

## **BAB 2**

### **DASAR TEORI**

#### **2.1 TINJAUAN PUSTAKA**

Penelitian oleh Pretty Veronica Ertyan, Porman Pangaribuan, dan Agung Surya Wibowo pada tahun 2019 dengan judul “Sistem Monitoring Dan Mengontrol Aquarium Dalam Pemeliharaan Ikan Hias Dari Jarak Jauh” membahas tentang alat pendeteksi kekeruhan air didalam aquarium. Pemantauan kondisi kualitas air aquarium akan menggunakan mikrokontroller arduino mega 2560, dan sensor mendeteksi kekeruhan air adalah sensor suhu DS18B20. Sistem ini dibuat secara bertahap mulai dari perancangan mekanik, konfigurasi mikrokontroler, sinkronisasi pembacaan sensor, tampilan monitoring dan pengukuran parameter terhadap android. Hasil akhir berupa monitoring yang sudah diuji pada aquarium dalam pemeliharaan ikan hias dari jarak jauh. Pada penelitian ini menggunakan sensor pH, sensor *turbidity*, dan sensor suhu. Tingkat keasaman untuk ikan hias air tawar 6 sampai 7,5, tingkat kekeruhan <25 NTU, dan suhu aquarium 20 °C sampai dengan 28°C. Penelitian ini menggunakan *thingspeak* dengan prinsip *Internet of Thing* [6].

Penelitian oleh Habibi Ramdani Safitri pada tahun 2019 dengan judul “Rancang Bangun alat Pemberi Pakan Dan Pengganti Air Aquarium Otomatis Berbasis Arduino Uno” membahas tentang aquarium otomatis yang sangat membantu kita dalam mengerjakan penggantian air dan pemberi pakan pada ikan , biasanya akan di buat waktu untuk mengganti air dan pemberi pakan akan menyita waktu apalagi dalam keadaan sibuk , maka dirancanglah sebuah alat pemberi pakan dan pengganti air aquarium otomatis berbasis arduino. Alat ini menggunakan arduino yang merupakan alat pengendali untuk menggerakkan sistem secara keseluruhan. Pada alat ini menggunakan sensor kekeruhan yang berfungsi untuk mendeteksi keruhnya air pada aquarium. Sensor *water level* untuk mengukur ketinggian air pada aquarium. Motor servo akan memberikan perintah pada arduino untuk membuka pemberi pakan pada ikan. *Relay* berfungsi untuk mematikan dan menghidupkan pompa air pada aquarium dan pompa air 1 dan pompa air 2 juga berfungsi untuk pembuangan air dan penggantian air. dan *buzzer*

digunakan untuk memberikan tanda bunyi untuk pemberi pakan dan pengganti air , Lcd akan berfungsi untuk menampilkan waktu, tanggal dan pakan peringatan penggantian air sedangkan Rtc untuk menyimpan waktu dan tanggalnya [7].

Penelitian oleh Riyan Kharisma dan Suryadhi Thaha pada tahun 2020 dengan judul “Rancang Bangun Alat Monitoring Dan Penanganan Kualitas Air Pada Akuarium Ikan Hias Berbasis *Internet of Things* (IoT)” membahas tentang alat untuk memantau kondisi air dengan penanganannya, menggunakan sensor Ph meter sebagai pembaca kadar Ph, sensor suhu DS18B20 sebagai pembacaan suhu dalam air dan sensor TDS sebagai sensor pembacaan kadar salinitas pada air. Menggunakan mikrokontroler NodeMCU ESP8266 untuk mengakses data yang direspon oleh sensor – sensor tersebut. Data yang telah didapat dari sensor akan diakses oleh NodeMCU dan diteruskan melalui IoT untuk dikirimkan ke server dan diteruskan ke *user* (telegram) secara otomatis. Saat kondisi air tidak sesuai dengan kondisi normal dan datanya kurang atau lebih dari data air normal yang dimasukkan maka akan muncul notifikasi otomatis dan akan dilakukan penanganan secara otomatis [8].

Penelitian oleh Ivan Bagus Prasetyo, Aditya Akbar Riadi dan Ahmad Abdul Chamid pada tahun 2021 yang berjudul “Perancangan *Smart Aquarium* Menggunakan Sensor *Turbidity* Dan Sensor Ultrasonik Pada Akuarium Ikan Air Tawar Berbasis Arduino Uno” membahas tentang Perancangan *smart aquarium* ini menggunakan 2 buah sensor, sensor *turbidity* yang bertugas sebagai pembaca tingkat kekeruhan air dan sensor ultrasonik sebagai pembaca jarak air akuarium yang dikendalikan oleh Arduino Uno sebagai pusat kendali pada sensor, LCD (*Liquid Crystal Display*), *buzzer* dan modul Sim GSM M800L. Pergantian air dilakukan oleh pompa air berdasarkan kondisi tingkat kekeruhan air. Tujuan dari penelitian adalah merancang sebuah sistem kontrol dan monitoring kualitas air pada akuarium berbasis Arduino yang dapat mempermudah pecinta ikan dalam memonitoring dan penggantian air pada akuarium. Pengujian alat yang dilakukan didapat hasil dimana sistem penggantian air dapat berjalan sesuai dengan tingkat kekeruhan air berjalan dengan baik sesuai yang diharapkan [9].

Penelitian oleh Tito Rikanto dan Arita Witanti pada tahun 2021 dengan judul “Sistem Monitoring Kualitas Kekeruhan Air Berbasis *Internet Of Things*”

membahas tentang alat otomatis untuk mengontrol kualitas air. Penelitian ini berbasis *Internet of Things* menggunakan Arduino UNO, ESP8266 dan LDR sebagai sensornya. Pengujian modul menggunakan 30 sampel air berbeda kondisi dengan total uji sampel sebanyak 90 kali dan pengujian dengan konsep waterflow. Hasil penelitian berupa perangkat modul yang dapat memantau dan mengontrol kualitas kekeruhan air diharapkan mampu memberikan hasil akurat dalam memonitoring kualitas air pada bak tampung serta dapat mempermudah dan membantu manusia untuk terhindar dari penyakit akibat kualitas air yang tidak baik Berdasarkan hasil perhitungan uji sampel, penerapan mikrokontroler berbasis *Internet of Things* dalam memonitoring kualitas kekeruhan air didapat tingkat keakurasian modul dalam mendeteksi kualitas air sebesar 96,67% [10].

**Tabel 2.1 Daftar Tinjauan Pustaka**

No	Judul	Tahun	Sensor	Komunikasi
1	Perancangan <i>Smart Aquarium</i> Menggunakan Sensor <i>Turbidity</i> Dan Sensor Ultrasonik Pada Akuarium Ikan Air Tawar Berbasis Arduino Uno	2021	Sensor <i>turbidity</i> dan sensor ultrasonik	Seluler
2	Sistem Monitoring Kualitas Kekeruhan Air Berbasis <i>Internet Of Things</i>	2021	Sensor LDR	<i>Wireless</i>
3	Rancang Bangun Alat Monitoring Dan Penanganan Kualitas Air Pada Akuarium Ikan Hias Berbasis <i>Internet of Things (IoT)</i>	2020	Sensor pH, sensor DS18B20 dan sensor TDS	<i>Wireless</i>
4	Sistem Monitoring Dan Mengontrol Aquarium Dalam Pemeliharaan Ikan Hias Dari Jarak Jauh	2019	Sensor pH, sensor kekeruhan dan sensor DS18B20	<i>Wireless</i>

5	Rancang Bangun alat Pemberi Pakan Dan Pengganti Air Aquarium Otomatis Berbasis Arduino Uno	2019	Sensor kekeruhan dan sensor <i>waterlevel</i>	Elektronik
---	--	------	---	------------

## 2.2 DASAR TEORI

### 2.2.1 Ikan Maskoki

Ikan maskoki yang memiliki nama latin “*Carrasius auratur*” awalnya dibudidayakan oleh masyarakat China berkisar pada tahun 960 hingga 1729. Yang awalnya bentuk maskoki ini hampir sama seperti ikan mas, karena memang jenis ikan ini berasal dari satu kerabat. Bedanya maskoki tidak mempunyai sepasang sungut di mulutnya. Lalu pada masa dinasti Ming “1368-1644” popularitas maskoki mulai ramai menanjak, maka dari situlah bermunculan maskoki dengan bentuk tubuh yang bervariasi dan unik [11].



Gambar 2.1 Ikan maskoki (*Carrasius auratur*)[11]

Cara memelihara ikan maskoki

#### 1. Derajat keasaman (pH)

Nilai pH yang ideal untuk menumbuh kembangkan maskoki berkisar dari 7.0 sampai 8.0. Hal yang perlu diperhatikan adalah membiarkan terlebih dahulu maskoki yang akan dipelihara untuk beradaptasi dengan kualitas air yang ada. Tidak disarankan untuk melakukan penggantian air atau melakukan perubahan kualitas air secara drastis.

#### 2. Komposisi

Maskoki sangat sensitif terhadap kadar *chlorine* yang digunakan sebagai disinfektan pada air olahan. Oleh sebab itu, dalam menyiapkan air pastikan

tersebut bebas dari *chlorine*. Hal ini dapat dilakukan dengan mengendapkan air terlebih dahulu, atau dengan cara menambahkan anti-*chlorine*.

### 3. Suhu

Suhu ideal untuk maskoki berada pada kisaran 20°C – 25 °C. Fluktuasi perubahan suhu direkomendasikan tidak melebihi dari 5 °C, terutama dalam proses pergantian air atau proses transportasi. Fluktuasi suhu diatas 5 °C akan sangat membahayakan ikan maskoki.

### 4. Kepadatan

Kepadatan ikan dalam volume air tertentu akan sangat menentukan tingkat keberhasilan memelihara maskoki, kemudahan dalam mengelola air dan menghindarkan terjadinya stress yang dapat memicu terjadinya berbagai masalah lanjutan lainnya. Disarankan agar maskoki dipelihara dengan kepadatan serendah mungkin, atau dipelihara sesedikit mungkin dalam suatu wadah [11].

## 2.2.2 *Internet of Things (IoT)*

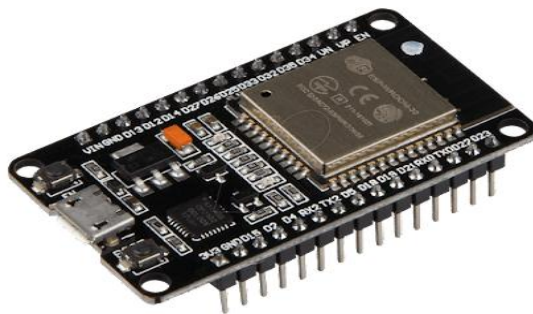
*Internet of things* dalam pengertian secara luas membuat semua yang ada di dunia terkoneksi ke dalam internet yang tersambung secara terus menerus. *Internet of things* bisa mengontrol, mengirim data, dan sebagainya yang memanfaatkan internet sehingga bisa dilakukan dengan jarak jauh tanpa mengenal jarak. Konsep dasar dari *internet of things* adalah dengan menggabungkan obyek, sensor, *controller*, dan internet yang bisa menyebarkan informasi kepada pengguna. Obyek akan dideteksi oleh sensor yang akan diproses oleh *controller* dan dilanjutkan untuk mengirim data yang sudah diolah sehingga menjadi sebuah informasi yang berguna dan secara *real-time* kepada pengguna [12].

### 2.2.3 Network Time Protocol (NTP)

Merupakan standar internet protokol untuk menyinkronkan waktu komputer dengan beberapa referensi internet. Lebih mudah dipahami pengertian NTP untuk mendapatkan data waktu dari internet berdasarkan beberapa referensi server sesuai dengan lokasi yang di tentukan. Sama halnya dengan waktu tanggal dan jam pada laptop atau *smartphone* yang digunakan, selain terdapat rtc (*Real Time Clock*) juga terdapat internet yang bisa mensinkronkan waktu, jadi hanya memilih *timezone*, selebihnya waktu secara otomatis akan mengikuti yang sebenarnya dengan referensi dari NTP server. Kemudian cara menginstal *library* NTPClient.h Arduino IDE yaitu membuka *software* Arduino IDE, pilih *Sketch*, pilih *include library*, pilih *manage libraries*, mencari NTPClient lalu klik *install* [13].

### 2.2.4 NodeMCU ESP32

ESP32 merupakan mikrokontroler berdaya rendah pada seri chip (SoC) dengan *Wi-Fi* & kemampuan *Bluetooth* dua mode. Keluarga ESP32 termasuk chip ESP32-D0WDQ6 (dan ESP32-D0WD), ESP32-D2WD, ESP32-S0WD, dan sistem dalam paket (SiP) ESP32-PICO-D4. Pada intinya, ada mikroprosesor *Tensilica Xtensa LX6 dual-core* atau *single-core* dengan *clock rate* hingga 240 MHz. ESP32 sudah terintegrasi dengan *built-in antenna switches*, RF balun, *power amplifier*, *low-noise receive amplifier*, *filters*, and *power management modules*. Didesain untuk perangkat seluler, perangkat elektronik yang dapat dipakai, dan aplikasi IoT, ESP32 juga bekerja dengan menggunakan daya sangat rendah melalui fitur hemat daya termasuk *fine resolution clock gating*, *multiple power modes*, and *dynamic power scaling* [14].



Gambar 2.2 NodeMCU ESP32 [14]

### 2.2.5 Sensor pH SEN0161



**Gambar 2.3 Sensor pH SEN0161 [15]**

Sistem kerja pH sensor terletak pada *probe* pH yang terbuat dari bahan kaca. Reaksi kimia pada ujung *probe* pH menyebabkan perbedaan tegangan dan perbedaan tegangan ini yang nantinya diukur dan dijadikan satuan pH. Secara pengertian, pH itu merupakan nilai yang digunakan untuk mengukur tingkat keasaman atau alkalinitas dalam suatu larutan. *Range* nilai pH yaitu antara angka 0 hingga 14. Jika nilai pH dibawah 7 maka dinyatakan asam sedangkan jika nilai ph diatas 7 maka dinyatakan basa [15]. Sensor pH memiliki spesifikasi seperti pada tabel 2.2

**Tabel 2.2 Spesifikasi Sensor pH SEN0161**

<b>Nama</b>	<b>Keterangan</b>
Tegangan Modul	3.3 Volt
Tingkat Keakuratan	$\pm 0.1\text{pH}$ (25°C)
Batas Temperatur	0°C S/d 60°C
Rentang Pengukuran	pH 0 sampai 14

### 2.2.6 Sensor *Turbidity* SKU SEN0189

Sistem sensor yang dibuat terdiri dari sebuah detektor fotodioda TSL 250 dan juga dioda laser sebagai sumber cahayanya yang mempunyai panjang gelombang 650 nm. Keduanya komponen ini dirangkai sehingga membentuk posisi sudut 90° antara kedua komponen ini. Rangkaian sistem sensor ini berfungsi untuk mendeteksi tingkat kekeruhan air dengan cara melewatkan air diantara *detector* dan sumber cahayanya. Fotodioda TSL 250 sebagai *detector* sangat peka terhadap perubahan intensitas cahaya yang masuk ke dalamnya. Sumber cahaya yang ditembakkan dalam hal ini adalah dioda laser akan mengenai air, dan apabila dalam air tersebut banyak sekali terdapat partikel dalam kata lain

keruh, maka cahaya tersebut sebagian akan ada yang diteruskan dan sebagian akan dihamburkan. Intensitas cahaya yang diterima oleh fotodiode TSL 250 ini adalah intensitas cahaya yang dihamburkan oleh partikel yang ada dalam air. Intensitas cahaya yang diterima oleh fotodiode TSL 250 akan dikonversi menjadi sinyal tegangan dan sinyal tegangan keluaran dari alat ini menunjukkan nilai tegangan yang sebanding dengan tingkat kekeruhan air [16].



**Gambar 2.4** Sensor *Turbidity* SKU SEN0189 [16]

### 2.2.7 Sensor *Dallas DS18B20*



**Gambar 2.5** Sensor Suhu *Dallas DS18B20* [17]

DS18B20 adalah sensor suhu digital seri terbaru dari *Maxim IC* yang mampu membaca suhu dengan ketelitian 9 sampai 12-bit, memiliki rentang  $-55^{\circ}\text{C}$  hingga  $125^{\circ}\text{C}$  dengan ketelitian ( $\pm 0,5^{\circ}\text{C}$ ). Sensor ini menggunakan komunikasi *1-Wire* yang artinya hanya membutuhkan satu pin saja. Sensor DS18B20 memiliki keunikan yaitu 64-bit, yang memungkinkan DS18B20 terhubung ke beberapa fungsi yang sama melalui satu kabel yang sama [17]. Sensor *Dallas DS18B20* memiliki spesifikasi seperti pada tabel 2.3 :



Tabel 2.3 Spesifikasi Sensor Dallas DS18B20

Nama	Keterangan
DC Supply Voltage	3-5.5 V
Tingkat Keakuratan	$\pm 0.5^\circ$ (Pada Kisaran $-10^\circ\text{C} - 85^\circ\text{C}$ )
Batas Temperatur	$-55^\circ\text{C}$ S/d $+125^\circ\text{C}$
Output	Digital 1-wire
Resolusi ADC	9-bit
9-bit	750

### 2.2.8 Motor Servo SG90

Merupakan suatu motor yang dapat diatur dan dikendalikan menggunakan pulsa. Motor ini terdiri dari beberapa komponen seperti sebuah motor DC, potensiometer, serangkaian *gear* dan rangkaian kendali. Dimana potensiometer memiliki fungsi untuk membaca sudut digital *encoder* dari putaran servo dan motor servo ini beroperasi secara *close loop* yang kemudian poros motor dihubungkan pada rangkaian kendali, apabila putaran poros belum mencapai posisi yang diperintahkan ada *coding* maka rangkaian kendali akan memperbaiki posisi sampai diposisi yang diperintahkan.

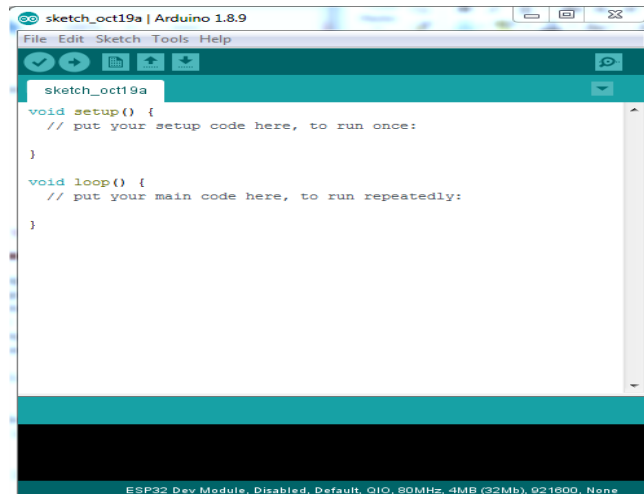
Terdapat dua jenis motor servo yaitu motor servo *continuous* dan motor servo standar. Perbedaan dari kedua jenis ini yaitu pada motor servo standar hanya dapat bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai  $90^\circ$  sehingga defleksi keduanya sebesar  $180^\circ$ , sedangkan motor servo *continuous* yang pada bagian *feedback*-nya dilepas sehingga dapat bergerak dua arah dengan tanpa batasan defleksi sudut putar mencapai sudut  $360^\circ$  [18].



Gambar 2.6 Motor Servo SG90 [18]

### 2.2.9 Arduino IDE

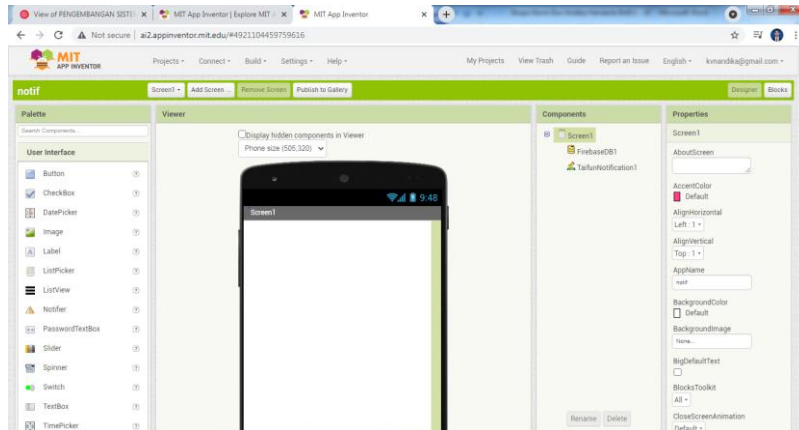
ArduinoIDE, adalah sebuah editor yang di gunakan untuk menulis program, meng*compile*, dan mengunggah ke papan Arduino. Arduino *Development Environment* terdiri dari editor teks, toolbar, dengan tombol-tombol untuk fungsi umum, dan sederet menu [19].



Gambar 2.7 Tampilan Arduino IDE

### 2.2.10 MIT App Inventor

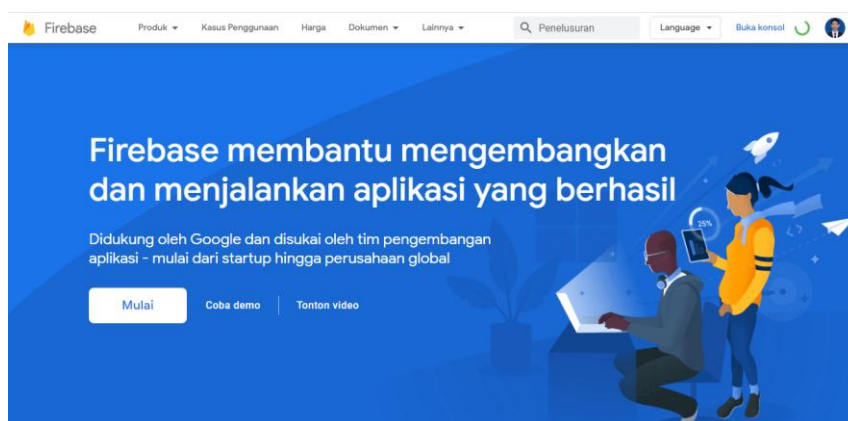
MIT App *Inventor* merupakan *platform* yang menyederhanakan proses pembuatan aplikasi sederhana tanpa harus belajar atau menggunakan terlalu banyak bahasa pemrograman. Dapat menggunakan berbagai *layout* dan komponen yang tersedia untuk mendesain aplikasi android sesuai kebutuhan. Pada MIT App *Inventor* terdapat dua halaman utama, yaitu halaman *designer* dan halaman *blocks*. Halaman desainer digunakan untuk mendesain tampilan aplikasi dan menyediakan berbagai komponen dan *layout* sesuai kebutuhan. Pada saat yang sama, blok man berfungsi untuk memprogram pengoperasian aplikasi android sesuai dengan tujuannya [20].



**Gambar 2.8 Tampilan MIT App Inventor**

### 2.2.11 Google Firebase

*Firebase* merupakan penyedia layanan *real time database* dan *backend* kepunyaan Google. *Firebase* ini merupakan solusi yang ditawarkan untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya *Firebase*, pengembang aplikasi dapat fokus mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan *backend*. *Firebase Real Time Database* merupakan fitur yang memberikan sebuah *NoSQL database* yang bisa diakses secara *Real Time* oleh penggunanya. *Firebase Real Time Database* memiliki kelebihan yaitu aplikasi dapat menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan *sync* data segera setelah memperoleh akses internet [21].

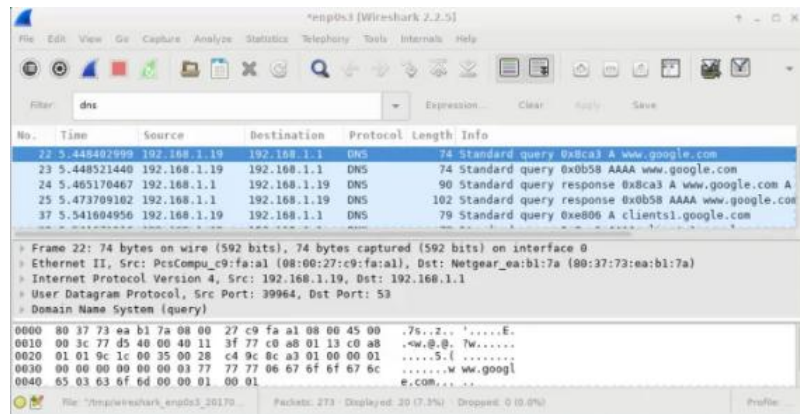


**Gambar 2.9 Tampilan Google Firebase**

### 2.2.12 Wireshark

Wireshark merupakan sebuah aplikasi *capture* paket data berbasis *open-source* yang berguna untuk memindai dan menangkap trafik data pada jaringan internet. Aplikasi ini umum digunakan sebagai alat *troubleshoot* pada jaringan yang bermasalah, selain itu juga biasa digunakan untuk pengujian *software* karena kemampuannya untuk membaca konten dari tiap paket trafik data.

Wireshark memiliki fungsi untuk pekerjaan analisis jaringan. Cara kerjanya yaitu dengan ‘menangkap’ paket-paket data dari protokol-protokol yang berbeda dari berbagai tipe jaringan yang umum ditemukan di dalam trafik jaringan internet. Paket-paket data tersebut ‘ditangkap’ lalu ditampilkan di jendela hasil *capture* secara *real-time* [22].



Gambar 2.10 Tampilan Software Wireshark [22]

### 2.2.13 Quality of Service (QoS)

*Quality of Service* (QoS) atau kualitas layanan merupakan metode pengukuran yang digunakan untuk menentukan kemampuan sebuah jaringan seperti; aplikasi jaringan, *host* atau *router* dengan tujuan memberikan *network service* yang lebih baik dan terencana sehingga dapat memenuhi kebutuhan suatu layanan [23]. Parameter-parameter *quality of service* menurut standar THIPHON :

#### 1. *Throughput*

*Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [24].

**Tabel 2.4 Kategori *Throughput* Menurut TIPHON**

<b>Kategori <i>Throughput</i></b>	<b>Indeks</b>	<b><i>Throughput</i></b>
Sangat Bagus	76-100%	4
Bagus	51-75%	3
Sedang	26-50%	2
Buruk	25%	1

Rumus persamaan mencari nilai *throughput* :

$$Throughput = \frac{\text{data diterima}}{\text{lama waktu pengamatan}}$$

2. *Delay*

*Delay* merupakan waktu yang dibutuhkan data untuk menempuh jarak dari titik asal ke titik tujuan [24].

**Tabel 2.5 Kategori *Delay* Menurut TIPHON**

<b>Kategori <i>Latency</i></b>	<b><i>Delay</i></b>	<b>Indeks</b>
Sangat Bagus	< 150 m/s	4
Bagus	150 s/d 300 m/s	3
Sedang	300 s/d 450 m/s	2
Buruk	> 450 m/s	1

Rumus persamaan mencari nilai *delay* :

$$Delay = \frac{\text{total delay}}{\text{total packet yang diterima}}$$

3. *Jitter*

*Jitter* diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket diakhir perjalanan [24].

**Tabel 2.6 Kategori Jitter Menurut TIPHON**

<b>Kategori Jitter</b>	<b>Jitter (m/s)</b>	<b>Indeks</b>
Sangat Bagus	0 m/s	4
Bagus	0 s/d 75 m/s	3
Sedang	75 s/d 125 m/s	2
Buruk	125 s/d 225 m/s	1

Rumus persamaan mencari nilai *jitter* :

$$Jitter = \frac{\text{total variasi delay}}{\text{total packet yang diterima}}$$

#### 4. *Packet Loss*

*Packet loss* merupakan suatu parameter yang memberikan suatu kondisi yang menunjukkan jumlah *packet* yang hilang [24].

**Tabel 2.7 Kategori Packet Loss Menurut TIPHON**

<b>Kategori Packet Loss</b>	<b>Indeks</b>	<b>Throughput</b>
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Buruk	25%	1