

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan mengenai kajian pustaka dan landasan teori yang digunakan oleh peneliti. Bab ini dibagi kedalam dua sub bab besar, yaitu kajian pustaka yang menerangkan sumber referensi terkait penelitian sebelumnya, dan dasar teori yang menerangkan tentang teori – teori pendukung penelitian ini.

2.1 Kajian Pustaka

Kajian pustaka terdiri dari beberapa referensi ilmiah yang berasal dari penelitian sebelumnya. Penelitian sebelumnya yang dimaksud adalah *AirSense: Opportunistic Crowd-Sensing based Air Quality Monitoring System for Smart City* (Dutta, 2016), *An Embedded System Model for Air Quality Monitoring* (Jangid, 2016), *Air-Kare: A Wi-Fi Based, Multi-Sensor, Real-Time Indoor Air Quality Monitor* (Yang, 2015), Pengembangan IoT *Middleware* Berbasis Event-Based Dengan Protokol Komunikasi CoAP, MOTT Dan Websocket (Anwari, 2017), dan *Context Aware Computing for The Internet of Things: A Survey* (Charith Perera, 2014).

2.1.1 AirSense

Penelitian yang berjudul “AirSense: Opportunistic Crowd-Sensing based Air Quality Monitoring System for Smart City” (Dutta, 2016). Dutta menawarkan sebuah perangkat yang ringan, rendah daya, dan murah untuk memonitor kualitas udara. Perangkat ini disebut dengan Air Quality Monitoring Device (AQMD). Dutta juga mendemonstrasikan bagaimana AQMD dan *smartphone* dalam keramaian berkolaborasi mengumpulkan dan berbagi data menarik ke penyimpanan *cloud*.

Dutta menggunakan Arduino Pro Mini dalam merancang AQMD. Sensor yang digunakan untuk menentukan kualitas udara adalah MQ-135. Sensor tersebut peka terhadap gas amonia dan benzena. Koneksi antara sensor dan perangkat akhir pengguna menggunakan *Bluetooth*. Perangkat akhir pengguna berupa *smartphone* atau tablet berbasis android yang kemudian perangkat tersebut dipasang sebuah aplikasi untuk menerima data dari Arduino Pro Mini melalui *Bluetooth*. Data kemudian dikirimkan ke server melalui perangkat akhir pengguna, dan aplikasi pada perangkat akan menampilkan peta kualitas udara dan keramaian di lokasi yang dipilih. Peta ini disebut Air Quality Index Map (AQImap).

2.1.2 An Embedded System Model for Air Quality Monitoring

Penelitian yang berjudul “An Embedded System Model for Air Quality Monitoring” (Jangid, 2016). Jangid menawarkan sebuah rancangan sistem *embedded* untuk memantau kualitas udara. Dalam rancangannya, Jangid menggunakan Arduino yang memiliki *processor* ATmega328. Sensor yang ditempelkan pada Arduino ada tiga macam, yaitu MQ-7, DHT-22, dan Sharp GP2Y1010AU0F. Untuk masalah komunikasinya, Arduino dipasangkan dengan modul GSM sebagai *wireless* modem agar dapat menjangkau jaringan internet. Pada perancangan sistem ini, Arduino juga akan dilengkapi dengan sebuah layar untuk melihat data hasil pembacaan oleh sensor. Apabila data yang dibaca sensor

melebihi batas *treshold*, maka Arduino akan mengirimkan pesan SMS kepada pengguna. Kemudian pengguna akan membaca dan membalas sms tersebut dengan perintah khusus. Di sini Jangid menggunakan *Air purifier* dan *dehumidifier* sebagai aktuator. Pengguna mengirimkan pesan perintah khusus yang nantinya akan dapat menyalakan atau mematikan *Air purifier* dan *dehumidifier*.

2.1.3 Air-Kare

Penelitian yang berjudul “Air-Kare: A Wi-Fi Based, Multi-Sensor, Real-Time Indoor Air Quality Monitor” (Yang, 2015). Yang berpendapat bahwa udara bersih yang baik untuk kesehatan dipengaruhi oleh suhu udara, kelembapan udara, kandungan *Volatile Organic Compounds* (VOC), dan kandungan debu. Sensor yang digunakan Yang adalah MQ138 untuk mengukur nilai VOC, AMT2001 untuk mengukur suhu dan kelembapan udara, dan Sharp GP2Y1010AU0F untuk mengukur banyaknya debu dalam udara. Yang menggunakan Arduino Yun yang sudah terintegrasi dengan WiFi untuk membangun node sensor yang ada. Alasan Yang memilih teknologi WiFi untuk alat berkomunikasi dengan perangkat lain dikarenakan jarak jangkanya dan kecepatan perngiriman datanya. Berikut disediakan Tabel 2.1 yang tentang perbandingan teknologi WiFi, *Bluetooth*, dan ZigBee.

Tabel 2.1 Perbandingan teknologi WiFi, Bluetooth, dan ZigBee

| Teknologi | Kecepatan | Jarak | Keamanan | Konsumsi daya |
|-----------|-----------|---------|----------|---------------|
| WiFi | 11-54Mbps | 20-200m | Rendah | 10-50mA |
| Bluetooth | 1Mbps | 20-200m | Tinggi | 20mA |
| ZigBee | 100kbps | 2-20m | Menengah | 5mA |

Sumber: <http://standards.ieee.org/getieee802/download/802.11-2012.pdf> (IEEE, 2012)

Penelitian yang dilakukan oleh Yang mengacu pada penelitian dan beberapa barang yang sudah ada. Hampir semua produk untuk mendeteksi kualitas udara hanya dapat mendeteksi 1 macam polutan. Beberapa produk dapat mendeteksi beberapa parameter sekaligus, seperti BeGood BGFm-08 yang dapat mendeteksi formaldehida, TVOC dan kelembapan serta KHD-FA buatan Qing Feng Kang Hua yang dapat mendeteksi formaldehida, kelembapan, dan suhu. Namun detector ini sangat mahal, hingga berharga 1080CNY (Rp 2.083.478,- dalam rupiah), sementara yang terakhir berharga 2680CNY (Rp 5.170.113,- dalam rupiah) (Yang, 2015). Keduanya juga tidak bisa diakses secara nirkabel dan tidak dapat disimpan datanya di cloud. Data hanya dapat dibaca dari LCD perangkat detector saja.

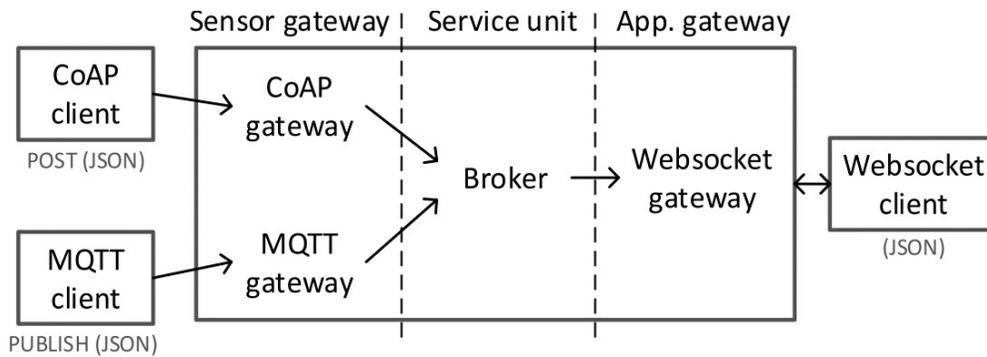
Ching Biau Tzeng sebelumnya telah mengimplementasikan sistem pemantau kualitas udara menggunakan chip CC2430 ZigBee (Ching-Biau Tzeng, 2011). Sistem ini dapat mendeteksi parameter karbon dioksida (CO₂), kelembapan udara, dan suhu udara. Sayantani Bhattacharya juga mengembangkan solusi nirkabel untuk pemantau kualitas udara di dalam ruangan (S., 2012). Solusi ini dapat mengukur gas (CO dan CO₂), aerosol (PM2.5, PM10), suhu ruangan dan kelembapan. Solusi

yang dikembangkan sebelumnya kurang akan kemampuan untuk mengukur data sensor secara tepat waktu dan kepadatan dari banyak pengguna.

2.1.4 Middleware

Penelitian ini berjudul “PENGEMBANGAN IOT *MIDDLEWARE* BERBASIS EVENT-BASED DENGAN PROTOKOL KOMUNIKASI COAP, MQTT DAN WEBSOCKET” (Anwari, 2017). Anwari mengatakan bahwa *Internet of Things* merupakan perpaduan antara internet dan *ubiquitous computing*. IoT melibatkan interaksi antara beragam perangkat seperti sensor, agregator, aktuator dan aplikasi dalam berbagai macam domain. Pada dasarnya IoT terdiri dari dua komponen utama yakni internet dan things. Internet merupakan gabungan infrastruktur jaringan dalam skala masif dan berkembang dinamis berdasarkan standar dan protokol komunikasi yang mendukung interoperabilitas. Sedangkan things merupakan benda atau perangkat baik konkret maupun virtual yang memiliki identitas, atribut, karakteristik dan dapat berkomunikasi satu sama lain melalui sebuah *interface*. Salah satu permasalahan utama Internet of Things adalah interoperabilitas.

Middleware yang digunakan merupakan *middleware* multi-protokol berbasis CoAP dan MQTT. *Middleware* dikembangkan menggunakan paradigma *event-driven* dan arsitektur *ends-to-middle* oleh *Middleware* yang telah dikembangkan oleh Anwari memiliki arsitektur yang terdiri dari tiga bagian, yaitu sensor *gateway*, *service unit*, dan *application gateway*. Arsitektur *middleware* dijelaskan pada gambar 2.1 sebagai berikut.



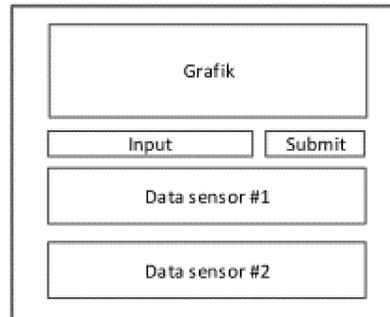
Gambar 2.1 Arsitektur *middleware*

Sumber: Pengembangan IOT *Middleware* Berbasis *Event-Based* dengan Protokol Komunikasi Coap, Mqtt dan Websocket (Anwari, 2017)

Middleware dijalankan menggunakan raspberry pi 2 model B yang sudah terpasang USB Adapter TL-WN722N – TP-Link sebagai *wireless* adapter dan dapat dimanfaatkan oleh raspberry pi sebagai *access point*. Pada gambar 4.3 dapat dilihat bahwa pengiriman data menggunakan pola *publish-subscribe*. *Sensor gateway* menyediakan antarmuka bagi sensor untuk mengirimkan data melalui protokol mqtt dan coap. *Service unit* berfungsi untuk mengatur topik yang *subscribe* oleh *subscriber* dan bertanggung jawab mengirimkan data yang diminta

oleh *subscriber*. *Application gateway* menyediakan antarmuka bagi *subscriber* untuk meminta topik dan menerima data melalui protokol websocket.

Untuk menampilkan data yang dikirimkan oleh sensor ke pengguna, dibuatlah sebuah aplikasi web sederhana menggunakan bahasa pemrograman web seperti HTML, CSS dan Javaprogram. Aplikasi web ini dijalankan pada sebuah datacenter yang terhubung dengan middleware melalui jaringan LAN. Aplikasi ini dapat diakses oleh browser pada smartphone ataupun laptop. Perancangan aplikasi web dijelaskan pada Gambar 2.2 di bawah ini.



Gambar 2.2 Rancangan aplikasi web

Sumber : PENGEMBANGAN IOT *MIDDLEWARE* BERBASIS EVENT-BASED DENGAN PROTOKOL KOMUNIKASI COAP, MQTT DAN WEBSOCKET (Anwari, 2017)

Seperti yang ditunjukkan rancangan di atas, aplikasi ini memiliki beberapa komponen bagi user untuk berinteraksi. Untuk menampilkan data dari middleware, user mengisi nama topik di kolom input kemudian meng-klik tombol submit. Selanjutnya setiap kali sensor mengirimkan data, data tersebut akan ditampilkan secara real-time pada kolom data sensor sekaligus dikonversi menjadi grafik pada kolom grafik. User dapat mengulangi proses yang sama untuk membaca data dari sensor lainnya.

Dari segi teknis, ketika user mengakses dari browser, aplikasi ini menginisialisasi koneksi dengan middleware melalui protokol Websocket. Selanjutnya ketika user memasukkan nama topik dan menekan submit, aplikasi akan melakukan subscribe untuk topik tersebut ke middleware. Ketika sensor mengirimkan data baru, server menyimpan data tersebut terlebih dahulu ke basis data MongoDB barulah mengirim data tersebut ke aplikasi untuk ditampilkan.

2.1.5 Context Aware Computing for The Internet of Things: A Survey

Konteks permasalahan merupakan fitur utama dari komputasi yang menyebar luas, yang sudah ada sejak ditemukan pada awal tahun 1990. Fokus dari komputasi konteks permasalahan berevolusi dari aplikasi *desktop*, aplikasi web, komputasi perangkat bergerak, hingga *Internet of Things* (IoT) pada akhir – akhir ini. Komputasi berdasarkan konteks permasalahan menjadi terkenal bearing dikenalkannya komputasi yang menyebar luas oleh Mark Weiser (Charith Perera, 2014).

Dalam 2 dekade belakangan ini, banyak ilmuwan yang membuat banyak prototip, sistem, dan solusi menggunakan teknik komputasi berbasis konteks permasalahan. Fokus penelitian berbeda – beda tergantung setiap pekerjaan yang dilakukan. Akan tetapi ada satu yang tidak dapat diganti, yaitu jumlah sumber data. Dalam situasi ini, sangat mungkin mengambil dan menganalisa data dari semua sumber karena masih bisa dihitung. Akan tetapi, bayangan tentang IoT pada era dimana jutaan sensor terhubung ke internet, yang artinya tidak mungkin untuk memproses semua data yang didapat dari sensor – sensor tersebut. Oleh karena itu, komputasi yang berbasis konteks permasalahan akan memerankan peran yang sangat penting dalam memilih data mana yang harus di proses.

Ada 3 fitur yang dapat didukung komputasi berbasis konteks, yaitu *presentation*, *execution*, dan *tagging*. Meskipun pandangan IoT tidak diketahui ketika peneliti mengidentifikasi fitur – fitur ini, tetapi fitur ini sangat dapat diaplikasikan ke paradigma IoT. Fitur – fitur tersebut adalah:

- *Presentation*, yang berarti konteks dapat digunakan untuk memilih informasi dan pelayanan seperti apa yang dapat diberikan ke pengguna. Contohnya ketika seorang pengguna datang ke toko dan melihat telepon pintarnya, maka akan ada daftar belanja yang tampil secara otomatis. Data tersebut bisa berasal dari kulkas pintar yang mengetahui persediaan bahan makanan di kulkas.
- *Execution* yang berarti eksekusi layanan secara otomatis merupakan salah satu fitur penting dalam IoT. Contohnya adalah ketika seorang pengguna sedang berjalan pulang kerumah, maka pendingin ruangan dan mesin pembuat kopi akan segera menyala secara otomatis.
- *Tagging* yang berarti penandaan. Dalam paradigma IoT, ada sejumlah besar data yang berasal dari sensor setiap harinya, karena data yang berasal dari sebuah sensor tidak dapat menghasilkan informasi yang cukup. Data yang banyak tersebut membutuhkan penandaan agar data mudah untuk digabungkan dengan data sensor lainnya.

Salah satu peneliti memperkenalkan mekanisme mendefinisikan tipe konteks. Dalam garis besar, tipe konteks dapat dibagi kedalam 2 bagian, yaitu konteks primer dan konteks sekunder. Konteks primer adalah segala informasi yang didapatkan tanpa menggunakan konteks yang sudah ada atau tanpa melakukan operasi penggabungan data sensor. Contohnya adalah pembacaan data sensor GPS yang menunjukkan lokasi seseorang. Konteks sekunder adalah segala informasi yang dapat dihitung menggunakan konteks primer. Konteks sekunder dapat berupa penggabungan beberapa data sensor atau pengambilan data untuk ditampilkan pada aplikasi web. Berikut merupakan contoh dari pengkategorian tipe konteks pada gambar 2.3 dibawah ini.

| | | Categories of Context (Operational Perspective) | |
|--|----------|---|--|
| | | Primary | Secondary |
| Categories of Context (Conceptual Perspective) | Location | Location data from GPS sensor (e.g. longitude and latitude) | Distance of two sensors computed using GPS values Image of a map retrieved from map service provider |
| | Identity | Identify user based on RFID tag | Retrieve friend list from users Facebook profile Identify a face of a person using facial recognition system |
| | Time | Read time from a clock | Calculate the season based on the weather information Predict the time based on the current activity and calendar |
| | Activity | Identify opening door activity from a door sensor | Predict the user activity based on the user calendar Find the user activity based on mobile phone sensors such as GPS, gyroscope, accelerometer |

Gambar 2.3 Contoh kategori dari konteks

Sumber: Context Aware Computing for The Internet of Things: A Survey (Charith Perera, 2014)

Tingkat konteks dapat dibagi menjadi 3 berdasarkan interaksi dari pengguna, yaitu:

- *Personalisation* yang artinya pengguna diperbolehkan memilih pengaturan, kesukaan, dan keluaran yang di inginkan kepada sistem. Contohnya pengguna menginginkan suhu ruangan tetap terjaga di suhu yang telah ditentukan, dimana pendingin ruangan dan pemanas ruangan bekerja sama untuk mencapai hal ini
- *Passive context-awareness* yang berarti sistem memantau lingkungan secara konstan dan memberikan pilihan kepada pengguna, sehingga pengguna dapat mengambil tindakan. Contohnya ketika memasuki toko, telepon pintar akan memberitahukan bahwa ada diskon untuk beberapa produk di dalam toko
- *Active context-awareness* yang berarti sistem memantau kondisi secara terus menerus dan otomatis, dan kemudian mengambil tindakan secara otomatis.

2.2 Dasar Teori

Dasar teori terdiri dari beberapa teori – teori pendukung yang akan digunakan pada penelitian ini. Teori pendukung tersebut dapat dilihat sebagai berikut.

2.2.1 Karakteristik Karbon Monoksida

Karbon monoksida adalah gas yang terbentuk ketika terjadi pembakaran yang tidak selesai, apakah itu di mesin kendaraan atau di dalam peralatan memasak

yang menggunakan batubara atau bahan bakar fosil lainnya, khususnya gas. Karbon monoksida tidak berwarna, tidak berbau, dan tidak terasa (Sac, 2015)

Sel darah merah membawa oksigen yang vital ke jaringan tubuh. Keracunan karbon monoksida terjadi sewaktu gas ini, bukannya oksigen, diserap oleh sel darah merah. Akibatnya, tubuh kekurangan oksigen. Menyerap emisi rendah karbon monoksida selama suatu periode waktu dapat mengakibatkan kerusakan otak permanen. Gejala-gejalanya mencakup sakit kepala, mengantuk, lesu, pusing, mual, serta pingsan dan pada kasus yang parah, gejala yang terjadi yaitu denyut jantung lemah, koma, dan kegagalan pada fungsi pernapasan. Jika seseorang tiba-tiba teracuni karbon monoksida, maka harus segera dilakukan pemberian oksigen dan bantuan pernapasan sebelum terjadi asfiksiasi atau kekurangan oksigen yang mencapai otak. Jika sampai terjadi kekurangan oksigen di otak, maka dapat terjadi kematian pada korban keracunan gas karbon monoksida. Berdasarkan data dari (International Programme on Chemical Safety, 1999), disebutkan batasan tingkat karbon monoksida dalam ruangan yang akan disampaikan pada tabel 2.2 berikut.

Tabel 2.2 Batasan tingkat karbon monoksida dalam ruangan

| Kadar CO di udara | Keterangan |
|-------------------|---|
| 1-4 ppm | Batas normal CO yang dianjurkan |
| 3-7 ppm | Dapat meningkatkan gejala asma sebesar 14% untuk anak-anak |
| 5-6 ppm | Resiko tinggi untuk penurunan angka kelahiran jika terjadi kontak terus menerus selama lebih dari 3 bulan |
| 8-9 ppm | Standar untuk penggunaan peralatan rumah tangga berventilasi penghasil CO, dengan maksimal penggunaan 8 jam sehari di dalam ruangan |
| >10 ppm | Peningkatan angka kematian akibat kegagalan jantung |

Sumber: (International Programme on Chemical Safety, 1999)

2.2.2 Internet of Things

Internet of things pada dasarnya menghubungkan semua perangkat yang memiliki tombol on dan off ke internet. Perangkat yang dimaksud bisa berupa telepon genggam, mesin penggiling kopi, mesin cuci, lampu, jam tangan pintar, dan hampir semua barang yang dapat dibayangkan. Bagian dari sebuah sistem juga dapat dikendalikan, seperti pintu pada rumah, bor pada mesin pengebor minyak. Istilah *Internet of Things* sering disebut sebagai teknologi masa kini, yaitu teknologi yang memanfaatkan perangkat komputer berukuran mini dan dapat terhubung dengan jaringan lokal ataupun internet. Perangkat yang digunakan didesain untuk menggunakan daya yang kecil, sehingga perangkat tersebut hanya bisa menjalankan perintah – perintah sederhana. *Internet of Things* sudah banyak diaplikasikan pada *Smart Home* saat ini. Perangkat ini diatur untuk melakukan tugas – tugas tertentu saja seperti layaknya sistem yang tertanam untuk membaca

data dari sensor. *Internet of Things* juga dapat digunakan sebagai perangkat perantara antara sensor dengan pengguna dan dapat berperan juga untuk mengontrol aktuator. Kelebihan teknologi *Internet of Things* adalah perangkat yang digunakan dapat terkoneksi dengan jaringan lokal maupun internet sehingga data yang dihasilkan dapat dilihat dari jarak jauh. Perusahaan analis Gartner mengatakan bahwa setidaknya akan ada 26 juta perangkat yang terhubung pada tahun 2020. *Internet of Things* adalah jaringan raksasa dari perangkat yang terhubung, termasuk manusia. Hubungannya dapat berupa hubungan antara manusia dengan manusia, manusia dengan mesin, dan mesin dengan mesin (Morgan, 2014).

2.2.3 Pengolahan data sensor MQ-7

Sensor MQ-7 merupakan sensor yang sangat sering digunakan pada beberapa penelitian dikarenakan harganya yang murah dan daya tahannya yang tergolong cukup lama. Namun data yang diberikan dari sensor hanyalah data pembacaan voltase sensor, yang berarti data tersebut harus di olah kembali supaya mudah dimengerti oleh pengguna. Dalam melakukan konversi nilai pembacaan sensor, dibutuhkan spesifikasi lengkap dari sensor tersebut.

Mengacu pada lembar spesifikasi sensor (HANWEI ELECTRONICS CO ., 2018), disebutkan bahwa rasio resistansi sensor MQ-7 (R_s/R_o) akan bernilai 1 pada saat konsentrasi gas CO 100 ppm. Berdasarkan hal itu, maka dapat disimpulkan bahwa saat konsentrasi gas CO 100 ppm, maka nilai $R_s = R_o$. Dari informasi inilah dapat dicari nilai R_o , dikarenakan nilai R_o tidak disebutkan secara resmi pada lembar spesifikasi sensor milik Hanwei. Dari perhitungan yang dilakukan oleh (Vaisala, 2017), didapatkan hasil pengujian menggunakan kalibrator Krisbow KD09-224 pada tabel 2.3 sebagai berikut:

Tabel 2.3 Hasil pengujian sensor dengan kalibrator

| No | PPM | Rata-rata ADC | Vout (Volt) | $R_s(k\Omega)$ | R_s/R_o |
|----|-----|---------------|-------------|----------------|-----------|
| 1 | 20 | 337.2 | 1.65 | 20.3 | 2.95 |
| 2 | 30 | 369.6 | 1.81 | 17.62 | 2.56 |
| 3 | 40 | 416.6 | 2.04 | 14.51 | 2.11 |
| 4 | 50 | 467 | 2.28 | 11.93 | 1.73 |
| 5 | 60 | 494.8 | 2.42 | 10.66 | 1.55 |
| 6 | 70 | 525.4 | 2.57 | 9.46 | 1.37 |
| 7 | 80 | 564.6 | 2.76 | 8.12 | 1.18 |
| 8 | 90 | 577.4 | 2.82 | 7.73 | 1.12 |
| 9 | 100 | 606.6 | 2.96 | 6.89 | 1 |
| 10 | 110 | 619 | 3.03 | 6.5 | 0.94 |
| 11 | 120 | 639.2 | 3.12 | 6.03 | 0.88 |
| 12 | 130 | 663.4 | 3.24 | 5.43 | 0.79 |
| 13 | 140 | 676.2 | 3.3 | 5.15 | 0.75 |
| 14 | 150 | 691.8 | 3.38 | 4.79 | 0.7 |

| | | | | | |
|----|-----|-------|------|------|------|
| 15 | 160 | 699.2 | 3.42 | 4.62 | 0.67 |
| 16 | 170 | 717.8 | 3.51 | 4.25 | 0.62 |
| 17 | 180 | 730 | 3.57 | 4.01 | 0.58 |
| 18 | 190 | 743.6 | 3.63 | 3.77 | 0.55 |
| 19 | 200 | 760.6 | 3.72 | 3.44 | 0.5 |

Sumber: (Vaisala, 2017)

Berikut merupakan rumus yang digunakan Vaisala dalam melakukan perhitungan pada tabel diatas.

Untuk menghitung V_{out} digunakan rumus:

$$V_{out} = (RL/R_s + RL) \times V_{cc} \quad (2.1)$$

Sehingga rumus untuk menghitung R_s sebagai berikut:

$$R_s = (V_{cc} \times RL/V_{out}) - RL \quad (2.2)$$

V_{out} = Tegangan keluaran sensor

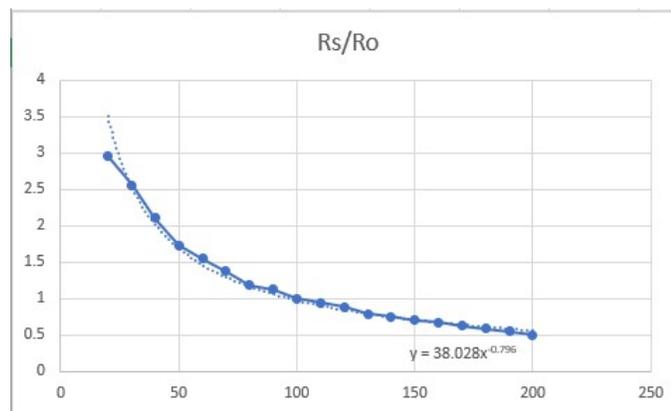
R_s = Resistansi sensor pada konsentrasi gas yang berbeda

R_o = Resistansi sensor saat konsentrasi gas CO 100 ppm

RL = Resistansi beban sensor

Vcc = Tegangan masukan sensor

Dari tabel tersebut, maka akan dicari persamaan garis matematisnya menggunakan bantuan Microsoft Excel. Dengan menggunakan regresi (trendline) power maka diperoleh persamaan yang merupakan hubungan antara ppm CO dengan R_s/R_o . Trendline tersebut dipilih karena melihat bentuk kurva pada grafik, selain itu juga dari sisi komputasi dengan menggunakan trendline tersebut akan lebih mudah diterapkan pada mikrokontroler, seperti yang ditampilkan pada gambar 2.4 berikut ini.



Gambar 2.4 Grafik persamaan garis antara konsentrasi CO dengan R_s/R_o

Dari lembar spesifikasi sensor buatan Hanwei, diketahui bahwa ketika sensor mendeteksi CO sebesar 100 PPM, maka nilai R_s berada diantara 2-20k.

Berdasarkan tabel 2.3 diatas, dapat dilihat bahwa nilai Rs saat konsentrasi CO 100 PPM adalah 6.89kΩ, yang berarti pengujian menggunakan kalibrator Krisbow KD09-224 berhasil. Sehingga untuk mencari nilai PPM dari data *raw* yang dihasilkan oleh sensor, dapat diimplementasikan menggunakan persamaan garis matematis yang sudah dibuat, yaitu:

$$y = 38.028x^{-0.796} \quad (2.3)$$

dimana x adalah nilai dari Rs/Ro yang didapatkan dari sensor.

2.2.4 Evaluasi Performa MQTT dan CoAP

Wireless Sensor Network (WSN) mengalami pembangunan dalam skala besar untuk mendapatkan parameter pada lingkungan seperti suhu, kelembapan, dan kualitas udara di sebuah kota. Pembangunan WSN biasanya terdiri dari node sensor dan *gateway*. Node sensor mengirim data kepada *gateway*, yang kemudian *gateway* mengirimkan data ke jaringan tertentu untuk meneruskan data dari node sensor (Thangavel, Ma, & Tan, 2014).

Untuk mengirim semua data sensor yang diterima oleh *gateway*, dibutuhkan protokol yang memiliki *bandwidth* efisien, energi efisien, dan dapat dijalankan pada perangkat yang memiliki sumber daya kecil. Hasilnya, protokol seperti *Message Queue Telemetry Transport* (MQTT) dan *Constrained Application Protocol* (CoAP) menjadi protokol yang paling sering digunakan untuk memenuhi kebutuhan dari dunia WSN. Berikut akan disediakan pada tabel 2.4 mengenai perbedaan besar antara protokol MQTT dan CoAP.

Tabel 2.4 Perbedaan antara protokol MQTT dan CoAP

| | MQTT | CoAP |
|---------------------------|---|---|
| <i>Layer</i> Aplikasi | Satu <i>layer</i> | Satu <i>layer</i> dengan 2 sub <i>layer</i> (<i>Messages Layer</i> dan <i>Request Response Layer</i>) |
| <i>Layer</i> Transport | Bekerja pada TCP | Bekerja pada UDP |
| Mekanisme Reliabilitas | 3 tingkat <i>Quality of Service</i> (QoS) | <i>Confirmable messages</i> , <i>Non-confirmable messages</i> , <i>Acknowledgements and retransmissions</i> |
| Arsitektur yang di dukung | <i>Publish-Subscribe</i> | <i>Request-Response</i> , <i>Resource observe/Publish-Subscribe</i> |

Sumber: Performance Evaluation of MQTT and CoAP via a Common Middleware (Thangavel, Ma, & Tan, 2014)

Salah satu cara WSN dalam mengirimkan data dari *gateway* ke *client* yaitu dengan menggunakan arsitektur *publish-subscribe*. Pada arsitektur ini, *client* yang ingin mengetahui data harus mendaftarkan ketertarikannya akan suatu topik data ke *gateway* (*broker*). *Client* yang menghasilkan data akan mengirimkan datanya ke *broker*, dan kemudian *broker* meneruskan data baru ke *client* yang menginginkan data tersebut. Salah satu keuntungan dari arsitektur ini adalah mengurangi jarak antara *client* yang membutuhkan data dengan *client* yang mengirimkan data. Sebagai contohnya, node sensor yang mengirimkan data tidak perlu mengetahui identitas *client* yang tertarik pada data sensor tersebut dan sebaliknya, *client* yang membutuhkan data tersebut tidak perlu mengetahui identitas node sensor yang mengirimkan data sensor. Pengurangan jarak seperti ini memungkinkan arsitektur *publish-subscribe* dapat digunakan pada skala yang besar. Arsitektur *publish-subscribe* ini didukung oleh protokol *Machine to Machine* (M2M) seperti MQTT dan CoAP.

2.2.5 Quality of Service

Quality of Service (QoS) merupakan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan yang diharapkan. Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah *throughput*, *delay* dan *jitter*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Sebagai contoh, komunikasi suara (seperti VoIP atau IP *Telephony*) serta video *streaming* dapat membuat pengguna merasa kebingungan ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan *throughput* yang terlalu kecil, dengan *delay* yang tidak dapat diprediksi, atau *jitter* yang berlebihan. Fitur *Quality of Service* (QoS) ini dapat menjadikan *throughput*, *delay*, dan *jitter* dapat diprediksi dan dicocokkan dengan kebutuhan aplikasi yang digunakan di dalam jaringan yang ada (Mahdi Miraz, 2017).

QoS didesain untuk membantu pengguna (*client*) menjadi lebih produktif dengan memastikan bahwa pengguna mendapatkan performansi yang handal dari aplikasi-aplikasi berbasis jaringan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan. Tujuan dari QoS adalah untuk memenuhi kebutuhan-kebutuhan layanan yang berbeda, yang menggunakan infrastruktur yang sama. QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan yang disediakan, baik secara kualitatif maupun kuantitatif. Banyak hal dapat terjadi pada paket saat paket – paket itu dikirimkan dari asal ke tujuannya. Hal itu dapat dihitung menggunakan beberapa parameter pengukuran seperti yang akan disebutkan berikut ini.

Throughput adalah jumlah data yang dikirim atau diterima oleh suatu jaringan atau jumlah data yang diproses dalam satuan waktu tertentu. *Throughput* memiliki satuan khusus yaitu bit per second (bit/s atau bps). *Throughput* dapat menjadi lebih rendah daripada *input tax* bergantung pada

kondisi jaringan (Carvalho & Magedanz, 2009). Rata-rata *Throughput* dapat dihitung dengan rumus pada persamaan 2.4.

$$\textit{Throughput} = \frac{\textit{Ukuran file (bps)}}{\textit{waktu (s)}} = \frac{\textit{Transfer (bps)}}{\textit{interfal (s)}} \quad (2.4)$$

Sumber: (Carvalho & Magedanz, 2009)

Delay adalah parameter yang intrinsic pada suatu jaringan, mengingat end-point dari suatu jaringan pasti berjarak dan informasi akan membutuhkan waktu untuk sampai ke ujung lain dari sebuah jaringan. *Delay* juga biasa disebut latency. Waktu *delay* dapat meningkat jika mengalami antrian yang panjang dalam jaringan (congestion), atau harus melewati rute yang lebih jauh untuk menghindari congestion. *Delay* dapat dihitung secara langsung (waktu total dari saat paket dikirim sampai paket diterima) atau round-trip (waktu yang dibutuhkan dari asal paket sampai ke tujuan lalu kembali ke asal lagi) (Carvalho & Magedanz, 2009).

Jitter adalah bentuk variasi dari *delay* yang erat kaitannya dengan kedatangan paket. *Jitter* disebabkan oleh panjang queue dalam satu pengolahan data dan reassemble data di akhir pengiriman. Semakin besar beban trafik akibat variasi *delay* akan membuat peluang terjadinya tumbukan antar paket juga tinggi sehingga nilai *jitter* akan makin besar dan nilai QoS menurun (Fabricio, 2009). Nilai *jitter* dapat dihitung dengan persamaan 2.5 berikut:

$$\textit{Jitter} = \frac{\textit{Total varian delay}}{\textit{total paket yang diterima}-1} \quad (2.5)$$

Sumber: (ETSI TR, 2002)