Macam-macam Protocol IoT

IOT Protocols

- Constrained Application Protocol (CoAP)
- Message Queue Telemetry Transport Protocol (MQTT)
- Advanced Message Queuing Protocol (AMQP)
- Data Distribution Service (DDS)
- HTTP

HTTP

Pengenalan HTTP

Hypertext Transfer Protocol (HTTP) adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia.

HTTP adalah dasar dar World Wide Web (WWW) yang digunakan untuk memuat halaman web menggunakan tautan hypertext.

Saat ini HTTP sudah banyak berkembang sehingga memiliki banyak kegunaanya itu tidak hanya untuk mengambil dokumen hypertext, tetapi juga gambar dan video atau untuk mengirim konten ke server, misalnya hasil formulir HTML.

HTTP

Request dan Response

Sistem HTTP Response merupakan sebuah aktivitas dimana server memberikan jawaban terhadap request yang dikirim oleh client sebelumnya. Respon yang diberikan oleh server setelah mengirimkan data yang diinginkan adalah mengirimkan status line atau response header .Berikut ini adalah beberapa status header:

- 1xx, artinya request dari client diterima server, dan dilanjutkan dengan memberikan tindakan selanjutnya dari client
- 2xx, artinya server berhasil merespon permintaan client
- 3xx, artinya adanya redirection (pengalihan permintaan client)
- 4xx, artinya error pada client. Sehingga server tidak dapat membaca dan memproses permintaan client (kesalahan sintak HTML, atau sintak yang ditulis tidak full)
- 5xx, artinya terdapat kesalahan internal, biasanya berasal dari kesalahan pada database server.

HTTP

Cara Kerja HTTP

HTTP membantu user dalam mengakses berbagai web resources seperti halaman website, dokumen, video, dll.

Pada web server. User menggunakan browser agar dapat mengakses berbagai macam resource didalam web server dengan menggunakan URL.

Protokol HTTP membantu koneksi yang membantu user untuk web server dengan URL tersebut.

HTTP

Cara Kerja HTTP

Contoh penggunaan http adalah saat kita mengunjungi https://indobot.co.id/blog/monitoring-suhu-air-tambak-udang-berbasis-iot/

Protokol	Alamat Web Server	Lokasi Source didalam Web Server
https://	Indobot.co.id	monitoring-suhu-air-tambak-udang-berbasis-iot/

Melalui URL tersebut, protokol HTTP bertugas untuk mengirimkan permintaan (request) dan memberikan respon (response) dari request URL yang diberikan. Contohnya seperti menghapus (deleted), memperbarui (update), atau mengganti (replace) file yang ada didalam server.

HTTP

Request dan Response

HTTP request merupakan aktivitas dimana client meminta data kepada server melalui alamat URL tertentu. Biasanya proses request dilakukan oleh client melalui web browser seperti Chrome, Firefox, Safari, dll.

MQTT

Pengenalan MQTT

MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi pengiriman pesan yang dibentuk dengan TCP/IP berdasarkan model messaging publish-subscribe dan dirancang khusus untuk machine to machine yang tidak memiliki alamat khusus.

Publisher mengirim pesan, subscriber menerima pesan yang dibutuhkan, dan broker akan menyampaikan pesan yang diminta, dari pengirim ke penerima. Publisher dan subscriber adalah klien MQTT yang hanya berkomunikasi dengan broker MQTT. Contoh publisher dan subscriber adalah mikrokontroller, komputer, smartphone, dll.

MQTT

Broker, Publish, Subscribe dan Topic

1. Broker

Broker pada MQTT berfungsi untuk mengorganisir data publish dan subscribe dari berbagai device. Broker bisa diibaratkan sebagai server data yang memiliki alamat IP khusus. Beberapa contoh dari Broker yang ada seperti Mosquitto, HiveMQ, dan Mosca.

MQTT

Broker, Publish, Subscribe dan Topic

2. Publish

Publish merupakan cara suatu perangkat untuk mengirimkan datanya ke subscribers. Pengirim data tersebut dinamakan publisher. Biasanya pada publisher adalah sebuah device yang terhubung dengan sensor tertentu.

Contoh publisher adalah Arduino, ESP32, RaspberryPi, Android, dll.

MQTT

Broker, Publish, Subscribe dan Topic

3. Subscribe

Subscribe merupakan cara suatu device untuk menerima berbagai macam data dari publisher melalui broker. Penerima data disebut subscriber.

Subscriber dapat berupa aplikasi monitoring sensor dan sebagainya, subscriber ini yang nantinya akan meminta data dari publisher melalui broker.

MQTT

Broker, Publish, Subscribe dan Topic

4. Topic

Topic seperti halnya pengelompokan data disuatu kategori tertentu yang dibuat oleh user. Pada sistem kerja MQTT protokol ini, topic bersifat wajib untuk membedakan data satu dengan yang lainnya.

Pada setiap transaksi data antara Publisher dan Subscriber harus memiliki suatu topic tertentu. Contoh penggunaan topic: “temperature”, “humidity“, dll.

MQTT

Cara Kerja MQTT

MQTT sejak awal memang dirancang untuk komunkasi Machine-to-Machine. MQTT mengirim data sebagai byte array, yang mana ini menjadi kelebihan protokol MQTT karena data yang dikirimkan sangat kecil. Pada saat Protokol MQTT di uji dengan jaringan 3G, Protokol MQTT 93 kali lebih cepat dari pada HTTP, karena paket headernya hanya 2 byte. Publisher mengirimkan data/messages dengan topic tertentu kepada broker. Pengiriman data dapat melalui WiFi, GSM, LTE, dll. Subcriber kemudian mengambildata/messages yang diperlukan dengan topik tertentu dibroker.

MQTT

Mosquitto

Eclipse Mosquitto™ adalah Message broker yang mengimplementasikan protokol MQTT versi 3.1 dan 3.1.1. Broker Mosquitto cukup handal terutama dalam sistem internet of things karena sudah digunakan oleh banyak user.

Mosquitto telah mendukung berbagai sistem operasi mulai dari MacOS, Microsoft Windows, dan berbagai varian distro Linux.

MQTT

Keamanan Mosquitto

Mosquitto mendukung keamanan dengan mengenkripsi port yang digunakan dalam transmisi data. Beberapa port yang dapat digunakan yaitu:

- 1883: MQTT, unencrypted, unauthenticated
- 1884: MQTT, unencrypted, authenticated
- 8883: MQTT, encrypted, unauthenticated
- 8884: MQTT, encrypted, client certificate required
- 8885: MQTT, encrypted, authenticated
- 8886: MQTT, encrypted, unauthenticated
- 8887: MQTT, encrypted, server certificate deliberately expired
- 8080: MQTT over Web Sockets, unencrypted, unauthenticated
- 8081: MQTT over Web Sockets, encrypted, unauthenticated
- 8090: MQTT over Web Sockets, unencrypted, authenticated
- 8091: MQTT over Web Sockets, encrypted, authenticated

CoAP

Pengenalan CoAP

- CoAP merupakan suatu protokol yang didesain untuk IoT.
- CoAP merupakan singkatan dari Constrained Application Protocol yang dibuat oleh Internet Engineering Task Force (IETF)
- Hampir sama dengan HTTP
- RESTFul (Representational State Transfer): Client/server structur,stateless:
  - Get: Membaca informasi
  - PUT: Memperbaharui informasi
  - POST : Membuat informasi baru
  - DELETE : menghapus informasi
- CoAP ditujukan pada perangkat yang memiliki keterbatasan dalam energi, RAM, bandwith data, memori, hingga catu daya.

CoAP

Pengenalan CoAP

CoAP adalah alternatif protokol yang bisa digunakan disamping HTTP yang lebih rakus terhadap Resource. CoAP sendiri, merupakan standar dari Internet Engineering Task Force (IEFT) dan ditetapkan pada RFC7252 https://tools.ietf.org/html/rfc7252.

CoAP melakukan optimasi secara drastis, ia mengubah dua hal. Pertama, CoAP berada diatas UDP bukan TCP. Di UDP, paket -paket IP disebut sebagai datagram, tidak mengenal *acknowledge* atau urutan sebagaimana TCP. Hal ini berarti datagram bisa hilang atau diterima dengan urutan yang berbeda dari yang seharusnya saat dia dikirim. Pertimbangan semacam ini yang harus dipikirkan oleh implementasi CoAP.

Application layer	HTTP, CoAP, EBHTTP, LTP, SNMP, IPfix, DNS, NTP, SSH, DLMS, COSEM, DNP, MODBUS
Network/Communication layer	IPv6/IPv4, RPL, TCP/UDP, uIP, SLIP, 6LoWPAN,
PHY/MAC layer	IEEE 802.11 Series, 802.15 Series, 802.3, 802.16, WirelessHART, Z-WAVE, UWB, IrDA, PLC, LonWorks, KNX

CoAP

Tipe Pesan

- **Confirmable (CON):** Pesan yang berisi request dan memerlukan Acknowledgment
- **Non Confirmable(NON):** Pesan yang digunakan berulang secara teratur tanpamemerlukan acknowledgement
- **Acknowledgement(ACK):** Pesan yang berisi response
- **Reset(RST):** Pesan yang digunakan ketika pesan CON tidak diterima dengan benar atau terdapat konteks yang hilang

CoAP

CoAP Frame Structure

CoAP memiliki 2 bits core version, kemudian 2 bits pesan yang dikirimkan, 4 bits dari panjang tokenya, namun token length akan bernilai 0 apabila kita tidak menggunakannya, namun token akan sangat berguna untuk mengidentifikasi komunikasi request response. Kemudian akan mengirimkan code, pesan, token, dan seterusnya

CoAP

CoAP Frame Structure

- **Ver :** 2 bit unsigned integer. Menyebutkan versi CoAP
- **Type:** 2 bit unsigned integer. Mengindikasikan pesan (confirmalbe ()), non confirmable (1), ACK(2), or Reset(3)
- **Token Length:** 4 bit unsigned integer, mengindikasikan panjang dari tokennya
- **Code:** response dari suatu kodenya
- **Massage ID:** Mengidentifikasi dari setiap pesan yang dikirimkan

CoAP

Distribusi Data yang Efisien

CoAP juga memperkenalkan **pola komunikasi baru yang disebut sebagai Multicast dan Event / Observe**. Karena CoAP berdasarkan pada UDP dan UDP mendukung multicasting, menggunakan Internet Group Management Protocol atau IGMP, pesan-pesan CoAP bisa juga menggunakan multicast, setidaknya ketika enkripsi tidak digunakan.

Ini bisa dilakukan dengan mengirim pesan CoAP ke alamat multicast yang telah ditentukan sebelumnya

CoAP

Observe Pattern

- **Pattern observe (event subscription)** berfungsi untuk sensor yang melaporkan data secara rutin dan aman, optimisasi yang terbesar, dengan menggunakan patern ini.

- **Subscription** merupakan pihak yang melakukan request (requestor) atau subscriber hanya perlu mengirim permintaan sekali.

- Sama seperti request pada biasanya, perangkat akan mengembalikan respon yang berisi data yang ingin dikembalikan. Perbedaannya terletak pada resource yang berubah. Ketika hal ini terjadi perangkat akan otomatis mengirim respon baru ke subscriber tanpa harus menunggu request yang sesuai.

CoAP

Observe Pattern

- Pada situasi yang normal, hal ini akan mengurangi jumlah pesan yang dikirim dengan faktor (pendekatan) dua.
- Dari empat pesan, dapat dikurangi menjadi satu bergantung pada arah komunikasi.
- Jika data dikirim dari perangkat ke penerima, maka jumlah pesan yang dikirim dapat dipangkas hingga 4 pada kasus normal tanpa acknowledge.
- Jika data dikumpulkan oleh client, jumlah pesan yang diterima bergantung pada seberapa sering pengumpulan dilakukan

CoAP

Kelebihan CoAP

- Pengoperasinnya tidak membutuhkan daya yang besar
- CoAP digunakan untuk perangkat yang dengan energi rendah seperti sensor mikrokontroler, switch dan komponen lainnya.
- Protokol komunikasi yang lebih ringan daripada protokol HTTP.
- CoAP memiliki fitur yang menyerupai HTTP yang menyediakan model interaksi request/response antara aplikasi dan endpoint.

AMQP

AMQP atau *Advanced Message Queuing Protocol*
adalah standar layer aplikasi protokol untuk *Message Oriented Middleware*.
Pada dasarnya, AMQP didesan sebagai standar pertukaran pesan dan dirancang untuk lingkungan *middleware* yang mendukung efisiensi pada komunikasi data yang saat ini sudah sangat beragam.

AMQP

Pemrosesan AMQP IoT Protokol ini terdiri dari 3 komponen yaitu *Exchange, Message Queue, dan Binding*.

Exchange merupakan bagian dari broker yang menerapkan AMQP model yang bertugas menerima pesan dan meletakkannya kedalam *queue*.

Message queue merupakan sebuah entity yang berfungsi untuk menampung pesan dan *consumer* mana yang akan menerima pesan tersebut. Sedangkan *bindings* berfungsi mendistribusikan pesan dari *exchanges* ke *queue*.

Ketiga komponen ini memastikan pertukaran dan penyimpanan pesan berhasil dan membantu dalam membangun hubungan antar pesan. Protokol AMQP ini biasa digunakan dalam industri perbankan. Setiap kali pesan dikirim oleh *server*, protokol melacak pesan sampai setiap pesan dikirim ke pengguna atau tujuan yang dituju tanpa adanya kegagalan.

XMPP

XMPP dirancang secara unik dengan menggunakan *'push mechanism'* agar dapat bertukar pesan secara *realtime*.

XMPP sendiri merupakan protokol yang fleksibel dan dapat diintegrasikan dengan berbagai perubahan.
XMPP ini dikembangkan menggunakan XML terbuka *(Extensible Markup Language)* yang berfungsi sebagai indikator keberadaan yang menunjukkan status ketersediaan *server* atau perangkat yang mengirim atau menerima pesan.

XMPP

Keunggulan XMPP adalah tidak perlu adanya master server pusat untuk menjalankannya.

Untuk security XMPP sudah disertakan spesifikasi untuk keamananannya menggunakan SASL (Simple Authentifikasi Secure Layer) untuk proses Authentifikasi
dan untuk enkripsinya menggunakan TSL (Transport Layer Security).

XMPP *server* dan *client* berkomunikasi melalui pengiriman
Message Stanza berupa XML code yang fleksibel. Message stanza ini sering disebut dengan packet, yang bisa berupa Message, Presence dan IQ.

banyak aplikasi pesan instan menggunakan XMPP seperti *Google Talk* dan *WhatsApp*.

XMPP juga digunakan dalam *game online*, situs web berita, dan *Voice over Internet Protocol* (VoIP).

End of Session

Reference :

PROA DTS Digital Talent Scholarship Kominfo – Indobot Academy
edspert.id – Internet of Things
www.kmtech.id/mengenal-internet-data-protokol-untuk-internet-of-things
sis.binus.ac.id/2019/11/15/iot-data-protocols/