BAB II
TINJAUAN PUSTAKA

2.1 Studi Literatur

Beberapa penelitian telah dilakukan untuk mengatasi masalah Keamanan Data Internet of Things (IOT). Beberapa ringkasan Studi Pustaka digunakan untuk mengetahui sejauh mana penelitian telah dilakukan.

Dilakukan oleh (Ariyanto, 2018) dengan judul "Algoritma RC4 dalam Proteksi Transmisi dan Hasil Query Untuk Ordbms Postgresql". Pada penelitian ini dibahas mengenai bagaimana mengimplementasikan algoritma kriptografi RC4 dalam proteksi terhadap query dan hasil query, pengamanan dilakukan dengan cara melakukan enkripsi dan dekripsi selama keduanya berada di dalam jaringan. Pengimplementasian dari penelitian ini yaitu membangun sebuah software yang akan diletakkan di sisi client yang berfungsi mengakses database yang diletakkan di sisi server. Software yang dibangun memiliki fasilitas untuk mengenkripsi dan mendektipsi query dan hasil query yang dikirimkan dari client ke server dan juga sebaliknya. Dengan demikian tramsmisi query dan hasil query dapat terjamin keamanannya.Terjaminnya keamanan transmisi query dan hasil query dapat dikatakan berhasil jika software berhasil mengenkripsi query dan hasil query yang ditransmisikan sehingga apabila terjadi penyadapan terhadap keduanya, penyadap tidak akan mengerti isi data tersebut. Kesimpulan dari penelitian ini yaitu software yang dibangun berhasil mengenkripsi query dan hasil query yang ditransmisikan antara aplikasi client dan server database.

Selanjutnya dilakukan oleh (Saragi et al., 2020) dengan judul "Pengamanan File Teks Menggunakan Algoritma RC4 Bister". Pada penelitian ini dibahas mengenai bagaimana mengimplementasikan algoritma kriptografi RC4 pada sebuah file. Penggunaan algoritma kriptografi RC4 bertujuan untuk melindungi dari serangan yang dapat membahayakan data yang dimiliki seperti dari serangan Brute Force. Untuk melindungi suatu data atau pesan agar tidak dibaca oleh pihak yang tidak berwenang serta mencegah pihak – pihak tersebut

melakukan perubahan data, menghapus data serta menyisipkan sesuatu yang tidak pantas yang mengganggu keaslian dari data tersebut.

Selanjutnya dilakukan oleh (Seputra et al., 2020) dengan judul "Kriptografi Simetris RC4 pada Transaksi Online Booking Engine System". Pada penelitian ini dibahas mengenai bagaimana mengimplementasikan algoritma kriptografi RC4 pada penjualan/transaksi paket wisata secara online melalui internet booking engine system (BES). Algoritma RC4 diimplementasikan guna melakukan enkripsi data pelanggan yang disimpan dalam basis data. Informasi yang sudah disimpan tersebut akan ditampilkan kembali ke plaintext melalui proses dekripsi menjadi informasi yang sebenarnya. Modul kritptografi disisipkan pada aplikasi, yaitu pada akhir proses transaksi sebelum data tersimpan. BES yang dibuat menggunakan framework codeigniter sangat mendukung untuk implementasi kriptografi dengan efisien. Pengujian algoritma dilakukan terhadap 922 baris data dengan melihat waktu dan memori yang digunakan dalam melakukan enkripsi hingga dekripsi. Hasil pengujian performa menunjukkan bahwa panjang kunci mempengaruhi waktu pemrosesan data.

Selanjutnya penelitian yang dilakukan (Andrico & Syafrullah, 2018) dengan judul "Aplikasi Enkripsi Database Menggunakan Algortima RC4 Berbasis Desktop". Pada penelitian ini dibahas mengenai implementasi kriptografi algoritma RC4 untuk membangun suatu aplikasi yang dapat mengenkripsi data dan informasi yang ada didalam database sehingga menghalangi pihak yang tidak bertanggung jawab untuk mengubah, mengambil, atau menyalahgunakan data dan informasi tersebut dengan menggunakan algoritma RC4.

Berdasarkan studi literatur yang telah dilakukan, maka sistem yang akan dibangun memiliki beberapa perbedaan, yaitu menggunakan sensor PIR dan DHT-11 serta menggunakan NodeMCU untuk mendeteksi gerakan dan mengambil data suhu dan kelembaban. Data yang dibaca oleh sensor tersebut di enkripsi lalu dikirim ke web server untuk di dekripsi. Sehingga data yang dikirim tidak dapat dilihat oleh pengguna yang tidak terautentikasi.

2.2 Landasan Teori

2.2.1 Internet of Things

IoT dapat didefinisikan sebagai infrastruktur jaringan global yang dinamis dengan konfigurasi mandiri. IoT memiliki kemampuan untuk membuat segala sesuatu di sekitar kita dapat terkoneksi ke Internet dengan perilaku yang cerdas (H. et al., 2015). Internet of things juga dapat didefinisikan sebagai suatu konsep atau program dimana sebuah objek memiliki kemampuan untuk mentransmisikan atau mengirimkan data melalui jaringan tanpa menggunakan bantuan perangkat komputer dan manusia. Internet of things atau sering disebut dengan IoT saat ini mengalami banyak perkembangan. Perkembangan IoT dapat dilihat mulai dari tingkat konvergensi teknologi nirkabel, microelectromechanical (MEMS), internet, dan QR (Quick Responses) Code. IoT juga sering diidentifikasi dengan RFID (Radio Frequency Identification) sebagai metode komunikasi. Selain itu, juga mencakup teknologi berbasis sensor, seperti teknologi nirkabel, QR Code yang sering kita jumpai. Kemampuan dari IoT sendiri tidak perlu diragukan lagi. Banyak sekali teknologi yang telah menerapkan sistem IoT, sebagai contoh sensor cahaya, sensor suara dari teknologi Google terbaru, yaitu Google Ai, dan Amazon Alexa. Dan yang terbaru saat ini, penerapan Smart City yang sudah dilakukan di beberapa negara maju, seperti China dan Jerman. Sehingga, segala bentuk aktivitas penduduk suatu kota dapat termonitoring dengan baik oleh sistem dengan jaringan basis data berskala besar.

2.2.2 Data di Perangkat IoT

Secara umum data dapat diartikan sebagai catatan kumpulan fakta. Data merupakan bentuk jamak dari *datum*, berasal dari bahasa latin yang artinya “sesuatu yang diberikan”. Dalam penggunaan sehari – hari, data berarti pernyataan yang diterima apa adanya. Pernyataan ini adalah hasil pengukuran atau observasi suatu variabel yang bentuknya bisa berupa angka, kata, atau gambar. Namun yang dimaksud dengan data pada perangkat Internet of Things adalah masukan yang berasal dari sebuah sensor yang selanjutnya akan

dikirimkan ke suatu media penyimpanan seperti web server dan selanjutnya diproses untuk menentukan suatu keluaran tertentu.

Data di Internet of Things sangat rentan terhadap semua jenis serangan. Tanpa adanya keamanan data di Internet of Things tentunya dapat menimbulkan resiko dimana informasi yang sensitif dan berharga dapat di akses oleh orang yang tidak bertanggung jawab. Selain itu data yang diretas kemungkinan akan rusak atau hilang, yang menyebabkannya kerusakan pada sistem IoT. Oleh karena itu keamanan data di Internet of Things sangat diperlukan (H. et al., 2015).

2.2.3 Kriptografi

Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. Pesan yang dikirim oleh pengirim perlu melalui suatu tahapan yang disebut enkripsi agar dapat diterima dengan aman. Enkripsi adalah proses untuk mengubah pesan asli (*plaintext*) menjadi pesan rahasia (*ciphertext*). Sebaliknya, proses untuk mengubah *ciphertext* menjadi *plaintext* disebut dekripsi. Berdasarkan jenis kunci yang digunakan, sistem kriptografi dapat dibedakan menjadi *symmetric key* dan *asymmetric key*. Selain berdasarkan jenis kunci yang digunakan, sistem kriptografi juga dapat dibedakan menjadi *block cipher* dan *stream cipher* (Fernando & Lukas, 2017).

Symmetric key atau enkripsi simetris adalah metode kriptografi yang menggunakan satu kunci yang sama untuk enkripsi *plaintext* menjadi *ciphertext* dan melakukan dekripsi *ciphertext* menjadi *plaintext*. Enkripsi simetris merupakan algoritma enkripsi tertua dan tercepat secara performa. Kunci yang digunakan dapat berupa angka, kata-kata atau sekumpulan karakter acak. Penerima informasi dan pengirim akan memegang kunci yang sama untuk proses pertukaran data di dalam jaringan komputer (Kurniawan, 2018).

Stream cipher terdiri dari dua komponen utama, yaitu *mixing function* dan *key stream*. Fungsi pencampuran (*mixing function*) biasanya menggunakan operasi XOR, sedangkan generator *key stream* adalah unit utama dalam enkripsi. Jika *stream cipher* menghasilkan deret nol, *stream cipher* yang dibuat akan identik

dengan *plaintext* aslinya. Algoritma *block cipher* merupakan algoritma yang mengubah blok N-bit data *plaintext* di bawah penentuan kunci rahasia dan menghasilkan blok N-bit data yang dienkripsi lainnya (Sharif & Mansoor, 2010).

Perangkat IoT yang dibahas dalam penelitian ini lebih fokus pada perangkat mikrokontroler. Umumnya data yang dikirim dari mikrokontroler ke server masih berbentuk plaintext, artinya data itu bisa dibaca oleh siapa saja. Metode enkripsi dapat digunakan untuk mengamankan data tersebut. Metode enkripsi yang tepat digunakan yaitu metode enkripsi simetris, karena metode enkripsi simetris lebih cepat dan efisien dibandingkan dengan metode enkripsi asimetris (Ariyanto, 2018).

2.2.4 Rivest Code 4

Rivest Code 4 (RC4) sebagai salah satu algoritma enkripsi stream chiper yang dirancang untuk dapat diimplementasikan secara efektif dan efisien. RC4 sangat populer untuk aplikasi internet, antara lain RC4 digunakan dalam standard *wireless equivalent privacy* (WEP) dan *transport layer security* (TLS). Algoritma kriptografi RC4 dibuat oleh RSA Data Security Inc (RSADSI) berbentuk *stream chiper*. Algoritma RC4 sukses menjadi salah satu algoritma *stream cipher* yang banyak digunakan dalam metode kriptografi. Kepopuleran algoritma ini sangat didukung oleh kecepatan, efisiensi, dan ke efektifan dalam penggunaan sumber daya sehingga sangat baik untuk diimplementasikan pada hardware maupun software (Jawad et al., 2020).

Algoritma RC4 memiliki tiga proses untuk mengenkripsi *plaintext* menjadi *ciphertext*. Pertama diakukan inisialisasi array skey dan sbox dengan panjang 256 byte. Variabel sbox berisi angka dengan nilai awal 0 sampai 255. Variabel sbox digunakan untuk melakukan subtitusi pada proses *key scheduling algorithm* (KSA). Panjang key yang digunakan adalah 256 byte, jika panjang key kurang dari 256 byte maka key akan di ulang hingga panjangnya mencapai 256 byte pada variabel skey.

```
58  key = (unsigned char *)"Bambang Fitriadi Wiansyah";
59  int sbox[256], skey[256];
60  //init skey and sbox
61  for (i = 0; i < 256; i++) {
62    sbox[i] = i;
63    skey[i] = key[(i % strlen((char*)key))];
64  }
```

Gambar 2.1 Inisialisasi Variabel key dan sbox

Tahap kedua dilakukan *key scheduling algorithm* dengan melakukan subtitusi pada variabel sbox dengan menggunakan array pada variabel skey.

```
65  //KSA
66  for (i = 0; i < 256; i++) {
67    j = (j + sbox[i] + skey[i]) % 256;
68    temp = sbox[i];
69    sbox[i] = sbox[j];
70    sbox[j] = temp;
71  }
```

Gambar 2.2 Proses Key Scheduling Algorithm (KSA)

Tahap terakhir dilakukan *pseudo-random generation algorithm* (PGRA) bertujuan untuk menghasilkan *keystream*. Setiap putaran, bagian *keystream* sebesar 1 byte dengan nilai antara 0 sampai dengan 255. Tahap ini menjadi penting karena akan banyak membutuhkan iterasi untuk memodifikasi state dari *keystream*. Setiap perulangan, PGRA melakukan increament terhadap i, kemudian menambah state yang terdiri dari matriks i dan j.

```
72  //PGRA
73  ciphertext = "";
74  for (i = j = n = 0; n < (int)strlen((char *)plaintext); n++) {
75    i = (i + 1) % 256;
76    j = (j + sbox[i]) % 256;
77    temp = sbox[i];
78    sbox[i] = sbox[j];
79    sbox[j] = temp;
80    t = (sbox[i] + sbox[j]) % 256;
81    keyStream = sbox[t];
82    ciphertext += (plaintext[n] ^ keyStream);
83    if (n < (int)strlen((char *)plaintext) - 1) {
84      ciphertext += ",";
85    }
86  }
```

Gambar 2.3 Pseudo-random Generation Algorithm (PGRA)

Algoritma RC4 mengenkripsi dengan mengombinasikan plainteks menggunakan bit-wise XOR yang memiliki panjang kunci dari 1 sampai 256 byte. Kombinasi kunci tersebut digunakan untuk menginisialisasikan tabel sepanjang 256 byte yang berfungsi sebagai generasi dari pseudo-random untuk kemudian di operasikan XOR dengan plaintext untuk menghasilkan ciphertext. Setiap elemen dalam tabel saling ditukarkan minimal sekali. Proses dekripsi dilakukan dengan cara yang sama seperti halnya proses enkripsi dengan kunci yang sama (Seputra et al., 2020).

2.2.5 Black Box Testing

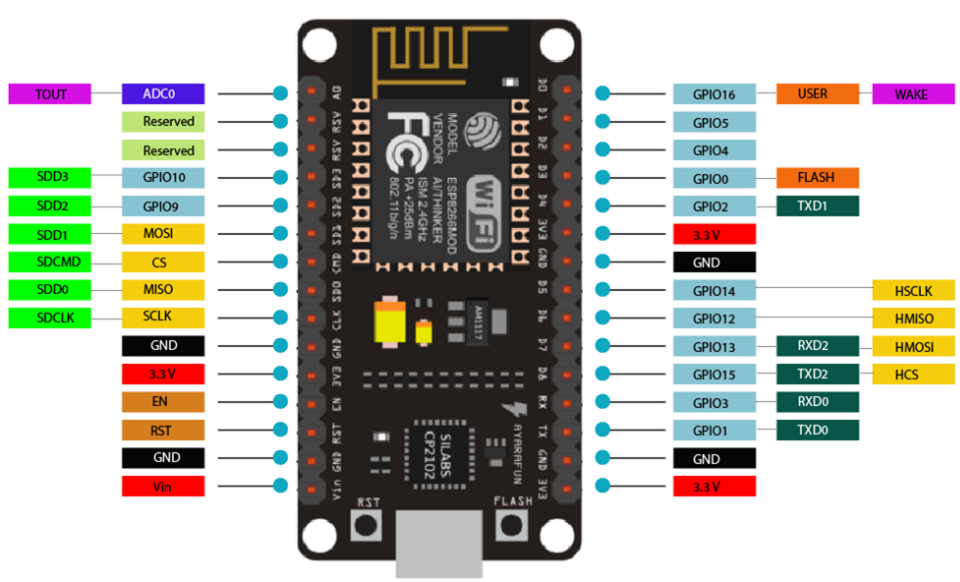
Black Box Testing merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program (Syafnidawaty, 2021). Pengujian *black box* juga disebut sebagai pengujian fungsional, teknik pengujian fungsional yang merancang kasus uji berdasarkan informasi dari spesifikasi. Dengan pengujian *black box*, penguji perangkat lunak tidak memiliki akses ke *source code* internal itu sendiri. Pengujian *black box* tidak berkaitan dengan mekanisme internal suatu sistem. Pengujian ini hanya berfokus pada output yang dihasilkan sebagai respons terhadap input dan kondisi eksekusi yang dipilih oleh suatu sistem. Penguji perangkat lunak hanya mengetahui bahwa informasi dapat dimasukkan ke dalam sistem, dan sistem akan mengirimkan sesuatu kembali. Hal ini dapat dilakukan berdasarkan pengetahuan spesifikasi kebutuhan. Penguji tahu apa yang diharapkan dari sistem untuk dikirim dan menguji untuk memastikan sistem mengirimkan apa yang seharusnya dikirim (Nidhra & Dondeti, 2012).

2.3 Perangkat Keras Yang Digunakan

2.3.1 NodeMCU

NodeMCU pada dasarnya adalah pengembangan dari esp8266 dengan *firmware* berbasis e-Lua. Pada NodeMcu dilengkapi dengan mikro usb port yang berfungsi untuk pemorgaman maupun *power supply*. Selain itu juga pada NodeMCU di lengkapi dengan tombol *push button* yaitu tombol *reset* dan *flash*. NodeMCU menggunakan bahasa pemorgamanan Lua yang merupakan

package dari esp8266. Selain dengan bahasa Lua, NodeMCU juga support dengan sofware Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE. Sebelum digunakan *board* ini harus di *Flash* terlebih dahulu agar mendukung *tool* yang akan digunakan. Arduino IDE menggunakan *firmware* yang cocok yaitu firmware keluaran dari Ai-Thinker yang support AT Command. Untuk penggunaan *tool loader firmware* yang di gunakan adalah firmware NodeMCU. NodeMCU memiliki spesifikasi sebagai berikut :



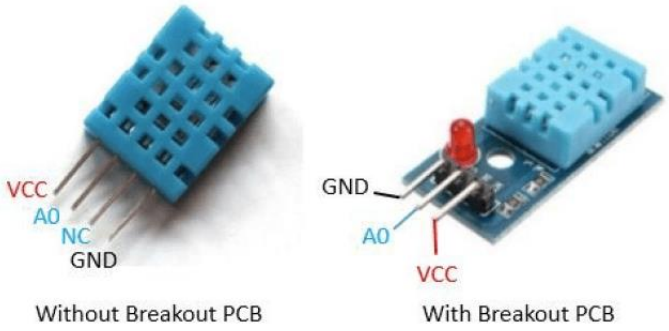
Gambar 2.4 Pinout NodeMCU

- Mikrokontroller : ESP8266-12E
- Ukuran Board : 57mm x 30mm
- Tegangan input : 3.3V – 5V
- GPIO : 13 Pin
- 10 bit ADC Pin : 1 Pin
- Flash Memory : 4 MB
- Clock Speed : 40/26/24 MHz
- WiFi : IEEE 802.11 b/g/n
- Frekuensi : 2.4 GHz–22.5 Ghz
- USB Port : Micro USB

2.3.2 Sensor DHT11

Sensor DHT11 adalah module sensor yang berfungsi untuk mensensing objek suhu dan kelembaban yang memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler. Module sensor ini tergolong kedalam elemen resistif seperti perangkat pengukur suhu seperti contohnya

yaitu NTC. Kelebihan dari module sensor ini dibanding module sensor lainnya yaitu dari segi kualitas pembacaan data sensing yang lebih responsif yang memliki kecepatan dalam hal sensing objek suhu dan kelembaban, dan data yang terbaca tidak mudah terinterverensi. Sensor DHT11 pada umumya memiliki fitur kalibrasi nilai pembacaan suhu dan kelembaban yang cukup akurat. Penyimpanan data kalibrasi tersebut terdapat pada memori program OTP yang disebut juga dengan nama koefisien kalibrasi. Sensor DHT11 memiliki spesifikasi sebagai berikut :



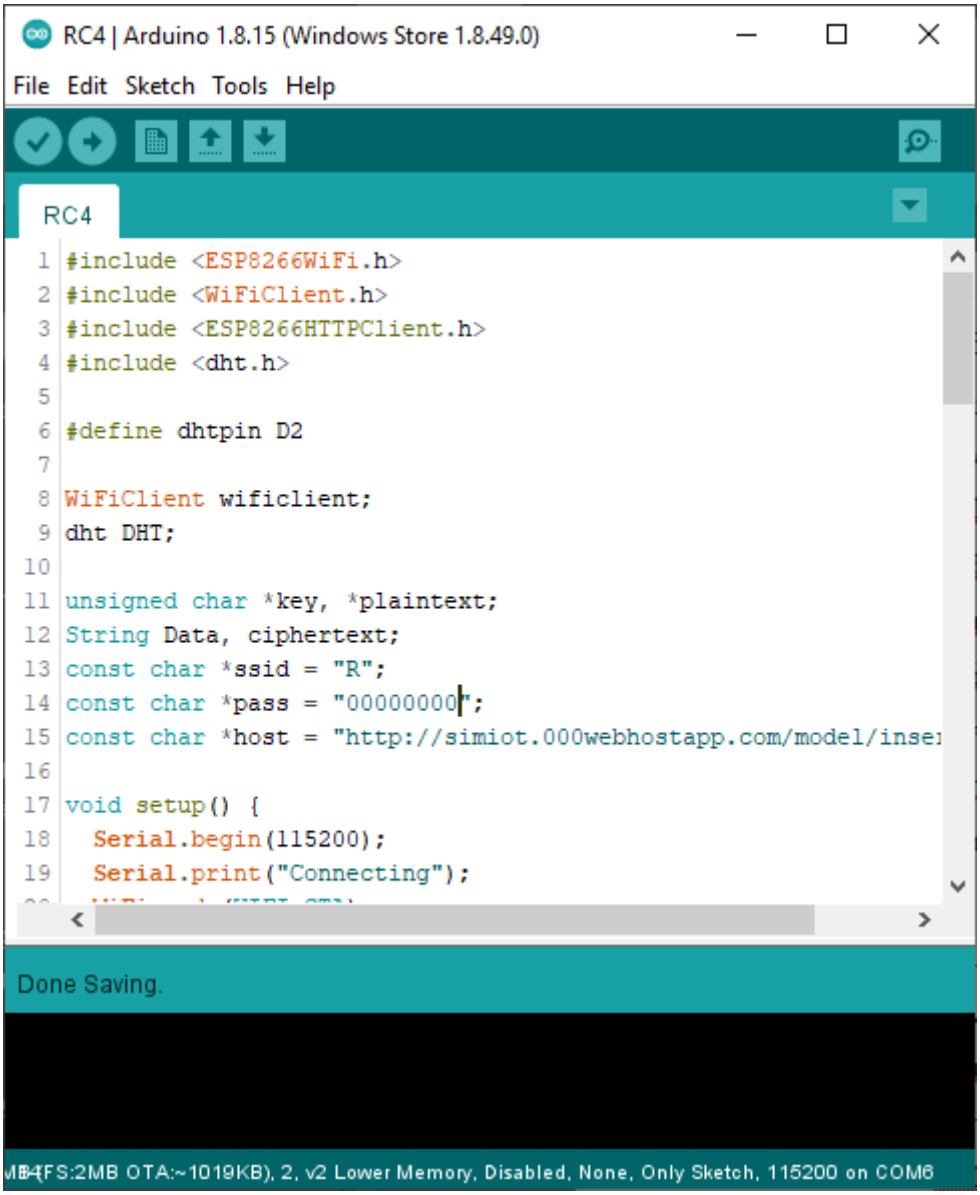
Gambar 2.5 DHT11 - Sensor Suhu dan Kelembaban

- Tegangan masukan : 5 Vdc
- Rentang temperatur : 0 – 50°C kesalahan ± 2°C
- Kelembaban : 20 – 90% RH ± 5% RH error

2.4 Perangkat Lunak Yang Digunakan

2.4.1 Arduino IDE

Software Arduino IDE adalah salah satu hal dibutuhkan dalam membuat project nodeMCU. IDE sendiri adalah singkatan dari *Integrated Development Environment* yang bisa diartikan sebagai lingkungan terintegrasi untuk melakukan pengembangan. Software untuk Arduino ini juga digunakan untuk nodeMCU memiliki bahasa pemrogramannya sendiri, yaitu bahasa C yang telah disederhanakan serta dilengkapi dengan library yang membuat pengguna jadi lebih mudah dalam melakukan pemrograman. Fungsi integrated development environment atau IDE Arduino adalah sebagai software yang digunakan untuk menuliskan, memverifikasi, men-debug, mengkompilasi, dan meng-upload program (*sketch*) dari komputer ke-*board* nodeMCU.

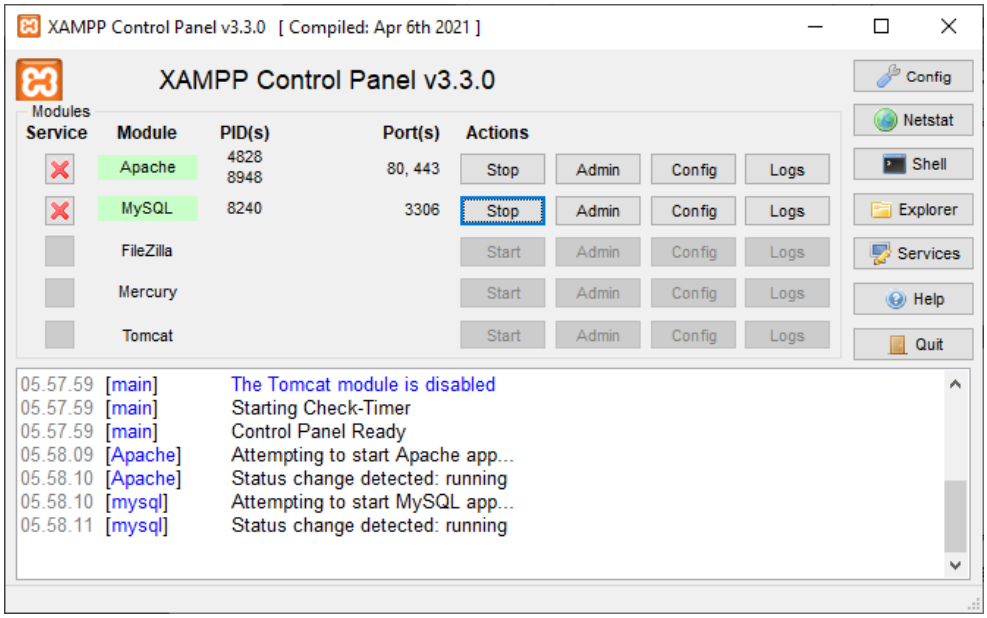


Gambar 2.6 Arduino IDE

2.4.2 XAMPP

XAMPP adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL/MariaDB, PHP, dan Perl. Sementara imbuhan huruf "X" yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti Linux, Windows, Mac OS, dan juga Solaris. Program aplikasi XAMPP berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses

pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website dalam lingkup lokal sebelum di deploy ke sistem yang sebenarnya.



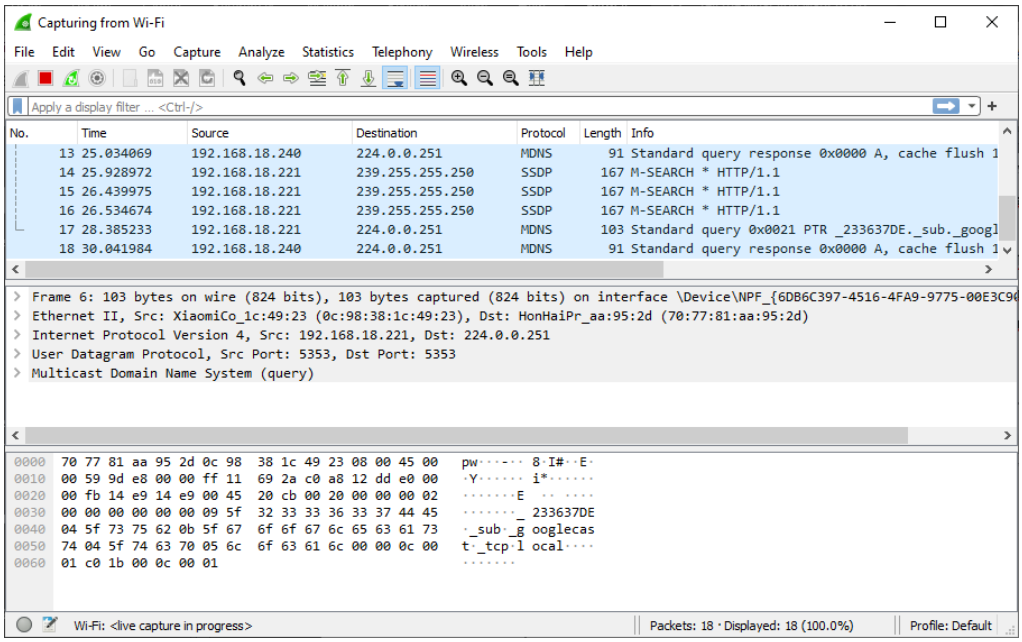
Gambar 2.7 XAMPP Control Panel

2.4.3 MySQL

MySQL adalah jenis database yang bersifat open source. Dalam pembuatan suatu aplikasi yang kompleks dan dapat dijalankan secara dinamis diperlukan database untuk menyimpan berbagai data sebagai sumber informasi. Website dan aplikasi berbasis mobile membutuhkan database server untuk menampung banyak informasi. Misalnya dalam hal URL, username, password, informasi pengguna. MySQL dapat mengatur semua jenis database sehingga dapat dikelola dengan baik. MySQL merupakan DBMS (Database Management System) dengan menggunakan perintah SQL (Structured Query Language) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL terbagi menjadi dua lisensi, yang pertama adalah Free Software dimana software tersebut dapat diakses oleh siapa saja. Dan kedua adalah Shareware dimana perangkat lunak berpemilik memiliki keterbatasan dalam penggunaannya.

2.4.4 Wireshark

Wireshark adalah tool yang ditujukan untuk penganalisaan paket data jaringan (Riadi et al., 2021). Wireshark disebut juga Network packet *analyzer* yang berfungsi menangkap paket – paket jaringan dan berusaha untuk menampilkan semua informasi dipaket tersebut sedetail mungkin. Sebenarnya *network packet analyzer* berfungsi sebagai alat untuk memeriksa apa yang sebenarnya terjadi di dalam jaringan baik kabel maupun *wireless*. Dengan adanya wireshark ini semua sangat dimudahkan dalam hal memonitoring dan menganalisa paket yang terdapat pada suatu jaringan.



Gambar 2.8 Capture Data pada Aplikasi Wireshark