

Pengiriman Data Berbasis Internet of Things untuk Monitoring Sistem Hemodialisis Secara Jarak Jauh

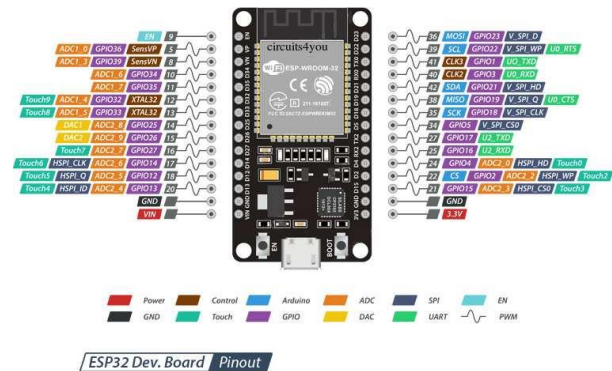
Ainun Arsyi Sahifa, Rachmad Setiawan, dan Muhammad Yazid
 Departemen Teknik Biomedik, Institut Teknologi Sepuluh Nopember (ITS)
 e-mail: rachmad@ee.its.ac.id

Abstrak—Hemodialisis merupakan alat yang sangat penting, karena alat ini membantu manusia yang mengalami gangguan pada fungsi ginjalnya. Sehingga Hemodialisis, merupakan alat sebagai pengganti fungsi ginjal pada manusia. Pada penelitian ini dilakukan pengembangan pada Hemodialisis sehingga lebih meningkatkan pelayanan rumah sakit yang diberikan kepada pasien dan memudahkan petugas rumah sakit dalam melakukan penjangaan. Penelitian ini dapat mengirimkan data Hemodialisis supaya pasien dan petugas rumah sakit dapat melihat data pada hemodialisis, selain itu pada penelitian ini menambahkan fitur untuk memantau pasien dari jarak jauh (*Monitoring System*) dengan memanfaatkan konsep *Internet of Things* menggunakan WiFi dan MQTT dengan Arduino ESP32 dan mengimplementasikan penelitian ini dalam *Platform IoT*. Sistem yang dikembangkan telah berhasil untuk mengirimkan data dengan menggunakan protokol MQTT yang dibuat sesuai dengan rancangan. Pada penelitian ini yang digunakan sebagai *publisher* adalah mikrokontroler yang berfungsi untuk menerima data nilai sensor yaitu ESP32 (*master*) dan ESP32 (*slave*) yang dihubungkan secara serial, lalu *broker* yang digunakan adalah lokal MQTT broker pada ThingsBoard dan *subscriber* pada ThingsBoard saat *log-in* menjadi admin atau customers. Lalu web server ThingsBoard sudah berjalan dengan baik dengan dibuktikan, pesan dapat terpublish dan ThingsBoard dapat menampilkan nilai tekanan darah, aliran darah, temperatur dialisat, dan konduktivitas cairan dialisat.

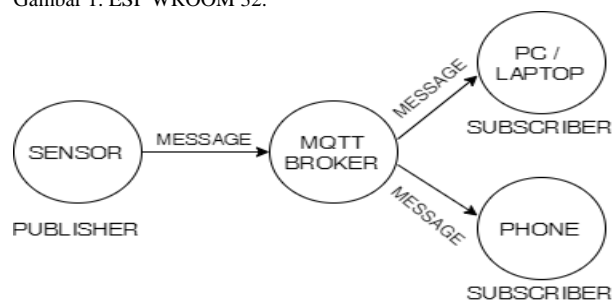
Kata Kunci— ESP32, Internet of Things, Monitoring System, MQTT, Wi-Fi

I. PENDAHULUAN

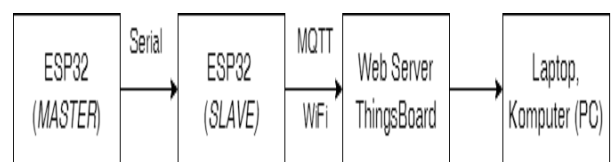
GINJAL merupakan organ yang penting di dalam tubuh kita. Fungsi utama dari ginjal yaitu untuk mengatur cairan dalam tubuh dan menyeimbangkan elektrolit, dimana fungsi tersebut untuk mempertahankan *volume* cairan dan komposisi ion *Intracellular* dan *Extracellular* [1]. Saat ginjal tidak bisa berfungsi dengan baik, cairan akan tertahan dan beberapa ion zat terlarut akan menumpuk. Ada dua kemungkinan opsi untuk menyelamatkan pasien yang terkena penyakit ginjal. Opsi yang pertama yaitu dengan melakukan transplantasi ginjal atau datang secara rutin untuk membersihkan darahnya dengan alat dialysis [1]. Dengan banyaknya pasien yang lebih memilih untuk melakukan cuci darah daripada transplantasi ginjal dikarenakan karena resiko yang mereka takut hadapi. Dengan banyaknya pasien yang lebih memilih untuk cuci darah, maka perlu dilakukan penambahan alat pencuci darah (Hemodialisis) dan melakukan beberapa inovasi atau pembaharuan, supaya dapat meningkatkan kualitas pelayanan untuk pasien di rumah sakit yang terkait. Dalam rangka meningkatkan kualitas pelayanan, maka dilakukan peningkatan disesuaikan dengan keadaan teknologi masa kini, yang sedang banyak digunakan, yaitu



Gambar 1. ESP WROOM 32.



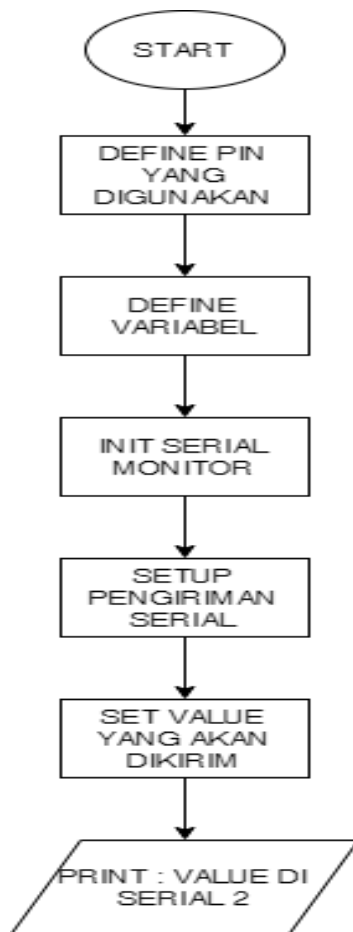
Gambar 1. Alur Protokol MQTT



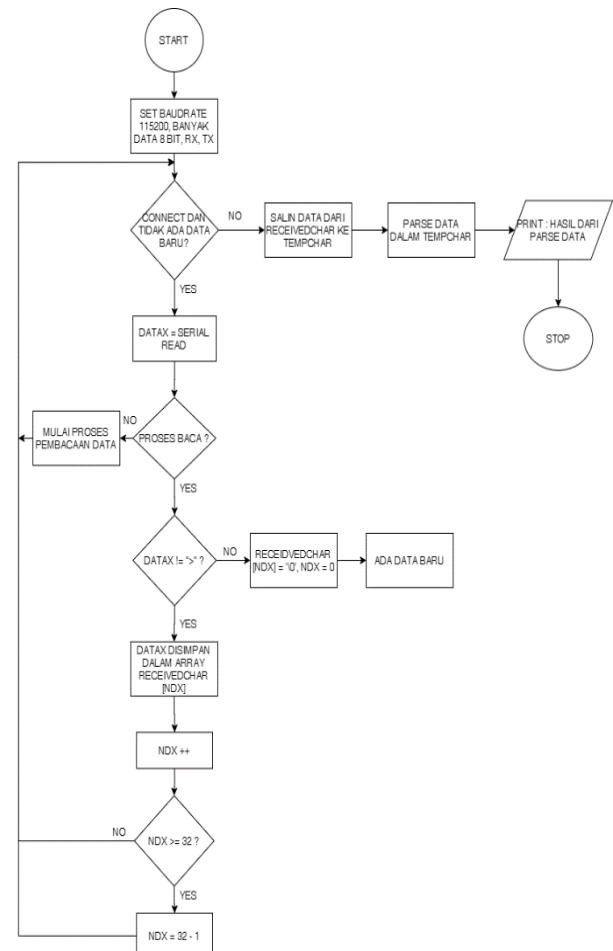
Gambar 2. Diagram Blok Sistem.

teknologi bernama *IoT* (*Internet of Things*). Pengiriman hasil data menggunakan metode IoT, yang memanfaatkan teknologi sekarang.

Internet of Things (IoT) merupakan sistem “*Smart Device*” yang dapat menghubungkan dan melakukan komunikasi dengan *users* dan sistem yang lainnya [2]. Sehingga dapat mengintegrasikan ke dalam sistem komputerisasi, menghasilkan hasil dengan lebih efisien, aplikasi baru, dan dapat menumbuhkan pertumbuhan ekonomi. Untuk aplikasi IoT sudah sangat banyak diterapkan dalam berbagai bidang, contohnya yaitu pada bidang kesehatan, rumah, kendaraan, infrastruktur kota, industri, *agricultural* (pertanian), dan *wild life*. Biasanya alat yang digunakan dihubungkan pada mikrokontroler, sensor, aktuator, dan konektivitas internet [3] [4]. Dihubungkan ke mikrokontroler agar terbentuk suatu *embedded system*. Yang dimaksud *embedded system* yaitu sistem yang sudah tertanam, yang didesain untuk melakukan satu atau lebih fungsi yang *real-time*. Jika alat itu menjadi bagian dari *embedded system*, maka jika ada kerusakan pada



Gambar 4. Diagram Blok Komunikasi Serial.



Gambar 3. Diagram Blok Parsing Data Esp32 (slave).

alat, alat tersebut harus dibongkar dan diset ulang. Sedangkan untuk sensor yaitu untuk mendeteksi benda yang akan diambil datanya, supaya nanti data tersebut dapat ditransmisi. Sedangkan untuk aktuator, karena fungsinya untuk menggerakkan atau mengontrol suatu sistem, maka aktuator juga bisa dikendalikan dengan IoT. Dan yang terakhir, jika ingin menerapkan IoT, maka harus terhubung dengan internet. Data yang diambil dari mikrokontroler dengan ESP32 akan mengirimkan ke *database local (ThingsBoard)*, dimana hasil dari data itu akan ditampilkan pada tampilan *web server ThingsBoard*.

II. TINJAUAN PUSTAKA

A. Wifi Module

Modul *WiFi* yang digunakan adalah ESP32. Dimana ESP32 ini merupakan generasi setelah ESP8266 yang memiliki beberapa kelebihan dari ESP8266. Diantaranya adalah, memiliki inti CPU serta *WiFi* yang lebih cepat, GPIO yang lebih, dan dapat digunakan sebagai *Bluetooth* juga. Dengan perincian 32-bit prosesor, dengan frekuensi 160kHz, dan memiliki 52kB RAM [5]. Untuk spesifikasi yang dipakai pada penelitian ini adalah ESP WROOM 32, dapat dilihat pada Gambar 1.

B. Hemodialisis

Hemodialisis adalah salah satu perawatan yang dapat digunakan untuk menapis zat sisa dan kelebihan air dari darah, layaknya kerja ginjal yang sehat. Hemodialisis membantu untuk mengatur tekanan darah dan keseimbangan mineral-mineral penting dalam darah. Selain itu, hemodialisis

dapat membantu seseorang untuk bertahan hidup lebih lama dan menambah opsi untuk pasien selain melakukan transplantasi ginjal sehingga mengurangi resiko yang ditakuti oleh pasien, namun tidak menyembuhkan gagal ginjal. Dalam sistem hemodialisis terdapat empat komponen yang penting untuk dilakukan monitoring. Empat komponen itu yaitu,

1) Sirkuit Darah

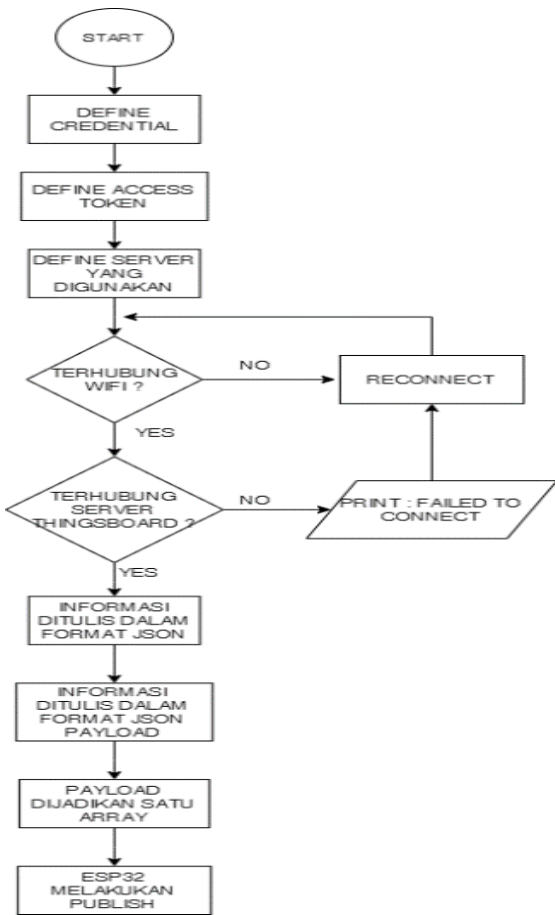
Pada sirkuit ini, pasien dimasukkan jarum atau kateter untuk dapat mengambil darah dari dalam tubuh pasien. Fungsi dari komponen ini yaitu untuk mengirimkan darah ke dalam *dialyzer* pada aliran darah tertentu dan mengembalikan darah yang sudah masuk dari *dialyzer* Kembali lagi ke dalam tubuh. Sehingga membuat aliran darah berperan penting dalam pembersihan urea (*urea clearance*), dimana dengan cara ini dapat mengetahui kemampuan fungsi ginjal

2) Dialyzer

Dialiser merupakan komponen utama dalam proses hemodialisis, karena yang dilakukan oleh dialiser Sebagian besar dilakukan oleh ginjal normal pada umumnya. Di dalam proses ini terjadi pertukaran zat-zat dan cairan dalam darah dan dialisat, maka letak dari dialiser ini ada di tengah-tengah antara sirkuit darah dan sirkuit dialisat.

3) Ultrafiltrasi

Pada proses hemodialisis terjadi mekanisme ultrafiltrasi, mekanisme ini bertujuan untuk mengurangi kelebihan cairan dalam dalam tubuh (*volume control*). Pada mekanisme ultrafiltrasi terjadi pembuangan cairan karena adanya perbedaan tekanan antara sirkuit darah dan sirkuit dialisat, jika tekanan antara sirkuit darah dan sirkuit dialisat seimbang maka mekanisme ultrafiltrasi akan berhenti.



Gambar 8. Diagram Blok Pengiriman ke Web Server ThingsBoard.

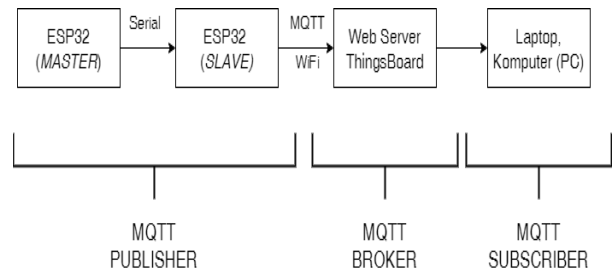
4) Sirkuit Dialisat

Dialisat merupakan cairan yang terdiri dari air dan elektrolit dari serum yang dipompa melewati dialiser ke darah pasien. Komposisi cairan dialisat diatur disesuaikan hingga mendekati komposisi ion darah normal dan sedikit dimodifikasi supaya dapat memperbaiki elektrolit pasien (d disesuaikan dengan pasien). Dialisat dibuat dengan mencampurkan konsentrat elektrolit dengan bikarbonat dan air murni. Konduktivitas merupakan hal yang penting, karena komposisi monitor dialisat berdasarkan konduktivitasnya. Sehingga jika salah menggunakan konsentrasinya maka komposisinya juga akan salah. Lalu dialisat yang dianjurkan untuk digunakan adalah dialisat bikarbonat untuk mengurangi komplikasi [6].

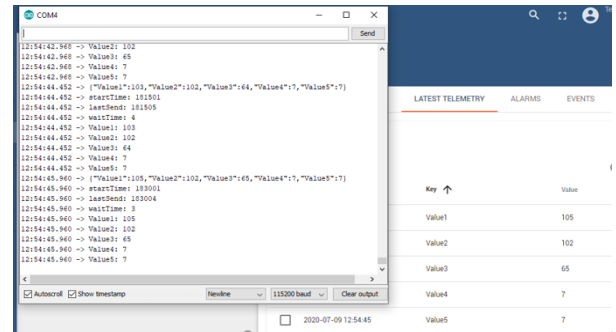
C. MQTT

Kepanjangan dari MQTT adalah *Message Queuing Telemetry Transport*. MQTT merupakan protokol pesan *publish/subscribe* yang ringan. MQTT merupakan salah satu protokol utama untuk penyebaran IoT (*Internet of Things*). Dalam protokol MQTT semua perangkat yang berlangganan maka disebut dengan klien. Semua klien dapat menyebarkan pesan, pesan berasal dari *broker* [7]. Pesan di sini yang dimaksud adalah data dari Hemodialisis yang perlu ditransmisikan. Sebelumnya disebutkan mengenai *broker*, untuk *broker* MQTT adalah untuk menyaring pesan data berdasarkan yang diperlukan lalu menyebarkan data pada pengakses untuk penelitian ini.

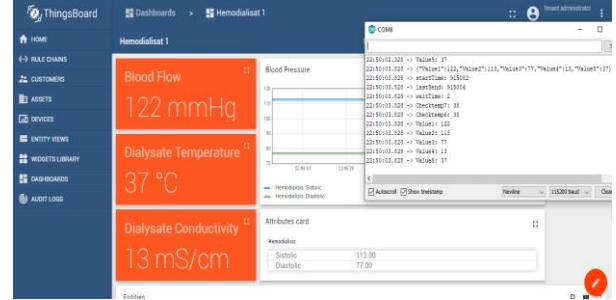
Dalam MQTT terdapat beberapa istilah yang biasa digunakan yaitu, (a)MQTT *Client* : Aplikasi *client* yang terhubung ke internet dan mengimplementasikan



Gambar 5. Rancangan Sistem Pengiriman Data Hemodialisis Berbasis Internet of Things.



Gambar 6. Hasil *publish* data pada topik *Telemetry*



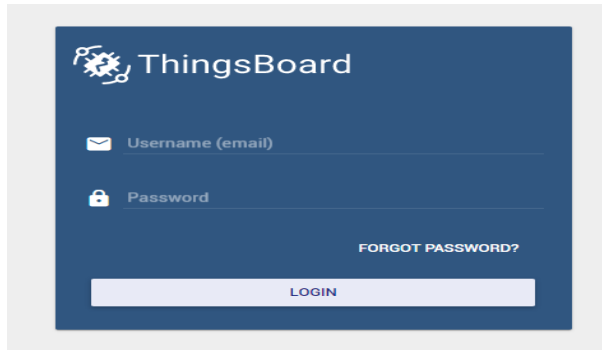
Gambar 7. Hasil dari pengiriman MQTT.

MQTT di atas TCP/IP untuk mengirim atau menerima pesan.; (b)MQTT Topic : Identifikasi pesan yang digunakan untuk menandai, mengklasifikasikan pesan dalam konsep hierarkis.; (c)Publisher : MQTT Client yang mengirim data melalui jaringan; (d)Subscriber MQTT yang meng-Subscribe (meminta) topik tertentu di jaringan MQTT; (e)MQTT Broker : Broker utamanya adalah server aplikasi yang mengontrol MQTT client dengan konektivitas, otentikasi, pengiriman pesan dan penerimaan pesan; (f)Message : Data yang sedang dimunculkan (Publish).; (g)Konektivitas, otentikasi, pengiriman pesan dan penerimaan pesan; (h)Message Data yang sedang dimunculkan (Publish).

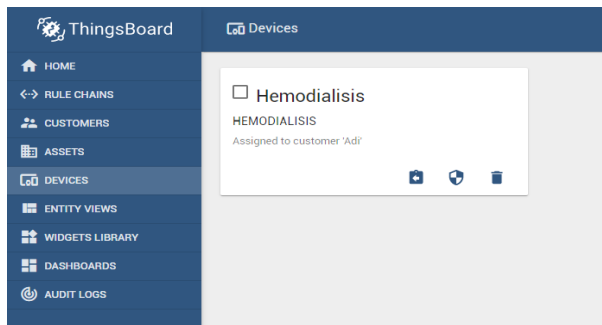
Untuk berkomunikasi dengan menggunakan protokol MQTT, *Publisher* (seperti data, sensor, *embedded system*) harus mendefinisikan dua elemen yang akan masuk ke MQTT *Broker*, dua elemen itu adalah pesan dan topik. Pesan merupakan data string yang ingin dibagikan *publisher* (penerbit) kepada *subscriber* (pelanggan) melalui MQTT *Broker*. Sementara itu, topik merupakan string yang digunakan oleh *Broker* untuk menyaring dan memutuskan pelanggan mana yang harus menerima pesan mana [8]. Suatu topik dapat terdiri dari suatu level topik, dimana masing-masing topik dapat dipisahkan dengan tanda “ / “, contohnya suatu karbon monoksida (CO) yang terletak di tempat parkir suatu toko perbelanjaan, dapat mempublikasikan (*Publish*) tingkat karbon monoksida menggunakan topik hirarkis

```
22:20:58.083 -> {"Value1":156,"Value2":119,"Value3":76,"Value4":6,"Value5":6}
22:20:58.083 -> startTime: 826672
22:20:58.083 -> lastSend: 826674
22:20:58.083 -> waitTime: 2
22:20:59.069 -> Value1: 156
22:20:59.104 -> Value2: 119
22:20:59.104 -> Value3: 76
22:20:59.104 -> Value4: 6
22:20:59.104 -> Value5: 6
```

Gambar 9. Hasil keluaran nilai dalam JSON file.



Gambar 10. Tampilan Login

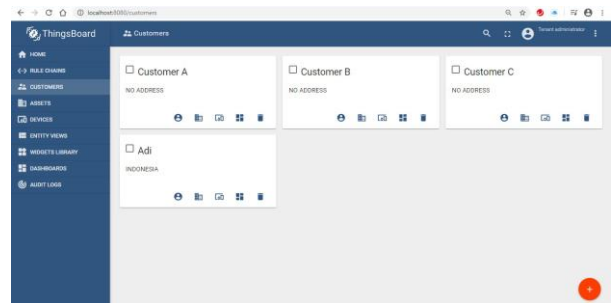


Gambar 11. Tampilan Alat.

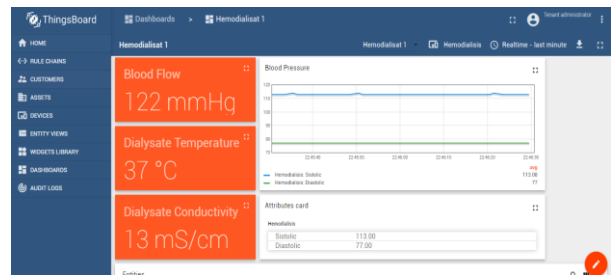
D. Shopping Mall/Parking Lot/CO

Setelah itu, data tersebut akan masuk ke dalam MQTT Broker, dimana Broker akan memberikan keputusan mana saja Subscriber yang dapat mengakses data tersebut. Para client akan mendapatkan data atau menerima pesan dengan topik yang sama ketika mereka melakukan subscribe. Untuk lebih jelas mengenai alur dari protokol MQTT dapat dilihat pada Gambar 2. Dimana ada suatu publisher yang akan mempublikasikan suatu informasi (sensor) maka terdapat dua pelanggan (Subscriber) yang akan mendapatkan pesan atau data tersebut. Dengan demikian, Subscriber untuk topik sensor dapat mengakses dan mendapat informasi mengenai datanya.

Dalam penggunaannya komunikasi dengan protokol MQTT sekarang lebih banyak dipakai daripada HTTP, karena protokol MQTT menggunakan daya yang lebih sedikit dari protokol HTTP pada setiap jamnya saat pengiriman data, artinya protokol MQTT dapat mengurangi konsumsi daya dan juga lebih cepat dari protokol HTTP[8], selain itu dengan menggunakan protokol MQTT ini juga mempunyai keamanan yang cukup, karena terdapat Topic yang dapat menunjukkan secara spesifik data mana yang mau dilihat dan Topic mempunyai sifat sensitif, dimana besar kecilnya huruf sangat mempengaruhi dan juga terdapat Broker yang fungsinya tidak hanya langsung menyebarkan informasi data,



Gambar 12. Tampilan Daftar Customer.



Gambar 13. Tampilan Dashboard.

entity_id	key	long_v	
3a557fe0-94fd-11ea-ba7e-5f711f15570b		7	120
3a557fe0-94fd-11ea-ba7e-5f711f15570b		8	109
3a557fe0-94fd-11ea-ba7e-5f711f15570b		9	69
3a557fe0-94fd-11ea-ba7e-5f711f15570b		10	7
3a557fe0-94fd-11ea-ba7e-5f711f15570b		11	7
3a557fe0-94fd-11ea-ba7e-5f711f15570b		7	119
3a557fe0-94fd-11ea-ba7e-5f711f15570b		8	109
3a557fe0-94fd-11ea-ba7e-5f711f15570b		9	69
3a557fe0-94fd-11ea-ba7e-5f711f15570b		10	7
3a557fe0-94fd-11ea-ba7e-5f711f15570b		11	7

Gambar 14. Hasil pada Database.

melainkan disaring terlebih dahulu siapa saja yang dapat mengakses informasi data tersebut.

III. PERANCANGAN SISTEM

A. Diagram Blok Sistem

Untuk Penelitian yang dilakukan dibagi menjadi dua bagian untuk menghasilkan hasil sesuai yang diinginkan. Pembagiannya yaitu terdapat Hardware dan Software, karena menggunakan Arduino, maka terdapat pembagian Hardware. Namun pada system ini akan lebih memperhatikan pembagian Software, karena rancangan penelitian ini berbasis Web server dimana harus membuat perancangan bagaimana berjalannya Web Server ThingsBoard, dan melakukan pemrograman untuk Arduino ESP32 sendiri. Rencana rancangan penelitian ini dapat dilihat pada Gambar 3. Dimana data akan diambil dari sensor-sensor oleh ESP32, setelah itu dari ESP32 pertama data ditransmisikan ke ESP32 yang kedua lalu akan diproses dan diteruskan ke web server ThingsBoard untuk dapat menampilkan datanya melalui protokol MQTT. Setelah itu, data dapat diakses pada perangkat keras yang sesuai di ThingsBoard.

B. Software

Dalam penelitian ini yang dimaksud dengan software yaitu tentang isi dari program-program dari keseluruhan sistem. Pada sub bab ini akan menjelaskan program-program apa saja

yag telah dibuat sehingga terbuatlah suatu sistem yang diinginkan.

1) Program Komunikasi Serial antar ESP32

Setelah mengambil data dari potensiometer dan ditampilkan dalam ESP32 (*master*) hal selanjutnya yang akan dilakukan adalah mengirim data dari ESP32 (*master*) ke ESP32 (*slave*) dengan menggunakan metode komunikasi serial, dapat dilihat pada Gambar 4.. Dimana kaki Tx,Rx antar ESP32 disambungkan dengan kebalikannya, sedangkan *ground* disambungkan ke *ground*, dan 3.3 v disambungkan 3.3 v. Dalam melakukan *setup* supaya dapat melakukan pengiriman serial, maka yang harus dilakukan adalah menuliskan `Serial2.begin(115200, SERIAL_8N1, Rx, Tx)`. Lalu ketika sudah melakukan set nilai yang akan dikirim (analogread potensiometer) dilakukan `Serial2.print` terhadap nilai yang sudah di set tadi, sehingga nilai tadi dapat ditampilkan di ESP32 yang kedua (*slave*).

Selanjutnya, Ketika ESP32 (*master*) mengirimkan secara serial ke ESP32 (*slave*), ESP32 (*slave*) melakukan proses parsing data (pembacaan data satu-satu/pemisahan data) terlebih dahulu, karena saat ESP32 (*master*) mengirimkan data ke ESP32 (*slave*) masih dalam satu array yang sama. Untuk dapat mengetahui jalannya parsing data dapat dilihat pada Gambar 5. Dalam melakukan parsing data, selama sudah tersambung dan tidak ada data baru, maka *datax* mulai membaca serial. Jika sudah mulai melakukan proses serial *read*, akan membaca array yang masuk, jika pembacaan *datax* tidak sama dengan tanda ">" maka *datax* disimpan dalam array `receiverchar[ndx]`, jika *datax* sama dengan tanda ">" maka ada data baru. Selama *datax* tidak sama dengan tanda ">" maka akan terus melakukan *increment* hingga *ndx* sama dengan 32. Jika ada data baru, maka data tadi di salin ke dalam *tempchar*, lalu dilakukan parsing data dalam *tempchar* dengan memisahkan nilainya, lalu hasil nilai diprint hasil data *parse*.

2) Program mengirim data ke ThingsBoard

Sebelum memulai memrogram di ESP32, ada beberapa *setup* yang harus dilakukan terlebih dahulu untuk mendukung ESP32 dapat melakukan pengiriman data ke *ThingsBoard*. Pengiriman data dilakukan oleh ESP32 (*slave*) dengan cara mengirim data melalui jaringan wifi yang ada di sekitarnya. Selain itu, untuk mengirim data ke *web server ThingsBoard* menggunakan metode MQTT. Sebelum mengirim ke *web server ThingsBoard*, terlebih dahulu harus memasukkan topik. Algoritma dalam melakukan pengiriman data dapat dijelaskan dengan proses berikut dan dapat dilihat pada Gambar 6.

- Proses 1 : Menuliskan *credential* dari wifi yang akan digunakan
- Proses 2 : Menuliskan access token (tujuan data yang akan dikirim)
- Proses 3 : Menuliskan server yang digunakan (IP)
- Proses 4 : Melakukan koneksi terhadap WiFi
- Proses 5 : Melakukan koneksi ke *ThingsBoard* server
- Proses 6 : Jika belum terhubung melakukan koneksi ulang selama 5 detik
- Proses 7 : Hasil dari serial data dituliskan ke dalam payload (pesan)
- Proses 8 : Payload dijadikan dalam satu array
- Proses 9 : ESP32 melakukan publish ke topik yang sudah ditentukan

3) Protokol MQTT (Message Queuing Telemetry Transport)

Karena penelitian ini merupakan penelitian yang berbasis IoT (*Internet of Things*) maka protokol yang baik digunakan yaitu protokol MQTT. Untuk perancangan protokol MQTT pada penelitian ini dapat dilihat pada Gambar 7. MQTT merupakan protokol yang dilakukan di TCP/IP (*Transmission Control Protocol/Internet Protocol*) dimana merupakan suatu komunikasi data di dalam jaringan internet. Kerja dari MQTT ini yaitu menerapkan *Publish* dan *Subscribe* data, lalu akan terhubung dengan MQTT *Broker* yang mempunyai suatu topik tertentu. Pada penelitian ini, seperti yang ditunjukkan di Gambar 7, dapat dilihat bahwa yang bekerja sebagai MQTT *Publisher* yaitu ESP32 (*master*) dan ESP32 (*slave*). MQTT *Publisher* ESP32 (*master*) dan ESP32 (*slave*) karena dapat mengirimkan data sensor ke MQTT *Broker* dan akan diterima oleh *Subscriber*.

MQTT *Publisher* biasanya merupakan alat yang terhubung dengan beberapa sensor. Sedangkan yang berperan sebagai MQTT *Broker* yaitu *ThingsBoard*, dimana pada *ThingsBoard* yang mengontrol berjalannya data *publish* dan *subscribe* dari berbagai alat. Sebagai MQTT *Broker*, untuk dapat mengkoneksikan *ThingsBoard* perlu menuliskan server dari *ThingsBoard* tersebut. Setelah itu yang bekerja sebagai MQTT *Subscriber* adalah Laptop dan Komputer petugas rumah sakit. MQTT *Subscriber* ini bekerja sebagai penerima data dari *Publisher*, biasanya MQTT *Subscriber* ini merupakan bentuk monitoring dari sensornya tersebut dimana nanti juga sebagai *Subscriber* akan meminta data dari *Publisher*. Untuk dapat menjalankan protokol MQTT maka diperlukan topik, untuk penelitian ini topik yang digunakan yaitu,

v1/devices/me/telemetry

Penamaan topik, bebas diberi nama apa saja sesuai dengan pemahaman yang membuat programnya. Jadi pada penelitian ini MQTT berfungsi untuk mengirimkan data sensor ke petugas rumah saki, dokter dan kerabat pasien yang untuk dapat melihat secara langsung salah satu dari kerabat atau pasien yang sedang melakukan pengobatan. Supaya data dapat dikirimkan, harus membuat file JSON. Dimana isi dari JSON merupakan informasi mengenai satu pesan yang berisi informasi data sensor. Program yang sudah dijalankan di Arduino IDE dapat memulai (menjalankan) protokol MQTT untuk mengirimkan data ke *thingsboard*.

IV. HASIL DAN PEMBAHASAN

A. Koneksi ESP32 ke ThingsBoard dan Protokol MQTT

Pengujian ketiga ini, melakukan pengujian terhadap program arduino untuk memastikan bahwa program arduino dapat mengirimkan data ke *ThingsBoard* dan pemantauan dapat dilakukan. Untuk hasil dari pengujian dapat dilihat pada Gambar 8, dimana ditunjukkan dari program arduino sudah bisa mengirimkan data ke *ThingsBoard*. Karena data dapat dikirimkan ke *ThingsBoard* dan dapat diakses ke komputer untuk melihat data pasien. Protokol ini berjalan dengan baik

karena dapat menampilkan data tekanan darah (sistol dan diastol), aliran darah, temperature dialisat dan konduktivitas dialih Gambar 9.

ESP32 berlaku sebagai Publisher yang akan mengambil data dari sensor dimana akan diteruskan ke MQTT broker, dimana pada penelitian ini yang bertindak sebagai MQTT Broker adalah ThingsBoardnya, lalu akan mengirimkan data ke perangkat keras jika perangkat keras melakukan subscribe. Dengan hasil yang ditunjukkan pada Gambar 8 dan Gambar 9, bahwa nilai potensiometer (sensor) yang banyak dapat dimasukkan ke dalam satu pesan yang diawali oleh tanda kurung kurawal “{”. Dimana nilai dari sensor-sensor tersebut siap untuk dikirim ke ThingsBoard. Untuk dapat mengirimkan data ke ThingsBoard, maka harus menuliskan topik terlebih dahulu supaya dapat memunculkan nilai pada ThingsBoardnya. Untuk topik yang digunakan yaitu “v1/devices/me/telemetry” Gambar 8.

Dimana pada program Arduino dituliskan dalam format file JSON yang berisikan dengan informasi pesan yang akan dikirim, lalu informasi topik supaya dapat menampilkan datanya dan informasi proses datanya. Dengan protokol MQTT pula keamanannya terjamin, karena ada yang Namanya topik dan *access token*, dimana *access token* merupakan IP dari mikrokontroler namun dengan identitas random, sehingga dapat menjaga data yang akan dikirimkan. Dalam program, nilai sensor sudah dimasukkan ke dalam satu array “{Value1=...,Value2,...,}” array itu lalu dinamakan pesan, dimana saat melakukan perintah *publish*, pesan yang di dalamnya informasi data sensor dapat mengirimkan data ke topik dengan nama v1/devices/me/telemetry pada Thingsboard. Arti dari topik tersebut adalah, dimana kita ingin menampilkan data dari proses telemetry (pengiriman) ke thingsboard yang ada pada satu pesan tertentu, dan data itu dapat diakses oleh siapa saja yang sudah tervalidasi dapat mengakses (*me*) dan device mana yang sedang ingin melakukan *request* untuk dapat melihat datanya. Hasil keluaran nilai dalam JSON file dapat dilihat pada Gambar 10.

B. Tampilan Web Server Things Board

ThingsBoard merupakan platform *IoT* sekaligus menjadi MQTT Broker yang memiliki menu *login*, *customer* (petugas rumah sakit dan dokter), daftar alat dan dapat menampilkan *dashboard*, dimana pada ThingsBoard akan menampilkan data yang terkirim dalam bentuk grafik dan angka maupun widget yang lainnya. Prosesnya dapat pada Gambar 11 sampai Gambar 14.

C. Hasil Data Hemodialisis

Pada pengujian ini memastikan bahwa data yang diterima oleh ThingsBoard sudah tersimpan dalam *database* lokal ThingsBoard. Untuk hasil data yang tersimpan dalam *database* lokal dapat dilihat pada Gambar 15. Data dapat dilihat dengan mengunduh di *PostgreSQL*

V. KESIMPULAN

Sistem yang dikembangkan telah berhasil untuk mengirimkan data dengan menggunakan protokol MQTT yang diterapkan pada monitoring alat hemodialisis yang dibuat sesuai dengan rancangan. Dalam rancangan terdapat elemen *publisher*, *broker*, dan *subscriber*. Pada penelitian ini yang digunakan sebagai *publisher* adalah mikrokontroler yang berfungsi untuk menerima data nilai sensor yaitu ESP32 (*master*) dan ESP32 (*slave*) yang dihubungkan secara serial, lalu *broker* yang digunakan adalah lokal MQTT broker pada ThingsBoard dan *subscriber* pada ThingsBoard saat *log-in* menjadi admin atau customers. Sehingga elemen-elemen tersebut dapat digunakan dalam implementasi konsep *Internet of Things* menggunakan protokol MQTT pada sistem monitoring alat Hemodialisis. Web Server ThingsBoard sudah berjalan dengan baik dengan dibuktikan bahwa ketika melakukan *publish* dari ESP32 (*slave*) untuk dikirim ke MQTT Broker ThingsBoard, pada ThingsBoard dapat menampilkan nilai tekanan darah (sistol dan diastol), aliran darah, temperature dialisat, dan konduktivitas dialisat pada topik yang dituliskan (Topik Telemetry pada ThingBoard). Dan sebagai admin maupun customers, dapat mengakes (melakukan subscribe) data nilai sensor yang ada di MQTT Broker.

DAFTAR PUSTAKA

- [1] F. Javed, A. V. Savkin, G. S. H. Chan, J. D. MacKie, and N. H. Lovell, “Identification and control for automated regulation of hemodynamic variables during hemodialysis,” *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1686–1697, 2011, doi: 10.1109/TBME.2011.2110650.
- [2] S. Dhingra, R. B. Madda, A. H. Gandomi, R. Patan, and M. Daneshmand, “Internet of things mobile-air pollution monitoring system (iot-mobair),” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5577–5584, 2019, doi: 10.1109/JIOT.2019.2903821.
- [3] M. Asghari, S. Yousefi, and D. Niyato, “Pricing strategies of iot wide area network service providers with complementary services included,” *J. Netw. Comput. Appl.*, vol. 147, no. May, p. 102426, 2019, doi: 10.1016/j.jnca.2019.102426.
- [4] M. Kashyap, V. Sharma, and N. Gupta, “Taking mqtt and nodemcu to iot: communication in internet of things,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1611–1618, 2018, doi: 10.1016/j.procs.2018.05.126.
- [5] R. K. Kodali and A. Valdas, “Mqtt based monitoring system for urban farmers using esp32 and raspberry pi,” in *Proceedings of the 2nd International Conference on Green Computing and Internet of Things, ICGCIoT 2018*, Aug. 2018, pp. 395–398, doi: 10.1109/ICGCIoT.2018.8752995.
- [6] R. N. Fine and A. R. Nissenson, *Handbook of Dialysis Therapy*, 4th ed. Philadelphia: Saunders/Elsevier, 2008.
- [7] M. S. Hossain, M. Rahman, M. T. Sarker, M. E. Haque, and A. Jahid, “A smart iot based system for monitoring and controlling the sub-station equipment,” *Internet of Things*, vol. 7, p. 100085, 2019, doi: 10.1016/j.iot.2019.100085.
- [8] N. Tantitharanukul, K. Osathanunkul, K. Hantrakul, P. Pramokchon, and P. Khoenkaw, “Mqtt-topics management system for sharing of open data,” in *2nd Joint International Conference on Digital Arts, Media and Technology 2017: Digital Economy for Sustainable Growth, ICDAMT 2017*, Apr. 2017, pp. 62–65, doi: 10.1109/ICDAMT.2017.7904935.