



Protokol MQTT

Totok Budioko

Protokol MQTT

MQTT adalah protokol pengiriman pesan yang bersifat client-server dan publish/subscribe. Protokol MQTT ringan, sederhana, terbuka dan dirancang agar mudah diimplementasikan. Karakteristik seperti ini membuatnya ideal untuk digunakan pada berbagai situasi termasuk pada lingkungan yang terbatas seperti pada konteks komunikasi mesin ke mesin (M2M) dan Internet of Things (IoT) yang membutuhkan kode yang kecil dan bandwidth yang terbatas.

MQTT dibuat oleh Andy Stanford-Clark dari IBM, dan Arlen Nipper (dari ArcomSystems, kemudian menjadi CTO di Eurotech)



Protokol MQTT

Kemampuan Unik MQTT Untuk IoT

1. Lightweight and Efficient
2. Reliable Message Delivery
3. Connect IoT Devices at Massive Scale
4. Secure Bi-Directional Communication
5. Keep Alive and Stateful Sessions

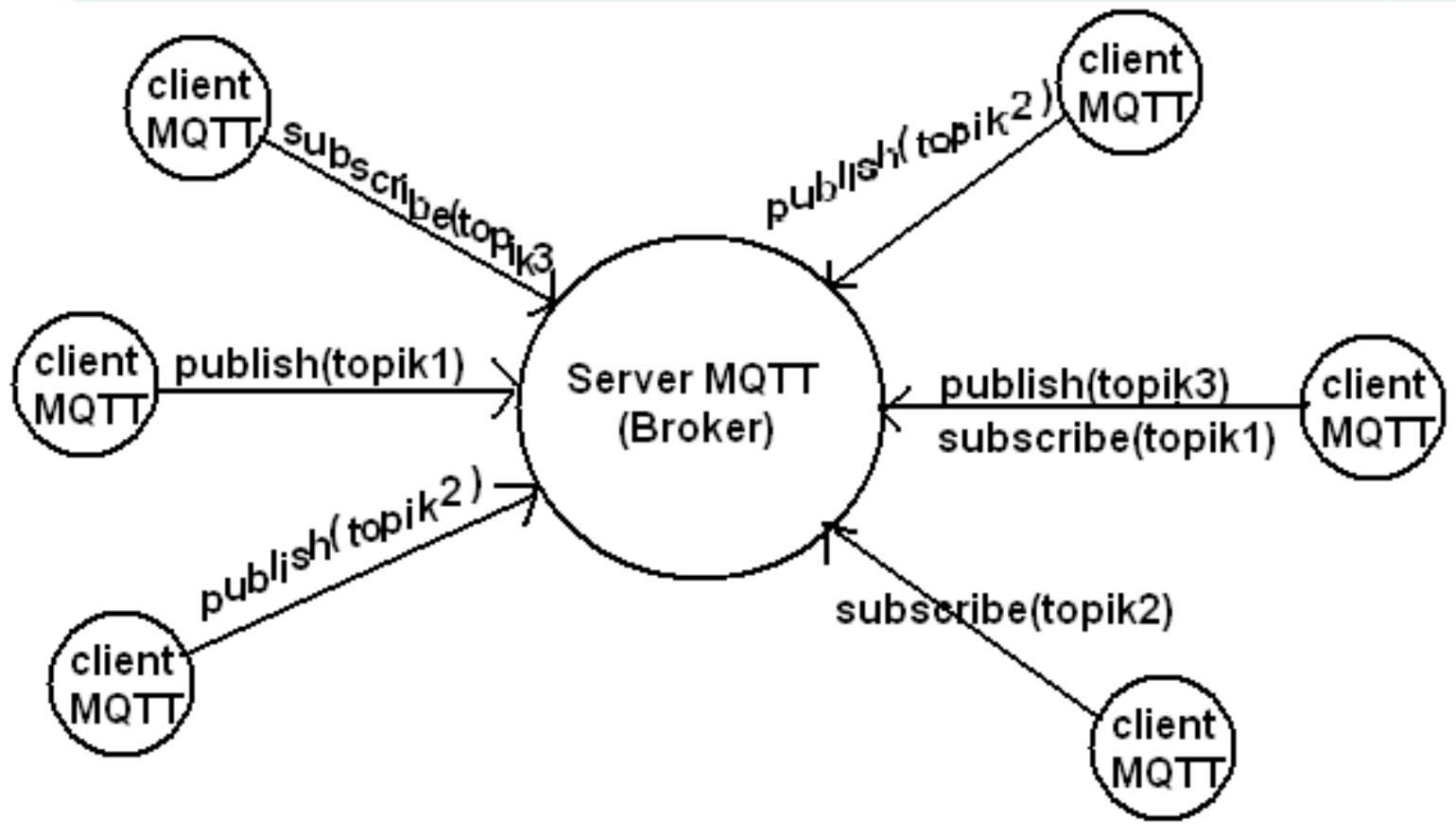


Protokol MQTT

MQTT menggunakan model client server. Server pada MQTT berlaku sebagai broker. Broker akan menerima client yang mengirimkan data. Pengiriman data ke broker disebut dengan “publish” yang dibatasi dengan topik. Jadi suatu client yang akan mengirimkan data ke server/broker MQTT melakukan “publish” pada topik tertentu. Client yang akan menggunakan data harus melakukan “subscribe” ke server/broker pada topik tertentu. Client dapat melakukan “publish” pada beberapa topik dan “subscribe” pada beberapa topik. Arsitektur MQTT diperlihatkan pada Gambar 1



Protokol MQTT



Gambar 1. Arsitektur MQTT



Protokol MQTT

MQTT mempunyai tiga *decoupling*, yaitu :

1. Space decoupling: Publisher dan subscriber tidak memerlukan untuk saling mengetahui (mis. Alamat IP dan Port)
2. Time decoupling: Publisher dan subscriber tidak perlu berjalan pada waktu yang sama.
3. Synchronization decoupling: Operasi dari kedua komponen tidak perlu melakukan interupsi selama melakukan pengiriman dan penerimaan.



Protokol MQTT

Komunikasi Pada Protokol MQTT

Komunikasi pada protokol MQTT secara langsung terjadi antara client dan server/broker. Antar client tidak boleh berkomunikasi secara langsung. Tahapan komunikasi adalah sebagai berikut.

A. Client Publish

1. Koneksi client ke server/broker
2. Publish

B. Client Subscribe/Unsubscribe

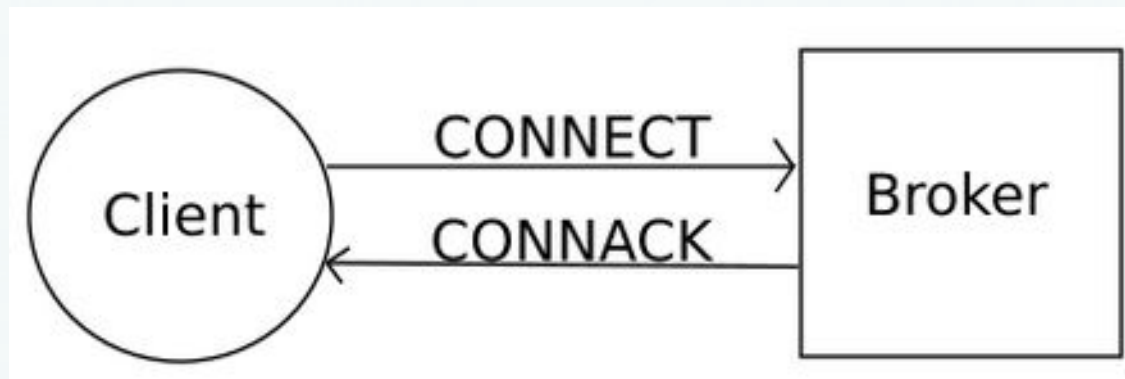
1. Koneksi Client ke Server/broker
2. Subscribe/Unsubscribe



Protokol MQTT

Client Connect

Untuk client yang melakukan publish pertama yang harus dilakukan adalah membuat koneksi ke broker. Untuk koneksi ke broker, client harus mengirim paket “CONNECT” dan broker akan memberikan respon dengan paket “CONNACK” sebagaimana diperlihatkan pada Gambar 2.2.



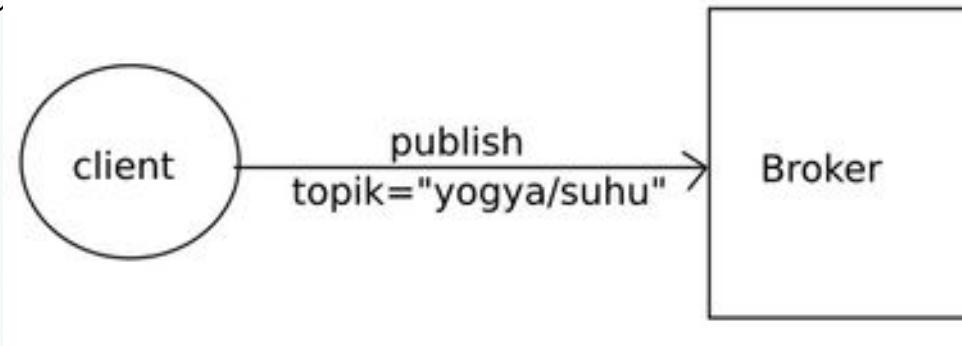
Gambar 2.2. Koneksi client ke broker



Protokol MQTT

Client Publish

Suatu *client* yang berhasil terkoneksi dengan *broker* dapat melakukan *publish* , yaitu mengirimkan paket ke *broker* pada suatu topik seperti diperlihatkan pada Gambar 2.3. . MQTT menggunakan topik sebagai *filter* di *broker*. Setiap paket harus berisi topik yang dapat digunakan oleh *broker* untuk meneruskan paket ke *client* yang melakukan *subscribe*. Pada umumnya paket mempunyai muatan yang berisi data untuk dikirimkan dalam format byte. Muatan dapat berupa data biner, text, XML atau



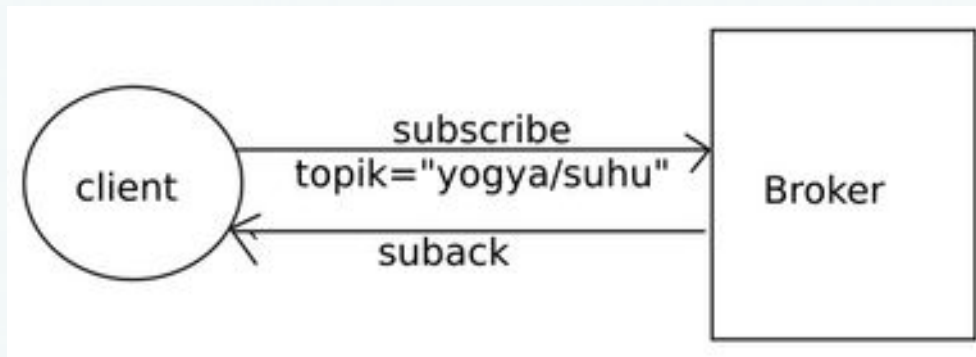
Gambar 2.3. Client mempublish pada topik=yogya/suhu.



Protokol MQTT

Client Subscribe

Data yang di *publish* oleh *client* pada suatu topik dapat diterima oleh *client* yang lainnya dengan melakukan *subscribe* pada topik dan *broker* yang sama. Untuk melakukan *subscribe*, *client* harus mengirimkan paket SUBSCRIBE dan jika berhasil akan dijawab oleh *broker* dengan paket SUBACK sebagaimana diperlihatkan pada Gambar 2.4.



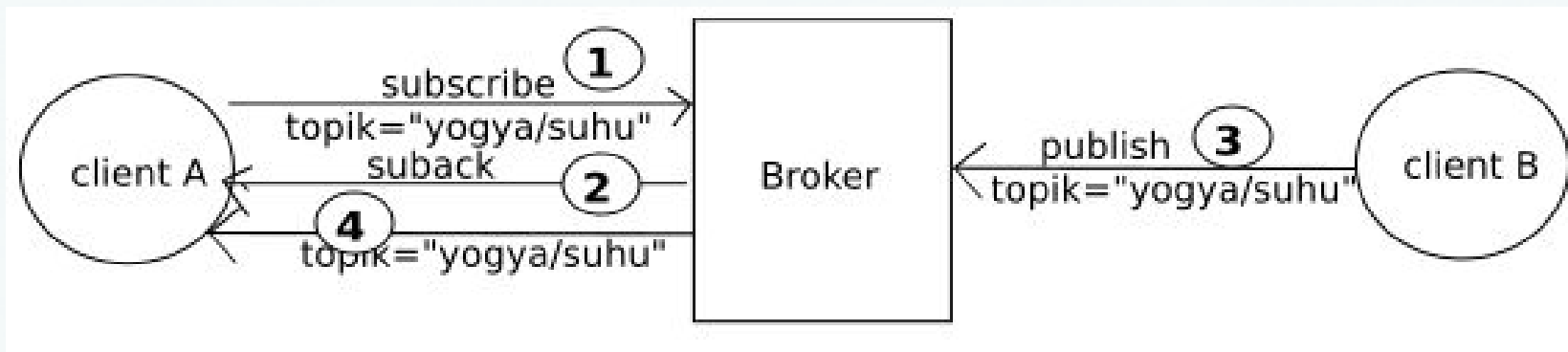
Gambar 2. 4 Client subscribe pada suatu topik



Protokol MQTT

Client Subscribe

Client yang sudah mengirimkan paket *SUBSCRIBE* dan menerima paket *SUBACK* akan mendapatkan kiriman data dari *client* yang melakukan *publish* pada topik tersebut, seperti diperlihatkan pada Gambar 2.5.

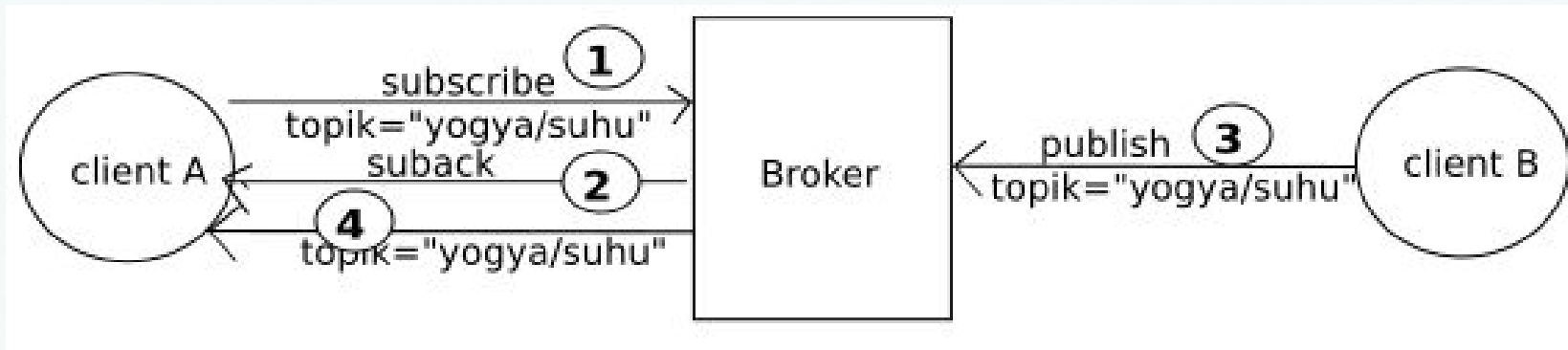


Gambar 2.5. *Client* yang *subscribe* pada topik dan *broker* tertentu

Protokol MQTT

Client Unsubscribe

Paket *UNSUBSCRIBE* merupakan kebalikan dari *SUBSCRIBE*. *Client* yang akan memutuskan untuk tidak menerima kiriman data dari suatu topik dapat mengirimkan paket *UNSUBSCRIBE*. Paket *UNSUBSCRIBE* berisi identifikasi paket dan list topik yang akan diputus.



Gambar 2.5. *Client* yang *subscribe* pada topik dan *broker* tertentu



Protokol MQTT

Topik

Pada protokol MQTT, topik menggunakan format string UTF-8 yang digunakan broker untuk menfilter pesan untuk setiap client yang terkoneksi. Topik dapat bertingkat. Setiap tingkat dipisahkan dengan karakter garis miring (/).

Topik bersifat case sensitive sehingga huruf kapital dengan non kapital dianggap berbeda. Contoh Topik sebagai berikut.

Yogya/sensor/suhu berbeda dengan yogya/sensor/suhu

Yogya/sensor/CO2 berbeda dengan Yogya/sensor/co2

Client dapat melakukan subscribe ke beberapa topik dengan melakukan subscribe satu persatu atau menggunakan karakter *Wildcard*.



Protokol MQTT

Topik

Karakter *Wildcard* ada dua macam yaitu *Wildcard* satu tingkat dan *Wildcard* banyak tingkat. *Wildcard* satu tingkat menggunakan karakter + (tambah) dan *Wildcard* banyak tingkat menggunakan karakter * (bintang). Karakter *Wildcard* hanya boleh digunakan untuk *client* yang *subscribe* dan tidak diperbolehkan untuk *client* yang *publish*. Contoh karakter *Wildcard* :

Wildcard satu tingkat:

topik= yogya/sensor/+

Client yang subscribe ke topik di atas akan *subscribe* di topik berikut ini.

- yogya/sensor/suhu
- yogya/sensor/kelembaban
- yogya/sensor/co2
- yogya/sensor/gempa



Protokol MQTT

Topik

Wildcard banyak tingkat:

topik= yogya/*

Client yang subscribe ke topik di atas akan subscribe di topik berikut ini.

- yogya/sensor/suhu
- yogya/sensor/kelembaban
- yogya/triffict/janti
- yogya/triffict/demangan



Protokol MQTT

Topik

Penggunaan praktis yang baik untuk topik adalah sebagai berikut [2]:

1) Jangan menggunakan karakter / (garis miring) pada awal nama topik.

Contohnya: */yogya/sensor*

2) Jangan menggunakan spasi pada nama topik.

Contohnya: *yogya selatan/sensor*

3) Buat topik yang pendek namun bermakna.

Contohnya: *yogya/sensor/suhu/janti*

4) Gunakan karakter ASCII jangan menggunakan karakter yang tidak terbaca.



Protokol MQTT

Kualitas Layanan (QoS)

Kualitas layanan (QoS) adalah persetujuan antara pengirim pesan dan penerima pesan yang mendefinisikan jaminan pengiriman untuk pesan tertentu. Pada protokol MQTT pengiriman dan penerimaan pesan terdiri atas dua bagian yaitu,

- 1) pengiriman pesan dari *client (publisher)* ke *broker*
- 2) pengiriman pesan dari *broker* ke *client (subscriber)*.

Client yang mengirim pesan ke *broker* mendefinisikan QoS pesannya. *Broker* akan mengirimkan pesan ke *client* menggunakan QoS pada saat proses *subscribe* ke topik.

Level QoS pada protokol MQTT ada tiga, yaitu:

- 3) QoS level 1
- 4) QoS level 2
- 5) QoS level 3

6)



Protokol MQTT

Broker MQTT

1. Mosquitto <https://mosquitto.org/>
2. CloudMQTT <https://www.cloudmqtt.com/>
3. HiveMQ <https://www.hivemq.com>
4. Flespi <https://flespi.com>
5. VernerMQ <https://vernemq.com/>
6. ActiveMQ <https://activemq.apache.org/>
7. Joram <https://joram.ow2.io/>
8. RabbitMQ <https://www.rabbitmq.com/>
9. EMQ <http://emqtt.io/> <https://www.emqx.io/>



Protokol MQTT

MQTT Desktop Client Tools

- 1. MQTTX (<https://mqttx.app/>)*
- 2. MQTT Explorer (<https://mqtt-explorer.com/>)*
- 3. MQTT.fx (<https://softblade.de/en/download-2/>)*



Protokol MQTT

OS for internet of things

1. Contiki <http://contiki-os.org/>
2. MBED www.mbed.org
3. RIOT <https://riot-os.org/>
4. FreeRTOS www.freertos.org (library for connection to mqtt broker)
5. Ubuntu IoT www.ubuntu.com (Raspberry Pi, NUC)
6. Mongoose OS <https://mongoose-os.com>

