

**ANALISIS PENGIMPLEMENTASIAN PROTOKOL MQTT
BERBASIS CLOUD DALAM SISTEM MONITORING
KETERSEDIAAN AIR DENGAN WEB DAN ANDROID**

*Analysis of Cloud-based MQTT Protocol Implementation on Water Availability
Monitoring System with Web and Android*

PROPOSAL PROYEK AKHIR

Diajukan sebagai syarat untuk mengambil Mata Kuliah Proyek Akhir

oleh :

MUHAMMAD REVALDI RAHMAN

6705184091



**D3 TEKNOLOGI TELEKOMUNIKASI
FAKULTAS ILMU TERAPAN
UNIVERSITAS TELKOM
2021**

LEMBAR PENGESAHAN

Proposal Proyek Akhir dengan judul :

ANALISIS PENGIMPLEMENTASIAN PROTOKOL MQTT BERBASIS CLOUD
DALAM SISTEM MONITORING KETERSEDIAAN AIR
DENGAN WEB DAN ANDROID

*Analysis of Cloud-based MQTT Protocol Implementation
on Water Availability Monitoring System with Web and Android*

oleh :

MUHAMMAD REVALDI RAHMAN
6705184091

Telah diperiksa dan disetujui untuk diajukan sebagai syarat mengambil
Mata Kuliah Proyek Akhir
pada Program Studi D3 Teknologi Telekomunikasi Universitas Telkom

Bandung, 21 Januari 2021

Menyetujui,

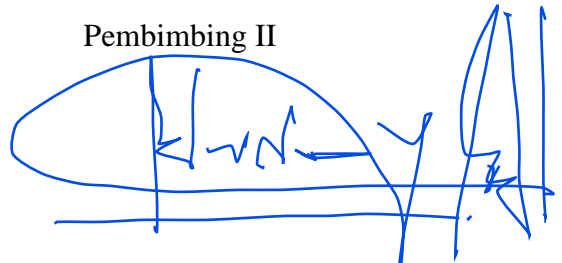
Pembimbing I



Dadan Nur Ramadan, S.Pd., M.T.

NIP. 14820047

Pembimbing II



Dr. Indrarini Dyah Irawati, S.T., M.T.

NIP. 07780053

ABSTRAK

Air sebagai salah satu kebutuhan pokok makhluk hidup memegang peranan penting dalam kehidupan sehari-hari, khususnya air bersih. Untuk menjaga ketersediaan air di lingkungan sekitar, sudah banyak yang menggunakan tandon air. Secara umum untuk memantau ketersediaan air pada tandon dilakukan secara manual, namun hal tersebut masih dinilai kurang efisien.

Sebagai solusi dari permasalahan tersebut, sudah banyak penelitian *Internet of Things* (IoT) untuk memantau ketersediaan air. Umumnya protokol komunikasi yang digunakan adalah protokol HTTP, yang menggunakan *bandwidth* cukup besar serta ukuran paket yang besar. Sehingga kurang sesuai jika digunakan pada sistem yang latensinya tinggi. Penelitian terdahulu membuktikan bahwa protokol MQTT memiliki performa yang lebih baik daripada HTTP.

Dengan demikian, penelitian ini dimaksudkan untuk membuat sistem pemantauan ketersediaan air yang mengimplementasikan protokol MQTT berbasis *cloud*, dimana *broker* MQTT berada pada *platform* Microsoft Azure. Serta untuk memudahkan dalam mengakses data, dibuat antar muka berupa *website* dan aplikasi *android*. Hasil yang diharapkan dari sistem ini adalah sistem berjalan dengan baik serta dapat dianalisis kinerja protokol MQTT yang diterapkan.

kata kunci : MQTT, *cloud*, *broker*, Microsoft Azure

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Rumusan Masalah	2
1.4 Batasan Masalah	2
1.5 Metodologi	3
BAB II DASAR TEORI	4
2.1 <i>Message Queuing Telemetry Transport (MQTT)</i>	4
2.2 Mikrokontroler	6
2.3 <i>Sensor Ultrasonic</i>	6
2.4 Microsoft Azure	6
2.5 Google Firebase	7
2.6 Website	7
2.7 Android	8
BAB III MODEL SISTEM	9
3.1 Blok Diagram Sistem	9
3.2 Tahapan Perancangan	10
3.3 Perancangan <i>Broker MQTT</i>	11
3.4 Perancangan Perangkat	12
3.5 Perancangan <i>Website</i>	16
3.6 Perancangan Aplikasi <i>Android</i>	17
BAB IV BENTUK KELUARAN YANG DIHARAPKAN	18
4.1 Keluaran yang Diharapkan	18
4.2 Jadwal Pelaksanaan	18
DAFTAR PUSTAKA	19

BAB I

PENDAHULUAN

1.1 Latar Belakang

Air sebagai salah satu kebutuhan pokok makhluk hidup memegang peranan penting dalam kehidupan sehari-hari. Seiring berkembangnya jaman, kebutuhan air juga meningkat, khususnya kebutuhan air bersih. Berdasarkan data yang diperoleh dari Badan Pusat Statistik, tercatat bahwa hingga tahun 2018 kebutuhan air bersih di Indonesia meningkat setiap tahunnya [1].

Untuk menjaga ketersediaan air bersih di lingkungan rumah sudah banyak yang menggunakan tandon air. Masalah yang muncul adalah tidak diketahuinya ketersediaan air yang ada pada tandon. Secara umum untuk pemantauan ketinggian air dapat dilakukan secara manual, namun kegiatan tersebut dinilai tidak cukup efisien karena memerlukan waktu dan tenaga tambahan.

Sebagai solusi dari permasalahan tersebut sudah banyak penelitian yang memanfaatkan konsep IoT (*Internet of Things*) yang dapat memantau ketinggian air secara otomatis dan *realtime*. Umumnya komunikasi yang digunakan pada penelitian terdahulu adalah dengan menggunakan protokol HTTP (*Hypertext Transfer Protocol*) yang menghubungkan langsung dari mikrokontroller ke *server*. Protokol HTTP memiliki beberapa kekurangan seperti pemakaian *bandwidth* yang besar, dan ukuran paket yang besar sehingga kurang dapat diandalkan saat berjalan pada sistem yang *bandwidth* nya rendah atau dengan latensi yang tinggi.

Untuk mengatasi permasalahan tersebut dapat digunakan protokol MQTT (*Message Queueing Telemetry Transport*) yang merupakan protokol komunikasi yang dirancang khusus untuk mendukung komunikasi “*Machine to Machine*”. MQTT merupakan protokol komunikasi sederhana dan ringan, yang memiliki fungsi *Publish* dan *Subscribe* untuk komunikasi 2 arah, baik antar *server* maupun dengan perangkat. MQTT juga didesain untuk sistem berkemampuan terbatas, *bandwidth* yang rendah, latensi yang tinggi, dan jaringan yang kurang dapat diandalkan. Berdasarkan penelitian yang telah dilakukan didapatkan hasil bahwa protokol MQTT dapat memberikan performa yang lebih baik dibandingkan dengan protokol HTTP, terlebih jika menggunakan banyak *device*. [2]

Oleh karena itu, proyek akhir ini bermaksud untuk merancang suatu sistem *monitoring* ketersediaan air yang mengimplementasikan protokol MQTT berbasis *cloud* dengan antar muka *website* dan aplikasi *android*. Sekaligus menganalisis kinerja protokol MQTT berbasis *cloud* tersebut dengan parameter pengujian berupa besar *delay* pengiriman dari *client* ke *broker*.

1.2 Tujuan

Adapun tujuan dari Proyek Akhir ini, sebagai berikut:

1. Dapat merancang dan membuat sistem *monitoring* ketersediaan air dengan antar muka *website* dan aplikasi *android*.
2. Dapat mengimplementasikan protokol MQTT berbasis *cloud* pada sistem yang dirancang.
3. Dapat menganalisis kinerja protokol MQTT yang diimplementasikan pada sistem.

1.3 Rumusan Masalah

Adapun rumusan masalah dari Proyek Akhir ini, sebagai berikut:

1. Bagaimana bentuk antar muka sistem *monitoring* ketersediaan air yang dirancang?
2. Bagaimana cara masing-masing *node* sensor mengirim data yang diperoleh?
3. Bagaimana kinerja dari protokol MQTT yang diimplementasikan pada sistem?

1.4 Batasan Masalah

Dalam Proyek Akhir ini, dilakukan pembatasan masalah sebagai berikut:

1. Sistem yang dirancang menggunakan protokol MQTT.
2. *Broker* MQTT berada di *cloud*.
3. Menggunakan NodeMCU dan Arduino Uno sebagai mikrokontroller.
4. Menggunakan sensor ultrasonic HY-SRF05 untuk mengukur volume air.
5. Menggunakan *platform* Microsoft Azure sebagai penyedia *broker* MQTT.
6. Menggunakan layanan *Google Firebase Realtime Database* sebagai *database* NoSQL.

7. Parameter pengujian adalah besar *delay* yang dibutuhkan dalam proses komunikasi MQTT.
8. Antar muka yang dibuat berupa *website* dan aplikasi *android* yang dapat menampilkan data secara *realtime*.

1.5 Metodologi

Metodologi pada penelitian ini, sebagai berikut:

1. Studi Literatur

Hal yang dilakukan adalah mencari informasi dan pendalaman materi-materi yang terkait melalui referensi yang tersedia di berbagai sumber.

2. Perancangan Sistem

Melakukan perancangan sistem monitoring ketersediaan air, mulai dari perancangan alat serta perancangan program secara keseluruhan.

3. Implementasi Sistem

Membuat sistem yang telah dirancang, memastikan mikrokontroler dapat mengirim data dengan baik, memastikan data dapat terkirim ke *database*, dan dapat menampilkan melalui *website* dan aplikasi *android*.

4. Troubleshooting

Jika terjadi ketidaksesuaian data atau terjadi *error*, maka dilakukan pencarian kesalahan dan cara mengatasinya.

5. Analisis

Melakukan analisis untuk menentukan apakah sistem dapat berfungsi dengan baik serta untuk mengetahui kinerja sistem secara keseluruhan.

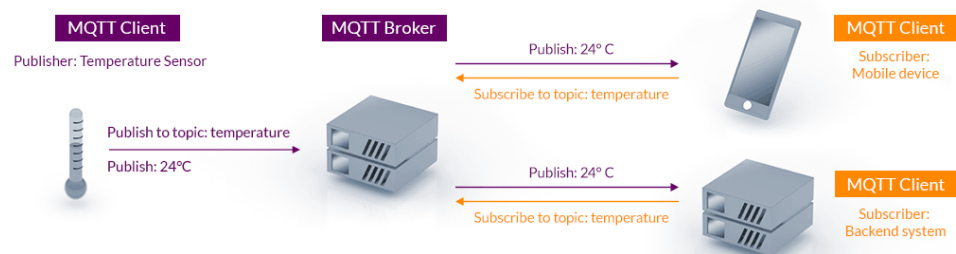
BAB II

DASAR TEORI

2.1 *Message Queuing Telemetry Transport* (MQTT)

Message Queuing Telemetry Transport (MQTT) merupakan salah satu protokol keluaran IBM (*International Business Machine*) yang berjalan di atas TCP/IP. Protokol MQTT pertama kali dikembangkan pada tahun 1999 yang didesain khusus untuk perangkat dengan sumber daya terbatas. MQTT bertujuan untuk memastikan pengiriman pesan yang andal dengan menggunakan *bandwidth* yang kecil serta waktu *delay* yang lebih singkat [3].

MQTT menerapkan konsep *Publish/Subscribe*, yang terdiri dari perangkat *publisher* untuk pengiriman pesan dan perangkat *subscriber* untuk penerimaan pesan. Kedua perangkat atau *client* tersebut dihubungkan oleh suatu *broker* yang juga berperan untuk menyaring pesan berdasarkan topik. Konsep *Publish/Subscribe* ini ditunjukkan oleh Gambar 2.1 di bawah ini.



Gambar 2.1 Konsep MQTT

Dengan menerapkan konsep ini, *client* tidak perlu berkenalan langsung dengan *client* lain atau dikenal dengan sebutan *space decoupling*. Serta masing-masing *client* tidak harus berjalan dalam waktu yang bersamaan, dikenal dengan sebutan *time decoupling*. Selain itu dikenal juga sebutan *synchronization decoupling*, yang memungkinkan sinkronisasi terjadi tanpa mengganggu proses *client* dalam pengiriman ataupun penerimaan pesan [4].

Pada MQTT sendiri terdapat 3 jenis tingkatan *Quality of Service* (QoS) yang membedakan mekanisme pengiriman pesan [5]. Jenis tingkatan *Quality of Service* (QoS) tersebut adalah sebagai berikut.

1. QoS Level 0

Tingkatan QoS 0 menerapkan prinsip pengiriman “*best-effort*”, artinya tidak ada jaminan pesan saat proses pengiriman. Tingkat QoS 0 ini juga dikenal dengan sebutan “*fire and forget*” karena isi pesan tidak disimpan, dan tidak dikirim kembali oleh penerima.

2. QoS Level 1

Tingkatan QoS 1 menjamin pesan terkirim setidaknya satu kali kepada penerima. Pengirim akan menyimpan pesan yang dikirim, hingga mendapatkan paket PUBACK dari penerima. Jika pengirim tidak menerima paket PUBACK setelah waktu yang cukup lama, pengirim akan mengirim kembali pesan tersebut. Saat penerima mendapatkan pesan dengan tingkat QoS 1, pesan tersebut dapat langsung diproses.

3. QoS Level 2

Tingkatan QoS 2 merupakan tingkatan layanan tertinggi di MQTT. Tingkatan ini menjamin pesan hanya diterima satu kali oleh penerima yang dituju. QoS 2 merupakan layanan dengan pengiriman paling lambat, namun paling aman. Jaminan pesan tercipta karena terjadi setidaknya 2 kali *request/response* antara pengirim dengan penerima. Pertama, pengirim akan mengirim pesan dengan paket PUBLISH, kemudian penerima akan membalas dengan paket PUBREC setelah paket PUBLISH diterima. Jika pengirim tidak mendapatkan paket PUBREC dari penerima, paket PUBLISH akan dikirim kembali dengan tambahan *flag* DUP. Setelah penerima mendapat paket PUBREC, paket PUBLISH yang disimpan dapat dibuang dan mengirim respon paket PUBREL. Jika penerima mendapat paket PUBREL, maka semua pesan paket yang tersimpan akan dibuang dan membalas dengan paket PUBCOMP. Setelah paket PUBCOMP diterima pengirim, *packet identifier* dapat digunakan kembali untuk pengiriman berikutnya.

Table 2.1 Perbedaan QoS Level

QoS Level	Message Payload
QoS 0	PUBLISH
QoS 1	PUBLISH, PUBACK

QoS 2	PUBLISH, PUBREC, PUBREL, PUBCOMP
-------	----------------------------------

2.2 Mikrokontroler

Mikrokontroler merupakan sebuah komputer berukuran kecil yang biasa digunakan untuk mendeteksi masukan analog atau digital maupun untuk mengendalikan perangkat lain. Saat ini, banyak perangkat elektronik yang memiliki mikrokontroler didalamnya karena mudah untuk digunakan serta dapat terhubung dengan komputer maupun jaringan. Umumnya, mikrokontroler berfungsi sebagai pengendali yang mengatur proses kerja suatu rangkaian elektronik. Dalam sebuah IC mikrokontroler terdapat CPU, memori, *timer*, saluran komunikasi serial dan parallel, port untuk *input/output*, ADC, PWM, dan lain-lain [6].

2.3 Sensor *Ultrasonic*

Sensor *ultrasonic* merupakan sensor yang membaca jarak antara sensor dengan benda yang ada di hadapannya. Sensor bekerja dengan mengirimkan gelombang suara di atas frekuensi pendengaran manusia, kemudian gelombang yang dipantulkan oleh benda akan ditangkap. Selanjutnya sensor akan menghitung waktu tempuh gelombang untuk menentukan jarak dengan benda.

Umumnya sensor *ultrasonic* memiliki 4 pin yaitu VCC, *trigger*, *echo*, dan GND. Pin *trigger* berfungsi memancarkan gelombang *ultrasonic* dengan frekuensi 40kHz, sementara pin *echo* berfungsi menangkap gelombang yang dipantulkan oleh benda tersebut [7].

2.4 Microsoft Azure

Microsoft Azure merupakan sebuah implementasi *Platform as a Service* (PaaS) dari *cloud computing* yang dikembangkan oleh Microsoft. Dengan menggunakan Microsoft Azure, pengembang dapat membuat aplikasi web, *cloud service*, jembatan koneksi antar *platform*, dan sebagainya. Salah satu yang menjadi keunggulan Microsoft Azure adalah kemampuannya untuk menjalankan aplikasi dari berbagai teknologi dan *platform* seperti .NET, PHP, maupun JAVA [8].

Beberapa keuntungan menggunakan teknologi Microsoft Azure diantaranya:

- Pengembang hanya membuat perangkat lunak atau sistem saja, karena Microsoft akan melindungi sistem tersebut.
- Mudah dalam menambah jumlah kapasitas penyimpanan data.
- Terintegrasi dengan aplikasi Microsoft Visual Studio [9].

Dari sekian layanan yang disediakan oleh Microsoft Azure, pada sistem ini menggunakan salah satu layanan Azure, yaitu *Compute*. Dimana *compute* sendiri merupakan bagian dari Azure yang berguna dalam proses komputasi, baik secara *foreground* maupun *background job*. Layanan *compute* yang digunakan adalah Azure *Virtual Machine* yang menyediakan server virtual yang dapat digunakan dalam banyak cara, layaknya server fisik.

2.5 Google Firebase

Firebase merupakan sebuah *Backend as a Service* yang disediakan oleh Google. Firebase menyediakan berbagai layanan yang mempermudah pengembang dalam mengembangkan aplikasinya dengan dilengkapi berbagai fitur seperti autentikasi pengguna, *storage*, *cloud messaging*, *real-time database*, dan sebagainya.

Dari beberapa layanan tersebut pada sistem ini menggunakan layanan *real-time database* yang merupakan *database* NoSQL. Pada *real-time database*, data disimpan dalam bentuk JSON dan akan disinkronkan secara *realtime* ke setiap klien yang terhubung. Klien yang dapat terhubung ke layanan ini pun beragam, mulai dari *Web*, *Android*, *iOS*, hingga mikrokontroler [10].

2.6 Website

Website merupakan kumpulan halaman di internet yang berisi informasi dengan tujuan tertentu dan saling terhubung serta dapat diakses secara luas melalui halaman depan yang ditunjukkan dengan URL *website* [11]. Untuk membangun suatu *website* terdapat beberapa bahasa yang umum digunakan, seperti:

- [1] HTML (*Hypertext Markup Language*)

HTML merupakan bahasa yang disisipkan (*embedded language*) pada suatu dokumen dengan memberi tanda tertentu yang disebut dengan *tag*. Dari *tag* tersebut, *browser* dapat menentukan tampilan teks atau dokumen yang akan ditampilkan [12].

[2] CSS (*Cascading Style Sheet*)

CSS merupakan bahasa *stylesheet* yang digunakan untuk mengatur *style* pada suatu dokumen. Umumnya CSS dikombinasikan dengan HTML untuk memperindah tampilan suatu *website* [12].

[3] JavaScript

JavaScript merupakan skrip program berbasis *client* yang dieksekusi oleh *browser* sehingga dapat membuat halaman *website* menjadi lebih interaktif [12].

2.7 Android

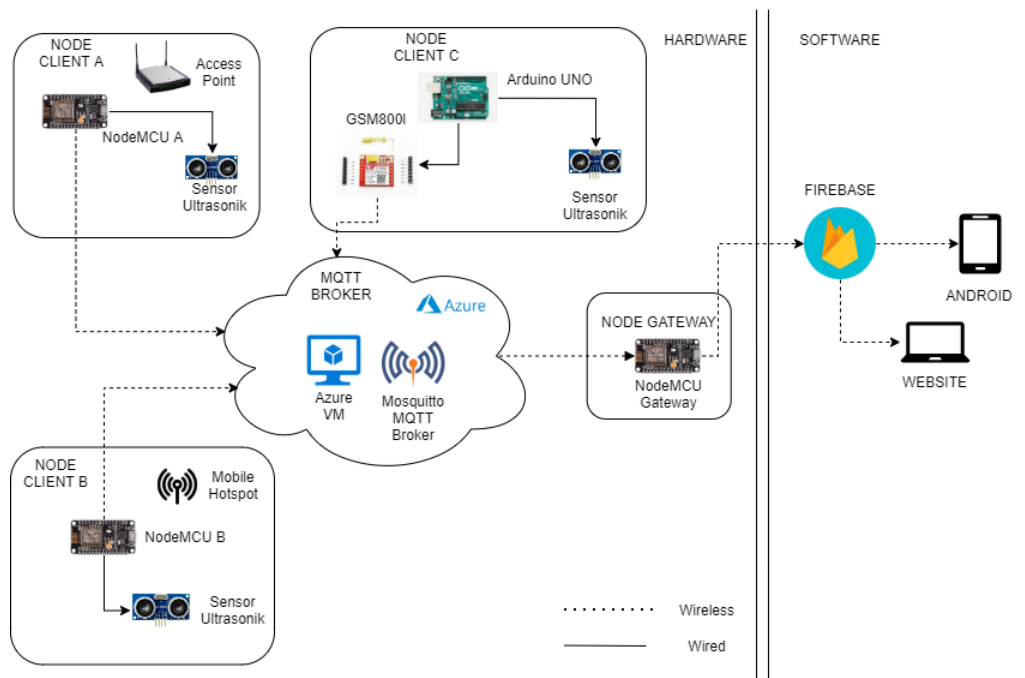
Android merupakan sistem operasi berbasis ponsel bergerak atau *mobile phone* yang menggunakan *kernel* Linux. Pertama kali dirilis oleh Google pada November 2007 dengan tujuan sebagai sarana pengembang untuk mengembangkan *software* pada ponsel [13]. Android bersifat *open-source* yang banyak digunakan karena dapat dengan mudah ditemukan dan juga fleksibel untuk dioperasikan. Selain terdapat pada ponsel, Android juga terdapat pada beberapa perangkat elektronik lain seperti jam tangan pintar (*smartwatch*), tablet, televisi, hingga kendaraan [14].

BAB III

MODEL SISTEM

3.1 Blok Diagram Sistem

Pada bab ini akan dijelaskan mengenai perancangan sistem *monitoring* ketinggian air dengan *web* dan *android* yang mengimplementasikan protokol MQTT berbasis *cloud* yang terdiri dari sub bab model sistem, diagram alir perancangan sistem secara keseluruhan, perancangan perangkat, perancangan *website*, serta perancangan aplikasi *android*. Adapun model sistem secara keseluruhan dapat dilihat pada Gambar 3.1 di bawah ini.



Gambar 3.1 Model Sistem Keseluruhan

Sistem yang dirancang akan menggunakan Mosquitto sebagai broker MQTT, yang ter-*install* di dalam *Virtual Machine* pada Microsoft Azure. Dirancang 3 buah node yang berperan sebagai *publisher* dan 1 buah node sebagai *subscriber*. Tiap node *publisher* akan mengambil data ketersediaan air yang didapatkan dari sensor *ultrasonic*, kemudian akan *publish* ke topik MQTT masing-masing. Untuk proses koneksi dengan broker, node *publisher* A menggunakan jaringan WiFi yang

terhubung dengan *Access Point*, node *publisher* B menggunakan jaringan WiFi yang terhubung dengan *Mobile Hotspot*, sedangkan node *publisher* C akan menggunakan modul GSM SIM800l untuk mengirimkan data melalui internet, sehingga masing-masing node tidak berada pada jaringan yang sama. Selanjutnya node *subscriber* akan *subscribe* ke tiap topik dari masing-masing *publisher*, lalu data yang didapat akan disimpan ke *Firebase Realtime Database*. Setelah tersimpan di *database*, selanjutnya data akan ditampilkan melalui antar muka *website* dan juga aplikasi *android*.

3.2 Tahapan Perancangan

Proses perancangan sistem ini dilakukan dengan metode eksperimental dan prosesnya dapat dilihat pada Gambar 3.2, tahapan pembuatannya adalah sebagai berikut:

1. Perancangan *broker* MQTT

Langkah awal dalam pembuatan sistem ini adalah perancangan *broker* MQTT yang terdiri dari pemilihan jenis *broker* MQTT, pemilihan *platform cloud*, serta pengujian.

2. Perancangan perangkat keras sebagai *client* MQTT

Tahap ini dilakukan untuk merancang perangkat keras yang akan berperan sebagai node *client* MQTT, mulai dari penentuan mikrokontroller dan komponen, hingga pengujian perangkat.

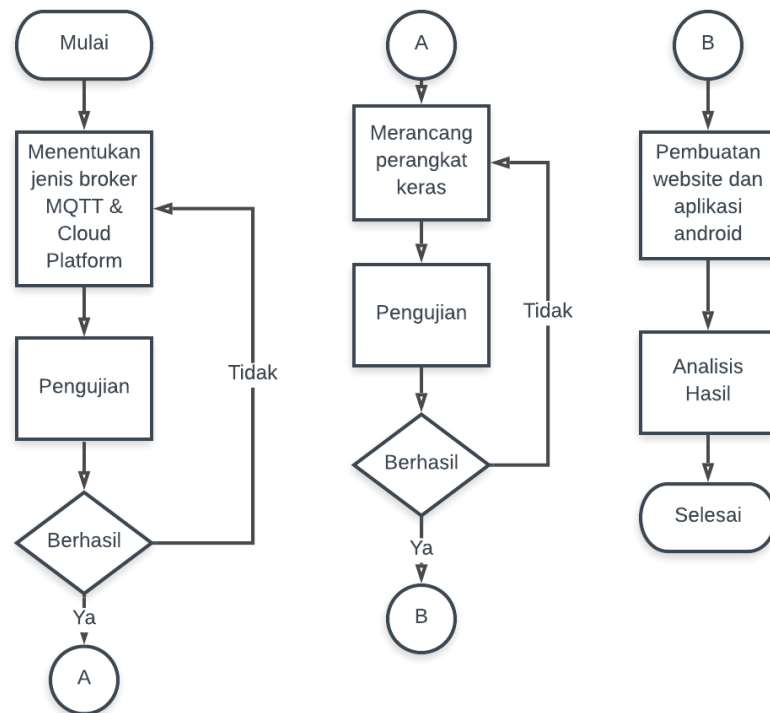
3. Perancangan *website*

Tahap ini dilakukan untuk membuat antar muka bagi pengguna berupa *website* yang akan menampilkan data dari masing-masing node.

4. Perancangan aplikasi *android*

Tahap ini dilakukan untuk membuat antar muka bagi pengguna berupa aplikasi *android* yang akan menampilkan data dari masing-masing node.

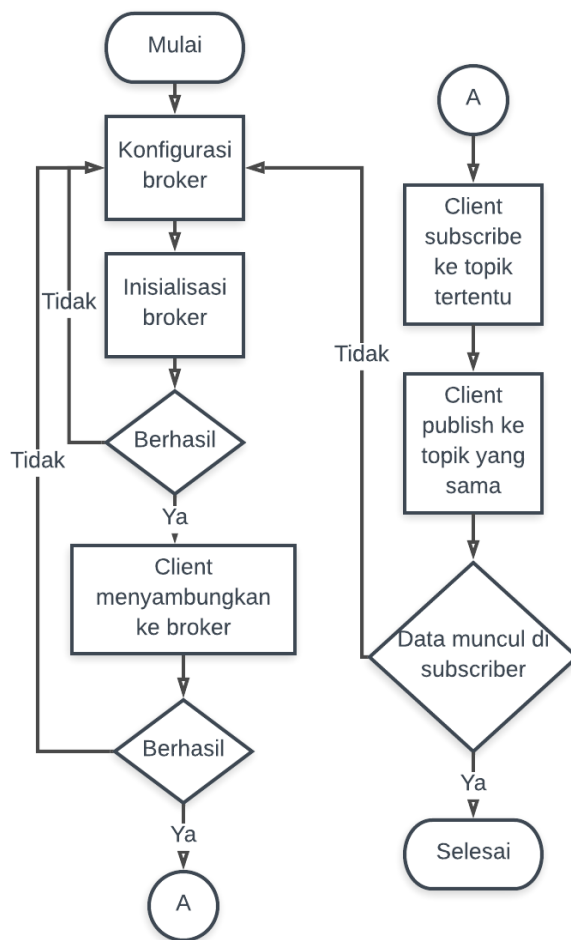
Berikut ini merupakan diagram alir yang menunjukkan tahapan dalam perancangan Proyek Akhir.



Gambar 3.2 Diagram Alir Perancangan

3.3 Perancangan *Broker* MQTT

Pada Proyek Akhir ini akan dirancang sistem *monitoring* ketersediaan air yang mengimplementasikan protokol MQTT berbasis *cloud* dengan antar muka *website* dan *android*. Perancangan sistem dimulai dari *broker* MQTT yang berperan sebagai inti dari sistem ini. Setelah *broker* berjalan dengan baik, selanjutnya akan dirancang perangkat keras yang berperan sebagai *client* MQTT, lalu dibuat *website* dan aplikasi *android* untuk menampilkan data dari masing-masing *client*. Setelah berjalan dengan baik, selanjutnya komunikasi antara *broker* dengan *client* akan dianalisis untuk mengetahui kinerja dari protokol MQTT berbasis *cloud* ini. Berikut diagram alir *broker* MQTT yang dirancang.



Gambar 3.3 Diagram Alir Broker MQTT

3.4 Perancangan Perangkat

Pada proyek akhir ini perangkat akan berperan sebagai node *client*, dimana masing-masing node dibedakan untuk konfigurasinya. Secara keseluruhan, alat yang digunakan adalah sebagai berikut:

1. Arduino UNO

Arduino UNO merupakan papan mikrokontroller open-source berbasis ATmega328P. Dengan dilengkapi 14 pin input/output digital, dan 6 pin input analog, Arduino UNO banyak digunakan untuk membaca hasil sensor serta memberikan output dari hasil bacaan. Arduino UNO berisi semua yang diperlukan untuk mendukung mikrokontrollernya, cukup dengan menghubungkannya ke komputer melalui kabel USB atau dengan AC-DC adapter [15].

Untuk memprogram Arduino UNO, dapat menggunakan software Arduino IDE dimana bahasa pemrograman yang digunakan adalah bahasa C yang telah disederhanakan. Pada sistem ini, Arduino UNO digunakan sebagai mikrokontroller yang akan mengambil data dari sensor ultrasonic, yang kemudian akan mengirimkan data yang diperoleh ke *broker* melalui modul GSM8001. Bentuk fisik Arduino UNO ditunjukkan oleh Gambar 3.4 di bawah ini.



Gambar 3.4 Arduino UNO

2. NodeMCU

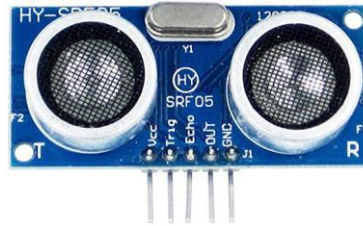
NodeMCU merupakan papan elektronik berbasis chip ESP8266 yang dapat menjalankan fungsi mikrokontroller dan koneksi ke internet melalui WiFi. Dilengkapi dengan 17 pin GPIO, yang juga mengintegrasikan PWM, I2C, dan ADC [16]. NodeMCU dapat dikembangkan menjadi sebuah aplikasi monitoring maupun controlling pada Internet of Things (IoT). Dengan menggunakan bahasa pemrograman Lua, NodeMCU juga dapat diprogram menggunakan Arduino IDE [17]. Pada sistem ini, NodeMCU juga digunakan sebagai mikrokontroller untuk mengambil data dari sensor, yang kemudian mengirimkannya ke *broker*. Bentuk fisik NodeMCU ESP8266 dapat dilihat pada Gambar 3.5 di bawah ini.



Gambar 3.5 NodeMCU

3. Sensor *Ultrasonic* HY-SRF05

Sensor HY-SRF05 adalah sensor *ultrasonic* yang digunakan pada sistem ini, pada sensor ini terdapat sebuah pin tambahan yaitu OUT, yang berfungsi menyatukan pin trigger dan echo. Sensor ini dapat membaca jarak dari 2-450 cm dengan akurasi hingga 3 mm [18]. Sensor HY-SRF05 ditunjukkan oleh Gambar 3.6 di bawah ini.



Gambar 3.6 Sensor HY-SRF05

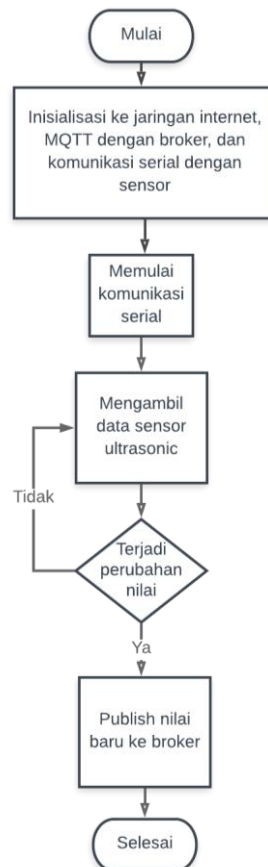
4. Modul GSM SIM800L

SIM800L merupakan modul GSM yang dapat mengakses layanan GPRS untuk pengiriman data ke internet dengan sistem M2M (Machine to Machine). Menggunakan *AT-Command* yang mirip dengan modul GSM lainnya [9]. Dengan antar muka standar industri, SIM800L memiliki *quad-band* dengan frekuensi 850/900/1800/1900 MHz untuk suara, SMS, data dan fax. Berdimensi kecil yaitu dengan ukuran 15.8 x 17.8 x 2.4 mm dan berat sekitar 1.35 g. Serta konsumsi daya yang rendah, mencapai 0.7 mA saat *sleep mode* [19]. Modul GSM SIM800l ditunjukkan oleh Gambar 3.7 di bawah ini.

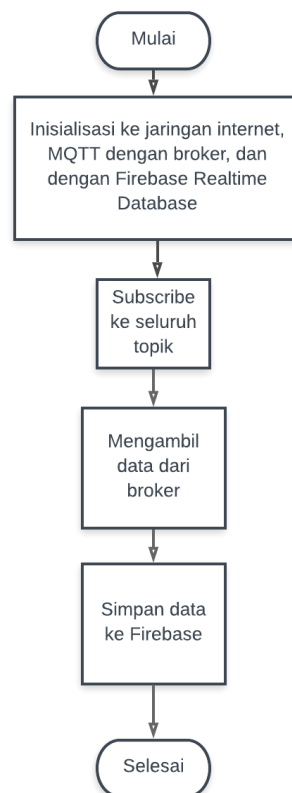


Gambar 3.7 Modul GSM SIM800L

Berikut ini merupakan diagram alir perangkat yang akan dibuat. Perangkat yang berperan sebagai *client publisher* ditunjukkan oleh Gambar 3.8, sedangkan perangkat yang berperan sebagai *client subscriber* ditunjukkan oleh Gambar 3.9.



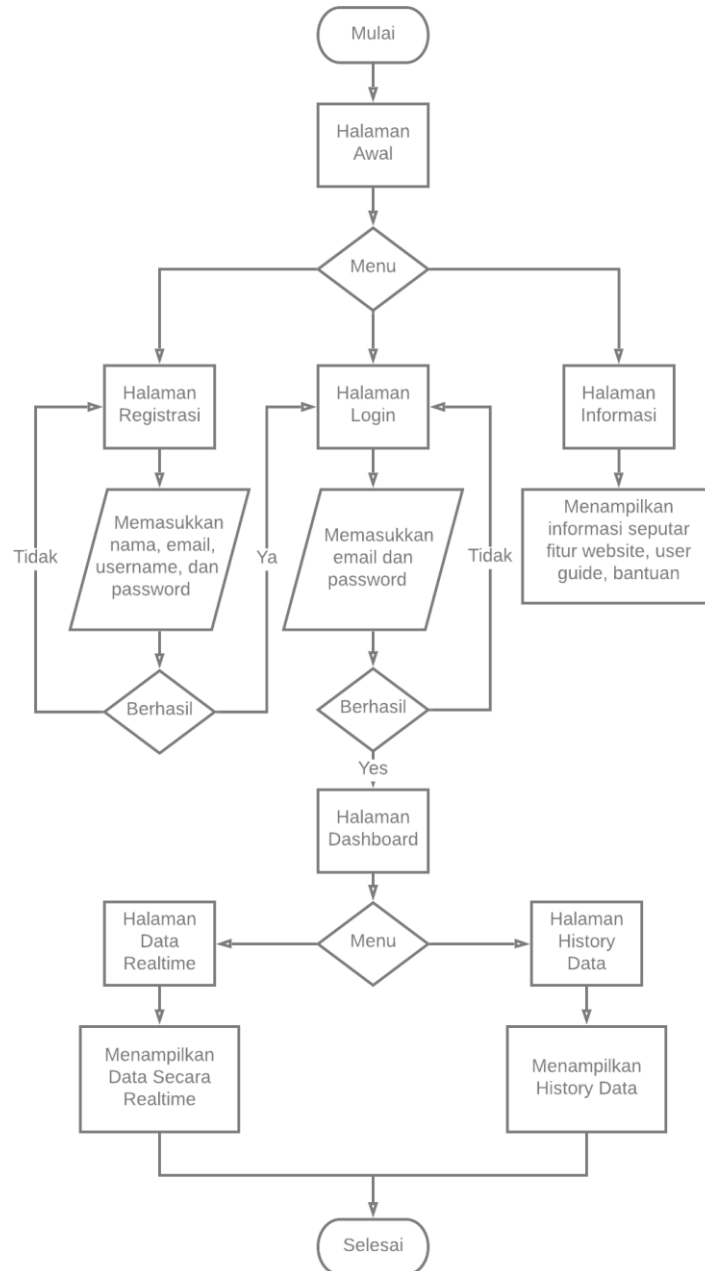
Gambar 3.8 Diagram Alir *Client Publisher*



Gambar 3.9 Diagram Alir *Client Subscriber*

3.5 Perancangan Website

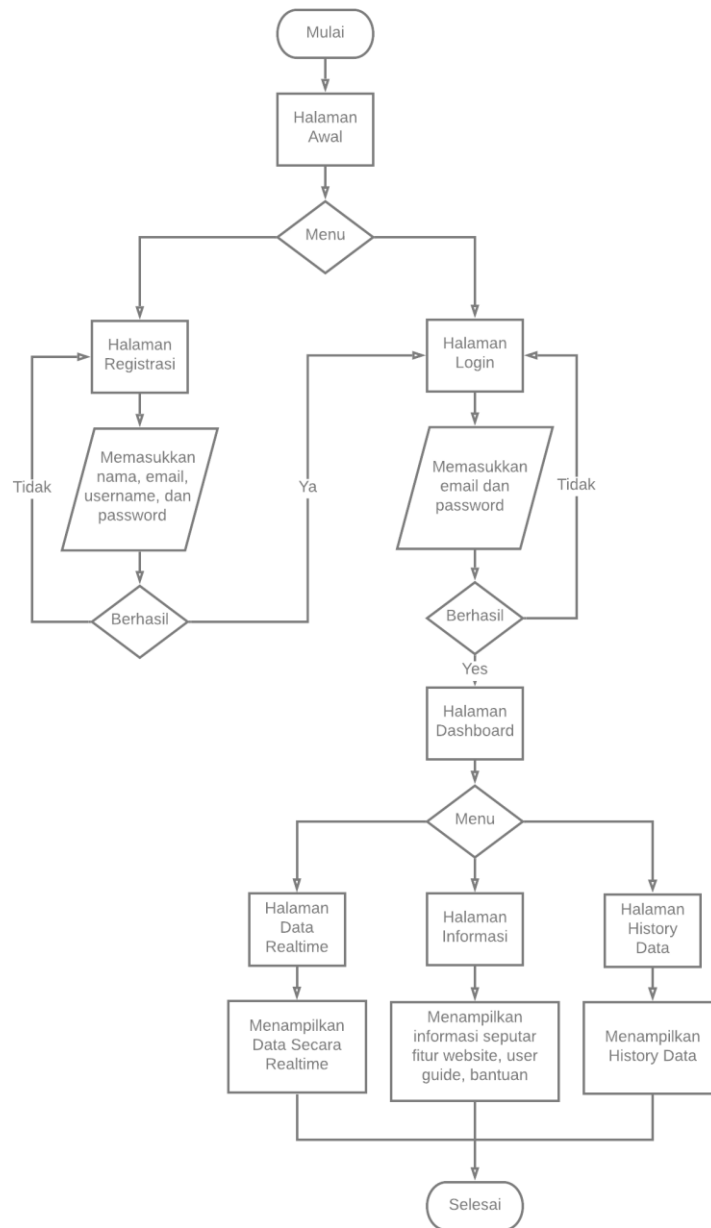
Berikut ini merupakan diagram alir *website* yang akan dibuat.



Gambar 3.10 Diagram Alir Website

3.6 Perancangan Aplikasi *Android*

Berikut ini merupakan diagram alir aplikasi *android* yang akan dibuat.



Gambar 3.11 Diagram Alir Aplikasi *Android*

BAB IV

BENTUK KELUARAN YANG DIHARAPKAN

4.1 Keluaran yang Diharapkan

Perancangan pada Proyek Akhir akan dibuat sistem dengan spesifikasi dan fungsi sebagai berikut :

1. *Broker* MQTT dapat menghubungkan antar *client*.
2. Perangkat dapat mendeteksi ketersediaan air melalui sensor *ultrasonic*.
3. Perangkat *subscriber* dapat menyimpan data ke *Firestore Realtime Database*.
4. Antar muka berupa *website* dan aplikasi *android* dapat terhubung dengan *Firestore* untuk menampilkan data yang tersimpan.
5. Dapat menganalisis kinerja protokol MQTT berbasis *cloud*, dengan parameter berupa besar *delay* pengiriman dari *client* ke *broker*.

4.2 Jadwal Pelaksanaan

Adapun jadwal pengerjaan Proyek tingkat bisa dilihat pada Tabel 4.1, sebagai berikut :

Tabel 4.1 Jadwal Pelaksanaan

Judul Kegiatan	Waktu					
	Jan	Feb	Mar	Apr	Mei	Jun
Studi Literatur						
Perancangan dan Simulasi Sistem						
Pengujian						
Analisis						
Pembuatan Laporan						

DAFTAR PUSTAKA

- [1] B. P. Statistik, "Badan Pusat Statistik," 2018. [Online]. Available: <http://www.bps.go.id>. [Accessed 10 12 2020].
- [2] N. B. R. M. Windryani N.P., "Analisa Perbandingan Protokol MQTT dengan HTTP Pada IOT Platform Patriot," *e-Proceeding of Engineering*, vol. 6, no. 2, p. 3193, 2019.
- [3] S. Manandhar, "MQTT Based Communication in IoT," 2017.
- [4] "Mqtt essentials part 2 - Publish & Subscribe," HiveMQ, 2015. [Online]. Available: <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>. [Accessed 18 January 2021].
- [5] F. Ilham, A. G. Putrada and S. Prabowo, "Analisis Performansi QoS MQTT pada Sistem Monitoring Sungai," *e-Proceeding of Engineering*, vol. 6, no. 1, p. 2013, 2019.
- [6] M. A. D. Alghifary, M. A. Murti and C. Setianingsih, "Perancangan Perangkat Manajemen dan Kendali Beban Listrik Berbasis Internet of Things," Telkom University, Bandung, 2020.
- [7] R. Fadhil, T. N. Damayanti and D. N. Ramadan, "PROTOTYPE SENSOR PARKIR MENGGUNAKAN REALTIME DATABASE DENGAN KOMUNIKASI MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT)," Telkom University, Bandung, 2019.
- [8] A. Gumelar and A. M. Bachtiar, "PEMBANGUNAN BACKEND UNTUK APLIKASI PENGAWASAN PENGGUNAAN INTERNET ANAK "DODO KIDS BROWSER" DENGAN TEKNOLOGI MICROSOFT AZURE," Universitas Komputer Indonesia, Bandung, 2016.
- [9] A. R. Hakim, "ANALISIS PERBANDINGAN SISTEM CLOUD AZURE DAN GOOGLE CLOUD," *Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 1, no. 1, 2016.
- [10] "Firebase Realtime Database," Google Firebase, [Online]. Available: <https://firebase.google.com/docs/database>. [Accessed 18 January 2021].
- [11] Waryanto, "Pengertian Website Lengkap dengan Jenis dan Manfaatnya," Niagahoster, 22 January 2018. [Online]. Available: <https://www.niagahoster.co.id/blog/pengertian-website/>. [Accessed 16 January 2021].

- [12] R. Imam and A. R. Nugraha, "PERANCANGAN SISTEM INFORMASI E-MARKETPLACE ORIGINAL CLOTHING INDONESIA BERBASIS WEB," *Jurnal Manajemen dan Teknik Informatika (JUMANTAKA)*, vol. 1, no. 1, pp. 161-170, 2018.
- [13] "Developer Guides | Android Developers," Google Developers, [Online]. Available: <https://developer.android.com/guide>. [Accessed 19 January 2021].
- [14] "What Is Android | Android," Android, [Online]. Available: <https://www.android.com/what-is-android/>. [Accessed 19 January 2021].
- [15] "Arduino Uno," Arduino, [Online]. Available: <http://arduino.cc/>. [Accessed 18 January 2021].
- [16] "NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet," components101, 22 April 2020. [Online]. Available: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>. [Accessed 18 January 2021].
- [17] I. A. Munawar, D. N. Ramadan and A. E. Prianda, "Implementasi Soil Moisture Sensor dengan MQTT Sebagai Pengukur Kelembapan Tanah Untuk Penyiraman Taman di Komplek Sanggar Indah Banjaran," Telkom University, Bandung, 2020.
- [18] "HY-SRF05 Datasheet," Datasheetpdf.com, [Online]. Available: <https://datasheetpdf.com/datasheet/HY-SRF05.html>. [Accessed 19 January 2021].
- [19] "SIM800L Datasheet," Datasheetpdf.com, [Online]. Available: <https://datasheetpdf.com/pdf/989664/SIMCom/SIM800L/1>. [Accessed 19 January 2021].
- [20] R. Affrilianto, D. Triyanto and Suhardi, "RANCANG BANGUN SISTEM PELACAK KENDARAAN BERMOTOR MENGGUNAKAN GPS," *Jurnal Coding Sistem Komputer Untan*, vol. 5, no. 3, pp. 1-11, 2017.
- [21] M. Farantino, V. Suryani and A. A. Wardana, "Deteksi Serangan Dos Pada IoT Berbasis Protokol MQTT dengan Metode Support Vector Machine," Telkom University, Bandung, 2020.



UNIVERSITAS TELKOM
FAKULTAS ILMU TERAPAN
KARTU KONSULTASI
SEMINAR PROPOSAL PROYEK TINGKAT


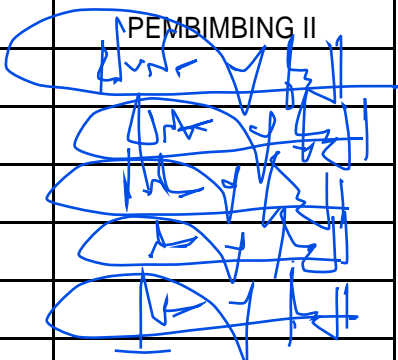
NAMA / PRODI : Muhammad Revaldi Rahman / D3 Teknologi Telekomunikasi NIM : 6705184091

JUDUL PROYEK AKHIR :

Analisis Pengimplementasian Protokol MQTT Berbasis *Cloud* dalam Sistem *Monitoring* Ketersediaan Air dengan *Web* dan *Android*

CALON PEMBIMBING : I. Dadan Nur Ramadan, S.Pd., M.T.

II. Dr. Indrarini Dyah Irawati, S.T., M.T.

NO	TANGGAL	CATATAN HASIL KONSULTASI	TANDA TANGAN CALON PEMBIMBING I
1		BAB 1 (SELESAI)	
2		BAB 2 (SELESAI)	
3		BAB 3 (SELESAI)	
4		BAB 4 (SELESAI)	
5		FINALISASI PROPOSAL	
6			
7			
8			
9			
10			
NO	TANGGAL	CATATAN HASIL KONSULTASI	TANDA TANGAN CALON PEMBIMBING II
1		BAB 1 (SELESAI)	
2		BAB 2 (SELESAI)	
3		BAB 3 (SELESAI)	
4		BAB 4 (SELESAI)	
5		FINALISASI PROPOSAL	
6			
7			
8			
9			
10			