BIG DATA (TFC 303)

Pertemuan 5 – Data Acquisition

ALIFIA REVAN PRANANDA

Department of Information Technology
Faculty of Engineering
Universitas Tidar

TODAY'S MATERIALS

- ☑ Definition of Data Acquisition
- ☑ Data Acquisition Example

DEFINITION OF DATA ACQUISITION

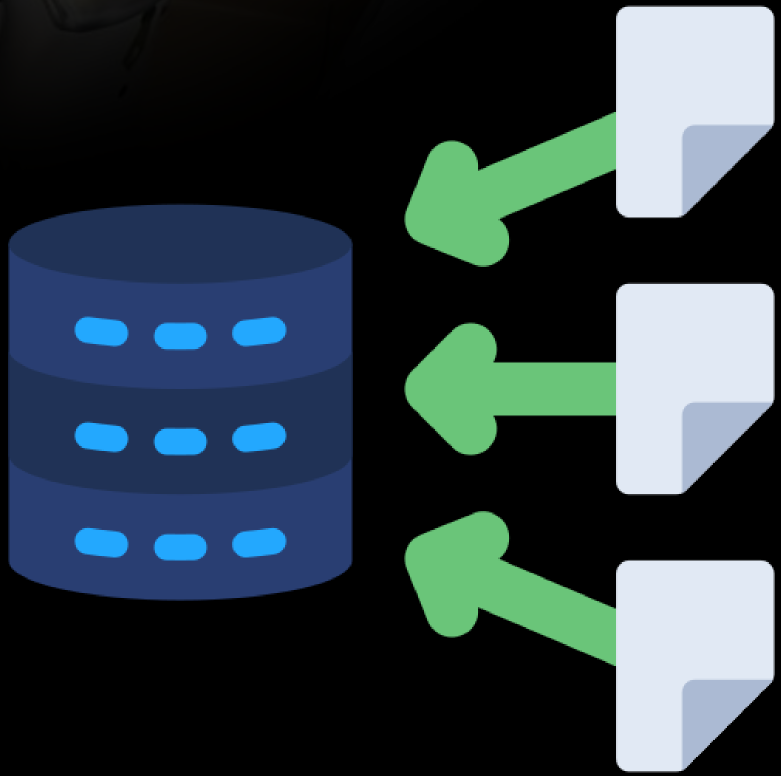
WHAT IS DATA ACQUISITION?

- "DATA ACQUISITION" is the process of gathering, filtering, and cleaning data before it is put in a data warehouse or any other storage solution on which data analysis can be carried out.
- Data acquisition is one of the major big data challenges in terms of infrastructure requirements.
- Most data acquisition scenarios always related to how to manage high-volume, high-velocity, high-variety of data.
- Good quality of data will produce good data interpretation.

WHY DO WE NEED TO USE IT?

- Sensors receive data
  - ✓ It is possible to generate noise.
- Sensors data were transferred to computer
  - ✓ We have to change the data type.
- Read the data
  - ✓ Data redundancy may exist in here.
- Analyse and visualize the data.

DATA ACQUISITION CONSIDERATION



1) **Business Needs** : why are these data required? What will be done with them?

2) **Business Rules** : the constraints under which the business operates

3) **Data Standards** : gold standard of data

4) **Accuracy Requirements** : targeted performance

5) **Cost** : Cost is always a consideration. Sometimes it's cheaper to buy than to collect

6) **Currency of Data** : for many types of work, the data need to be fairly current. For others, data may need to cover a specified time period. For others, data need to be in a specific season.

7) **Time Constraints** : You should determine how soon you need the data

8) **Format** : Do you need the data as spatial data, photos, flat files, Excel files, XML files? This may not apply, but you need to determine that for each project

MAIN COMPONENT OF DATA ACQUISITION

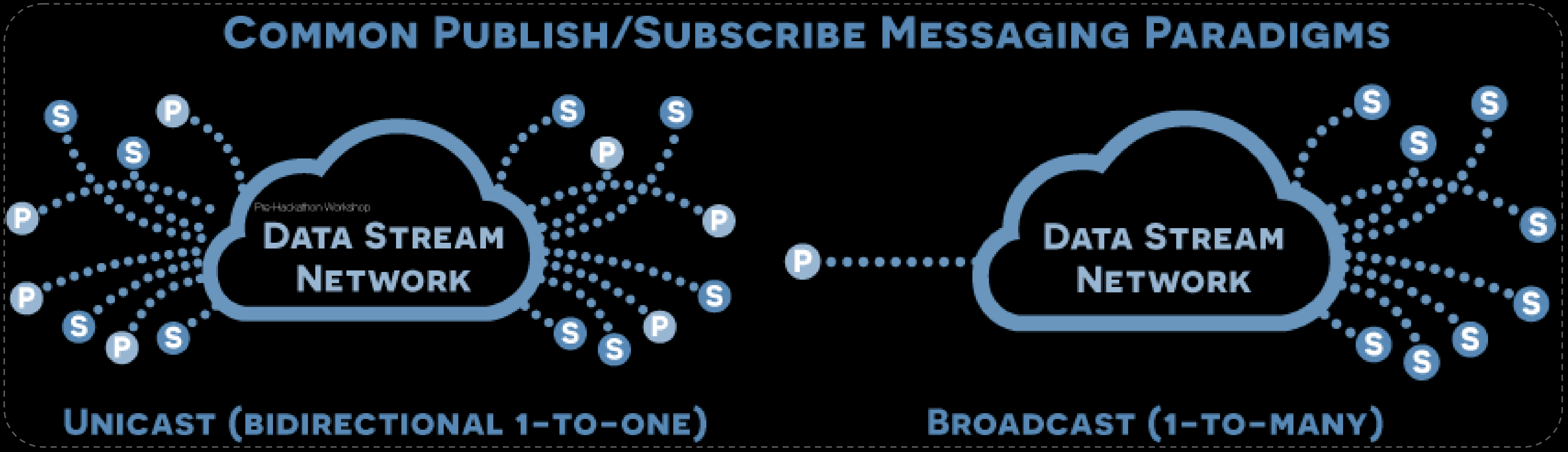
To produce good quality of data, three main components are required:

- **PROTOCOLS** that allow the gathering of information for distributed data sources of any type (unstructured, semi-structured, structured).
- **FRAMEWORKS** with which the data is collected from the distributed sources by using different protocols.
- **TECHNOLOGIES** that allow the persistent storage of the data retrieved by the frameworks.

DATA ACQUISITION MODEL

GATHERING DATA (USING PUBLISH-SUBSCRIBE)

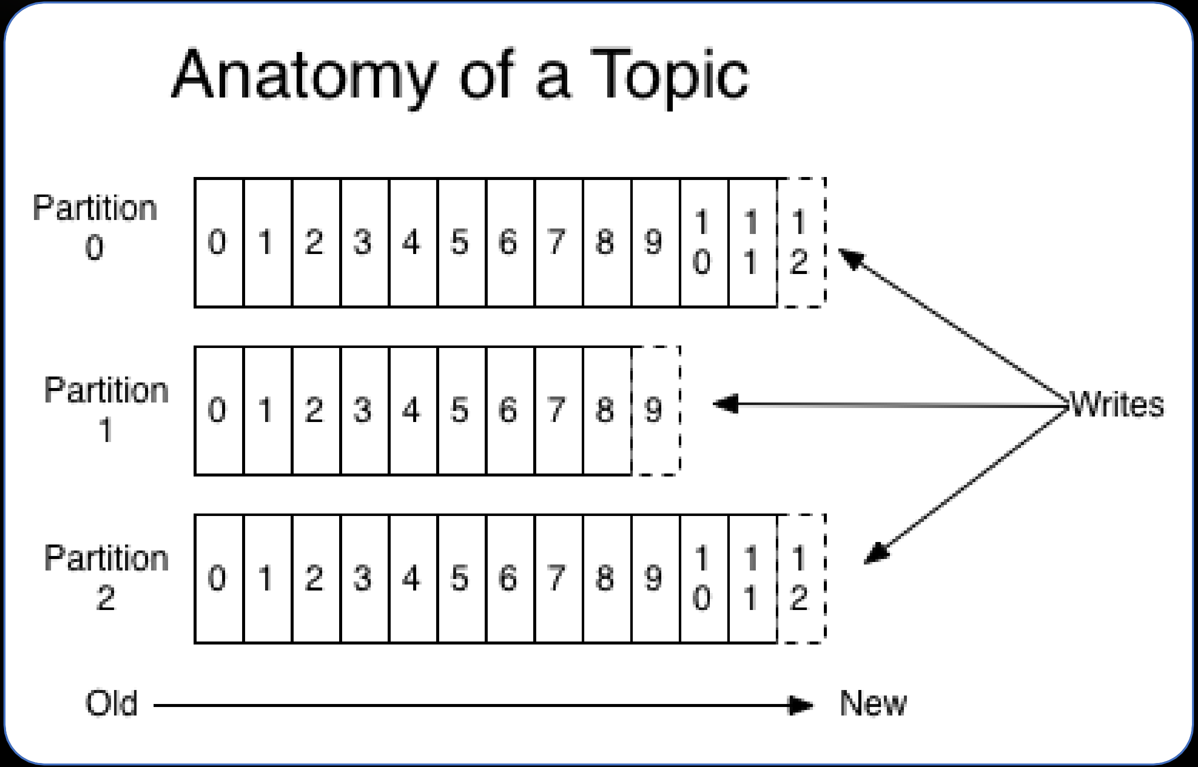
Publish–subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead categorize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.



GATHERING DATA (USING PUBLISH-SUBSCRIBE)

Architecture of the Publisher/Subscriber model

The main level of abstraction of the model is “ **the topics** ”.

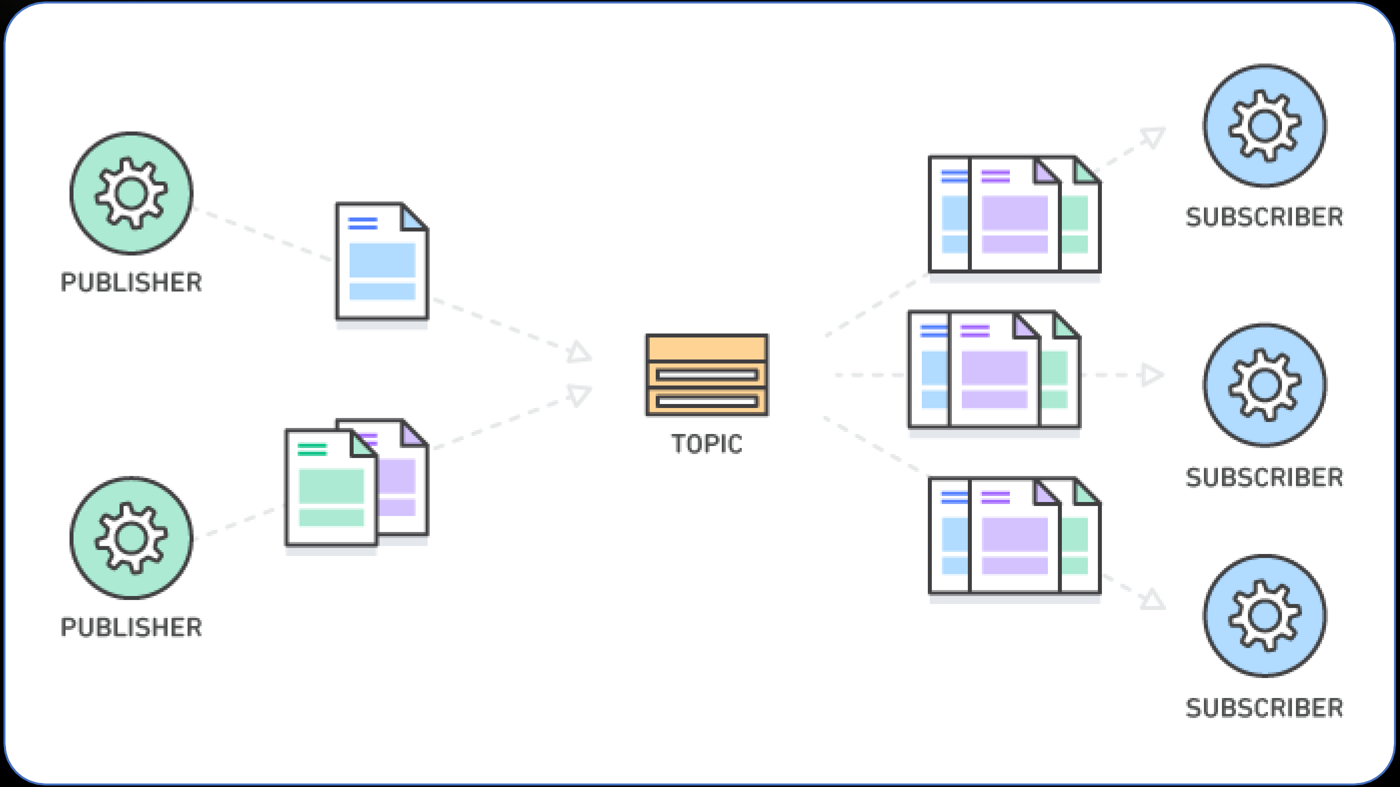


- **A TOPIC IS** a category or feed name to which records are published.
- It can be understood as a communication channel that receives information through some kind of protocol (FTP, TCP/IP, UDP, HTTP, SMTP, etc) and whose information is retained in order of arrival and can be accessed by more than one source of consultation (multi-subscriber).
- The structure of a topic can vary depending on the streaming system that we implement.

GATHERING DATA (USING PUBLISH-SUBSCRIBE)

Architecture of the Publisher/Subscriber model

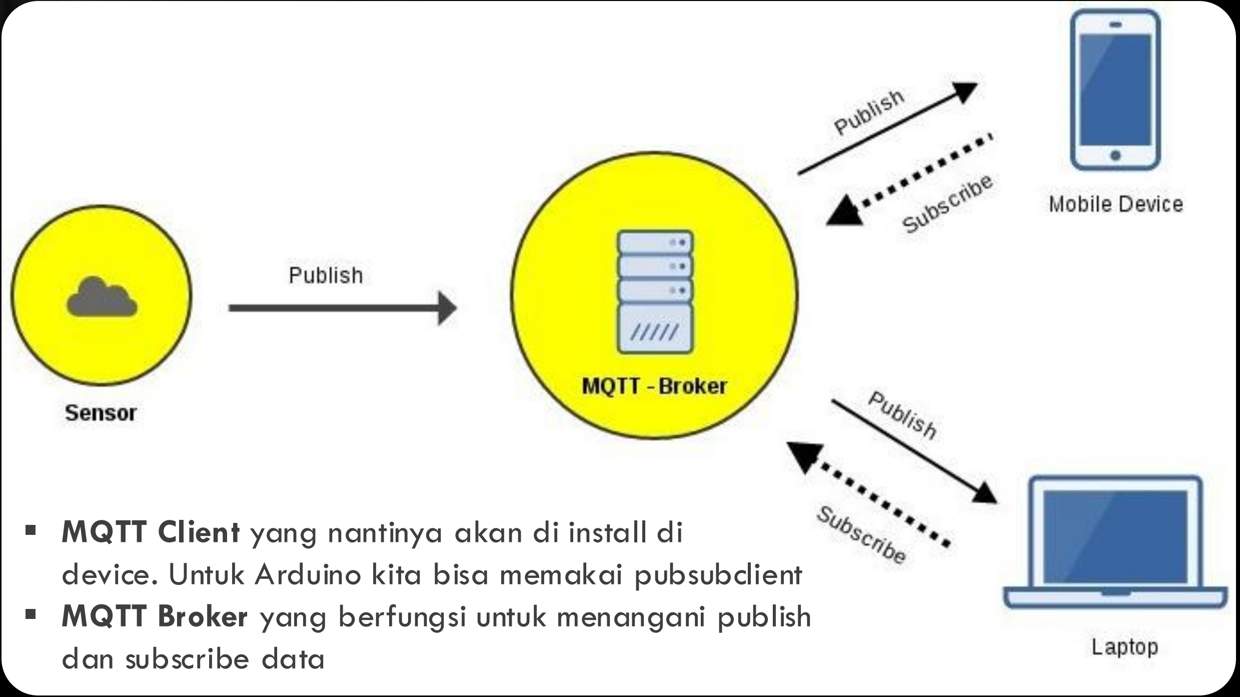
Here is the illustration of how the topic works:



- Topics are composed of ordered, immutable sequence of records that is continually appended to — a structured commit log.
- The records in the topics are each assigned a sequential id number called the offset that uniquely identifies each record within the topic.

GATHERING DATA (USING PUBLISH-SUBSCRIBE)

PROTOCOL OF PUBLISH-SUBSCRIBE



- **MQTT Client** yang nantinya akan di install di device. Untuk Arduino kita bisa memakai pubsubclient
- **MQTT Broker** yang berfungsi untuk menangani publish dan subscribe data

- MQTT adalah sistem publish-subscribe di mana kita dapat menerbitkan dan menerima pesan sebagai klien.
- MQTT adalah sebuah protokol jaringan yang menjadi penghubung komunikasi mesin ke mesin.
- Protokol ini berguna untuk koneksi ke lokasi terpencil di mana bandwidth menjadi sesuatu yang mahal/langka.

MQTT cukup sederhana dan dirancang untuk perangkat terbatas atau dengan bandwidth rendah, karenanya dapat menjadi **solusi terbaik untuk diterapkan pada Internet of Things (IoT)**.

GATHERING DATA (USING PUBLISH-SUBSCRIBE)

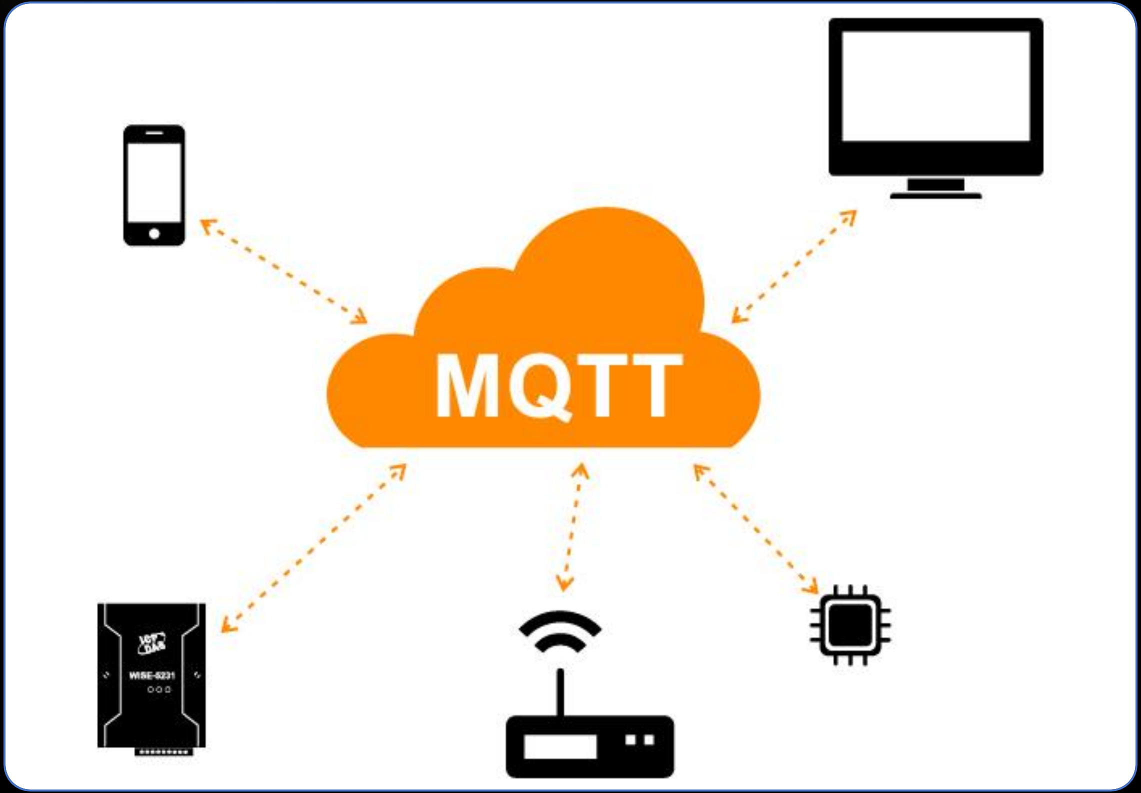
MENGAPA MQTT PENTING UNTUK IOT?

- Sistem IoT memiliki beberapa perangkat seperti sensor dan embedded device.
- Ukuran media penyimpanan internal dan RAM dari perangkat IoT biasanya relatif cukup kecil seperti Arduino, Nucleo STM32.
- Sehingga, terdapat beberapa optimasi yang harus diperhatikan dalam pertukaran data pada sistem IoT agar transfer data bisa seefisien mungkin. Optimasi tersebut meliputi:
  - Menekan ukuran paket data sekecil mungkin sehingga trafik bisa meningkat.
  - Meminimalisasi proses komputasi untuk encoding dan decoding dari paket data.
  - Data hanya menggunakan ruang penyimpanan yang sekecil mungkin.

GATHERING DATA (USING PUBLISH-SUBSCRIBE)

KARAKTERISTIK MQTT

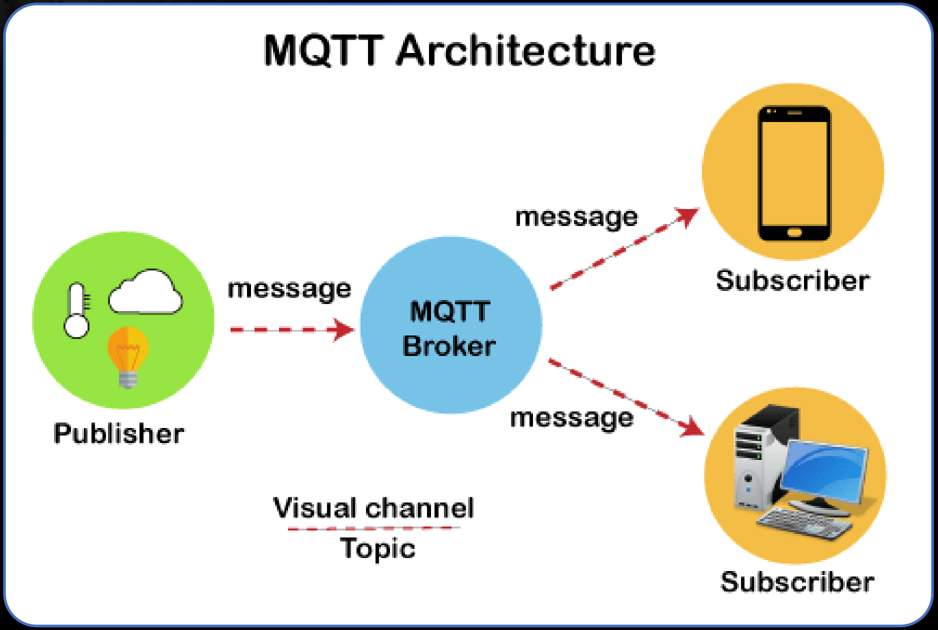
MQTT memiliki beberapa fitur unik yang jarang ditemukan di protokol lain.



- Merupakan protokol mesin ke mesin, yaitu protokol yang menyediakan komunikasi antar perangkat.
- Dirancang sebagai protokol pesan sederhana dan ringan yang menggunakan sistem publish/subscribe untuk bertukar informasi antara klien dan server.
- MQTT tidak mengharuskan klien dan server membuat koneksi pada saat yang sama.
- Memberikan transmisi data yang lebih cepat dan realtime.
- Memungkinkan klien untuk melakukan subscribe pada pilihan topik tertentu sehingga klien dapat menerima informasi yang mereka cari.

GATHERING DATA (USING PUBLISH-SUBSCRIBE)

KOMPONEN MQTT



1. **Pesan** : data yang dibawa keluar oleh protokol ke seluruh jaringan.
2. **Klien** : Pada MQTT, subscriber dan publisher adalah dua peran yang dijalankan oleh klien. Klien memiliki dua operasi yakni **publish** (klien mengirim data ke server) dan **subscribe** (klien menerima data dari server).
3. **Server atau Broker** : perangkat atau program yang memungkinkan klien untuk subscribe pesan dan publish pesan
4. **Topik** : Dalam MQTT dikenal istilah topik yaitu berupa UTF-8 string yang perannya hampir sama seperti topik pada chat hanya saja lebih sederhana dan berfungsi sebagai filter untuk broker atau server dalam mengirimkan pesan ke tiap klien yang terhubung.

FILTERING DATA (USING PUBLISH-SUBSCRIBE)

In the publish-subscribe model, subscribers typically receive only a subset of the total messages published. The process of selecting messages for reception and processing is called **FILTERING**.

There are two type of filtering: topic-based and content-based.

- **Topic-based system**: messages are published to "topics" or named logical channels. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe. The publisher is responsible for defining the topics to which subscribers can subscribe.
- **Content-based system**: messages are only delivered to a subscriber if the attributes or content of those messages matches constraints defined by the subscriber. The subscriber is responsible for classifying the messages.

Some systems support a hybrid of the two; publishers post messages to a topic while subscribers register content-based subscriptions to one or more topics.