

## BAB II KAJIAN PUSTAKA & DASAR TEORI

Bab ini berisi kajian pustaka dan dasar teori yang digunakan untuk penelitian. Kajian Pustaka akan membahas terkait penelitian yang sudah ada yang dapat mendukung dalam proses penelitian yang diajukan. Dasar teori berisi teori yang diperlukan pada penelitian ini.

### 2.1. Kajian Pustaka

Kajian Pustaka merupakan sub bab untuk membahas penelitian yang telah dilakukan sebelumnya yang digunakan untuk mendukung penelitian yang diusulkan. Penelitian dilakukan oleh Kodali & Soratkal (2017) yang berjudul "*MQTT based Home Automation System Using ESP8266*" yakni mengimplementasikan protokol MQTT di sebuah *board* NodeMCU yang terdapat modul ESP8266. Penelitian dibuat untuk automasi sistem rumah yang cerdas dan hemat energi dengan sensor terhubung kepada NodeMCU. Broker MQTT yang digunakan dalam penelitian tersebut yaitu MQTT Mosquitto yang merupakan salah satu platform yang berbasis *open source*. Dari hasil percobaan yang telah dilakukan oleh penulis sistem *publish* dan *subscribe* berjalan dengan baik pada NodeMCU.

Penelitian yang dilakukan oleh Mulya, dkk (2017) yang berjudul "*Implementasi Gateway berbasis NRF24L01 dan ESP8266 pada Protokol Message Queue Telemetry Transport - Sensor Network (MQTT-SN)*" yakni mengaplikasikan sistem yang berfungsi untuk *client* mengenali ID *gateway*. Penulis menggunakan mekanisme *discovery* yang terdapat pada protokol MQTT-SN dengan beberapa *device* yang digunakan seperti Arduino Pro Mini dan NRF24L01 pada *client* dan menggunakan Arduino Pro Mini, NRF24L01, ESP8266 dan LCD pada *device gateway*. Dari hasil pengujian yang dilakukan oleh penulis, bahwa *client* dapat mengenali ID *gateway* dan dapat terkoneksi dengan *gateway* dengan waktu sekitar 123.406  $\mu$ S.

Penelitian selanjutnya dilakukan oleh Hanifah, dkk (2017) yang berjudul "*Implementasi Quality of Service pada Protokol Message Queue Telemetry Transport – Sensor Network (MQTT-SN) Berbasis Arduino dan NRF24L01*" dengan penelitian menguji dan membandingkan beberapa *Quality of Service* (QoS) yang terdapat pada MQTT-SN yaitu QoS 0, 1 dan 2. Penulis melakukan penelitian menggunakan Arduino Nano serta modul komunikasi NRF24L01. Dimana QoS 0 hanya mengirimkan pesan kepada *gateway*, QoS 1 mengirim pesan dan menunggu *acknowledgement* dari *gateway*, sedangkan QoS 2 mengirim pesan dan melakukan konfirmasi kepada *publisher* mengenai data yang dikirim. Dari hasil penelitian yang dilakukan oleh penulis ketiga QoS berjalan dengan baik.

Berdasarkan beberapa penelitian diatas, protokol MQTT dapat diimplementasikan dengan baik terlebih kepada penelitian yang berfokus pada

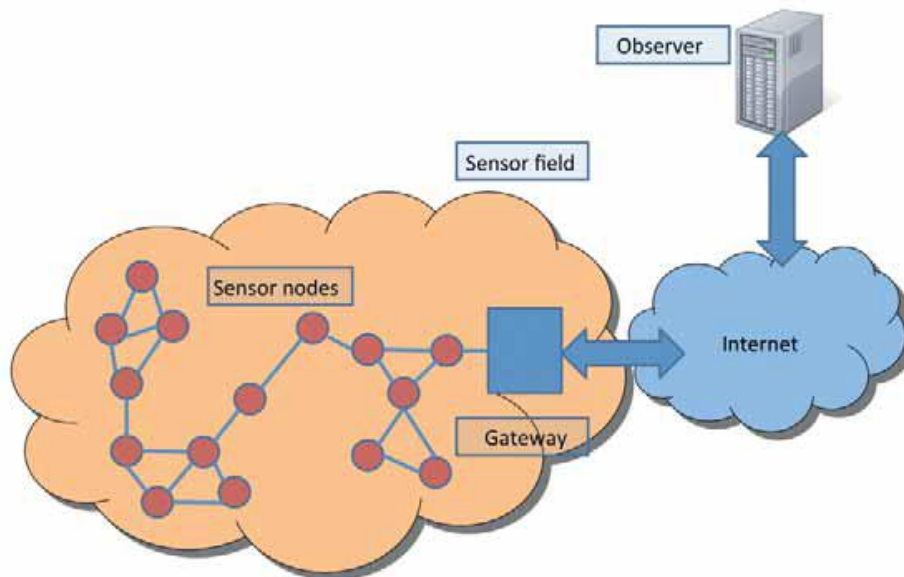
protokol MQTT-SN dimana penulis melakukan penelitian pada protokol MQTT-SN ini yang menggunakan Arduino Nano dan modul komunikasi NRF24L01.

## 2.2. Dasar Teori

Dasar teori membahas berbagai teori yang digunakan untuk mendukung penelitian ini. Pada subbab dibawah menjelaskan berbagai referensi dan teori pendukung mengenai gambaran dari jaringan WSN, protokol MQTT-SN, serta komponen yang digunakan untuk penelitian ini seperti mikrokontroler dan modul komunikasi NRF24L01.

### 2.2.1. *Wireless Sensor Network*(WSN)

*Wireless Sensor Network* atau WSN merupakan sebuah jaringan *node-node* yang saling merasa dan mengontrol lingkungan dan dapat berkomunikasi antara manusia atau komputer dan lingkungan sekitarnya (BRÖRING ,et al., 2011). Arsitektur dari WSN biasanya terdiri dari *node sensor* dan aktuator, *gateway*, dan *client*. Dimana *node sensor* akan disebar secara acak pada sebuah lingkungan yang memerlukan data sensor. *Node sensor* yang tersebar akan melakukan pengambilan data dari lingkungan tersebut dan mengirim data kepada *gateway*. Pada Gambar 2.1 merupakan *node sensor* akan melewati *node sensor* lainnya agar sampai kepada *gateway*. Ketika data sensor diterima *gateway* maka *gateway* akan berkomunikasi dengan internet atau satelit.

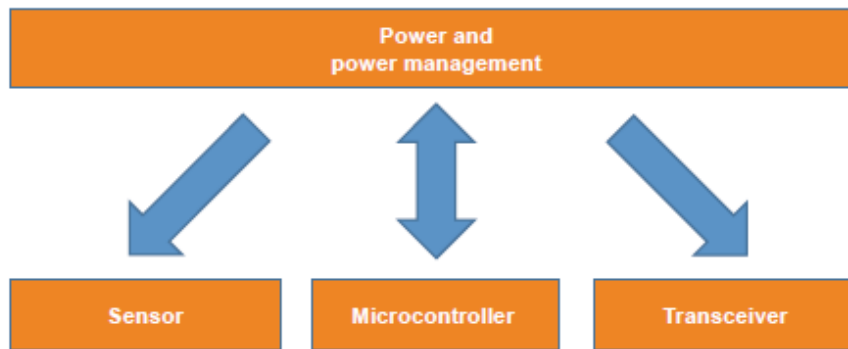


**Gambar 2.1 Arsitektur *Wireless Sensor Network***

Sumber: (IEC Market Strategy Board, 2014)

*Node sensor* merupakan sebuah bagian penting dalam jaringan WSN. *Node sensor* terdiri dari *power*, *sensor*, mikrokontroler, dan modul komunikasi sesuai

dengan Gambar 2.2. *Power* yang merupakan sumber dari *node* sensor dalam melakukan pengambilan, pengolahan, dan pengiriman data sehingga sangat penting dalam penggunaan *power* tersebut. Sensor bertugas melakukan pengambilan data yang berasal dari lingkungan luar dan mengirimkan kepada mikrokontroler agar diproses dan dilanjutkan dengan mengirim data melalui modul komunikasi.

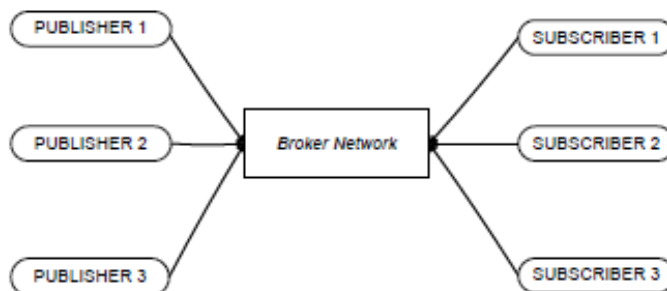


**Gambar 2.2 Struktur Node Sensor**

Sumber: (IEC Market Strategy Board, 2014)

### 2.2.2. Protokol MQTT

*Message Queue Telemetry Transport*(MQTT) merupakan protokol yang *lightweight* atau terbilang ringan dibanding dengan beberapa protokol lainnya seperti HTTP dan COAP dan lain-lain. MQTT hampir sama dengan HTTP yang berjalan pada TCP/IP, namun HTTP merupakan protokol simetris sedangkan MQTT merupakan protokol asimetris (Yokotani & Sasaki, 2016). MQTT merupakan protokol yang dirancang oleh IBM dengan sistem *publish/subscribe* yang cocok digunakan untuk sistem berbasis *Machine To Machine*(M2M) (Tarigan, Sitepu, & Hutagalung, 2014). Ditengah dari *Publisher* serta *Subscriber* terdapat *broker server*, dimana *Publisher* akan mengirim data menuju broker dan selanjutnya data akan dikirim kepada *Subscriber* yang telah tersambung dan disesuaikan dengan *topic* yang tersedia.



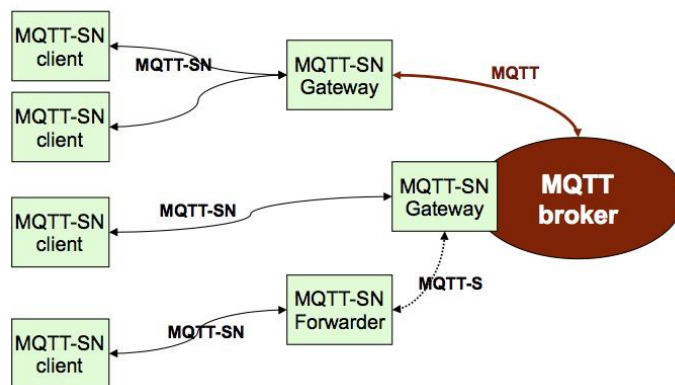
Sumber : (Tarigan, Sitepu, & Hutagalung, 2014)

**Gambar 2.3 Sistem Sederhana MQTT**

Pada gambar 2.3 merupakan konfigurasi sistem dari MQTT, ketika *broker server* yang menjembatani *node publisher* serta *subscriber*. Ada beberapa *platform* yang mendukung MQTT *broker* seperti Trafero Stack, HiveMQ, mosquitto, MQTT.js, Apache Apollo, dan lain-lain.

### 2.2.3. Protokol MQTT – SN

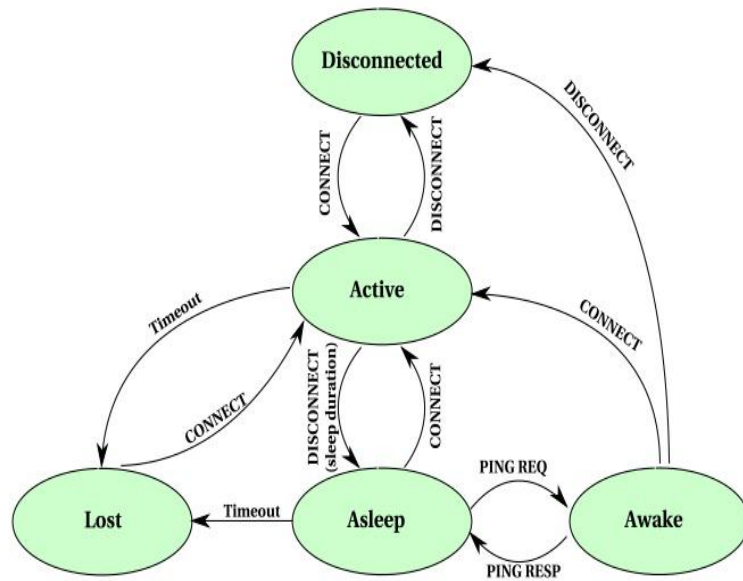
*Message Queue Telemetry Transport for Sensor Network*(MQTT-SN) merupakan versi MQTT yang dapat menyesuaikan dengan lingkungan komunikasi *network*. MQTT-SN sangat baik digunakan untuk implementasi *cost* yang murah, pemrosesan dan penyimpanan yang terbatas.



**Gambar2.4 Arsitektur MQTT-SN**

Sumber: (Stanford-Clark & Truong, 2013)

Pada gambar 2.4 MQTT-SN terdapat beberapa komponen seperti *clients*, *gateway*, dan *forwarders*. *Node client* terhubung dapat terhubung dengan *broker* dengan terhubung lebih dulu dengan *gateway*. Dari *Gateway* menuju *broker* dapat menggunakan MQTT. Dalam MQTT-SN terdapat fitur yang dapat menghemat penggunaan energi pada *node client* yaitu *sleeping client*.



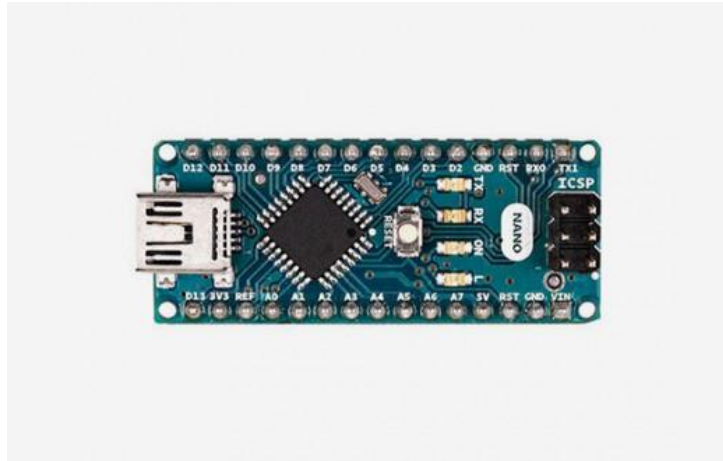
**Gambar2.5 Siklus *Sleeping Client***

Sumber : (Stanford-Clark & Truong, 2013)

Gambar 2.5 dijelaskan terdapat 4 kondisi dari *node client* yaitu: *active*, *asleep*, *awake*, *disconnect*, dan *lost*. *Client* dikatakan *active* ketika *gateway* saling terhubung dan *gateway* menerima pesan dari *node client*, dikatakan *lost* ketika *node client* terhubung dengan *gateway* namun tidak menerima pesan dari *node client*. Sedangkan *disconnect* dijalankan ketika *client* mengirim pesan kepada *gateway* untuk terputus tanpa ada durasi yang ditentukan. Jika *node client* ingin *sleep*, *client* harus mengirim pesan ke *gateway* bahwa akan terputus dengan waktu yang ditentukan, maka *client* akan masuk kondisi *asleep*. *Client* dikatakan *awake* ketika *client* mengirim pesan ke *gateway* bahwa di telah bangun.

#### **2.2.4. Mikrokontroler**

Mikrokontroler merupakan otak dari *node* yang berfungsi mengolah data yang diambil oleh sensor. Mikrokontroler yang digunakan dalam penelitian ini yaitu Arduino Nano yang memiliki cara kerja sama dengan Arduino Uno dan Arduino lainnya namun memiliki ukuran yang lebih kecil dibanding dengan Arduino Uno.



**Gambar2.6 Arduino Nano**

Sumber : (Arduino, 2017)

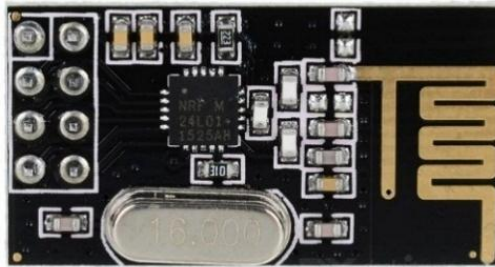
Gambar 2.7 merupakan bentuk fisik dari Arduino Nano. Menggunakan Atmega328 untuk pemrosesan yang berjalan ketika diberi tegangan sebesar 5V dengan kristal sebesar 16MHz. Arduino Nano dapat menghabiskan power sebesar 19 mA. Memiliki pin analog sebanyak 8 buah dan pin digital sebanyak 22 buah.

**Tabel 2.1. Tabel Spesifikasi Arduino Nano**

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

### 2.2.5. NRF24L01

NRF24L01 merupakan chip 2.4 GHz *transceiver* dengan *embedded baseband protocol*. nRF24L01 bekerja pada *world wide ISM* berfrekuensi 2.400 – 2.525GHz. nRF24L01 memiliki fitur seperti 126 RF *channel*, *common RX* dan *TX pin*, *Integrated channel filters*, otomatis paket *handling*, konsumsi *power* yang sangat rendah dan lain-lain.



**Gambar 2.8 NRF24L01**

Sumber : (Nordic Semiconductor, 2017)

Spesifikasi dari NRF24L01 diatas seperti 900nA *deep mode sleep*, 11.3mA Radio TX pada 0dBm, cocok dengan 16MHz  $\pm 60$ ppm *crystal*, otomatis paket *assembly*, otomatis retransmit, SPI bisa mencapai 10Mbps, dengan temperatur sekitar -40 sampai +80 °C. Gambar 2.7 nRF24L01 merupakan gambaran dari bentuk nRF24L01 memiliki ukuran yang kecil yang dapat berkomunikasi sejauh 500 meter.