	78000

Jumlah Makanan			
Makanan A	Makanan B	Makanan C	Makanan D
3	5	5	5

Gambar 4.23 Riwayat Transaksi

Menu riwayat transaksi yang dapat dilihat pada gambar 4.23 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 akan 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 4.24 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 akan otomatis terbaru pada menu riwayat transaksi seperti pada gambar 4.25.

Gambar 4.24 Tampilan Menu Makanan



Gambar 4.25 Pembaruan Jumlah Makanan

Analisis Data

Pengujian seluruh sistem terdiri dari gabungan antara pengujian sebelumnya yaitu pengujian 4.1, pengujian 4.2, pengujian 4.3, dan pengujian 4.4, sehingga hasil pengujian seluruh sistem akan bergantung pada keberhasilan pengiriman data pada pengujian sebelumnya. Dari seluruh pengujian pengiriman data utama yang telah dijelaskan pada pengujian 4.1, pengujian 4.2, pengujian 4.3, dan pengujian 4.4, 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.

BAB V

PENUTUP

5.1 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. Tugas akhir 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.

5.2 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.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Adiputra, D. S. (2015). Mesin Penjual Softdrink Otomatis Berbasis ATMega8535 . *ELEMENTER* , Vol. 1, No. 2.
- Giyartono, A. (2015). Aplikasi Android Pengendali Lampu Rumah Berbasis Mikrokontroler ATMEGA328. *TINF*, 003.
- Haryono, F. (2017). Rancang Bangun Alat Memilih Minuman Dengan Perintah Suara Berbasis Arduino. *Jurnal Ilmiah Go Infotech*, Volume 23 No. 1.
- 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*. Retrieved from Riwayat dan Asa Vending Mesin di Indonesia: <https://biz.kompas.com/read/2017/12/08/114616128/riwayat-dan-asa-vending-machine-di-indonesia>
- Kompas. (2019, 10 16). *Asal mula mesin penjual otomatis di Jepang*. Retrieved from Asal mula mesin penjual otomatis di Jepang: <https://travel.kompas.com/read/2019/10/16/100700427/ini-fakta-dan-sejarah-vending-machine-di-jepang?page=all>
- Nurrochman, A. R. (2019, Juli 22). *Rancang Bangun Vending Machine Minuman Berkarbonasi Berbasis Arduino Mega 2560*. Retrieved from 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.