

## **BAB 2**

### **DASAR TEORI**

#### **2.1 TINJAUAN PUSTAKA**

Pada penelitian oleh Hanum Shirotu Nida, Mahar Faiqurahman, Zamah Sari pada tahun 2017 dengan judul “Prototype Sistem Multi-Telemetry Wireless Untuk Mengukur Suhu Udara Berbasis Mikrokontroler ESP8266 Pada Greenhouse” penelitian ini membahas tentang prototype sistem multi-telemetry wireless untuk mengukur suhu udara pada greenhouse dengan menggunakan sensor suhu DHT11 dan modul WiFi ESP8266 sebagai sensor node yang diimplementasikan pada greenhouse. Sensor suhu akan dipasang di dalam beberapa greenhouse dan data yang dibaca oleh sensor kemudian dikirimkan ke server dengan menggunakan modul WiFi ESP8266 yang tersambung dengan sensor. Server menerima data yang dikirimkan oleh modul WiFi dan menyimpannya ke dalam database kemudian menampilkan data dalam bentuk website. Dalam penelitian ini diuji nilai sensor DHT11, heap memory ESP8266, jarak atau jangkauan ESP8266, uji coba data missing handling dan kestabilan jaringan. Berdasarkan hasil pengujian diketahui bahwa sensor DHT11 memiliki rata-rata kesalahan ukur suhu  $0.92^{\circ}\text{C}$  dan kelembapan 3.1%. Modul WiFi ESP8266 mampu menyimpan dan mengirim buffer hingga 100 data dan dapat melakukan pengiriman dalam jangkauan 50 meter. Data *missing* handling memanfaatkan *buffer* untuk menyimpan data selama server sedang tidak dapat diakses oleh sensor *node* agar data tidak hilang. Kestabilan pengiriman data atau koneksi sensor *node* dengan server dipengaruhi oleh jumlah *access point* yang sedang berkomunikasi di sekitar *access point* server dengan menggunakan *channel* yang sama. Penulis mengambil pembahasan mengenai suhu dan kelembapan yang digunakan pada penelitian ini [8].

Selanjutnya penelitian oleh Hardian Setya Dharma Putra, Resmana Lim, Iwan Handoyo Putro pada tahun 2019 dengan judul “Pemantauan Kualitas Udara Polutan Gas CO dan CO<sub>2</sub> Berbasis IoT”. Penelitian ini membahas tentang alat pemantau kualitas udara polutan CO dan CO<sub>2</sub> berbasis IoT (*Internet of Things*). Dalam pembuatan alat pemantau polutan udara ini terdapat perangkat yang

digunakan yaitu sensor MQ-7 untuk mendeteksi gas CO dan sensor MH-Z19 untuk memantau gas CO<sub>2</sub>. Hasil dalam pembuatan sistem pemantau polutan udara berbasis IoT adalah mudahnya pemantauan gas polutan udara khususnya gas CO dan CO<sub>2</sub>. Dalam pembacaan besaran gas CO dan CO<sub>2</sub> alat pemantau polutan udara berbasis IoT memiliki tingkat kesalahan pembacaan gas CO sebesar 0% dan gas CO<sub>2</sub> sebesar 97%-98%. Besar *error* pembacaan sensor gas CO<sub>2</sub> dikarenakan sensor untuk kalibrasi tidak memiliki spesifikasi yang sama dengan sensor yang digunakan. Pemantauan polutan udara terdiri dalam tiga tampilan yaitu tampilan berupa *website*, aplikasi *Blynk* dan perangkat LCD 20x4. Penulis mengambil pembahasan mengenai karbondioksida yang digunakan pada penelitian ini. [9].

Selanjutnya penelitian yang dilakukan oleh Arida Amalia Rosa<sup>1</sup>, Bryan Alexis Simon dan Kevin Sherdy Lieanto tahun 2020 dengan judul “Sistem Pendeteksi Pencemaran Udara Portabel Menggunakan Sensor MQ-7 dan MQ-135” membahas tentang sistem pendeteksi pencemaran udara. Dalam penelitian ini untuk menunjang program Kementrian Lingkungan Hidup tentang meningkatkan kesadaran masyarakat akan pentingnya memperhatikan ISPU (Indeks Standar Pencemaran Udara). Dalam penelitian ini menggunakan sensor MQ-7 sebagai pendeteksi gas karbon monoksida (CO), sensor MQ -135 sebagai pendeteksi gas karbon dioksida (CO<sub>2</sub>) dan Arduino uno sebagai platform dari *physical computing* yang bersifat *open source*. Hasil dari penelitian ini adalah penggunaan sensor MQ-7 dan MQ-135 sudah cukup ideal untuk melakukan pendeteksian gas- gas berbahaya pada batasan tertentu melalui bacaan digital. Namun untuk melakukan pengukuran dengan akurasi tinggi, sensor harus melewati proses heating dan konfigurasi tersendiri. Penulis mengambil pembahasan mengenai sensor MQ-135 yang digunakan pada penelitian ini. [10].

Selanjutnya penelitian oleh Fitri Puspasari, Trias Prima Satya, Unan Yusmaniar Oktawati, Imam Fahrurrozi, dan Hristina Prisyanti pada tahun 2020 dengan judul “Analisis Akurasi Sistem Sensor DHT22 berbasis *Arduino* terhadap *Thermohyrometer* Standar” penelitian ini membahas tentang kinerja sensor dan membandingkan hasil pengukuran antara sensor DHT22 dan *thermohyrometer*. Dalam analisis akurasi sistem ini menggunakan sensor DHT22 untuk mengukur suhu dan kelembaban sedangkan untuk pembandingnya menggunakan

*thermohygrometer*. Hasil dalam analisis akurasi sistem sensor DHT22 berbasis *arduino* terhadap *thermohygrometer* standar ini adalah perbandingan hasil nilai kesalahan rata-rata pada pengukuran suhu dan kelembaban antara sensor DHT22 dengan *Thermohygrometer* standar menghasilkan nilai 2,99% untuk kelembaban dan -2,31% untuk suhu. Berdasarkan hasil tersebut dapat disimpulkan akurasi dikatakan baik dan dapat diterima karena sesuai dengan data sheet sensor DHT22 yaitu kelembaban yang terukur harus memiliki range antara 2-5% dan  $\pm 5^{\circ}\text{C}$  untuk nilai suhu. Penulis mengambil pembahasan mengenai sensor DHT22 yang digunakan pada penelitian ini.[11]

Selanjutnya penelitian oleh Arif Indra Irawan, Raditiana Patmasari, Muhammad Rahmat Hidayat pada tahun 2020 dengan judul “Peningkatan Kinerja Sensor DS18B20 pada Sistem IoT Monitoring Suhu Kolam Ikan” penelitian ini membahas tentang peningkatan kinerja sensor DS18B20 dalam monitoring suhu kolam ikan. Penelitian ini bertujuan untuk meningkatkan kinerja sensor suhu DS18B20 yang digunakan pada sistem *Internet of Things* (IoT) untuk monitoring suhu kolam ikan. Parameter yang digunakan untuk menguji kualitas sistem sensor suhu tersebut adalah akurasi, *Root Mean Square Error* (RMSE), dan *response time*. Analisis dilakukan menggunakan metode regresi linier dan *measurement rate* berdistribusi normal. Hasil penelitian menunjukkan bahwa setelah digunakannya metode regresi linier saat kalibrasi, akurasi pengukuran, RMSE, dan *response time* mengalami peningkatan. Penulis mengambil pembahasan mengenai peningkatan kinerja sensor dengan menggunakan metode regresi linier yang digunakan pada penelitian ini. [12].

Selanjutnya penelitian oleh Nadjum Middinali dan Yusnita Rahayu pada tahun 2019 dengan judul “Pembangunan Sistem Monitoring Data Kualitas Udara Berbasis IoT Di Universitas Riau” membahas tentang sistem monitoring kualitas udara yang berbasis IoT di Universitas Riau. dalam penelitian ini membuat aplikasi pemantauan perangkat kualitas udara secara *real-time* menggunakan *Internet of Things*. Pada penelitian menggunakan perangkat *hardware* yaitu *waspmote plug and sense* untuk pemantauan udara kandungan PM10 (Partikel Asap) dan *Waspmote OEM* untuk pemantauan udara kandungan gas CO (Karbonmonoksida) dan O3 (Ozon), dimana *transfer data* menggunakan *Zigbee*.

Aplikasi *Microsoft Visual Basic 2010* digunakan sebagai pemrosesan dan koneksi ke database MySQL Aplikasi akan memproses data *waspmote* menjadi data ISPU untuk kondisi referensi, konten udara PM10, CO dan O3, akan menampilkan ISPU di situs Web. Penelitian ini juga membandingkan hasil pemantauan sensor dengan AQMS di BLH dan BMKG Pekanbaru. Media pendukung untuk menjalankan dan mengolah data pemantauan dilakukan dengan aplikasi yang telah dibuat di *Microsoft Visual Basic 2010*, aplikasi akan mengolah data dalam bentuk angka yang dikirimkan ke *database MySQL* untuk diupload ke jaringan internet sesuai dengan ISPU (Indeks Standar Pencemaran Udara) untuk selanjutnya ditampilkan di *website*. Penulis mengambil pembahasan mengenai *internet of things* yang digunakan pada penelitian ini[13].

Selanjutnya penelitian oleh Jacqueline Waworundeng, Oktoverano Lengkong pada tahun 2018 dengan judul “Sistem Monitoring dan Notifikasi Kualitas Udara dalam Ruangan dengan Platform IoT” penelitian ini membahas tentang *prototype* alat pendeteksi kualitas udara di dalam ruangan dengan menggunakan mikrokontroler *Wemos* dan sensor MQ135 yang terhubung dengan platform IoT sebagai sistem monitoring dan notifikasi. Sistem ini berpotensi untuk digunakan sebagai sistem pemantauan kualitas udara di dalam ruangan untuk meningkatkan kesadaran tentang pentingnya kualitas udara yang sehat. Dalam penelitian ini alat yang digunakan yaitu modul sensor MQ135 sebagai detektor kualitas udara, mengirimkan sinyal input untuk diproses oleh mikrokontroler *Wemos board*.

Modul wifi yang terdapat pada *Wemos board* mengirimkan nilai yang terbaca oleh sensor ke platform IoT *Thingspeak* yang merekam data *logging* dalam bentuk grafik. Dalam hal ini, *Thingspeak* berfungsi sebagai bagian dari sistem monitoring. Sedangkan sebagai sistem notifikasi digunakan platform IoT *Blynk apps*. *Blynk apps* terhubung secara tidak langsung ke *prototype* alat pendeteksi kualitas udara melalui internet. Nilai yang terbaca dari sensor diproses sesuai program dan jika memenuhi level sensor yang ditentukan maka sistem memberikan notifikasi kepada user melalui *Blynk apps*. Penelitian ini mengacu pada model *prototype*. Berdasarkan skenario pengujian yang dilakukan dengan menempatkan alat detektor kualitas udara di dalam sebuah ruangan selama 20

menit, didapati bahwa rata-rata level kualitas udara dalam ruangan tersebut menunjukkan nilai analog 489,3 dimana ruangan tersebut terindikasi memiliki polusi udara tingkat menengah. Penulis mengambil pembahasan mengenai *internet of things* yang digunakan pada penelitian ini. [14].

## **2.2 DASAR TEORI**

Pada dasar teori ini terdapat beberapa pembahasan mengenai Pengenalan kualitas udara, Metode Regresi Linier, Pengenalan konsep *Internet Of Things* (IoT), Nodemcu ESP8622, Sensor DHT22, Sensor MQ-135, LCD I2C dengan ukuran 16x2 cm , Buzzer, LED, *Google FIREBASE*, MIT APP Inventor, *Quality Of Service* (QoS).

### **2.2.1 Kualitas Udara**

Kualitas udara dalam ruang sangat penting untuk karyawan dan setiap orang yang berada di dalam bangunan atau gedung. Meskipun ada upaya untuk menjaga kualitas udara dalam ruang agar tetap baik, keluhan dari penghuni mulai dari bau, sakit kepala, iritasi mata atau tenggorokan, alergi reaksi dan gejala lain kadang-kadang terjadi [5]. Ada beberapa faktor yang mempengaruhi kualitas udara yaitu :

#### **A. Suhu**

Suhu dalam ruang rumah yang terlalu rendah dapat menyebabkan gangguan kesehatan hingga *hypotermia*, sedangkan suhu yang terlalu tinggi dapat menyebabkan dehidrasi sampai dengan *heat stroke*. Perubahan suhu udara dalam rumah dipengaruhi oleh beberapa faktor antara lain penggunaan bahan bakar biomassa, ventilasi yang tidak memenuhi syarat, kepadatan hunian, bahan dan struktur bangunan, kondisi geografis dan kondisi topografi. Upaya Penyehatan yang bisa dilakukan adalah jika suhu udara di atas 30°C diturunkan dengan cara meningkatkan sirkulasi udara dengan menambahkan ventilasi mekanik/buatan dan jika suhu kurang dari 18°C, maka perlu menggunakan pemanas ruangan dengan menggunakan sumber energi yang aman bagi lingkungan dan kesehatan [3].

#### **B. Kelembapan**

Kelembapan yang terlalu tinggi maupun rendah dapat menyebabkan suburnya pertumbuhan mikroorganisme. Faktor yang menyebabkan kelembapan

tinggi maupun rendah adalah konstruksi rumah yang tidak baik seperti atap yang bocor, lantai, dan dinding rumah yang tidak kedap air, serta kurangnya pencahayaan baik buatan maupun alami. Ada beberapa cara atau upaya penyehatan yang dilakukan jika kelembapan udara kurang dari 40%, maka dapat dilakukan upaya penyehatan seperti menggunakan alat *humidifier* (alat pengatur kelembapan udara) untuk meningkatkan kelembapan, membuka jendela rumah, menambah jumlah dan luas jendela rumah serta memodifikasi fisik bangunan (meningkatkan pencahayaan, sirkulasi udara). Jika kelembapan udara lebih dari 60%, maka dapat dilakukan upaya penyehatan seperti memasang genteng kaca atau menggunakan alat *humidifier* (alat pengatur kelembapan udara) untuk menurunkan kelembapan [3].

### C. Karbon Dioksida

Dampak yang ditimbulkan pada konsentrasi di atas nilai ambang batas yang disyaratkan yaitu 1.000 ppm, dapat menyebabkan mengantuk, sakit kepala, dan dapat menurunkan aktivitas fisik. Pada konsentrasi 3% (30.000 ppm), bersifat narkotik ringan dan menyebabkan peningkatan tekanan darah serta gangguan pendengaran. Pada konsentrasi 5% (50.000 ppm), menyebabkan stimulasi pernapasan, pusing-pusing, dan kesulitan pernapasan yang diikuti oleh sakit kepala. Pada konsentrasi >8% (80.000 ppm,) dapat menyebabkan sakit kepala, berkeringat terus menerus, tremor, dan kehilangan kesadaran setelah paparan selama 5-10 menit. Faktor yang menyebabkan tinggi kadar CO<sub>2</sub> adalah penggunaan bahan bakar seperti arang, kayu, minyak bumi, dan batu bara, merokok di dalam rumah serta kepadatan penghuni yang tinggi di dalam ruang. Upaya penyehatan yang dapat dilakukan dalam mengendalikan kadar CO<sub>2</sub> antara lain menggunakan ventilasi alami atau mekanik dalam rumah agar terjadi pertukaran udara, menggunakan bahan bakar rumah tangga yang ramah lingkungan seperti LPG dan listrik, tidak merokok di dalam rumah serta tidak mehidupkan mesin kendaraan bermotor dalam ruangan tertutup, serta menanam tanaman di sekeliling rumah [3].

### 2.2.2 Regresi Linear Sederhana

Regresi Linear Sederhana merupakan metode statistik yang memiliki fungsi menguji sejauh mana hubungan sebab akibat antara Variabel Faktor

Penyebab (X) terhadap Variabel Akibat (Y). Faktor penyebab dilambangkan dengan X atau disebut *Predictor* sedangkan Variabel Akibat dilambangkan dengan Y atau disebut *Response*. Regresi Linear Sederhana atau SLR (*Simple Linear Regression*) merupakan metode statistik yang dipergunakan dalam produksi untuk melakukan peramalan atau prediksi tentang karakteristik kualitas maupun kuantitas [15].

Model Persamaan 2.1 menunjukkan rumus Regresi Linear Sederhana.

$$Y = a + bX \quad (2.1)$$

Keterangan:

Y = Variabel Akibat (Dependen)

X = Variabel Faktor ( Independen)

a = Konstanta

b = Koefisien Regresi (Kemiringan), besaran Akibat yang ditimbulkan oleh Faktor.

Persamaan 2.2 dan 2.3 menunjukkan rumus untuk mencari nilai a dan b di persamaan 2.1.

$$a = \frac{(\sum y) (\sum x^2) - (\sum x) (\sum xy)}{n(\sum x^2) - (\sum x)^2} \quad (2.2)$$

$$b = \frac{n(\sum xy) - (\sum x) (\sum y)}{n(\sum x^2) - (\sum x)^2} \quad (2.3)$$

### 2.2.3 *Internet Of Things (IoT)*

IoT merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan *global* melalui sensor yang tertanam dan selalu aktif. IoT dapat didefinisikan kemampuan berbagai *device* yang bisa saling terhubung dan saling bertukar data melalui jaringan internet. IoT merupakan sebuah teknologi yang memungkinkan adanya sebuah pengendalian, komunikasi, kerjasama dengan berbagai perangkat keras, data melalui jaringan internet.

Sehingga bisa dikatakan bahwa IoT adalah ketika kita menyambungkan sesuatu (*things*) yang tidak dioperasikan oleh manusia, ke internet. Namun IoT

bukan hanya terkait dengan pengendalian perangkat melalui jarak jauh, tapi juga bagaimana berbagi data, memvirtualisasikan segala hal nyata ke dalam bentuk internet, dan lain-lain. Internet menjadi sebuah penghubung antara sesama mesin secara otomatis. Selain itu juga adanya user yang bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. Manfaatnya menggunakan teknologi IoT yaitu pekerjaan yang dilakukan oleh manusia menjadi lebih cepat, mudah dan efisien [16].

#### 2.2.4 NodeMCU ESP8622



Gambar 2.1 NodeMCU ESP8266[17]

Pada Gambar 2.1 menunjukkan gambar NodeMCU 8266 yang merupakan *firmware open-source* dan kit pengembangan yang memainkan peran penting dalam merancang produk IoT. Beberapa pin GPIO di papan memungkinkan untuk menghubungkan papan dengan *periferal* lain dan mampu menghasilkan PWM, I2C, SPI, dan komunikasi serial UART. ESP8266 adalah pengontrol mikro yang dirancang oleh sistem *espressif*, esp8266 sendiri adalah solusi jaringan *wifi* mandiri yang menawarkan sebagai jembatan dari pengontrol mikro yang ada ke *wifi* dan juga mampu menjalankan aplikasi mandiri. Modul ini dilengkapi dengan konektor USB bawaan dan beragam *pin-out*. Dengan kabel *micro* USB dapat menghubungkan *devkit* nodemcu ke laptop dan memflashnya tanpa masalah, seperti *arduino*. Cara paling dasar untuk menggunakan modul ESP8266 adalah dengan menggunakan perintah serial, karena *chip* pada dasarnya adalah *wifi/serial* pemancar[17]. Tabel 2.1 membahas tentang spesifikasi NodeMCU 8266.



Tabel 2.1 Spesifikasi NodeMCU 8266[18].

Spesifikasi	Keterangan
<i>Voltage</i>	3.3V
<i>Current Consumption</i>	10 $\mu$ a – 170mA
<i>Flash memory attachable</i>	16MB <i>Max</i> (512K Normal)
<b>GPIO</b>	17( <i>multiplexed with other functions</i> )
<i>Processor Speed</i>	80-160Mhz
<b>RAM</b>	32K+80K
<i>Flash Memory</i>	4 Mb
<i>WiFi</i>	IEEE 802.11 b/g/n
<b>ADC</b>	1 <i>input with 1024 step resolution</i>
<b>USB Port</b>	<i>Micro USB</i>

### 2.2.5 Sensor DHT22



Gambar 2.2 Sensor DHT22 [19]

Pada Gambar 2.2 menunjukkan gambar DHT22 yang merupakan sensor pengukur suhu dan kelembapan berbiaya rendah. Terdiri dari termister untuk pengukuran suhu dan pengukuran kelembapan sensor kelembapan kapasitif. Fitur yang lebih penting dari sensor lainnya adalah Anda bisa mendapatkan data setelah dua detik. Sangat mudah digunakan, cukup sambungkan *pin* pertama dari kiri dengan *supply* kemudian dapatkan data dari *pin* kedua. *Pin* terakhirnya digunakan untuk *ground*. Ini memiliki akurasi dan presisi yang lebih tinggi, karena fitur ini dapat menggantikan sensor suhu dan kelembapan SHT10 impor yang mahal.

Sensor ini digunakan untuk mengukur suhu dan kelembapan lingkungan untuk memenuhi permintaan yang tinggi. Sensor ini memiliki keandalan dan stabilitas yang baik. Sangat mudah untuk menggabungkan dengan papan ekspansi *Arduino* sensor khusus, yang mudah digunakan untuk persepsi suhu dan kelembapan. DHT22 adalah sensor kelembapan digital, dirancang untuk antarmuka sensor analog. Terdiri dari resistor 4,7 K hingga 10 K, yang dapat digunakan sebagai *pull up* dari *pin* data ke *Vcc* [20]. Tabel 2.2 membahas tentang spesifikasi sensor DHT22.

**Tabel 2.2 Spesifikasi DHT22 [19].**

<b>Spesifikasi</b>	<b>Keterangan</b>
<i>Model</i>	DHT22
<i>Power supply</i>	3.3-6V DC
<i>Output signal</i>	<i>digital signal via single-bus</i>
<i>Sensing element</i>	<i>Polymer capacitor</i>
<i>Operating range</i>	<i>humidity 0-100%RH; temperature -40~80Celsius</i>
<i>Accuracy</i>	<i>humidity +-2%RH(Max +-5%RH); temperature &lt;+-0.5Celsius</i>
<i>Resolution or sensitivity</i>	<i>humidity 0.1%RH; temperature 0.1Celsius</i>
<i>Repeatability</i>	<i>humidity +-1%RH; temperature +-0.2Celsius</i>
<i>Humidity hysteresis</i>	<i>+0.3%RH</i>
<i>Long-term Stability</i>	<i>+0.5%RH/year</i>
<i>Sensing period</i>	<i>Average: 2s</i>
<i>Interchangeability</i>	<i>fully interchangeable</i>
<i>Dimensions</i>	<i>small size 14*18*5.5mm; big size 22*28*5mm</i>

### 2.2.6 Sensor CO2 MQ-135

Pada Gambar 2.3 menunjukkan sensor MQ-135 yang merupakan jenis sensor kimia yang sensitif terhadap senyawa NH<sub>3</sub>, NO<sub>x</sub>, alkohol, benzol, asap (CO), CO<sub>2</sub>, dan lain-lain. Sensor ini bekerja dengan cara menerima perubahan nilai resistansi (analog) bila terkena gas. Sensor ini memiliki daya tahan yang baik untuk penggunaan penanda bahaya polusi karena praktis dan tidak memakan daya yang besar. Penyesuaian sensitifitas sensor ditentukan oleh nilai resistansi dari

MQ-135 yang berbeda-beda untuk berbagai konsentrasi gas-gas. Jadi, Ketika menggunakan komponen ini, penyesuaian sensitifitas sangat diperlukan. Selain itu, kalibrasi pendeteksiian konsentrasi NH<sub>3</sub> sebesar 100 ppm atau alkohol sebesar 50 ppm di udara [10]



**Gambar 2.3** Sensor MQ-135 [10].

Dalam penelitian ini hanya mengambil nilai karbondioksida atau CO<sub>2</sub> dari sensor MQ-135,berikut Tabel 2.3 membahas tentang spesifikasi sensor MQ-135.

**Tabel 2.3** Spesifikasi MQ-135 [21].

Parameter	Kondisi Teknis	Keterangan
<i>Circuit Voltage</i>	5V ± 0,1	AC atau DC
<i>Heating Voltage</i>	5V ± 0,1	AC atau DC
<i>Load Resistance</i>	Bisa menyesuaikan	
<i>Heater Resistance</i>	33 Ω ± 5%	Suhu ruangan
<i>Heating Consumption</i>	< 800 mW	

Hasil keluaran sensor MQ 135 masih berupa *value* sensor belum menunjukkan nilai PPM (*Part Per Million*) dari suatu gas, maka data diolah untuk diubah menjadi informasi PPM dengan rumus sebagai berikut :

$$PPM = a \left( \frac{R_S}{R_O} \right)^b \quad (2.4)$$

Keterangan :

- a)  $a = \text{Scaling Factor}$
- b)  $R_s = \text{Resistance of sensor}$
- c)  $R_o = \text{Resistance Default of sensor}$
- d)  $b = \text{Exponent}$

Dari rumus di atas nilai variable yang menentukan nilai ppm adalah nilai  $R_s$ , di mana :

$$R_S = R_L \frac{(1023 - V_{RL})}{V_{RL}} \quad (2.5)$$

- a)  $R_L = \text{Adjustment Resistor}$
- b)  $V_{RL} = \text{Value analog input from sensor}$

### 2.2.7 LCD I2C 16x2



**Gambar 2.4 LCD I2C 16x2 [22].**

Pada Gambar 2.4 menunjukkan gambar LCD I2C yang merupakan modul LCD yang dikendalikan secara serial sinkron dengan protokol I2C/IIC (*Inter Integrated Circuit*) atau TWI (*Two Wire Interface*). Normalnya, modul LCD dikendalikan secara *parallel* baik untuk jalur data maupun kontrolnya. Namun, jalur *parallel* akan memakan banyak *pin* di sisi controller (misal *Arduino*, *Android*, komputer, dll). Setidaknya Anda akan membutuhkan 6 atau 7 pin untuk mengendalikan sebuah modul LCD. Dengan demikian untuk sebuah controller

yang ‘sibuk’ dan harus mengendalikan banyak I/O, menggunakan jalur *parallel* adalah solusi yang kurang tepat [23], Tabel 2.4 membahas tentang spesifikasi LCD I2C.

**Tabel 2.4 Spesifikasi LCD I2C [22].**

<i>Brief Data:</i>	<i>Description</i>
<i>Compatible with:</i>	<i>Arduino Board or other controller board with I2C bus.</i>
<i>Display Type:</i>	<i>Negative white on Blue backlight.</i>
<i>I2C Address:</i>	<i>0x38-0x3F (0x3F default)</i>
<i>Supply voltage:</i>	<i>5V</i>
<i>Interface:</i>	<i>I2C to 4bits LCD data and control lines.</i>
<i>Contrast Adjustment:</i>	<i>Built-in Potentiometer.</i>
<i>Backlight Control:</i>	<i>Firmware or jumper wire.</i>
<i>Board Size:</i>	<i>80x36 mm.</i>

## 2.2.8 Buzzer



**Gambar 2.5 Buzzer[24].**

Pada Gambar 2.5 menunjukkan gambar *Buzzer* Listrik yang merupakan komponen elektronika yang dapat mengubah sinyal listrik menjadi sebuah getaran suara *Buzzer* pada umumnya sebuah perangkat audio yang sering digunakan pada rangkaian anti-maling, alarm jam tangan, bel rumah, peringatan mundur pada truk, dan perangkat-perangkat bahaya lainnya. Jenis *Buzzer* yang sering ditemukan dan digunakan yakni *Buzzer* yang berisi *Piezoelectric*, karena *Buzzer Piezoelectric* mempunyai kelebihan seperti harganya yang lebih murah, ringan, dan mudah dalam menggabungkannya ke rangkaian elektronika lainnya. *Buzzer* termasuk dalam keluarga *Tranduser* yang disebut juga *Beeper* [24].

### 2.2.9 *Light Emitting Diode (LED)*



**Gambar 2.6** *Light Emitting Diode (LED)*[25].

Pada gambar 2.6 menunjukkan gambar *Light Emitting Diode* atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada *Remote Control* TV ataupun *Remote Control* perangkat elektronik lainnya.

Bentuk LED mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan Lampu Pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya. Oleh karena itu, saat ini LED (*Light Emitting Diode*) yang bentuknya kecil telah banyak digunakan sebagai lampu penerang dalam LCD TV yang mengganti lampu *tube* [25].

### 2.2.10 *Firebase*

*Firebase* merupakan layanan dari *Google* yang digunakan untuk mempermudah para developer perangkat lunak untuk mengembangkan aplikasinya. Berfungsi sebagai BaaS (*Backend as a Service*) merupakan solusi yang ditawarkan oleh *Google* agar developer dapat fokus mengembangkan keseluruhan aplikasi. Setiap layanan pastinya memiliki keunggulan dan

kekurangan. Menurut situs muhaaz.com *Firestore* memiliki beberapa kelebihan dan kekurangan, antara lain :

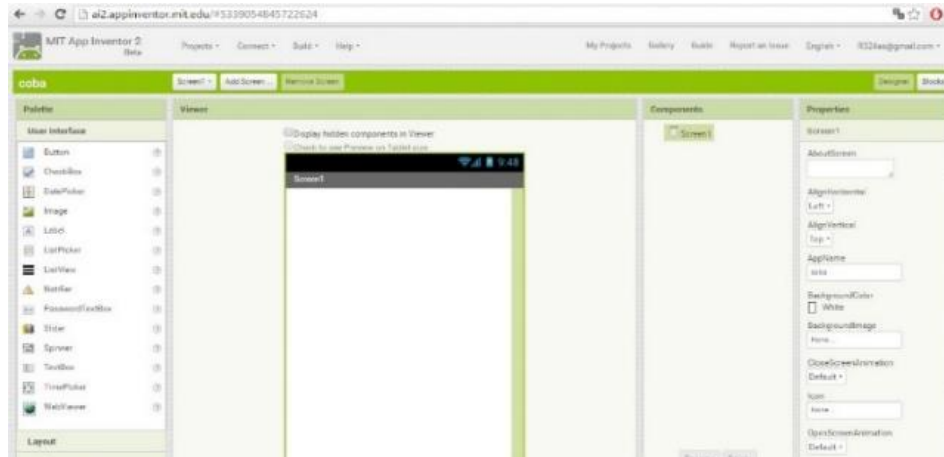
- A. Keunggulan *Firestore* diantaranya adalah :
- 1) Adanya versi gratis ( *free* ). Jadi para developer bisa mencoba menggunakan *Firestore* untuk waktu yang tidak terbatas.
  - 2) *Firestore* cepat dan juga *responsive*.
  - 3) *Firestore* ternyata sudah menggunakan JSON jadi kita tidak lagi menggunakan SQL.
  - 4) *Firestore* bisa digunakan untuk Android, iOS, Java, Objective-C, Swift, Node.js, dan JavaScript.
  - 5) *Firestore* *user friendly* sehingga mudah untuk digunakan.
  - 6) *Firestore* memiliki dokumentasi yang lengkap juga *customer support* yang sigap menjawab permasalahan dari para developer.
- B. Kekurangan *Firestore* diantaranya adalah :
- 1) Versi gratis dari *Firestore* hanya memberikan 100 koneksi dan *storage* sebesar 1 GB, jadi jika ingin bertambah kita harus mengupgradenya.
  - 2) Aplikasi yang menggunakan *database* secara terpusat yang dapat di *update* oleh banyak *user*, biasanya akan terjadi *overkill*.
  - 3) Data yang ada pada *Firestore* bukanlah *host* milik kita, sehingga untuk *recovery* akun *user* akan menjadi mustahil[26].

### 2.2.11 MIT APP Inventor

MIT *App Inventor* merupakan aplikasi inovatif yang dikembangkan oleh *Google* dan MIT dengan tujuan mengenalkan dan mengembangkan pemrograman daripada sebuah android dengan mentransformasikan bahasa pemrograman yang kompleks berbasis teks menjadi berbasis visual (*drag and drop*) berbentuk blok-blok.

Pada Gambar 2.7 menunjukkan tampilan MIT *App Inventor*, *App Inventor* adalah aplikasi web sumber terbuka yang dikembangkan oleh *Google* dimana saat ini dikelola oleh *Massachusetts Institute of Technology* *App Inventor* memungkinkan pengguna baru dapat memprogram suatu komputer agar dapat membuat aplikasi suatu perangkat lunak untuk sistem operasi pada platform

android. App Inventor juga menggunakan tampilan antarmuka grafis yang serupa dengan antarmuka pengguna pada *Scratch* maupun Star Logo TNG untuk *drag-and-drop* suatu objek visual dalam menciptakan aplikasi yang dapat beroperasi pada perangkat Android [27].



**Gambar 2.7** Tampilan MIT *App Inventor*[27]

### 2.2.12 *Wireshark*

Pada skripsi ini *software Wireshark* digunakan untuk mendapatkan hasil dari kualitas *Quality Of Service* (QoS) saat menjalankan pengiriman data menggunakan internet kepada sebuah *website* ataupun aplikasi di android. *Wireshark* adalah salah satu dari alat analisa jaringan yang biasa dipakai oleh seorang *Network Administrator* untuk melakukan pemecahan masalah yang ada dalam jaringan, menganalisis, perangkat lunak atau untuk pengembangan sebuah *protocol* dalam komunikasi, dan atau dalam pendidikan. Pertama kali *wireshark* muncul dengan nama *Ethereal*, lalu pada bulan Mei tahun 2006 proyek ini mengganti namanya menjadi *Wireshark* karena ada permasalahan mengenai merk dagang. Bahasa pemrograman yang dipakai dalam *wireshark* adalah bahasa C dengan *public licenci* GNU.

*Wireshark* banyak digemari karena *interface wireshark* yang telah menggunakan tampilan grafis atau GUI. Seperti namanya, aplikasi *Wireshark* dapat menangkap beberapa paket data yang berkeliaran dalam lalu lintas jaringan yang dilihat. Seluruh jenis informasi paket dalam bermacam-macam format *protokol* pun bisa dengan mudah ditangkap dan dianalisis. Manfaat dari penggunaan aplikasi *wireshark* ialah menangkap informasi atau data paket yang dikirim dan diterima dalam jaringan komputer, mengetahui aktifitas yang terjadi



dalam jaringan komputer, mengetahui dan menganalisis kinerja jaringan komputer yang dimiliki seperti kecepatan akses data dan koneksi jaringan ke internet serta mengamati keamanan dari jaringan komputer [28].

### 2.2.13 Quality Of Service (QoS)

*Quality of service* (QoS) atau kualitas layanan digunakan untuk memenuhi persyaratan lalu lintas dari aplikasi sensitif, seperti suara dan video *real-time*, dan untuk mencegah penurunan kualitas yang disebabkan oleh *packet loss*, penundaan dan *jitter*. QoS mengontrol dan mengelola sumber daya jaringan dengan menetapkan prioritas untuk tipe data tertentu pada jaringan [29]. Berikut beberapa parameter QoS dan kategori parameter menurut standar THIPON:

#### A. Delay (*Latency*)

*Delay (latency)* merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan [29]. Persamaan 2.4 menunjukkan perhitungan *delay (latency)* :

$$\text{Rata - rata delay} = \frac{\text{Total Delay}}{\text{Total Paket Diterima}} \quad (2.4)$$

Pada Tabel 2.5 membahas tentang kategori *Delay*.

Tabel 2.5 Kategori *Delay* [30].

Kategori Latensi	Besar <i>Delay</i> (ms)	Indeks
Sangat Bagus	<150	4
Bagus	150 s/d 300	3
Sedang	300 s/d 450	2
Jelek	>450	1

#### B. *Throughput*

*Throughput* yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps (*bit per second*) [29]. Persamaan 2.5 menunjukkan perhitungan *throughput* :

$$\text{Throughput} = \frac{\text{Paket Data Diterima}}{\text{Lama Pengamatan}} \quad (2.5)$$

Pada Tabel 2.6 membahas tentang kategori *Throughput*.

Tabel 2.6 Kategori *Throughput* [30].

Kategori <i>Throughput</i>	<i>Throughput</i> (bps)	Indeks
Sangat Bagus	>2100k	4
Bagus	1200k-2100k	3
Sedang	700k-1200k	2
Jelek	338k-700k	1
Sangat Jelek	0-338k	0

C. *Packet Loss*

*Packet loss* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan [29]. Persamaan 2.6 menunjukkan perhitungan *packet loss*.

$$\mathbf{Packet\ Loss} = \frac{\mathbf{Paket\ Data\ Dikirim} - \mathbf{Paket\ Data\ Diterima}}{\mathbf{Paket\ Data\ Dikirim}} \times \mathbf{100\%} \quad \mathbf{(2.6)}$$

Pada Tabel 2.7 membahas tentang kategori *Packet Loss*.

**Tabel 2.7 Kategori *Packet Loss* [30].**

<b>Kategori Degradasi</b>	<b>Packet Loss (%)</b>	<b>Indeks</b>
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1